The Director's Lens: An Intelligent Assistant for Virtual Cinematography

Christophe Lino, Marc Christie, Roberto Ranon, William Bares

The Director's Lens: An Intelligent Assistant for Virtual Cinematography *

Christophe Lino
IRISA/INRIA Rennes
Campus de Beaulieu
35042, Rennes Cedex, France
christophe.lino@inria.fr

Marc Christie
IRISA/INRIA Rennes
Campus de Beaulieu
35042, Rennes Cedex, France
marc.christie@irisa.fr

Roberto Ranon
HCI Lab, University of Udine
via delle Scienze 206
33100, Udine, Italy
roberto.ranon@uniud.it

William Bares
Millsaps College
1701 North State St Jackson
MS 39210
bareswh@millsaps.edu

ABSTRACT

We present the *Director's Lens*, an intelligent interactive assistant for crafting virtual cinematography using a motion-tracked hand-held device that can be aimed like a real camera. The system employs an intelligent cinematography engine that can compute, at the request of the filmmaker, a set of suitable camera placements for starting a shot. These suggestions represent semantically and cinematically distinct choices for visualizing the current narrative. In computing suggestions, the system considers established cinema conventions of continuity and composition along with the filmmaker's previous selected suggestions, and also his or her manually crafted camera compositions, by a machine learning component that adapts shot editing preferences from user-created camera edits. The result is a novel workflow based on interactive collaboration of human creativity with automated intelligence that enables efficient exploration of a wide range of cinematographic possibilities, and rapid production of computer-generated animated movies.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Animations, Video

General Terms

Algorithms, Human Factors

Keywords

Virtual Cinematography, Motion-Tracked Virtual Cameras, Virtual Camera Planning

*Area chair: Dick Bulterman

MM'11, November 28–December 1, 2011, Scottsdale, Arizona, USA.

1. INTRODUCTION

Despite the numerous advances in the tools proposed to assist the creation of computer-generated animations, the task of crafting virtual camera work and edits for a sequence of 3D animation remains a time-consuming endeavor requiring skills in cinematography and 3D animation packages. Issues faced by animators encompass: (i) creative placement and movement of the virtual camera in terms of its position, orientation, and lens angle to fulfill desired communicative goals, (ii) compliance (when appropriate) with established conventions in screen composition and viewpoint selection and (iii) compliance with continuity editing rules which guide the way successive camera placements need to be arranged in time to effectively convey a sequence of events.

Motivated in part to reduce this human workload, research in automated virtual camera control has proposed increasingly sophisticated artificial intelligence algorithms (e.g. [10, 5, 9, 17, 2, 20]) that can in real-time generate virtual camera work that mimics common textbook-style cinematic sequences. However, this combination of increased automation and decreased human input too often produces cinematography of little creative appeal or utility since it minimizes the participation of filmmakers skilled in "taking ideas, words, actions, emotional subtext, tone and all other forms of non-verbal communication and rendering them in visual terms" [6]. Additionally, existing systems rely on precoded knowledge of cinematography and provide no facility to adapt machine-computed cinematography to examples of human-created cinematography.

This paper introduces the *Director's Lens*, whose novel workflow is the first to combine the creative intelligence and skill of filmmakers with the computational power of an automated cinematography engine.

1.1 Overview

Our Director's Lens system follows a recent trend in previsualization and computer-generated movies, by including a motion-tracked hand-held device equipped with a small LCD screen that can be aimed like a real camera (see Figure 4) to move a corresponding virtual camera and create shots depicting events in a virtual environment.

In the typical production workflow, the filmmaker would film the same action from different points of view, and then

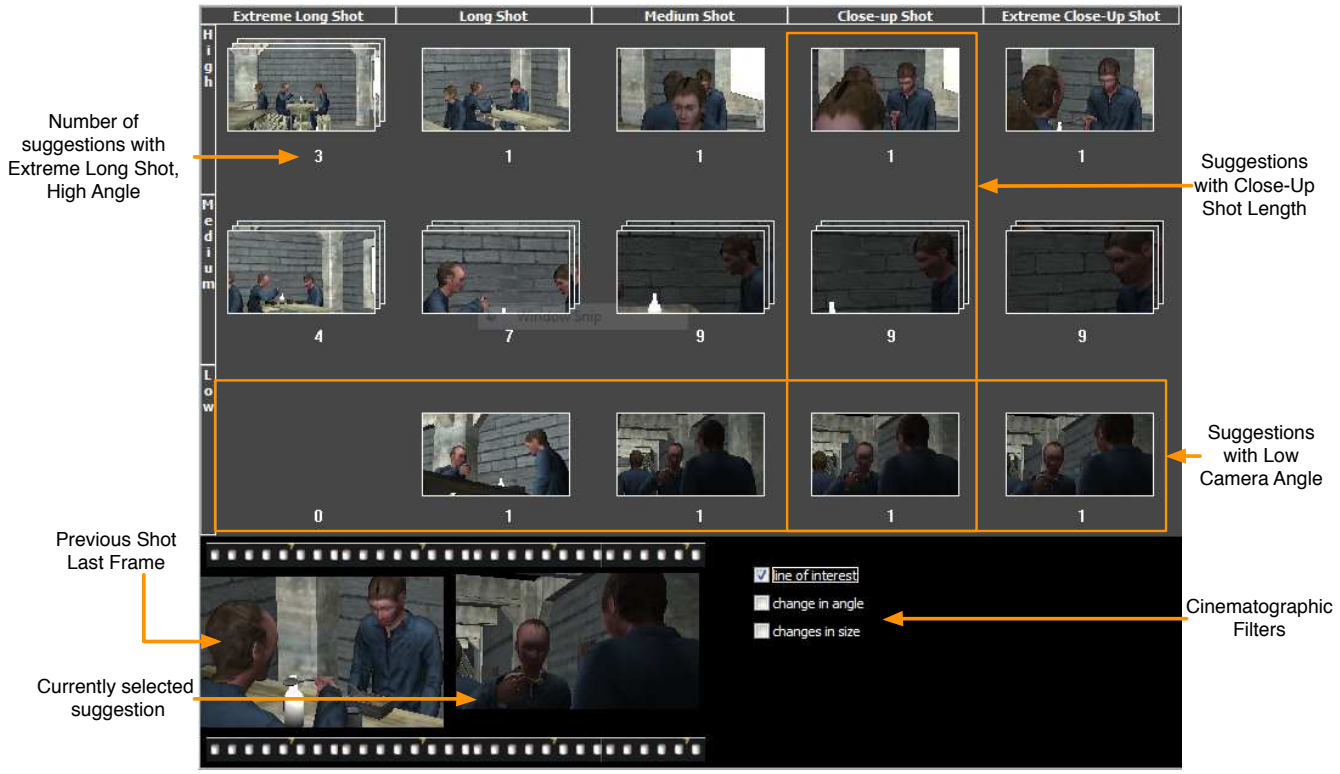

Figure 1: Screenshot of *Director's Lens* user interface in *Explore Suggestions* mode.

later select, trim and combine shots into sequences and ultimately create a finished movie. Our system instead proposes a workflow in which shooting and editing are combined into the same process. More specifically, after having filmed a shot with the motion-tracked device, the filmmaker decides where a cut should be introduced (i.e. trim the shot) and the system will compute a set of suitable camera placements for starting the subsequent shot. These suggestions represent semantically and cinematically distinct choices for visualizing the current narrative, and can keep into account consistency in cinematic continuity and style to prior shots. The filmmaker can visually browse the set of suggestions (using the interface shown in Figure 1), possibly filter them with respect to cinematographic conventions, and select the one he or she likes best.

Once a suggestion is selected, it can be refined by manually moving the virtual camera into a better position, angle, or adjusting lens zoom, and shooting can start again. In addition, movies with shots composed of still cameras can be quickly generated with just keyboard and mouse.

The system exploits an annotated screenplay which provides the narrative context in the form of text descriptions of locations, subjects, and time-stamped actions, with links to the 3D models employed in the scene. The system uses the screenplay to provides on-screen information about timing of actions and dialog while shooting and provides our cinematography engine requisite knowledge about the 3D geometry and animation being shot. To generate the suggestions, the cinematography engine leverages both its knowledge of classical cinematography rules [3], and the knowledge it gets by analyzing the compositions created by the filmmaker.

1.2 Contributions and Outline

In the emergent *Virtual Moviemaking* approach [4], real-time 3D graphics and motion-tracked virtual cameras are used in all production phases to provide creative team members with the ability of both reduce the time needed for previsualization, and to enable them to enjoy immediate and direct control of the process, from pre-production to production. For example, designed virtual sets can be interactively explored to accurately plan shots and refine creative ideas or virtual props before actual shooting takes place.

Our system aims at improving Virtual Moviemaking processes by introducing, for the first time, the combination of manual control of the virtual camera with an intelligent assistant. This provides a filmmaker with a way to interactively and visually explore, while shooting, a wide variety of possible solutions for transitioning from a shot to another, including some perhaps not considered otherwise, and instantly see how each alternate would look like without having to walk and turn around with the motion-tracked device (or move a virtual camera around with keyboard and mouse). In addition, in the innovative production workflow we propose, shooting and editing are combined in a seamless way, thus enabling very rapid production of animated storyboards or rough movies for previsualization purposes.

Furthermore, a novice filmmaker could use the system as a hands-on learning tool to quickly create camera sequences that conform to the textbook guidelines of composition and continuity in cutting from one shot to the next.

Finally, our work constitutes the first effort towards automatically learning cinematography idioms from live human camera work, and therefore building systems that are able

to learn from examples of human experts, instead of trying to pre-encode a limited number of situations.

This paper is structured as follows. In Section 2 we review related work. Section 3 describes how the system computes the suggestions, while Section 4 illustrates the Director's Lens interface layout and functionality. Section 5 discusses implementation details, limitations and results from preliminary users' feedback. Finally, Section 6 concludes the paper and outlines future work.

2. RELATED WORK

Cinematographers over time developed a rich canon of conventions for filming commonly recurring types of shots (a continuous take of camera recording) and sequences (an ordered series of shots). For example, cinematic convention suggests that in filming two characters facing one another in conversation, alternating shots of the two characters should depict one gazing left-to-right and the other right-to-left. In filming the first shot of such a sequence, the camera is placed on one side of an imaginary line-of-interest passing through the characters, and successively placing the camera on the same side of this line preserves continuity by repeating the established facing directions of the characters. [3].

In composing the visual properties of a shot, a cinematographer may vary the size of a subject in the frame or the relative angle or height between camera and subject. Shot sizes include extreme close-up, close-up, medium, long, and extreme long in which a subject's size in the frame appears progressively smaller or more distant. A filmmaker can also use editing decisions such as shot duration and the frequency of cuts to artful effect. [3].

2.1 Automated Virtual Camera Planning

The research field of automated camera planning, which combines expertise in Computer Graphics and Artificial Intelligence has sought to reduce the burden of manually placing and moving virtual cameras by bringing to bear increasingly sophisticated intelligence to emulate the composition and editing decisions found in traditional film.

Early efforts in automated camera control computed virtual camera locations from pre-programmed displacement vectors relative to the subject(s) being viewed, e.g. [12]. Attempts to model the editing transitions between shots have relied on idiom-based approaches which encode established conventions for filming common scenarios (e.g. groups of actors in conversation) using hierarchical state machines to model and transition between commonly used shot types [15]. These approaches fall short when a user needs to film scenarios not anticipated by the pre-coded state machines.

Seeking to better handle unanticipated scenarios and arbitrary user-specified viewing goals, constrained-optimization camera planners compute virtual camera placements from user-specified declarative constraints on how subjects should appear by view angle, size and location in the frame, and avoidance of occlusion. Typically, heuristic search-based or optimization techniques repeatedly generate and evaluate the quality of candidate camera shots until a satisfactory camera placement is found [10, 5]. To reduce the computational complexity, the 3D scene space can be partitioned into regions that yield semantically distinct shots by distance, view angle, and relative position of subjects [8]. Extensions to tackle camera path-planning problems have been proposed in offline [7] or online contexts [14]. The quality of the output often depends on carefully formulating the constraint parameters and importances making it difficult for users to obtain a specific desired shot.

Automatically producing virtual camera behavior coupled with narrative planning entails encoding and reasoning about coherent narrative discourse and cinematography to visualize a story. Along this line, a few systems have been proposed, e.g. CamBot [11] and Darshak [17] which can compute a full movie from a script while maintaining consistent rhetorical structure. These approaches however operate in offline contexts.

A recent proposal [20] combines all these lines of research and is able to, starting from a script of the movie, compute camera motion and edits in real-time by partitioning the 3D scene space into *Director Volumes* that combine semantic and visibility information about subjects, and then searching optimal cameras (also enforcing composition rules) inside the best volume.

In summary, existing automated virtual camera planners can in realtime generate camera placements, movements, and edits to visualize sequences of live or pre-recorded events in virtual 3D worlds. The resulting computer-generated camera work is generally adequate for mechanically reproducing cinematic sequences that conform to anticipated textbook scenarios such as groups of virtual actors in conversation. However, crafting truly artful cinematography often requires human judgement of when and how to creatively depart from textbook rules (see [22]). Unfortunately, existing automated virtual camera planners provide little or no facility for a human filmmaker to interject his or her creativity into the planner's operation.

Passos et. al. [21] also recognize the need to look beyond relying on programmers to encode cinematography knowledge in their system, which uses a neural network, to enable users to train an editing agent by selecting between a small set of virtual cameras each filming a moving car from different vantage points. Our system instead uses a much deeper model of cinematography accounting for continuity, cinematic style, and composition constraints and utilizes an intuitive motion-tracking and touch-screen interface that allows a user to craft his own compositions.

2.2 Computer-based Assistants for Movie Production

Commercial systems like FrameForge 3D [16] enable a user to create a virtual set and place cameras for producing storyboards and previsualization, but do include intelligent automated assistance to find suitable camera angles and framings or to plan camera paths.

Intelligent storyboarding research systems provide assistance in creating storyboards. For example, Longboard [18] integrates a sketch-based tablet interface to a discourse and camera planner. The user defines a script and then sketches a storyboard, from which the system produces animatics and renders scenes, possibly adding shots and actions if the user or the script did not specify them. In a form of interplay similar to that of our system, the user has the possibility of either accepting or rejecting the suggested shots by specifically adding constraints on certain frames or by adding new frames to the storyboard. However, Longboard is meant to be used as an offline generator, not as a shooting assistant.

Adams and Venkatesh [1] have proposed a mobile assistant for amateur movie makers where the user selects a narrative

template and a style, and the system displays shot directives (i.e. how to film a specific part of the event) by generating 3D simplified first-person renderings that illustrate what the camera should frame and how. The renderings are meant to assist the user while shooting in the field.

2.3 Motion-tracked Cameras for Movies

Early virtual reality researchers studied different user interaction metaphors for users to control the position and orientation of a virtual camera by moving a six-degree of freedom motion sensor [24]. This technique of animating a virtual camera using a hand-held motion sensor has found high-profile application in the previsualization (planning) and filming of feature films including *Polar Express* (2004), *Beowulf* (2007), and *Avatar* (2009). Computer game productions such as *Command and Conquer: Red Alert 3, and Resident Evil 5* have also used these devices to create in-game cinematics. Commercially-available systems (e.g., Intersense VCAM, NaturalPoint Insight VCS) couple camera-like bodies or camera mounts with 3DOF (orientation) or 6DOF (orientation and position) sensors, whose readings are mapped onto the virtual camera parameters (position, orientation) and provide buttons or levers to adjust lens zoom angle. These devices have replaced typical mouse, keyboard, and joystick camera controls with intuitive motion-sensing control, but they operate with the same complex software interfaces of 3D modeling tools, which are not designed around cinematic organizational concepts of scene, sequence, and shot. Nor can they easily manage and browse a variety of alternate cinematic visualizations side-by-side.

3. COMPUTING SUGGESTIONS

The core of our system relies on the automated generation of a large range of suggested cameras and on the ranking of these suggestions according to the last frame of the current shot.

In order to generate appropriate shots with relation to the story, our system requires the provision of an annotated screenplay. This screenplay specifies a set of actions (possibly overlapping in time) as they occur in a sequence of scenes by specifying, for each one, the starting and ending times, the involved subjects/objects (if any), a relevance value, and a textual description of the action. Actions may be attached to a character or an object (e.g. "Parsons walks to the table"), or describe any general event (e.g. "Characters are entering the canteen"). In addition, the screenplay includes the names of particular 3D model parts for the subjects, which are required to perform appropriate screen composition in computing suggestions (e.g. for a character, body part, head part and eyes parts). An excerpt of the annotated screenplay for a scene from Michael Radford's *1984* is provided below:

```
Screenplay "1984"
  Scene "Canteen"
  Location "Canteen"
   Actor Smith
     body "SmithBody"
     head "SmithHead"
     leftEye "SmithLeftEye"
     rightEye "SmithRightEye"
   Actor Syme
    ...
    Action "A pours gin"
      Relevance 5
  begin 0 end 4
     Character "Smith"
    Action "A drinks gin"
      Relevance 5
      begin 4 end 12
  Character "Smith"
    Action "A turns"
      Relevance 3
      begin 10 end 18
      Character "Smith"
    Action "A speaks to B"
      Relevance 9
      begin 18 end 21
      Character "Smith"
      Character "Syme"
    ...
end screenplay
```

As the user requests a list of suggestions (at time in the movie we denote t_s) our system performs a three-step computation:

- from the screenplay the system selects the list of all actions overlapping time t_s or occurring within a second after time t_s (this enables to suggest a shot which anticipates an action);
- for each selected action, the set of key subjects (characters, objects, buildings) is extracted and a dynamic spatial partitioning process generates a wide collection of viewpoints ensuring both a complete coverage of the key subjects and actions as well as significant enough difference between viewpoints (see detailed description in next Section). Each viewpoint represents a suggested starting point for a shot;
- each suggestion is then ranked by considering the enforcement of cinematic continuity rules between the current viewpoint and the suggested viewpoint, the quality of the composition in the shot (how elements are spatially organized on the screen), the relevance of the suggestion with relation to the action and the quality of the transition between the current shot and the suggestion.

3.1 Dynamic Spatial Partitions

We employ *Director Volumes*, a technique proposed by [20] to compute dynamic spatial partitions in the space of viewpoints around key subjects (a key subject may be a character or any object of the environment). As displayed in Figure 3, the area around some subjects configurations (a typical two-character configuration in the Figure, but one, three or more character configurations are also considered) is partitioned with relation to shot distance (Close-Up, Medium Shot, Long shot, Extreme Long Shot) and with relation to relative angle of the camera against the key subjects (Internal, External, Apex, Parallel and Subjective see [3]). For a one-subject shot this would be Front, Left, Right, Profile, 3/4 Left, 3/4 Right. These dynamic spatial partitions are based on Binary Space Partitions (BSPs) which are recomputed every time a change occurs in the 3D environment

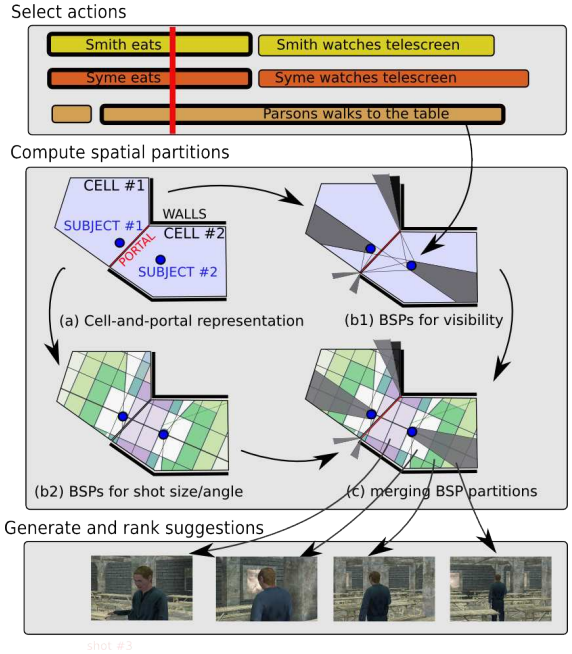

Figure 2: The automated computation of suggestions. Each selected action in the screenplay leads to the computation of a set of spatial partitions (each partitionned area represents a typical shot of the action). Suggestions are computed and ranked for each area.

(due to character or object motion). Each partition is qualified with a semantic tag representing its shot distance and relative angle to subjects.

In a parallel step, we compute a second level of dynamic spatial partitions to identify visibility of key-subjects. We analyze the 3D environment to build a visibility graph [23] (a graph which connects convex cells extracted from the environment which share a common portal, the portal representing a potential occluder). We use the graph to propagate full visibiltiy, partial visibility and no visibility for each key-subject and tag spatial partitions accordingly.

Finally, we rely on BSP-tree merging techniques to compose the information from the Director Volumes and the visibility graph into a single representation. Using this technique, all regions generated by the partitions represent viewpoints with different tags, meaning that whatever viewpoint is selected in a region, it will yield a similar result in terms of shot distance, relative angle to framed subjects and visibility.

For each region, our process generates three suggestions: a high-angle shot, a low-angle shot, and a medium angle shot, and for each of these suggestions, a default screen composition is computed by enforcing the rule of the thirds (key-subjects are spatially arranged on the screen to appear at the intersection of the line of the thirds [3]).

The overall process is described in Figure 2.

3.2 Ranking Suggestions

Once a set of suggestions is computed (for a two-subject configuration, over 240 shots are generated and for a one-subject configuration, 120 shots are generated), our system performs a quality ranking process, whose result is used when displaying the suggestions to the user (see Section 4).

The quality $q_s > 0$ of a suggestion s is defined as the product of qualities assessing specific features of s:

$$q_s = Q_{cont}(s) \cdot Q_{comp}(s) \cdot Q_r(s) \cdot Q_t(s)$$

with

$$Q_{cont}(s) = Q_{loi}(s) \cdot Q_{change}(s)$$

where Q_{cont} measures the enforcement of *continuity* rules, Q_{loi} measures compliance with *line-of-interest* rule and last, Q_{change} measures compliance with the change-in-angle-or-size rule. Q_{comp} represents the satisfaction of composition rules, Q_r represents the relevance of the suggestion with relation to the current action's relevance, and Q_t represents the quality of the transition between the current shot and the suggestion. All the $Q_i(s)$ functions return positive real values.

In the following subsection, we detail how $Q_i(s)$ functions are computed.

3.2.1 *Respecting Continuity in the Cut*

Cinematographic conventions related to continuity in cuts are well established. Such conventions typically maintain the spatial coherency in cuts (do not invert key subjects on the screen), the motion coherency (key subjects moving in the same direction on successive shots) and the composition continuity (successive shots of the same subjects will maintain similar composition). We rely on the following continuity rules to establish the ranking $Q_{cont}(s)$:

- **Line-of-interest continuity**: in actions which involve two or more key subjects, once the camera is located on one side of the line-of-interest (imaginary line linking two key subjects), the camera should not cross this line in successive shots, unless using an extreme long shot (that re-establishes the key subjects in relation to the environment). The system ranks all suggestions on the opposite side of the Line-of-interest with a low value.
- **Change in angle or size**: When considering two shots portraying the same subject, there should be at

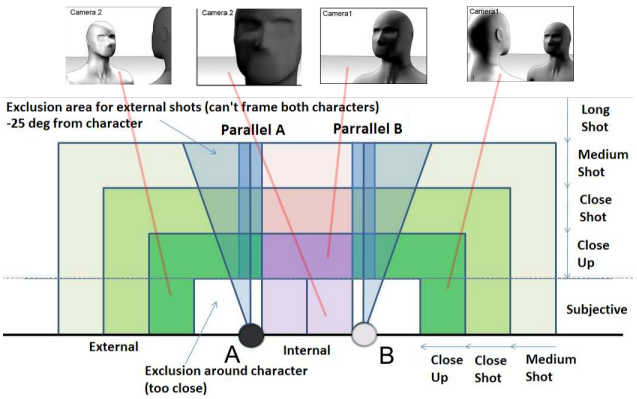

Figure 3: Spatial partitions around two subjects define areas of characteristic viewpoints referred to as Director Volumes (see [20]).

least a thirty-degree difference in orientation with relation to the key subject, or a notable difference in the size of the subject in the shot. For this purpose we compute the difference in angle and size between the current shot and the suggested shot. Only suggestions with noticeable difference in size are ranked positively (here we consider a change of at least two units in size, a unit being a step between two shots in the range of Extreme Close Shot, Close Shot, Medium Close Shot, Long Shot and Extreme Long Shot). Suggestions that subtend an angle lower than 30 degrees to the subject are ranked with a low value.

3.2.2 Respecting Classical Composition Rules

Suggestions computed by the system try to enforce the classical rule of the thirds. When considering characters, the composition is set so that their eyes (an element to which spectators look at in priority when gathering elements in a picture) are located at the intersection of two equally spaced horizontal lines and two equally spaced vertical lines on the screen. We thus measure as an Euclidean distance the difference between the ideal composition and the composition computed by the system to asses the quality of a viewpoint. Bad compositions are ranked with low values.

3.2.3 Relevance of the Shot w.r.t. Current Action

Relevance is measured by exploring the capacity of the shot to enhance a viewer's comprehension of the action. Each action has a relevance value that encodes its importance for the story (*e.g.* representing whether the action is a foreground action, an establishing action, or a background action). Shots that depict more relevant actions, from relevant viewpoint enforce the comprehension of the story and will have a higher quality.

3.2.4 Quality in Transitions Between Shots

Quality in transitions (a transition is a cut between shots) is measured by using transition matrices. A transition matrix is a stochastic matrix used to describe the transitions of a Markov chain. We encode our transitions with a right stochastic matrix, i.e a square matrix where each row consists of nonnegative real numbers summing to 1. Each value t_{ij} of the matrix T (i being the row index, and j the column index) corresponds to the probability of performing a cut from Director Volume i to Director volume j. We use three different matrices depending on whether the transition is performed inside the same action, between related actions, or between unrelated actions. The quality of the transition is given by an affine function $y = ax + b$, where $a > 0, b > 0$ and x is equal to the value t_{ij}^k related to the transition in the corresponding matrix T_k. The values in the matrices therefore represent user preferences in performing cuts, and in our approach, the values are updated by a learning process described in Section 3.3.

3.3 Learning from the User Inputs

We rely on a simple reinforcement learning technique to update the probabilities in the transition matrices, using the cuts already performed by the user. Three distinct transition matrices are encoded depending on whether the transition is performed:

1. **during the same action (matrix T_A)**: the two consecutive shots are conveying the same action;
2. **between related actions (matrix T_R)**: the two consecutive shots are conveying two different actions, but the actions have a causal link (e.g. in case of dialog, the first action being "Syme speaks Smith" and the second one "Smith answers to Syme" for instance). A causal link is established when the two actions share the same key subjects;
3. **between unrelated actions (matrix T_U)**: the two consecutive shots are conveying two different actions, and there is no causal link between them;

These transition matrices actually define preferences in using some transitions between shots over others. A shot is identified by its type (orientation to subject) and its size (extreme long to extreme close-up). Our system learns the transition matrices from the user inputs, by analyzing the successive choices in shot types performed by the user. The learning process operates as follows. Each time the user selects a suggestion as the new shot, we first determine the transition matrix T_k to consider by analysing whether (1) the successive conveyed actions are the same, (2) the actions are different but are causally linked, or (3) the actions are different and they have no causal link. Then all the values t_{ij}^k of the row i of the corresponding matrix T_k (where i is the shot type of the current shot), are updated in the following way: let the newly selected shot be of type n, the value t_{in}^k will be increased, and the values $t_{ij}^k, j \neq n$ of row i will be accordingly updated such that $\sum_i t_{ij} = 1$.

The probabilities in the matrices influence the quality of a shot by ranking preferred cuts higher (the quality of transition $Q_t(s)$ is expressed as a function of t_{ij}^k).

4. THE DIRECTOR'S LENS SYSTEM

In this Section, we describe the interface and user interactions with the Director's Lens system. To demonstrate the system, we will use as example a reconstruction from a sequence of Michael Radford's *1984* movie, together with a set of 70 actions which describe in details the events occurring over the scene. The scene is composed of 5 key subjects (Syme, Parsons, Julia, Smith and an anonymous member of the party).

4.1 User Input Devices

The Director's Lens system features a 7-inch HD LCD touch-screen mounted to a custom-built dual handgrip rig (see Figure 4) that can be paired with either a 3DOF or 6DOF motion sensor. When used with a 3DOF sensor (rotation controls camera aim direction), the two thumb sticks control camera position, and buttons adjust lens zoom, movement/zoom speed, and record/playback functions. When paired with an optical 6DOF tracking system, a rigid cluster of reflective markers is fixed to the device and one of the thumb sticks is configured to increase or decrease lens zoom angle. The spatial configuration (position/orientation) read by the tracking sensor is synchronized with the spatial configuration of the virtual camera in the system, so that (besides a distance scaling factor that can be adjusted) movements of the motion-tracked device will correspond to analogous movements of the virtual camera in the 3D scene.

The system can also be used with just a desktop or notebook computer and a mouse. In this mode, the preferred workflow is primarily one of clicking to select a suggested

Figure 4: Our hand-held virtual camera device with custom-built dual handgrip rig and button controls, a 7-inch LCD touch-screen.

camera placement for each shot to very rapidly compose a virtual 3d movie made with shots featuring static cameras.

4.2 User Interface and Interaction

The user's interaction with Director's Lens is divided into two interface modes: (i) *explore suggestions* (Figure 1) and (ii) *camera control/movie playing* (Figure 5).

4.2.1 Explore Suggestions Interface

The *explore suggestions* interface (Figure 1) displays the suggestions for beginning a new shot from any instant in the recorded movie. The suggestions are presented as small movie frames, arranged in a grid whose rows and columns correspond to visual composition properties of the suggested cameras. More specifically, the horizontal axis from left-to-right varies by decreasing shot length (or distance) in order of extreme long, long, medium, close-up, and extreme close-up. The vertical axis from top-to-bottom presents suggestions from a variety of camera heights in order of high, medium, and low. For example, Figure 1 shows the suggestions that are proposed to transition from a shot depicting the actions "Syme eats" to a new shot where two parallel actions occur: "Syme eats" (which continues from the previous shot) and "Smith speaks to Syme" (the last frame of the previous shot is shown in the reel at the bottom left). Notice, for example, that all suggestions in the top row are viewed by cameras positioned above the virtual actors.

When more than one suggestion is generated for each shot length and camera angle, the system displays a stack of frames in that grid position, where the top one is the most highly ranked (of all the suggestions in the stack). If the user wishes to explore a stack (s)he can expand the actually displayed suggestions for a grid column, row, or cell, by clicking on the corresponding column/row heading. For example, Figure 5 shows the suggestions grid expanded to display all suggestions for medium length / medium angle shots.

When the suggestions do not refer to the first instant in the movie (i.e., there are previous shots), the system allows the user to visually filter the presented suggestions on the basis of respect of cinematography rules, namely the line of interest, and minimum change in angle or size with respect to the previous shot's last frame. For example, in Figure 1 we have selected to display only suggestions that respect the line of interest rule (i.e., the camera is on the same side of the line of interest as in the previous shot last frame), while Figure 5 shows just suggestions where the camera is at least 30 ° difference in orientation with relation to the key subject (with respect to the previous shot of the last frame).

The user selects a suggested shot by touching its icon in the grid. The larger image framed in the reel at the bottom left represents the previous shot to assist the user in choosing or modifying a suggestion to best follow its predecessor, and shows the currently selected suggestion to its right. The user finally confirms his or her choice and goes to the *camera control/movie playing* interface.

4.2.2 Camera Control / Movie Playing Interface

The *camera control/movie playing* screen (see Figure 6) allows the filmmaker to view the recorded movie or manually take control of the camera to record a shot. The interface features a single large viewport to display the 3D scene as viewed from a virtual camera corresponding to the filmmaker's choice for the current instant in the movie or corresponding to the position/orientation of the motion-tracked device (when the tracking of the device is enabled).

When camera tracking is not enabled, the *Play* button allows one to play-pause the animation, and visualize it from the currently recorded shots, while *Previous Shot* and *Next Shot* buttons enables to move in the recorded movie by shot. With these functionalities, the user can review the recorded movie, and decide where to introduce a cut - this is implicitly introduced when one asks the system to explore suggestions.

When camera tracking is enabled, the virtual camera is driven by the motion tracking and lens zoom controls of the hand-held device. The tracking starts from the currently used camera, so that, if the user is coming from the *explore suggestions* interface after having selected a suggestion, tracking starts from the virtual camera in the configuration associated with that suggestion. By using the *Turn on Recording* button, the user starts/Stops recording of the stream of virtual camera position, orientation, and lens angle input data. Starting recording also starts playback of the pre-produced animated character action and digitized audio dialog.

A graphic overlay can appear over the camera image to display an animated timeline of the unfolding character actions and dialog. Horizontal bars along each row represent one character's actions or dialog. The length of each bar represents the duration of that event. Bars scroll by as the animation plays to indicate the passage of time. This allows the filmmaker to move the camera at the precise time in anticipation of an upcoming screenplay action or spoken dialog event. The overlay also displays the framing properties (e.g. medium, low, etc.) of the current camera view.

Figure 7 demonstrates an example in which the system suggests shots which match the composition of the user's previous shot. In this scene, Smith pours a bottle while seated facing Syme. The user has composed a shot (shown in the right image in the Figure) in which Smith fills the rightmost third of the frame and gazes to the left and the camera is to his lefthand side. Consequently, when we click to enable the line of interest filter, the system generates suggestions in which no viewpoints violate the line of interest by placing the camera on Smith's righthand side. Furthermore, all shots display Smith on the rightmost third of the

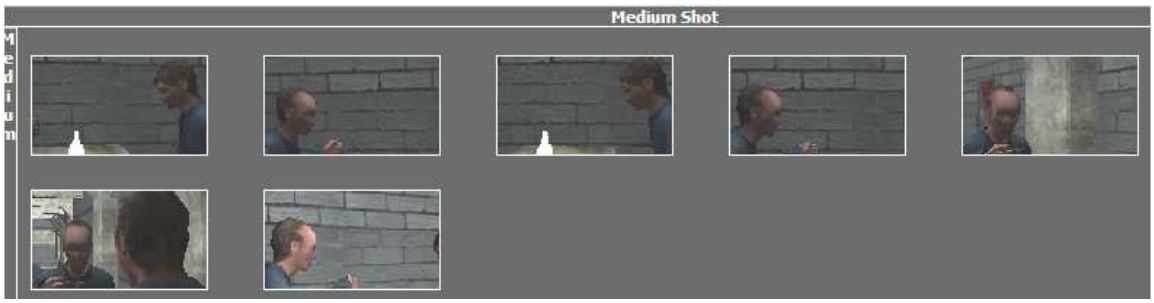

Figure 5: Screenshot of the interface in Explore Suggestions mode with expansion of medium shots and medium camera angles that were visualized in Figure 1, with additional enabling of the *Change in Angle* visual filter. Compare the resulting suggestions with the last frame of the previous shot that is shown in the bottom left part of Figure 1.

Figure 6: Screenshot of the interface in Camera Control / Movie Playing mode. The overlay horizontal bars show an animated timeline of screenplay events and their duration, together with an indication of the camera angle and framing that is currently being used in the movie.

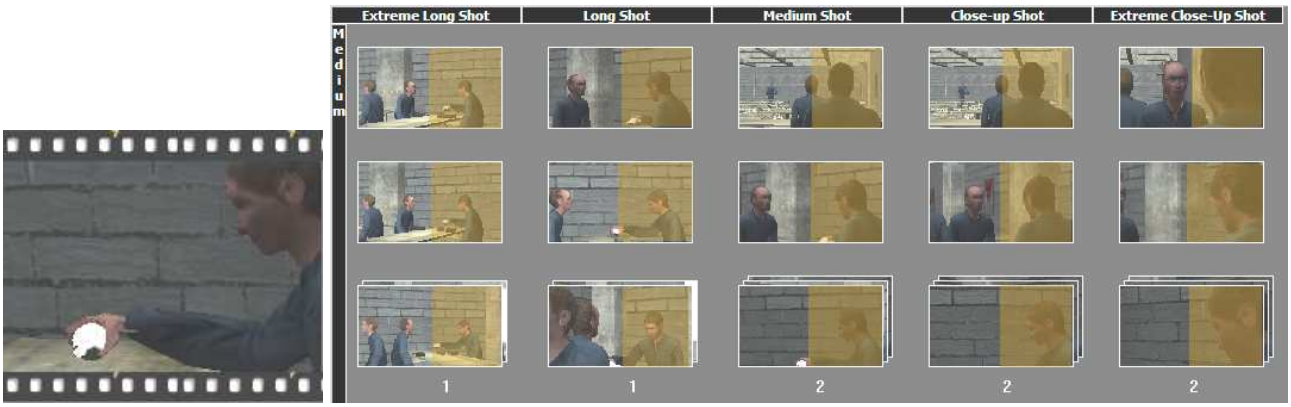

Figure 7: Given user's composition for the current shot (shown in the left image), the system suggests shots (right image) which satisfy continuity rules of editing.

frame. Also note suggested compositions leave more room ahead of Smith's face, as did the user's composition.

5. RESULTS

Our system is coded in C++ and uses the OGRE real-time rendering engine, and a custom-written abstraction layer to interface to a variety of button and joystick input devices and motion trackers. In our tests we used 6DOF rigid body reflective markers with an A.R.T. optical tracker with 16 cameras covering a workspace of 4×8 meters. The application runs on a computer whose video output is routed over an HDMI cable to the small LCD touch screen. The largest computational cost of the cinematography engine is in finding visibility and director's volumes, which depend on the geometric complexity of the scene and the variations in position of the key subjects. In the pictured example, the computation of suggestions takes around a second on a Intel Core i7 2.13 GHz notebook, which is acceptable since this computation is not required to happen in real-time.

5.1 Applicability to Other Scenarios

The examples shown in the paper features mostly dialog actions involving two or three characters, and also generic actions (e.g. "Parsons walks to the table") for which proper suggestions are computed. In general, the system can model any event where one character or object, or group of objects (*e.g.*, cars driving, architectural elements) is involved, and propose suggestions to properly frame it using different angles and shot lengths. Note that for many situations (e.g. shooting of architecture, or crowds) there are not many cinematographic conventions other than using establishing shots first, and then closer shots. However, in general, the cinematography knowledge the system uses can be expanded by designing additional Director Volumes for specific cases or using the automatically learned tables of transitions. For situations that involve, in the same instant, more than three characters, the system could be extended by employing the method of hierarchical lines of action [19].

Scenes involving very fast action can be a bit problematic as the system might generate suggestions that are good for the considered instant, and decrease in quality after a short delay, for example because the framed character moves aways or is covered by some moving objects. This kind of situation would require the system either to consider animations occurring in an interval of time, instead of an instant, or to suggest camera movements, instead of just starting frames. However, this would be of much greater computational cost than deriving suggestions for single instants. In these cases, however, the filmmaker can at worst ignore the suggestions and deal with the situation by taking manual control.

Finally, the system computes all suggestions using a single fixed lens field of view. However, the user can quickly adjust the field of view by manual control. The system's search algorithm can be extended by taking into account variations on lens angle. In practice, the set of lens angles to consider would be limited to model the cinematography practice of using a small number of user-selected "prime" or standard lenses.

5.2 User Feedback

While we do not currently have results from formal and extensive user evaluations, a few videographers have tried the system, reporting great satisfaction in using it.

In general, users found the system workflow refreshing compared to mouse-based solutions, were very willing to try it, and could rapidly use it to produce results. In our tests, after a very brief introduction to the system, users familiar with camerawork could shoot a 3 minutes video of the 1984 scene in around 10-15 minutes, including taking decisions on camera angles and cuts. For comparison, producing a video for same animation sequence took an expert user a few hours of work using 3DS Max. The reason for the huge difference in time comes from the fact that with typical animation software tools one is forced to a workflow where cameras have to be manually positioned and animated, and then the shot can be watched, and often one has to go back and forth between cameras modifications and testing the result. Furthermore, there's the complexity of manually keeping track of timings and motion curves, making sure that edits occur in the correct place and time.

While the use of a motion-tracked virtual camera is surely pivotal in reducing the amount of time needed, we can hypothesize that suggestions also played a role, as in general users needed to make very small movements to refine the suggestions and find their ideal starting camera positions. Moreover, from our feedback with professionals in the development of motion-tracked virtual cameras, we know that having to walk around in the tracked space to explore angles and distances is perceived as a real issue. Finally, users more experienced in real cameras work appreciated on-screen information overlays of character actions and their timing while shooting.

The interface for browsing suggestions by shot distance and height was considered easy to learn and effective for a videographer, and the possibility of quickly visualizing many alternatives was very appreciated, though not all users agreed with each suggestion. This is expected since the perceived quality of a suggestion is a subjective factor, which changes from one videographer to another. However, users' attitude towards the cinematography engine work was positive, mainly because the the system does not force one to any shooting or editing style, and suggestions can be ignored: however, many times they caused the videographer to consider solutions he had not thought about, thus enhancing creative thinking and exploration.

6. CLOSING REMARKS

While a few experiments have been done in combining human expertise with automatic camera planners, we believe that this work is the first attempt at innovating the shooting process of CG animations by providing intelligent and adaptive automatic support in creating shots. In the shooting of CG movies, videogame cinematic scenes and machinima, the system has the potential of making the process much easier and quicker than current methods. From our early tests, unlike existing automated systems which rely on pre-coded cinematic knowledge to cover all anticipated scenarios, our interactive approach proves effective even if no suggested viewpoint is "perfect" in the user's creative judgement.

Future work includes conducting extensive user evaluations to precisely assess the user satisfaction of the various system features, and measure task completion time with and without the automated assistance capability.

We plan also to focus on methods to improve the recorded

camera tracks. For example, one could think of a "steadycam" filter to avoid using a tripod for non-handheld camera style, or apply post-processing algorithms to recorded video to improve apparent camera movement as proposed by Gleicher and Liu [13].

Motion-sensing virtual cameras offer an opportunity to easily capture shots in the context of screenplay actions providing a rich corpus for machine learning. Further work in this novel direction could eliminate the re-programming needed to encode knowledge for new numbers and arrangements of actors, shots, and transitions, and also provide an effective way to incorporate more sophisticated stylistic capabilities into automatic camera planners.

6.1 Acknowledgements

Thanks to Dean Wormell of InterSense for discussions on the use of motion-sensing virtual cameras in movie productions. Thanks to Nick Buckner, Tyler Smith, Alex Olinger, and Lihuang Zhu for constructing our virtual camera device. William Bares acknowledges support of a Millsaps College sabbatical award. Roberto Ranon acknowledges the financial support of the PRIN project *Descriptive Geometry and Digital Representation: intelligent user interfaces to support modeling and navigation in 3D graphics application for architecture and design*. This work has been funded in part by the European Commission under grant agreement IRIS (FP7-ICT-231824).

7. REFERENCES

[1] B. Adams and S. Venkatesh. Director in your pocket: holistic help for the hapless home videographer. In *Proceedings of the 12th ACM International Conference on Multimedia*, pages 460–463. ACM Press, 2004.

[2] D. Amerson, S. Kime, and R. M. Young. Real-time cinematic camera control for interactive narratives. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 369–369. ACM Press, 2005.

[3] D. Arijon. *Grammar of the Film Language*. Hastings House Publishers, 1976.

[4] Autodesk. The new art of virtual moviemaking. Autodesk whitepaper, 2009.

[5] W. Bares, S. McDermott, C. Boudreaux, and S. Thainimit. Virtual 3D camera composition from frame constraints. In *Proceedings of the 8th ACM international conference on Multimedia*, pages 177–186. ACM Press, 2000.

[6] B. Brown. *Cinematography: Theory and Practice: Image Making for Cinematographers, Directors, and Videographers*. Focal Press, 1st edition, 2002.

[7] M. Christie, E. Languénou, and L. Granvilliers. Modeling camera control with constrained hypertubes. In *Proceedings of the Constraint Programming Conference (CP 2002)*, pages 618–632. Springer-Verlag, 2002.

[8] M. Christie and J.-M. Normand. A semantic space partitioning approach to virtual camera control. In *Proceedings of the 2005 Eurographics Conference*, pages 247–256. Blackwell Publishing, 2005.

[9] N. Courty, F. Lamarche, S. Donikian, and E. Marchand. A cinematography system for virtual storytelling. In *Proceedings of the 2003 International Conference on Virtual Storytelling*, volume 2897, pages 30–34, November 2003.

[10] S. M. Drucker and D. Zeltzer. Camdroid: a system for implementing intelligent camera control. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics*, pages 139–144. ACM Press, 1995.

[11] D. Elson and M. Riedl. A lightweight intelligent virtual cinematography system for machinima generation. In *Proceedings of the 3rd Conference on AI for Interactive Entertainment (AIIDE)*, 2007.

[12] S. Feiner. Apex: An experiment in the automated creation of pictorial explanations. *IEEE Computer Graphics & Applications*, pages 29–37, 1985.

[13] M. L. Gleicher and F. Liu. Re-cinematography: improving the camera dynamics of casual video. In *Proceedings of the 15th international conference on Multimedia*, pages 27–36. ACM Press, 2007.

[14] N. Halper, R. Helbing, and T. Strothotte. A camera engine for computer games: managing the trade-off between constraint satisfaction and frame coherence. In *Proceedings of the 2001 Eurographics Conference*, pages 174–183. Blackwell Publishing, 2001.

[15] L. He, M. F. Cohen, and D. H. Salesin. The virtual cinematographer: a paradigm for automatic real-time camera control and directing. In *Proceedings of the 23rd ACM conference on Computer graphics and interactive techniques (SIGGRAPH '96)*, pages 217–224. ACM Press, 1996.

[16] Innoventive-Software. Frameforge3d, 2007.

[17] A. Jhala. *Cinematic discourse generation*. PhD thesis, 2009.

[18] A. Jhala, C. Rawls, S. Munilla, and R. M. Young. Longboard: A sketch based intelligent storyboarding tool for creating machinima. In *Proceedings of the 2008 FLAIRS Conference*, pages 386–390, 2008.

[19] K. Kardan and H. Casanova. Virtual cinematography of group scenes using hierarchical lines of actions. In *Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*, pages 171–178. ACM Press, 2008.

[20] C. Lino, M. Christie, F. Lamarche, G. Schofield, and P. Olivier. A real-time cinematography system for interactive 3D environments. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 139–148. ACM Press, 2010.

[21] E. B. Passos, A. Montenegro, E. W. G. Clua, C. Pozzer, and V. Azevedo. Neuronal editor agent for scene cutting in game cinematography. *Comput. Entertain.*, 7:57:1–57:17, 2010.

[22] B. Peterson. *Learning to See Creatively*. Watson-Guptill, 1988.

[23] S. J. Teller and C. H. Sequin. Visibility preprocessing for interactive walkthroughs. In ACM, editor, *Proceedings of the 18th ACM annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 61–69, 1991.

[24] C. Ware and S. Osborne. Exploration and virtual camera control in virtual three dimensional environments. *Proceedings of the 1990 symposium on Interactive 3D graphics (SI3D 90)*, pages 175–183, 1990.