Real-time Cinematic Camera Control for Interactive Narratives

Daniel Amerson and Shaun Kime

Department of Computer Science
North Carolina State University
Raleigh, North Carolina 27607
{dbamerso, sckime}@unity.ncsu.edu

Abstract

In conventional, visual narrative media such as film, the placement and movement of the camera can be as important a device as the events of the narrative in conveying story and meaning. Hollywood has developed a set of common cinematographic techniques to convey meaning for different subject matters. For interactive narratives in 3D virtual worlds, the camera can be manipulated in the same way as traditional cinema. We propose a system for real-time camera control in interactive narratives called FILM, Film Idiom Language and Model, that uses these common cinematographic techniques to construct camera placements based on input from a narrative planner. Information about common film idioms is encoded in a scene tree using the FILM language. Objects within the FILM system use this knowledge in conjunction with the planner inputs to constrain the location and orientation of the camera for viewing a given action at execution time.

Introduction

The evolution of computer gaming has presented an interesting problem for developers. Original gaming environments were limited in the amount and quality of multimedia information they could present. Because of this fact, the potential for storytelling by developers was limited by the medium. As games have developed from two-dimensional sprite based games into fully three-dimensional worlds, the potential for storytelling has increased dramatically. Current 3D game engines such as Unreal Tournament(UT) or Quake 3 possess functions to fully control a virtual camera in the game environment. Because these worlds are so much richer in media, the potential for storytelling and deeper immersion into interactive narratives is possible.

In this paper, we discuss automated cinematography as it relates to interactive narratives in virtual worlds. Due to the interactive nature of these environments, automated camera controllers cannot fully utilize all of the idioms in the domain of cinematography. For instance, while the exact flow of action is often prescripted in cinematography, in interactive environments we can only roughly predict the flow of action in the narrative because users may substantially change the course of action at any time. Any system that depends on certainty of prediction to generate effective camera movements will be very brittle in these environments. As an alternative model of interactive cinematography, we present a system that is currently under development called FILM. FILM, (Film Idiom Language and Model) is one component of the larger Mimesis Project (Young 1999), an architecture for the generation and control of interactive 3D narratives.

Related Work

It is important to note that substantial investigations into interactive cinematography have already been made. One existing system, DCCL (Declarative Camera Control Language) also outlines a similar system for the creation of camera controls that obey cinematographic constraints (Christianson et al., 1996). This system lays a strong conceptual foundation for our work, but was not designed to function in a real-time environment. Instead, the system takes a pre-existing animation trace and chooses the conventions that best show the events that the trace contains. Another system, UCAM, uses a constraint-based approach to select for shot composition (Bares and Lester 1997). This system works extremely well in real time, but does not fully capitalize on the cinematographic body of work. The Virtual Cinematographer system (He, Cohen, & Salesin, 1996) exploits a scene hierarchy to model cinematography conventions in a real-time environment. We have used this notion as a basis for a representation for our scene tree. In contrast to these individual systems, we propose a hybrid system that uses abstractly defined cinematographic idioms as constraints to choose the best camera placement for any shot at any moment within any geometry.

Film as a Metaphor

Most people are comfortable with narratives that are conveyed using video and sound because Hollywood cinema has created a structured style for conveying stories that is familiar to viewers. This style is so structured that certain types of events are almost always filmed from the same camera positions and angles. These structured scenes

and shots are referred to as film idioms. To capitalize on this familiarity with cinematic styles our camera control algorithm is based on these film idioms.

Logically, a film can be broken down into scenes. A scene is a unit of a film that conveys the sense of a series of events occurring in a continuous space and time. In film, there are many common templates for camera positioning and motion, which are called film idioms. We encode film idioms at the scene level in the FILM system. Scenes can further be broken down into shots or takes. A shot in cinema is an uninterrupted segment of exposed film. A shot lasts from the time the camera is turned on to the time the camera is turned off. Each of these shots can be described using the language developed by directors and cinematographers from years of experience (For more information regarding the language and conventions of filmmaking, we recommend Film, An Introduction [Phillips, 1999]).

The structure of the camera control architecture proposed here is based on the notion of classical filmmaking practices. In the FILM system, decision-making individuals like the director and cinematographer are represented by software objects of the same name, and possible camera movements are broken down into scenes composed of shots. The director selects scenes based on the subject nature of the shot. Unlike true cinema, however, the state of the world is not wholly determined when the director makes decisions. For example, in film the director can control how actors move. Environments such as those built by the Mimesis Project are interactive, so predictions must be made about how the state of the world will change. Because of this, shots and scenes are specified with constraints that can be relaxed in case of conflict rather via description written in an absolutely declarative language.

The Mimesis Project

The Mimesis Project uses 3D gaming platforms to tell interactive narratives in a rich multimedia environment. These narratives tell stories that adapt to fit the actions of the user. We use existing 3D engines to leverage the time and effort necessary to produce these interactive environments. At the time of this article, we have chosen the UT engine as our first development platform. This engine was chosen for initial development due to ease at which the engine can be modified without the purchase of a full engine license. In brief, the Mimesis system consists of an AI server that communicates with a modified UT game server. The AI server is comprised of a HTN-style decomposition planner, called Longbow (Young 1994), and several supporting subsystems. Longbow generates plans that define and communicate a narrative through a sequence of actions. The UT server communicates the user's actions in the world to the AI server and carries out the plan that is generated by the AI server. If the user's actions violate the steps in the plan, then new actions are taken to either alter the threatening action so that it no longer threatens the plan or replan to change the narrative to reflect the user's action. Communication between the two ends of the pipeline occurs through a socket connection. In the proposed implementation, the FILM system would allow spectators to view a cinematic display of the narrative as it unfolds.

Camera Control Architecture

The FILM system is defined by a pipeline of objects beginning with the narrative planner, Longbow. This planner generates the information that must be conveyed to a user during a given scene (typically, the actions carried out by characters and their actions' consequences). This information is passed to a domain specific Translator that is able to map the Longbow action descriptors in terms that the next module, the Director, can understand. The Director uses the translated scene requirements to select a specific film idiom from a predefined scene tree. It then binds any unbound variables in the chosen scene to objects in the virtual world. This scene specification is then passed down to a graphics engine specific Cinematographer. The Cinematographer takes the constraints that define the individual shots in a scene and chooses the best camera position in the world to satisfy those constraints (described below).

Scene Tree

The scene tree encodes the knowledge about film idioms that is needed to construct appropriate camera placements throughout the storytelling experience. Scenes are encoded using the FILM language which is expressive enough to cover a wide variety of idioms. Each scene has certain components that allow it to be heuristically searched by the Director when selecting a shot. The tree is constructed in levels. At the root is a generic shot followed by a level that is differentiated by shot type. The next three levels are differentiated by number of participants, emotional effect of the shot, and a keyword list for finest shot selection. Each subsequent level in the tree contains a scene that is more refined and conveys more information to the user. This property allows us to make quick searches of the tree. Once we have a keyword match, we can eliminate all of a node's siblings and only search its children. If the node is ranked the same as its children, this means that the children scenes will provide more information about the world, but this information is not required to fulfill our objectives. In this case, we will err on the side of caution and choose the scene that meets all of our objectives and only our objectives. Using the children nodes may add information that will confuse the user. For instance, if our objective is to show a conversation shot between two people, it is not necessary to show the emotional intent of that shot because it is not specified. If we choose one of the children, the scene may show an angry conversation, which will be

confusing and unnatural. It is only when the child node fulfills more of our objectives that the child node is selected.

The FILM Language

The FILM language is designed to be both expressive enough to characterize the camera behaviors of any film idiom and also flexible enough to dynamically adapt to any given shot for any world geometry. The basic units of a FILM idiom specification are the scene and the shot. These objects in the FILM specification represent the analogous concepts in cinema. Each scene contains some subject information and a shot list. Each shot in the list is specified by certain required primitives and constraints on the proper positioning of the camera. An example of a simple scene follows. In this scene an actor moving in some direction is filmed in the center of the shot for an unspecified period of time. After that time elapses the camera's position is fixed and the actor walks offscreen.

```
NAME: Movement1
SCENE_TYPE: Movement
NUMBER_OF_PARTICIPANTS: 1
EMOTIONAL_EFFECT: Neutral
KEYWORD_DESCRIPTION:
SHOT_LIST:
   NAME: Tracking
   CUT_ON: timeOutEvent(?time)
   BASIS: ?actor.viewVector
   CONSTRAINTS:
      lensType(NORMAL_LENS) WEIGHT=.9
      lookAt(?actor) WEIGHT=1
      relativeLocation(?actor, LONG_SHOT, (1,0,0))
WEIGHT=.7
      maintainRelativeLocation(?actor, LONG_SHOT,
(1,0,0) WEIGHT=.8
   NAME: Exit
   CUT_ON: timeOutEvent(?time)
   BASIS: ?actor.viewVector
   CONSTRAINTS:
      maintainLensType() WEIGHT=.8
      maintainRotation() WEIGHT=.9
      maintainAbsoluteLocation() WEIGHT=1
```

In the example above, the lines prior to the shot list describe the requisite information to place the scene in the scene tree correctly. The shot list defines the sequence of shots in this scene. Each shot has a text name and three variables, CUT-ON, BASIS, and CONSTRAINTS. CUT-ON defines the ending event that causes the Cinematographer to move to the next shot on the list. In the above example the events are of simple duration, but more complicated events can be constructed allowing flexibility in the idiom specification. BASIS is a vector indicating the positive axes of the shot. This basis allows a frame of reference for defining camera location and direction relative to the principle subjects of a scene. The final component of a shot is the sequence of constraints that restrict the instantiation of a shot. Each constraint consists of some function with its arguments. Arguments can be explicitly specified such as lensType(NORMAL_LENS), or they can be unbound variables indicated in the form ?TYPE (e.g. lookAt(?actor)). Each constraint also has a weight, indicating its relative importance to the shot, so that constraints can be appropriately relaxed if all constraints cannot be satisfied simultaneously.

The Translator

Because the information output by the narrative planner is not formatted for the Director object, a Translator mediates between the planner and the Director. The Translator contains certain domain specific information that is used to recast its input into a format that allows the Director to select scenes and formulate bindings for variables. The Translator is essentially a mapping from planner operators to the Director's input structure allowing the Director to be domain independent.

The Director

In traditional film production, the director is responsible for the overall decision making process. He ensures that the narrative is conveyed effectively using the film techniques at his disposal. In the FILM system, the Director has a similar role. The Director object takes the translated input and determines which scene specified in the scene tree best fits the specification. To do this the Director requires certain information from the Translator, namely the type of scene, number of participants, and emotional or affective context of the story at the current point in its telling. Using this information the Director performs a depth first search with no backtracking to select the best scene from the scene tree. This search method works upon the assumption of tree construction that specifies each level of the tree contains more critical information for scene specification than lower levels of the tree. Once a scene is selected, the Director binds any unbound variables in the scene specification and passes the information to the Cinematographer. These unbound variables may be time constants, locations, rotations, actors, or props in the world.

The Cinematographer

Once the director has specified the scene and the constraints that effect the filming of that scene, it is the Cinematographer's job to choose the optimal position for the camera. The Cinematographer uses the constraints passed in by the Director to select an optimal placement for the camera, but this selected optimal position may or may not be a valid position in the virtual environment. Some possible reasons that can make an optimal position invalid are that the shot may be occluded by another object in the world or a pair of the constraints may not be easily satisfied in tandem. In this case, we relax the constraints out from our optimal point to determine the closest point that will best satisfy our constraints. Constraints are

relaxed based on the weights defined in the scene specification. Currently we employ an ad hoc procedural model for the relaxation of constraints. In this model we define the relaxation of particular constraint types using predefined procedures. Relaxation of a location constraint, for instance, can be implemented by a function that moves the camera along an orbit in the XY-plane or changes the magnitude of the camera distance from its target. In the future, it may be possible (and even desirable) to use a more general constraint solver like the ones used by UCAM (Bares and Lester, 1997) or the museum tour system (Drucker and Zetzer, 1995).

Since the Cinematographer directly communicates with the graphics engine, the Cinematographer is the graphics engine specific component of the system. In order to effectively consider how to place the camera, the Cinematographer needs to be able to map the Director's specification of a shot to the actual graphics engine's capabilities. For instance, UT's base unit of length is not the same as the Director's base unit of length. By accessing a database of facts specific to the graphics engine being used to render the world (e.g. the height of a player model's head from the ground, fields of view for given lens types), the Cinematographer can construct camera placements in absolute world coordinates that satisfy the relative constraints of the Director.

Conclusion

The FILM system combines the expressiveness of a cinematography expert system with the ability to adapt a film idiom to fit with the environment on the fly. This blending is accomplished through two objects, the Director and Cinematographer. The Director translates the subject information into camera constraints which the Cinematographer can then apply to create camera placements in world coordinates. By utilizing this model of film idioms, the camera's position and movement can convey narrative information along with the events of the narrative. Upon completion, the implementation of this architecture in the UT engine will show how abstract film idiom knowledge can be selected and converted into virtual camera movements.

References

Bares, W. H.; Gregoire, J. P.; Lester, J.C.; 1998. Real-time constraint-based cinematography for complex interactive 3D worlds. In Proceedings of the Tenth National Conference on Innovative Applications of Artificial Intelligence, 1101-1106.

Bares, W. H; Lester, J.C. 1997. Cinematographic user models for automated real-time camera control in dynamic 3D environments. In Proceedings of the Sixth International Conference on User Modeling, 215-226.

Christianson, D. B.; Anderson, S. E.; He, L.-W.; Salesin, D. H.; Weld, D. S.; and Cohen, M.F. 1996. Declarative camera control for automatic cinematography. In Proceedings of the Thirteenth National Conference on Artificial Intelligence, 148-155.

Drucker, S. and Zeltzer, D. 1995. CamDroid: A system for implementing intelligent camera control. In Proceedings of the 1995 Symposium on Interactive 3D Graphics, 139-144.

He, L.; Cohen, M.; and Salesin, D. 1996. The virtual cinematographer: A paradigm for automatic real-time camera control and directing. In Proceedings of the ACM SIGGRAPH '96, 217-224.

Philips, W.H. 1999. Film: an introduction. Bedford / St. Martin's. Boston, MA.

Young, R. M.; Pollack, M. E.; Moore, J. D. 1994. Decomposition and causality in partial-order planning. In the Proceedings of the Second International Conference on Artificial Intelligence and Planning Systems, 188-193.

Young, R. M. 1999. Notes on the use of plan structures in the creation of interactive plot. In the Working Notes of the AAAI Fall Symposium on Narrative Intelligence, .