

Virtual Camera Planning: A Survey

Marc Christie¹, Rumesh Machap², Jean-Marie Normand¹, Patrick Olivier² and Jonathan Pickering³

¹ University of Nantes

² University of Newcastle Upon Tyne

³ University of York

Abstract. Modelling, animation and rendering has dominated research computer graphics yielding increasingly rich and realistic virtual worlds. The complexity, richness and quality of the virtual worlds are viewed through a single media that is a virtual camera. In order to properly convey information, whether related to the characters in a scene, the aesthetics of the composition or the emotional impact of the lighting, particular attention must be given to how the camera is positioned and moved. This paper presents an overview of automated camera planning techniques. After analyzing the requirements with respect to shot properties, we review the solution techniques and present a broad classification of existing approaches. We identify the principal shortcomings of existing techniques and propose a set of objectives for research into automated camera planning.

1 Introduction

At a very basic level one of the objectives of photography and cinematography is to capture and convey information. Deciding where to position a camera or how to move a camera necessarily raises questions as to what information is to be conveyed and how this will be achieved. We propose three levels of description for the properties of an image: geometric, perceptual and aesthetic. Geometric properties capture the absolute and relative screen position, orientation and sizes of objects. Perceptual properties refer to intermediate stages of the visual processing pipeline, for example, the occurrence of visual gestalts and other properties that impinge on our recognition of objects and their spatial relations with each other. Aesthetic properties relate to notions of shot composition and are typified by terms frequently used (but hard to algorithmically characterize) by artists and art scholars, for example, compositional balance and unity.

In transposing these notions to virtual environments, researchers have been working on approaches to assist and automate positioning and path planning for virtual cameras. A common approach is to invoke declarative techniques by which a user articulates the properties required in the shot (*e.g.* what should be on the screen, where on the screen, which vantage angle) and a solver computes a solution, set of solutions, or best approximation to a solution. To date most actual systems rely solely on geometric properties. This paper presents a survey

of virtual camera planning techniques, and we structure our review by referring to two criteria:

1. the expressivity of the set of properties *i.e.* the assumptions pertaining to the properties, the qualitative and quantitative characteristics as well as the range of possible properties;
2. the characteristics of the solving mechanisms (*e.g.* generality, optimisation, local-minima failure, and computational cost).

In Section 2 we present the principles of camera planning and cinematography as they apply to the use of real world cameras. Section 3 reviews the uses of the cameras in computer games, a demanding practical field of application, and in Section 4 we review existing research before concluding with our requirements for future research.

2 Camera Planning and Cinematography

Direct insight into the use of real-world cameras can be found in reports of photography and cinematography practice [1, 22, 21]. Cinematography encompasses a number of issues in addition to camera placement including shot composition, lighting design and staging (the positioning of actors and scene elements) and an understanding of the requirements of the editor. For fictional film and studio photography camera placement, lighting design and staging are highly interdependent. However, documentary cinematographers and photographers have little or no control of staging and we consider accounts of camera placement in cinematography within this context. Indeed, real-time camera planning in computer graphics applications (*e.g.* computer games) is analogous to documentary cinematography whereby coherent visual presentations of the state and behavior of scene elements must be presented to a viewer without direct modification to the position or orientation of the elements.

2.1 Camera Positioning

Whilst characterizations of cinematography practice demonstrate a degree of consensus as to best practice, there is considerable variation in its articulation. Accounts such as Arijon's [1] systematically classify components of a scene (*e.g.* according to the number of principal actors) and enumerate appropriate camera positions and shot constraints. Not surprisingly, Arijon's procedural description of camera placement has been cited as the motivation for a number of existing automatic camera planning systems. By contrast accounts such as Mascelli's [22] are a less prescriptive and formulate camera planning in terms of broader motivating principles, such as narrative, spatial and temporal continuity.

2.2 Shot Composition

Camera positioning ensures the general spatial arrangement of elements of the scene with respect to the camera, thereby placing a coarse constraint on the

composition of a shot. That is, the position (and lens selection) determines the class of shot that is achievable, which is typically classified accordingly to the proportion of the subject included in the shot as: close up (*e.g.* from the shoulders), close shot (*e.g.* from the waist), medium shot (*e.g.* from the knee), full shot (*e.g.* whole body) and long shot (*e.g.* from a distance). However, precise placement and orientation of the camera is critical to achieving the layout of the scene elements in shot — referred to as the *composition* of the shot.

Composition is variously characterized in terms of shot elements including lines, forms, masses, and (in the cases of action scenes) motion. In turn, shots are organized to achieve an appropriate (usually single) center of attention, appropriate eye scan, unity, and compositional balance (arrangements of shot elements that affords a subconsciously agreeable picture). As psychological notions these terms are problematic to characterize and the empirical investigation of visual aesthetics is very much in its infancy. However, the validity or significance of the notions themselves cannot be questioned; eye tracking studies have demonstrated significant differences between the viewing behavior of observers of subjectively agreed balanced and unbalanced (through artificial modification) works of art [24].

Scenes that comprise significant amounts of motion and action pose different problems for cinematographers and editors, although general heuristics such as the triangle principle, use of a line of action, and compositional rules can be extended to these more complex configurations. The challenge for camera planning is to algorithmically formulate these principles in a manner appropriate to the particular application to be addressed.

3 Camera Control in Computer Games

Camera systems are increasingly becoming decisive elements to the success of computer games. With the advent of near photo-realistic graphics and the use of powerful and captivating story-driven plots to bring life to games, cameras in games have been neglected. As we have seen in Section 2 film practitioners have characterized standard camera configurations and transitions in terms of a number of cinematographic principles. The use of cinematographic properties and intelligent camera controls have the potential to heavily influence the look and feel of games and give game designers access to the film director visual toolbox.

Camera systems in modern day games can be classified into three different categories;

1. *First person camera systems*: users control the camera (allowing them to have to feel of being the character in virtual environment and looking from their eyes). There are a multitude of games that use first person camera views, notably the Doom series.
2. *Third person camera systems*: the camera system relies on tracking the character from a set of fixed positions, constantly changing the cameras posi-

tion based on the environment and the users interactions with the environment. This mode of camera systems presents a problem when the views presented by the camera do not co-adhere with the current events in the game, *e.g.* when a character leans on to a wall, the camera defaults to moving to the front of the player, disrupting the game play by blocking the view of the opponents.

3. *Action replay camera systems*: replays are heavily used in modern games, to highlight notable scenes in a certain game, and it is imperative that the images generated by the camera system during the replay are meaningful.

For example, *Tomb Raider*⁴ is a successful series of computer games that has been both widely praised and criticized for its use of a dynamic camera. The game utilizes a third-person view with respect to the main character, where the camera is *attached* to the character. The camera system employed was rather limited in expressing informative shots when in tight spots; often leading to situations where the camera displayed awkward views, preventing the user from playing the game as it was intended. The camera system computed its next best position without significant consideration of the visual properties and the ongoing action within the environment.

*Full Spectrum Warrior*⁵, an action war military simulator, uses an advanced (in computer games terms) camera system to allow the player to effectively manage teams of soldiers. Its main feature is the *auto look* facility which helps the user by presenting a shot that handles occlusion to prevent the view from being blocked by an object (wall) through the use of ray casting. The fly-by sequences performed by the camera also avoid collisions with environmental objects by applying the same occlusion detection method. *Jump cuts* are used to handle situations when the only evident path is to move through a wall or an obstacle, whereby the camera jumps to the scene beyond the wall, avoiding the unnatural view of going through a wall. While *Full Spectrum Warrior* does constitute a step forward in the use of cameras in games, it lacks the cinematic expression that is apparent in film. Cinematic camera transitions, which preserve the flow of a story, are not apparent in camera systems games as they require heavy computations to handle positioning, orientation, occlusion handling and other image properties as well as defining good shots based on a certain narrative.

Camera shots are the heart of producing truly interactive visual applications. The consideration of cinematic properties and literature (narrative) should provide cues to automatic generation of camera shots that are both intelligent and meaningful, enabling games that give the look and feel of film. Games present an inherent difference to film as most games are controlled by the player and define a dynamic environment. Therefore the problem for automated camera systems for games is intrinsically more complex than for their static counterparts, or for systems where the temporal evolution of action is known in advance. While automation of camera shots based on cinematographic principles present meaningful shots, the use of editing techniques (which are very rare indeed within

⁴ Tomb Raider, Eidos plc, www.eidos.com

⁵ Full Spectrum Warrior, Pandemic Studios, www.pandemicstudios.com

games today) can preserve the game-play by presenting jump-shots or cut-scenes to show the user only what is intended. Indeed, such technologies would reduce the confusion that is evident in many games.

4 Virtual Camera Planning Systems

In this section, we present a review of a number of approaches to assist or automate the setting up of a virtual camera. The problem is a complex one involving the geometric properties of 3D model worlds together with the higher level problem of describing the desired properties of the output. Our review is structured around two criteria: the expressivity of the properties and the nature of the solving mechanisms.

4.1 Expressivity

Most systems share a similarly declarative approach and identify a common set of properties by which a shot may be specified. We distinguish *on-camera* and *on-screen* properties. On the one hand, *on-camera* properties define the location, orientation and movement of the camera in the 3D scene. This encompasses scale shots (*e.g. establishing shot, long shot, close shot*) that define the distance from the camera to the main subject, the vantage point (*e.g. low angle, high angle*), and classical camera motions such as *travelling, dolly, and zoom* shots. Most recent static camera planning [14, 25, 19, 28] and dynamic camera planning approaches [32, 12, 4, 18, 20, 11] rely on this grammar by providing a description language. Pickering *et al.* [29] propose an IDL language (Image Description Language) that meets these requirements.

By contrast, *on-screen* properties allow the specification of where the objects are located in the screen (in a relative, absolute or approximate manner) and other properties of the shot itself (*e.g. size, orientation of objects and possible occlusion*). Here the approaches differ in the range of properties they consider and in their qualitative or quantitative nature. For example, Blinn's initial work [7] and its derivatives [8, 10] abstract objects as points and their exact location on the screen must be provided. Such accounts are intrinsically restricted to no more than two objects on the screen otherwise the specification becomes over-constrained. Some interactive approaches [17] offer a precise control of characteristic points (*e.g. edges of a cube*) by controlling the path of their image on the screen. In a less contrived way, some approaches allow the specification of object position with respect to on-screen areas such as rectangular frames [20, 11, 3, 28]. These match the level of specification found in storyboards, but can prove inadequate for complex objects.

The occlusion property (*i.e. the object to view should not be, or should only be partially, occluded*) is fundamental in camera planning. Most approaches use abstractions of the objects, such as spheres, and compute occlusion using their projection on the screen [25, 11]. Though easy to compute, it is clear that for

many objects and scenes sphere-based approximation of objects cannot adequately model occlusion. Bares *et al.* [6] prune the search space by projecting the occluders onto discretised hemispheres located in the center of each object and by intersecting all the volumes. More sophisticated techniques that rely on raycasting [23] and hardware rendering capacities [12, 18] provide effective occlusion tests.

While a broad range of properties has been implemented, many compositional techniques have been neglected such as perspective lines, simple primitives and virtual curves shaped by a set of objects of interest in the screen.

4.2 Solving Mechanisms

We propose a description of solving mechanisms based on four classes of approaches:

- algebraic systems** represent the problem in vector algebra and directly compute a solution;
- interactive systems** set up cameras in response to directives from a user, who continuously observes the output image;
- reactive real-time systems** rely on robot motion planning mechanisms in a dynamic virtual environment;
- optimisation and constraint-based systems** model the properties as constraints and objective functions and rely on a broad range of solving processes that differ in their properties (*e.g.* completeness, incompleteness, and softness).

Algebraic systems In an algebraic system a camera set-up is regarded as the solution to a vector algebra problem defined on the model world being viewed. This limits the systems to examples of this class, imposing a lack of flexibility on the system. The congruent advantage being that the dedicated solution is usually computationally efficient. The assumptions made on the graphical entities are strong as objects are to be considered as points and the expressivity is limited to locating two entities in the screen space.

The earliest example of such a system was the work of Blinn [7] at NASA. Working on the visualization of space probes passing planets, he developed a system for setting up a camera so that the probe and planet appeared on-screen at given coordinates with given sizes. Attempts to generalize these systems have relied on the use of idioms, standard lay-outs of subjects and cameras commonly used in cinematography. In these systems solution methods are devised for each lay-out, and the input consists of a model world together with a list of idioms to apply. Such system have been developed by Butz [8] and Christianson [10].

Vector algebra approaches have also been studied in purely 2D-based applications such as cel animation (motion-picture cartoons) or virtual 2D guided tours (presentation of different artworks such as frescos, tapestries or paintings). The need for camera planning algorithms for cartoon animation has been tackled by Wood *et al.* in [32], in association with the Walt Disney Animation Studios.

Their system generates the panoramas by relying on some basic camera moves such as pan, tilt-pan, zoom and truck. Two dimensional “multimedia guided tours” aim is to help the visitors during their visit by providing additional information on the artworks. Zancanaro *et al.* explored the opportunities to use the PDAs in multimedia museum tours in [34, 33]. Their approach consists of planning camera movements on *still* images *i.e.* the frescos or paintings exposed in the museum. In a similar way, Palamidese [26] proposes the description of art by planning camera movements that first show details and then zooms out to show an entire work.

Interactive systems Interactive control systems provide the user with a view of the model world and modify the camera set-up in response to input by the user. The immediate problem with such systems is how a user’s input devices will map onto the properties of the camera. Ware and Osborne [31] reviewed the possible mappings, which they referred to as camera control metaphors. One of the first systems to implement such control was developed by Phillips [27], for the human figure modelling system *Jack*. *Jack* could not properly manipulate model figures about axes parallel or perpendicular to the view direction. The camera control system prevented this occurring by repositioning the camera. It could also reposition the camera to make a selected object visible if it was off screen and manage occlusion via z-buffer rendering.

A system that allows a user to move a camera through a dynamic environment using a pointing device, was developed by Gleicher and Witkin [17]. In this approach the difference between the actual screen properties and the desired properties input by the user was treated as a force. This was applied to the camera set-up, which itself was treated as a body in a classical mechanics system.

Reactive Real-time Approaches Applications of camera control to reactive environments require specific approaches that emphasizes on continuity, smoothness and occlusion criteria w.r.t. a target. Virtual camera planning while following a single target is very close to visual servoing in robotics (specifying a task as the regulation in the image of a set of visual features), and can share similar solving techniques. In [12], a visual servoing approach is proposed that integrates constraints on the camera trajectory. If the primary task (following the object) does not instantiate all the camera parameters, secondary tasks may be added (*e.g.* occlusion or lighting).

A similar application in the domain of computer games has been explored by Halper *et al.* [18]. Here an ad-hoc incremental solving process has been proposed to satisfy at each frame a set of constraints on the screen (*e.g.* height angle, angle of interest, size, visibility).

Optimisation and Constraint-based Approaches Static virtual camera planning also referred as virtual camera composition (VCC) and dynamic virtual camera planning (VCP) problems can both be viewed as constrained and/or

optimisation problems. The set of properties of the shots (*e.g.* framing, orientation, zoom factor) can be expressed as numerical constraints (properties that must hold) or objective functions (properties to be maximized/minimized) on the camera variables (respectively the camera’s path variables). The general process is to search the space of camera set-ups for one that maximizes/minimizes the objective functions while satisfying the constraints. A broad range of procedures is then available and the solvers differ in the way to manage over-constrained and under-constrained cases, in their complete or incomplete search capacities, in local minima failures and possible optimisation processes (generally finding the best solution with respect to an objective function).

We classify the “constraint-and-optimisation”-based approaches the following way:

complete methods perform an exhaustive exploration of the search space, thus providing the user with the whole set of solutions to the problem, or none if the problem is inconsistent. This is generally achieved by a computational and time expensive dynamic programming approach (*split* and *explore*).

incomplete methods are mostly based on a stochastic investigation of the search space, thereby computing a unique solution. The output of an incomplete method can either be a solution to the problem, or an approximation of a solution as these methods try to minimize errors given objective functions.

hybrid methods rely on cooperation between different techniques to simultaneously manage constraints and objective functions.

The classification of the solving strategies for the VCC and VCP problems is summarized in Figure 1.

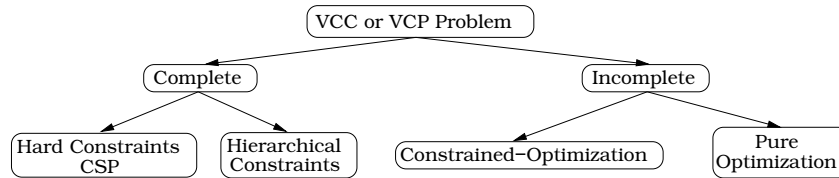


Fig. 1. Taxonomy of the optimisation and constraint-based approaches to VCC (Virtual Camera Composition) and VCP (Virtual Camera Planning) problems.

Complete Approaches. The Constraint Satisfaction Problem (*CSP*) framework has proven to succeed in some camera composition and motion planning approaches. However these can differ in the way they handle over-constrained problems. We distinguish the *hard constraints* approaches [20, 11] from the *hierarchical constraints* ones [4]. Hard constraint approaches such as Jardillier & Langu  nou [20] (the *Virtual Cameraman*) use pure interval methods to compute

camera paths which yield sequences of images fulfilling temporally indexed image properties. The use of pure interval methods has been improved by Christie *et al.* in [11] by describing camera trajectories as sequences of parameterized elementary camera movements called *hypertubes*. Unlike most approaches which only guarantee the correctness of user-defined properties for a set of points on a camera trajectory (generally the starting and ending points plus some keypoints taken on the camera path), interval-based methods guarantee the fulfillment of the properties during a whole film sequence. Unfortunately, the benefits of completeness of interval-based techniques are counterbalanced by the computational effort required, and the fact that there is no mechanism for constraint relaxation. However, since the method explores the whole search space, the user has a guarantee that there are no solutions whenever the method fails.

In contrast, hierarchical constraints approaches are able to relax some of the constraints in order to give the user an approximate solution of the problem. Bares *et al.* propose to use a partial constraint satisfaction system named CONSTRAINTCAM [4] in order to provide alternate solutions when constraints cannot be completely satisfied. If CONSTRAINTCAM fails to satisfy all the constraints of the original problem, the system relaxes weak constraints and, if necessary, decompose a single shot problem to create a set of camera placements that can be composed in multiple viewports [5] providing an alternate solution to the user.

Incomplete Approaches. In general, incomplete approaches allow more flexibility in the definition of the problem; the main difference is in the kind of solving procedure applied to the problem. The incomplete search can be based on pure *optimisation* procedures (*e.g.* descent methods) or stochastic search methods (*e.g.* local search, genetic algorithms, simulated annealing).

Indeed as early as 1992, Drucker *et al.* [13] proposed CINEMA, a general system for camera movement. CINEMA was designed to address the problem of combining different paradigms (*e.g.* *eyeball in hand*, *scene in hand*, or *flying vehicle* metaphors [31]) for controlling camera movements. This early approach incited the authors to explore the constraint satisfaction methodology to address some of the CINEMA's problems. In [14, 15] Drucker and Zeltzer improved their previous ideas and developed the CAMDROID system which specifies behaviours for virtual cameras in terms of task level goals (objective functions) and constraints on the camera parameters. They regroup some primitives constraints into *camera modules* which represent a higher level means of interaction with the user. The constraints of each module are then combined by a constraint solver and solved by a camera optimizer based on the *CFSQP* (Feasible Sequential Quadratic Programming coded in C) package, which has been designed to solve large scale constrained non-linear optimisation problems [14].

Unfortunately this initial attempt at constrained optimisation effort does not offer a systematic solution for handling constraint failures that can occur frequently in dynamic environments with complex scene geometries and requires a good initial guess for correct convergence.

In order to address the major drawbacks of their CONSTRAINTCAM system, Bares *et al.* [3, 2] proposed the use of a heuristic search method that uses a constraint's allowable minimum and maximum values to reduce the size of the 7 degrees of freedom search space of possible camera positions (x, y, z) , orientations (dx, dy, dz) and field of view angles (fov). Their method combine both constraints and objective functions in a constrained-optimisation algorithm. This was addressed by Pickering and Olivier [29], who defined an Image Description Language (IDL), a context free grammar allowing properties of images to be defined. The IDL could be parsed into hierarchies of constraints and objectives, which were then subject to constrained optimisation using a genetic algorithm [28].

However, optimisation-based techniques rise the delicate question of how to provide the weights of each property and how to aggregate them in the objective function in a way that is stable and efficient whatever the description may be. Moreover, most incomplete techniques require a fine and specific tuning of the solving process (parameters of the simulated annealing, cross-over probabilities in GA, and stopping condition).

Hybrid Approaches. Some parallel works have proposed hybrid approaches through a cooperation of different methods. The novel feature is that a first step computes volumes solely defined on camera positions to build models of the feasible space [3, 28]. And a second step relies on a classical stochastic search as in [28] or heuristic search [3]. This provides an efficient pruning of the search space and retains the search process in *interesting* volumes.

5 Discussion

In order to manage complex 3D scenes, an abstraction of the geometry is necessary: objects are mostly considered as simple primitives (points or bounding volumes such as spheres) and provide imprecise and possibly erroneous results since most complex objects can not be represented with simple bounding volumes. Some accounts do consider the precise geometry for occlusion purposes [18, 28], but due to the computational cost of this process, have to rely on hardware rendering capacities. Improving the quality of abstraction of the objects is a difficult but necessary work that requires proposing an adequate model as well as effective solving mechanisms. Complex situations, such as filming a character through the leaves of a tree, have not yet been addressed.

The expressiveness of the set of properties is mostly related to the application context and the retained solution mechanism. Algebraic, interactive and real-time approaches generally rely on quantitative relations (*e.g.* exact locations or orientations of objects on the screen) [7, 12, 18, 10, 17, 3], while optimisation and constraint-based systems allow for qualitative relations through the use of square or oval frames to constrain the location of objects [25, 19, 20, 11]. In the latter, qualitative relations allow to relax the hardness of the properties. Expressiveness is provided through the use of property softness too. In Drucker's

work [13], the choice between constraints and objective functions is hidden to the user, but usually one has to fix the softness of the constraints through scalar coefficients [5, 3, 25] which can be awkward to set. The question remains as to how one decides on the weights of each property and the aggregation of the weights in a single objective function. Hard constraint-based approaches do not require any user settings [20, 11] and uses constraint relaxation to compute approximate solutions [4].

The solution technique adopted constrains both the geometry abstraction and expressiveness of the approaches. However, compared to algebraic and real-time approaches, constraint and optimisation-based approaches provide a powerful framework to add new constraints or objective functions relative to any specific property. The problem is the computational cost of these mechanisms, although hybrid approaches provide some valuable results as efficient search techniques can be applied in promising areas of the search space.

Long-term research is required into higher levels notions such as perceptual and aesthetic properties. To some extent this requires an adequate cognitive model in order to assist the user in his mental time and space construction of the world. Indeed, some work has adopted common set of editing rules to effectively engage the user [16, 30]. However, the editing choices rely primarily on the nature of actions in the environment rather than on an emotional state of the user. Furthermore, the incorporation of aesthetic properties requires an important collaboration with cognitive psychology and relies on empirical characterization of the nature of composition.

References

1. D. Arijon. *Grammar of the Film Language*. Hastings House Publishers, 1976.
2. W. Bares and B. Kim. Generating Virtual Camera Compositions. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces*, pages 9–12, New York, NY, USA, 2001. ACM Press.
3. W. Bares, S. McDermott, C. Boudreaux, and S. Thainimit. Virtual 3D Camera Composition from Frame Constraints. In *MULTIMEDIA '00: Procs. of the eighth ACM international conference on Multimedia*, pages 177–186. ACM Press, 2000.
4. W. H. Bares, J. P. Gregoire, and J. C. Lester. Realtime Constraint-Based Cinematography for Complex Interactive 3D Worlds. In *Procs of AAAI-98/IAAI-98*, pages 1101–1106, 1998.
5. W. H. Bares and J. C. Lester. Intelligent Multi-Shot Visualization Interfaces for Dynamic 3D Worlds. In *IUI '99: Proceedings of the 4th international conference on Intelligent user interfaces*, pages 119–126, New York, NY, USA, 1999. ACM Press.
6. W. H. Bares, D. W. Rodriguez, L. S. Zettlemoyer, and J. C. Lester. Task-sensitive cinematography interfaces for interactive 3d learning environments. In *Proceedings Fourth International conference on Intelligent User Interfaces*, pages 81–88, 1998.
7. J. Blinn. Where am I? what am I looking at? *IEEE Computer Graphics and Applications*, pages 76–81, July 1988.
8. A. Butz. Animation with CATHI. In *Proceedings of American Association for Artificial Intelligence/IAAI '97*, pages 957–962. AAAI Press, 1997.

9. A. Butz, A. Krüger, and P. Olivier, editors. *Smart Graphics, Third International Symposium, SG 2003, Heidelberg, Germany, July 2-4, 2003, Proceedings*, volume 2733 of *Lecture Notes in Computer Science*. Springer, 2003.
10. D. B. Christianson, S. E. Anderson, L. He, D. H. Salesin, D. S. Weld, and M. F. Cohen. Declarative Camera Control for Automatic Cinematography. In *Proceedings of the American Association for Artificial Intelligence 1996*, pages 148–155, 1996.
11. M. Christie and E. Languéno. A Constraint-Based Approach to Camera Path Planning. In Butz et al. [9], pages 172–181.
12. N. Courty and E. Marchand. Computer animation: A new application for image-based visual servoing. In *In Proceedings of IEEE Int. Conf. on Robotics and Automation, ICRA '2001*, volume 1, pages 223–228, 2001.
13. S. M. Drucker, T. A. Galyean, and D. Zeltzer. Cinema: A System for Procedural Camera Movements. In *SI3D '92: Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 67–70, New York, NY, USA, 1992. ACM Press.
14. S. M. Drucker and D. Zeltzer. Intelligent Camera Control in a Virtual Environment. In *Proceedings of Graphics Interface '94*, pages 190–199, Banff, Alberta, Canada, 1994.
15. S. M. Drucker and D. Zeltzer. Camdroid: A System for Implementing Intelligent Camera Control. In *Symposium on Interactive 3D Graphics*, pages 139–144, 1995.
16. D. A. Friedman and Y. A. Feldman. Knowledge-based cinematography and its applications. In *Proceedings of the 16th European Conference on Artificial Intelligence, ECAI'2004*, pages 256–262. IOS Press, 2004.
17. M. Gleicher and A. Witkin. Through-the-lens camera control. In *Proceedings of ACM SIGGRAPH'92*, pages 331–340, 1992.
18. N. Halper, R. Helbing, and T. Strothotte. A camera engine for computer games: Managing the trade-off between constraint satisfaction and frame coherence. In *In Proceedings of the Eurographics'2001 Conference*, volume 20, pages 174–183, 2001.
19. N. Halper and P. Olivier. CAMPLAN: A Camera Planning Agent. In *Smart Graphics 2000 AAAI Spring Symposium*, pages 92–100, March 2000.
20. F. Jardillier and E. Languéno. Screen-Space Constraints for Camera Movements: the Virtual Cameraman. In N. Ferreira and M. Göbel, editors, *Procs. of EUROGRAPHICS-98*, volume 17, pages 175–186. Blackwell Publishers, 1998. ISSN 1067-7055.
21. S. Katz. *Film Directing Shot by Shot: Visualizing from Concept to Screen*. Michael Wiese Productions, 1991.
22. J. Mascelli. *The Five C's of Cinematography: Motion Picture Filming Techniques*. Cine/Grafic Publications, Hollywood, 1965.
23. S. McDermott, J. Li, and W. Bares. Storyboard Frame Editing for Cinematic Composition. In *IUI '02: Proceedings of the 7th international conference on Intelligent user interfaces*, pages 206–207, New York, NY, USA, 2002. ACM Press.
24. C.F. Nodinem, J.J. Locher, and E.A. Krupinski. *The role of formal art training on perception and aesthetic judgement of art compositions*. Leonardo, 1993.
25. P. Olivier, N. Halper, J. Pickering, and P. Luna. Visual Composition as Optimisation. In *AISB Symposium on AI and Creativity in Entertainment and Visual Art*, pages 22–30, 1999.
26. P. Palamidese. A Camera Motion Metaphor Based on Film Grammar. *Journal of Visualization and Computer Animation*, 7(2):61–78, 1996.
27. C. B. Phillips, N. I. Badler, and J. Granieri. Automatic viewing control for 3d direct manipulation. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 71–74. ACM Press New York, NY, USA, 1992.

28. J. H. Pickering. *Intelligent Camera Planning for Computer Graphics*. PhD thesis, Department of Computer Science, University of York, 2002.
29. J. H. Pickering and P. Olivier. Declarative Camera Planning Roles and Requirements. In *Proceedings of the Third International Symposium on Smart Graphics*, volume 2733 of *Lecture Notes in Computer Science*, pages 182–191. Springer, 2003.
30. B. Tomlinson, B. Blumberg, and D. Nain. Expressive autonomous cinematography for interactive virtual environments. In Carles Sierra, Maria Gini, and Jeffrey S. Rosenschein, editors, *Proceedings of the Fourth International Conference on Autonomous Agents*, pages 317–324, Barcelona, Catalonia, Spain, 2000. ACM Press.
31. C. Ware and S. Osborne. Exploration and virtual camera control in virtual three dimensional environments. In *SI3D '90: Proceedings of the 1990 symposium on Interactive 3D graphics*, pages 175–183, New York, NY, USA, 1990. ACM Press.
32. D. N. Wood, A. Finkelstein, J. F. Hughes, C. E. Thayer, and D. H. Salesin. Multiperspective panoramas for cel animation. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 243–250, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
33. M. Zancanaro, C. Rocchi, and O. Stock. Automatic video composition. In Butz et al. [9], pages 192–201.
34. M. Zancanaro, O. Stock, and I. Alfaro. Using cinematic techniques in a multimedia museum guide. In *Proceedings of Museums and the Web 2003*, March 2003.