Cinematographic User Models for Automated Realtime Camera Control in Dynamic 3D Environments

William H. Bares and James C. Lester*

Multimedia Laboratory, Department of Computer Science,
North Carolina State University, Raleigh, NC U.S.A.

Abstract. Advances in 3D graphics technology have accelerated the construction of dynamic 3D environments. Despite their promise for scientific and educational applications, much of this potential has gone unrealized because runtime camera control software lacks user-sensitivity. Current environments rely on sequences of viewpoints that directly require the user's control or are based primarily on actions and geometry of the scene. Because of the complexity of rapidly changing environments, users typically cannot manipulate objects in environments while simultaneously issuing camera control commands. To address these issues, we have developed UCAM, a realtime camera planner that employs cinematographic user models to render customized visualizations of dynamic 3D environments. After interviewing users to determine their preferred directorial style and pacing, UCAM examines the resulting cinematographic user model to plan camera sequences whose shot vantage points and cutting rates are tailored to the user in realtime. Evaluations of UCAM in a dynamic 3D testbed are encouraging.

1 Introduction

Dynamic 3D environments hold great promise for a broad range of educational and scientific applications. By enabling users to participate in immersive experiences as they learn about complex systems or perform complicated tasks, 3D environments can help students and scientists achieve greater levels of performance. Fortunately, recent advances in graphics hardware have created significant opportunities for making dynamic 3D environments available on a broad scale. With the proliferation of inexpensive graphics accelerators, dynamic 3D environments can soon become an indispensable, cost-effective means of delivering customized instruction and visualizations in domains as diverse as molecular biology, computer engineering, and medicine. By enabling users to view complex processes and interact with objects in 3D animations in realtime, dynamic 3D environments can significantly increase the effectiveness of knowledge-based learning environments and scientific visualization systems.

Effective realtime camera control is critical to the successful deployment of dynamic 3D environments. In dynamic 3D environments, virtual cameras track the objects of interest to depict the most salient aspects of complicated scenes. Two approaches have been proposed for camera control. Some systems require users to directly control low-level camera positioning and

* Support for this work was provided by the IntelliMedia Initiative of North Carolina State University and donations from Novell, Inc.

orientation parameters, while others automatically control camera movement without considering users' visualization preferences. The first approach is problematic when users must perform complex tasks while simultaneously issuing camera control commands, a particularly acute problem in highly dynamic environments. Although the second approach frees users from camera control, it fails to consider their individual visualization preferences.

Dynamic 3D environments must accommodate a broad range of users. Each individual brings his or her own idiosyncratic visualization preferences for experiencing a particular 3D environment. While some users prefer informative styles, others prefer visualizations with a dramatic flair. Some users may be unfamiliar with a task that involves unusually complex visualizations; these situations call for slower camera pacing, gradual transitions, and an informational visualization style. Moreover, users that are intimately familiar with a particular aspect of a task may prefer a faster camera pace and rapid transitions.

Given the broad range of students and scientists who will interact with dynamic 3D environments, user-sensitive automated camera control is quickly becoming essential. However, user-sensitive camera control poses a difficult challenge: determining the positions and directions of virtual cameras in realtime is enormously difficult because shots must clearly depict the portion of the scene most relevant to the user while at the same time taking into account his or her visualization preferences. These functionalities call for a user modeling approach to representing users' visualization preferences and to performing customized realtime camera planning. While a growing body of work considers user modeling for multimedia presentation systems (André et al., 1993, McKeown et al., 1992, Roth et al., 1991, van Mulken, 1996) the problem of user modeling for camera control in dynamic 3D environments has not been addressed: current environments rely on sequences of viewpoints that directly require the user's control or are based primarily on actions and geometry of the scene (Butz, 1997, Christianson et al., 1996, Drucker and Zeltzer, 1995, Karp and Feiner, 1993, Mackinlay et al., 1990, Ware and Osborn, 1990).

To address these problems, we have developed the *cinematographic user modeling* framework for user-sensitive realtime camera control in dynamic 3D environments. This domain-independent framework has been implemented in UCAM,[1] a user-sensitive realtime camera planner. After constructing an expressive representation of users' visualization preferences, UCAM creates customized immersive experiences by exploiting its cinematographic user model to plan camera positions, view directions, and camera movement in response to users' manipulations of objects in 3D environments. UCAM has been evaluated in a 3D environment with subjects from both technical and art backgrounds. Subjects interacted with UCAM to perform two families of tasks: a visualization task in which they specified their cinematographic preferences to create a 3D visualizations of long sequences of actions, and a navigation task in which they maneuvered a virtual vehicle through a cityscape. The results of this evaluation are encouraging and demonstrate that user-sensitive automated realtime camera control significantly improves users' interactions with dynamic 3D environments.

2 Customized Camera Planning in 3D Environments

Realtime camera planning in dynamic 3D environments entails selecting camera positions and view directions in response to changes to objects in the environment that are caused by users'

[1] User-Customized Automated Montage

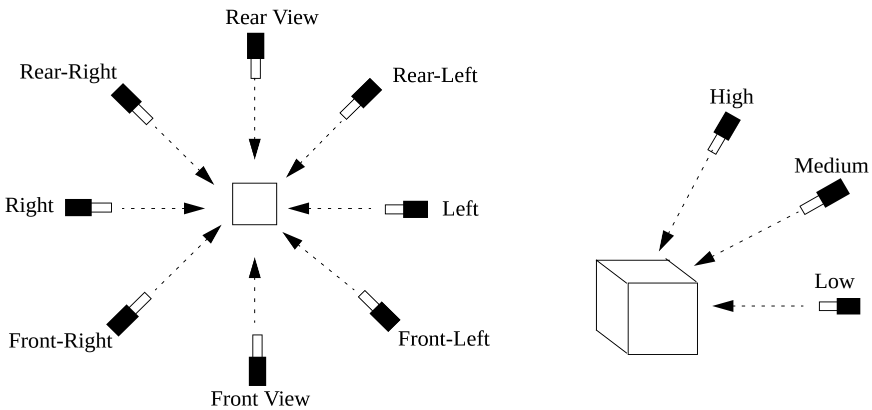

Figure 1. Camera viewing angles and elevations for 3D environments.

manipulations or a simulation. A virtual camera must track the objects by executing cuts,[2] pans, and zooms (pull-ins and pull-outs) to make on-the-fly decisions about camera viewing angles, distances, and elevations. Planning camera shots and camera positions while preserving continuity requires solving precisely the same set of problems that are faced by cinematographers, with the additional constraint that they must be solved in realtime.

In dynamic 3D environments, camera planners must continually make decisions about shot types and camera positions. *Shot types* are characterized by several dimensions, including the size of the subject as it appears in the frame and the relative position of the camera to the subject. For example, the subject occupies all of the frame for a close-up shot. In a long shot, the subject occupies a small portion of the frame. Different shot types are more useful in particular situations. Long shots are preferred for depicting wide-ranging action or showing relative size or position of subjects. Close-up shots are useful for emphasizing detail and for focusing on a single subject (Millerson, 1994). Camera *positions* are defined by the viewing angle and elevation relative to the subject (Figure 1). For example, the camera can be placed directly in front of the subject or to the right of the subject; it can be placed slightly below the subject and gaze up towards it to exaggerate its size, or high above the subject gazing downwards. High and far shots present more information about the scene but tend to be less interesting (Mascelli, 1965). Preserving continuity during transitions such as cuts, panning, and tracking is critical. However, it is also difficult because camera planners must maintain the viewer's interest with a variety of shots without introducing jarring visual discontinuities (Mascelli, 1965).

Given the complexities of camera planning, users performing tasks in 3D environments should be able to delegate the myriad micro-level camera planning decisions to an automated camera planner in order to focus their attention on their own tasks. For example, a biologist observing the effects of a T-cell's traversal of the lymph system should not be forced to continually make decisions about how to adjust the virtual camera's elevation, orientation, and zoom level, when to make cuts, and how to pan left and right. Rather than making a series of incremental, ongoing modifications to the virtual camera, perhaps even by specifying 3D motion spline paths along

[2] A *cut* is an instantaneous change from one shot to another without an intervening transition.

which the virtual camera will travel, users should be able to describe how they would like to experience the environment and then have a user-sensitive camera planner interpret these preferences in realtime in response to changes in the environment.

Customized automated camera control for dynamic 3D environments calls for a user modeling framework for representing and reasoning about users' environmental viewing preferences. A growing number of projects have attacked the problem of user modeling for multimedia systems. These include the COMET (McKeown et al., 1992), SAGE (Roth et al., 1991), WIP (André et al., 1993) and PPP (van Mulken, 1996) work on customized presentation planning, and PPP André and Rist (1996) and DESIGN-A-PLANT (Lester et al., 1997, Stone and Lester, 1996) for customized behavior of animated interface agents. However, user modeling for customized camera planning in 3D environments has not been addressed. Most 3D environment projects require the user to operate the camera (Mackinlay et al., 1990, Ware and Osborn, 1990). Several recent efforts have begun to address intelligent camera control, but they do not employ a user model to represent users' visualization preferences. CAMDROID (Drucker and Zeltzer, 1995) allows the user to design a network of camera modules and constraints but has no user model. The VIRTUAL CINEMATOGRAPHER (Christianson et al., 1996) and ESPLANADE (Karp and Feiner, 1993) employ film idioms to successfully maintain camera shot sequences that are consistent with film conventions, but they cannot customize animations since no user model is maintained. CATHI (Butz, 1997), which is part of the PPP project (André and Rist, 1996) permits users to state visualization preferences such as the use of spotlights, depth of field, and animation duration, as well as animation preferences that include two cinematic styles. However, the cinematic styles are specified by the choice of one of two grammars of film rules rather than more fine grained user modeling of individual cinematic attributes such as camera pacing, viewpoint style, and transition style.

3 Cinematographic User Modeling

To address the problem of customized realtime camera control for dynamic 3D environments, we have developed the domain-independent *cinematographic user modeling* framework and implemented it in UCAM, a realtime cinematographic user modeling system (Figure 2). By constructing cinematographic user models and creating a camera planner that exploits the models to select camera shots and enact camera transitions, UCAM creates interactive viewing experiences that are highly customized to individual users' preferences in realtime. Cinematographic user models enable users—including users with no cinematographic expertise—to become "directors" of their experiences through a two step process:

1. **User Model Construction:** To accommodate the majority of users' lack of familiarity with cinematography, it is critical that a "director studio" provide them with a tool that is simple yet expressive. As users describe their visualization preferences for how they wish to interactively experience a 3D environment, UCAM constructs a cinematographic user model. Represented in a *Cinematographic Specification Language* (CSL), cinematographic user models consist of probabilistic micro-level camera planning directives including specifications for shot type selection, camera orientation, minimum shot durations, and angular difference thresholds for cut/pan decisions.

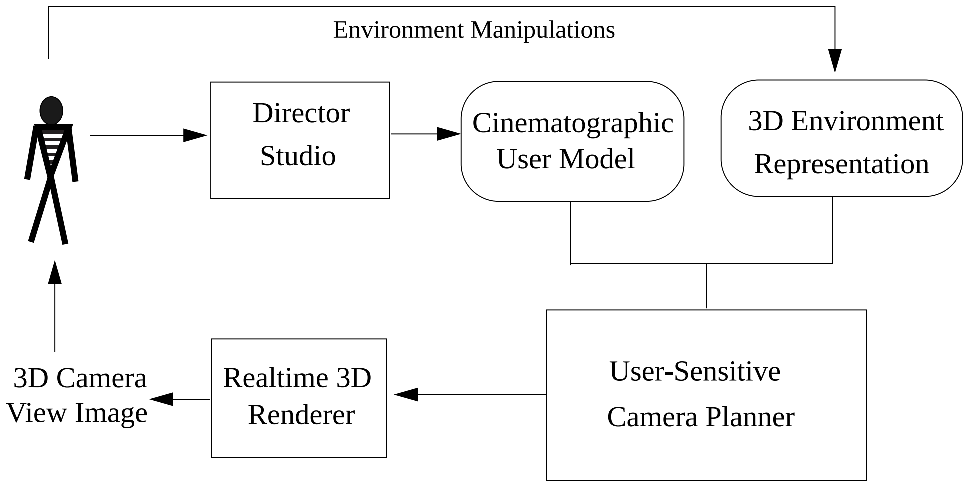

Figure 2. The UCAM architecture.

2. **Customized Realtime Camera Planning:** A camera planner interprets these models to plan camera positions, view directions, and camera transitions in realtime. As users navigate through an environment (perhaps traversing an expansive complex landscape) and manipulates objects in the scene, the planner computes executable directives for *shot type* (e.g., close-up, long), *viewing angle* (e.g., front-left), *viewing elevation*, *transitions* (cut, tracking, panning), *shot duration*, and *panning* and *tracking speeds* in realtime to frame objects of interest as they move about.

These directives are then passed to the renderer, which composes the next frame depicting the 3D environment. The net effect of viewing these rapidly rendered frames is a seamless immersive experience that is customized for users' visualization preferences.[3] UCAM also permits users to modify their visualization preferences at any time.

3.1 Constructing Cinematographic User Models

In interacting with a customized camera planning system, it is critical that users can easily express their visualization preferences without being overwhelmed by an enormous number of selections. UCAM therefore provides a menu-based "director studio" with which users classify their visualization preferences along three dimensions: viewpoint style, camera pacing, and transition style as shown in the first two columns of Table 1.

Specific viewpoint styles can be achieved with different shot types and elevations, which together bring about a specific cinematographic impact (Mascelli, 1965). Users can select either an informational or a dramatic viewpoint. Specifying an *informational* viewpoint style will produce a visualization that is more clear and informative by employing more medium and long shots, as well as more medium and high elevation shots. Specifying a *dramatic* viewpoint style

[3] UCAM completes the planning-rendering cycle every 1/8 of a second on a PC.

Table 1. Semantics of visualization preferences

Visualization Preference	*Value*	*Cinematographic User Model Camera Directives*
Viewpoint Style	Informational	Medium and long shots more probable Medium and high elevation shots more probable
	Dramatic	Close-up and near shots more probable Low and medium elevation shots more probable
Camera Pacing	Slow	Longer shot duration ($\geq$ 35 frames) Pan in increments of 1°
	Fast	Shorter shot duration ($\leq$ 15 frames) Pan in increments of 4°
Transition Style	Gradual	Always pan and track between different shots
	Jump	Cut if angular distance between shots $>$ 60°

will produce an experience that is more dramatic by having the camera planner prefer close-up and near shots and low and medium elevation shots.

Users may also state pacing and transition preferences. *Slow* pacing will produce an interactive experience that is perceived as slower by increasing shot durations and reducing the speed of tracking and panning shots. *Fast* pacing will produce an experience that seems more intense by decreasing shot durations and increasing tracking and panning speeds. Preferences for transition styles can be either gradual or jumping. A *gradual* transition preference will achieve a more relaxed experience by causing the camera planner to opt for panning and tracking between shots, while a *jump* transition preference will produce a more staccato experience by causing the camera planner to cut from shot to shot. Users can state their preferences for these dimensions in any order.

The director studio builds a cinematographic user model by mapping high-level visualization preferences to low-level camera planning directives expressed in the probabilistic CSL. The semantics of the user's visualization preferences are summarized in the third column of Table 1. To illustrate, suppose a user expresses her preferences for a dramatic viewpoint style, fast pace, and jump transitions. The director studio creates a cinematographic user model with a 30% probability for close-ups, 40% for near shots, 20% for medium shots, and 10% for far shots. It selects camera elevation probabilities of 50% for low, 50% for medium, and 0% for high. It chooses a minimum shot duration of 15 frames, a panning/tracking rate of 4° per unit time, and a minimum cut angle of 60°.[4] All of these factors will be considered by the camera planner in making shot modification determination decisions, shot selection decisions, and camera transition decisions at each instant of the visualization.

3.2 User-Sensitive Realtime Camera Planning

UCAM's camera planner exploits the visualization directives represented by cinematographic user models to create customized experiences for users interacting with 3D environments. As a

[4] Specific probabilities and cut angle values in the implementation were developed empirically to reflect the semantics of Table 1.

```
loop
    3DEnvironment ← update-environment
    if user modifies visualization preferences then
      CinematicUserModel ← construct-UM (VisualizationPrefs)
    if NumFrames < MinimumShotDuration then
      (* no change to CamShot but move camera to track object *)
      CamPosition ← select-new-position (3DEnvironment, null-transition)
    else {
      CamShot ← select-new-shot (CinematicUserModel, 3DEnvironment)
      NumFrames ← 0
      if AngleToNewShot < MinimumCutAngle then
        CamTransition ← pan to new position
      else
        CamTransition ← cut to new position
      CamPosition ← select-new-position (3DEnvironment, CamTransition)}
    NumFrames ← NumFrames + 1
    NewFrame ← render (3DEnvironment, CamShot, CamPosition)
until user exits visualization
```

Figure 3. Realtime user-sensitive camera planning.

user manipulates objects in an environment, the camera planner considers his or her preferred viewpoint style, pacing, and transition style to make continuous runtime decisions about camera positioning. A visualization begins with the user expressing his or her preferences through the director studio interface, which are used to construct a cinematographic user model. After initializing the camera shot and position to display the *primary object of interest*, e.g., for a physiologist this might be a particular molecule she wishes to track, the camera planner then performs the user-sensitive camera control algorithm shown in Figure 3. The planning-rendering loop begins with an update to the 3D environment model. This update may stem from either the user's manipulation of objects in the environment, a new state of a simulation, or both. The camera planner then makes three sets of decisions on each iteration of the loop to support the user's visualization preferences:

- *Shot modification determination:* The camera planner determines when a new shot should be selected based on the camera pace preference.
- *Camera shot selection:* If it has been determined that a new shot should be planned, the camera planner composes the new shot based on the viewpoint style preference.
- *Camera transition selection:* If a new shot will be presented, the camera planner determines whether to pan from the current shot to the new shot or whether to cut directly to the new shot based on the transition style preference.

When users state their preference for a faster or slower pace, UCAM changes shots more or less frequently. This is accomplished by the user model stipulating either a smaller or larger *MinimumShotDuration*.[5] Even if the camera planner opts to maintain the current distance, angle,

[5] In UCAM, the *MinimumShotDuration* for a *slow* pace is 35 frames and for a *fast* pace is 15 frames.

and elevation relative to the primary object, movement of the object in the environment may require it to modify the camera position. For example, if a red blood cell in a cardiovascular environment moves from one ventricle of the heart to another, the camera must track it.

User-sensitive camera shot selection decisions are made when the camera planner has determined that a new shot should be planned. UCAM composes new shots by selecting a new camera-subject distance (zooming in or out) and/or changing the camera's elevation. Because variability is of paramount importance in maintaining the user's interest, in most applications it is critical that a camera planner *not* employ a fixed sequence of shots. UCAM therefore employs a probabilistic shot selection algorithm that exploits probabilities on camera directives stipulated by the user model. Recall that more dramatic visualizations are produced with more frequent use of close-ups, near shots, and shots that are of lower elevation (Mascelli, 1965), while the converse holds for more informational visualizations. For variety, UCAM also varies the camera's view angle by selecting from one of the eight possible angles.[6] Together, these probabilistic decisions produce camera positions and orientations that reflect the user's viewpoint style preferences.

When UCAM selects a new shot, it enacts a transition that will achieve the user's visualization preferences. This decision is made by considering (1) camera transition directives specified in the user model and (2) the relative positions of the camera's current location/orientation with respect to the location/orientation it must travel to in order to make the new shot. Users who prefer gradual transitions will experience smoother visualizations through camera panning and tracking, while users who prefer jump transitions experience the environment with more frequent cutting from one shot to another. However, because of continuity concerns, jump transitions are often inappropriate: jumping from one shot to another shot which is only slightly different produces a very jarring effect (Mascelli, 1965). UCAM therefore compares the angular difference between the current and new shots with the *MinCutAngle* represented in the user model to make its decisions. If a user prefers jump transitions, the *MinCutAngle* threshold will be 60°, while the *MinCutAngle* for users preferring gradual transitions is ∞. The effect of these thresholds is as follows: for users who prefer gradual transitions, the camera will always transition with a combination of panning and tracking; for users who prefer jump transitions, a cut will be selected only if the angular difference exceeds the *MinCutAngle*.

Once the new shot is selected, UCAM computes the new coordinates for the camera. To accomplish this, it transforms the camera position from coordinates in the local coordinate system anchored at the object of interest to coordinates in the global XYZ coordinate system. The planning-rendering loop is completed by rendering the current scene in the environment from the new camera position. UCAM executes the body of the algorithm for each frame to produce a continuous interactive visualization as the user manipulates objects in the environment.

[6] To permit a cleaner evaluation, the version of UCAM employed in the evaluation described below assigns equal probabilities to each of the eight possible view angles. However, UCAM also has a sophisticated *occlusion avoidance system* that modifies the viewing angle to eliminate occlusions of the primary object by objects that come between the virtual camera and the primary object. As the user moves through the dynamic environment, the camera planner invokes the occlusion avoidance system to obtain an obstruction-free view of the primary object. The occlusion avoidance system was disabled during the evaluation.

4 An Implemented User-Sensitive Camera Planner

UCAM is a full-scale realtime implementation of the cinematographic user modeling framework.[7] To investigate UCAM's behavior, we constructed a navigable 3D environment testbed, CARPARK. In the CARPARK environment, users drive a sports car through a scenic maze of rectangular city blocks populated by tree-lined parks. On their journey they can pass other cars as they make their ways towards an enormous stop sign.

Suppose a user (User 1) prefers to experience his navigation of CARPARK at a slow pace that is less dramatic and with gradual transitions. As the user steers his car through the city, stopping and turning as he sees fit, UCAM creates a customized experience whose visualizations are tailored to his preferences. Incremental screen shots along his tour are shown in the left column of Figure 4. For the initial shot, UCAM depicts the car from a front-left angle at a high elevation and far viewing distance (a). In (b), the camera keeps the same front-left viewing angle, but gradually zooms to a medium distance shot. In (c), the camera has panned and tracked to a rear-right view to show the car rounding the corner. For the final shot in (d), the camera gradually pans and tracks around the car until it reaches the rear-left viewing angle as the driver brings the car to a halt.

Suppose that another user (User 2) prefers to experience her navigation of CARPARK at a faster pace that is more dramatic and includes jump transitions. For purposes of comparison, suppose that User 2 issues precisely the same navigation commands at precisely the same locations and with precisely the same timing as User 1.[8] The resulting experience is much more dramatic and seems much faster (Figure 4, right column). UCAM chooses more close-up-to-the-action views and sometimes sweeps in low to the ground to increase the sense of excitement. The initial shot in (a) shows the car from a front-left angle at low elevation and at a medium viewing distance. UCAM then cuts directly to the near front-left shot in (b), where the driver's car has just passed a parked car. As the driver's car rounds the corner, UCAM instructs the camera to zoom in for a close-up view from a front-left viewing angle and medium elevation. To transition from the shot in (b) to the shot in (c), although User 2 prefers jump transitions, UCAM employs a gradual zoom in and pan because the angular difference between the shot in (b) and the shot in (c) is too small for a cut without a jarring effect. For the final shot (d), the camera cuts to a low view of the car arriving at the stop sign.

5 Evaluation

To gauge the effectiveness of the cinematographic user modeling framework for creating customized visualization experiences in dynamic 3D environments, an empirical evaluation was conducted. The evaluation was designed to determine if (1) cinematographic user modeling could accurately represent users' preferences by producing visualizations that met their expectations, and (2) the resulting visualizations were clear and/or visually appealing.

---

[7] UCAM is implemented in C++ and employs the OpenGL graphics library for 3D rendering. It runs at 8 frames/second with 16 bits/pixel color on a Pentium 133 Mhz PC equipped with a 2D video board and 32 megabytes of memory. It consists of approximately 14,000 lines of code.

[8] Because CARPARK writes out a navigation script of a user's interactions, it is possible to replay a navigation and experience it with different visualization preferences.

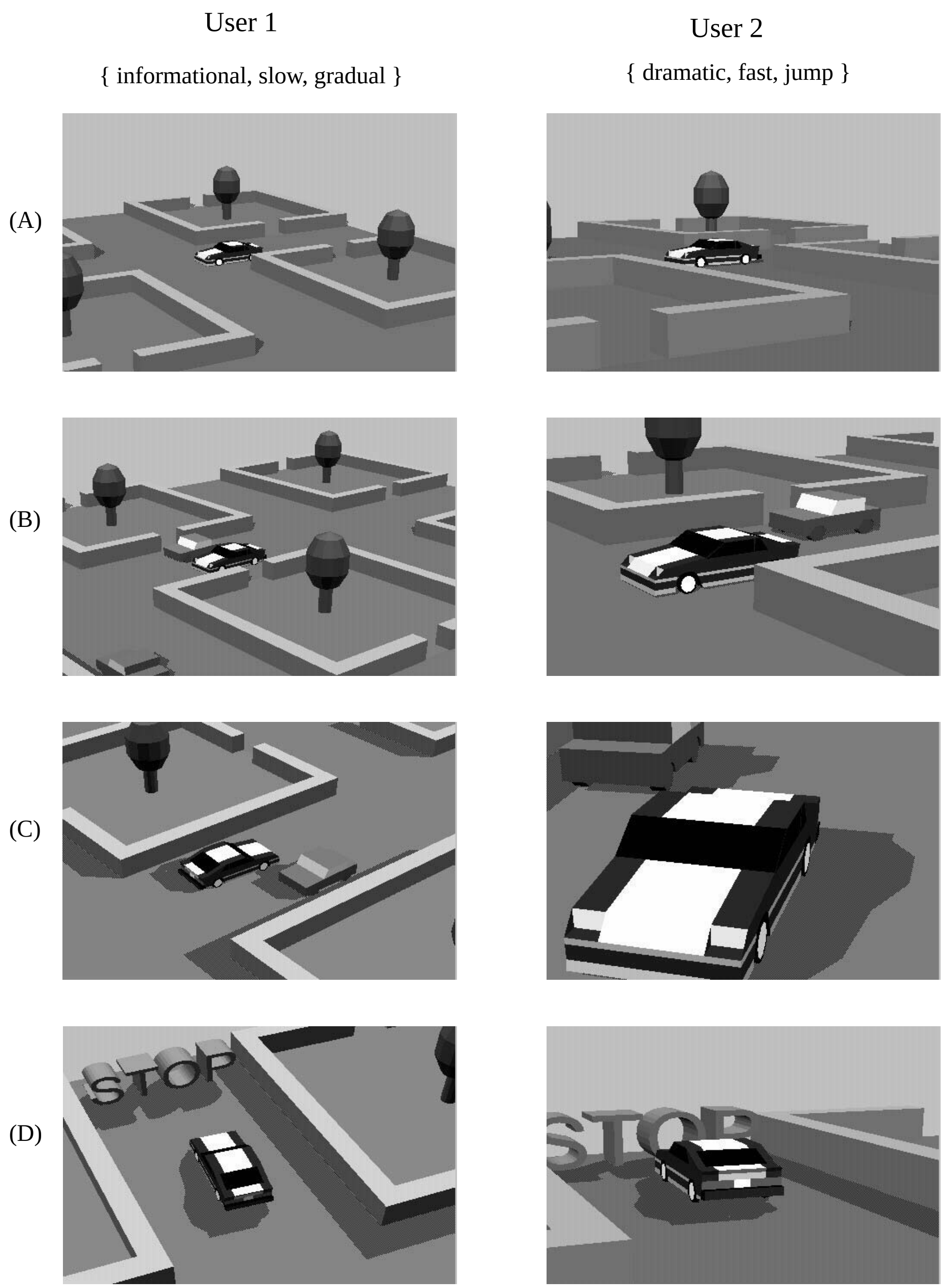

Figure 4. UCAM's customized tours of the CARPARK testbed.

The subjects of the study were 10 skilled computer users, all of whom were familiar with 3D computer animation and multimedia applications. To obtain a broad spectrum of responses, subjects were chosen from both artistic and technical backgrounds. Of the 10 subjects, 6 were graphic designers and 4 were computer scientists. Subjects interacted with two different versions of the CARPARK testbed. In the first version, users interacted with UCAM's director studio interface; in the second, they interacted with a direct camera control interface that presents a large number of options for specifying the same low-level camera control parameters automatically controlled by UCAM.

Subjects were given two sets of tasks to perform. In the first set of tasks, they were asked to serve as the director for a 3D movie, which was generated from a navigation script of the CARPARK testbed environment. They interacted with UCAM to specify two different sets of visualization preferences and observe the effects of these preferences as UCAM created the customized visualizations. They then repeated this process with the direct camera control interface. For their second set of tasks, users navigated through the CARPARK environment in realtime.

Results of the evaluation suggest that cinematic user modeling is an effective means for achieving customized camera control of dynamic 3D environments in realtime. Specific findings include the following: The dramatic viewpoint style was rated more interesting in 8 of the 11 times it was selected but found more difficult to follow in 6 of the 11. The informative viewpoint style was found easier to follow 7 of the 9 times when it was selected. The jump transition style was rated more interesting 6 of 9 times but was rated more difficult to follow in 5 of 9. The gradual transition style was found easier to follow 7 of the 10 of times it was selected. The slow camera pace was found easier to follow 5 of the 7 times it was selected, but was judged less interesting, while the fast camera pace was found to be more interesting 12 of 13 times and easy to follow 8 of 13 times.

The cinematographic user modeling approach was preferred by 8 of 10 for the 3D movie viewing directing task over the direct camera control. With the direct control, subjects reported being distracted by having to pay more attention to the camera controls. For the 3D navigation task, half of the subjects preferred cinematographic user modeling. Given the extreme simplicity of the CARPARK task, it is interesting to note that 8 of the 10 subjects (including those preferring direct control) reported being distracted by having to attend to both the camera control and the navigation in the direct control version.

Overall, the cinematographic user model system produced the expected visual result in 70 percent of the trials. Situations in which expectations were not met are attributable to the animation speed not being fast enough, the lack of establishing shots, and "line crossing" problems, i.e., the sudden apparent reversal of an object's direction of motion. Soon-to-be-released 3D graphic accelerators will provide a solution to the first concern by delivering an order-of-magnitude greater speed. As a result of the evaluation, UCAM has been extended to address the second and third concerns. In short, cinematographic user modeling enabled users to quickly specify their visualization preferences, it accurately modeled their preferences, and it permitted them to focus their attention on the task at hand.

6 Conclusion

Dynamic 3D environments offer great potential for a broad range of educational and scientific visualization tasks. We have proposed the cinematographic user modeling framework for dynamically customizing 3D environment experiences to users' visualization preferences. By considering these preferences, it plans camera positioning and orientation in realtime as users interact with objects in 3D environments. An empirical evaluation of an implemented domain-independent cinematographic user modeling system suggests that the approach can accurately model users' visualization preferences. This work represents a promising first step toward creating adaptive 3D environments. Perhaps the greatest challenge ahead lies in extending cinematographic user models to account for context-sensitivity to users' tasks. We will be investigating these issues in our future research.

References

André, E., and Rist, T. (1996). Coping with temporal constraints in multimedia presentation planning. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 142–147.

André, E., Finkler, W., Graf, W., Rist, T., Schauder, A., and Wahlster, W. (1993). WIP: The automatic synthesis of multi-modal presentations. In Maybury, M. T., ed., *Intelligent Multimedia Interfaces*. AAAI Press. chapter 3.

Butz, A. (1997). Anymation with CATHI. To appear in *Proceedings of the Innovative Applications of Artificial Intelligence Conference*.

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.

Karp, P., and Feiner, S. (1993). Automated presentation planning of animation using task decomposition with heuristic reasoning. In *Proceedings of Graphics Interface '93*, 118–127.

Lester, J. C., FitzGerald, P. J., and Stone, B. A. (1997). The pedagogical design studio: Exploiting artifact-based task models for constructivist learning. In *Proceedings of the Third International Conference on Intelligent User Interfaces*, 155–162.

Mackinlay, J., Card, S., and Robertson, G. (1990). Rapid controlled movement through a virtual 3D workspace. In *Proceedings of ACM SIGGRAPH '90*, 171–176.

Mascelli, J. (1965). *The Five C's of Cinematography*. Cine/Grafic Publications, Hollywood.

McKeown, K. R., Feiner, S. K., Robin, J., Seligmann, D., and Tanenblatt, M. (1992). Generating cross-references for multimedia explanation. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 9–15.

Millerson, G. (1994). *Video Camera Techniques*. Focal Press, Oxford, England.

Roth, S. F., Mattis, J., and Mesnard, X. (1991). Graphics and natural language as components of automatic explanation. In Sullivan, J. W., and Tyler, S. W., eds., *Intelligent User Interfaces*. New York: Addison-Wesley. 207–239.

Stone, B. A., and Lester, J. C. (1996). Dynamically sequencing an animated pedagogical agent. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 424–431.

van Mulken, S. (1996). Reasoning about the user's decoding of presentations in an intelligent multimedia presentation system. In *Proceedings of the Fifth International Conference on User Modeling*, 67–74.

Ware, C., and Osborn, S. (1990). Exploration and virtual camera control in virtual three dimensional environments. In *1990 Symposium on Interactive 3D Graphics*, 175–184.