Realtime Constraint-Based Cinematography for Complex Interactive 3D Worlds*

William H. Bares and **Joël P. Grégoire** and **James C. Lester**
Multimedia Laboratory
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-7534
{whbares, jgregoi, lester}@eos.ncsu.edu

Abstract

In 3D interactive fiction systems, a virtual camera must "film" the behaviors of multiple autonomous characters as they unpredictably interact with one another, are modified by the viewer, and manipulate artifacts in 3D worlds with complex scene geometries. It must continuously plan camera movements to clearly shoot the salient visual features of each relevant character. To address these issues, we have developed a 3D interactive fiction system with a narrative planner that, together with a bank of autonomous character directors, creates cinematic goals for a constraint-based realtime 3D virtual cinematography planner. As interactive narratives unfold, a cinematic goal selector creates view constraints to film the most salient activities performed by the characters. These constraints are then passed to a camera planner, which employs a partial constraint-based approach to compute the position and orientation of the virtual camera. This framework has been implemented in a prototype 3D interactive fiction system, COPS&ROBBERS, a testbed with multiple characters interacting in an intricate cityscape.

Introduction

One of the most promising unexplored fields of AI applications lies in the entertainment industry, and perhaps no family of applications offers more potential than interactive fiction systems. While natural language story generation has been a goal of AI for more than two decades (Meehan 1976) and text-based interactive fiction systems have been the subject of increasing attention (Murray 1997), it is the prospect of coupling sophisticated inference with believable characters (Bates 1994) that offers the potential of realizing the long-held goal of fiction generation in a visually compelling environment.

*Support for this work was provided by a grant from the NSF (Faculty Early Career Development Award IRI-9701503), the IntelliMedia Initiative of North Carolina State University and an industrial gift from Novell, Inc.

A key problem in bringing 3D interactive fiction systems to fruition is accommodating the viewing demands of an interactive, and therefore unpredictable, narrative. Intelligent realtime camera planning is critical for the dynamic worlds of 3D interactive fiction in which multiple autonomous characters inhabit complex environments. As narratives dynamically unfold, characters unpredictably interact with one another and with artifacts in the environment as they meander through sprawling landscapes and intricate cityscapes. From moment to moment, the virtual camera must "film" all of these interactions in realtime as they play out in the surrounding physical context, regardless of how visually complicated it may be. This entails continuously planning camera movements so that it clearly shoots the salient visual features of each relevant character, artifact, and structure. To do so, it must plan vantage angles and distances that allow the viewer to immediately comprehend the actions as they occur in the scene while continuously avoiding intervening occluding obstacles.

Previous work on automated camera planning does not provide a general-purpose solution that addresses these requirements. One family of systems employs camera positions that are pre-specified relative to the subject(s) being viewed (Bares & Lester 1997; Butz 1997; Feiner 1985; Karp & Feiner 1990). The IBIS system (Seligmann & Feiner 1991) takes a similar approach but also supplements it with limited ability to overcome viewing failures by using cutaways of occluding objects and creating multi-view illustrations. A second family of systems encodes idiom-based knowledge of cinematography as sequences of shots to depict commonly occurring actions (Karp & Feiner 1993; Christianson *et al.* 1996; He, Cohen, & Salesin 1996). Each shot of an idiom is encoded as a camera position pre-specified relative to the subject(s). Consequently, they work well for filming subjects performing stereotypical actions in a predictable fashion in simple surroundings, e.g., filming a conversation between two actors. However, both of these approaches fail when the camera must simultaneously film arbitrary combinations of subject objects with arbitrary constraints on vantage and/or distance, or when unexpected structures in the surroundings occlude the subjects of in-

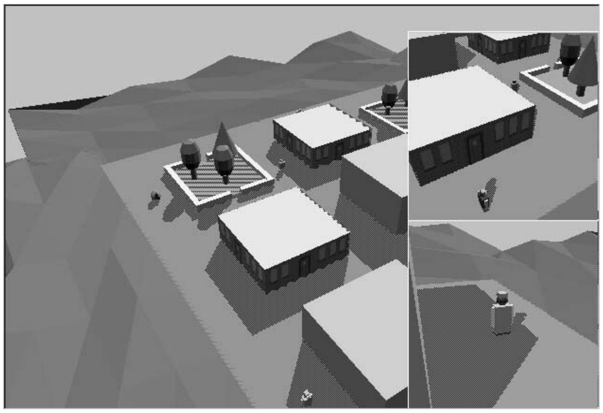

Figure 1: Example scene from COPS&ROBBERS

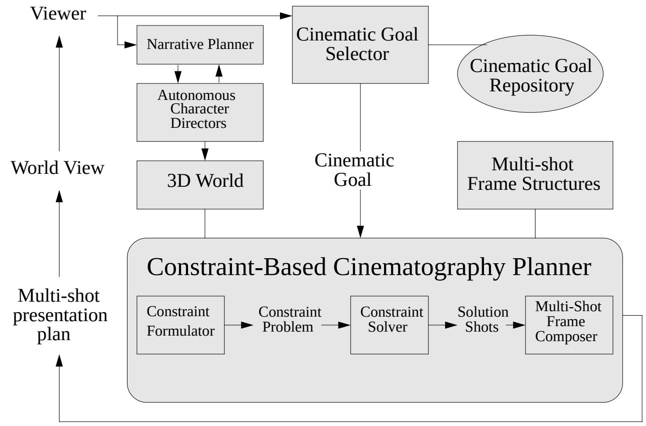

Figure 2: The 3D Interactive Fiction Architecture

terest. Lacking methodologies for reasoning about the viewing problem in the context of the surrounding environment, they can only respond by selecting one of the pre-specified relative camera positions, e.g., placing the camera at a point displaced by a given vector from the subject(s).

In contrast, the constraint satisfaction approach to automated cinematography casts camera planning as a constraint satisfaction problem (Drucker & Zeltzer 1995). Given a request to view particular subjects of interest and a specification of how each should be viewed, the CAMDROID constraint solver attempts to find a solution. In general, the constraint-based approach offers significantly greater flexibility than alternate approaches. However, this initial constraint-based effort does not offer a systematic solution for handling constraint failures that can occur frequently in dynamic environments with complex scene geometries.

To address these issues, we have developed a general-purpose, constraint-based framework for intelligent realtime 3D cinematography. By reasoning from a kind of "cinematic first principles" of scene geometries, camera planners can clearly present the behaviors of multiple autonomous characters as they unpredictably interact with one another and manipulate artifacts in complex 3D worlds. By employing partial constraint satisfaction, they can provide alternate solutions when constraints cannot be completely satisfied. When constraints fail, they relax weak constraints and, if necessary, decompose a single shot to create a set of camera placements, which they then present with either a temporal sequence of shots or composite shots with simultaneous multiple viewports.

This framework has been implemented in an CONSTRAINTCAM, a realtime camera planner for complex dynamic 3D virtual worlds. To investigate its performance, CONSTRAINTCAM's behavior has been studied in COPS&ROBBERS (Figure 1) a 3D interactive fiction testbed in which a policeman and two robbers compete with one another to capture a money bag lost in a cityscape populated by a multitude of potentially occluding buildings. CONSTRAINTCAM films the characters in realtime as they exhibit goal-directed stochastic behaviors such as searching for the money bag, pursuing each other, grabbing the money bag from each another, and attempting to return it to the bank or their hideouts. The camera planner can monitor the autonomous action, and automatically shift the camera's focus to interesting events as the plot progresses. To further stress test the camera, the viewer may at any time modify the course of the narrative by changing any of characters' attributes of eyesight range and speed. CONSTRAINTCAM's realtime performance and the results of a focus group study with the COPS&ROBBERS testbed have yielded encouraging results.

3D Interactive Fiction Cinematography

In the 3D interactive fiction architecture, the narrative planner invokes autonomous character directors to initiate the behaviors of each of the characters. It directs them to accomplish their tasks by navigating the world defined by the scene layouts and 3D models in the environment. At any time, viewers can modify the characters' behaviors, which will affect the activities of the narrative planner and update its directives to the cinematic goal selector. During the narrative, the cinematography planner will receive cinematic goals which specify which characters or objects the camera should film. As the action in the world unfolds, the cinematic goal selector triggers cinematic goals to reflect new developments in the action. For example, when one character begins pursuing another character, a new cinematic goal to view the pursuit is automatically selected and passed to the constraint-based cinematography planner. In addition, viewers can post a new cinematic goal to see the behaviors or relative locations of particular sets of characters.

When the cinematography planner receives a goal, it formulates a cinematic constraint problem consisting of a set of subjects and, for each subject, a set of subject inclusion, vantage angle, shot distance, and occlusion avoidance constraints. The constraint solver analyzes the constraints and attempts to find a single camera placement that will satisfy all of the constraints with respect to the scene geometry of the surrounding environment. If a solution in the form of a single shot cannot be found, then the constraint solver builds an incompatible constraints pair graph to guide the relaxation of weak constraints or the decomposition of the constraint problem to form a multi-shot solution. The multi-shot frame composer exploits a repository of multi-shot frame structures to create sequential or composite presentations of the multiple shot solution. The entire process repeats continuously in realtime to reflect new developments in the world.

3D Narrative Planning

The *3D narrative planner* begins by initializing the goals of each of the characters in the interactive story. For example, in the COPS&ROBBERS testbed, three characters, a cop and two robbers, are all given the initial goal of seeking a lost money bag. The bank of *autonomous character directors* then determines the behaviors appropriate for accomplishing each character's current goal. Characters navigate through the 3D world and interact, both with one another and with world artifacts, e.g., the money bag. Characters navigate by examining a grid, each cell of which represents either an obstacle or a navigable passageway. They sight each other by scanning grid cells for their objective until reaching either their eyesight range or an obstacle cell. Characters walk about at random until they have sighted their objective, at which point a shortest-path algorithm is used to direct them efficiently towards their objective. When a character achieves a goal, the narrative planner dictates the successor goal. For example, in the COPS&ROBBERS testbed, when the cop spots a robber stealing the money bag, he is instructed by the narrative planner to give pursuit. At any point, the viewer can modify the ongoing narrative by changing characters' attributes, e.g., eyesight range, which affects how he/she interacts with all of the other characters.

In addition to incrementally posting goals for the characters, the narrative planner tracks *epochs* of the story, each of which is defined by significant turning points in the narrative's events. Types of epochs include:

- *Visual Identification:* For example, a character sights the money bag and heads towards it.
- *Transportation:* For example, a character carries the money bag to his destination.
- *Pursuit:* For example, the cop sights a robber with the money bag and gives chase.

The *cinematic goal selector* monitors the progress of the current epoch and posts cinematic viewing goals that call for the camera to depict the most salient activities in the current epoch. To illustrate, in the COPS&ROBBERS interactive fiction testbed, suppose a policeman has just sighted a money bag and begins to head towards it. The camera planner should compute a view that indicates the policeman's progress towards the money bag. Informally, this *cinematic goal* specifies that we want a camera view that clearly shows him, the terrain ahead, and the money bag free of occlusions so that the viewer can determine their relative locations.

Constraint-Based Camera Planning

Once a cinematic goal indicating the characters of interest has been passed to the cinematography planner, the constraint formulator creates a constraint problem by specifying the set of characters of interest and the preferred way in which each character is to be viewed. The cinematography planner accommodates four types of constraints, each of which can be applied to any character, includes a relative strength, and a marker indicating whether that constraint can be relaxed. *Subject inclusion* constraints specify which characters to include in the camera's field of view. *Vantage angle* constraints indicate the permissible and optimal relative orientations between the camera and the character. *Shot distance* constraints specify the minimum, maximum, and optimal distances between the camera and a character. Finally, *occlusion avoidance* constraints indicate whether the camera position should be displaced from the optimal vantage angle when necessary to prevent the character from being occluded by obstacles.

The cinematic constraint solver must identify a region of space that is satisfactory to all constraints within which it can position the camera. The cinematic constraint algorithm (Figure 3) begins by determining the consistent region of 3D space in which the camera can be placed to satisfy each of the constraints (Step 1). The critical step converts the consistent regions, which are expressed as spatial relations relative to individual subjects, into a corresponding representation in terms of a common "global" composite spherical coordinate system with origin at the midpoint of all subjects.

If the intersection of the consistent regions I (Step 2) is non-empty, then the cinematographer searches for the spherical coordinates point within I that is nearest the optimal vantage for viewing the subject(s) (Step 3). The camera distance from the aim point (central to the subject objects) is computed via intersecting distance intervals corresponding to the consistent regions for the viewing distance constraints. The solution point is then converted from spherical coordinates to Cartesian coordinates to determine the camera position. However, if the intersection I is empty, then constraint relaxation and possibly shot decomposition methods (discussed below) are employed to compute an alternate solution.

If no solution can be found for the given constraint problem, then the cinematography planner attempts to find an alternative (Steps 4, 5, and 6). The cinematography planner first identifies the combinations of constraints that are incompatible, then tries to find a maximal solution which satisfies as many of the higher priority constraints as possible. Combinations of in-

1. Compute consistent region of space $R_{s,i}$ that satisfies constraint $C_{s,i}$, for all $1 \leq s \leq NumSubjects$, and $1 \leq i \leq NumConstraints_s$.
2. Compute the intersection I of all consistent regions.

$$I = \bigcap_{\substack{1 \leq s \leq NumSubjects \\ 1 \leq i \leq NumConstraints_s}} R_{s,i}$$

3. If the intersection I is non-empty, then compute and return shot solution.
4. If relaxation or decomposition is desired, then identify incompatible constraint pairs of the form $C_{s1,i1}$, $C_{s2,i2}$ such that $R_{s1,i1} \not\cap R_{s2,i2}$ where $1 \leq s1, s2 \leq NumSubjects$, $1 \leq i1 \leq NumConstraints_{s1}$, $1 \leq i2 \leq NumConstraints_{s2}$ and $s1 \neq s2$.
5. Perform relaxation by considering each constraint of an incompatible constraint pair $C_{s1,i1}$, $C_{s2,i2}$ in order of weakest to strongest. If a constraint $C_{s1,i1}$ can be relaxed, then mark it relaxed and delete all incompatible constraint pairs $C_{s1,i1}$, $C_{x,y}$ that conflicted with relaxed constraint $C_{s1,i1}$. Continue until either all constraints are considered or no incompatible constraint pairs remain. If no incompatible pairs remain, then return the solution for the relaxed constraint problem.
6. Perform decomposition by placing $C_{s1,i1}$ and $C_{s2,i2}$ in separate constraint sub-problems P_1 and P_2 if there exists an incompatible constraint pair arc between $C_{s1,i1}$, $C_{s2,i2}$. Into each resulting sub-problem P_i, insert all constraints $C_{x,y}$ that are compatible with the constraints in sub-problem P_i. Return a solution for each sub-problem.
7. Given the camera solution of $NumShots$, select a graphical shot decomposition strategy to present the resulting shots in either sequence, or multiple viewport layout.

Figure 3: The Cinematic Constraint Algorithm

compatible constraints are identified by constructing an *incompatible constraints pair graph*. The cinematographer first creates a node for each constraint $C_{s,i}$. Next, it adds an arc connecting a pair of nodes $C_{s1,i1}$, $C_{s2,i2}$ if their consistent regions $R_{s1,i1}$ and $R_{s2,i2}$ fail to intersect (Step 4).

The cinematography planner then repeatedly relaxes weak constraints until no incompatible constraint pairs remain (Step 5). Relaxation is accomplished by scanning over the constraints $C_{s,i}$ that are involved in at least one arc of the incompatible constraint pairs graph. Constraints are considered for relaxation in order of lowest priority. If the constraint $C_{s,i}$ being considered is deemed "weak," then it marks that constraint as being relaxed and deletes all incompatible constraint graph arcs that involve this constraint. It continues until either no more constraints remain to be considered or no more arcs remain in the graph. If no incompatible constraint pairs arcs remain, then it submits the resulting relaxed constraint problem to the constraint solver. If relaxation was successful, it returns a single shot camera solution.

If relaxation is not possible, then the cinematography planner considers decomposing the original viewing constraint problem into several sub-problems (Step 6). It attempts to satisfy as many constraints as possible in each sub-problem to avoid needless camera shots. It therefore places each constraint of an incompatible constraint pair $C_{s1,i1}$, $C_{s2,i2}$ in a distinct sub-problem P_i (unless that constraint has already been placed in a new sub-problem) and then inserts all possible compatible constraints into each sub-problem. Thus, for sub-problem P_i including constraint $C_{s1,i1}$, it adds constraint $C_{s2,i2}$ if no arc in the incompatible constraint pairs graph connects them.

If the cinematography planner has decomposed a constraint problem into a multi-shot solution, it must then determine how to present the set of multiple camera shots (Step 7). In some cases, it may be important to display several shots in the solution set simultaneously using composite shots. Composited shots use both a main viewport and an inset viewport so that the viewer can compare the subjects in the shots or gauge some relationship between their attributes such as relative location or size. In other cases, it may be preferable to focus on the details in each shot individually, in which case a sequence of shots is shown.

Implementation

The constraint-based cinematography framework has been implemented in ConstraintCam, a realtime 3D camera planner.[1] ConstraintCam's shot composition behaviors are driven by the partial constraint satisfaction and relaxation methods presented above. For physically complex environments and the up to 16 simultaneously active constraints in the testbed interactive fiction world discussed below, ConstraintCam's full implementation of the the constraint-based cinematography algorithm (Figure 4) executes in approximately 25 milliseconds for worst case four shot solutions on a 233 MHz Pentium II with 64 MB with a Permedia2 3D accelerator running Windows NT 4.0. Depending on the number of viewports to draw per frame, it achieves frame rates of between 7 and 15 frames/second.

The 3D Interactive Fiction Testbed

To investigate ConstraintCam's behavior, it has been studied in Cops&Robbers, a 3D interactive fiction testbed with multiple characters interacting with each other in an intricate cityscape. Three autonomous characters, Murphy the policeman and two robbers, Sam and Jake, try to capture the lost money bag dropped by a careless bank teller. If the policeman finds the money bag, he dutifully returns it to the bank. But if either of the two miscreants find the unclaimed money, they will scurry off to Joe's Place to spend their new

[1] ConstraintCam, which together with the 3D interactive fiction world testbed consist of approximately 55,000 lines of C++, employs the OpenGL 3D graphics library for realtime 3D rendering.

found loot. If the cop catches either robber carrying the money, he will immobilize him and return the money bag to the bank. When the narrative begins, the initial locations of the three characters are randomly assigned, and then they begin to meander randomly through the town searching for the lost money bag.

To challenge CONSTRAINTCAM's ability to handle large sets of cinematography constraints that emerge in an unpredictable fashion, at any time viewers can (1) affect the plot's outcome by specifying the speed and eyesight range of each of the three characters, (2) specify preferences for the optimal vantage angle for viewing each character, and (3) post cinematic goals by indicating which of the three characters, the money bag, the bank, and/or Joe's Place to view. To challenge CONSTRAINTCAM's occlusion performance, the interactive fiction testbed was populated with a number of buildings including a bank, Joe's Place, five other buildings, two city parks, and a surrounding mountain range. In the most challenging case, the viewer can request that the camera track all four principals (the policeman, the two robbers, and the money bag), each with four constraints. The viewer can also indicate preferences for one of three varying amounts of information content per screen, which influences the degree to which multi-shot decompositions employ simultaneous inset viewports.

3D Interactive Fiction Example

This example illustrates how the cinematographer films events in the dynamic interactive fiction testbed in response to cinematic goals. The viewer begins with a preference for the cinematographer to present a low information content per screen (no insets), and specifies a cinematic goal to show the whereabouts of the three characters. Due to their separation the camera cannot find a shot that satisfies all constraints. The low information content per screen preference results in only one shot displayed per screen. The multiple shot solution is presented using a sequential multi-shot frame structure that opens with a relaxed constraint overview shot of all characters to be followed by detail shots of each character (Figure 4 a).

After about one minute of wandering the first robber Sam spots the money bag and begins moving in to claim the prize. The Cinematic Goal Selector automatically triggers a new cinematic goal to show the location of Sam relative to the money bag in response to the escalating world actions. In this case, the camera constraint solver was able to find a single shot that clearly showed both Sam and the money bag (Figure 4 b).

The viewer decides to stir things up by increasing the police officer's speed and eye sight range to improve his odds of spotting Sam with the money and catching him. He then executes his search more aggressively, and soon rounds the corner and spots Sam carrying the money bag. A hot pursuit ensues as the officer tries to catch Sam and reclaim the money for its rightful owner. The Cinematic Goal Selector enacts a cinematic goal to show the locations of the policeman and Sam. In this case the result is a single shot of both characters with a viewing angle setup to favor the optimal viewing angle (behind-the-back) for the policeman, the highest priority subject in this pursuit goal (Figure 4 c).

After immobilizing the robber, the policeman, reclaims the money bag to return it to the bank. A cinematic goal is triggered to depict the his progress towards the bank. With the bank located several blocks away and behind another building, the cinematography planner computes a multiple shot solution. The newly specified medium information content per screen permits a multi-shot frame structure that includes a relaxed constraint overview shot to establish the relative locations of the bank and the officer, and uses the inset viewport to present the shots of the decomposition, the first of which gives a detail shot of the bank (Figure 4 d).

Focus Group Study

To assess the performance of CONSTRAINTCAM with real users, an informal focus group study was conducted with 8 subjects interacting with the COPS&ROBBERS testbed. The study was conducted to (1) investigate how effective CONSTRAINTCAM is at communicating key events of dynamic narrative structures in unpredictable, complex environments and (2) to determine how well it responds to user's possibly idiosyncratic view requests. The findings suggest that CONSTRAINTCAM can consistently track multiple subjects interacting with each other and moving between obstacles (buildings of the city) in realtime. CONSTRAINTCAM's ability to depict complex scenes by relaxing constraints to create bird's-eye overview shots composited with inset shots from constraint decomposition were particularly well received. Some subjects suggested improving shot continuity and transitions, and also avoiding inset shots that present information that is redundant with respect to other viewports.

Conclusions and Future Work

As interactive 3D worlds appear in an expanding range of entertainment systems, they place an increasingly greater demand on intelligent virtual cinematographers that can film their activities. We have proposed a narrative generation *cum* camera planning framework for filming the worlds of interactive fiction. By coupling a narrative planner that includes a bank of autonomous character directors with a cinematic goal selector that formulates viewing constraints, a constraint-based camera planner can compose moment-by-moment shots of the most salient actions in the scene. While this work addresses many of the core issues in cinematography for interactive 3D fiction systems, much remains to be done. A particularly intriguing line of investigation is that of complementing the visual impact of virtual 3D narrative cinematography with natural language generation for running commentary. We will be exploring these issues in future research.

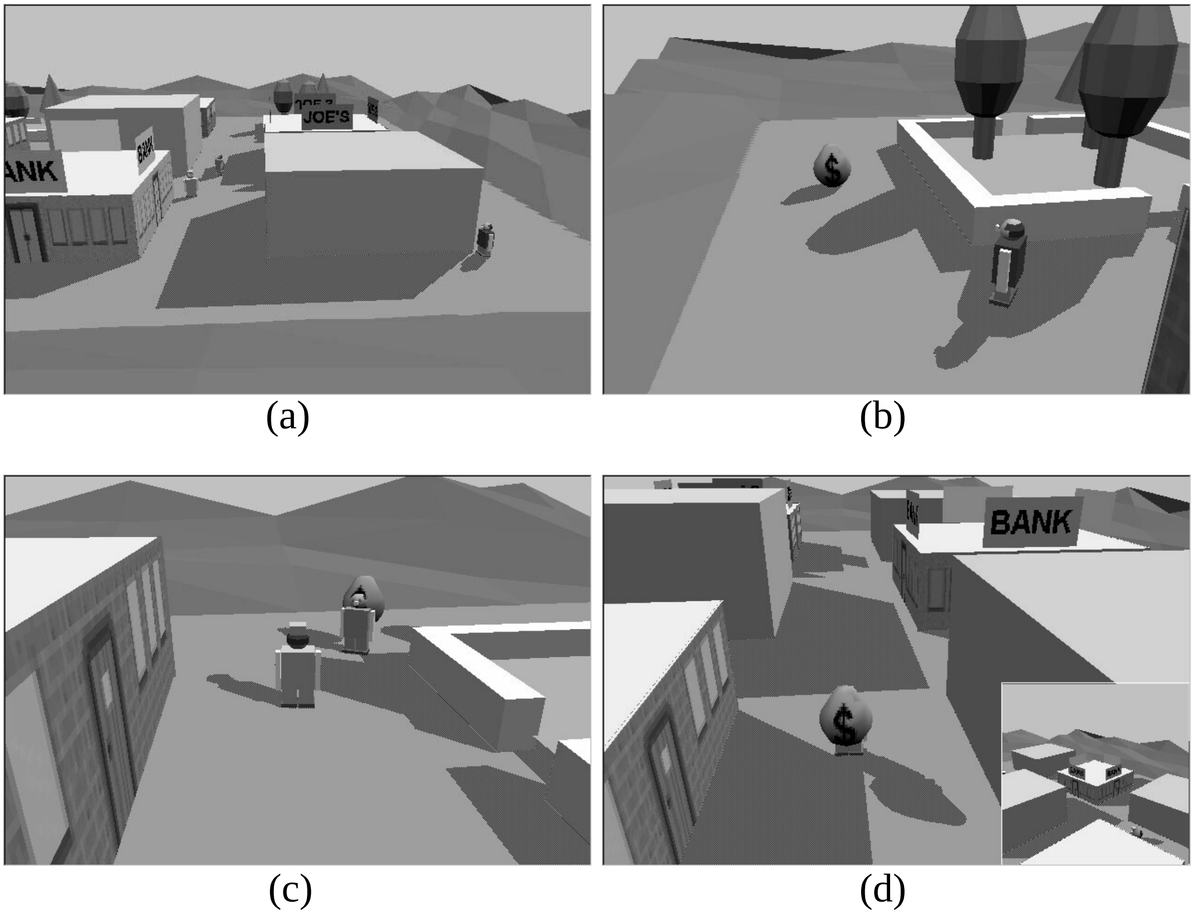

Figure 4: Example COPS&ROBBERS narrative sequence

References

Bares, W. H., and Lester, J. C. 1997. Realtime generation of customized 3D animated explanations for knowledge-based learning environments. In *AAAI-97: Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 347–354.

Bates, J. 1994. The role of emotion in believable agents. *Communications of the ACM* 37(7):122–125.

Butz, A. 1997. Anymation with CATHI. In *Proceedings of the Ninth Innovative Applications of Artificial Intelligence Conference*, 957–62.

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.

Feiner, S. 1985. APEX: An experiment in the automated creation of pictorial explanations. *IEEE Computer Graphics and Applications* 29–37.

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

Karp, P., and Feiner, S. 1990. Issues in the automated generation of animated presentations. In *Proceedings of Graphics Interface '90*, 39–48.

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.

Meehan, J. 1976. *The Metanovel: Writing Stories by Computer*. Ph.D. Dissertation, Yale University, New Haven, Connecticutt.

Murray, J. H. 1997. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. The Free Press.

Seligmann, D. D., and Feiner, S. K. 1991. Automated generation of intent-based 3D illustrations. *Computer Graphics* 25(4):123–132.