StyleCam: Interactive Stylized 3D Navigation using Integrated Spatial & Temporal Controls

Nicholas Burtnyk[2,1], *Azam Khan*[1], *George Fitzmaurice*[1], *Ravin Balakrishnan*[2], *Gordon Kurtenbach*[1]

[1]Alias|wavefront
210 King Street East
Toronto, Ontario
Canada M5A 1J7
akhan, gf, gordo@aw.sgi.com

[2]Department of Computer Science
University of Toronto
Toronto, Ontario
Canada M5S 3G5
n.burtnyk@utoronto.ca, ravin@cs.toronto.edu

ABSTRACT

This paper describes StyleCam, an approach for authoring 3D viewing experiences that incorporate stylistic elements that are not available in typical 3D viewers. A key aspect of StyleCam is that it allows the author to significantly tailor what the user sees and when they see it. The resulting viewing experience can approach the visual richness and pacing of highly authored visual content such as television commercials or feature films. At the same time, StyleCam allows for a satisfying level of interactivity while avoiding the problems inherent in using unconstrained camera models. The main components of StyleCam are camera surfaces which spatially constrain the viewing camera; animation clips that allow for visually appealing transitions between different camera surfaces; and a simple, unified, interaction technique that permits the user to seamlessly and continuously move between spatial-control of the camera and temporal-control of the animated transitions. Further, the user's focus of attention is always kept on the content, and not on extraneous interface widgets. In addition to describing the conceptual model of StyleCam, its current implementation, and an example authored experience, we also present the results of an evaluation involving real users.

KEYWORDS: interaction techniques, camera controls, 3D navigation, 3D viewers, 3D visualization.

1. INTRODUCTION

Computer graphics has reached the stage where 3D models can be created and rendered, often in real time on commodity hardware, at a fidelity that is almost indistinguishable from the real thing. As such, it should be feasible at the consumer level to use 3D models rather than 2D images to represent or showcase various physical artifacts. Indeed, as an example, many product manufacturers' websites are beginning to supply not only professionally produced 2D images of their products, but also ways to view their products in 3D. Unfortunately, the visual and interactive experience provided by these 3D viewers currently fall short of the slick, professionally produced 2D images of the same items. For example, the quality of 2D imagery in an automobile's sales brochure typically provides a richer and more compelling presentation of that automobile to the user than the interactive 3D experiences provided on the manufacturer's website. If these 3D viewers are to replace, or at the very least be at par with, the 2D imagery, eliminating this difference in quality is critical.

The reasons for the poor quality of these 3D viewers fall roughly into two categories. First, 2D imagery is usually produced by professional artists and photographers who are skilled at using this well-established artform to convey information, feelings, or experiences, whereas creators of 3D models do not necessarily have the same established skills and are working in an evolving medium. However, this problem will work itself out as the medium matures.

The second issue is more troublesome. In creating 2D images a photographer can carefully control most of the elements that make up the shot including lighting and viewpoint, in an attempt to ensure that a viewer receives the intended message. In contrast, 3D viewers typically allow the user to interactively move their viewpoint in the scene to view any part of the 3D model.

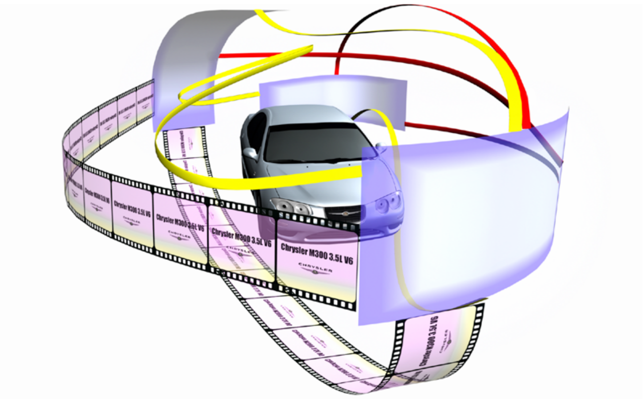

Figure 1. StyleCam authored elements

Published in *ACM CHI Letters, 4(2)*. p. 101-110. ACM UIST 2002 Symposium on User Interface Software & Technology

This results in a host of problems: a user may “get lost” in the scene, view the model from awkward angles that present it in poor light, miss seeing important features, experience frustration at controlling their navigation, etc. As such, given that the author of the 3D model does not have control over all aspects of what the user eventually sees, they cannot ensure that 3D viewing conveys the intended messages. In the worse case, the problems in 3D viewing produce an experience completely opposite to the authors intentions!

The goal of our present research is to develop a system, which we call StyleCam (Figure 1), where users viewing 3D models can be guaranteed a certain level of quality in terms of their visual and interactive experience. Further, we intend that the system should not only avoid the problems suggested earlier, but also have the capability to make the interactive experience adhere to particular visual styles. For example, with StyleCam one should be able to produce an interactive viewing experience for a 3D model of an automobile “in the style of” the television commercial for that same automobile. Ultimately, a high-level goal of our research is to produce interactive 3D viewing experiences where, to use an old saying from the film industry, “every frame is a Rembrandt”.

1.1. Author vs. User Control

Central to our research is differentiating between the concept of authoring an interactive 3D experience versus authoring a 3D model which the user subsequently views using general controls. If we look at the case of a typical 3D viewer on the web, in terms of interaction, the original author of the 3D scene is limited to providing somewhat standard camera controls such as pan, tumble and zoom. Essentially, control of the viewpoint is left up to the user and the author has limited influence on the overall experience.

From an author’s perspective this is a significant imbalance. If we view an interactive experience by cinematic standards, an author (or director) of a movie has control over several major elements: content/art direction, shading/lighting, viewpoint, and pacing. It is these elements that determine the overall visual style of a movie. However, in the interactive experience provided by current 3D viewers, by placing control of the viewpoint completely in the hands of the user, the author has surrendered control of two major elements of visual style: viewpoint and pacing.

Thus we desire a method for creating 3D interactive experiences where an author can not only determine the content and shading but also the viewpoints and pacing. However, intrinsic in any interactive system is some degree of user control and therefore, more accurately, our desire is to allow the author to have methods to significantly influence the viewpoints and pacing in order to create particular visual styles. Thus, we hope to strike a better balance between author and user control. In order to achieve this end, StyleCam incorporates an innovative interaction technique that seamlessly integrates spatial camera control with the temporal control of animation playback.

2. CONCEPTUAL MODEL

In order to provide author control or influence over viewpoints and pacing, we need a way for an author to express the viewpoints and the types of pacing they are interested in. Thus we have developed three main elements upon which our StyleCam approach is based.

1. Camera surfaces – an author-created surface used to constrain the users’ movement of the viewpoint

2. Animation clips – an author-created set of visual sequences and effects whose playback may be controlled by the user. These can include:
   - sophisticated camera movements.
   - Slates – 2D media such as images, movies, documents, or web pages.
   - visual effects such as fades, wipes, and edits.
   - animation of elements in the scene.

3. Unified UI technique – The user utilizes a single method of interaction (dragging) to control the viewpoint, animation clips, and the transitions between camera surfaces.

2.1. Camera Surfaces

In the motion picture industry a money-shot is a shot with a particular viewpoint that a director has deemed “important” in portraying a story or in setting the visual style of a movie. Similarly, in advertising, money-shots are those which are the most effective in conveying the intended message. We borrow these concepts of a money-shot for our StyleCam system. Our money-shots are viewpoints that an author can use to broadly determine what a user will see.

Further, we use the concept of a camera surface as introduced by Hanson and Wernert [19, 36]. When on a camera surface, the virtual camera’s spatial movement is constrained to that surface. Further, each camera surface is defined such that they incorporate a single money-shot. Figure 2 illustrates this notion.

Camera surfaces can be used for various purposes. A small camera surface can be thought of as an enhanced money-shot where the user is allowed to move their viewpoint a bit in order to get a sense of the 3-dimensionality of what they are looking at. Alternatively, the shape of the surface could be used to provide some dramatic camera movements, for example, sweeping across the front grill of a car. The key idea is that camera surfaces allow authors to conceptualize, visualize, and express particular ranges of viewpoints they deem important.

Intrinsic in our authored interactions is the notion that multiple camera surfaces can be used to capture multiple money-shots. Thus authors have the ability to influence a user’s viewpoint broadly, by adding different camera surfaces, or locally by adjusting the shape of a camera surface to allow a user to navigate through a range of viewpoints which are similar to a single particular money-shot. For example, as shown in Figure 2, camera surfaces at the front and rear of the car provide two authored viewpoints of these parts of the car in which a user can "move around a bit" to get a better sense of the shape of the front grille and rear tail design.

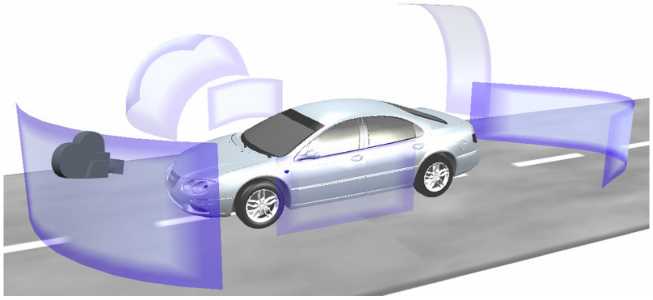

Figure 2. Camera surfaces. The active camera is at the money-shot viewpoint on the first camera surface.

The rate at which a user moves around on a camera surface (Control-Display gain) can dramatically affect the style of the experience. In order to allow an author some control over visual pacing, we provide the author with the ability to control the rate at which dragging the mouse changes the camera position as it moves across a camera surface. The intention is that increasing/decreasing this gain ratio results in slower/faster camera movement and this will influence how fast a user moves in the scene, which contributes to a sense of pacing and visual style. For example, if small mouse movements cause large changes in viewpoint this may produce a feeling of fast action while large mouse movement and slow changes in movement produce a slow, flowing quality. Figure 3 illustrates an example of variable control-display gain, where the gain increases as the camera gets closer to the right edge of the camera surface.

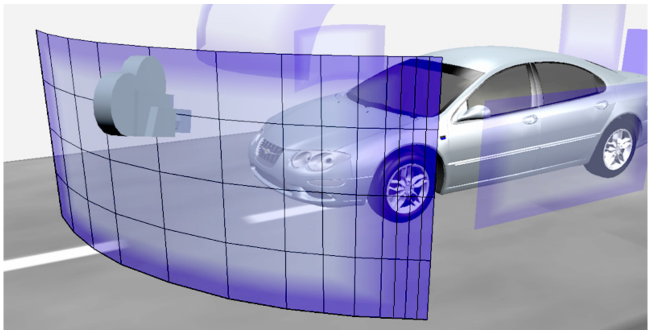

Figure 3. Variable control-display gain on a camera surface

2.2. Animation Clips

To support transitions between two camera surfaces, we use animation clips as illustrated in Figure 4. An animation clip can be thought of as a "path" between the edges of camera surfaces. When a user navigates to the edge of a camera surface, this triggers an animation. When the animation ends, they resume navigating at the destination camera surface. One obvious type of animation between the camera surfaces would simply be an automatic interpolation of the camera moving from its start location on the first camera surface to its end location on the second camera surface (Figure 4a). This is similar to what systems such as VRML do. While our system supports these automatic interpolated animations, we also allow for authored, stylized, animations. These authored animations can be any visual sequence and pacing, and are therefore opportunities for introducing visual style. For example, in transitioning from one side of the car to the other, the author may create a stylized camera animation which pans across the front of the car, while closing in on a styling detail like a front grille emblem (Figure 4b).

The generality of using animation clips allows the author the stylistic freedom of completely abandoning the camera-movement metaphor for transitions between surfaces and expressing other types of visual sequences. Thus animation clips are effective mechanisms for introducing slates — 2D visuals which are not part of the 3D scene but are momentarily placed in front of the viewing camera as it moves from one camera surface to another (Figure 4c). For example, moving from a view of the front of the car to the back of the car may be accomplished using a 2D image showing the name of the car. This mechanism allows the use of visual elements commonly found in advertising such as real action video clips and rich 2D imagery. In the computer realm, slates may also contain elements such as documents or webpages.

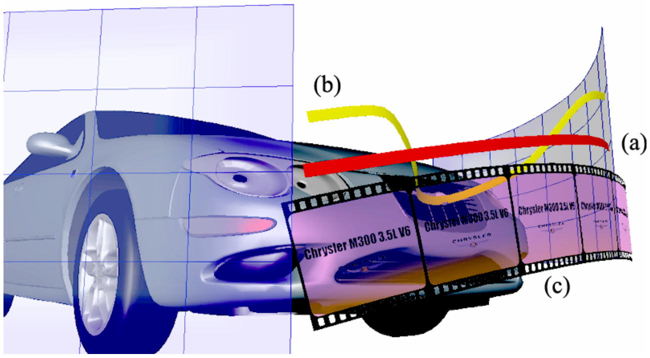

Figure 4. Three example animated transitions between camera surfaces. (a) automatic transition, (b) authored stylized transition, (c) slate transition.

The use of animation clips also allows for typical visual transitions effects such as cross fades, wipes etc.

In addition to using animation clips for transitions between camera surfaces, StyleCam also supports the animation of elements in the 3D scene. These scene element animations can occur separately or concurrently with transition animations. For example, while the animation clip for the visual transition may have the camera sweeping down the side of the car, an auxiliary animation may open the trunk to reveal cargo space.

The animation of scene elements can also be used to affect extremely broad changes. For example, entire scene transitions (similar to level changes in video games) may occur when a user hits the edge of particular camera surface.

At the author's discretion, temporal control of animation clips can either be under user control or uninterruptable.

Overall, in terms of visual expression, these varying types of animation clips allow an author to provide rich visual experiences and therefore significantly influence the pacing and style of a user's interaction.

2.3. Unified User Interaction Technique

While animation clips are effective for providing a means to move between camera surfaces and introduce visual styling elements, they also highlight the fundamental issue of arbitrating between user control and system control. At the heart of our system are two distinct types of behavior: 1) user control of the viewpoint, and 2) playback of animation clips. In other systems these two types of behavior are treated as distinct interactions. Specifically, the user must stop dragging the camera viewpoint, then click on something in the interface to trigger the animation, dividing their attention and interrupting the visual flow. In our system we wanted to use animations as a seamless way of facilitating movement between camera surfaces. Thus we needed a mechanism for engaging these animations that did not require an explicit mouse click to trigger animation. Ideally we wanted to leave the user with the impression that they "dragged" from one camera surface to another even though the transition between the surfaces was implemented as an authored animation.

These two behaviors are fundamentally different in that viewpoint control is spatial navigation and animation control is temporal navigation. From a user interaction standpoint, spatial behavior can be thought of as "dragging the camera" while temporal control is "dragging a time slider" or "scrubbing". Given this we required an interaction model which allowed these two types of drags to be combined together in a way that was well defined, controllable, and corresponded to user's expectations.

Figure 5, which uses the finite-state-machine model to describe interaction as introduced by [5, 26], shows the interaction model we developed. The key feature of this model is *the ability to transition back and forth from spatial to temporal control during a contiguous drag.* As a user drags the camera across a camera surface (State 1, Spatial Navigation) and hits the edge of the surface, a transition is made to dragging an invisible time slider (State 2, Temporal Navigation). As the user continues to drag, the drag controls the location in the animation clip, assuming that the author has specified the clip to be under user control. Upon reaching the end of the animation, a transition is made back to dragging the camera, however, on a different, destination camera surface (State 1).

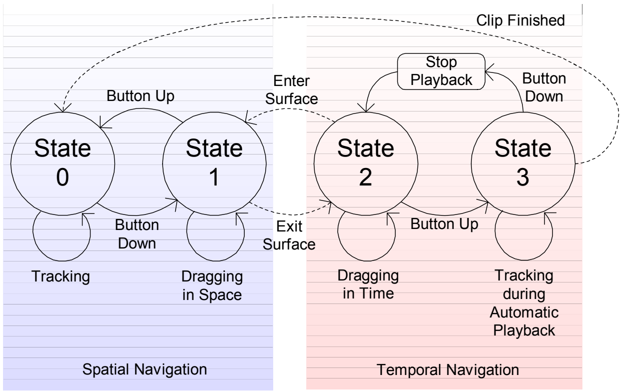

Figure 5. StyleCam interaction model.

The interaction model also handles a variety of reasonable variations on this type of dragging behavior. A user may stop moving when dragging an animation clip, thus pausing the animation. If, however, when in State 2 the user releases the mouse button during a drag, automatic playback is invoked to carry the user to the next camera surface (State 3). Should the user press the mouse button during this automatic playback, playback is stopped and temporal control by the user is resumed (return to State 2). We found in practice that this interaction design enhanced the user's feeling of being in control throughout the entire experience.

3. DESIGN RATIONALE

At first glance, it may appear that the incorporation of animation clips into StyleCam unnecessarily complicates its authoring and use. After all, without animated transitions, we would not have had to develop an interaction technique that blended between spatial and temporal control. Indeed, when we first began our research, our hope was to create a system that simply involved spatial control of a constrained camera.

Our first variation used a single camera surface that surrounded the 3D object of interest. The camera was constrained to remain normal to this single camera surface at all times. While this gave the author more control than using a simple unconstrained camera, we found that it was difficult to author a single camera surface that encompassed all the desirable viewpoints and interesting transitions between those viewpoints. In order to guarantee desirable viewpoints, we introduced the concept of money-shots that were placed on the single camera surface. The parameters of the camera were then determined based on its location on the camera surface and a weighted average of the surrounding money-shots. At this point, it was still difficult to author what the user would see when not directly on a money-shot. In other words, while money-shots worked well, the transitions between them worked poorly.

To address this problem of unsatisfactory transitions, we first replaced the concept of a single global camera surface with separate local camera surfaces for each money-shot.

Then, to define transitions between these local camera surfaces, we introduced the idea of animating the camera. This led to the use of the three types of animation clips as described earlier. Simply playing back the animation clips between camera surfaces gave users the sense that they lost control during this period. To maintain the feeling of continuous control throughout, we developed our integrated spatial-temporal interaction technique.

4. AN EXAMPLE EXPERIENCE

We illustrate how StyleCam operates by an example. Figure 6 illustrates the system components and how they react to user input, as well as screen shots of what the user actually sees. The user starts by dragging on a camera surface (position A). The path A-B shows the camera being dragged on the surface (spatial navigation). At B, the user reaches the edge of the camera surface and this launches an animation that will transition the user from B to E. The zig-zag path from B to D indicates that the user is scrubbing time on the animation (temporal navigation). Position C simply illustrates an intermediate point in the animation that gets seen three times during the interaction. At position D, the user releases the mouse button, whereupon the system automatically completes playing back the remainder of the animation at the authored pacing. At position E, the user enters another camera surface and resumes spatial navigation of the camera as shown by path E-F. When the user exits this camera surface at position F, another animation is launched that will transition the user to position J. Since the user releases the mouse button at position F, the animation from F to J is played back at the authored pacing. Since this animation is a slate animation, the intermediate shots at positions G, H, and I along the path F to J are of slates containing information on the car fading in and out as the camera pans over the top of the car. The net result of this StyleCam experience is a view of the car that is far more visually rich and influenced by an author who intends to convey a certain message, rather than using simple camera controls as is typical in current 3D viewers.

5. RELATED WORK

Much prior research has explored camera techniques for 3D virtual environments. Many of the techniques use a 2D mouse or stylus as an input device and introduce metaphors to assist the user. Perhaps the most ubiquitous metaphor, the cinematic camera, enables users to tumble, track and dolly a viewpoint. Various other metaphors have been explored by researchers, including orbiting and flying [32], through-the-lens control [18], points and areas of interests

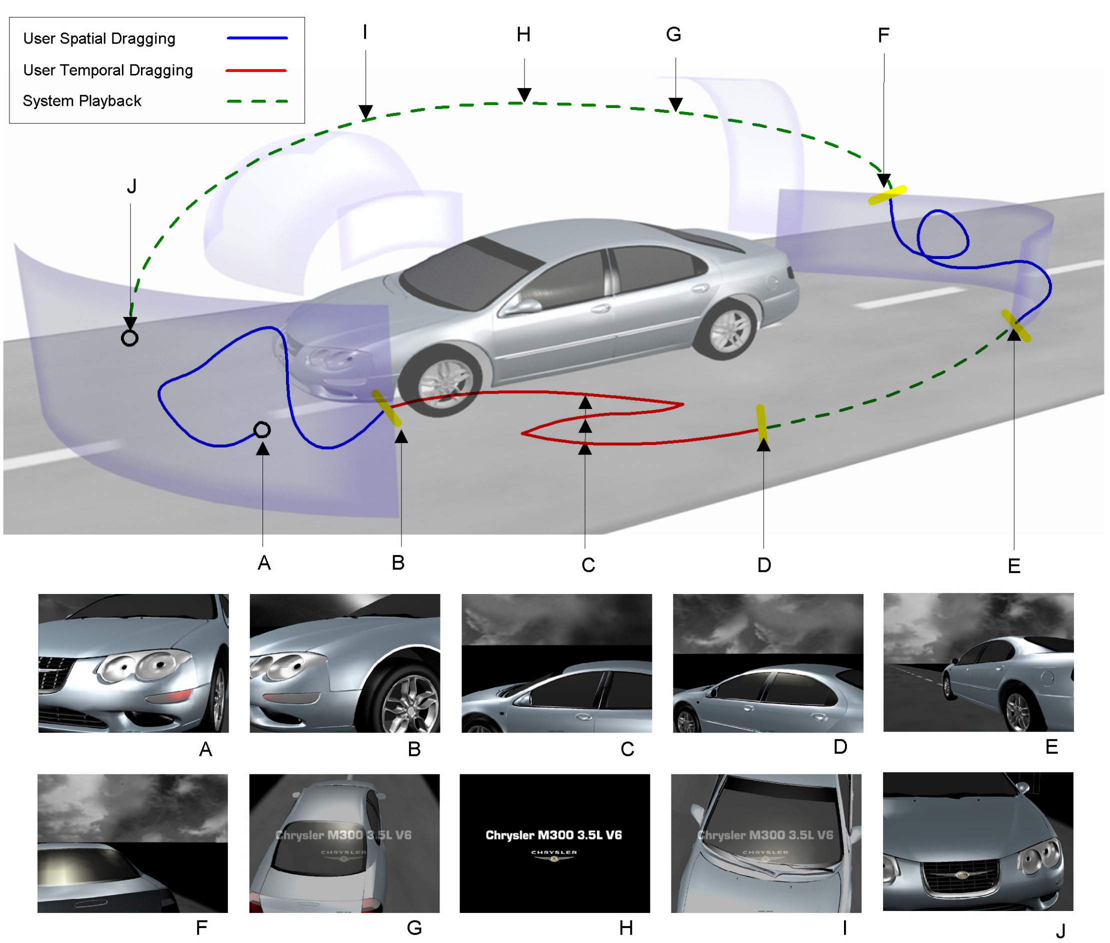

Figure 6. Example StyleCam experience. Top: system components and their reaction to user input. Bottom: what the user sees.

[22], using constraints [24, 29], drawing a path [21], two-handed techniques [1, 38], and combinations of techniques [30, 37]. Bowman et. al. present taxonomies and evaluations of various schemes [3, 4].

Other techniques involve automatic framing of the areas of interest as typically found in game console based adventure games which use a ”chase airplane” metaphor for a third person perspective. Systems that utilize higher degree-of-freedom input devices offer additional control and alternative metaphors have been investigated, including flying [7, 34], eyeball-in-hand [35], and worlds in miniature [31]. The major difference between this body of prior research and our work is that we attempt to give the author substantially more influence over the types of views and transitions between them as the user navigates in the virtual space.

Beyond techniques for navigating the scene, extra information can also be provided to aid navigation. These include global maps in addition to local views [12, 14], and various landmarks [9, 33]. Others have investigated integrating global and local views, using various distorted spaces including “fisheye” views [6, 15]. At present, in an attempt to keep the visual space uncluttered, our work does not have mechanisms for providing global information to the user, however, this is something we may incorporate as our system progresses.

Approaches which give the author more influence include guided tours where camera paths are prespecified for the end user to travel along. Galyean [17] proposes a “river analogy” where a user, on a metaphorical boat, can deviate from the guided path, the river, by steering a conceptual “rudder”. Fundamental work by Hanson and Wernert [19, 36] proposes “virtual sidewalks” which are authored by constructing virtual surfaces and specifying gaze direction, vistas, and procedural events (e.g., fog and spotlights) along the sidewalk. Our system builds upon the guided tour and virtual sidewalk ideas but differs by providing authoring elements that enable a much more stylized experience. Specifically, we offer a means of presenting 3D, 2D, and temporal media experiences through a simple, unified, singular user interaction technique that supports both spatial and temporal navigation.

Robotic planning algorithms have been used to assist or automatically create a guided tour of a 3D scene, in some cases resulting in specific behaviors trying to satisfy goals and constraints [10, 11]. Individual camera framing of a scene has been used to assist in viewing or manipulation tasks [27]. Rules can be defined for cameras to automatically frame a scene that follow cinematic principles such as keeping the virtual actors visible in the scene; or following the lead actor [20]. Yet another system [2] allows authors to define storyboard frames and the system defines a set of virtual cameras in the 3D scene to support the visual composition. This previous work assists in the authoring aspects by ceding some control to the system. Our work too involves some automatic system control, but we emphasize author control.

Image based virtual reality environments such as QuicktimeVR [8] utilize camera panning and zooming and allow users to move to defined vista points. The driving metaphor has also been used for navigating interactive video, as seen in the Movie-Maps system [23]. More recently, the Steerable Media project [25] for interactive television aims to retain the visual aesthetic of existing television but increase the level of user interactivity. The user is given the ability to control the content progression by seamlessly integrating video with augmented 2D and 3D graphics. While our goals are similar in that we hope to enhance the aesthetics of the visual experience, we differ in that our dominant media type is 3D graphics with augmented temporal media (animations and visual effects) and traditional 2D media (video, still images).

Lastly, we note that widely available 3D viewers or viewing technologies such as VRML, Cult3D, Shockwave, Viewpoint, Virtools, and Pulse3D, are becoming very popular but offer the standard camera controls of vista points, track, tumble, and zoom. We hope our explorations will ultimately assist in offering new experience and interaction approaches for future incarnations of these 3D viewers.

6. IMPLEMENTATION

StyleCam is implemented using Alias|wavefront’s MAYA 3D modeling and animation package. We use MAYA to author the 3D content to be visualized, the required camera surfaces, animation clips, and required associations between them. A custom written MAYA plugin allows the user to control their view of the 3D content based on their mouse input and the authored camera surfaces, animation clips, and associations.

The following description of our implementation assumes some knowledge of MAYA, although we have endeavoured to be as general as possible without sacrificing accuracy.

6.1. Authoring

First, money-shots are created by defining a MAYA camera with specific position, orientation, and other camera parameters. Then, a camera surface which intersects the position of the money-shot camera is defined by creating an appropriate non-trimmed NURBS surface within MAYA. To include an optional camera look-at point, the author simply defines a point in 3D space (using a MAYA locator). Finally, to make these components easily locatable by the plugin, they are grouped under a named MAYA node within its dependency graph.

Then, StyleCam animation clips are created as one would normally create animations in MAYA, using its TRAX non-linear animation editor. Animation clips at this stage are given meaningful, consistent, names in order to facilitate their identification later when associating them with events.

StyleCam allows the author to create scripts and associate them with events. Supported events are session startup, camera surface entry, camera surface exit, and camera surface timeout (Figure 7).

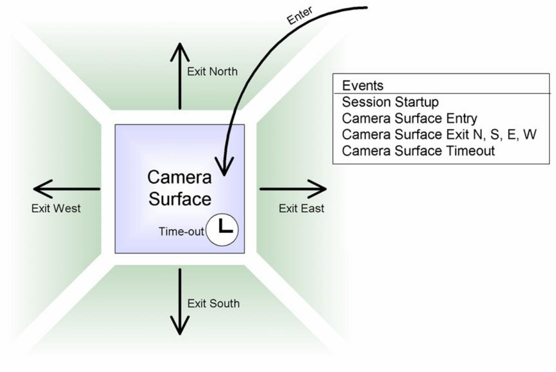

Figure 7. StyleCam events

The session startup event is triggered only once when the user initially begins using StyleCam to view a scene. Exit events are triggered when the user leaves a camera surface from one of four directions. Associated scripts can specify destination camera surfaces and types of transitions to be performed. Time-out events are triggered when the mouse is idle for a given duration while on a particular camera surface, and can be used to launch an automatic presentation. StyleCam's event and script mechanism provides for the use of logic to dynamically alter the presentation. For example, scripts can ensure that some surfaces are only visited once, while others are shown only after certain surfaces have already been visited.

6.2. Interaction

When the StyleCam plugin is activated, the first money-shot of the first camera surface is used as the initial view. If a look-at point is defined for this camera surface, the orientation of the user camera is set such that the camera points directly at the look-at point. Otherwise, the orientation is set to the normal of the camera surface at the money-shot viewpoint's position.

User's mouse movements and button presses are monitored by the StyleCam plugin. Mouse drags result in the camera moving along the current camera surface. Specifically, for a given mouse displacement (dx, dy), the new position of the camera on the camera surface (in *uv*-coordinates local to the camera surface) is given by

$$(u1,v1) = (u0,v0) + c*(\mathrm{dx}, \mathrm{dy})$$

where ($u0$, $v0$) is the last position of the camera, and c is the gain constant. If either the u or v coordinate of the resulting position is not within the range [0,1], the camera has left the current camera surface. At this point, the author-scripted logic is executed to determine the next step. First, the destination money-shot is resolved. Next, an appropriate transition is performed to move to the next camera surface.

We implement variable control-display gain on a camera surface (Figure 3) by varying the separation between the isoparms on the NURBS surface.

As shown in Figure 4, StyleCam supports three types of transitions: automatic, authored, and slate.

Automatic transitions are those that smoothly move the camera from one camera surface to another without requiring any authored animation clips. This is done by having the system perform quaternion [28] interpolation of camera orientation, combined quaternion and linear interpolation of camera position, and linear interpolation of other camera properties such as focal length. Using quaternion interpolation ensures smooth changes in orientation while defining a smooth arcing path for the position. At each time step in the transition, two quaternions representing the required fractional rotations of the position and orientation vectors of the camera are calculated and applied to the source vectors. In addition, the magnitude of the position vector is adjusted by linear interpolation between the source and destination position vector magnitudes. The result is a series of intermediate camera positions and orientations as Figure 8 illustrates.

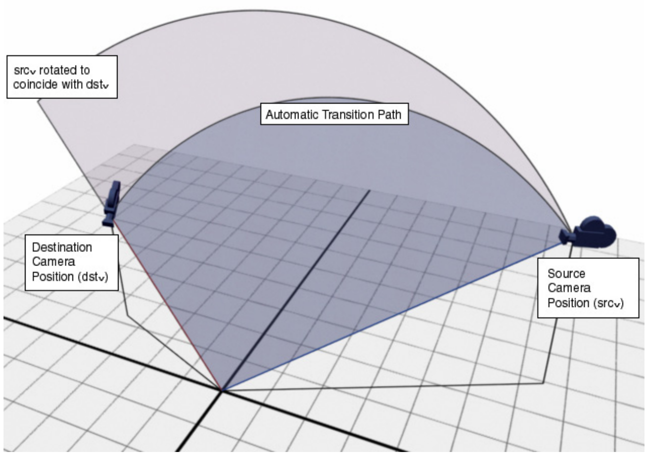
src$_v$ rotated to coincide with dst$_v$
Automatic Transition Path
Destination Camera Position (dst$_v$)
Source Camera Position (src$_v$)

Figure 8. Combined quaternion and linear interpolation

Authored transitions involve the playback of preauthored animation clips. This gives the author complete control over the user experience during the transition including the pacing, framing and visual effects.

Slate transitions are a special case of authored transitions. Used to present 2D media, slate transitions are authored by placing an image plane in front of the camera as it transitions between camera surfaces. Various visual effects can be achieved by using multiple image planes simultaneously and by animating transparency and other parameters of these image planes. While the slate transition is in progress, the camera is simultaneously being smoothly interpolated towards the destination camera surface. This essentially allows for a "soft" fade from a camera view, to a slate, and back, as Figure 9 illustrates.

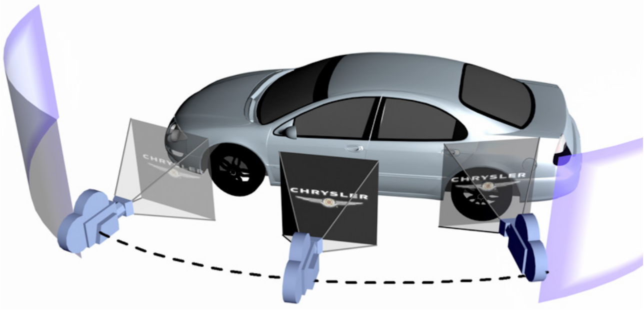

Figure 9. Slate transitions

StyleCam supports temporal control or "scrubbing" of animations. During navigation mode, the user's mouse drags control the camera's position on the camera surface. However, when the user moves off a camera surface into an animated transition, mouse drags control the (invisible) timeslider of the animation. Time is advanced when the mouse is dragged in the same direction that the camera exited the camera surface and reversed if the directions are also reversed. When the mouse button is released, the system takes over time management and smoothly ramps the time steps towards the animation's original playback rate.

Our present implementation supports scrubbing only for automatic transitions. Authored and slate transitions are currently uninterruptible. There is however no technical reason why *all* transitions cannot support scrubbing. In future versions we intend to give the author the choice of determining whether or not any given transition is scrubable. This is important since in some cases it may be desirable to force the animation to playback uninterrupted at a certain rate.

7. EVALUATION

We conducted an informal user study to get a sense of users' initial reactions to using StyleCam. Seven participants, three of whom had experience with 3D graphics applications and camera control techniques, and four who had never used a 3D application or camera controls, were asked to explore a 3D car model using StyleCam. In order to ensure the study resembled our intended casual usage scenario, we gave participants only minimal instructions. We explained the click-and-drag action required to manipulate the camera, a brief rationale for the study, and to imagine they were experiencing an interactive advertisement for that car. We did not identify the various components (camera surfaces, animated transitions, etc) nor give any details on them. This was deliberately done so that the participants could experience these components in action for themselves and give us feedback without knowing in advance of their existence.

One very promising result was that none of the participants realized that they were switching between controlling the camera and controlling the time slider on the animations. They felt that they had the same type of control throughout, indicating that our blending between spatial and temporal control worked remarkably well. Also the simplicity of the interaction technique – essentially a single click and drag action – was immediately understood and usable by all our users.

Another reaction from all the participants was that, to varying degrees, they sometimes felt that they were not in control of the interaction when the uninterruptable animations occurred. This was particularly acute when the information in the animations seemed unrelated to their current view. In these cases, participants indicated that they had no idea what triggered these animations and were often annoyed at the sudden interruptions. However when the information was relevant the interruptions were not as annoying and often actually appreciated. In some cases participants indicated that they would have liked to be able to replay the animation or to have it last longer. This highlights the importance of carefully authoring the intermingling of uninterruptable animations with the rest of the interaction experience.

Participants also indicated that they would have liked the ability to click on individual parts of the car model in order to inspect them more closely. This request is not surprising since we made no effort in our current implement to support pointing. However, we believe that in future research StyleCam could be extended to include pointing.

As we expected, all the participants with prior 3D graphics camera experience stated that they at times would have liked full control of the camera, in addition to the constrained control we provided. Participants without this prior experience, however, did not ask for this directly although they indicated that there were some areas of the car model that they would have liked to see but could not get to. However, this does not necessarily imply full control of the camera is required. We believe that this issue can be largely alleviated at the authoring phase by ascertaining what users want to see for a particular model and ensuring that those features are accessible via the authored camera surfaces. Interestingly, the participant with the most 3D graphics experience commented that the automatic transitions and smooth camera paths during those transitions were very good and that "for those who don't know 3D and stuff, this would be very good"!

8. DISCUSSION & CONCLUSIONS

Central to our StyleCam system is the integration of spatial and temporal controls into a single user interaction model. The implications of this interaction model go far beyond a simple interaction technique. The blending of spatial and temporal control presents a completely new issue that an author needs to understand and consider when creating these interactive visual experiences. As evident from the comments of our users, temporal control can feel very much like spatial control even when scrubbing backwards in an animation when the animation consists of moving the viewing camera around the central object of interest. However, if the animation is not around the central object of interest, for example in some of our slate animations, temporal control can produce very different sensations. These include the feeling of moving backwards in time, interruption of a well paced animation, jarring or ugly visuals, and sometimes even nonsensical content.

As a result, the author needs to be extremely cognizant of these artefacts and make design decisions as to when and where to relinquish control - and how much control - to the user. At one extreme, the author can specify that certain animations are completely uninterruptible by the user. In the experience we authored for our user study, we included several of these types of transitions. As discussed earlier, whether users favored this depended heavily on the content. In other words, in some cases, as authors, we did not make the right decision. Further improvements could include partially interruptable animations. For example, we may not allow movement backwards in time but allow the user to control the forward pacing. This will largely solve the nonsensical content problem but may still result in occasionally jarring visuals.

If we intend to support these various types of control, we must also be able to set the users' expectations of what type of control they have at any given time. It is clear that the current StyleCam switching between spatial and temporal control without any explicit indication to the user that a switch is happening works in most cases. In the cases where it fails, either the visual content itself should indicate what control is possible, or some explicit mechanism is required to inform the user of the current or upcoming control possibilities. In addition to the obvious solution of using on-screen visual indicators (e.g., changing cursors) to indicate state, future research could include exploring "hint-ahead" mechanisms that indicate upcoming content if the user chooses to stay on their current course of travel. For example, as the user reaches the edge of a camera surface, a "voice-over" could say something like "now we're heading towards the engine of the car". Alternatively, a visual "signpost" could fade-in near the cursor location to convey this information. These ideas coincide with research that states that navigation routes must be discoverable by the user [16].

It is very clear from our experiences with StyleCam that the user's viewing experience is highly dependent on the talent and skill of the author. It is likely that skills from movie making, game authoring, advertising, and theme park design would all assist in authoring compelling experiences. However, we also realize that authoring skills from these other genres do not necessarily directly translate due to the unique interaction aspects of StyleCam.

While StyleCam has the appropriate components for creating compelling visual experiences, it is still currently a research prototype that requires substantial skills with MAYA. We envision a more author-friendly tool that is based on the conceptual model of StyleCam.

Some future avenues that we intend to explore include supporting soundtracks, extensions to enable pointing to elements in the 3D scene, and mechanisms for authoring animation paths using alternate techniques such as Chameleon [13].

Finally, it is important to note that StyleCam is not limited to product or automobile visualization. Other domains such as visualization of building interiors and medical applications could also utilize the ideas presented in this paper. Figures 10, 11, and 12 illustrate some examples.

ACKNOWLEDGEMENTS

We thank Scott Guy and Miles Menegon for assistance in figure and video creation.

REFERENCES

1. Balakrishnan, R., & Kurtenbach, G. (1999). Exploring bimanual camera control and object manipulation in 3D graphics interfaces. *ACM CHI 1999 Conference on Human Factors in Computing Systems*. p. 56-63.

2. Bares, W., McDermott, S., Boudreaux, C., & Thainimit, S. (2000). Virtual 3D camera composition from frame constraints. *ACM Multimedia*. p. 177-186.

3. Bowman, D.A., Johnson, D.B., & Hodges, L.F. (1997). Travel in immersive virtual environments. *IEEE VRAIS'97 Virtual Reality Annual International Symposium*. p. 45-52.

4. Bowman, D.A., Johnson, D.B., & Hodges, L.F. (1999). Testbed environment of virtual environment interaction. *ACM VRST'99 Symposium on Virtual Reality Software and Technologies*. p. 26-33.

5. Buxton, W., ed. Three-state model of graphical input. Human-computer interaction - INTERACT'90, ed. D. Diaper. 1990, Elsevier Science Publishers B. V. (North-Holland): Amsterdam. 449-456.

6. Carpendale, M.S.T., & Montagnese, C.A. (2001). A framework for unifying presentation space. *ACM UIST'2001 Symposium on User Interface Software and Technology*. p. 61-70.

7. Chapman, D., & Ware, C. (1992). Manipulating the future: predictor based feedback for velocity control in virtual environment navigation. *ACM I3D'92 Symposium on Interactive 3D Graphics*. p. 63-66.

8. Chen, S.E. (1995). QuickTime VR: An image-based approach to virtual environment navigation. *ACM SIGGRAPH'95 Conference on Computer Graphics and Interactive Techniques*. p. 29-38.

9. Darken, R., & Sibert, J. (1996). Wayfinding strategies and behaviours in large virtual worlds. *ACM CHI'96 Conference on Human Factors in Computing Systems*. p. 142-149.

10. Drucker, S.M., Galyean, T.A., & Zeltzer, D. (1992). CINEMA: A system for procedural camera movements. *ACM Symposium on Interactive 3D Graphics*. p. 67-70.

11. Drucker, S.M., & Zeltzer, D. (1994). Intelligent camera control in a virtual environment. *Graphics Interface*. p. 190-199.

12. Elvins, T., Nadeau, D., Schul, R., & Kirsh, D. (1998). Worldlets: 3D thumbnails for 3D browsing. *ACM CHI'98 Conf. on Human Factors in Computing Systems*. p. 163-170.

13. Fitzmaurice, G.W. (1993). Situated information spaces and spatially aware palmtop computers. *Communications of the ACM, 36*(7). p. 38-49.

14. Fukatsu, S., Kitamura, Y., Masaki, T., & Kishino, F. (1998). Intuitive control of bird's eye overview images for navigation in an enormous virtual environment. *ACM VRST'98 Sympoisum on Virtual Reality Software and Technology*. p. 67-76.

15. Furnas, G. (1986). Generalized fisheye views. *ACM CHI 1986 Conference on Human Factors in Computing Systems*. p. 16-23.

16. Furnas, G. (1997). Effective view navigation. *ACM CHI'97 Conference on Human Factors in Computing Systems*. p. 367-374.

17. Galyean, T.A. (1995). Guided navigation of virtual environments. *ACM I3D'95 Symposium on Interactive 3D Graphics*. p. 103-104.

18. Gliecher, M., & Witkin, A. (1992). Through-the-lens camera control. *ACM SIGGRAPH' Conf. on Computer Graphics and Interactive Techniques*. p. 331-340.

19. Hanson, A.J., & Wernet, E. (1997). Constrained 3D navigation with 2D controllers. p. 175-182.

20. He, L., Cohen, M.F., & Salesin, D. (1996). The virtual cinematographer: a paradigm for automatic real-time camera control and directing. *ACM SIGGRAPH'96 Conference on Computer Graphics and Interactive Techniques*. p. 217-224.

21. Igarashi, T., Kadobayashi, R., Mase, K., & Tanaka, H. (1998). Path drawing for 3D walkthrough. *ACM UIST 1998 Symposium on User Interface Software and Technology*. p. 173-174.

22. Jul, S., & Furnas, G. (1998). Critical zones in desert fog: aids to multiscale navigation. *ACM Symposium on User Interface Software and Technology*. p. 97-106.

23. Lippman, A. (1980). Movie-maps: an application of the optical videodisc to computer graphics. *ACM SIGGRAPH'80 Conference on Computer Graphics and Interactive Techniques*. p. 32-42.

24. Mackinlay, J., Card, S., & Robertson, G. (1990). Rapid controlled movement through a virtual 3D workspace. *ACM SIGGRAPH 1990 Conference on Computer Graphics and Interactive Techniques*. p. 171-176.

25. Marrin, C., Myers, R., Kent, J., & Broadwell, P. (2001). Steerable media: interactive television via video synthesis. *ACM Conference on 3D Technologies for the World Wide Web*. p. 7-14.

26. Newman, W. (1968). A system for interactive graphical programming. *AFIPS Spring Joint Computer Conference*. p. 47-54.

27. Phillips, C.B., Badler, N.I., & Granieri, J. (1992). Automatic viewing control for 3D direct manipulation. *ACM Symposium on Interactive 3D Graphics*. p. 71-74.

28. Shoemake, K. (1985). Animating rotation with quartenion curves. *ACM SIGGRAPH Conf Computer Graphics & Interactive Techniques*. p. 245-254.

29. Smith, G., Salzman, T., & Stuerzlinger, W. (2001). 3D Scene manipulation with 2D devices and constraints. *Graphics Interface*. p. 135-142.

30. Steed, A. (1997). Efficient navigation around complex virtual environments. *ACM VRST'97 Conference on Virtual Reality Software and Technology*. p. 173-180.

31. Stoakley, R., Conway, M., & Pausch, R. (1995). Virtual reality on a WIM: Interactive worlds in miniature. *ACM CHI 1995 Conference on Human Factors in Computing Systems*. p. 265-272.

32. Tan, D., Robertson, G., & Czerwinski, M. (2001). Exploring 3D navigation: combining speed-coupled flying with orbiting. *ACM CHI'2001 Conference on Human Factors in Computing Systems*. p. 418-425.

33. Vinson, N. (1999). Design guidelines for landmarks to support navigation in virtual environments. *ACM CHI'99 Conference on Human Factors in Computing Systems*. p. 278-285.

34. Ware, C., & Fleet, D. (1997). Context sensitve flying interface. *ACM I3D'97 Symposium on Interactive 3D Graphics*. p. 127-130.

35. Ware, C., & Osborne, S. (1990). Exploration and virtual camera control in virtual three dimensional environments. *ACM I3D'90 Symposium on Interactive 3D Graphics*. p. 175-183.

36. Wernert, E.A., & Hanson, A.J. (1999). A framework for assisted exploration with collaboration. *IEEE Visualization*. p. 241-248.

37. Zeleznik, R., & Forsberg, A. (1999). UniCam - 2D Gestural Camera Controls for 3D Environments. *ACM Symposium on Interactive 3D Graphics*. p. 169-173.

38. Zeleznik, R., Forsberg, A., & Strauss, P. (1997). Two pointer input for 3D interaction. *ACM I3D Symposium on Interactive 3D Graphics*. p. 115-120.