Visual Composition as Optimisation

Patrick Olivier; Nicolas Halper; Jon Pickering; Pamela Luna
Department of Computer Science, University of York, Heslington, York, YO10 5DD
patrick.olivier@cs.york.ac.uk

Abstract

Camera planning is the problem of positioning a camera within a world, such that the resulting image has some predefined set of visual properties. We are developing a graphical presentation planning system which incorporates CAMPLAN, a camera planning subsystem for polygonal graphics. CAMPLAN uses a genetic algorithm to optimise the camera with respect to a set of image objectives (properties of the image). The motivations for CAMPLAN are outlined and an informal evaluation of the system is presented in which we show how successfully more restrictive objectives can impose stylistic regularity over similar graphical scenes. We conclude with a discussion of our current work and outline future directions for graphical presentation planning.

1 Introduction

In computer graphics a virtual camera provides the means of specifying views of its world in a manner simulating that of a real camera. The virtual camera is considered to be an object with its own co-ordinates, orientation and field of view. These parameters determine the view associated with the camera and through geometric projection, and the various operations of the rendering pipeline, the idealised operation of a real camera can be simulated.

For the non-expert, placing the camera so as to produce a desired view of a scene is a difficult task. Furthermore there are a number of domains for which it would be advantageous to be able to automatically place a camera. For example, providing multimedia support for diagnosis and maintenance of engineering artefacts (e.g. communications equipment or automobile engines) requires the generation of depictions of components for every potential diagnostic or maintenance scenario. Similarly, in the entertainment industry, there is a need for the automatic generation of graphics in applications such as action summarises for multiplayer games. Even in their simplest form, both of these tasks, if performed manually, place significant demands on a graphic designer. Furthermore, if we require such presentations to be sensitive user preferences, experience and expertise, manual solutions are infeasible.

This paper describes CAMPLAN, the camera planning component of a graphical presentation planning system that we are currently developing. Section 2 comprises a sketch of existing camera planning systems. In sections 3 and 4 we describe and evaluate CAMPLAN. Finally, in section 5 we point to current and future directions both for the development of CAMPLAN and graphical presentation planning in general.

2 Graphical presentation planning

The camera planning problem resides within the latter two stages of the four stage graphical presentation planning pipeline characterised by Doree Seligmann (Seligmann, 1993):

(a) Generation of communicative goal: decide what it is that the image should accomplish. For example, it may be to explain something to the user, to convey the value of some property or to get the user to perform some task.

(b) Selection of presentation strategy: determine the visual effect that will be used to satisfy the communicative goal. That is, the types of visual cues and effect that will be used in a picture, for example, emphasise an object in a scene (without specifying how the emphasis will be achieved).

(c) Selection of presentation method: given the presentation strategy specify the different ways by which it may be realised. For example, emphasis may be achieved by using an extra light source to increase the brightness of an object, or by requiring the scene element concerned to have a prominent size and/or location in the image.

(d) Image generation: based upon the selection of the presentation methods, the graphical model of the artefact must be modified to achieve the specified visual properties. This process will include optimising the camera position to achieve visual properties, but will also involve selecting material and lighting model parameters, and potentially even configuring the spatial arrangement of the scene elements.

There are a number of existing techniques for camera planning, although it is characteristic of such methods that they are highly dependent on the specific application contexts that their originators were concerned with. In Seligmann's IBIS system (Seligmann, 1993)
default camera positions were used, specified relative to the objects of interest. Blinn described how vector algebra can be used to position a camera given the desired position of two objects in the viewplane (Blinn, 1993), and this method was used in the compiler for the Declarative Camera Control Language (DCCL) which allowed the specification cinematic idioms in camera planning for animations (Christianson et al, 1996). The limitation of this approach is that it uses point abstractions of the objects, and cannot therefore can account for the range of visual effects that arise from the fact that real scene elements have finite extents (e.g. occlusion between scene elements).

More recently, Bares and Lester have developed ConstraintCAM, a real-time camera visualisation interface for dynamic 3D worlds (Bares et al, 1998) (Bares & Lester, 1999). ConstraintCAM allows the specification and real-time solution of three classes of constraint: viewing angle, viewing distance and occlusion avoidance. Although the expressiveness of this set is limited (e.g. it is not possible to locate objects at particular positions in the image) the strength of the system is the utilisation of solution techniques that allow real-time satisfaction of constraints sets.

The camera planning component of Steven Drucker's CINEMA system (Drucker, 1994) is the principal motivation for CAMPLAN. Drucker formulated the task of finding an image with a particular set of scene element position properties as a constrained optimisation problem whose solution was sought numerically using Newton's method. The range of properties that Drucker allowed was limited to the positions of scene elements (in fact, only point idealisations of scene elements) in the image, and relative orientations between scene elements and the camera. Due to the local nature of such numerical methods we have empirically demonstrated that this approach is less effective as a solution method for more complex scenes and large sets of properties.

Lastly, (Jardillier & Languénou, 1998) have reported an approach which uses interval methods to find camera paths which yield sequences of images fulfilling temporally indexed image properties. Although the classes of properties is once again very limited, and the method computationally expensive, the interval-based approach has the advantage of guaranteeing the maintenance of visual properties for the duration of their temporal indexing. This contrasts favourably with techniques which rely on only a sample of the positions along a camera's path.

CAMPLAN is an attempt to address the principal shortcomings of all these approaches: the restriction placed on the range of image properties that may be specified and the unrealistic point-based characterisations of scene elements.

3 The CAMPLAN system

We are developing a presentational graphics system, along the lines of Seligmann's four stage architecture, which will incorporate CAMPLAN as the camera planning subsystem. CAMPLAN extends ideas from (Drucker, 1994) and imposes a division of camera planning into three sub-problems:

- **specification of shot objectives**: shots are specified not only in terms of explicit spatial relationships between the camera and scene elements, but also in terms of the objectives (visual and spatial properties) of the desired image;
- **evaluation of objectives**: for any position of the camera, each objective must be well defined and efficient to evaluate with respect to the underlying graphical modelling paradigm (i.e. geometric abstraction of the graphical model of the scene should be resisted);
- **acquisition of a camera state**: there must be a mechanism by which the camera state (location, orientation and field of view) can be established such that the fulfillment of the specified objectives is maximised.

In the following subsections we describe the requirements and operation of CAMPLAN through briefs account of the resolution of each of these problems. Readers interested in a full characterisation of the implementation of CAMPLAN are referred to (Halper, 1999).

3.1 The specification of shot properties

Shot properties can be explicit spatial relationships between objects in the scene and the camera or properties evaluated over the image itself. Examples of the former include the requirement that an particular object is facing the camera or that an object is in front of the camera's near-clipping plane (although not necessarily "in shot"). However, the use of such explicit spatial relationships between objects in the scene and the camera conflicts with the declarative approach to camera planning that CAMPLAN aims to implement. The use of such properties is in practice restricted to requiring scene elements to reside between the camera's near and far clipping planes.\(^1\)

Image properties (objectives) are properties of the projections of scene elements into the image plane, or rela-

\(^1\) This is required as most of the image properties discussed remain well defined even when objects are behind the camera. For example, using the standard projection transforms, without clipping, all objects will be projected onto the view-plane whether they are in front of or behind the camera.
tions between properties of the projected scene elements. Table 1 lists a examples of pairs of objectives implemented by the current version of CamPlan. The integer following the objective name indicates the number of arguments involved in the specification of the property. The objectives HorizSize/3 and VertSize/3 distinguish between the required vertical and horizontal extents of an object, and their 3 arguments identify (1) the scene element for which the property holds; (2) the magnitude of the extent of the projection, in screen co-ordinates; and (3) a tolerance.

The objectives BetweenX/3 and BetweenY/3 require the screen coordinates of a scene element to be bounded by specified maxima and minima. More complex quantitative and qualitative relationships between the projected extents of two scene elements may be specified using the RelObjectsPosition objectives, including all of the thirteen qualitative relations that can hold between two one dimensional intervals (Allen, 1983) and quantitative variations on these parameterised by the magnitudes of the extents of the objects.2

3.2 Evaluation of shot properties

The means of evaluating objectives is dependant on the underlying graphical model of the scene. For example, if all the scene elements are spheres then we can evaluate most of the properties in table 1 very efficiently via closed form mathematical expressions. Consider the NotOccludedBy objective which specifies that one scene element is not occluded by another.

Suppose object1 and object2 have the camera co-ordinates (x1, y1, z1) and (x2, y2, z2). In considering the NotOccludedBy property, for object2 to occlude object1, it must be between the camera and object1. Thus, assuming object1 is in front of the camera (z1>0), if either of the constraints: z2<0 or |d1|<|d2|, where |d1| = √(x1² + y1² + z1²), then object2 cannot occlude object1. If neither of the constraints are satisfied, then consider figure 1. The constraint: θ1 + θ2 < θ is satisfied where: θ = cos⁻¹((d1.d2)/(|d1||d2|)) and θ1 = sin⁻¹(|r1||d2|), θ2 = sin⁻¹(|r2||d1|).

Table 1. Image objectives

<table>
<thead>
<tr>
<th>HorizSize/3</th>
<th>VertSize/3</th>
<th>PositionX/3</th>
<th>PositionY/3</th>
<th>PositionXY/4</th>
<th>NotOccludedBy/2</th>
<th>BetweenX/3</th>
<th>BetweenY/3</th>
<th>BetweenObjectsX/3</th>
<th>BetweenObjectsY/3</th>
<th>RelObjectsPositionX/5</th>
<th>RelObjectsPositionY/5</th>
<th>InViewport/3</th>
<th>EntirelyInView/1</th>
<th>SizeOnViewport/3</th>
<th>SizeOnView/4</th>
<th>RelSizeOnViewport/4</th>
<th>RelSizeOnView/4</th>
<th>OccludedInView/3</th>
<th>OccludedOnView/3</th>
</tr>
</thead>
</table>

In practice, CamPlan implements more computationally expensive evaluation methods over polygonal representations of scene elements. Furthermore, the models themselves allow the specification of the part-whole structure of each scene element which enables the specification of scene properties over named parts of scene elements. For polygonal models occlusion constraints are evaluated in a two stage manner, first over a bounding sphere approximation of the polygonal object (as above), followed by an adaptive scanline visible

---

2 Although we do not develop this point further, we believe that qualitative relations between image elements constitute a more natural way to specify properties in an image.
surface algorithm for which the resolution is dynamically assigned depending on the tolerance of the specified occlusion property. For a more complete account see (Halper, 1999).

3.3 Acquisition of a camera state

Assuming the evaluation methods outlined in the previous section, the process of acquiring a camera location and orientation, that yields an image maximising the fulfilment of the objectives, can readily be cast as an optimisation problem. CAMPLAN uses a genetic algorithm (a non-deterministic optimisation method) to find the optimal camera position. All seven elements of the camera state vector (position/3, angles/3 and field of view) are encoded in the chromosome and the configuration of the genetic algorithm is as follows: the population is randomly initialised within a space specified in the input; selection is by binary tournament; crossover is two point; mutation modifies a gene by a small random perturbation from its current value; and only 10% of the population is replaced in each generation.

The fitness function is a linear combination of normalised values corresponding to the degree of fulfilment of the objectives described in sections 3.1 and 3.2. The results described in the next section have been obtained without any experimentation with the form of the fitness function or any other parameters of the genetic algorithm, other than using a population size and number of generations large enough to give rise to an optimum camera state for which the image maximally fulfils the specified objectives.

4 Examples

Human photographers and cinematographers will casually satisfy communicative goals such as "locate the door" by applying suitable composition styles or cinematic idioms. However, our approach to camera planning depends on the specified image properties defining sets of images that satisfy the attendant communicative goals. Since the images produced by CAMPLAN are intended to function in the same way as human-composed photographs the appearance of styles when similar communicative goals are applied to similar scenes may be anticipated. Hence we attempted to evaluation of the ability of the system to generate recognisable styles. This was done by observing CAMPLAN’s satisfaction of progressively more restrictive sets of image properties for the appearance of common styles.

The same sets of properties are applied to each scene in turn and the results of five random runs are given. Isomorphism between the elements of the different scenes are characterised as sets: $A=\{\text{lamp, trash, cake}\}$, $B=\{\text{table, house, table}\}$, $C=\{\text{chair, rclown, pine01}\}$, $D=\{\text{plant, yclown, pine02}\}$, $E=\{\text{cabinet, bclown, pine03}\}$. The same objectives are applied across the scenes for isomorphic elements. For example, for every image in which $\text{pine01}$ is involved in the specification of a property, there are images in which $\text{yclown}$ and $\text{plant}$ are identically involved.
4.1 Objective: ** EntirelyInViewport**

Figure 3 shows the set of solutions for each scene using the set of objectives given below. These simply specify that all the scene elements must lie in the viewport:

- **EntirelyInViewport** objectA
- **EntirelyInViewport** objectB
- **EntirelyInViewport** objectC
- **EntirelyInViewport** objectD
- **EntirelyInViewport** objectE

Although the projections of all five objects are required to be entirely within the viewport, the location of the camera may take any value in the range \((-1000, 0..500, \pm 1000)\). Since the region of space within which camera location will yield an image where the scene elements are of a reasonable size is small compared to the size of the total space, it is predictable that the resulting images comprise views in which the scene elements occur in the distance.

4.2 Objective: ** HorizSize**

The undesirably small size of the scene elements in the figure 3 can be addressed by constraining the screen size of a particular element (e.g. the largest) such that to have particular dimensions in screen space. In **CamPlan** the height and width of the viewport is two units, and so the additional constraint in the new set given below requires object B (the tables and the house) to be between 40% and 60% of the screen width.

```
EntirelyInViewport objectA
EntirelyInViewport objectB
EntirelyInViewport objectC
EntirelyInViewport objectD
EntirelyInViewport objectE
HorizSize objectB 1.0 0.2
```

Figure 4 shows example solutions for each of the scenes after the addition of this objective. Although the camera positions are closer to the scene elements than before, there remains a significant variability both in the direction from which the scene is viewed, and in the positions of the scene elements in the viewport.³

4.3 Objective: **ObjectCloserThan**

As with all the previous image objectives, when we place a restriction on the direction from which an object is viewed, it must be as independent as possible of implicit knowledge about the scene.

```
EntirelyInViewport objectA
EntirelyInViewport objectB
EntirelyInViewport objectC
EntirelyInViewport objectD
EntirelyInViewport objectE
HorizSize objectB 1.2 0.2
ObjectCloserThan objectA objectC
ObjectCloserThan objectA objectD
```

³ In the furniture scene (top row), the distance of the cabinet behind the table, and the requirement that all objects are in the viewport, results in the fact that views fulfilling this objective having to be shots from in front of the table.
Thus, rather than referencing orientation information of any particular object, we restrict the orientation of the view by requiring one of the objects to be closer to the camera than two others. The resulting images are shown in figure 5. A degree of compositional regularity has resulted, although there is still some variation in the position of the elements in the viewport.

The addition of this objective relies on a number of assumptions including the fact that the closer object is not significantly larger than its separation from the background objects, and that the three objects (in this case objects A, C and D) are not collinear.
4.4 Objective: **PositionY**

Restricting the position of the scene elements in the viewport may be achieved by placing a restriction on the image coordinates of one of the scene elements.

\[
\text{EntirelyInViewport objectA} \\
\text{EntirelyInViewport objectB} \\
\text{EntirelyInViewport objectC} \\
\text{EntirelyInViewport objectD} \\
\text{HorizSize objectB 1.0 0.2} \\
\text{ObjectCloserThan objectA objectC} \\
\text{ObjectCloserThan objectA objectD} \\
\text{PositionY objectB 0.6 0.3}
\]

In this case the centre of the bounding sphere for object B, (the object with the size restriction) is required to lie in the bottom 15-45% of the image. The resulting images are shown in figure 6 and we can observe a further increase in the compositional consistency. However, it is apparent, from the party scene (middle row) and the house scene (bottom row), that objects may fully, or nearly fully, occlude each other.

4.5 Objective: **OccludedInViewport**

The final objective is the removal of the possibility that any of the scene elements can be either fully, or nearly fully, occluded by another scene element. Thus we require all of the objects to be at least 20% unoccluded. Examples of the images resulting from the addition of these final objectives are shown in figure 7.

\[
\text{EntirelyInViewport objectA} \\
\text{EntirelyInViewport objectB} \\
\text{EntirelyInViewport objectC} \\
\text{EntirelyInViewport objectD} \\
\text{HorizSize objectB 1.0 0.2} \\
\text{ObjectCloserThan objectA objectC} \\
\text{ObjectCloserThan objectA objectD} \\
\text{PositionY objectB 0.6 0.3} \\
\text{OccludedInViewport objectA 0.0 80.0} \\
\text{OccludedInViewport objectB 0.0 80.0} \\
\text{OccludedInViewport objectC 0.0 80.0} \\
\text{OccludedInViewport objectD 0.0 80.0} \\
\text{OccludedInViewport objectE 0.0 80.0}
\]

5 A research program

In the following subsections we outline a number of additional aspects of camera planning and presentation planning that we are either currently undertaking or intend to address in the near future.

5.1 Dynamic camera planning

In the preceding sections we have described the application CAMPLAN to planning static shots of scenes. In fact, we have completed preliminary work in extending our framework to the planning of camera paths for static and dynamic scenes. By requiring the path of a camera to be quadratic, between known start- and end-points (established using the static version of CAMPLAN), a camera path can be found by optimising a control point to maximise temporally indexed visual properties specified by a user. Samples of camera posi-
tions along the path are evaluated, and the cumulative fitness of the path is estimated as a linear combination of the fitness at the sampled positions. In figure 8, the cones indicate the position and orientation of the virtual camera. In figure 8(b) the small sphere moves to the left and a dynamic shot is required that maintains the full visibility and central image position of the moving sphere. In figure 8(a) the focus of the scene shifts from one outside sphere to the other, whilst at the same maintaining the visibility of the central sphere. Visual properties may persist over the whole path, or be temporally indexed. The current system is limited to spheroid approximations of the scene elements and an immediate goal is the efficient extension of this to polygonal worlds.

5.2 Evaluation and design of objectives

The evaluation of section 4 is very limited in its scope. Each scene has the same number of objects, three of which are identical or similar in their spatial characteristics, and one which is significantly larger than the others. Although within these restrictions it is apparent that some degree of stylistic regularity can be imposed, an immediate requirement for research into camera planning is the development of a both a systematic evaluation framework, and a means of eliciting image objectives. As with existing work (Christianson et al, 1996), one potentially fruitful source of image objectives are the accounts of cinematic practice (Arijon, 1976).

5.3 Lighting and material properties

Camera position is just one of the many determiners of the appearance of a graphical scene. Of the remaining aspects, the position and properties of light sources and the material properties of scene elements are very significant. We intend to investigate the application of analogous optimisation techniques to these additional variables. This will in turn require the development of evaluators for properties relating to the illumination of objects, for example, the desired distribution of light on a curved surface, or the contrast between disjoint surfaces that overlap in the image.

5.4 Visual perception and aesthetics

Another source of inspiration for the design of image objectives is the visual cognition literature, and we envisage that sets of cognitively motivated constraints will be useful in tuning results initially derived from objectives motivated by insights from graphic design. For example, cognitive theories of recognisability (Biederman, 1987), depth perception (Rolland, 1995), and figure-ground separation (Koffka, 1935) offer many insights into how the potential for visual ambiguity can be minimised. We also intend to investigate the feasibility of an algorithmic characterisation of concepts from graphic design (Lauer, 1975) and visual aesthetics (Bethers, 1964), for example, unity, emphasis, balance, contrast, pattern, movement and rhythm.

Acknowledgements

We would very much like to acknowledge the efforts of Nick Olivier and Chris McGuire for developing original (spheroid) version of the CAMPLAN software as part of their Masters dissertations at the University of Wales, Aberystwyth. Financial assistance for the continuation of this work is being provided by the University of York, through the Innovative Research and Priming Fund.

References


