Proceedings of the Fifth Artificial Intelligence for Interactive Digital Entertainment Conference

Dynamic Artificial Potential Fields for Autonomous Camera Control

Paolo Burelli and **Arnav Jhala**
Center For Computer Games Research
IT University Of Copenhagen
pabu@itu.dk, arjh@itu.dk

Abstract

Camera control in real-time interactive 3D applications is a challenging problem. Developing a generalized system able to produce high quality visual results and smooth camera movements in dynamic environments remains an open problem in the research community. In this paper, we describe the implementation and evaluation of Artificial Potential Fields for automatic camera placement. We first describe the recasting of the frame composition problem as a solution to a two particles suspended in an Artificial Potential Field. We demonstrate the application of this technique to control both camera location and camera aim direction. We show that this technique can be successfully used to solve both camera animation and frame composition tasks in dynamic environments in real-time through an example implemented on an existing 3D game engine.

Introduction

In interactive 3D applications, such as games, a virtual camera represents the point-of-view of the user. Therefore, camera control is one of the fundamental aspects of user's interaction with the virtual world. The quantity and quality of information perceived by the user in a 3D environment is dependent on the camera control employed by the system. An autonomous camera control system is a tool that permits the application designer to control these aspects without pre-scripting the camera parameters at design time. Such a system enables the designer to define a set of visual requirements and automatically configures the camera view according to these requirements at run-time.

Several techniques for autonomous camera control have been proposed in the past (for a comprehensive overview see (Christie and Olivier 2006)) to address this problem. Most of these techniques either focus on frame composition, or on camera motion. While there are efficient approaches for solving one of these problems at a time, these approaches are not effective for both composition and movement at the same time.

We propose a system that employs Artificial Potential Fields to perform both camera animation and frame composition tasks within the same camera framework. Artificial Potential Fields (APFs) is a well-known technique in the field of robotics used for robot navigation in dynamic environments; it has been applied to camera control (Beckhaus, Ritter, and Strothotte 2000) to address camera path-planning in exploratory tasks within virtual environments.

Our system extends this work by modelling the camera movement and camera orientation parameters, both with multiple APFs. integrating frame composition concepts to both camera positionning and orientation. To do this we first propose a model to recast frame composition requirements into an Artificial Potential Fields representation and then we propose the adoption of multiple potential fields to control both camera position and camera orientation. We show that this technique permits the development of an autonomous camera system capable of controlling the camera in dynamic 3D environments according to frame composition requirements with real-time performances.

In the remainder of the paper, we first describe our APF representation of geometric constraint satisfaction. Then, we show how the proposed system is able to deal with over-constrained frame requirements (no solution for given constraints) and under-constrained frame requirements (multiple camera configurations as solutions to the given constraints). We show how this enables the designer to choose whether camera is fully controlled by the system or can by partially controlled by the user. Finally, a working example implementation of the system is described.

Related work

The problem of automatically control the camera in virtual camera in 3D environments has received significant attention from the research community (Christie and Olivier 2006). Earliest approaches (Ware and Osborne 1990; Blinn 1988; Gleicher and Witkin 1992) focused on the mapping between the degrees of freedom (DOF) for input devices to 3D camera movement.

Direct control of the several camera's DOFs showed to be often problematic for the user (Drucker and Zeltzer 1994) so researchers started to investigate how to automatically place and configure the camera. Christie and Olivier (Christie and Olivier 2006) classify different approaches into three main categories according to the the modelling algorithm: algebraic systems, reactive systems and generalized approaches.

One of the first examples of algebraic systems, was developed by Blinn already in 1988; it was am automatic camera control system for planets visualization in a space simulation at NASA (Blinn 1988). Blinn's work has been followed by many others trying to produce more flexible autonomous camera system and to integrate aspects like camera motion and frame composition (Arijon 1991). Gleicher and Witkin (Gleicher and Witkin 1992) introduced through-the-lens camera control technique, a reactive technique inspired by visual-servoing which permits the user to manipulate the camera by controlling and constrainting projected images parameters.

As through-the-lens based systems are computationally efficient, they are ideal for tasks such as object tracking in interactive applications. Their aim, however, is to maintain specific image features (i.e. keep an object in the center of the screen) and require a preliminary camera initialization.

Generalized approaches have modelled camera control as a constraint satisfaction or optimization problem. These approaches require the designer to define a set of required frame properties which are then modelled either as an objective function to be maximized by the solver or as a set of constraints that the camera configuration must satisfy.

Optimization based systems, like Halper and Olivier's CAMPLAN (Halper and Olivier 2000), always find the best camera configuration for the given requirements but their computational cost is high. On the other hand, constraint fulfilling systems (Jardillier and Languènou 1998) are much more efficient but may not return any result if there is not configuration respecting all the frame requirement.

Bares (Bares and Lester 1999) addressed the issue by identifying conflicting constraints and produce multiple camera configurations corresponding to the minimum number of non-conflicting subsets. Bourne (Bourne and Sattar 2005) extended Bares's solution by adding a *weight* property to each constraint to define a relaxation priority. A third set of generalized approaches (Pickering 2002; Christie and Normand 2005; Burelli et al. 2008), combines constraints satisfaction to select feasible volumes (therefore reduce the size of the search space) and optimization to find the best camera configuration within these spaces.

These approaches on finding the best camera configuration to obtain a specified shot, other researchers have instead investigated the problem of camera animation. Beckhaus(Beckhaus, Ritter, and Strothotte 2000) employed Artificial Potential Fields to generate collision-free camera paths through a virtual environment, Xiao (Xiao and Hubbold 1998) employed force fields to simulate gravity in camera motion and help the camera controller to avoid collisions.

We extend Beckhaus's work and apply Artificial Potential Fields (Khatib 1986), a well established technique for robot motion planning, to camera control addressing both camera movement and frame composition problems. Frame constraints, typical of the generalized approaches, are modelled into potential fields and these are used to compute both camera position and camera orientation.

CamOn

CamOn is an autonomous camera system capable of generating smooth camera animations and solving camera composition tasks. The system iteratively animates the camera to converge to an optimal camera position, at each iteration CamOn takes the current camera position, frame description and scene description as input, and returns a new camera position as output.

Frame description requires identification of the frame subjects, definition of subject importance and definition of composition rules. Composition defines disposition of visual elements in an image (Arijon 1991); following the model proposed by Bares (Bares et al. 2000) we have translated these composition rules into soft constraints.

Current version of the system supports three constraints:

- **Visibility**
  This constraint defines what fraction of the subject must be visible in the frame, fraction is defined as a number between 0.0 if subject is not visible to 1.0 if subject is fully visible.
- **ProjectionSize**
  This constraint defines the size of subject in the frame, size is defined as the quotient between frame height or width and the relative longest side of the subject's projected bounding box.
- **ViewAngle**
  This constraint defines the angle from which the camera should shoot the subject, angle is defined using spherical coordinates.

Each of these constraints for each subject and the scene geometry is modelled into the system using Artificial Potential Fields. Camera position and look-at point are modelled as particles moving along the potential field so created this way.

Artificial Potential Fields

Artificial Potential Fields (Khatib 1986) is an iterative technique commonly adopted in the area of robotics for controlling the navigation of robots in dynamic environments. Robots are modelled as particles moving in a field of potentials attracted by low potential areas; the position to be reached generates an attractive force (a low potential zone) and obstacles generate repulsive forces (high potential zones). At each iteration the particle moves along the force resulting from the sum of all repulsive (obstacle avoidance) and attractive (goals) forces influencing current particle position; the particle continues to move until it reaches a stable state.

CamOn system employs Artificial Potential Fields to move camera and solve frame composition tasks: obstacle avoidance is modelled using repulsive forces and frame composition is obtained by translating frame constraints into forces affecting both the position and the look-at point of the camera.

Each frame constraint produces one force attracting or repulsing camera position and one force attracting or repulsing camera look-at point, the system treats these two aspect of the camera as two different particles moving into two different potential fields. An example of the two potential fields created by a frame constraint (the target sphere has to be fully visible in the projected frame) can be seen in Figure

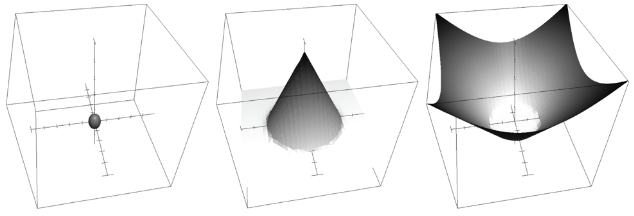


Figure 1: Example APF produced by a visibility constraint on a sphere (from left to right: sphere, position APF, orientation APF)

1, the two potential fields shown are a sample of the 3D field measured along the horizontal plane passing through the sphere centre. The particle is attracted by the low potential (white) areas. The two potential fields are linked by constraint satisfaction values as it is explained more in depth in the next section. At each iteration CamOn moves both camera's position and orientation; this makes constraint satisfaction values change concurrently and the fields dynamically change during camera movement according to the new particles positions.

CamOn handles over-constrained and under-constrained tasks effectively. When there is no camera configuration that can satisfy all the constraints imposed to the camera (over constrained task) the system converges to the nearest configuration which optimizes frame requirements (giving higher priority to more important subjects). When more than one configuration potentially satisfies frame requirements the system converges to one of the feasible configurations but lets the camera freedom of movement within the limits imposed by frame constraints. This feature permits to let the user to control the camera without loosing the possibility to impose some constraints on camera position, orientation and movement.

Recasting frame constraints into potential fields

Translating frame constraints into potential fields requires identification of position and orientation goals corresponding to each frame constraint.

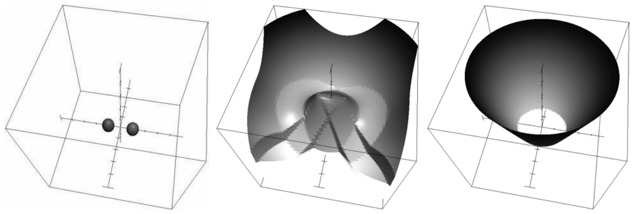


Figure 2: Example APF produced by a two front-shot frame description (from left to right: spheres, position APF, orientation APF)

Ideal camera positions and orientations for each constraint are modelled as low potential zones; any other part of the solution space has a potential proportional (the exact relation can vary from constraint to constraint) to the distance from the ideal position and to the constraint satisfaction of the corresponding camera configuration.

The forces imposed by each constraint are defined as follow:

- **Visibility**
  Assuming α as the current visible fraction value and α_{exp} the desired one, $\vec{A}$ as the camera position and $\vec{B}$ as subject position, the force $\vec{V}_{pos}$ applied to the camera position is:

$$\vec{V}_{pos}(\vec{B}) = \begin{cases} \frac{(\vec{B}-\vec{A})_{norm}(\alpha-\alpha_{exp})}{(1-\alpha_{exp})} & \alpha < \alpha_{exp} \\ 0 & \alpha \geq \alpha_{exp} \end{cases} \tag{1}$$

  Assuming α as the current visible fraction value and α_{exp} the desired one, $\vec{A}$ as the camera view direction and $\vec{B}$ as subject position relative to the camera, the force $\vec{V}_{orient}$ applied to the camera look-at point is:

$$\vec{V}_{orient}(\vec{B}) = \begin{cases} \frac{(\vec{B}-\vec{A})_{norm}(\alpha-\alpha_{exp})}{1-\alpha_{exp}} & \alpha < \alpha_{exp} \\ \frac{(\vec{A}-\vec{B})_{norm}(\alpha_{exp}-\alpha)}{\alpha_{exp}} & \alpha \geq \alpha_{exp} \end{cases} \tag{2}$$

- **ProjectionSize** (only position)
  Assuming α as the current projection size value and α_{exp} the desired one, $\vec{A}$ as the camera position and $\vec{B}$ as subject position, the force $\vec{V}_{pos}$ applied to the camera position is:

$$\vec{\Delta} = \begin{cases} (\vec{B}-\vec{A}) & \alpha < \alpha_{exp} \\ (\vec{A}-\vec{B}) & \alpha \geq \alpha_{exp} \end{cases}$$

$$\vec{V}_{pos}(\vec{B}) = \begin{cases} \vec{\Delta}_{norm}(\frac{|\vec{\Delta}|(\alpha-\alpha_{exp})}{1-\alpha_{exp}})^2 & \alpha < \alpha_{exp} \\ \vec{\Delta}_{norm}(\frac{|\vec{\Delta}|(\alpha_{exp}-\alpha)}{\alpha_{exp}})^2 & \alpha \geq \alpha_{exp} \end{cases} \tag{3}$$

- **ViewAngle** (only position)
  Assuming α and θ as the current view angle of the camera and α_{exp} and θ_{exp} the desired ones, $\vec{A}$ as the camera position, $\vec{B}$ as subject position, $\vec{F}$ as the normalized subject front vector and $\vec{F}_{rot(\alpha,\theta)}$ as the desired view direction (front vector rotated by α and θ), the force $\vec{V}_{pos}$ applied to the camera position is:

$$\begin{aligned} \lambda &= \frac{1}{2}(\frac{|\alpha_{exp}-\alpha|}{\pi} + \frac{|\theta_{exp}-\theta|}{2\pi}) \\ \vec{A}_{exp} &= |\vec{A}|\vec{F}_{rot(\alpha,\theta)} \\ \vec{V}_{pos}(\vec{B}) &= (\vec{A}_{exp}-\vec{A})_{norm}(\lambda|\vec{A}_{exp}-\vec{A}|)^2 \end{aligned} \tag{4}$$

Force value at each point is described as a linear combination of scalar weighted forces, where each force corresponds to a frame constraint and each scalar value to the relative subject importance; the resulting values define the gradients of the potential field.

Figure 2 shows the potential field associated to a frame composed by two spheres with equal importance value both with Visibility set to 1, ProjectionSize set to 1 and ViewAngle set to (0,0).

(a) Camera facing another robot (b) Camera orients towards the target (c) Camera approaches the target maximizing size (d) Camera aligns to the right view angle

Figure 3: Camera converging to a right-profile shot of a robot (listing 1)

Implementation and Experimental Results

CamOn is currently implemented as a library of classes for camera control related functions in Ogre3D [1]. The library includes functions for setting camera parameters, setting constraints, and automatically computing the motion of the camera. A set of desired camera parameters (location, orientation, and field-of-view) identifies a camera profile and it is defined as a set of subjects (each with an importance value) and a set of viewing constraints applied to each subject. Viewing constraints take the form of desired composition parameters like primary object, object occlusion, etc. At each frame the system computes the current camera position and orientation velocities which are then used to compute the new camera configuration; the two velocities are computed at each iteration with a 4th order Runge-Kutta. The library facilitates change of the number of iterations per frame and the Runge-Kutta speed factor to tune camera speed.

A small demo application has been developed to test CamOn behaviour and performance in a game-like 3D environment. The application shows a non-interactive intro sequence showing the main character (a green ninja) and the enemies he has to avoid (4 steady robots and 2 walking robots). The user can watch the default camera animation (a predetermined sequence of camera profiles) or can interactively decide to select a specific camera profile for composing a shot of any of the characters in the scene. Camera position and aim direction can be modified by the user continuously during the demo and the freedom of control depends on the currently loaded camera profile. The camera definition file included in the demo can also be modified to test the camera animations corresponding to different constraint values and weights.

In the next sections we show how the camera moves and satisfies a simple camera profile with one subject and how camera deals with moving subjects and over-constrained requirements. We'll explain CamOn behaviour with under-constrained frame requirements and finally we show system's run-time performance.

Camera movement

Figure 3 shows how CamOn converges to a right-profile shot of a target robot starting from another shot (figure 3(a)); the camera requires approximately 3 seconds to reach the steady configuration (figure 3(d)).

Visibility is the first constraint to be satisfied (figure 3(b)), meanwhile camera approaches the target subject to satisfy ProjectionSize (figure 3(c)) constraint and finally moves to the stable configuration to satisfy also the ViewAngle constraint (figure 3(d)).

The sequence depends on the function used to model different constraint forces and especially on the size of their area of interest.

Listing 1: Right-Profile shot camera profile

```
<subject name=”Robot1” importance=”1.0”>
  <visibility visibleFraction=”1.0”/>
  <viewAngle horizontal=”100” vertical=”45”/>
  <projectionSize size=”1.0” />
</subject>
```

The time required by the camera to reach the optimal configuration, within performance limits, depends primarily on the chosen animation speed and the starting camera position.

Performance limits are directly connected to the integration method used to search the optimal configuration through the potential fields. Increasing the convergence speed increases also computational error (in the fourth-order RungeKutta method the total accumulated error has order h^4 where h is the convergence speed factor). On the other hand, increasing the number of iterations per frame leads to a performance drop, so the speed limit depends on the tradeoff between camera placement accuracy and system performance.

Over-constrained profile

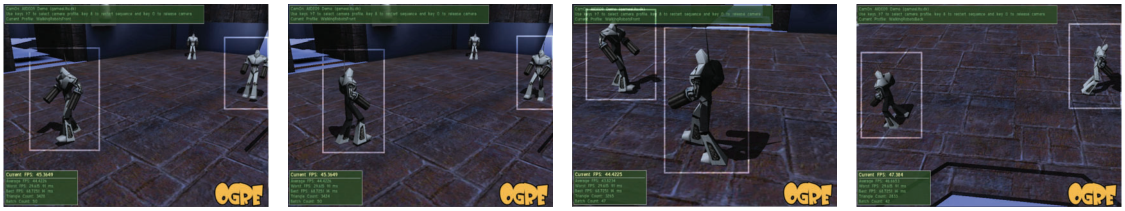
Figure 4 shows how CamOn deals with the camera requirements defined in listing 2 in a dynamic scene. The camera system is instructed to generate and maintain a three-quarter front shot of the two walking robots (4(a)), however, as robots during their patrol face opposite directions (4(b)), the problem becomes over-constrained and no camera configuration can satisfy the requirements anymore.

Listing 2: Three-quarter front two-shot profile

```
<subject name=”WalkingRobot1” importance=”1.0”>
  <visibility visibleFraction=”1.0”/>
  <viewAngle horizontal=”−30” vertical=”10”/>
  <projectionSize subject=”RobotW1” size=”1.0” />
</subject>
```

---

[1] Object-Oriented Graphics Rendering Engine - http://www.ogre3d.org



(a) Robots are distant but face the same direction (b) One robot changes suddenly walking direction (c) Robots get near (d) Robots get distant again facing opposite directions

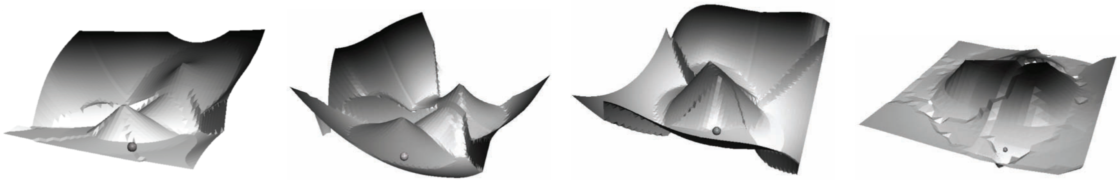


Figure 4: Camera searching an optimal configuration for a three-quarter front shot of two walking robots (listing 2); a 2D sample of the corresponding position potential field is shown below each frame with camera represented as a small sphere

```
<subject name=WalkingRobot2 importance="1.0">
    <visibility visibleFraction="1.0"/>
    <viewAngle vertical="−30" vertical=10/>
    <projectionSize size="1.0" />
</subject>
```

The camera starts searching for a new stable configuration while the potential fields change dynamically according to the robots movement; figure 4(c) shows how camera fully satisfies the Visibility and ProjectionSize constraints and figure 4(d) shows the camera satisfying only the visibility constraints due to the distance between the two robots.

Under-constrained profile

Listing 3 shows a profile of the same two robots in figure 4 which doesn't specify any preference about view angle. This is an example of under-constrained composition problem since there are several camera configurations that satisfy frame requirements.

Listing 3: Under-constrained two-shot profile

```
<subject name="WalkingRobot1" importance="1.0">
    <visibility visibleFraction="1.0"/>
    <projectionSize subject="RobotW1" size="1.0" />
</subject>
<subject name=WalkingRobot2 importance="1.0">
    <visibility visibleFraction="1.0"/>
    <projectionSize size="1.0" />
</subject>
```

In this case the camera control system converges from any starting configuration to the nearest configuration satisfying all the constraints; the user can move both camera position and orientation while the system continuously limits camera movements to keep constraints satisfaction.

Performance

The experiment was run on a Apple Macbook equipped with an Intel Core 2 Duo (2.0GHz) and 4Gb of RAM (DDR3 1.07 GHz) running MacOS 10.5.6; the demo was implemented using Ogre3D version 1.6 (Einhort).

Table 1 shows average execution time required to compute the force related to each implemented constraint (ViewAngle and ProjectionSize constraint don't generate any orientation force). CamOn computes at each frame the sum of the forces on the current particle position for both camera location and orientation; this is done 4 times due to the 4th order Runge-Kutta integration.

Constraint	Average time (ms)
ProjectionSize (Position)	0.08647
ViewAngle (Position)	0.02401
Visibility (Position)	0.04477
Visibility (Orientation)	0.04350

Table 1: Constraints execution times collected in approximately 10,000 runs

The shot shown in figure 3, with one subject and three constraints, requires approximately 0.86 milliseconds per frame; even with multiple subjects and multiple constraints per subject (a shot with 10 subjects would not take more than 9 ms per frame), the system maintains real-time performance.

Conclusions and future work

This paper describes CamOn, an autonomous camera control system base on Artificial Potential Fields. The system solves camera motion and frame composition problems by extending Beckhaus work on Artificial Potential Fields with frame constraint based forces.

Two different potential fields are generated from scene's geometry and frame constraints; camera position and camera look-at point are modelled as particles moving through the two potential fields. As potential fields don't need to be precomputed and can be calculated locally at each iteration step, CamOn performances are independent from scene size.

Further, the camera converges to a solution from any initial configuration and can converge to a solution automatically even if the user takes over control in the middle of the convergence process.

While CamOn satisfies all the four major requirements mentioned by Bourne (Bourne and Sattar 2005), viz. autonomy, reactivity, real-time performance, and dynamic environment handling, there are many aspects that can be still improved. Cinematic camera movement principles could be incorporated to influence camera motion. CamOn's current implementation only supports basic object view constraints, more composition constraints, like occlusion or in-frame position, could be introduced to match the expressiveness of state-of-art composition systems such as the one by Bares et. al. (2000). Our current implementation of Artificial Potential Fields suffers of local minimum problem which could occur in complex environments like mazes and lead to undesirable local minima with occluded views.

This issue might be solved using multiple particles placed near the attractive points and making the camera follow the one converging to the best value. A Global Optimization algorithm could also be run parellelly to to Artificial Potential Fields based algorithm to tune the forces according to the global optimum as soon as this is found.

This paper demonstrates the use of APFs for camera composition and movement in real-time interactive 3D virtual environments. Initial results of this implementation are encouraging. One of the long-term goals of this implementation is to provide designers with editing interfaces for defining APF profiles for different objects in the environment. Rather than modifying constraint values between 0 and 1, the colored visual representation of a potential field might be more accessible to designers. Potential fields for collision of the camera with dynamic world objects can also be automatically extracted from geometry information available in the spatial data-structure of the game engine.

References

Arijon, D. 1991. *Grammar of the Film Language*. Silman-James Press LA.

Bares, W. H., and Lester, J. C. 1999. Intelligent multi-shot visualization interfaces for dynamic 3d worlds. In *IUI '99*, 119–126. ACM.

Bares, W.; McDermott, S.; Boudreaux, C.; and Thainimit, S. 2000. Virtual 3d camera composition from frame constraints. In *MULTIMEDIA '00*, 177–186. ACM.

Beckhaus, S.; Ritter, F.; and Strothotte, T. 2000. Cubicalpath - dynamic potential fields for guided exploration in virtual environments. In *PG '00*. IEEE Computer Society.

Blinn, J. 1988. Where am i? what am i looking at? *IEEE Comput. Graph. Appl.* 8(4):76–81.

Bourne, O., and Sattar, A. 2005. Applying constraint weighting to autonomous camera control. In *AIIDE '05*, 3–8.

Burelli, P.; Di Gaspero, L.; Ermetici, A.; and Ranon, R. 2008. Virtual camera composition with particle swarm optimization. In *Smart Graphics*, 130–141. Springer-Verlag.

Christie, M., and Normand, J.-M. 2005. A semantic space partitioning approach to virtual camera composition. *Computer Graphics Forum* 24(3):247–256.

Christie, M., and Olivier, P. 2006. Camera Control in Computer Graphics. 89–113. Eurographics Association.

Drucker, S. M., and Zeltzer, D. 1994. Intelligent camera control in a virtual environment. In *Graphics Interface 94*, 190–199.

Gleicher, M., and Witkin, A. 1992. Through-the-lens camera control. In *Computer Graphics*, 331–340.

Halper, N., and Olivier, P. 2000. Camplan: A camera planning agent. In *Smart Graphics 2000 AAAI Spring Symposium*, 92–100.

Jardillier, F., and Languènou, E. 1998. Screen-space constraints for camera movements: the virtual cameraman. *Computer Graphics Forum* 17(3):175–186.

Khatib, O. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Rob. Res.* 5(1):90–98.

Pickering, J. 2002. *Intelligent Camera Planning for Computer Graphics*. Ph.D. Dissertation, University of York.

Ware, C., and Osborne, S. 1990. Exploration and virtual camera control in virtual three dimensional environments. *SIGGRAPH* 24(2):175–183.

Xiao, D., and Hubbold, R. 1998. Navigation guided by artificial force fields. In *CHI '98*, 179–186. Addison Wesley.