Automated Presentation Planning of Animation Using Task Decomposition with Heuristic Reasoning

Peter Karp and Steven Feiner
Department of Computer Science
Columbia University
New York, N.Y. 10027
karp@cs.columbia.edu
feiner@cs.columbia.edu

ABSTRACT

We describe ESPLANADE (Expert System for PLANning Animation, Design, and Editing), a knowledge-based animation presentation planner. ESPLANADE receives as input a script generated by a separate action planner and a set of communicative goals that specify what the animation is to communicate to the viewer. The presentation plan that ESPLANADE creates is a complete description of every frame of an animation and includes viewing specifications, viewport dimensions, and object properties. Instead of treating animation as a flat sequence of frames, our approach is to use a top-down task decomposition strategy with heuristic reasoning to build a frame-based semantic representation of an animation. The structure built by ESPLANADE is a hierarchy of sequences, scenes, and shots, based on a representation used by filmmakers. This approach allows the planning to be performed with contextual information that is inferred while building the semantic representation and results in a better animation. We demonstrate our system with a detailed example of a generated animation.

KEYWORDS: Presentation planning, computer animation, knowledge-based graphics, visual language, film grammar.

INTRODUCTION

In some situations it is appropriate to allow a user of a graphics system to control a virtual camera and roam the world freely. However, there are other situations where specific information must be communicated to the user. If the information involves explaining spatial, temporal, or causal relations then a well designed animation can be much more effective.

Much of the recent work in computer animation has concentrated on the planning of the actions to be depicted (e.g. [1, 20, 31, 3]). However, a number of other tasks are vital to designing a quality animation. *Presentation planning* involves selecting the actions and the order they will be shown, and specifying viewing specifications, viewport dimension and effects. Presentation planning of animation is a difficult task. Important details must be shown in context and unambiguously. Actions that are relevant to the presentation cannot be left out, obscured by objects or otherwise lost in the background. A viewer should not be disoriented or confused by a sudden change of viewpoint. Presentation planning also involves making decisions about when to show an action in an inset, split screen or full viewport, and when to use editing transitions such as wipes and fades.

We are developing ESPLANADE (Expert System for PLANning Animation, Design and Editing), a presentation planner with a knowledge-base of filmmaking techniques. Our research is motivated by the realization that films and animations are constructs of a complex yet coherent semantic system that has evolved rapidly during cinema's ninety-year history [4, 28]. Many of the visual elements of a film (e.g., dissolves and close-ups) have developed an accepted established usage. Others, like the iris mask effect, once popular in silent films, have fallen into disuse. We are using the semantics of the visual elements of film, encoded in heuristics, to direct ESPLANADE along productive paths during the automatic construction of an animation.

ESPLANADE takes a script generated by an action planner and a set of communicative goals and creates a coherent animated presentation automatically. The script contains the sort of information that can be generated by many planning systems [25]. The communicative goals provide ESPLANADE with information about the intent [26] of the animation. The ultimate objective of ESPLANADE is to generate computer graphic animation of sufficient quality that it can be used for presentation and instruction by a user.

In an earlier paper [16] we presented a brief overview of some of the techniques used in the filming and editing of movies, and we discussed some of the research issues involved in developing systems that could apply these techniques automatically. That paper introduced an early version of ESPLANADE and presented one of its animations. In contrast, this paper concentrates on the strategy and reasoning performed by the current version of ESPLANADE. We describe a novel hierarchical task decomposition strategy and the heuristic reasoning it uses. We also discuss the internal structures, including an object-oriented class hierarchy of film concepts. One major addition to the first version of ESPLANADE is the use of frame-based reasoning over a hierarchical representation. Since the underlying representation is richer than before, the range of inferencing we are able to perform has expanded and the quality of the animation produced by ESPLANADE has improved. We also discuss the semantics of several elements of movies and the heuristics that result from an understanding of the semantics. We demonstrate the use of heuristic reasoning with task decomposition for presentation planning with sample frames from an animation generated by ESPLANADE.

RELATED WORK

Several researchers are focusing on the problem of action planning to automate the generation of animation from high-level descriptions [10, 8, 29]. Kahn's [15] ANI is an early example of using inferencing in presentation planning. ANI plans the movement of characters using information from an input description of the dramatic actions in a story and generates a 2D symbolic animation. Another system that focuses on the problems of presentation planning was developed by Ridsdale [21]. It has an expert system that uses theater directing heuristics to plan a character's motion based on a behavioral model of the character, the actions to be performed, other actors in the environment and theatrical rules used to direct the audience's attention.

Rubin's Constraint-Based Editor [22] builds movies from a set of prerecorded videotaped shots. The material is recorded using standard video equipment and transferred to laser disks. Each shot is described manually and entered into a database. Rubin's automated editing system assembles a selection of shots, applying continuity rules to form a coherent visual narrative from a subset of the shot database. The system is restricted to static camera shots and does not make use of camera position information. Several recent papers address the problem of automatic viewpoint placement of virtual cameras [9, 18, 17, 14].

Our work on ESPLANADE relies in part on the results of our previous research in automating the design of static images, including the design of sequences of pictures of 3D actions in APEX [12], and the automatic generation of composite pictures that include multiple viewports in IBIS [27]. IBIS supports user-controlled animation in which the user changes the viewing specification while the system attempts to maintain the input communicative goals of what is essentially a single shot. In contrast, ESPLANADE designs a structured animation containing potentially many shots organized in a standard cinematic hierarchy of sequences, scenes and shots.

ESPLANADE (Expert System for PLANning Animation, Design and Editing)

Our testbed for experimenting with automated presentation planning, shown in Figure 1, includes several modules, each of which is currently run in sequence after the previous one has completed its task. The *action planner* generates a presentation-independent plan of the actions in the domain, which we call a *script*. The input to the action planner is a set of action goals that must be achieved in the domain. The type of information it generates is representative of the output of many planners, (e.g., see [25]) and includes information used in the planning process such as causal relationships between actions. We have built several simple planners that generate this input for ESPLANADE.

ESPLANADE's input also includes a set of communicative goals indicating information about the actions and objects in the domain that must be included in the presentation. ESPLANADE currently supports communicative goals that specify that an action is to be described or that the causal relationship between two actions is to be emphasized. For example, a communicative goal can specify that it is important to show that *turning knob A caused door B to open*. One way to satisfy the goal is to show the following actions in consecutive shots: (1) door *B* before it opens, (2) knob *A*, (3) door *B* opening, (4) the act of turning knob *A*.

ESPLANADE creates a presentation plan ready for viewing. The finished plan includes fully specified viewing parameters, the selected domain actions, visual effects and viewports. The *animation viewer* is used to play back and render the animation since ESPLANADE does not generate its plan in real-time. The *domain database* uses an object-oriented class hierarchy to describe the objects in the domain. The database includes component information that describes each object and its separate parts and can include physical properties, such as geometry, weight, temperature, or other properties relevant to the domain. The component information also specifies the operations that an object can perform. The action planner, ESPLANADE and the animation viewer each have access to the database.

Task Decomposition and Heuristic Reasoning

ESPLANADE uses hierarchical task decomposition in which a complex planning problem is decomposed and solved in a hierarchy of different abstraction levels [24]. Hierarchical planners generate a complete plan at a single level before beginning work on the next level. The goals become more specific when the planner moves to a lower level. An example of hierarchical task decomposition for motion planning is described by Zeltzer [32].

We use the hierarchical structure of a film to define the different abstraction spaces for our planner (see Figure 2). The structure of a film, conventionally imposed by filmmakers, is a multi-level hierarchy [4] of *sequences*, *scenes*, and *shots*. A *sequence* is a set of actions that oc-

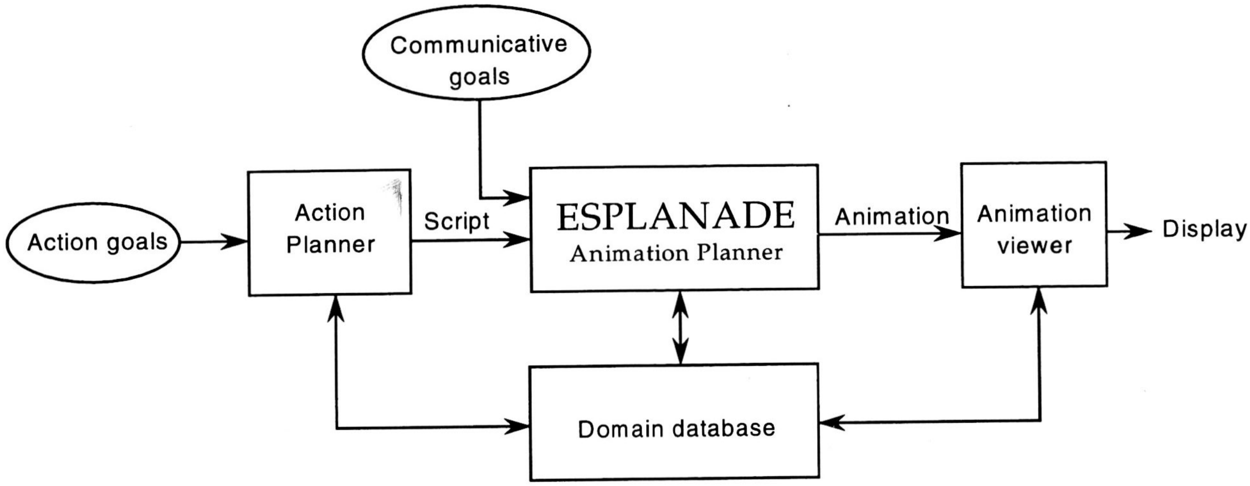

Figure 1: ESPLANADE architecture

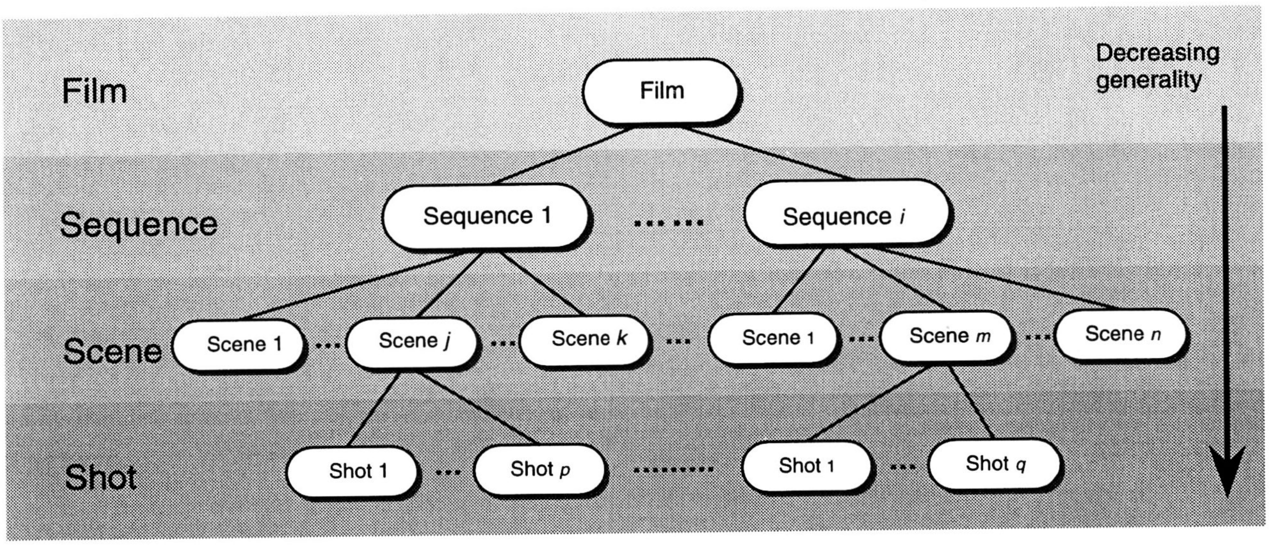

Figure 2: Film structure

cur in the same location. A *scene* is a part of a sequence that shows a set of related actions occurring during the same time period and within the same locale. A scene is composed of *shots*, the basic semantic unit of films. Each shot is a view of one or more actions made with a single camera.

On the *film* level, ESPLANADE determines the subject of the animation and the objects that are in it. On the *sequence* level, ESPLANADE determines the sequences that are part of the film. On the *scene* level, ESPLANADE determines the scenes that are part of a sequence and selects a scene structuring formalism. On the *shot* level, ESPLANADE determines the content, viewing specifications, transitions, and viewport of each shot.

The hierarchical planning strategy allows ESPLANADE to delay low-level decisions until more information from high-level goals is available. A non-hierarchical planner (e.g. [13]) may spend a lot of computation on details before solving goals that are critical to the solution. For example, a non-hierarchical presentation planner may attempt to solve for camera placement before selecting the actions to show. Hierarchical planners use heuristics or rules to guide the selection of an operator (from a set of operators) that will transform the current state into the next state. ESPLANADE uses heuristics derived from generally accepted filmmaking practices [5, 2, 19]. The use of filmmaking heuristics in ESPLANADE reduces the planning search space by early elimination of solution paths that should not be pursued. The heuristics are used to select the various transitions (*cut, fade, wipe,* and *dissolve*), viewport layout (*full, inset,* and *split*), and viewing specifications.

Film Level Processing

ESPLANADE builds a frame-based, hierarchical representation of an animation. The hierarchy has a single root node: the film. This represents the set of goals that the entire finished animation achieves. The hierarchy provides a source of information about each sequence, scene and shot as ESPLANADE extends the hierarchy. When planning is complete, the leaves of the hierarchy contain information about each shot in the animation. This information is output to a file that specifies on a frame-by-frame basis all viewport positions and dimensions, viewing specifications, fade levels, and object transformations.

Sequence Level Processing

The level below the root node is the sequence level. ESPLANADE analyzes the action planner's script to find a set of contiguous actions that occur in the same location. These actions are grouped together into a sequence and a new sequence node is entered beneath the root. Below the sequence level is the scene level. A scene is defined in ESPLANADE as a set of actions occurring during the same period and within the same locale. ESPLANADE analyzes the actions of the sequence to group actions together into scenes.

Scene Level Processing

Filmmakers structure scenes in a variety of ways. For example, Sharff [28] has identified seven basic models of scene structure used in commercial films:

Separation Fragmentation of scene into single images in alternation (e.g. A, B, A, B, A, B, . . .) used to establish relations between objects shown separately on the screen.

Parallel Action Two or more independent narrative lines running simultaneously and presented by alternation between scenes.

Slow Disclosure The gradual introduction of context beginning with a close-up shot.

Familiar Image A stabilizing image periodically reintroduced without variations.

Moving Camera Use of a moving camera to change view instead of cuts.

Multi-angularity A series of shots of contrasting angles and compositions (including reverse and mirror images).

Master Shot Discipline The opposite of slow disclosure; begins and ends with an establishing shot followed by closer shots.

We have implemented three scene-structuring techniques suitable for instructional animation that is intended to describe physical actions in a clear and concise fashion. This may require using inset windows, for example, which are not often found in commercial films. The selection of a scene structuring technique is based on the filmmaker's preferences and the initial communicative goals of the scene. One scene-structuring technique we implemented is the basic master shot discipline. Another, a composite of several techniques, was designed by an advising filmmaker [30] for use in ESPLANADE; we call it the *how-to structure* since it is useful for showing how to perform a task. This technique shows the action in a medium shot and uses an inset viewport of an important object that must be visible at all times. The third we call the *causal structure* and it uses a form of the parallel action structure listed by Sharff to show the causal relation between two actions. The structure is designed so any change of state appears on screen. If the two actions are essentially simultaneous (e.g. pressing a light switch and the light turning on) then the resulting shots will contain a temporal overlap with each shot showing different views of the same time period.

Shot Level Processing

Each scene-structuring technique is associated with a template that defines the *scope*[1] and image content of the shots that will form the scene. For example, a scene using the master shot discipline uses close-ups to show objects. ESPLANADE has one template for each scene-structuring technique. The template generates a default specification for the shots used in the scene. It is responsible for assigning the scope, the object to watch, and the domain time period for the shot. We define the *center-of-interest* as an object feature that is centered in the viewport. ESPLANADE currently uses a simple rule for assigning the center-of-interest and duration to the default shots. If an action involves multiple objects then a shot is created for each of the objects with each object as the center-of-interest. The duration initially assigned to a shot is the duration of the action divided equally by the number of shots used to view the action.

At the shot level, ESPLANADE performs a "patch-up" pass through the shot list to make additional adjustments to satisfy the communicative goals of the animation. During this pass, some shots are modified and some are removed or replaced. The duration and the starting time of shots are changed so that all the actions that must be shown to satisfy a communicative goal will be shown at the correct time and in an appropriate order.

For example, if the location of a sequence needs to be established, then ESPLANADE will add a master shot to the first scene of the sequence. The other shots in the scene need to have their start time and durations adjusted. Another patch-up involves correcting a sequence of shots that does not correctly show the context of an object. If a small object is shown in close-up before it is properly shown in context, the viewer may find it difficult to locate the object in the domain. In this case, ESPLANADE inserts a medium shot before the closeup.

Finally, ESPLANADE determines the transitions between each pair of adjacent shots using several rules described later. At this point in the planning process ESPLANADE has a complete high-level description for each

[1]The scope describes a shot as either a long shot, medium, or close-up.

shot used in the animation. The description includes the center-of-interest, scope, transitions, viewports, ordering and duration. The next step, ESPLANADE's incremental phase, provides control over the generation of the frame-by-frame output primitives for the animation viewer.

In the following sections, we describe the low-level representations used by ESPLANADE and discuss some of the processing phases in more detail.

Image Pipeline and Event Supervisors

ESPLANADE implements a structural representation of a parameterized *image pipeline* (see Figure 3). The pipeline has parameters to control a virtual camera, a fader, and viewport dimensions. Each shot is associated with a pipeline. ESPLANADE assigns priorities to viewports so they will be displayed in the proper order. For example, if a full-screen viewport and an inset viewport are to be displayed simultaneously then the inset viewport is assigned a higher priority to display it on top of the full-screen viewport. Overlapping viewports are also used to create the wipe transition effect. A typical wipe has one viewport expanding from the center of a full-screen viewport until it completely covers the underlying viewport. The fader control in the image pipeline is used to set the opacity of the image in a viewport. The fader control is used to perform fades from black or dissolves when two or more images must blend together on screen.

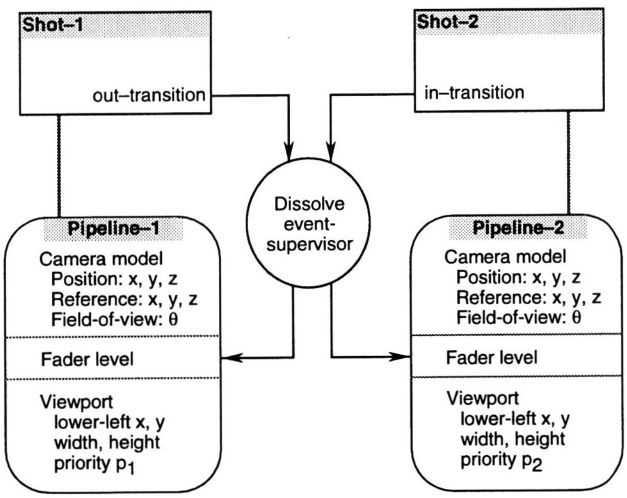

Figure 3: Dissolve event supervisor and image pipeline

A shot and its associated image pipeline are connected by an *event supervisor* that has the responsibility for controlling pipeline parameters dynamically. There are event supervisors for initializing and controlling all dynamic cinematic elements; for example, camera movements, zooms, scope settings (long, medium, and closeup), and transitions (cut, fade, wipe, and dissolve). Every dynamic cinematic element is active between its start time and end time and is interpolated between its start state and its goal state under the control of an event supervisor. For example, if two shots are joined by a dissolve then a dissolve event supervisor is instantiated as shown in Figure 3. The dissolve supervisor is responsible for synchronizing the changes to the fader level of the outgoing shot with the changes to the fader level of incoming shot.

Incremental Phase

During the *incremental phase*, ESPLANADE performs all the remaining tasks needed to generate an animation except for the rendering. We call this the incremental phase because changes to the state of the system, including changes to the domain, are caused by events triggered by an incrementing clock. There are several reasons for traversing the data representation before actually rendering the animation. Most events are fully instantiated before the incremental phase begins; however, ESPLANADE performs viewing specification planning[2], for example, during the incremental phase. Since decisions about a viewing specification must be based on the position of objects in the environment, viewing specification planning is performed during the incremental phase because an object's position information can be determined for each frame of the current shot.

ESPLANADE begins the incremental phase after instantiating all the event supervisors. During this phase a clock, representing *screen time* relative to the start of the film, is incremented. This causes a set of events to be activated. An active event modifies one parameter in the image pipeline. When an event is first triggered, the parameter is initialized to the start state specified by the event supervisor. The parameter is updated every tick during the period the event is active. There is a special event, called a *domain clock event*, that modifies domain time when the event is triggered. This feature allows ESPLANADE to present actions in the domain in an arbitrary order and at any speed. By altering the ratio of the domain clock rate to the screen clock rate the speed of domain actions can be varied. When a domain clock event is triggered at the start of a shot, the domain time is initialized to the beginning of the domain time interval that the shot will present. The entire domain state must be modified to conform to the new domain time.

After all events have been triggered for the current frame and the state of the system has been updated, ESPLANADE retrieves the state of the image pipelines and the domain state and writes the information into a file for the animation viewer. The clock is incremented again, triggering a set of events for the next frame of animation. The process is repeated until all the frames have been processed and output.

Transitions

Transitions are used to join together two shots. ESPLANADE selects a transition from a set of basic transitions: *cut, fade, wipe,* and *dissolve.* The choice of one transition over another is based on its semantics in

[2]ESPLANADE currently uses default viewing specification information that is associated with each object in the domain.

common film usage [25]. For example, the dissolve is commonly used to join shots from two scenes separated by a large time period, while the cut is often used to join shots together that are a part of the same scene. A cut can also be used by default when no other transition is appropriate. We express our heuristics in a rule-based production language. A rule is executed only when all the condition statements on the left-hand side, representing the current state of the planner, are true. We present here in English-language paraphrase, several rules that ESPLANADE currently uses for selecting a dissolve, cut, fadein, or fadeout:

```
(rule select-dissolve
   if the goal is to find a transition
      and shot-2 follows shot-1
      and shot-2 is
          the first shot in scene-2
      and shot-1 is
          the last shot in scene-1
      and difference between
          the time period
          of the events in shot-1
          and the time period
          of the event in shot-2 is
          large
   then the starting transition
        for shot-2 is a dissolve
        and the ending transition
        for shot-1 is a dissolve)

(rule select-cut
   if the goal is to find a transition
      and shot-2 follows shot-1
      and shot-1 and shot-2 are
          part of the same scene
   then the starting transition
        for shot-2 is a cut
        and the ending transition
        for shot-1 is a cut)

(rule select-fadein
   if the goal is to find a transition
      and there are no shots
      before shot-1
   then the starting transition
      is a fadein)

(rule select-fadeout
   if the goal is to find a transition
      and there are no shots
          after shot-1
   then the ending transition
        is a fadeout)
```

Viewing Specifications

Filmmaking heuristics are also used to plan other high-level descriptions of a shot, including the scope (*long*, *medium* or *close-up*), the camera movement (*pan*, *dolly*, *tracking*, or *static*)[3] and the center-of-interest. The description of a shot is used to constrain the search for an exact camera position and path. For example, if the shot is described as a long shot of the driver's side of a car and it must be centered in the viewport then we have identified the half space to locate the camera. Additional constraints may specify that the camera must be located above the action or that some other object must appear in the shot.

EXAMPLE ANIMATION

The domain we are using is a simplified model of a coiled steel storage warehouse with an overhead crane, a button panel, and some rolls of steel. The crane is operated by pushing the buttons (up, down, left, right, front, back, attach, and detach). The attach and detach button need only be pressed momentarily to perform that operation. The other buttons must be pressed and held to move the crane. The action goals are to move roll-7 and roll-2 with the crane. The communicative goals are to show moving roll-7 and roll-2 and to show the causal relation between pressing the up button when roll-7 is lifted and raising the crane's hoist. The script includes the causal information that the up button is the agent of change, the pressing of the button is the action, the hoist is the object, and the hoist rising is the reaction.

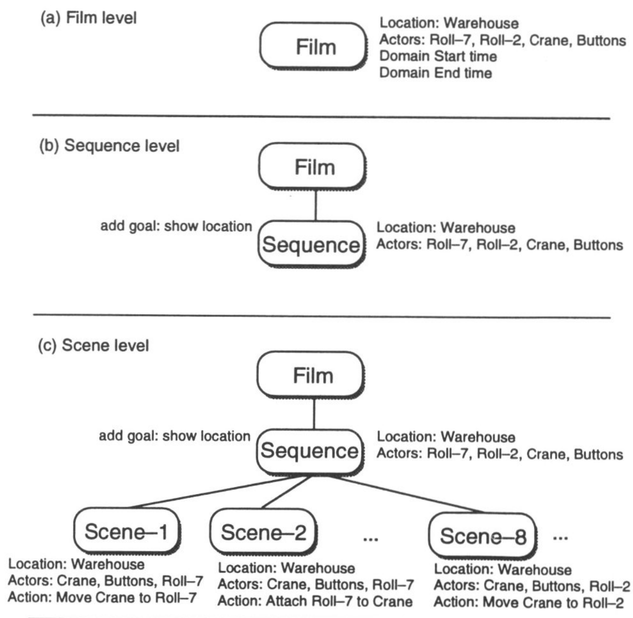

Figure 4: Top down plan

[3] A *pan* is a camera rotation about the y-axis; a *dolly* shot uses a camera that moves either towards or away from a static center of interest; a *tracking* shot uses a camera that follows along with a moving object.

Film Level

Film level goals include choosing the parts of the script to show and determining the objects that participate in these actions. Since ESPLANADE has been requested to produce an animation that shows the two move roll actions, the animation will contain all the objects that participate in these actions (roll-7, roll-2, crane, buttons). The script and cast of participating objects are put into a new instance of the root film node as shown in Fig. 4(a).

Sequence Level

ESPLANADE analyzes the script used at the film level to identify groups of actions that occur at the same location. Since the actions occur at a single locale, a single sequence is created. A new goal is added that requires the location be shown. The script and participating objects are the same as those at the film level, as shown in Fig. 4(b).

Scene Level

ESPLANADE's creates a scene for each action that must be shown. The resulting scene decomposition is shown in Fig. 4(c). ESPLANADE then selects a scene structure for each scene. The master shot discipline is selected as the preferred scene-structuring technique for showing actions. The third scene is structured using the causal scene structuring technique to show the causal relation between pressing the up button and the lifting of the crane.

Shot Level

Since Scene-1, for example, is using the master shot discipline, the default set of shots has separate shots showing each important object in the action as shown in Fig. 5(a). By default the shots are arranged in a series with each shot's domain time interval ending where the next shot's interval begins.

In the patch-up phase, ESPLANADE analyzes the default set of shots and makes several changes to satisfy the communicative goals and filmmaking continuity requirements. These changes are depicted in Fig. 5(b).

ESPLANADE determines that the show location goal needs to be satisfied so a shot of the warehouse is added as the first shot to the first scene. A film rule requires that a viewer must be aware of the context of an object that is shown in a close-up. Therefore, a medium shot of the button panel must be added before the first close-up of the buttons so the viewer knows the location of the buttons in the warehouse when the close-up is shown.

The animation's transitions are determined from the heuristics described earlier. ESPLANADE selects a fadein as the transition for the first shot and a fadeout for the last shot. Cuts are chosen to join two adjacent shots in the same scene. A cut is the default choice to join the last shot in the first scene with the first shot in the second scene, as shown in Fig. 5(c).

Incremental Phase

Events are triggered during each tick. Events inform ESPLANADE that the cameras must be positioned and directed, the fader level incremented, viewports displayed, the domain time must be set, and the domain state updated. Once the entire system state is updated for the current tick, it is output to a file. With this process a complete animation is generated frame-by-frame and ready for presentation with the animation viewer.

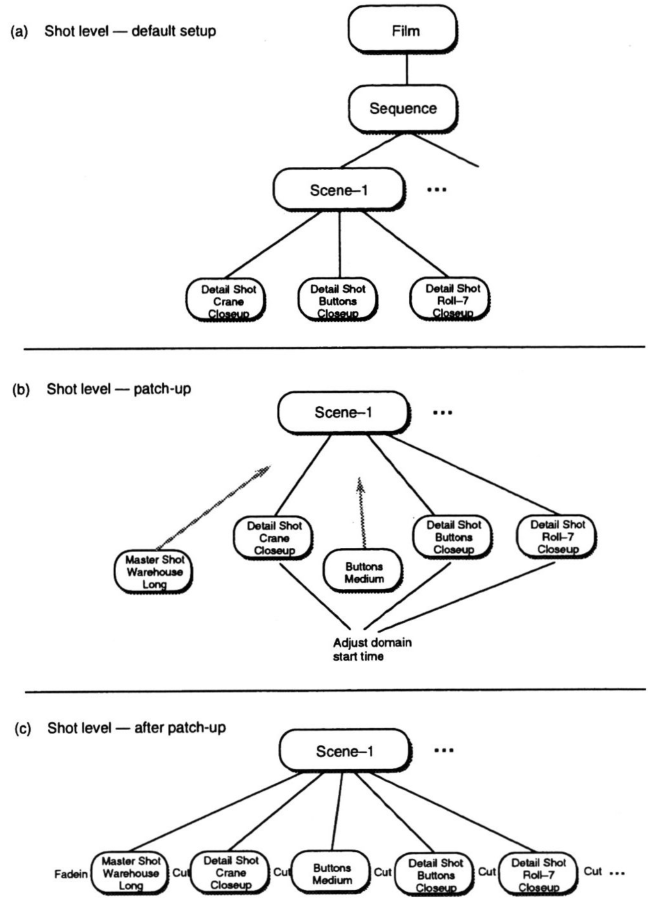

Figure 5: Shot level decisions

Sample Frames

Figure 6 shows sample frames from the beginning of the generated animation. (The animation frames should be read across and down.) The animation begins with a fadein (A1) to a long shot of the warehouse (A2). This is followed by a close-up shot of the crane moving (A3). A medium shot of the buttons (B1) was added by ESPLANADE as a transitional shot to the close-up of the buttons (B2). The medium shot shows that the buttons are located near the rolls. The close-up shot of the buttons shows the text next to the buttons and which buttons are pressed (light).

A close-up shot of roll-7 is used when the crane is above it (B3). The next three shots (C1–3) show the crane grabbing roll-7 and the attach button pressed. (ESPLANADE assigned one shot to each object involved in the action: the crane, the buttons, and roll-7). The six-frame sequence

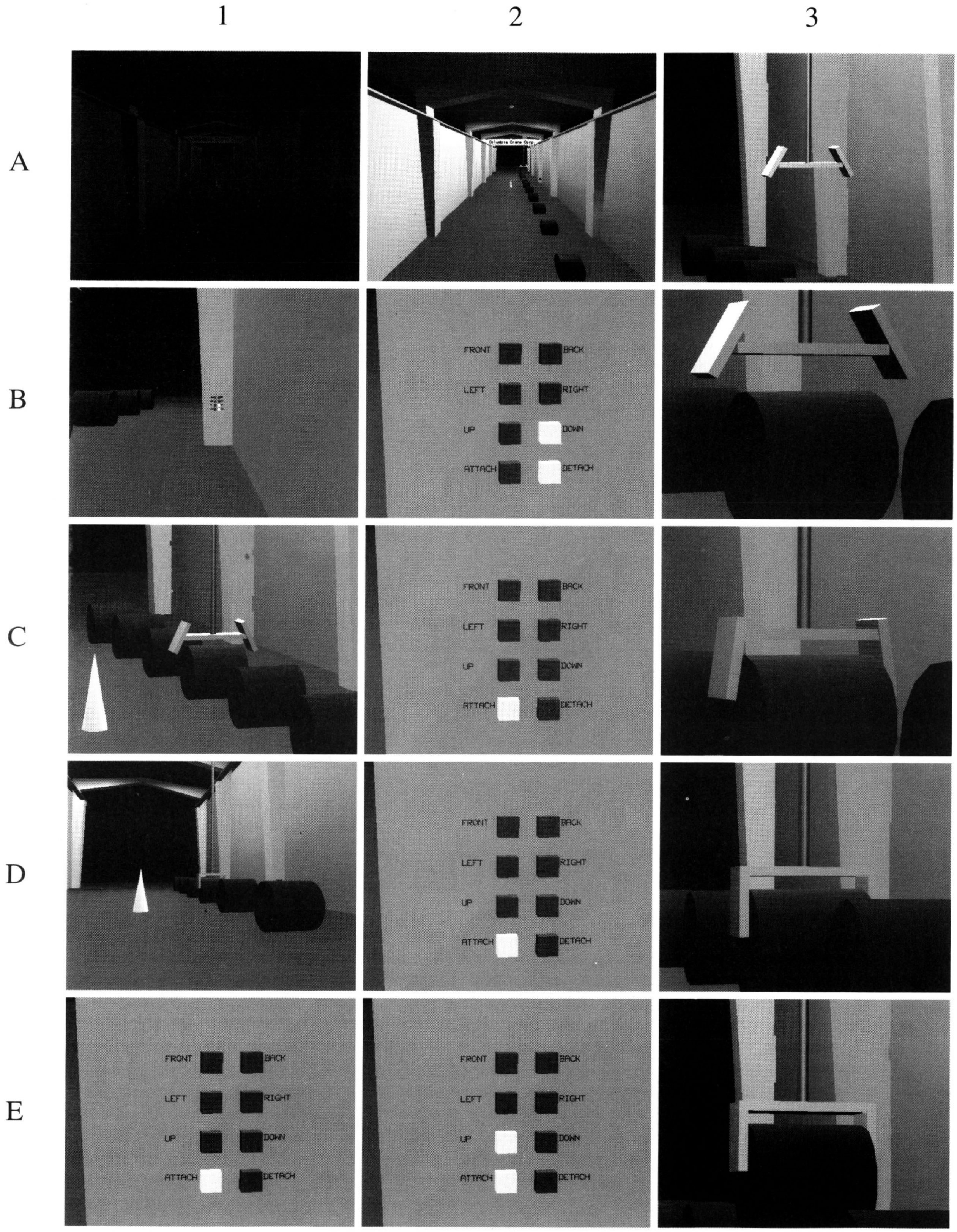

Figure 6: Selected shots from the example animation generated by ESPLANADE

(D1–3) and (E1–3) was generated to satisfy the causality communicative goal. First, the crane and the buttons are shown before the action occurs (D1,D2). Then the crane is seen as it begins to lift the roll (D3). Since the up button must already be pressed to lift the roll, ESPLANADE uses a temporal overlap to show what has caused the roll to rise. The time period shown in (D3) is repeated so the up button can be shown being pressed (E1,E2) to reveal what caused the roll to rise. The last shot (E3) is a close-up of the crane continuing up.

IMPLEMENTATION

ESPLANADE is implemented under UNIX using the CLIPS version 5.0 production system language [6] and C++. The animation viewer and rendering component run in real-time on an HP 9000/730 TurboVRX, using 3D polygon-based object geometry, and interprets a script created during (non-real-time) execution of ESPLANADE. The animation shown in part in Figure 6, is 564 seconds long (4512 frames at 8 frames/sec). It took 700 seconds to plan and consists of 62 separate shots.

CONCLUSIONS AND FUTURE WORK

We have described an experimental hierarchical presentation planner, ESPLANADE, that generates simple animations automatically that fulfill a set of input communicative goals. Task decomposition and filmmaking heuristics are used to reduce the search space for planning an animation. The inherent hierarchical structure of films provides a model for decomposing the planning problem into several abstract levels.

Additional rules are being added to improve the animation. For example, ESPLANADE currently creates a separate shot for each object participating in an action. If the objects are close together, they will appear in both shots. The two shots can either be replaced by a single shot of the two objects or the second shot can be modified so that it will contribute new information about the action. We are also implementing additional communicative goals, based on the set used in IBIS [27]. Additional scene-structures will be needed to express these goals.

Currently, each ESPLANADE animation is at least as long as the total elapsed time taken by the domain actions being presented. In fact, an animation may be even longer if temporally overlapping actions are presented in sequence, as in the example described in this paper. We are particularly interested in supporting presentation goals that determine properties of the presentation itself, such as the length of the animation [15, 23]. This would make it possible for ESPLANADE to be told to create an animation of a specific length or one in which certain actions are constrained to occur at specified screen times. This is a necessary capability if ESPLANADE's animations are to be designed in conjunction with material generated in other media such as speech [11]. We think that the use of speech is particularly important because it can help remove potential ambiguities from the animations that ESPLANADE currently generates. For example, the use of temporal overlap and sequential presentation to show causality in frames D1–E3 of Figure 6 could be explicitly reinforced in spoken narration.

Another long-term goal involves exploring interaction between the action planner and ESPLANADE. The current action planner generates a viewpoint-independent plan independent of the communicative goals. We would like to investigate a planning model that supports interaction between the action planner and the presentation planner so that actions can be planned in a way that will enhance the presentation's satisfaction of the communicative goals.

ACKNOWLEDGMENTS

Our current work is supported in part by the Office of Naval Research under Contract N00014-91-J-1872, by the New York State Center for Advanced Technology in CIS under Contract NYSSTF-CAT-92-053, and by an equipment grant from the Hewlett-Packard Company. Jueychong Ong built the 3D model of the crane world, implemented the animation viewer and parts of the graphics back-end, and produced the figures for this paper. Dewey Thompson, Dave Blair, and Bill Memory offered advice on film production and editing techniques and assisted in prototyping ESPLANADE's rules. Alex Kent at Kawasaki Steel provided information on steel coil material handling.

References

[1] Norman I. Badler, Jonathan D. Korein, James U. Korein, Gerald M. Radack, and Lynne Shapiro Brotman. Positioning and animating human figures in a task-oriented environment. *The Visual Computer*, 1(4):212–220, Dec 1985.

[2] Frank E. Beaver. *Dictionary of Film Terms*. McGraw-Hill, NY, 1983.

[3] Armin Bruderlin and Thomas W. Calvert. Goal-directed, dynamic animation of human walking. *Computer Graphics (Proceedings SIGGRAPH '89)*, 23(3):233–241, July 1989.

[4] John M. Carroll. *Toward a Structural Psychology of Cinema*. Mouton Publishers, The Hague, 1980.

[5] David Cheshire. *The Book of Movie Photography*. Alfred A Knopf, New York, 1974.

[6] C. Culbert. CLIPS reference manual–Version 5.0. Technical report, NASA/Johnson Space Center, TX, January 1991.

[7] Marc Davis and Warren Sack. IDIC: assembling video sequences from story plans and content annotations. MIT Media Lab, Learning and Common Sense Section, Working Paper: Cambridge, MA., 1993.

[8] Karin Drewery and John Tsotsos. Goal directed animation using English motion commands. In *Proceedings Graphics Interface '86*, pages 131–135, 1986.

[9] Steven M. Drucker, Tinsley A. Galyean, and David Zeltzer. Cinema: A system for procedural camera movements. In *Proc. 1992 Symposium on Interactive 3D Graphics*, Boston, MA, March 30–April 1 1992. ACM.

[10] Jefrey Esakov, Norman I. Badler, and Moon Jung. An investigation of language input and performance timing for task animation. In *Proceedings Graphics Interface '89*, pages 86–93, 1989.

[11] S. Feiner, D. Litman, K. McKeown, and R. Passonneau. Towards coordinated temporal multimedia presentations. In M. Maybury, editor, *Intelligent Multimedia Interfaces*. AAAI Press, Menlo Park, CA, 1993.

[12] Steven K. Feiner. APEX: An experiment in the automated creation of pictorial explanations. *IEEE Computer Graphics and applications*, 5(11):29–37, 1985.

[13] R. E. Fikes and N.J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[14] Michael Gleicher and Andrew Witkin. Through-the-lens camera control. *Computer Graphics (Proceedings Siggraph '92)*, 26(2):331–340, July 1992.

[15] Kenneth M. Kahn. *Creation of Computer Animation from Story Descriptions*. PhD thesis, AI Lab, Massachusetts Institute of Technology, Cambridge, MA, January 1979.

[16] Peter Karp and Steven K. Feiner. Issues in the automated generation of animated presentations. In *Proceedings Graphics Interface '90*, pages 39–48, 1990.

[17] Tsukasa Noma and Naoyuki Okada. Automating virtual camera control for computer animation. In Nadia Magnenat-Thalmann and Daniel Thalmann, editors, *Creating and Animating the Virtual World*, pages 177–187, Tokyo, 1992. Springer-Verlag.

[18] Cary B. Phillips, Norman I. Balder, and John Granieri. Automatic viewing control for 3D direct manipulation. In *Proc. 1992 Symposium on Interactive 3D Graphics*, pages 71–74, Boston, MA, March 30–April 1 1992. ACM.

[19] Karel Reisz and Gavin Millar. *The Technique of Film Editing*. Focal Press, London, 1968.

[20] Craig W. Reynolds. Flocks, herds and schools: A distributed behaviorial model. *Computer Graphics (Proceedings SIGGRAPH '87)*, 21(4):25–34, July 1987.

[21] Garfield John Ridsdale. *The Director's Apprentice: Animating Figures in a Constrained Environment.* PhD thesis, Simon Fraser University, Burnaby, B.C., June 1987.

[22] Benjamin Rubin. Constraint-based cinematic editing. Master's thesis, Massachusetts Institute of Technology, 1989.

[23] Benjamin Rubin and Glorianna Davenport. Structured content modeling for cinematic information. *SIGCHI Bulletin*, 21(2):78–79, October 1989.

[24] Earl D. Sacerdoti. Planning in a hierarchy of abstraction. *Artificial Intelligence*, 5:115–135, 1974.

[25] R.C. Schank and R.P Abelson. *Scripts, plans, goals, and understanding*. Lawrence Erlbaum, Hillsdale, N.J., 1977.

[26] Dorée Seligmann and Steven K. Feiner. Specifying composite illustrations with communicative goals. In *Proc. UIST 89 (ACM SIGGRAPH Symp. on User Interface Software and Technology)*, pages 1–9, Williamsburg, VA, November 13-15 1989.

[27] Dorée Duncan Seligmann and Steven K. Feiner. Automated generation of intent-based 3D illustrations. *Computer Graphics (Proceedings SIGGRAPH '91)*, 25(4):123–132, July 1991.

[28] Stefan Sharff. *The Elements of Cinema–Toward a Theory of Cinesthetic Impact*. Columbia University Press, N.Y., 1982.

[29] Yosuke Takashima, Hideo Shimazu, and Masahiro Tomono. Story driven animation. In *Proceedings CHI + GI 1987*, pages 149–153, 1987.

[30] Dewey Thompson. Personal communication, 1992.

[31] Jane Wilhelms. Toward automatic motion control. *IEEE Computer Graphics & Applications*, 7(4):11–22, April 1987.

[32] David Zeltzer. Motor control techniques for figure animation. *IEEE Computer Graphics & Applications*, 2(9):53–59, November 1982.