

Computer Graphics and Constraint Solving: An Application to Virtual Camera Control

Jean-Marie Normand

December 17, 2006

Chapter 1

Computer Graphics and Constraint Solving: An Application to Virtual Camera Control

1.1. Introduction

Modelling, animation and rendering have dominated research in computer graphics yielding increasingly rich and complex virtual worlds. In order to manage the complexity of these new worlds, some applications have successfully relied on the use of *CSPs* (Constraint Satisfaction Problems) and their associated constraint solving techniques. Indeed such techniques seem highly appropriate to solve difficult problems (such as computer graphics applications) thanks to their declarative properties as well as the whole set of methods available that will ensure the efficiency of constraint solving approaches. However, actual constraint based techniques are somehow limited in the interaction offered to the user. This is especially true when considering that the modelling stage actually loses the semantics pertaining to the properties given by the user when transforming them into algebraic constraints. Indeed the constraints obtained by this step are solved without any consideration with respect to their membership to a property or another. Some *semantic* decomposition involving geometric constraints involve a semantic dimension and are able to give explanations in case of failures (for a broad overview of these decompositions see [JER 06]). Nevertheless, this is restricted to geometric constraints and cannot be seen as a very general method. When solving the CSP, information that could be gathered from the knowledge of the violated constraints is lost and nothing is provided to the user regarding the inconsistency of the problem. Main difficulties arise when one cannot determine the nature of

Chapter written by Jean-Marie NORMAND.

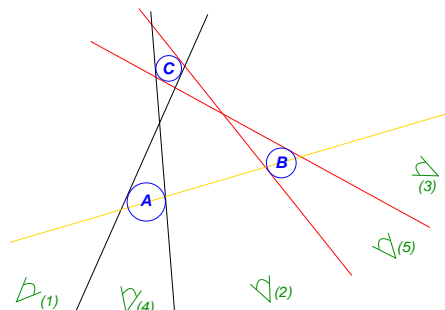


Figure 1.1. Possible distinct areas for viewing a couple of objects *A* and *B* (resp. on the left and right of the screen) w.r.t. to a third object *C*.

a CSP (over, under or well constrained) before solving it, and that the solving methods should be chosen according to this nature. This would also lead to replace the usual measure of similarity between two solutions (usually the Euclidean distance) by a closeness w.r.t. constraints satisfaction which would capture more accurately the difference among solutions.

In this paper, we rely on the problem of virtual camera composition (VCC) that consists in positioning a camera in a virtual world such that the resulting image satisfies a set of visual cinematographic properties, in order to illustrate the integration of semantics in the whole solving process.

The virtual camera composition task is generally achieved through a tedious and time-consuming process requiring a succession of “place the camera” and “check the result” operations. Current 3D modelers surprisingly lack integration of tools to assist the user in this task, despite the fact that cinema, in more than a hundred years, has provided a rich grammar that allows a director to unambiguously describe shots.

The paper is organized as follows, we next present a brief overview of our method before detailing the semantic space partitioning approach w.r.t. to visual properties on the image. Section 1.3 will address the description of our numeric algorithm which is based on a continuous extension of a local search algorithm. The interpretation and reasoning with the semantic volumes is presented in section 1.4. Finally results are presented in section 5 before drawing a conclusion.

1.1.1. Overview

Most of camera positioning methods rely upon optimization processes to compute satisfactory camera placements and therefore lead to a unique solution closely related to the objective function (see [CHR 06] for an overview).

However, the description of a cinematic shot can possibly yield different visual solutions. Therefore, by computing the set of semantically distinct solutions w.r.t. cinematographic properties, one provides the user meaningful results. Figure 1.1 presents a top view of a simple scene containing three objects A , B and C . Whenever the user describes a shot in which he constrains A and B respectively to lay on the left and on the right sides of the screen, it clearly yields three possible classes of camera configurations: (1) object C is on the left of A and B on the screen, (2) object C is between A and B , and (3) object C is on the right of A and B . Moreover, when considering possible occlusions, two classes can be added (4) A occludes C and (5) B occludes C . In such cases, classical optimization and incomplete CSP-based approaches fail in that a unique solution [DRU 95, OLI 99, PIC 02], or a reduced subset [JAR 98, CHR 02] of solutions is proposed, whereas all classes of solutions should be equally considered. These approaches actually lose the semantics of the problem while relying upon pure numerical approaches.

In this paper, we propose to integrate a semantic dimension in the solving and interaction processes to assist the user in his camera placement tasks. We follow a threefold declarative approach: (1) describe the desired solution with a set of cinematographic-based properties, (2) compute distinct classes of solutions satisfying the description with related cinematographic properties and (3) explore and interact with the classes of possible solutions.

In the description phase, as in previous approaches ([OLI 99, DRU 95]), a high-level grammar is offered including composition properties (framing objects on screen surface, relative object orientation and size) and shot properties (close shot, establishing shot, *etc.*). The geometry of the scene (locations and orientation of objects) is considered as an input provided by the user.

In the computational phase, two processes are combined. The first process partitions the search space according to cinematographic properties (*e.g.* area such that A occludes B on the screen) and builds the intersection of the space partitions (indeed the conjunction of the visual properties will correspond to an intersection of the *semantic volumes*). The second process computes a *nice* representative of each possible class of solutions via a continuous domain implementation of a local search meta-heuristic algorithm.

Finally, in a third phase, the user navigates in the possible solution sets and interacts with the semantic information provided in each area. This paper concentrates on the first two phases and offers solid foundations for high-level interactions with the user.

1.2. A Semantic Space Partitioning Approach

Our approach to virtual camera composition (VCC) is based on the primary idea of *visual aspects* [KOE 79]. The idea behind *visual aspects* is to gather all the viewpoints of a single polyhedron that share similar topological characteristics on the image. A change of appearance of the polyhedron with changing viewpoint, gives rise to boundaries in the search space. Computing all the boundaries enables the construction of regions of constant aspect, namely *viewpoint space partitions*.

In this paper, we propose an extension of viewpoint space partitions to multiple objects and replace the topological characteristics of a polyhedron by cinematographic properties such as occlusions, relative viewing angles, distance shots and relative object locations.

We introduce the notion of *semantic volume* as a volume of possible camera locations that give rise to qualitatively equivalent shots w.r.t. to cinematographic properties, *i.e.* semantically equivalent shots. Each volume is characterized by a set of semantic tags issued from film grammar [ARI 76] and each tag is associated to a satisfied property in the volume. The entire space of possible camera locations is thoroughly partitioned for each object, and each couple of objects.

We then derive from the user's description the subset of volumes to be considered and intersect them. This process leads to a set of non-connected regions of which each represents a different class of solutions in terms of visual aspect.

1.2.1. Projection Property

The projection property is based on the notion of scale shots in cinematography. It allows the artist to specify a viewing shot for an object. There are basically six different kinds of shots : the *Extreme Close-Up*, the *Close-Up*, the *Medium Close-Up*, the *Medium Long Shot* (or *Plan Américain*), the *Long Shot*, the *Extreme Long Shot*. Related semantic tags reflect all six kinds of shots (`ExtremeCloseUp(Object)` to `ExtremeLongShot(Object)`).

The underlying *semantic volume* is computed given the position of an object and a cinematographic scale shot specified by the user. One can deduce an optimal size corresponding to each scale shot presented in Figure 1.2. The object's bounding sphere and desired area in the frame are used to determine the range of camera distances. The minimum and maximum bounds of the interval correspond to two distances defining the inner and outer radiuses of an hollow sphere that includes the set of consistent positions for the camera.

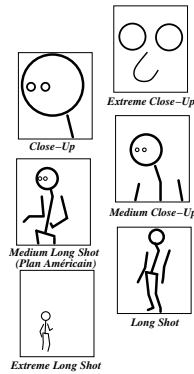


Figure 1.2. The six distances between the camera and a character according to Arijon [ARI 76].

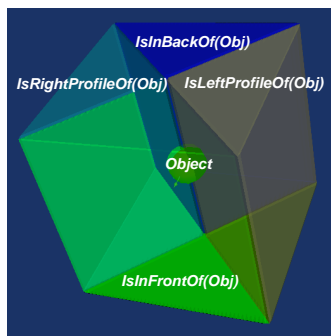


Figure 1.3. Four common relative viewing angles and related semantic tags.

1.2.2. Orientation Property

The orientation property lets the virtual cinematographer specify the viewing angle required to shoot an object or a character. A common set of 8 viewing angles is offered (e.g. relative to object A , there are $IsLeftProfileOf(A)$, $IsRightProfileOf(A)$, ..., $IsThreeQuarterFrontLeft(A)$ up to $IsThreeQuarterBackRight(A)$) and each can be composed with high and low relative angles $IsHighAngle(A)$ and $IsLowAngle(A)$.

Computing orientation *semantic volumes* consists in building a prism-shaped volume of possible camera locations (cf. Fig. 1.3), w.r.t. the vector to consider (front, back, left, etc.).

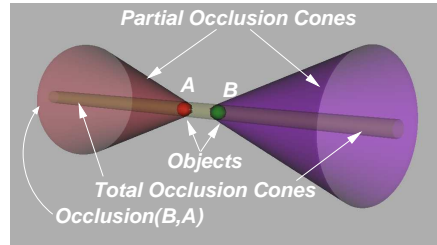


Figure 1.4. Occlusion cones computation (both partial and total occlusions).

1.2.3. Occlusion Property

The occlusion property gives the user the opportunity to specify some visibility constraints between two objects of the scene. The cinematographer can characterize a total, partial or non occlusion of an object by another. The *semantic volumes* induced by an occlusion property are computed given characteristic cones defined with respect to the positions of the two objects involved in the occlusion property. The inside bounds of the cones define the volumes of partial occlusion. The outside bounds of the cones define the volumes where no possible occlusion can occur (*cf.* Fig. 1.4).

1.3. Numerical solving stage

The output of the space partitioning approach consists in a semantic volume $s_v = \langle \mathcal{S}, \mathcal{V} \rangle$ containing possible camera locations. From there, the numerical stage computes a *nice* representative of each disjoint volume in \mathcal{V} . This consists in choosing in \mathcal{V} a consistent camera configuration (location, orientation and focal distance) that maximizes each property p_i corresponding to a semantic tag of \mathcal{S} (*i.e.* minimizing a cost function).

The problem therefore comes down to determine a septet of variables c , such that :

$$\begin{cases} \min \sum_i \text{cost}_{p_i}(c) \\ c \in \mathcal{V} \end{cases}$$

where $\text{cost}_{p_i}(c)$ stands for the function cost associated to property p_i .

Classical optimization and constrained optimization techniques require a differentiable objective function (*e.g.* general gradient descent algorithm). In order to manage our constraints we rely on stochastic-based *Local Search* techniques.

The algorithm relies on the exploration of the search space – here a *semantic volume* $\langle \mathcal{S}, \mathcal{V} \rangle$ – starting from an initial set-up, and exploring the neighborhood around

the current configuration. A neighborhood function randomly selects a number of neighbors in \mathcal{V} , each is evaluated, and the best one is kept as the new current configuration. *Diversification* and intensification techniques are provided to respectively prevent being stuck in local minima and concentrate on promising regions.

1.4. Exploitation of Semantic Volumes

The main contribution of this paper is to offer a semantic basis for exploring and interacting with the volumes. We illustrate two possible interactions: making requests on the computed *semantic volumes* and making requests on the whole 3D scene w.r.t. the computed volumes.

1.4.1. Making requests on the volumes

For a given description, each computed volume provided by the geometric solver stores some knowledge related to the satisfaction of the properties. From here, the characterization of each distinct volume can be semantically augmented according to properties the user has not mentioned. For example, if a semantic volume s_v is characterized by the relative location $\text{IsLeftOf}(A, B)$ tag, some further characterization of s_v can be computed by considering the orientation properties related to A and B (e.g. $\text{IsInFrontOf}(A) \wedge \text{IsInBackOf}(B)$). As a consequence, the user can select two semantically augmented volumes s_{v1} and s_{v2} from the same description and request for the differences between them. This request boils down to compute all tags in s_{v1} and in s_{v2} that do not belong to $s_{v1} \cap s_{v2}$.

1.4.2. Making requests on the scene

As each object and couple of objects in the scene generate their respective *semantic volumes*, it is trivial to make requests on a computed volume s_v against any possible object or property. The request comes down to computing a new geometric intersection and checking the number of connected volumes computed by tessellation.

1.5. Results

Two examples illustrate our approach. First the classical over-the-shoulder shot implying two characters and then a framing shot with 7 possible classes of solutions.

The classical over-the-shoulder shot is commonly encountered when filming a dialogue between two or more actors and consists in laying the camera behind one actor while framing the other. Figure 1.5 illustrates the related *semantic volume* and a result is presented in Figure 1.6.

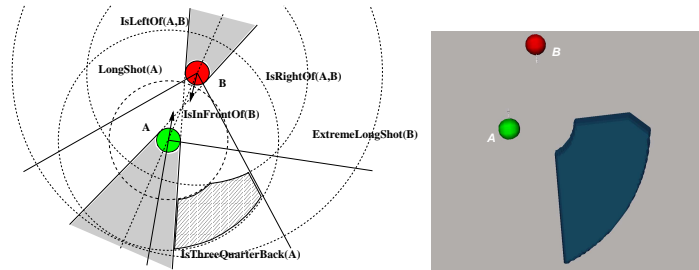


Figure 1.5. Top view of the search space computed by the over-the-shoulder shot (left) and tessellation of the implicit volume (right).



Figure 1.6. A result of the over-the-shoulder shot.

In the next problem, the geometry of the scene is composed of 5 coplanar objects (see top-view Fig. 1.7). The user frames objects A , B and C respectively in the left, middle and right of the screen, and constrains D and E to belong to the screen without any occlusion. All three shots satisfy the users description and illustrate different classes of solutions.

Table 1.1 presents the time spent during the geometrical (T_G) and numerical (T_N) steps in computation of one representative of each semantic volume. T_G is directly related to the granularity of the tessellation (examples 1 and 2 generate respectively 240 and 4036 polygons). T_N is the time spent in the local search with `nbSteps` = 25 and `nbTries` = 25. Such values provide a very good time-to-quality ratio. Although the total time to compute all representatives seems important ($7 \times 0.67 = 4.69$ seconds), time per representative is around 0.7 seconds which is quite acceptable for interaction purposes.

1.6. Discussion

The semantic space-partitioning approach offers the following features: cinematographic properties provide semantic volumes containing the possible solutions of the problem through a geometric process that filters non-satisfactory areas of the world.

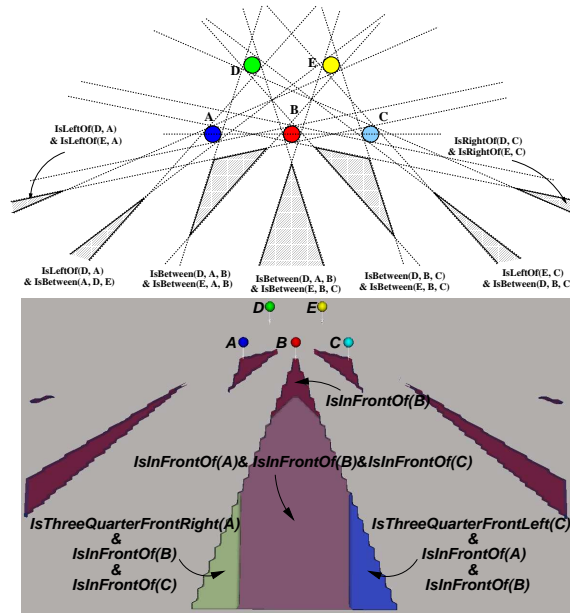


Figure 1.7. Top view of the five-framing shot (top) and tessellation related to possible camera locations (bottom) with further semantic information related to orientations of A, B and C.

Example	Nb Vol	T_G	T_N	Total time
1	1	0.21	0.60	0.81
2	7	0.53	0.67	5.22
				$(0.53+7 \times 0.67)$

Table 1.1. Time for computing a representative of each volume. T_G and T_N represents the time spend in the geometrical and numerical processes (Time in seconds on Linux OS, Pentium M 2GHz, 512Mo)

Whenever the intersection process leads to an empty result, there is a guarantee of contradiction in the user’s specification. The numerical process offers a *good* representative of each volume at relatively low computational cost.

The main interest of maintaining semantics throughout the whole solving process (as shown is the case of positioning a virtual camera), is that the user gains meaningful and interesting information on the solutions of his problem. Unlike other approaches we offer solutions as well as additional knowledge on the different classes of solutions (if existing). This allows the user to reason on the problem instead of “only” being

given a solution. Additional information could also be provided on constraints the user had not modelled (*cf.* section 1.4).

To conclude, in this paper we have presented an original approach to virtual camera composition that identifies classes of distinct solutions, provides means to characterize them and computes good representatives. By extending the notion of visual aspects, we introduced the notion of *Semantic Volumes* as a set of possible camera locations that share a same set of cinematographic characteristics. Experimental results show the suitability of our approach and opens exciting perspectives in providing natural and intuitive interfaces to virtual camera composition. This work also lays the groundwork to a new kind of solving processes that maintain semantics of the problem while solving it in order to provide the user with useful information on the solutions (or pseudo-solutions).

1.7. Bibliography

- [ARI 76] ARIJON D., *Grammar of the Film Language*, Silman-James Press, 1976.
- [CHR 02] CHRISTIE M., LANGUÉNOU E., GRANVILLIERS L., “Modeling Camera Control with Constrained Hypertubes”, *Procs of CP 2002*, September 9-13 2002.
- [CHR 06] CHRISTIE M., OLIVIER P., “Camera Control”, *Eurographics 2006, State of the Art Reports*, September 2006.
- [DRU 95] DRUCKER S. M., ZELTZER D., “CamDroid: a system for implementing intelligent camera control”, *Procs of the 1995 Symposium on Interactive 3D graphics*, p. 139–144, 1995.
- [JAR 98] JARDILLIER F., LANGUÉNOU E., “Screen-Space Constraints for Camera Movements: the Virtual Cameraman”, *Procs. of EUROGRAPHICS-98*, vol. 17, p. 175–186, 1998.
- [JER 06] JERMANN C., TROMBETTONI G., NEVEU B., MATHIS P., “Decomposition of Geometric Constraint Systems: a Survey”, *International Journal of Computational Geometry and Applications (IJCGA)*, vol. to appear, 2006.
- [KOE 79] KOENDERINK J., VAN DOORN J., “The internal representation of solid shape with respect to vision”, *Biological Cybernetics*, vol. 32, p. 211–216, 1979.
- [OLI 99] OLIVIER P., HALPER N., PICKERING J., LUNA P., “Visual Composition as Optimisation”, *AISB Symposium on AI and Creativity in Entertainment and Visual Art*, p. 22–30, 1999.
- [PIC 02] PICKERING J. H., *Intelligent Camera Planning for Computer Graphics*, PhD thesis, Department of Computer Science, University of York, September 2002.