



HAL
open science

A Real-time Cinematography System for 3D Environments

Christophe Lino, Marc Christie, Fabrice Lamarche, Patrick Olivier

► **To cite this version:**

Christophe Lino, Marc Christie, Fabrice Lamarche, Patrick Olivier. A Real-time Cinematography System for 3D Environments. 22es Journées de l'Association Française d'Informatique Graphique, Nov 2009, Arles, France. pp.213-220. hal-00572963

HAL Id: hal-00572963

<https://hal.science/hal-00572963>

Submitted on 2 Mar 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Real-time Cinematography System for 3D Environments

Christophe Lino¹, Marc Christie¹, Fabrice Lamarche¹ and Patrick Olivier²

¹INRIA, Université de Rennes

²Newcastle University

Abstract

We propose an automated system that constructs a movie in real-time from a sequence of low-level narrative elements (eg. character actions and motions, and object motions). The system computes appropriate viewpoints on these elements, and performs cuts following user-defined cinematic conventions and styles. Viewpoints are selected within specific regions of the environment (the Director Volumes) using the intersections of Visibility Volumes and Semantic Volumes (viewpoint partitions defined around elements of interest according to characteristic cinematographic views). We then reason over Director Volumes to choose where to position the camera, and when to perform cuts given cinematographic continuity rules. Visibility issues due to dynamic occluders are handled by a local visibility sampling technique based on hardware rendering. In building annotated viewpoint partitions that handle visibility, we propose both high-level representations and a means to reason over the editing process. This represents a novel and expressive approach that couples camera planning and editing, which stands in contrast to existing techniques that are mostly procedural and do not encounter for dynamic visibility in editing.

Nous proposons une méthode temps-réel d'automatisation de la construction d'un film à partir d'une liste d'éléments bas-niveau (eg. actions/mouvements des personnages, mouvement des objets). Notre système calcule des points de vue appropriés sur ces éléments, et effectue des coupures de montage selon des conventions cinématographiques définies par l'utilisateur et le style. Les points de vue sont sélectionnés dans des régions spécifiques de l'environnement (les volumes directeurs) en utilisant l'intersection de volumes de visibilité et de volumes sémantiques (partitions des points de vue définies autour des personnages principaux selon des placements caractéristiques de la cinématographie). Nous raisonnons ensuite sur ces volumes directeurs pour choisir où positionner la caméra, et quand effectuer des coupures étant donné des règles de continuité cinématographiques. Les problèmes de visibilité dû à des occultants dynamiques sont gérés par un échantillonnage local basé sur des rendus. En construisant une partition en points de vue annotés qui gèrent la visibilité, nous proposons une représentation haut-niveau et un moyen de raisonner sur le processus de montage. Cela représente une approche originale et expressive qui associe la planification du positionnement et des sauts de la caméra, qui contraste avec les techniques existantes qui sont pour la plupart procédurales et ne prennent pas en compte la visibilité dynamique dans le montage.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques. Interaction Techniques

1. Introduction

The provision of fully automated camera control for interactive environments, including viewpoint computation, viewpoint planning and editing, is a complex problem that raises three important issues.

1. Such a system needs to be underpinned by a narrative

model that both structures the way the actions are organized, and allows us to reason as to their relative importance and presentation. Following Young's bipartite model [You07], interactive narratives can be considered to have two levels: (1) the story, the chronologic account of events over time; and (2) the discourse, the way in which events are organized and presented to the spectator.

2. How can we encode cinematic conventions and relate these to the narrative goals of the application? In relation to camera control, these conventions are often expressed as a set of idioms, that is, camera configurations that can be assembled to bring a story to the screen in a way that allows the viewer to understand the spatial relations/context of each scene, and the sequence of actions occurring within it. For example, such idioms would describe conventional camera placements and movements for variety of settings involving two or more characters in dialogue. An automated camera control framework must be capable of characterizing these declaratively through the use of constraints and objectives that can be solved in real-time.
3. How to develop these efficient solution mechanisms which must address the innate complexity of well understood problems such as visibility determination and path planning? For example, the simultaneous computation of the visibility of multiple targets in a dynamic environment requires a pragmatic solution which will provide some guarantee as to the efficacy of results. Facilitating *jump-cut* continuity editing, where a user's view switches between two different viewpoints, requires the ability to reason more than just to local visibility of the current camera. Similarly, selecting appropriate *tracking* shots involves planning a camera's motion in a dynamic environment whilst simultaneously taking account of the visibility of scene elements from viewpoints along the spatio-temporal path.

We propose a fully automated system that addresses two issues, the construction of a cinematically expressive movie in real-time from a specified sequence of low-level narrative elements (e.g. character actions and motions, and object motions). Our system computes appropriate viewpoints on these elements, selected within specific regions of the environment referred to as *Director Volumes*) that capture both visibility and shot information pertaining to the scene. We then reason over Director Volumes to choose where to position the camera, how to plan paths between viewpoints, and when to perform cuts following user-defined cinematic conventions, given cinematographic continuity rules and styles. Our system constitutes a novel and significant step towards the expressive coupling of camera planning and editing in dynamic 3D graphical environments.

2. The Elements of Automated Cinematography

Camera control in computer graphics has been received significant attention in recent year. Contributions span a range of subproblems, from viewpoint computation and the specification of shot properties, to camera motion control and the planning visually felicitous paths. To contextualize our formulation automated cinematography we considered a number of significant contributions across the broad spectrum of problems to be solved, from viewpoint computation and camera motion planning, to editing.

Viewpoint Computation The ability to compute viewpoints with specifiable properties is of particular value in a range of applications, including visual servoing, medical and scientific visualization, and image-based rendering. For such applications, viewpoints typically maximize the visibility of salient features and highlight the spatial relations between object with a view to enhancing a viewer's perception and understanding of a scene. Vasquez *et al.* [VFSH03] proposed that a good view should maximize the amount of information about a scene for a viewer. Drawing on Information Theory they use the notion of viewpoint entropy to compute a minimal set of N good views of a 3D environment (*i.e.* a set that maximizes viewpoint entropy).

Others have considered the problem of viewpoint selection as one of information composition. Bares *et al.* [BK01] automatically generated camera shots using the composition heuristics of expert photographers. Camera constraints are specified using storyboard frames to indicate how a desired shot should appear. The 3D environment is then analyzed to determine camera parameter setting leading to a shot closely matching the specification. Christie *et al.* [CN05] proposed a partitioning of space, into semantic volumes, which capture the regions around scene elements for which a camera will have a consistent set of visual properties (*i.e.* qualitatively equivalent shots). However, in general approaches to viewpoint computation approach the problem without considering the temporal evolution of a scene.

Motion Planning Existing methods for camera motion planning augment classical path planning approaches with features such as path continuity and visibility along the path. Planning techniques can be divided into two main classes: local approaches, that constrain the search to a reduced set of candidate solutions, and global approaches, which require the construction of, and search over, an a priori representation of the whole environment. In terms of local approaches, techniques based on potential fields has been proposed to plan camera's position in dynamic environments [Bec02]; the remainder of DOF being set by fixing the field of view and pointing the camera at an object of interest. Image based visual servoing has also been used to compute camera's DOF based on constraints expressed in the camera's field of view [CM01]. Li *et al.* proposed to use a lazy PRM attached to the object of interest (e.g. a virtual character) to reactively plan suitable camera paths [LC08]. Although such local approaches are responsive to local changes and dynamic elements of the scene (*i.e.* occluders) they typically fail to produce camera paths with properties that account for the global properties of a scene. Integrating global knowledge of the environment significantly improves the quality of the path but generally relies on an offline analysis of static environments. For instance, [NO03] used a pre-computed PRM constructed over a complete environment in the automatic generation of camera movements. Within static environments, [AVF04] combined global planning, a search through a dis-

cretization (cells) of an environment, with a secondary local search of each cell. Recently, Oskam *et al.* [OSTG09] presented a single target tracking real-time system that uses a global visibility-aware roadmap together with an estimate of the pair-wise visibility between portions of the scene, and which adapts at run-time to dynamic occluders. Such global planning methods enable large-scale camera transitions and can be combined with local search to follow a target and avoid obstructed viewpoints.

Editing Incorporating the editing process (where and when to cut between viewpoints) in computer graphics applications has been relatively under-addressed [HCS96] [CAWH*96] [CLDM03] [FF04]. Typically this is realized as set of idioms (series of reference viewpoints linked to a given configuration of actors) using a procedural language (usually a finite state machine): when a given action occurs, apply the associated viewpoint transition. Such approaches fail to recognize the importance of narrative discourse level [You07]. That is, story elements (plot and character) are defined in terms of plans that drive the dynamics of a virtual environment, and discourse elements (the narrative’s communicative actions) are defined in terms of discourse plans whose communicative goals include conveying the story world plan’s structure. Integration of these narrative aspects has only recently been considered [ER07] [Jha09]. However, the realization has again been procedural in character, a script (*i.e.* a series of actions) is taken as input and an automatic visualization of the scenario *w.r.t.* procedural cinematic rules is produced.

The procedural application of idioms often leads to highly restrictive editing rules and cannot furnish the expressive range of real world cinematography. Instead, series of idioms are deterministically applied to a story, and the idiom application architecture (*e.g.* finite state machines) makes idiom composition impossible when two actions are unfolding in parallel. Crucially, such approaches do not address low-level issues, such as visibility, that must be simultaneously accounted for. Indeed, the fact that no representation of an environment takes account of the exact visibility in the whole environment is a major issue in automated camera control. Therefore, automated cinematography must incorporate a high-level editing model, in which rules (to specify spatio-temporal context) and idiom preferences (*i.e.* style) are non-deterministically applied, and incorporate an integrated facility to reason as to visibility in camera movement and cut planning.

3. Overview

We propose a real-time approach to virtual camera control and editing that directly addresses the visibility of key-subjects for both static and dynamic occluders, and provides an expressive means to reason over idioms, shots and shot transitions. An overview of this process is given (fig. 1).

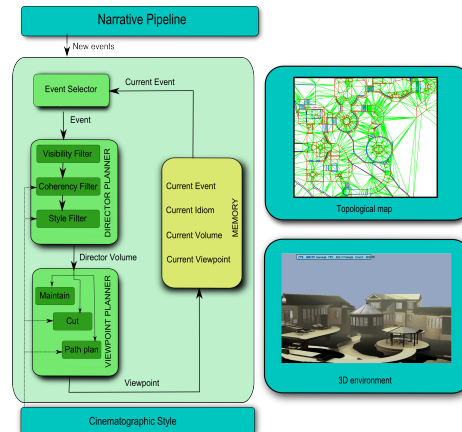


Figure 1: Overview of the camera control process: building and reasoning over occlusion-free Director Volumes.

Our system takes as input a 3D interactive environment (*i.e.* changes are not known beforehand) and a narrative pipeline that injects events in real-time. Each class of event has an associated set of viewpoints (Director Volumes) that convey the event according to established cinematographic conventions (*i.e.* idioms). At runtime, a viewpoint planner first selects the most relevant event and chooses the best idiom to convey this event by filtering the Director Volumes according to the visibility of key-subjects, thereby enforcing coherency in the sequence of viewpoints, and eliciting preferred volumes. A transition planner then determines appropriate viewpoint transitions, either by using a jump cut or a path-planning process between viewpoints.

4. Computing Director Volumes

As already described (section 2) real-time visibility computation is a prohibitively expensive process. We address this constraint by reducing the practical complexity through two assumptions as to the geometric properties of the environment. Firstly, we use the term *actor* to designate a key subject of an action, free to move in the environment and we estimate its visibility using 2D convex hull of the projection of its geometry onto the floor. Secondly, we restrict the setting to static 3D indoor environments which we represent as a 2D cell-and-portal structure [TS91]. Cells represent the rooms and portals represent doorways and connections between cells. Hard visibility constraints exist in relation to extremities of the portals (*e.g.* walls or doors). Each boundary between a wall and a portal supports a stabbing line that represents a separation between visibility and occlusion. By assuming that the indoor environment resides on plane (*i.e.* distinct floors) we extract a 2.5D topological representation. All geometric elements in the scene above a specified height (here 1.5m) are treated as occluders.

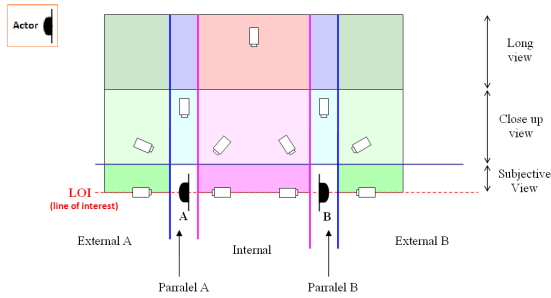


Figure 2: Example of semantic volumes for a key configuration of actors (adapted from [Ari76]).

4.1. Computing Semantic Volumes

Visual composition refers to the process of positioning the camera to furnish an image with the desired set of cinematographic properties. At the core of our approach is a representation, Semantic Volumes, that aggregates viewpoints that give rise to the qualitatively similar visual properties. The boundaries of the Semantic Volumes define when viewpoint changes occur.

We have fully specified sets of shots for key configurations of actors, which we use to characterize our Semantic Volumes (Figure 2). For example, in the case of a dialogue between two actors, a semi-space defined one side of the line of interest (LOI) captures the range of positions in which we can formulate over-the-shoulder shots of the two actors. We use a BSP to define these Semantic Volumes. This structure is dynamically and procedurally processed, and allows us to efficiently characterize and partition the environment into sets of viewpoints.

Some regions of space however cannot be selected to position the camera, since the actors are fully or partially occluded. Thus, an information on the visibility of the actors must be incorporated. In the sections that follow, we present a method to dynamically compute this visibility information.

4.2. A Topological Representation of the Environment

Topological representations of an environment are spatial structures that supports fast visibility computation and path planning around (typically static) obstacles. For the class of environments we consider these should be capable of addressing scene elements such as large vertical occluders, such as walls and pillars, and be capable of addressing configuration that constrain visibility (such as bottlenecks in the environment).

We use TopoPlan (Topological Planner) [Lam09] to preprocess the static environment geometry. TopoPlan analyses the scene geometry and generates a topological map as a triangular cell and portal decomposition based on a 2D constrained Delaunay triangulation.

The resulting representation contains pertinent information as to large vertical occluders and visibility bottlenecks, and allows us to rapidly compute the potential visibility of an object by taking into account of these large occluders.

4.3. Computing Visibility Volumes

Our process to compute the visibility of the actors has two elements: (1) a dynamic analysis of the potential visibility based on the precomputed static 2D topology and the configuration of the actors; and (2) a 3D dynamic refinement step, using a local visibility sampling that take into account dynamic occluders (*e.g.* the actors themselves) and elements that are not considered into topological analysis of the environment (*e.g.* furniture).

4.3.1. Using the static representation

Generating good shots of static and moving actors, requires us to maximize shot properties that relate to them, and in particular to their visibility. Whilst full visibility and complete occlusion is trivial to compute, partial visibility, which constitutes a large proportion of the shots of an actor, is more difficult to efficiently calculate. We use an original visibility analysis method, based on the precomputed topology, to create a visibility cartography for an actor.

Numerous visibility culling methods already exist to establish fully or partially occluded polygons [CF92] [BHS98] but generally such techniques neither helps us identify fully occluded volumes nor, in the case of a partially occluded polygon, provide it with a visibility degree estimation. Consequently, we instead base our visibility analysis on the use of stabbing lines to make a dynamic categorization of the visibility volume of an actor, by which we can also estimate a partial visibility degree. Where there is no occluder between a viewpoint and the actor (for example, any viewpoint within the same topological cell), the actor will always be fully visible, we therefore only focus on viewpoints that are located in other topological cells.

From a viewpoint p , two stabbing lines are defined such that each line is tangential to, but on opposite sides, of the convex hull of the actor (fig. 3) and are referred to as *extremal points*. We can rapidly compute the extremal points, in linear time, by observing that a point e on the convex hull is extremal if and only if the other points immediately preceding and following e are located on the same side of a stabbing line passing through e .

To determine the visibility of an actor a with a static occluder, where point p is an extremity of this occluder, the stabbing line associated to an extremal point e of the actor separates the region where e is visible and the region where e is occluded. Since the actor is fully included between stabbing lines, visibility categorization of the whole actor is then processed by combining visibility categorization of each stabbing line. Three regions are then obtained: a

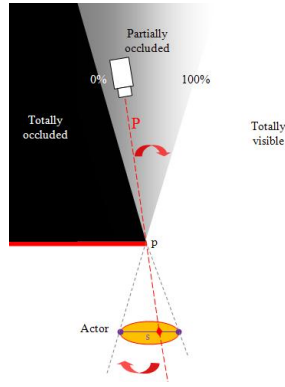


Figure 3: From a viewpoint, two stabbing lines are defined with respect to an actor. By combining both stabbing lines carried by an extremity p of an occluder, a categorization into a full occlusion region, a partial visibility region and a full visibility region of the actor is obtained. 2D visibility of an actor corresponds to a segment s linking its extreme points. From a viewpoint included in the region of partial visibility, the visible portion of the actor is evaluated by using relative angle between the plane P (passing by the viewpoint and the extreme point p of the occluder) and the stabbing lines of the actor. The visibility degree is then simply evaluated.

region where the actor is fully occluded, a region where the actor is fully visible, and an intermediate region where the actor is partially visible. From a viewpoint v , the 2D visibility of an actor corresponds to a segment s linking its extreme points. We thus propose to evaluate the visibility degree of the actor by computing the visible portion of s . By drawing a line through v and p , the degree of visibility of the actor can be estimated from the angle subtended.

Integration to topological analysis In reality the representation of the environment is based on the topological analysis and is a set of convex topological cells and portals. A portal comprises two points carried by one or more walls. Thus each point of a portal defines two stabbing lines for an actor, providing a subdivision of an adjacent topological cell into visibility cells. Moreover, since topological cells are convex, visibility cells will also be convex. The environment visibility characterization (Figure 4(a)) proceeds as: (1) the decomposition of topological cells into visibility cells by using stabbing lines ; and, (2) the recursive propagation of stabbing lines to adjacent cells.

Dynamic visibility analysis structure In this context, a BSP characterization is used for each topological cell. A node represents a stabbing line, and a leaf corresponds to a visibility cell of the actor. This structure provides: (1) an on-the-fly cell subdivision, ; and, (2) a reduction of the search complexity when characterizing a given viewpoint in the topological cell. However, occlusion due to other scene objects, not typically considered to be part of the environment (e.g. furniture), has not been considered. In the following

section, we show how these elements are incorporated in the dynamic visibility computation.

4.3.2. Using the dynamic environment

Christie et al. [CON08] presented an 3D approach to the real-time evaluation of the visibility of multiple target objects which simultaneously computes their visibility for a large sample of points. The visibility computation step involves performing a low resolution projection from points on pairs of target objects onto a plane parallel to the pair and behind the camera. By combining the depth buffers for these projections the joint visibility of the pair can be rapidly computed for a sample of locations around the current camera position. This pair-wise computation is extended for three or more target objects and visibility results aggregated in a temporal window to mitigate over-reactive camera behaviour.

In order to plan the next 3D camera position, we propose to rely on this local method. Once a 2.5D volume selected *w.r.t.* its potential visibility, and considering the current camera position/path, we propose to extend this volume into 3D, and then to sample the locally defined 3D potential positions around the current camera position/path. The new camera position is finally chosen among the set of the non-occluded viewpoints.

4.4. Director Volumes

In the previous sections, we have computed volumes corresponding to a semantic information, and volumes corresponding to a visibility information on an actor. Combining visibility volumes and semantic volumes, by using the compatibility between both BSP structures, allows to obtain a BSP partitioning into semantic volumes augmented with a visibility information on actors. This is the first step towards camera editing. We propose here a volume intersection method, in the aim to extract camera director volumes. BSP fusion methods have been proposed for a long time [NAT90]. We propose to intersect visibility volume relative to each actor, obtaining a multiple visibility volume partition (fig. 4(b)). The representation of semantic information as a BSP now allows to easily intersect it with visibility information, and to obtain a camera director partition of the whole environment (fig. 4(c)).

This information composition now provides a good base to reason on camera director volumes (semantic volumes augmented with a visibility information) to either make a jump, move in the environment, or maintain the camera position. In the following we will present our editing method, providing the camera paths and jumps given as input to the dynamic resolution of non-occluded camera positions.

5. Reasoning over Directors Volumes

In the application contexts we envisage narrative elements are not known in advance and are continuously introduced at

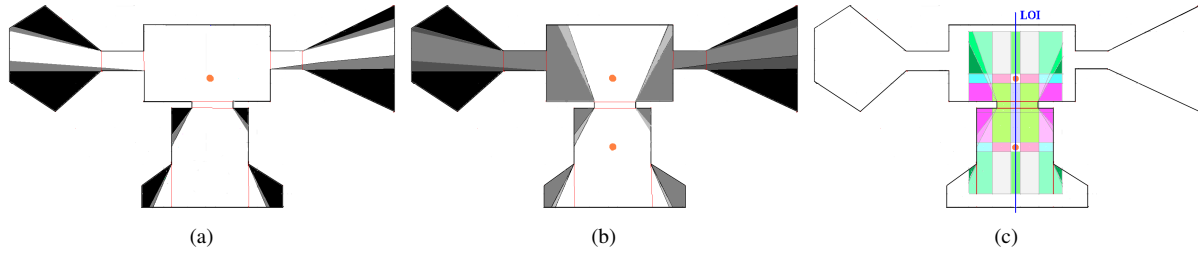


Figure 4: (a) Partitioning into visibility cells. Regions are divided into three groups: full visibility cells (white), full occlusion cells (black), and partial visibility cells (grey level) (b) Partitioning into multiple visibility cells (Two actors) (c) Semantic volumes linked to a configuration with two actors, and merged with the visibility information. Each semantic volume is colored, and the visibility degree is represented by the coloration degree of the cell.

run-time. Our automated cinematography module is therefore required to: (1) maintain rhetorical consistency with past shots; and (2) interactively interrupt the current shot to handle new elements as they become more relevant. Unlike most previous approaches [HCS96, ER07, Jha09] we enforce shot coherency rather than simply react to new narrative elements.

5.1. Narrative elements in Virtual Cinematography

The central notion around which our framework is design is the *narrative element*, and its coherent translation into cinematographic idioms. A narrative element is a component of the discourse and conveys relevant information or an action from the story. Each narrative element has a specific purpose and its conveyance leads to changes in a viewer's cognitive or emotional state. Narrative elements vary in accordance with the nature of the story, ranging from prototypical actions (a character stands up, walks, talks to another character) to more subtle notions such as the display of a relationship between characters (*e.g.* show dominance, conflict or isolation). As narrative elements are indexed in time, we refer to them as *events*. The *Narrative Pipeline* (Figure 1) therefore interactively injects events into the cinematography module.

Our editing model is built upon the notions of *idioms*, *coherency* and *style*.

Idioms An idiom is a sequence of reference shots that convey a narrative element. In contrast to approaches that consider only a single solution viewpoint [HCS96, CAwH*96], each shot is specified in terms of Semantic Volumes that relate to key subjects.

Coherency Coherency affords compliance with cinematography conventions that require maintenance of the spatio-temporal context when the camera changes viewpoint. Conventions are well established in relation to the actors' motions, spatial locations, the line-of-interest and actions. Though some conventions are implicitly encoded within idioms, it is necessary to maintain coherency when the

camera changes from one event to another, when visibility fails in the middle of an idiom, or in case of interleaved idioms (parallel editing).

Style Style is defined as a number of preferences as to: (1) shot pacing (which is the rate at which cuts are performed); (2) the camera speed (which favors either static or dynamic shots); (3) the minimum actor visibility in a shot (the level at which a character is considered occluded); and (4) order in which idioms should be considered where more than one applies.

The selection and sequencing of shots is fundamental to a viewer's understanding of the spatial, temporal and causal context. The appropriate choice of the next shot therefore needs to comply with a number of general rules, according to which we remove incoherent Director Volumes when switching viewpoints:

- Action continuity: between two shots, coherency must be maintained in the apparent speed and/or direction of motion.
- Screen direction: motion within the frame (up, down, left or right) must be maintained.
- Matching spatial relations: key spatial relations pertinent to the current action must be maintained, for example, in a dialogue if a character looks right towards another (off screen) character in one shot, this second actor should look towards left in the subsequent shot.

Following these rules, our editing process proceeds through a sequence of filters over the available director volumes, incrementally removing inappropriate volumes:

- filtering according to the idioms that relate to that event according to: (1) visibility, captured in Director Volumes (*i.e.* suppresses idioms whose semantic volumes are occluded); (2) style, through the application of preferences between idioms; and (3) the current viewpoint, each shot has a minimum/maximum duration;
- filtering according to coherency (*i.e.* relative position in relation to the previous semantic volume). For example, jump cuts must be between viewpoints that subtend an angle of at least 30 degree on the subject.

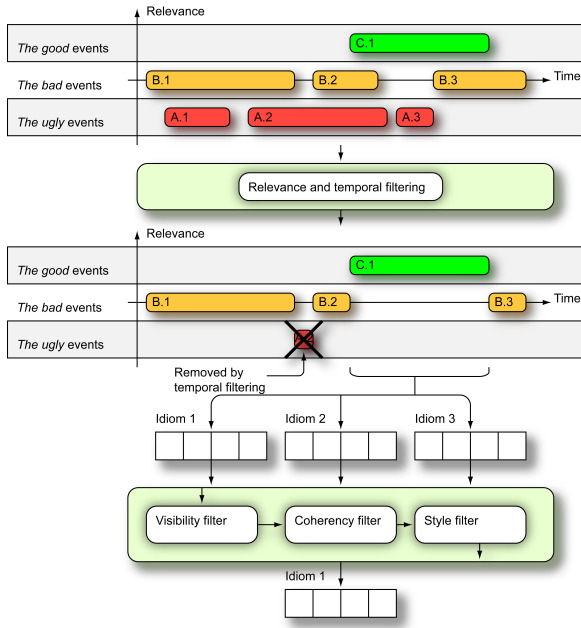


Figure 5: From story, extraction of a tree of possible idioms is processed, to plan the sequence of camera viewpoints

6. Results and Discussion

We present two examples that illustrate the key features or our automated cinematography system.

Our first example comprises two characters, A and B with a doorway. The narrative elements are typical character interactions for which idioms are well defined and overlap in time to illustrate the parallel editing feature of related actions:

- Event 1: **A moves towards B** can be captured using a sequence of three shots: an internal or parallel shot of character A; an apex shot of characters A and B; and an internal or parallel shot on character B.
- Event 2: **A crosses a doorway** can be captured using a sequence of two shots: a side shot, 3/4 rear, or rear shot; and a front, 3/4 front or side shot.

As the first event is triggered, the specified idiom is selected and the Director Volumes are filtered by merging Visibility and Semantic Volumes (here there is no coherence to maintain with previous shots). A viewpoint is then selected in one of the Director Volumes (1.1) which in turn will constrain the choices on the next shots (1.2 and 1.3) due to the line-of-interest coherency requirement. It is possible that the second event may be ignored, depending on the temporal overlap between idioms and the minimum duration of shots. However, once the new event is selected motion coherency is enforced due to the left-to-right motion of the subject on the screen, which filters the available volumes for shot 2.1,

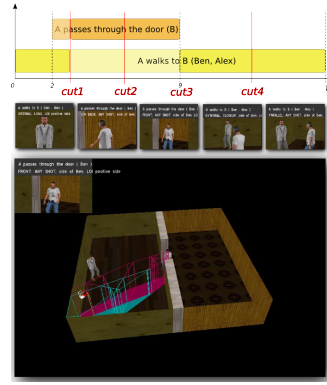


Figure 6: Conveying two related actions in parallel while maintaining coherency over the line of action and the line of interest.

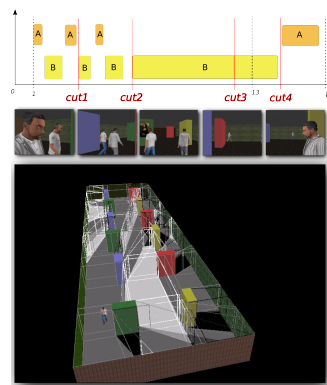


Figure 7: Following two actors in a dialogue scene. Camera avoids reactive switches with short utterances and uses one-character shots when apex shots are not available.

and forces the selection of a new viewpoint. The shots are displayed in Figure 6.

The second scene takes place in an environment with multiple pillars. Two characters A and B are engaged in a discussion while walking in the environment. The only narrative actions are the dialogue utterances. This example displays how our system overcomes some challenging problems. Firstly, around $t = 1$ s. utterances of characters A and B are very short. The system therefore ignores some of the events or postpones their shooting to avoid overly reactive cutting between shots. Secondly, around time $t = 5$ s., no apex shots are available to frame both characters. The filtering system therefore selects the next appropriate idiom in the list (an over-the-shoulder shot). Events and shots are displayed in figure 7.

6.1. Conclusion

In summary, we have presented a real-time cinematography system for virtual environments relies on an encoding of cin-

ematic idioms and coherency rules to produce appropriate edits and camera paths from a set of narrative actions. Our method relies on a spatial partitioning providing a characterization into visibility cells (fully visible, partially visible, or fully occluded) and characteristic viewpoints (the semantic volumes). To improve the visibility estimation, we utilized a local visibility sampling technique to handle minor and dynamic occluders. We then used these cells to reason at a symbolic level on how shot transitions should be performed, utilizing a user defined encoding for style and cinematic convention. The expressiveness of our system stands in stark contrast to existing approaches that are either procedural in character or do not account for dynamic visibility.

References

- [Ari76] ARIJON D.: *Grammar of the Film Language*. Hastings House Publishers, 1976.
- [AVF04] ANDÚJAR C. G., VÁZQUEZ P. P. A., FAIRÉN M. G.: Way-finder: Guided tours through complex walk-through models. *Proceedings of the Eurographics 2004 Conference (EG 2004)* 23, 3 (2004).
- [Bec02] BECKHAUS S.: *Dynamic Potential Fields for Guided Exploration in Virtual Environments*. PhD thesis, Fakultät für Informatik, University of Magdeburg, 2002.
- [BHS98] BITTNER J., HAVRAN V., SLAVIK P.: Hierarchical visibility culling with occlusion trees. *Computer Graphics International Conference 0* (1998).
- [BK01] BARES W., KIM B.: Generating virtual camera compositions. In *Proc. of the 6th international conference on Intelligent user interfaces (IUI 01)* (2001), ACM Press.
- [CAwH*96] CHRISTIANSON D. B., ANDERSON S. E., WEI HE L., SALESIN D. H., WELD D. S., COHEN M. F.: Declarative Camera Control for Automatic Cinematography. In *Proc. of the 13th National Conference on Artificial Intelligence and the 8th Innovative Applications of Artificial Intelligence Conference* (1996), AAAI Press / MIT Press.
- [CF92] CHIN N., FEINER S.: Fast object-precision shadow generation for area light sources using bsp trees. In *SI3D '92: Proc. of the 1992 symposium on Interactive 3D graphics* (1992), ACM.
- [CLDM03] COURTY N., LAMARCHE F., DONIKIAN S., MARCHAND E.: A cinematography system for virtual storytelling. In *Proc. of the International Conference on Virtual Storytelling 2003* (November 2003), vol. 2897.
- [CM01] COURTY N., MARCHAND E.: Computer animation: A new application for image-based visual servoing. In *Proc. of International Conference on Robotics and Automation, (ICRA 2001)* (2001).
- [CN05] CHRISTIE M., NORMAND J.-M.: A semantic space partitioning approach to virtual camera control. In *Proc. of the Eurographics Conference (EG 2005)* (2005), vol. 24, Computer Graphics Forum.
- [CON08] CHRISTIE M., OLIVIER P., NORMAND J.-M.: *Occlusion-free Camera Control*. Research Report RR-6640, INRIA, 2008.
- [ER07] ELSON D. K., RIEDL M. O.: A lightweight intelligent virtual cinematography system for machinima production. In *Proc. of the 3rd Conf. on AI for Interactive Entertainment* (2007).
- [FF04] FRIEDMAN D. A., FELDMAN Y. A.: Knowledge-Based Cinematography and Its Applications. In *Proc. of the 16th European Conference on Artificial Intelligence (ECAI 2004)* (2004), IOS Press.
- [HCS96] HE L., COHEN M. F., SALESIN D. H.: The virtual cinematographer: A paradigm for automatic real-time camera control and directing. In *SIGGRAPH* (Aug. 1996), ACM Computer Graphics.
- [Jha09] JHALA A.: *Cinematic Discourse Generation*. PhD thesis, Faculty of North Carolina State University, 2009.
- [Lam09] LAMARCHE F.: Topoplan: a topological path planner for real time human navigation under floor and ceiling constraints. *Computer Graphics Forum* 28, 2 (2009).
- [LC08] LI T.-Y., CHENG C.-C.: Real-time camera planning for navigation in virtual environments. In *SG '08: Proc. of the 9th international symposium on Smart Graphics* (2008), Springer-Verlag.
- [NAT90] NAYLOR B., AMANATIDES J., THIBAUT W.: Merging bsp trees yields polyhedral set operations. In *SIGGRAPH '90: Proc. of the 17th annual conference on Computer graphics and interactive techniques* (1990), ACM.
- [NO03] NIEUWENHUISEN D., OVERMARS M. H.: *Motion Planning for Camera Movements in Virtual Environments*. Tech. Rep. UU-CS-2003-004, Institute of Information and Computing Sciences, Utrecht University, 2003.
- [OSTG09] OSKAM T., SUMNER R. W., THUREY N., GROSS M.: Visibility transition planning for dynamic camera control. In *SCA '09: Proc. of the 2009 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2009), Eurographics Association.
- [TS91] TELLER S. J., SÉQUIN C. H.: Visibility preprocessing for interactive walkthroughs. In *SIGGRAPH '91: Proc. of the 18th annual conference on Computer graphics and interactive techniques* (1991), ACM.
- [VFSH03] VÁZQUEZ P.-P., FEIXAS M., SBERT M., HEIDRICH W.: Automatic view selection using viewpoint entropy and its application to image-based modelling. *Computer Graphics Forum* 22, 4 (2003).
- [You07] YOUNG R. M.: Story and discourse: A bipartite model of narrative generation in virtual worlds. *Interaction Studies* 8, 2 (2007).