Eurographics Workshop on Intelligent Cinematography and Editing (2015)
W. Bares, M. Christie, and R. Ronfard (Editors)

Toward More Effective Viewpoint Computation Tools

Christophe Lino

INRIA Rennes - Bretagne Atlantique / IRISA, France

Abstract

*The proper setting of cameras is an essential component in many 3D computer graphics applications. Commonly, viewpoint computation tools rely on the specification of visual criteria on a number of targets, each expressed as a constraint; then on the use of an optimization-based technique to compute a 7-degrees of freedom camera setting that best satisfy this set of constraints. Proposed methods can be evaluated in terms of their efficiency (required computation time), but there is a clear lack of a proper evaluation of their effectiveness (how aesthetically satisfactory the generated viewpoints are). In parallel, current methods rely on the maximization of a single fitness function built as a weighted sum (*i.e.* a pure tradeoff) over the satisfaction of each single criterion considered independently from all others. In contrast, cinematographers' sense of the effective satisfaction of a viewpoint is far from a tradeoff between visual criteria. These issues call for the provision of means to better evaluate the overall satisfaction of a composition problem, and methods to improve the search of a satisfactory viewpoint. In this paper, we present a work in progress which targets to steer computation tools in this direction. We first propose a range of aggregation functions which supplement the classical tradeoff function, and enable to express evolved relationships between criteria. We then propose to aggregate the individual satisfactions of criteria in hierarchical way instead of simply summing them. We finally propose to reduce the search to camera positions (*i.e.* from 7D to 3D), while constraining the framing more strongly by separately optimizing its orientation and focal length.*

Categories and Subject Descriptors (according to ACM CCS): J.5 [Computer Graphics]: ARTS AND HUMANITIES—H.3.4 [Computer Graphics]: Systems and Software—Performance evaluation (efficiency and effectiveness) G.1.6 [Computer Graphics]: Optimization—

1. Introduction

The fast-growing number of interactive computer graphics applications (such as 3D games or movie production tools), together with 3D virtual worlds becoming more and more realistic, call for efficient computation methods able to produce viewpoints satisfying a range of aesthetic criteria on a number of targets. Visual criteria include targets' on-screen size, position, view angle, or visibility, and may also include more subtle aspects such as the visual balance or depth of field in a shot.

Most efficient methods rely on an any-time generation of viewpoints through optimization techniques. This also calls for means to guarantee that the proposed methods are able to return a visually acceptable viewpoint at any time during the search, even if the overall problem cannot be fully satisfied at this stage. However, the current construction of a fitness function (as a weighted sum) leads to difficulties in properly expressing the viewpoints quality and effectively guiding the search through viewpoints which are visually more and more acceptable. Furthermore, the task of tuning weights is known to be a nightmare, while the obtained weight distribution is often suitable for a single problem only. Constructing the fitness function as a pure tradeoff also leads to the extreme difficulty to express preferences in terms of partial criteria satisfactions (*e.g.* two constraints are both satisfied at 50%, versus one is satisfied at 75% while the other is satisfied at 25%). Finally, the orientation and focal length parameters are strongly dependent on the position. We argue that considering the 7 camera parameters as completely independent leads to a difficulty to reach a proper framing of targets.

In contrast, though cinematographers' sense of the aesthetic satisfaction of a viewpoint cannot be easily put into equations, it is really far from a pure tradeoff between all visual criteria; they conversely often focus on some visual criteria that are greatly satisfied, or insufficiently satisfied,

DOI: 10.2312/wiced.20151076

and their evaluation of the satisfaction degree of some criterion is often made with regards to others' satisfaction (*e.g.* if some target is occluded or not framed by the camera, it's no big deal if other visual criteria are satisfied).

In this paper, we address two main concerns: (i) providing a mean to better express and guide the search toward visually acceptable viewpoints according to an input visual layout to satisfy, and (ii) improving the search process by separating the optimization over camera positions and the optimization over the other camera parameters, that strongly depend on the position (namely its orientation and focal length).

The contributions of this paper are:

- Novel aggregation functions supplementing the classically used tradeoff function; they provide powerful means to account for evolved relationships between criteria;
- A hierarchical aggregation method, which eases the design process of the fitness function while it enables an effective evaluation of the overall problem satisfaction;
- An intertwined optimization of the camera position on one side, and of its orientation and focal length (for a given position) on the other side .

This paper is organized as follows. First, we review the current state of the research on viewpoint computation tools. Second, we detail our novel aggregation functions, and describe our hierarchical aggregation method. Third, we detail our separate optimization of the orientation and focal length. We finally conclude on further potential improvements of viewpoint computation tools.

2. Related Work

Implementing cinematic knowledge related to visual composition requires considering a combination of both visual properties (*e.g.* targets' size or on-screen position) and properties on the camera (*e.g.* targets' view angle or focal length). With this in mind, researchers have proposed a range of declarative approaches, relying on the specification of visual and camera properties, then on the use of a constraint-based and/or optimization-based technique to compute the camera parameters that best satisfy those properties. An overview of existing methods is given in [CON08].

The most efficient computation method to date [RU14] relies on an any-time generation of viewpoints, based on a Particle Swarm Optimization (PSO) technique. PSO is a population-based method for global optimization. [RU14] approximates targets by bounded boxes and account for criteria such as their occlusion, screen inclusion/exclusion, view angle, size, on-screen position (as frames) and relative on-screen position (left/right), as well as geometric constraints on the camera position. Each particle is represented by a position and velocity, both aggregating the camera position, orientation and focal length. Interestingly, their method build upon algebraic solutions of some constraints to smartly select an initial population of particles into their solutions spaces; this greatly increases the chances to start from camera positions close the optimum of the fitness function. The search is finally performed as an iterative process, which at step n modifies the velocity and position of each particle based on the fitness values obtained at step $n-1$. During the search, the particle best visited positions are memorized.

More generally, optimization-based approaches suffer from the difficulty of modeling objective functions for each property, then aggregating these properties into a single fitness function (built as a weighted sum); this leads to a tedious process of tuning objective functions and their weights. Further, it leads to considering each criterion satisfaction separately. It is then difficult to handle complex relationships between visual criteria which make a viewpoint is (or is not) considered as satisfactory by human viewers. Finally, these techniques suffer from an insufficient consideration of the dependency of the camera orientation and focal length with relation to its position. Though some methods, such as [RU14], propose to consider optimizing a look-at point instead of directly optimizing the camera orientation, this also lead to some instability in the camera orientation due the randomness in the search algorithm. In a nutshell, most current optimization solvers suffer from the difficulty to properly guide the framing of targets.

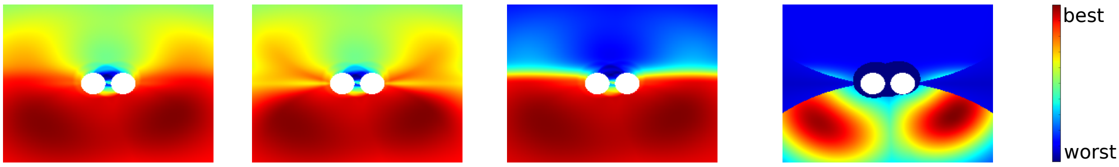
We here identified two issues. First, there is a lack in effectively guiding, while remaining efficient, the search process through visually satisfactory viewpoints. Second, though it is yet possible to make comparisons between method in terms of their efficiency, the community misses means to evaluate them in term of their effectiveness. In this paper, we propose a computational model targeting an effective evaluation of viewpoints' satisfaction, regarding an input specification. We also propose to search in 3D (camera positions), while better constraining the framing of targets.

3. Key visual criteria

This section reviews an extensive set of visual criteria, which we find relevant for evaluating the effective satisfaction of a viewpoint. We study the way each can be expressed as an objective function, and the relationships that may exist between criteria.

First, common practice in composing film shots furnish a set of explicitly declarable criteria. They can often be expressed as relatively simple visual or geometric constraints. Note that Ranon *et al.* [RCU10] provides an extensive review of such declarative constraints used in the community, and means to accurately measure their satisfaction. Consequently, this paper does not focus on this particular aspect.

Position A target's on-screen position can be described as the desired position (x,y) of its point of interest (*e.g.* eyes or head), with some freedom (*e.g.* a maximum distance between the target projection and its ideal position). Such



(a) balanced tradeoff (b) unbalanced tradeoff (c) unbalanced tradeoff (d) hierarchical aggregation

Figure 1: *Comparison between a simple tradeoff (as used in previous work) and a hierarchical aggregation of objective functions. A balanced tradeoff between all criteria, as in (a), leads to a really smoothed fitness function; a decrease on one criterion's satisfaction (*e.g.* visibility or screen ordering) can always be compensated by a sufficient increase on any other criterion's satisfaction. Giving more weight to a hard constraint, such as visibility in (b) or screen ordering in (c), leads to weaken the requirement to satisfy other constraints. A hierarchical aggregation, as shown in (d), provides a mean to more easily highlight satisfactory and non-satisfactory viewpoints.*

constraints' satisfaction can be evaluated through a distance metric taking as input both the desired position (x,y) and actual position (x',y') of the target.

Size A target's size is typically described as its body parts appearing on the screen, and their size (*e.g.* for a character, a close framing of its head in a Close-up shot, or a framing of both its head and its shoulders in a Medium Close-up shot). Such constraints satisfaction can be evaluated with a similarity value between the amount of the target appearing and the amount which should appear on the screen. Each body part might be abstracted as a bounding volume, then one might evaluate the amount of each volume (desired or not) which projects on the screen.

Viewing angle A target's viewing angle is characterized through two sub-criteria: an horizontal viewing angle (*e.g.* see the target from its front, side or back) and a vertical viewing angle (see the target from top to bottom, from bottom to top, or from its eye level). Such constraints can be expressed geometrically from the camera position, as a viewing direction with an allowed angular relaxation. Their satisfaction might be computed as an angular difference between the desired and actual viewing directions, taken from the target to the camera position.

Visibility The visibility of a target is often defined as the percentage of a target which is visible on the screen (*i.e.* not occluded by some other scene element located between the camera and the target). Practically, the visibility criteria satisfaction is often computed by either (i) casting a number of rays from the camera to the target geometry and checking how many rays collide with other scene elements, or (ii) using an occlusion query technique.

Some visual constraints relate to higher level aesthetic rules, which are not considered here. One might, for example, be concerned with subtle visual criteria such as the look-room, head-room or foot-room of a target (amount of space left in its gaze direction, over and underneath its head, and underneath its feet, respectively). One might also account for aesthetic criteria such as the visual balance of a shot (each on-screen object is abstracted as a visual mass; a shot is then considered well-balanced if the centroid of all visual masses is located at the center of the screen). Though they impact the quality of viewpoints, such criteria remain difficult to express visually or geometrically.

In addition, one should also consider implicit constraints related to the underlying camera configuration space, such as the inclusion or ordering of targets on the screen.

Inclusion The inclusion of a target can be defined as the percentage of a target which appear on the screen. Practically, its satisfaction can be computed by first projecting the desired target's geometry on the camera plane, then evaluating how much of this geometry is bounded by the screen extents.

Screen Ordering The screen ordering of targets directly derives from their declared positions. One should then ensure the relative on-screen positioning of targets (typically their left-to-right ordering). Each two-target ordering can be expressed as a geometrical constraint, by using a 2D line (or a vertical plane) passing through both targets; the camera should then be positioned on one side of this line or plane.

At this point, note that, from a viewer's point of view, the consideration of layout criteria (such as the screen position, size, or viewing angle) for a given target is subject to its sufficient inclusion and visibility.

One might finally consider satisfying continuity editing rules when searching viewpoints (they typically apply when cutting between two viewpoints in a movie). Note that, to our knowledge, these criteria have not yet been addressed with optimization-based techniques, due to the difficulty to properly model their aggregation with other constraints by using a weighted sum.

View angle change The view angle change rule state that one should ensure a minimum angle between the viewing angle of at least one target appearing on the screen. This is commonly expressed as a 30 degree angle between its viewing directions taken from the target to the camera.

Size change The size change rule state that one should ensure a minimum change in the shot size of at least one target appearing on the screen. This is classically expressed as a significant change in the shot size of the target (though in the literature there is no precise value provided on the amount of change which should occur).

180 degree rule The 180 degree rule state that one should maintain the relative on-screen positions of targets. This criteria is very similar to the on-screen ordering criteria.

Motion and gaze continuity The motion and gaze continuity rules state that one should enforce the apparent motion and gaze of a target appearing on the screen. Similar to the 180 degree rule, both rules can be expressed as a line or a plane that the camera should not cross when cutting.

Note that the view angle change and size change both consider complementary constraints, and these two rules also complement each other (one should satisfy at least one such constraint). Conversely, the 180 degree rule, and the motion and gaze continuity rules are supplementary constraints (one should strictly satisfy every such constraint).

From a general view, note that satisfying a criterion may also conflict with satisfying one another, especially when both criteria are related to the same target. This is a important aspect to consider when searching viewpoints.

4. Overview

Common viewpoint computation requires searching over 7 camera parameters (position, orientation and focal length) while relying on a pure tradeoff between visual criteria.

We firstly argue that a pure tradeoff (*i.e.* a weighted sum) prevent from expressing evolved relationships between criteria; its main feature is to smooth all satisfaction values over the whole search space. In the following sections, we propose novel aggregation functions addressing the problem of expressing such relationships. We then propose a hierarchical evaluation of criteria satisfaction, aiming to more easily and better express an overall problem satisfaction. We secondly argue that the camera orientation and focal length should be considered as strongly dependent from its position; they should consequently be optimized according to the camera position. We propose to start from an initial orientation and focal length (computed algebraically from visual criteria), then to rely on a few pan, tilt and zoom corrections to improve the framing of targets until convergence.

We detail our propositions in the next sections. To illustrate them, we have sampled camera positions around a pair of targets and computed the camera orientation and focal length to obtain the best possible value of fitness. We further provide 2D heat-maps representing the satisfaction of described functions around this pair of targets, for cameras positioned at their eye-level.

5. Aggregation functions

In this section, we assume each criterion i is cast into a satisfaction function f_i returning a value ranging from 0 (not satisfied at all) to 1 (fully satisfied). Building upon this definition, we define an aggregation function as a function taking one or more satisfaction functions as input and returning a new value in $[0;1]$; it can further be considered as kind of satisfaction function, which can then be given as input to a new aggregation function.

Common optimization-based viewpoint computation systems use a single *tradeoff* function to aggregate all criteria satisfaction. Figure 1 illustrates the difficulty, when using a pure tradeoff, to guide the search and evaluate the effective satisfaction of a viewpoint. The example composition problem comprises criteria such as targets' visibility, size, screen position and viewing angle. The figure also provides a visual comparison with the satisfaction value returned by using an evolved aggregation tree, showing a clear improvement regarding the expression of the overall problem satisfaction. Through this section, we present both the *tradeoff* function and our novel aggregation functions. For each function, we explain how it can be useful for expressing relationships between criteria.

5.1. Tradeoff

The *Tradeoff* function denotes the desire to balance the satisfaction of two or more constraints (*i.e.* to make a tradeoff between all criteria satisfactions). This aggregation function is expressed as a weighted sum of the objective functions of accounted constraints:

$$T(\{f_1,\ldots,f_m\},\{w_1,\cdots,w_m\},q) = \frac{1}{\sum_i w_i}.\sum_i w_i.f_i(q)$$

Decreasing the satisfaction of one criterion can be balanced by equally increasing the satisfaction of another criterion. Consequently, for two constraints i and j with respective weights $w_i = 1$ and $w_j = 2$, the tradeoff function will make no difference between the case where both constraints are satisfied at 50%, and the case where constraint i is satisfied at 80% while constraint j is satisfied at 35%.

5.2. Joint satisfaction

The *Joint* satisfaction function denotes the desire to jointly satisfy a subset of m constraints ($m \geq 2$). It can be viewed as the numerical intersection of their objective functions. The local search will then be guided by the locally worst-satisfied constraint. This aggregation function is expressed as the lowest satisfaction value returned by an input objective function:

$$J(\{f_1,\ldots,f_m\},q) = \min\{f_1(q),\ldots,f_m(q)\}$$

We illustrate the *Joint* satisfaction function through the aggregation of visibility criteria on a pair of targets (see figure 2).

5.3. Least satisfaction

The *Least* satisfaction function denotes the desire to satisfy at least one constraint among a subset of m constraints

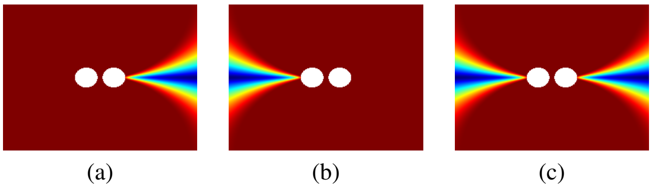


Figure 2: *Joint-satisfaction function. (a) satisfaction of the visibility criterion on the left target; (b) satisfaction of the visibility criterion on the right target; (c) joint satisfaction of both visibility criteria.*

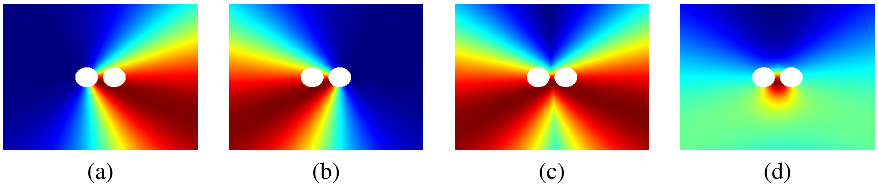


Figure 3: *Least-satisfaction function. (a) satisfaction of a vantage angle on the left target; (b) satisfaction of a vantage angle on the right target; (c) satisfaction of at least one of the vantage angles; (d) tradeoff between both angle satisfaction. This figure can clearly illustrate how a tradeoff between vantage constraint leads to viewpoints which may not satisfy at least one of them.*

($m \geq 2$), at the potential expense of other constraints of the set not being satisfied. It can be viewed as the numerical union of their objective functions. The search will then be guided by the locally best-satisfied constraint. This aggregation function is expressed as the highest satisfaction value returned by one of the input objective functions:

$$L(\{f_1,\ldots,f_m\},q) = \max\{f_1(q),\ldots,f_m(q)\}$$

We illustrate the *Least* satisfaction function through the aggregation of viewing angle criteria on a pair of targets (see figure 3).

5.4. Adjustment satisfaction

The *Adjustement* satisfaction function denotes the desire to adjust two or more criteria regarding each other. This is typically desired when a change on a shared camera parameter has opposite effects on the satisfaction of a set of criteria. It can be viewed as a tweaking process leading to a good compromise between the satisfaction of each criterion of the set. The search will then be guided by a tweaking constraint, for which decreasing the satisfaction of a criterion requires to over-compensate with an even higher increase of the satisfaction value of some other criteria of the set. This aggregation function is expressed as the product of all satisfaction values returned by input objective functions:

$$A(\{f_1,\ldots,f_m\},q) = \prod_i f_i(q)$$

We illustrate the *Adjustement* satisfaction function through the aggregation of targets' size criteria with the focal length criterion (see figure 4).

5.5. Prerequisite satisfaction

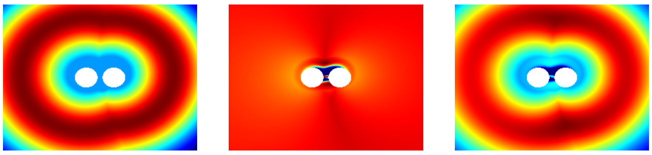
The *Prerequisite* satisfaction function denotes the desire to account for a given constraint only when a prerequisite (*i.e.* the satisfaction of another constraint) is ensured. Consequently, the local search will first prune regions of the space where the prerequisite is not ensured. It will then allow guiding the search by only accounting for the constraint, while ensuring the prerequisite is satisfied. This aggregation function is expressed through a threshold-based filtering, computed as follows:

$$P(p,\tau,f,q) = \begin{cases} 0 & \text{if} p(q) < \tau \\ f(q) & \text{if} p(q) \geq \tau \end{cases}$$

We illustrate the *Prerequisite* satisfaction function through the aggregation of hard constraints, such as inclusion or visibility of targets, with the visual layout criteria of targets (see figure 5).

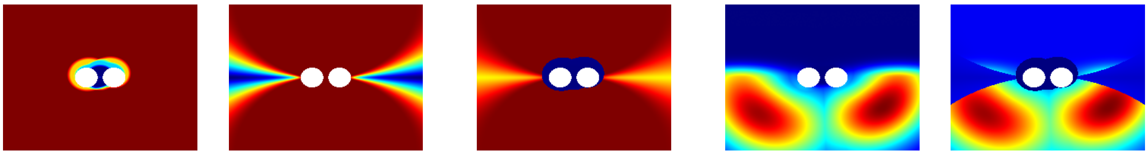
6. Fitness as an aggregation tree

As presented earlier, the overall satisfaction of a composition problem is subject to many evolved relationships between the satisfactions of the different visual or geometrical criteria. One can represent such relationships as a tree of inter-dependencies. We consequently propose to combine the satisfaction of declared criteria through a hierarchy of aggregation functions, each aiming to handle one single subproblem at a time. The main advantage of doing so is to make the fitness function easier to design by enabling to focus on simpler problems, instead of considering the whole problem at a glance. We think the evaluation of an overall problem satisfaction with a tree representation will enable a



(a) (b) (c)

Figure 4: *Adjustment satisfaction function. (a) satisfaction of the size criteria on both targets; (b) satisfaction of the ideal focal length criterion; (c) satisfaction of the adjustment between the size of targets and the ideal focal length.*



(a) inclusion (b) visibility (c) inclusion prerequisite (d) layout criteria (e) fitness function

Figure 5: *Prerequisite satisfaction function. (a) joint satisfaction of the targets' screen inclusion; (b) joint satisfaction of both targets' visibility; (c) prerequisite that targets should appear on the screen, before considering their visibility; (d) aggregated satisfaction of the other layout criteria; (e) fitness function expressed as a prerequisite to satisfy both the on-screen inclusion and an sufficient visibility of targets, before considering their on-screen layout.*

more evolved search process, while our guess is that the employed optimization algorithm will also be able, at each new iteration, to provide viewpoints which are (effectively) more satisfactory.

Algorithm 6.1: FITNESS(q)

$Visibility(q) \leftarrow J(\{vis_1, vis_2\}, q)$
$Inclusions(q) \leftarrow J(\{inc_1, inc_2\}, q)$
$Positions(q) \leftarrow T(\{pos_1, pos_2\}, \{1,1\}, q)$
$Individual_1(q) \leftarrow J(\{size_1, angle_1, height_1\}, q)$
$Individual_2(q) \leftarrow J(\{size_2, angle_2, height_2\}, q)$
$Individuals(q) \leftarrow A(\{Individual_1, Individual_2\}, q)$
$Layout(q) \leftarrow A(\{Individuals, flength, Positions\}, q)$
$OrderedLayout(q) \leftarrow P(ordering, 0.9, Layout, q)$
$Pre(q) \leftarrow P(Inclusions, 0.99, Visibility, q)$
$Fitness(q) \leftarrow P(Pre, 0.8, OrderedLayout, q)$

To illustrate this proposition, we considered the construction and evaluation of an aggregation tree for two-target shots, described in algorithm 6.1. For each target, a joint satisfaction of its size and view angle (both horizontally and vertically) is required, as they are complementary criteria; this intersection defines a region which best satisfies the individual layout of one target. Both individual constraints are then adjusted with the targets' screen positions and focal length constraints (as shown in [LC12], both criteria are strongly linked for each pair of targets); this adjustment represents the tweaking process required to best satisfy the on-screen layout of all targets while minimizing the distance to the desired focal length. Two prerequisites are then specified : (i) the targets' screen ordering should be ensured, and (ii) sufficient inclusion and visibility should be provided for all targets. Note that we do not claim that no other aggregation tree could perform well (or better) for certain cases. At least, the tree we crafted seems to work well for the two-target viewpoint examples that we designed (see the results section), without relying on any previous tuning process (when a tradeoff is made, weights are consistently set to 1).

7. Intertwined optimization

This section addresses the provision of a separate optimization on the camera orientation together with its focal length, assuming an input camera position is given. In a way similar to Blinn's algorithm [Bli88], we propose to iterate on (i) computing a new orientation from its focal length, and (ii) computing a new focal length from its orientation. To do so, we build upon the desired on-screen size and position of each target (see algorithm 7.1).

Algorithm 7.1: ORIENTATION_AND_FOCAL(pos)

$focal \leftarrow desiredFocalLength$
$w_1, w_2 \leftarrow worldPosition_1, worldPosition_2$
$p_1, p_2 \leftarrow desiredScreenPosition_1, desiredScreenPosition_2$
$s_1, s_2 \leftarrow desiredSize_1, desiredSize_2$
repeat
  $ori \leftarrow bestOrientation(pos, focal, p_1, p_2, w_1, w_2)$
  $focal \leftarrow bestFocal(pos, ori, s_1, s_2)$
until convergence of $focal$

The search is initiated by taking the desired focal length as initial value. In a first step, from this initial focal length, we compute the camera orientation that best satisfies the desired targets' on-screen positions. We do so by using a generalization of the look-at operator, described in [LC15]; practically, we compute an orientation such that the camera roll angle is set to 0 degrees and that each target's head center is positioned as close as possible to its desired position. In a second step, from this new orientation, we compute the focal length that best satisfies the targets' inclusion (*i.e.* the whole desired targets' geometry to view is included in the camera frustum). From each on-screen size criterion, we extract the corresponding target's geometry to view (*i.e.* vertices). From the previously computed orientation, we also extract the camera coordinate system, defined as a triplet of axes r, f and u (right, forward and up respectively). For each vertex i, we then compute a direction vector v_i taken from the camera to the vertex, and compute two projections of this vector: we project v_i on the plane (r, f), resulting in a vector v_i^x; and we project v_i on the plane (f, u), resulting in a vector v_i^y. The minimal horizontal field of view angle ϕ_i^x required to include this vertex is computed as twice the angle between vectors f and v_i^x. Similarly, the minimal vertical field of view angle ϕ_i^y required to include this vertex is computed as twice the angle between vectors f and v_i^y. ϕ_i^x and ϕ_i^y are then cast into focal lengths f_i^x and f_i^y respectively. The optimal focal length is computed as the minimum over all f_i^x and f_i^y. We iterate on these two steps until convergence of this focal length value.

This separate optimization process reduces the general search to the 3D space of camera positions. We then rely on a Particle Swarm Optimization similar to [RU14]. Then, each time we sample a new camera position, we optimize its orientation and focal length using the above method; by doing so, the evaluation of this position is done only once, and maximizes its fitness value. For this reason, we think this inter-twinned optimization has the potential to provide a better guidance of the search process through visually acceptable viewpoints.

8. Results

Benefits of our two propositions are illustrated by comparing four different search methods. We show the impact of using either a fitness function built as a pure tradeoff or as an evolved aggregation tree, and either searching in 7D or 3D. In the pure tradeoff version, all criteria have been given the same weight (*i.e.* assuming no prior tuning process). For comparison purposes, we designed two viewpoints specifications that voluntarily contain inconsistencies:

Viewpoint #1: Marty and George are looking at each other. As we need to film a dialogue, we want to film both from a front view (which conflicts with their respective orientation); we also want to film both in Medium closeup, *i.e.* with both the head and shoulders, and at their eye-level; Marty should appear on the top left third and George on the top right third of the screen; finally, the operator should ideally use a 25mm focal length.

Viewpoint #2: Lou is passing behind Marty; they are looking in opposite directions. We want to film both from front view (which conflicts with their respective orientation); we also want to film both in Closeup, *i.e.* with only the head (which conflicts with their respective distance), and at their eye-level; Marty should appear on the top left third and Lou on the top right third of the screen; finally, the operator should ideally use a 35mm focal length.

We considered a bounding box $\mathcal{B}$ of the desired targets' geometry to view, and defined our search space as 10 times wider than $\mathcal{B}$, which seems wide enough to contain any potential solution viewpoint. We then generated viewpoints, using a 5ms time window, through methods built as a combination of a search dimension (7D or 3D) and an aggregation method. Result screenshots are shown in figure 6 and the corresponding fitness functions are illustrated in figure 7.

As showed in figure 6, each of our propositions separately provides an improvement regarding the proper framing of targets, but does not bring sufficient effectiveness on its own. On one hand, our hierarchical aggregation penalizes weak inclusion or bad on-screen positioning of targets. However, by searching in 7D, it will also penalize a camera position from which the overall problem can be satisfied if its orientation and/or focal length do not provide a proper framing of the targets; this is likely to (at least) impact the algorithm efficiency by requiring to search more extensively. On the other hand, our inter-twined optimization better constrains the on-screen positions and sizes of targets, from any position. When making a pure tradeoff between visual criteria, the algorithm is however unable to reach a viewpoint with sufficient satisfaction (one target is framed from back view instead of front view, or the targets are framed too wide or too close). Finally, when combining both propositions, the search reaches a satisfactory viewpoint more rapidly than when using one of our propositions alone.

Figure 7 also clearly shows that a pure tradeoff function does not easily account for inconsistencies in viewpoint specifications. By smoothing all criteria satisfactions, it leads to difficulties in deciding where to precisely position the camera. Conversely, our aggregation tree model can express potential inconsistencies, thus providing means to define the region(s) where to position the camera and how to tweak camera parameters such as its focal length. Note that since we use a stochastic search algorithm, computational consistency is also an important feature to consider; though we do not yet compare results over multiple runs, future work should also focus on that aspect.

9. Conclusion

This paper identified two existing lacks of current viewpoint computation methods. We showed that the commonly used weighted sum function does not allow to properly evaluate the effective satisfaction of a composition problem. We proposed novel aggregation functions which provide means to

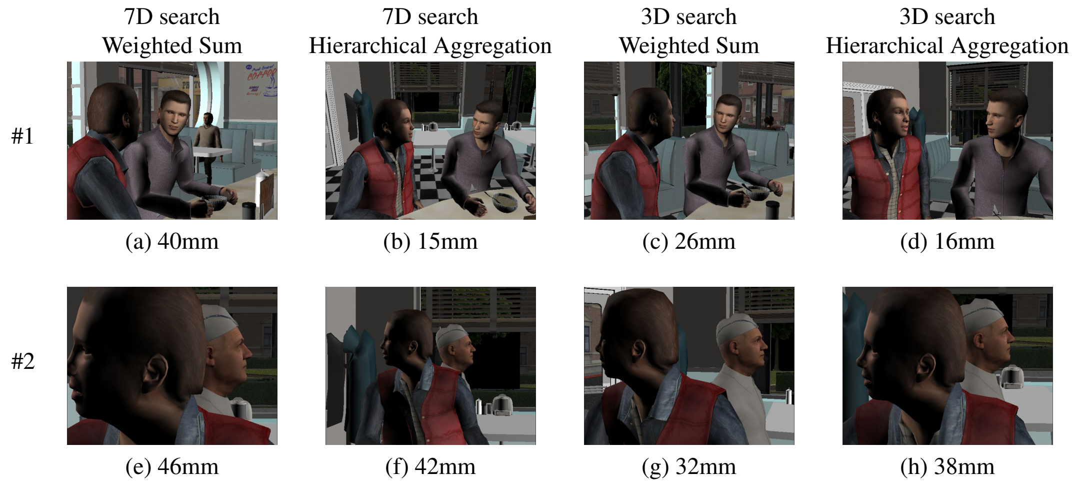


Figure 6: *Comparison between viewpoints obtained through a combination of a fitness function and a search method, for two viewpoint specifications (detailed in the results section). The impact of our two propositions (a hierarchical aggregation of objective functions, and a search reduced to camera positions, with its orientation and focal length optimized in a separate process) are compared to common practice (where the fitness function is computed as a weighted sum, and the search is done over all 7 camera degrees of freedom in a single process). Each viewpoint has been generated within a 5ms time window.*

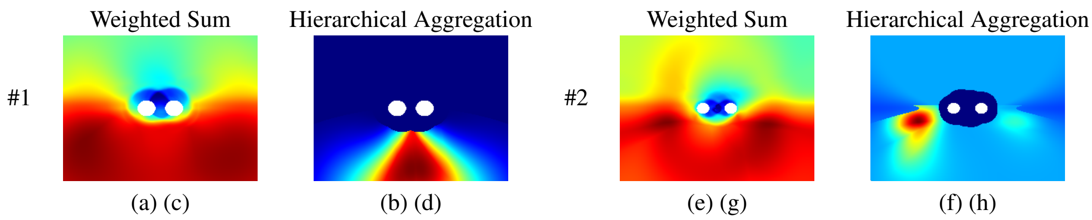


Figure 7: *Comparison between the fitness functions used for computing the two-target viewpoints; each viewpoint is defined as a combination of classical visual criteria (a screen position, a size, a vantage angle, and visibility requirements) on both target. Each problem comprises visual criteria that are inconsistent. It is clear that the pure tradeoff version has difficulties in properly expressing the overall problem satisfaction, conversely to our aggregation tree method.*

express evolved relationships between two or more criteria. We then proposed to build upon such functions to better express the fitness value of a viewpoint, and to construct the fitness function as an aggregation tree. We argued that considering the camera orientation and focal length independent from the position leads to difficulty in guiding the search process. We then proposed to reduce the search process to the space of camera positions, while separately constraining the framing through a separate optimization of the camera orientation and focal length. Our aggregation method opens perspectives to also consider continuity rules in the computation; such consideration would benefit to a wide a number of applications such as video games. As a future work, we also think that viewpoint computation tools could benefit from studying how expert cinematographers rank viewpoints with regards to all criteria of a composition problem. Particularly, one could focus on learning objective functions and aggregation trees from a set of annotated viewpoint examples.

References

[Bli88] Blinn J.: Where Am I? What Am I Looking At? *IEEE Computer Graphics and Applications 8*, 4 (July 1988). 6

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

[LC12] Lino C., Christie M.: Efficient Camera Composition for Virtual Camera Control. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2012). 6

[LC15] Lino C., Christie M.: Intuitive and Efficient Camera Control with the Toric Space. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2015* (2015). 7

[RCU10] Ranon R., Christie M., Urli T.: Accurately Measuring the Satisfaction of Visual Properties in Virtual Camera Control. In *Smart Graphics*, vol. 6133 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2010. 2

[RU14] Ranon R., Urli T.: Improving the efficiency of viewpoint composition. *IEEE Transactions on Visualization and Computer Graphics 20*, 5 (2014). 2, 7