Implementing Game Cinematography:
Technical Challenges and Solutions
for Automatic Camera Control in Games

Paolo Burelli - Aalborg University Copenhagen, Denmark

Abstract
Cinematographic games are a rising genre in the computer games industry and an increasing number of titles published include some aspects of cinematography in the gameplay or the storytelling. At present state, camera handling in computer games is managed primarily through custom scripts and animations and there is an inverse relationship between player freedom and cinematographic quality. In this paper, we present a description of a series of technical challenges connected with the development of an automatic camera control library for computer games and we showcase a set of algorithmic and engineering solutions.

Categories and Subject Descriptors (according to ACM CCS):   I.3.6 [Computer Graphics]: Methodology and Techniques—[Interaction techniques] I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search—[Heuristic methods]

1. Introduction

Camera control in computer games is one of the key aspects of player experience [Bur13]. The virtual camera represents the point of view of the player, this it deeply affects its ability to interact and understand the game narrative. Between standard manual and static camera placement, different control paradigms have been developed in the industry, mostly to accommodate specific forms of gameplay [HH09]. Research in automatic camera control, instead, aims at defining general methods to control efficiently an effectively the camera behaviour. Through automatic camera control, game designers and developers can define high-level and environment-independent requirements, such as the visibility of a particular object or the size of that object on the screen. Based on these requirements the camera should be automatically placed and animated while reacting to changes in the virtual environment.

Finding and tracking the optimal camera that maximises the fulfilment of the designer requirements is a complex optimisation problem; the complexity of the objective function evaluation and the short execution time imposed to ensure a real-time execution, make the problem computationally hard. Moreover, in real-time automatic camera control, the algorithm needs to find and track the best possible camera configuration in an environment that changes continuously and in an unpredictable manner.

On top of these purely algorithmic aspects, there are a number of challenges related to the engineering of a camera control system for computer games. Addressing issues, such as how to adapt the system to different types of environments and geometries or how to ensure smooth camera animations, is as important as designing an efficient camera optimisation algorithm. In this article, based on the design and implementation of an automatic camera control library for computer games named CamOn [†], we describe the challenges addressed and the solutions employed. Furthermore, we evaluate these solutions from a performance perspective on benchmark for real-time automatic camera control [BY15].

2. Background

Since the introduction of 3D computer graphics, the automation of virtual camera control represents as a challenging task [CON08]. The task is commonly modelled as a constraint satisfaction problem, in which a designer defines a set of desired frame properties that are modelled as an objective function to be maximised by a solver. These frame properties, often referred to as frame constraints, describe how the frame should look like in terms of composition [BMBT00].

[†] http://github.com/paoloburelli/camon
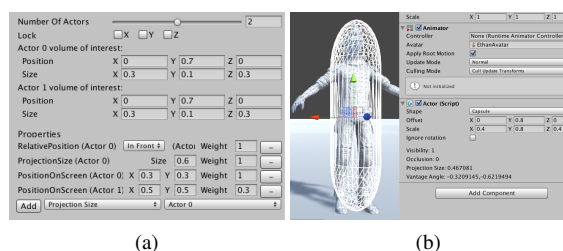
(a)   (b)

Figure 1: Examples of shot (a) and actor (b) definition

Pure optimisation approaches, such as the Smart Viewpoint Computation Library [RU14], implement a flexible search strategy that models all frame constraints as an objective function (a weighted sum of each constraint) allowing for partial satisfaction of any constraint. The flexibility of such approaches comes with the price of a high computational cost, which is a particularly critical factor in real-time dynamic virtual environments such as computer games. A number of solutions [BSG08, BJ09] attempted to address this computational employ by employing local search algorithms.

Inspired by the work of Ranon and Urli [RU14] on improving efficiency of static virtual camera optimisation, this article builds upon these early works on dynamic camera optimisation with local search [BJ09] by introducing a series of technical and algorithmic improvements targeted at increasing the quality of the shots produced.

3. Automatic Camera Control in Games

The problem of automatic camera control in interactive environments can be described as a composition of three interconnected subtasks: *shot definition and planning*, *virtual camera composition* and *camera animation*. Shot definition and planning can be described as the task of defining a sequence of shots to visualise one or more events in a virtual environment. Virtual camera composition is the process of translating these shots in actual camera configurations, while camera animation is the process of animating the camera during the shots and ensuring smoothness between them. In the process of translating this framework in a camera control library, we identified four components: *shot*, *actor*, *solver* and *camera operator*.

Shot A shot, in cinematography, describes an uninterrupted series of frames with common composition properties [Skl93]. We employ the shot concept to describe the requirements for the camera behaviour in terms of how the frame generated by the camera should look like. A shot is composed, therefore, by a number of actors – not necessarily humans – and a description of the composition properties of the frames [BMBT00]. Based on the work by Burelli

and Yannakakis [BY15] and Ranon et al. [RCU10], who performed a systematic analysis of frame constraints for virtual camera composition, the latest version of CamOn supports the following five properties:

- Visibility
- Projection Size
- Vantage Angle
- Relative Position
- Position On Screen

The visibility constraint permits to control whether an actor should be included in the frame and not hidden by any other object; both conditions are necessary to identify the target object as visible. The projection size constraint allows to control the area covered by the projection of an actor in the frame. The vantage angle constraint permits to require the camera to be positioned so that the angle between an actor's front vector and the direction of the camera equals to a specific value. The relative position constraint permits to control the relative placement of two actors in the frame, while the position on screen constraint allows the control of the absolute position on screen of a single actor.

A shot is defined as an environment independent entity that can be created, edited and loaded when necessary. Figure 1a shows an example of the definition of a shot with the number of actors and a list of frame constraints. Furthermore, for each actor, it is possible to define a volume of interest within the actor. This is necessary to focus the shot on a part of the actor allowing, for instance, a close-up shot to focus on the face.

Actor An actor is an entity representing a virtual world object in a shot, its purpose is to give a standard representation of different types of objects that can be used throughout the camera control flow to animate the camera. An actor represents the connected object with a proxy geometry that can be adapted to the object's shape and size (see Figure 1b). The necessities for such an abstraction are manifold. First, the virtual objects might contain animations that should not be taken into consideration by the camera; for instance, a 3D model oscillating during a walking cycle should not, in most cases, transmit its oscillation to the camera. Likewise, the virtual object might be affected by transformations that should not be considered by the camera, such as the rotation of a rolling ball.

Moreover, contrarily to a bounding box or a bounding sphere, the actor's representation is independent from the original geometry and can be adapted to the target object's shape without affecting the internal mechanisms of the game engine – e.g physics. For instance, the representation of a butterfly might not include the wings and, in the same way, the representation of a headless man might include the non-existing head.

Finally, the actor entity serves as a centralised point of evaluation for the actor's composition properties, allowing

to reduce redundant calculations connected to a per-property evaluation [RU14]. This allows, for instance, to reuse the actor's visibility when calculating the actor's projections size.

Camera Operator The camera operator component is the central component connecting shot, actors and solver. It instructs the solver with a specific shot and set of actors and, at every frame, it uses the solver's calculated camera configuration to drive the camera animation. This component allows control of the camera dynamic behaviour and it ensures that the camera is animated smoothly guaranteeing frame coherence. This process is called camera motion damping and it is performed by smoothing the changes of camera velocity [HH09]. The motion damping in CamOn is implemented through a spring-damping method connecting the current camera position with the ideal camera calculated by the solver.

Solver The solver component is responsible for interpreting the given shot, analysing the scene and finding the camera configuration that best matches the desired composition properties. In computer games and interactive applications the solver process is not intended as a finite optimisation that produces an optimal set of results at the end of its execution. It is, instead, a never-ending process that continuously adapts and tracks the best possible camera configuration while the environment is changing. This type of problem process is commonly defined as dynamic optimisation problem (DOP), in which the objective function of this problem is the linear combination of the objective functions corresponding to the frame constraints defined in the shot and the solution space contains all possible camera configurations.

In this article we propose an extension of one of the author's earlier works on dynamic camera control with local search [BJ09, BY10] based on the Artificial Potential Field algorithm. In the process of engineering the CamOn library, we have analysed which aspects of the solver can be improved based on the type of shot and the current state of the environment. We identified the three aspects: initialisation, movement compensation and weight management. These improvements are independent of the type of solver and can be applied to any dynamic optimisation algorithm for virtual camera composition

The improved initialisation is based on the results presented by Ranon and Urli [RU14], which suggest a strategy for particles initialisation to reduce the convergence time of Particle Swarm Optimisation for static camera composition. To adapt this strategy to a local search algorithm we propose to prioritise the frame constraints and to calculate the initial position based on the first vantage angle or relative position frame constraint. Furthermore, in order to void early convergence based on complete occlusion, the initial position is moved close enough to the target to guarantee the presence of no obstacles between the two.
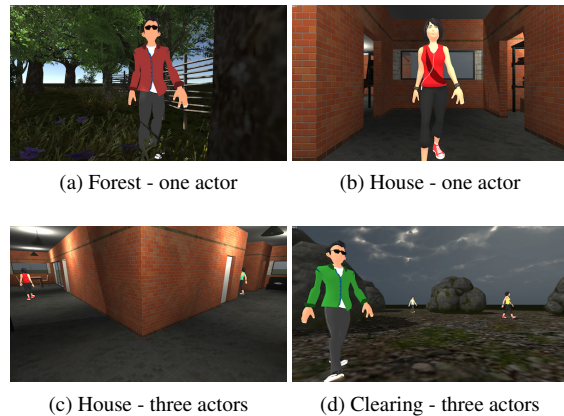
The second improvement, movement compensation, is

(a) Forest - one actor (b) House - one actor

(c) House - three actors (d) Clearing - three actors

Figure 2: *Screenshots taken showing the test scenarios.*

based on Halper's solution to enforce frame coherence and consists in interpolating the trajectory of the actors. In this context the interpolation is used to compensate the position of the last optimum so that the local search algorithm is able to explore areas of the space that are more likely to contain an optimal solution.

The last improvement focuses on dynamically tuning the weights of the frame constraints in the objective function. These weights are normally constant throughout the whole optimisation process; however, this might provoke some undesired behaviours in the local search process. For instance, in the cases in which one or more actors in the shot are completely non visible, the frame constraints corresponding to those subjects are have still an impact on the objective function. This means that the solver could potentially select solutions in which no actor is visible, but the camera is placed at the correct vantage angle for all actors.

To avoid these situations, the weight of the frame constraints referred to an actor can be scaled by a factor inversely proportional to the occlusion of that actor, giving higher propriety to the frame constraints referring to visible actors. Furthermore, using a second or higher power of the occlusion can be used to smoothly reduce the weights only at high levels of occlusion – e.g. when the actor is almost non visible. The same principle principle can be applied also to the function used to estimate the level of visibility of an actor. Using a power of the visibility – the current implementation uses a 4th power – instead of the original estimation allows the solver to handle more effectively sparse occlusion [CNO12]. Modifying in this manner the objective function induces the search algorithm to discard solutions that offer little improvement in visibility at a high cost of the other objective function.

Scenario	Improved	Basic	p-value
a	1.00	0.98	0.00
b	0.99	0.98	0.12
c	0.52	0.42	0.00
d	0.58	0.40	0.00

Table 1: Average shot quality in each scenario.

4. Performance Evaluation

To assess the impact of the improvements proposed in this article, we conducted an experimental evaluation comparing the original version of the APF algorithm for virtual camera composition [BJ09] with a version including these improvements. The experiment was carried using the benchmark application presented in [BY15] on an Intel i7 at 2.3 Ghz with 8GBytes of RAM and with an NVIDIA GeForce GT 650M graphics card running Mac OS X 10.10.

Among the eleven scenarios presented in the aforementioned benchmark, we selected the subset of scenarios that mimic more closely an in-game situation. The selected subset contains four scenarios in which a Projection Size, a Vantage Angle and a Visibility constraints are imposed on one or three actors in different environments as described in Figure 2. In each scenario, the two versions of the system have been compared on 20 seconds runs repeated for 20 times; the performance measure employed is the average quality of the frames produced, which is calculated using the average best function value commonly employed in dynamic optimisation [Sch07].

Table 1 shows the average values of accuracy over the 20 runs; the p-value in each scenario is the result of a two tailed t-test between the two tested versions. In the forest scenario with one actor, a small but significant improvement is achieved; the good results in this environment suggest that the improvements introduced have a positive impact in the system's ability to handle sparse occluders as this scenario is made of trees, grass and wooden fences. While no significant improvement is registered in scenario; in the third and fourth scenario, there is an increase of 23% and 45% in the accuracy. These scenarios include empty spaces divided by large occluders that can potentially generate situations in which only one or two actors can be visible at the same time (see Figures 2c and 2d). The ability of the improved algorithm to dynamically change the weights of the non-visible actors helps the solver in these situations to concentrate on the best solution candidates.

5. Conclusion

The article presented a series of domain specific improvements for automatic camera control in computer games. A performance evaluation was conducted, revealing how the proposed improvements applied to an existing camera control system are able to increase the system's performance.

We believe that the study presented can help practitioners and researchers to have a better understanding of the challenges of automatic camera control. Furthermore, the library developed as a result of this research study can serve as a starting point for future experiment about game cinematography aimed, for instance, at developing a better understanding of the relationship between cinematography, gameplay and player experience.

References

[BJ09] BURELLI P., JHALA A.: Dynamic Artificial Potential Fields for Autonomous Camera Control. In *AAAI Conference On Artificial Intelligence In Interactive Digitale Entertainment Conference* (Palo Alto, 2009), AAAI. 2, 3, 4

[BMBT00] BARES W. H., MCDERMOTT S., BOUDREAUX C., THAINIMIT S.: Virtual 3D camera composition from frame constraints. In *ACM Multimedia* (Marina del Rey, California, USA, 2000), ACM Press, pp. 177–186. 1, 2

[BSG08] BOURNE O., SATTAR A., GOODWIN S.: A Constraint-Based Autonomous 3D Camera System. *Journal of Constraints 13*, 1-2 (2008), 180–205. 2

[Bur13] BURELLI P.: Virtual Cinematography in Games : Investigating the Impact on Player Experience. In *International Conference On The Foundations Of Digital Games* (Chania, Greece, 2013), Society for the Advancement of the Science of Digital Games, pp. 134–141. 1

[BY10] BURELLI P., YANNAKAKIS G. N.: Combining local and global optimisation for virtual camera control. In *IEEE Symposium on Computational Intelligence and Games* (2010), pp. 403–410. 3

[BY15] BURELLI P., YANNAKAKIS G. N.: A Benchmark For Virtual Camera Control in Computer Games. In *Applications of Evolutionary Computation* (2015). 1, 2, 4

[CNO12] CHRISTIE M., NORMAND J.-M., OLIVIER P.: Occlusion-free camera control for multiple targets. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (July 2012), Eurographics Association, pp. 59–64. 3

[CON08] CHRISTIE M., OLIVIER P., NORMAND J. M.: Camera control in computer graphics. In *Computer Graphics Forum* (2008), vol. 27, pp. 2197–2218. 1

[HH09] HAIGH-HUTCHINSON M.: *Real-Time Cameras.* Elsevier, 2009. 1, 3

[RCU10] RANON R., CHRISTIE M., URLI T.: Accurately Measuring the Satisfaction of Visual Properties in Virtual Camera Control. In *International symposium on Smart Graphics* (Banff, Canada, 2010), Springer-Verlag, pp. 91–102. 2

[RU14] RANON R., URLI T.: Improving the Efficiency of Viewpoint Composition. *IEEE Transactions on Visualization and Computer Graphics 2626*, c (2014), 1–1. 2, 3

[Sch07] SCHÖNEMANN L.: Evolution Strategies in Dynamic Environments. In *Evolutionary Computation in Dynamic and Uncertain Environments*, Yang S., Ong Y.-S., Jin Y., (Eds.), vol. 51 of *Studies in Computational Intelligence*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 51–77. 4

[Skl93] SKLAR R.: *Film: An International History of the Medium.* Thames and Hudson, 1993. 2