Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2012)
P. Kry and J. Lee (Editors)

Efficient Composition for Virtual Camera Control

Christophe Lino, Marc Christie

IRISA / INRIA Rennes - Bretagne Atlantique, France

Abstract

Automatically positioning a virtual camera in a 3D environment given the specification of visual properties to be satisfied (on-screen layout of subjects, vantage angles, visibility) is a complex and challenging problem. Most approaches tackle the problem by expressing visual properties as constraints or functions to optimize, and rely on computationally expensive search techniques to explore the solution space. We show here how to express and solve the exact on-screen positioning of two or three subjects using a simple and very efficient technique. We express the solution space for each couple of subjects as a 2D manifold surface. We demonstrate how to use this manifold surface to solve Blinn's spacecraft problem with a straightforward algebraic approach. We extend the solution to three subjects and we show how to cast the complex 6D optimization problem tackled by most contributions in the field in a simple 2D optimization on the manifold surface by pruning large portions of the search space. The result is a robust and very efficient technique which finds a wide range of applications in virtual camera control and more generally in computer graphics.

Categories and Subject Descriptors (according to ACM CCS): Mathematics of Computing [G.1.6]: Numerical Analysis—Optimization Computer Graphics [I.3.6]: Methodology and Techniques—

1. Introduction

A large range of computer graphics applications rely on the computation of viewpoints over 3D scenes that need to display a number of characteristic composition properties (e.g. on-screen positioning, visibility, or vantage angle of one, two or more subjects). Simplified versions of such problems can be tackled with straightforward vector algebra approaches [Bli88]. More expressive versions are generally casted as non-linear optimization problems in a 7 degree-of-freedom search space, i.e. computing the camera position, orientation and field of view, given a number of composition properties expressed and aggregated in a single viewpoint quality function. Given the size of the search space and the computational cost in the evaluation of composition properties (typically the visibility of subjects), this optimization process is a time-consuming task and hampers the use of evolved composition techniques in most applications.

An analysis of the problem and current solutions reveals a central issue. Aggregating quality functions for all properties together reduces the capacities to guide the solving process through the search space, therefore leading to techniques which explore large areas of the search space without solutions. One transforms a very specific problem into a general search process for which it is difficult to propose efficient and general heuristics.

In this paper, we address an important visual property in virtual camera composition: the specification of exact on-screen coordinates at which targets should project. We show that the range of solutions for the exact composition of two subjects on the screen is a 2D manifold shaped as a spindle torus and that the manifold can be explored using two meaningful parameters φ and θ (see Figure 3). We use this formulation to easily solve Blinn's spacecraft problem [Bli88], and show how to generalize the approach to three and more subjects. The key contributions of our paper are:

- a novel way to express, as a 2D manifold, the solution space of the exact on-screen composition of two subjects;
- an algebraic formulation of on-screen composition for two subjects which solves Blinn's problem [Bli88] without resorting to an iterative technique;
- the expression of virtual camera composition as a search in a 2D space rather than in a 6D space (as performed in most of the literature) for two, three or more targets.

The paper is organized as follows. After presenting the

DOI: 10.2312/SCA/SCA12/065-070

state of the art in virtual camera composition, we show how the on-screen positioning of two subjects can be expressed in a 2D manifold. We then show how Blinn's spacecraft problem can be solved by using this manifold, and how to extend to three (or more) subjects, before concluding.

2. Related work

In 1988, Blinn [Bli88] tried to solve the problem of positioning and orienting a virtual camera to satisfy a given on-screen projection of two subjects with a camera at a distance d of one of the targets, and with some axis (passing through one of the targets) appearing vertical on the screen. By assuming the up vector of the camera was pointing "up" (as much as possible) and using the inverse projection matrix to extract the camera position and orientation from the on-screen projections of subjects, he proposed an iterative algorithm that computes an approximate solution (if one exists) of the problem. To the best of our knowledge, there is no general technique to compute the exact solution (or range of solutions) to this on-screen composition problem.

As a replacement to limitations in Blinn's techniques, researchers have focused on ways to express and solve more complex on-screen compositions. Given the complexity of the problem and the non-linearities in the projection equations, most approaches rely on constraint-based or constrained-optimization techniques in which on-screen properties are expressed as constraints/quality-functions on the camera parameters (see [CON08] for an overview of techniques). Approaches vary in their expressiveness (range of on-screen properties they handle) and efficiency (solving technique employed to search for solutions).

A good illustration is the CamPlan tool [HO00] which uses a genetic algorithm to optimize the camera *w.r.t.* the specified set of visual properties. By encoding the camera parameters as a gene, the authors provide a robust, yet computationally expensive, global search technique. In a similar way, Bares et al. [BMBT00] defined sophisticated camera compositions by creating storyboard frames, indicating how a desired shot should appear and used a heuristic-based optimization technique to compute solutions through a recursive sampling of the search space. Similar approaches in the idea have been followed by [CN05, ACOYL08, BY10, LCL*10]. Authors design one quality function per on-screen property (see [RCU10] which details and compares these properties). The overall quality of a camera configuration is then expressed as a function aggregating the quality of each property and solving techniques explore large regions of the search space leading to significant computationnal costs.

By focusing on the problem of composing two subjects on the screen, our objective is to show that large regions of space can be discarded from the search process. For this purpose, we illustrate the quality of the on-screen positioning of two subjects as a 2D heat map. Figure 1 presents a 2D topview of a scene comprizing two targets. Regions around the subjects are colored from blue to green and red, respectively representing low, average and high viewpoint quality. Note how the region with highest quality is restricted to a small continuous portion of the whole search space.

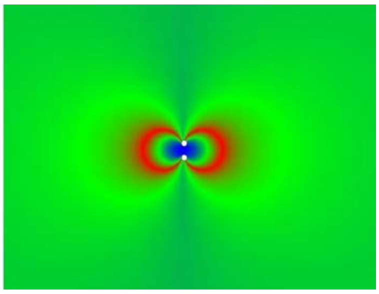

Figure 1: *Heatmap representing the quality of on-screen composition for two subjects (white points) for a region of camera configurations (topview of a the 3D scene). Each colored point represents a camera configuration. Red areas are regions of good quality viewpoints (*i.e. *good compositions) while blue areas are regions of bad quality viewpoints. Green regions represent average quality viewpoints. Note that the best viewpoints are very local to a given curve around the subjects.*

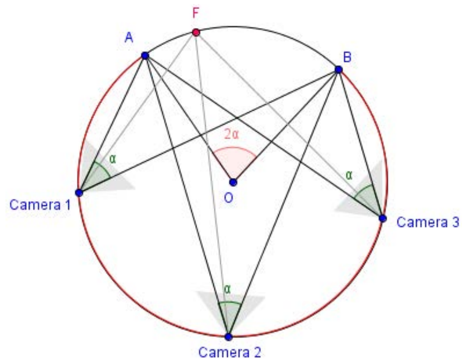

Figure 2: *The range of solutions for the exact composition of two subjects A and B on the screen is displayed in red. The set of points satisfying a given angle α is exactly the arc $\widehat{AB}$ of the inscribed circle centered on O. Moreover* $(\widehat{\overrightarrow{OB},\overrightarrow{OA}}) = 2\alpha$.

In the following section we show that the entire solution set for the exact on-screen positioning problem can be determined with simple mathematical concepts, and that this model serves as a cornerstone for the resolution of more complex camera composition problems.

Definition 1 The oriented angle between two vectors $\vec{u}$ and $\vec{v}$ is denoted as $(\widehat{\vec{u},\vec{v}})$.

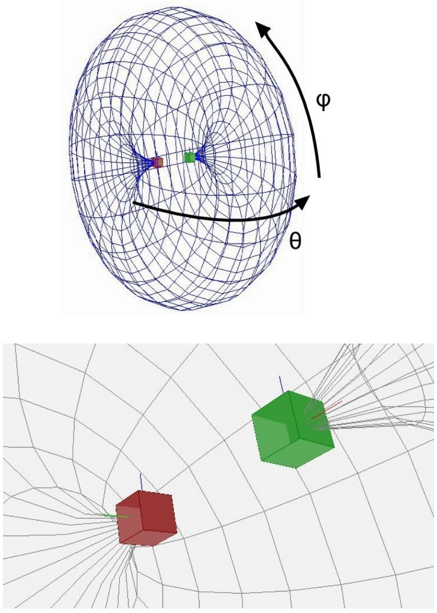

Figure 3: *Example of 3D solution set for 2 subjects. Subjects A and B are displayed resp. in red and green. Top: view of the solution set (a two parametric space* (ϕ,θ)*), for the on-screen projections of A and B, resp.* $p_A\,(-0.33;-0.33)$ *and* $p_B\,(+0.33;+0.33)$*. Bottom: the result obtained from the viewpoint parameterized with* $\phi = \frac{\pi}{6}$ *and* $\theta = \frac{\pi-\alpha}{2}$*.*

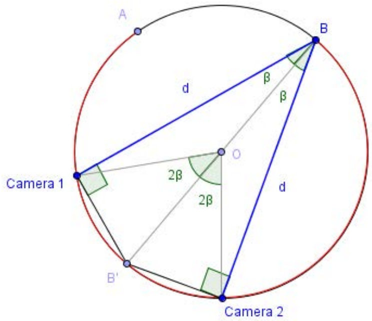

Figure 4: *Illustration of the method used to solve a distance-to-B constraint. The distance BP (P being the camera position) is d meters. The segment* BB' *is a diameter of the inscribed circle of radius r. The angle* $\beta = \cos^{-1}\left(\frac{d}{2r}\right)$ *corresponds to* $\widehat{(\overrightarrow{BP},\overrightarrow{BB'})} = \frac{1}{2}\widehat{(\overrightarrow{OP},\overrightarrow{OB'})}$*. The solution for* $\widehat{(\overrightarrow{OP},\overrightarrow{OB})}$ *is then* $\theta = \pi \pm 2\beta$*, with the restriction that P must lie on the arc* $\widehat{AB}$ *(in red).*

Definition 2 The non-oriented angle between two vectors $\vec{u}$ and $\vec{v}$ is denoted as $\left|\widehat{(\vec{u},\vec{v})}\right|$.

3. Solution in 2D

Let's first consider a 2-subject composition problem in 2D (we study the problem on a plane). In this case the screen dimension is 1D. The goal is to find a 2D camera position P such that two subjects of world positions A and B ($A \neq B$) project respectively at normalized screen positions $p_A, p_B \in [-1;+1]$, knowing the camera field of view ϕ. We here assume that A is the left-most subject (i.e. $p_A < p_B$).

The inscribed angle theorem states that the set of points P satisfying $\widehat{(\overrightarrow{PB},\overrightarrow{PA})} = \alpha$ is equal to the arc $\widehat{AB}$ (excluding A and B) of a circle $\mathcal{C}$ of center O passing through A and B, such that $\widehat{(\overrightarrow{OB},\overrightarrow{OA})} = 2\alpha$ (see figure 2). α is easily expressed from the on-screen distance between A and B and the fixed field of view ϕ. The arc $\widehat{AB}$ goes from A to B in the anti-clockwise direction, and the radius r of $\mathcal{C}$ is equal to $\frac{AB}{2\sin\alpha}$.

We now introduce two points in the camera coordinate system, projecting respectively at p_A and p_B on the screen:

$$p_A^{cam}\left(\frac{p_A}{S};1\right) \text{ and } p_B^{cam}\left(\frac{p_B}{S};1\right), \text{ with } S = \tan(\phi)$$

The solution of our 2D problem is the arc $\widehat{AB}$ such that $\alpha = \widehat{(\overrightarrow{p_B^{cam}},\overrightarrow{p_A^{cam}})}$

To compute the direction vector $\vec{f}$ of our 2D camera, we know to have $\widehat{(\overrightarrow{PA},\vec{f})} = \frac{\phi}{2}\cdot p_A$ and $\widehat{(\overrightarrow{PB};\vec{f})} = \frac{\phi}{2}\cdot p_B$. By using the same theorem we know that the line of direction $\vec{f}$ and passing through P intersects the circle $\mathcal{C}$ at a point F such that $\widehat{(\overrightarrow{OB},\overrightarrow{OF})} = \phi \cdot p_B$, and is tangent to $\mathcal{C}$ when points P and F coincide. The camera orientation is thus algebraically determined from the camera position (see figure 2).

As a conclusion, in 2D, the range of solutions to the exact on-screen composition of two subjects can be described with a single angle $\theta \in]0\,;\,2(\pi-\alpha)[$, where θ represents the angle $\widehat{(\overrightarrow{OP},\overrightarrow{OB})}$, P being the position of the camera.

4. Solution in 3D

Let's now consider this composition problem in 3D. The screen dimension is now 2D, and the goal is to find a 3D camera position P such that the two key subjects A and B ($A \neq B$) project respectively at normalized screen positions $p_A\,(x_A;y_A)\,, p_B\,(x_B;y_B) \in [-1;+1]^2$, knowing the camera field of view ϕ and its aspect ratio a. Note that ϕ define the field of view on the x axis (representing the left to right direction on screen) and that the field of view on the y axis (representing the bottom to top direction on screen) is given by $\frac{\phi}{a}$. We also assume that A is the left-most key subject (i.e. either $x_A < x_B$, or $x_A = x_B$ and $y_A \neq y_B$).

As previously, let us introduce two points described in the

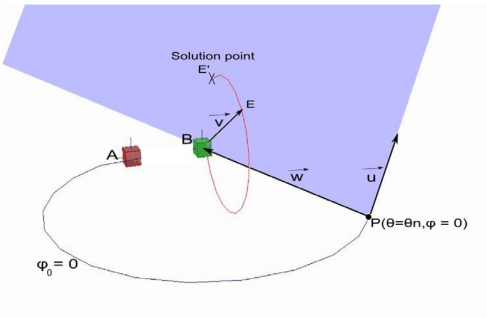

Figure 5: *Resolution of Blinn's problem. The vector (in black) starting from B should appear vertical and pointing "up" on screen. Subjects A and B are drawn resp. in red and green. The camera position P is a solution to the distance constraint (see figure 4), corresponding to a given value $\theta = \theta_n$, and located at a given vertical angle ($\varphi_0 = 0$) of the manifold. The red circle represents the appearance of $\vec{v}$ depending on the vertical angle φ applied to the viewpoint P. The half-plane (in blue) represents the set of vectors starting from B that appear vertical and pointing "up" on screen ($\vec{u}$ is the up vector of the camera when at P, and $\vec{w}$ represents $\overrightarrow{PB}$). The solution of Blinn's problem is given by the intersection (E') of the half-plane and the circle, and provide a mean to compute the rotation φ to apply.*

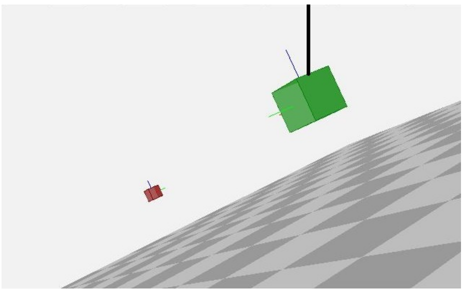

Figure 6: *Example solution of Blinn's spacecraft problem. Subjects A and B are drawn respectively in red and green. The camera must satisfy desired exact on-screen positions of A and B, and be d meters away from subject B. An additional constraint is that the vector (in black) starting from B must appear vertical and pointing "up" on screen.*

camera coordinate system, projecting respectively at p_A and p_B on screen:

$$p_A^{cam}\left(\frac{x_A}{S_x};\frac{y_A}{S_y};1\right) \quad p_B^{cam}\left(\frac{x_B}{S_x};\frac{y_B}{S_y};1\right)$$

with $S_x = \tan\phi$ and $S_y = \tan\frac{\phi}{a}$. Remember now the previous theorem. By considering a plane $\mathcal{P}$ defined by the points A, B, and a third point C not lying on the line (AB), the solution

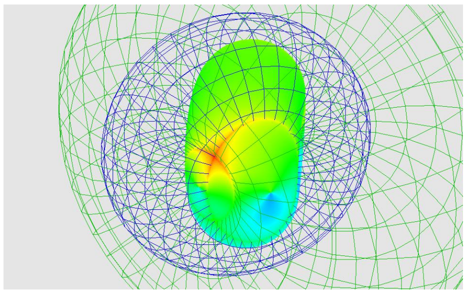

Figure 7: *Heatmap drawn on one of the manifolds. The cost function represents the sum of the values of the other two manifold equations. Red: points close to both other manifolds (drawn in blue and green), thus verifying the on-screen positioning of the 3 subjects. Blue: points farest to both other manifolds. Green: intermediate-distance points.*

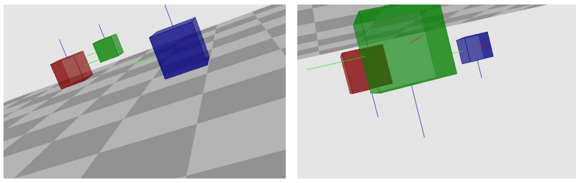

Figure 8: *Here are two solutions of the exact on-screen positioning of three subjects A (red), B (green) and C (blue). Their respective on-screen positions are $p_A(-0.50;+0.25)$, $p_B(-0.25;+0.50)$ and $p_C(+0.25;+0.50)$. These solutions correspond to the intersection of three solution set (described in figure 7), each solving the exact on-screen positioning of to of the subjects.*

of our 3D problem in $\mathcal{P}$ is the 2D arc $\widehat{AB}$ on plane $\mathcal{P}$, such that $\alpha = \widehat{(\overrightarrow{p_B^{cam}}, \overrightarrow{p_A^{cam}})}$

We now have the solution of our problem for an arbitrary plane. Let's consider that point C is the center of the inscribed circle in $\mathcal{P}$ that we called O. And let point I be the middle of segment AB. Note that from the definition of the solution in 2D, we know that $OI = \frac{AB}{2\tan\alpha}$. We can now consider a plane $\mathcal{Q}$, defined by the point I and the normal vector $\vec{n} = \overrightarrow{AB}$. The set of all possible positions for O is then the circle of center I and radius $\frac{AB}{2\tan\alpha}$ defined in the plane $\mathcal{Q}$. This, together with the solution in an arbitrary plane containing A and B, defines a unique solution set for the camera position (see figure 3).

We now show how to compute the camera orientation that satisfies the composition, assuming the camera position is P (belonging to a plane $\mathcal{P}$). We first initialize the camera orientation as a coordinate system made of three unit vectors: a forward (look-at) vector $\overrightarrow{f_i}$, a right vector $\overrightarrow{r_i}$, and an up vector $\overrightarrow{u_i}$. They are given by

$\overrightarrow{u_i} = \frac{\overrightarrow{PB}}{|\overrightarrow{PB}|} \times \frac{\overrightarrow{PA}}{|\overrightarrow{PA}|}$; $\overrightarrow{f_i} = \left(\frac{\overrightarrow{PB}}{|\overrightarrow{PB}|} + \frac{\overrightarrow{PA}}{|\overrightarrow{PA}|}\right)$ scaled to unit length ; and $\overrightarrow{r_i} = \overrightarrow{f} \times \overrightarrow{u}$. Note that $\overrightarrow{u_i}$ is normal to the supporting plane $\mathcal{P}$, and that $\overrightarrow{f_i}$ and $\overrightarrow{r_i}$ belong to $\mathcal{P}$.

We then build a quaternion q_i from these vectors, which represents a first "default" composition of A and B (i.e. $y_A = y_B = 0$ and $x_A = -x_B$). We now compute the rotation q_c such that, when applied to q_i, points A and B are projected in the appropriate locations on the screen (*i.e.* p_A^{cam} and p_B^{cam}). The solution camera orientation is then given by

$$q = q_i \cdot (q_c)^{-1} \qquad (1)$$

Note that q_c and q_i are algebraically determined from respectively the desired on-screen composition, and the camera position.

As a conclusion, in 3D, the range of solutions to the exact on-screen composition of two subjects can be described with two angles $\varphi \in]-\pi\,;\,+\pi]$ and $\theta \in]0\,;\,2(\pi-\alpha)[$, where φ represents the angle defining the supporting plane $\mathcal{P}$ and θ represents the angle $(\widehat{\overrightarrow{OP},\overrightarrow{OB}})$. Note that it is also possible to describe θ as a ratio defined in $]0\,;\,1[$ where the ratio 0 maps $\theta = 0$ and the ratio 1 maps $\theta = 2(\pi-\alpha)$. This makes θ independent from the value of α.

The position of a viewpoint $P_{\varphi,\theta}$ parameterized by (φ,θ) can then be computed as follows. We need to compute the reference position O_0 that we consider as the center of the inscribed circle for $\varphi = 0$ (e.g. such that $\overrightarrow{IO_0}$ has a z component equal to 0). We then build a quaternion q_φ as the rotation of φ radians around the axis $\overrightarrow{AB}$. The center of the inscribed circle in an arbitrary plane, that we call O_φ, is then given by $O_\varphi = (q_\varphi \cdot \overrightarrow{IO_0}) + \overrightarrow{I}$

A normal of the plane supporting this inscribed circle is $\overrightarrow{n_\varphi} = \overrightarrow{O_\varphi A} \times \overrightarrow{O_\varphi B}$. We then define a quaternion q_θ as the rotation of θ radians around the axis $\overrightarrow{n_\varphi}$, and finally obtain the position of $P_{\varphi,\theta}$ by $P_{\varphi,\theta} = (q_\theta \cdot \overrightarrow{O_\varphi B}) + \overrightarrow{O_\varphi}$

5. Solution of Blinn's Spacecraft Problem

The problem presented by Blinn [Bli88] is similar to the following. Where to put the camera and how to set its orientation knowing the result on-screen positions of two key subjects A and B, the distance d between the camera and the subject B, and a given vector (a non-null vector $\vec{v}$ starting from B) that should appear vertical and pointing "up" on the screen? An illustration of this problem is given in figure 6.

First, we can already state that the solution of this problem belongs to the 2D manifold defined in section 4. Moreover, by using the analytic solution in 2D, the distance constraint is quite simple to resolve. Indeed, let's consider a plane $\mathcal{P}$, and define a point B' as the symmetric point to B w.r.t. the center point O on the inscribed circle. We will use the properties between angles and sides in the triangle BPB' which is rectangle in P. Let's define the angle β corresponding to $(\widehat{\overrightarrow{BP},\overrightarrow{BB'}}) = \frac{1}{2}(\widehat{\overrightarrow{OP},\overrightarrow{OB'}})$. Then we can state that there are 0, 1 or 2 solutions to this distance problem:

- no solution if $d = 0$ or $d > 2r$
- a single solution if $d \in]0\,;\,AB]$: $\theta = \pi - 2\beta$
- a single solution if $d = 2r$: $\theta = \pi$
- two solutions if $d \in]AB\,;\,2r[$: $\theta = \pi \pm 2\beta$

The resolution method is illustrated in figure 4. This solution in 2D works in 3D since it is valid for every value of the parameter φ (vertical angle). The set of solutions to the distance constraint is therefore a circle on the 2D surface such that every point of the circle is at a distance d to B, *i.e.* this is the set of points such that $\theta = \theta_n$ on the manifold.

Now let's take a look at the "up" vector constraint. In the following, we will assume that the position of the camera satisfying all the composition constraints is P. As stated above, P belongs to a circle $\mathcal{C}_n$ defined by the parameter $\theta = \theta_n$ (solution to the distance constraint).

Let's consider $\overrightarrow{u}$ the up vector of the camera (provided from the quaternion q described in equation 1) at position P, and $\overrightarrow{w}$ the vector representing $\overrightarrow{PB}$. The vector $\overrightarrow{v}$ must appear vertical on screen, thus $\overrightarrow{u}$, $\overrightarrow{w}$ and $\overrightarrow{v}$ must be coplanar. Moreover $\overrightarrow{v}$ should point "up" on screen. We must thus have $\overrightarrow{u} \cdot \overrightarrow{v} > 0$. Blinn's problem now corresponds to finding a position P on the circle $\mathcal{C}_n$ such that these two constraints are satisfied.

From the last section, we know that the camera coordinates (i.e. its position and orientation) correspond to a rotation of the coordinates of the camera when at a reference position $P_{0,n}$ (parameterized by $(\varphi = 0, \theta = \theta_n)$) on the circle $\mathcal{C}_n$ by an angle of φ radians around the axis $\overrightarrow{AB}$. Our problem is then similar to searching the rotation of φ radians around $-\overrightarrow{AB}$ to apply to the vector $\vec{v}$ such that it appears vertical and pointing "up" on screen when the camera is set to $P_{0,n}$. Instead of searching the camera position P on the circle $\mathcal{C}_n$, we then build a circle $\mathcal{C}$ representing the range of possible positions for the extremity of the vector $\vec{v}$, and search a position on $\mathcal{C}$ such that the previous property is satisfied.

Before solving this problem, let us define the vectors $\overrightarrow{u_{0,n}}$ as the up vector of the camera when at $P_{0,n}$, $\overrightarrow{w_{n,0}}$ as the vector $\overrightarrow{P_{0,n}B}$, and $\overrightarrow{v_0}$ as the vector $\vec{v}$ rotated by an angle of φ radians around $-\overrightarrow{AB}$ (i.e. the solution vector). We then solve the problem as follows.

We first define a plane $\mathcal{P}$ by a normal vector $\overrightarrow{n} = \overrightarrow{u_{0,n}} \times \overrightarrow{w_{0,n}}$ (scaled to unit length) and the point $P_{0,n}$. By definition, B belongs to $\mathcal{P}$. We then define a half-plane $\mathcal{P_H} \subset \mathcal{P}$ delimited by the line $\mathcal{L}$ of direction $\overrightarrow{w_{0,n}}$ passing through $P_{n,0}$. The points belonging to $\mathcal{P_H}$ are located in the direction of $\overrightarrow{u_{0,n}}$. This half-plane defines the set of vectors that will appear vertical and pointing "up" on the screen.

We then define an other plane $\mathcal{Q}$ by a normal vector

$\vec{t} = \overrightarrow{AB}$ (scaled to unit length) and a point F given by $\vec{F} = \vec{B} + \vec{t} \cdot |\vec{v}| \cdot \cos(\pi - \gamma)$, where $\gamma = \widehat{(\vec{t}, \vec{v})}$ (N.B. F belongs to the line (AB)). The circle $\mathcal{C}$ is then defined as the circle of center F and radius $r' = |\vec{v}| \cdot \sin(\pi - \gamma)$ in the plane $\mathcal{Q}$. Note that the vertical angle φ that is solution to the problem will be given by the angle $\widehat{(\overrightarrow{FE'}, \overrightarrow{FE})}$.

The solution of Blinn's problem is then given by the intersection(s) of the half-plane $\mathcal{P}_{\mathcal{H}}$ and the circle $\mathcal{C}$. In other words, by computing the intersection $\mathcal{I}$ of $\mathcal{P}_{\mathcal{H}}$ and $\mathcal{Q}$ (a half-line or the half-plane $\mathcal{P}_{\mathcal{H}}$ itself), then the intersection(s) of $\mathcal{I}$ and the circle $\mathcal{C}$ in the plane $\mathcal{Q}$. The figure 5 provides an illustration of this geometric resolution, and the figure 6 provides an example of solution computed with this method.

6. Solution for 3 or more subjects

The approach is extensible to 3 or more subjects: lets consider 3 subjects with respective positions A, B, C (different from each other). The solution set for each pair of subjects is a 2D manifold. Consequently, any camera position P satisfying the 3-subject composition should belong to the intersection of three manifolds M, M' and M'' respectively defined by couples (A,B), (B,C) and (A,C), and such that the camera orientation at P matches for these three manifolds.

Let first define three angles α, α' and α'' corresponding to M, M' and M''. Similarly let q, q' and q'' be the orientations (defined as unit quaternions) at a given 3D point P, as defined respectively on M, M' and M''. Then, if P belongs to M (for instance) it verifies $\widehat{(\overrightarrow{PB}, \overrightarrow{PA})} - \alpha = 0$

We propose here an algorithm which searches on the surface of one manifold M the configuration which maximizes the viewpoint quality (defined as a distance to the other manifolds). Typically the on-screen position of two subjects is fixed, and we optimize the on-screen position of the third subject. The cost function to minimize is then

$$\min_{\theta,\varphi} \left(\left| \widehat{(\overrightarrow{PC}, \overrightarrow{PB})} - \alpha' \right| + \left| \widehat{(\overrightarrow{PC}, \overrightarrow{PA})} - \alpha'' \right| + \left| 1 - \left\langle q', q'' \right\rangle^2 \right| \right)$$

The heatmap corresponding to this cost function and an illustration of the solutions of this composition problem are given respectively in figures 7 and 8.

The main advantage of this technique is that it can easily extend to more than 3 subjects. Though the composition may not be strictly satisfied, it enables fixing the on-screen position of two (main) subjects, then optimizing the position of other subjects. All other visual properties can be accounted for in this optimisation on the manifold surface (target size, vantage angle, rectangular framing of targets).

7. Conclusion

We have introduced a simple parametric model to solve a range of problems that occur in the task of positioning a virtual camera given exact on-screen specifications. Our model solves Blinn's spacecraft problem [Bli88] by using an algebraic formulation rather than an iterative process. It casts camera optimization problems mostly conducted in 6D into searches inside a 2D space on a manifold surface. Interestingly, our model can be easily extended to integrate most of the classical visual properties employed in the litterature [RCU10]. For example, size of key subjects (or distance to camera) can be expressed as the set of viewpoints on the manifold which are at a given distance from the camera (resolves as 0, 1 or 2 lines on the manifold surface). In a similar way, vantage angle properties (eg. see the front of a subject) represent sub-regions of the manifold.

By reducing the search space to a manifold where the on-screen location of subjects are exact, we obviously restrict the generality of the technique. However, the benefits in terms of computational cost greatly favors our approach. Though the solution for two subjects appears easy to formulate with vector algebra, it has not been reported before and the model serves as an expressive way on which to build more evolved techniques. The techniques presented in the paper have the potential to replace most of the previous formulations related to camera control with a simpler and more efficient approach, and opens great possibilities to include more evolved on-screen composition techniques in a large range of applications in computer graphics.

References

[ACOYL08] Assa J., Cohen-Or D., Yeh I.-C., Lee T.-Y.: Motion Overview of Human Action. *ACM Transactions on Graphics* (2008). 2

[Bli88] Blinn J.: Where am I? what am I looking at? *IEEE Computer Graphics and Applications 8*, 4 (1988). 1, 2, 5, 6

[BMBT00] Bares W., McDermott S., Boudreaux C., Thainimit S.: Virtual 3D Camera Composition from Frame Constraints. In *Proceedings of the 8th ACM International Conference on Multimedia* (2000). 2

[BY10] Burelli P., Yannakakis G. N.: Global Search for Occlusion Minimisation in Virtual Camera Control. In *IEEE Congress on Evolutionary Computation* (Barcelona, 2010). 2

[CN05] Christie M., Normand J.-M.: A Semantic Space Partitioning Approach to Virtual Camera Composition. *Computer Graphics Forum 24*, 3 (2005). 2

[CON08] Christie M., Oliver P., Normand J.-M.: Camera Control in Computer Graphics. *Computer Graphics Forum 27*, 8 (2008). 2

[HO00] Halper N., Olivier P.: CAMPLAN: A Camera Planning Agent. In *Smart Graphics* (2000). 2

[LCL*10] Lino C., Christie M., Lamarche F., Schofield G., Olivier P.: A Real-time Cinematography System for Interactive 3D Environments. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2010). 2

[RCU10] Ranon R., Christie M., Urli T.: Accurately Measuring the Satisfaction of Visual Properties in Virtual Camera Control. In *Smart Graphics*. 2010. 2, 6