Optimising for Multiple Shots: An Analysis of Solutions Diversity in Virtual Camera Composition

Paolo Burelli
Center For Computer Games Research
IT University of Copenhagen
Denmark
pabu@itu.dk

Mike Preuss
Computational Intelligence Group
Dept. of Computer Science
Technische Universität Dortmund, Germany
mike.preuss@tu-dortmund.de

Georgios N. Yannakakis
Center For Computer Games Research
IT University of Copenhagen
Denmark
yannakakis@itu.dk

ABSTRACT

Automatically generating cinematics in computer games could, not only increase the quality of the player experience, but also allow more cinematographic visualizations of the game for other audiences, such as game replays, machinima, audience views or game comics. In all these cases, the virtual camera controller is expected to find a high quality camera configuration, often in as little time as possible; however, in several cases, one optimal camera configuration is not sufficient and multiple diverse cameras are needed. In this paper, we show how the intrinsic shape of the the camera composition objective function contains multiple alternative solutions, often visually different, that can be used to generate multiple different shots from the same description. It is already known that from an optimization standpoint, these problems are usually multimodal so that algorithms with restart and/or niching components are needed as we may otherwise miss the best optima. Here, we are especially interested in exploring the problem landscapes and see how the difficulty level of finding multiple good shots varies across several stereotypical scenes.

Categories and Subject Descriptors

G.1.6 [**Mathematics of Computing**]: Optimization; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism

Keywords

Automatic Camera Control, Virtual Camera Composition, Fitness Landscape, Niching, Global Optimization

1. INTRODUCTION

There can be no doubt that in times of the growing importance of procedural content generation and crowdsourcing, automated camera control is an important support technique in games and other 3D applications. The task to find one or multiple good camera positions can be modelled as an optimisation problem [7] in which the search space consists of all possible camera configurations and high level properties are aggregated into an objective function that is to be optimised.

Although the space of possible camera configurations is relatively low dimensional (at least 5 dimensions to define position and orientation), automatic camera control is a complex optimisation problem for two reasons: the evaluation functions corresponding to frame properties often generate landscapes that are very rough for a search algorithm to explore [6]. Moreover, if the controller is used in a real-time and interactive context, the evaluation of the objective function is computationally expensive with respect to the time available for computation, significantly reducing the number of evaluations available for the search process. In general, these problems seem to be highly multimodal, but the degree of ruggedness and the number of basins may vary a lot across different instances [6, 13].

Additionally, there is often a need to obtain multiple alternative good solutions. For instance, Thawonmas et al. [15] point out how one single best solution is unsatisfactory when generating comics out of game dialogues. However, the same conclusion could be easily generalised to any situation in which a scene with little movement needs to be filmed. Instead of the simple cure suggested by Thawonmas et al. (randomizing the shot definition), we instead use the multimodal nature of the problem to generate several solutions with comparable quality. For this purpose, we propose an analysis of the landscape of the virtual camera composition problem, in which we investigate the distributions and the diversity of the acceptable solutions and we suggest a multi-objective measure of visual diversity.

Finally, considering that the problem of providing multiple alternative good solutions naturally calls for application of restart/niching methods, as also employed in [13]. Here, we gather further experimental evindence concerning which methods are most suitable for what kinds of landscapes. Note that the optimization algorithms work on an aggregated single objective function which shall work by the implicit assumption that two solutions that are located far from each other in the search space also differ in the constituents of the objective function.

In the reminder of the paper we describe the current state-of-the-art in virtual camera composition (Sec. 2), we analyse the presence of multiple solutions in a set of representative camera control problems (Sec. 3) and we showcase how multiple diverse solutions can be found using a niching and restart based evolutionary strategy (Sec. 5).

2. RELATED WORK

Since the introduction of virtual reality, virtual camera control attracted the attention of a large number of researchers [7]. Early studies on virtual camera [16] investigated manual camera control metaphors for exploration of virtual environments and manipulation of virtual objects. However, direct control of the several degrees of freedom of the camera showed often to be problematic for the user [8] leading researchers to investigate for the automation of camera control.

In 1988, Blinn [3] showcased one of the first examples of an automatic camera control system. Blinn designed a system to automatically generate views of planets in a NASA space simulator. Although limited in its expressiveness and flexibility, Blinn's work inspired many other researchers trying to investigate efficient solutions and more flexible mathematical models able to handle more complex aspects such as camera motion and frame composition [1].

More generic approaches model camera control as an optimisation problem by requiring the designer to define a set of targetted frame properties which are then put into an objective function. These properties describe how the frame should look like in terms of object size, visibility and positioning. Olivier et al. [11] first formalised the camera control problem as an optimisation problem and introduced detailed definition of these properties. Since then, numerous search strategies have been applied to solve the problem, including population based algorithms, local search algorithms and combinations of the two [7]. These approaches offer different performances with respect to computational cost, robustness and accuracy; however, none of them addresses the problem of finding multiple diverse solutions.

Thawonmas et al. [15] identify variety of shots as a major problem in automatic generation of cinematics and they introduce a roulette-wheel selection mechanism to force variety in shot descriptions. However, by altering the shot properties, this approach does not only vary the shot visual aspect but potentially changes the shot meaning.

We suggest that shot diversity can be achieved by exploiting the characteristics of the virtual camera composition objective function. We demonstrate our hypothesis through an analysis of a set of standard virtual camera composition problems and by showing that such diversity can be achieved using a restart and niching based evolutionary strategy.

3. VIRTUAL CAMERA COMPOSITION

An optimal camera configuration is defined as the combination of camera settings which maximises the satisfaction of the requirements imposed on the camera, known as camera profile. A camera profile describes the characteristics of the image that the camera should generate in terms of composition properties. Based on the author's previous work on automatic camera control [5], the properties that can be imposed are: *Object Visibility*, *Object Projection Size*, *Object View Angle* and *Object Frame Position*.

The *Object Visibility* property defines whether an object (or a part of it) should be visible in the frame: it is calculated by casting 5 rays to the same number of vertices of the object's mesh. The vertices are the top, the lowest, the leftmost and the rightmost vertices on screen and the one closets to the object's center of mass. The overall visibility value V is defined by the following formula:

$$V = \frac{N_{in}}{N_{tot}} * \frac{5 - R_h}{5} \tag{1}$$

where N_{tot} is the total number of vertices composing the mesh of the object, N_{in} the number of vertices contained in the view frustum, and R_h is the number of ray cast hits — i.e. number of occluded vertices. The *Object Projection Size* property defines the size an object should have in the frame and it is calculated as the fraction between the area included in the bounding rectangle defined by the top, the lowest, the leftmost and the rightmost vertices on screen and the screen area. The *Object View Angle* property defines the angle from which the camera should frame the object, while *Object Frame Position* property defines the position that the projected image of the object (its center) should have in the frame, where the upper left angle has coordinates [0,0] while the lower right one has coordinates [1,1].

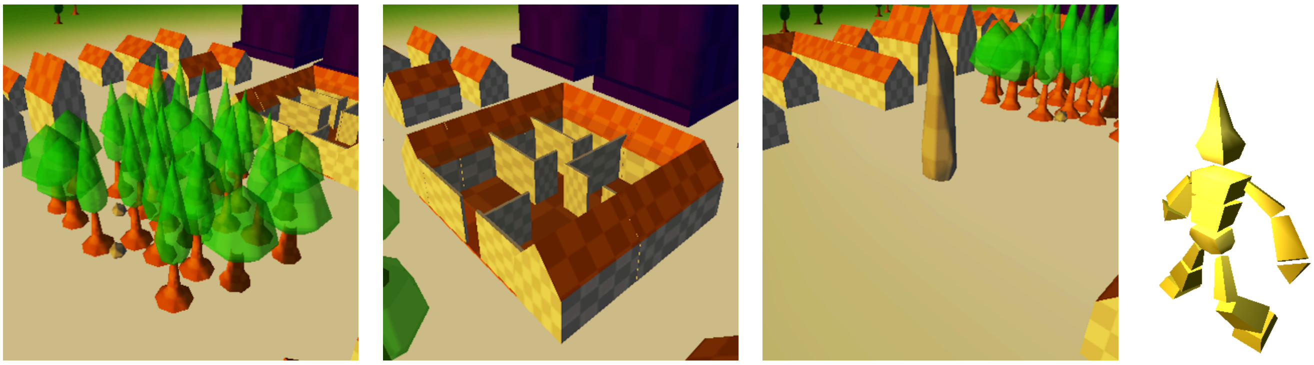
Each composition property corresponds to an objective function which describes the satisfaction of such property. The complete virtual camera composition objective function F is a linear combination of the objective functions corresponding to each property included in the camera profile and is is defined as follows:

$$\begin{aligned} F = &\sum_{i=1}^{Nv} Wv_i Fv_i + \sum_{i=1}^{Np} Wp_i Fp_i \\ &+ \sum_{i=1}^{Na} Wa_i Fa_i + \sum_{i=1}^{Nf} Wf_i Ff_i \end{aligned} \tag{2}$$

where Fv_i,Fp_i,Fa_i and Ff_i are respectively the objective function values of the i^{th} *Object Visibility* property, the i^{th} *Object Projection Size* property, the i^{th} *Object View Angle* property and the i^{th} *Object Frame Position* property; Nv, Np, Na and Nf are the numbers of properties imposed for property type; Wv, Wp, Wa and Wf are the weights of each property.

3.1 Test Problems

To analyse the distribution of the solutions in the camera composition optimisation problem and to assess the performance of the proposed solution we consider 3 test problems in three test environments. In each test problem, the camera controller has to frame a common game situation (e.g. a dialogue between virtual characters) according to a set of standard cinematographic visual properties. The test environments include a large variety of geometrical features of modern computer games such as closed rooms, walls or trees. The set of properties of the desired camera configurations and the virtual environments are designed to include a wide variety of optimisation challenges typical of the virtual camera composition problem such as lack of gradient or



(a) Forest environment (b) House environment (c) Square environment (d) Subject

Figure 1: Test problems' virtual environments and subject model.

multi-modality.

The test problems are set in three different environments: a forest (see Fig. 1(a)), a house (see Fig. 1(b)) and a square (see Fig. 1(c)). In each environment three different problems are tested with respectively one, two and three subjects (see Fig. 1(d)): a long shot of one subject, a side shot of two subjects facing each other and an over the shoulder shot of one subject facing two others.

The first problem contains three properties set on the subject: an *Object Visibility*, an *Object Projection Size* and an *Object View Angle*. The properties are parametrised so that the visibility should be full, the projection of the subject should occupy the whole screen and the objective should be facing the camera. In the second problem, the same three properties are imposed on each subject, the only difference being that the angle should be lateral. The last problem is based on the chat scene by Thawonmas et al. [15] and it includes three characters with one ideally chatting to the other two and should be framed with an over-the-shoulder shot. A a full *Object Visibility* property is imposed on all subjects, as well as an *Object Projection Size* and and *Object View Angle* properties; however, the parameters for these two properties differ. The two subjects standing side by side are expected to cover just one third of the screen and to be shot frontally, while the subject facing them should cover the screen completely and should be shot from his back. Moreover, the latter should also have a projection positioned in the lower right quadrant of the screen, at position [0.66,0.33].

Each problem is tested in three different environments, one indoor and two outdoor, with different geometrical characteristics. The forest environment is an outdoor virtual environment composed by a cluster of threes, the subjects are placed between these threes which act as partial as scattered obstacles. As displayed in Fig. 2(a), Fig. 2(b) and Fig. 2(c), such environment influences the objective function landscape by increasing the modality; this is mostly due to the fact that the three trunks are thin occluders which produce a slicing effect in the objective function landscape. The second environment analysed in this paper, the house, is an indoor environment with closed spaces separated by solid walls. As described in [6], walls act as large occluders inducing large areas of the objective function landscape to have little or no gradient. Figures 2(d), 2(e) and 2(f) display the aforementioned characteristic which is smoothed by the presence of other properties besides *Object Visibility* in the problem description. The last environment, the square, is the simplest one from a geometrical perspective. It is largely an empty space with one single obstacle placed in the center. All the test problems in this environment exhibit a mostly smooth objective function landscape with a lower number of modalities (see Fig. 2(g), Fig. 2(h) and Fig. 2(i)). While this environment does not appear particularly challenging in terms of optimisation complexity, it has been included in the study for completeness, and to analyse the solutions distribution also in the simplest cases.

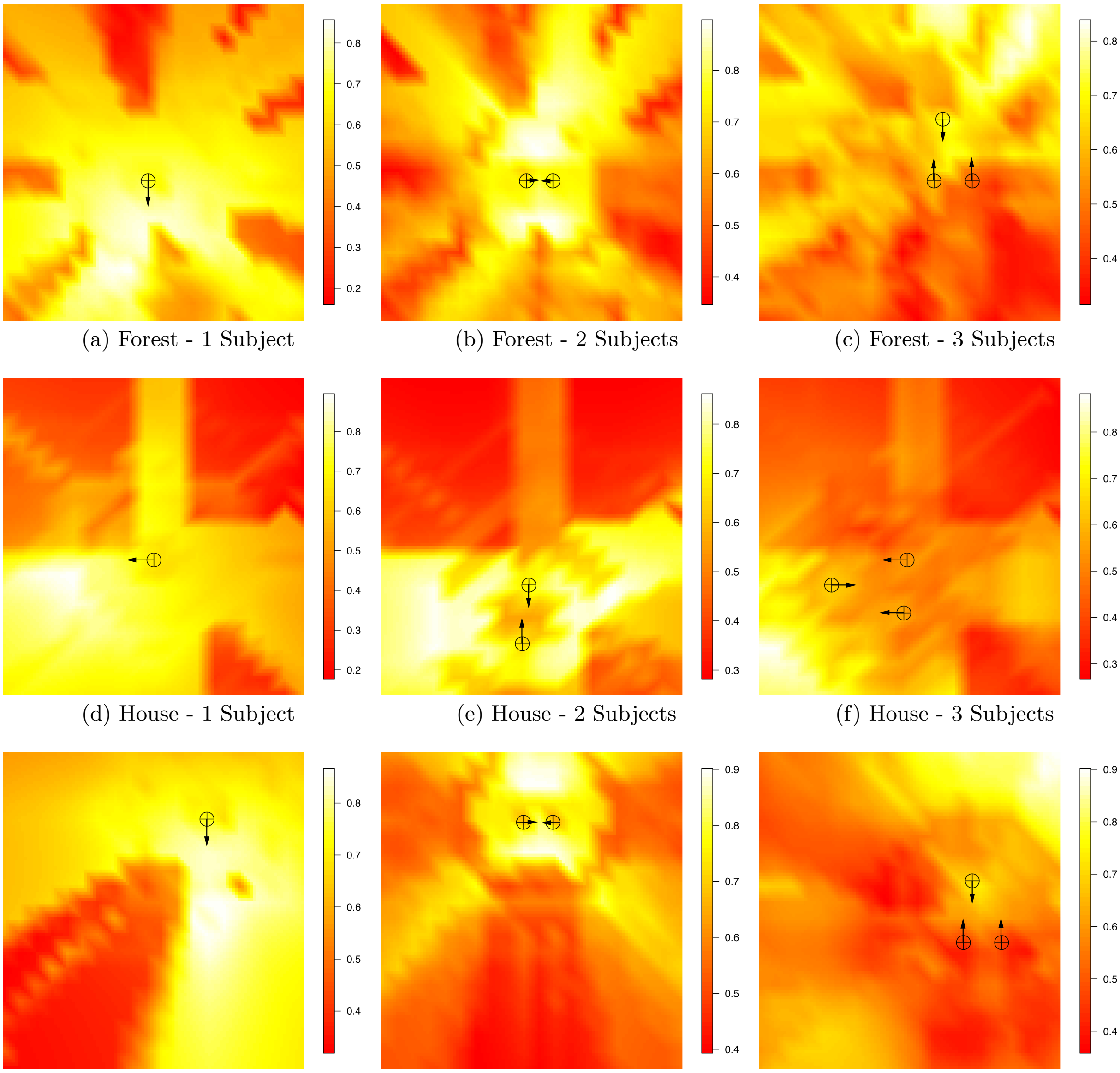
4. SOLUTIONS DIVERSITY

All the objective function landscapes displayed in Fig. 2 exhibit multiple modalities — i.e. multiple areas which contain either local or global optima — which might contain multiple alternative camera configurations. To analyse the arrangement and the number of these solutions we have first to define which solutions we consider valid, what are the characteristics that these solutions should have to be considered good alternatives and how we measure these characteristics.

For this analysis, we consider to be a valid solution any solution having an objective function value equal or greater than 0.95 times the value of the global optimum. The 5% threshold has been chosen, after multiple trials, to guarantee that the solutions considered have a quality not perceivably worse than the global optimum; moreover, all the test problems analysed have multiple solutions above this threshold. The solutions with an objective function value within the aforementioned range are all contained roughly in the white areas of the landscapes depicted in Fig. 2.

When looking for more than one solution to a composition problem a new measure becomes necessary: the solutions should appear diverse to be valuable; therefore, it is important to define numerically the concept of diversity. In the authors' first attempt to deal with shot diversity [13] the difference between two solution was defined as the euclidean distance between the two camera locations. However, while such a decision space distance gives an intuitive measure of difference, it does not guarantee or quantify the difference between the visual features of the two generated shots (nevertheless, we assume that there is a correlation between these two).

To be able to quantify these differences we need a numer-



(a) Forest - 1 Subject (b) Forest - 2 Subjects (c) Forest - 3 Subjects

(d) House - 1 Subject (e) House - 2 Subjects (f) House - 3 Subjects

(g) Square - 1 Subject (h) Square - 2 Subjects (i) Square - 3 Subjects

Figure 2: Maximum value of the problems' objective function sampled across the X and Z axis of the virtual test environments. The circles correspond to the position and orientation of the subjects in the virtual environment.

ical measure of the distance between the shots rather than the cameras placements. For this purpose, in this paper we attempt to measure the difference between shots within the objective function space instead of the search space. This is the vectorial space defined by the satisfaction values of the properties composing the overall objective function – i.e. we decompose the virtual camera composition problem into a multi-objective optimisation and we measure diversity as the euclidean distance within the space defined this way. The distance value between two solutions a and b, $D(a, b)$, is defined as follows:

$$D(a,b) = \frac{\sqrt{\sum_{i=1}^{k}(f_i(a) - f_i(b))^2}}{\sqrt{k}} \qquad (3)$$

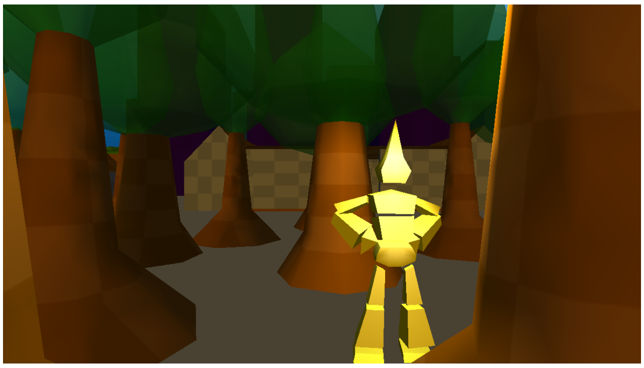
where k is the number of properties of the problem and $f_i(x)$ is the weighted satisfaction value of the i^{th} property of the problem for solution x. The distance function is divided by the square root of the number of properties, to make the measure independent to the value of k and normalised between 0 and 1.

Using this measure it is possible to analyse the nine test problems' landscapes from a solutions diversity perspective. The ideal solutions to the problem should have a minimal objective function difference with the global optimum — i.e. the overall quality should similar — but should be as distant as possible to the global optimum in the multi-objective space — i.e. the shots generated should have different visual features. Identifying the best solutions is, therefore, itself a multi-objective problem, in which the objective function difference has to be minimised, while the objective space distance has to be maximised. Figure 4 shows the solutions within the previously defined validity range for each test problems sorted by quality (objective function difference) and diversity (multi objective space distance). The most interesting solutions, highlighted in red color, are the ones belonging to the Pareto front, which are, therefore, non dominated by any other solution to the problem.

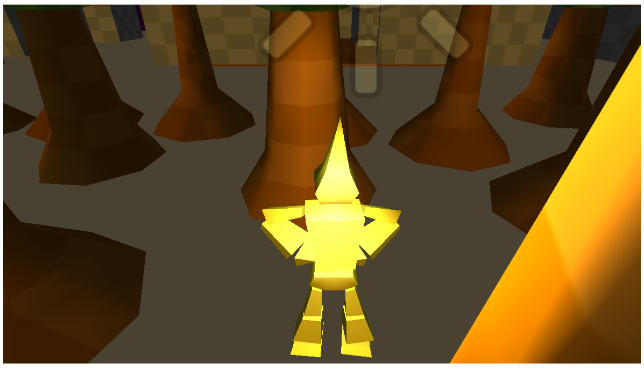
It is evident from the figures that each considered problem contains at least three solutions belonging to the Pareto front, with objective space distances up to 38% of the max distance. Moreover, these values have been obtained with a 32000 points sampling of the solutions pace; therefore we can expect that with a more refined sampling more Pareto solutions could be found for each problem. An example of two solutions belonging to the Pareto front is depicted in Fig. 4(b), the two shots are respectively the optimal solution to the long shot problem in the forest environment and and alternative solution with an overall quality only lower by 1.8% to the optimum but an objective space distance equal to 0.303.

It is interesting to notice how, even in a relatively simple composition problem such as a long shot, it is possible to find multiple alternative solutions with a significant visual difference. This would probably not be possible in a properly designed scene, where no obstacles compromise the shot; however, it is a common situation if the scene to be shot comes from a game or any other situation where staging is impossible or and ideal staging is not feasible.

We believe that, by exploiting the structure of the problem, it is possible to design a camera controller able to track multiple optimal solutions and, in the reminder of the paper, we show how this is possible using an evolutionary strategy employing niching and restart.



(a) Optimal solution



(b) Alternative optimal solution from the Pareto front

Figure 4: An example of two solutions belonging to the Pareto front of optimal solutions of the long shot in the forest environment. The objective space distance between the two solutions equals to 0.303 **while the difference between the two objective objective function values is** 0.018

5. FINDING MULTIPLE SOLUTIONS

As the problem at hand is obviously multimodal (see fig. 2), it would be dangerous to rely on a simple one-shot convergence process even if we wanted to find only one very good shot position. However, we now strive for detecting several good solutions quickly which makes it even more important to stop and restart as soon as possible. In [13], we have investigated how the CMA-ES without population increasing heuristic [1] and a CMA-ES derived niching method, NEA2, perform on some camera positioning problems. Instead of diversity of solutions in search space, we are now interested in diversity in objective space, but we presume that these methods are still doing much better than standard variants of Particle Swarm Optimisation (PSO) [10], Differential Evolution (DE) [14], and Sliding Octree (SO) [4] (which were also in the set of algorithms in the previous comparison). The only derivation of the default parameters we make for CMA-ES and NEA2 modification is the `TolFun` stopping criterion which is highly connected to the desired accuracy [9]. This is set to a value of 10^{-3} which is still below the needed accuracy. The effect of this setting is that fruitless searches in local optima are stopped earlier, thus more restarts can be done. Obviously, the change does not affect PSO, DE and SO, as they do not use restarts. They shall profit from doing so, but the many of the standard CMA-ES stopping criteria cannot easily be transferred to these methods.

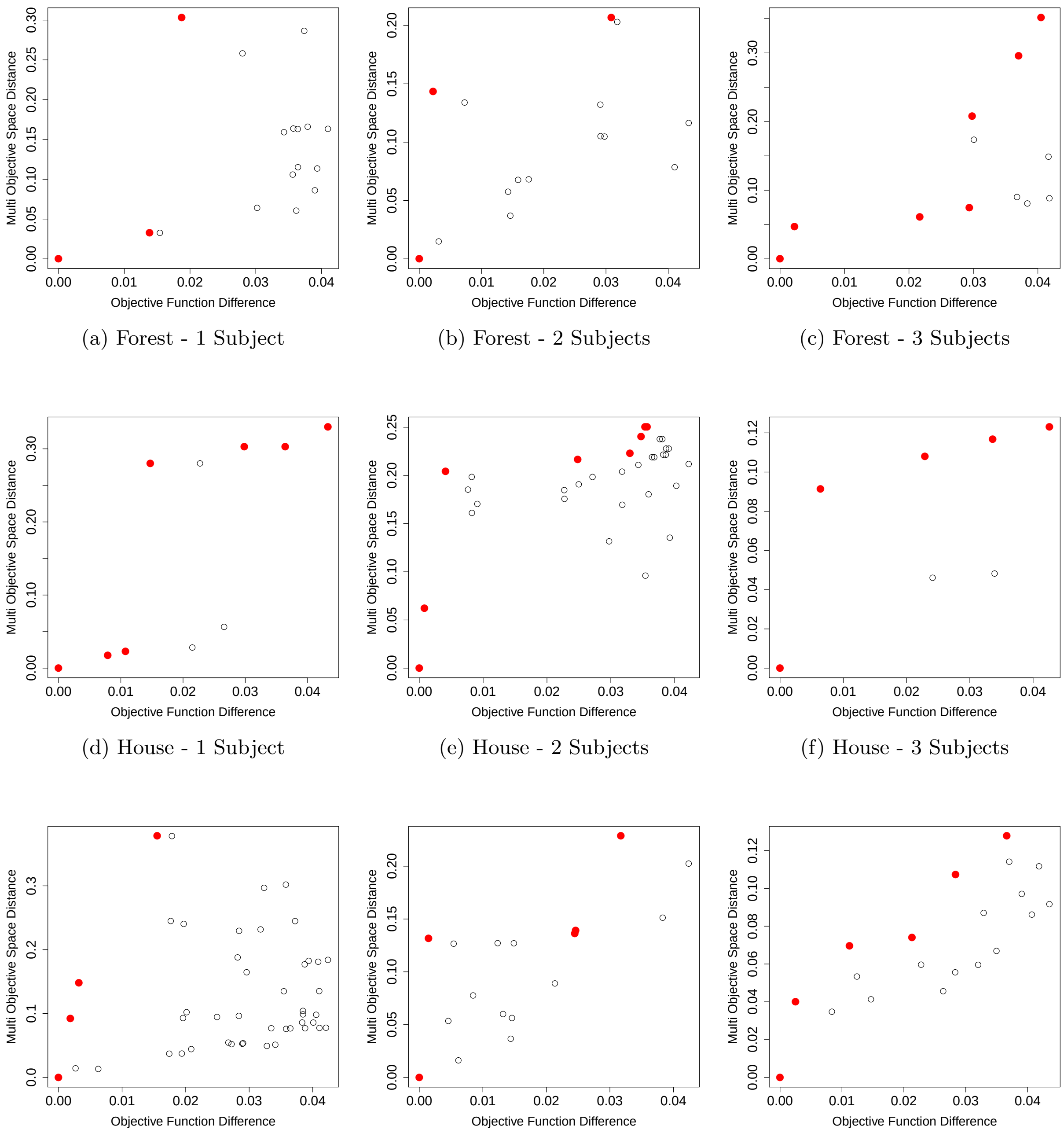
In [13], CMA-ES and NEA2 performed head-to-head, with a slight advantage of CMA-ES on the simpler problems and NEA2 on the more complex problems. This makes sense because the niching method uses an initial scan (with 200 samples in 5D) to obtain a partitioning of the search space in several basins of attraction, and uses this information to prevent multiple searches in the same basin. A typical CMA-ES run uses about 500 to 1000 evaluations in 5D before the first restart is initiated. Thus, NEA2 can only be quicker than the CMA-ES if the landscape is considerably complex.

5.1 Performance Measures

As described in section 4, we are interested in solutions with less than 5% quality difference to the presumed optimum, which is determined with the 32000 samples scan used already in section 4. The expected time to reach the desired quality for the first time is computed over several repeated runs after the *expected runtime* (ERT)[2] definition suggested in [2], with $\#fevals$ being the sum of all eval-

[1]Nikolaus Hansen - The cma evolution strategy: A tutorial. Version of June 28, 2011

[2]The term may be misleading as it is defined in evaluations, for absolute times it has to be multiplied with 16 ms.



(a) Forest - 1 Subject (b) Forest - 2 Subjects (c) Forest - 3 Subjects

(d) House - 1 Subject (e) House - 2 Subjects (f) House - 3 Subjects

(g) Square - 1 Subject (h) Square - 2 Subjects (i) Square - 3 Subjects

Figure 3: Optimal solutions (value > 0.95 * global optimum) of each test problems sorted by multi-objective space distance to the global optimum and objective function value difference with the global optimum. The red dots represent the solution belonging to the Pareto front. The plots have been generated by sampling each test problem linearly with approximately 32000 samples.

uations that were spend before reaching the target value f_{target}, and $\#succ$ standing for the number of successful runs:

$$ERT = \frac{\#fevals > f_{target}}{\#succ} \tag{4}$$

The next important task is to find a solution of similar quality but with a large objective space distance as measured by equation (3). If a first good solution was found, we scan the remainder of the run (until it finally ends after 5000 evaluations), detect the most diverse good solution, and also compute its ERT value.

5.2 Experiment

With the following experiment we want to find out which of the suggested algorithms, CMA-ES, NEA2, PSO, DE or SO is capable of reliably reaching good fitness values at least once and then generate a second good but diverse solution. Standard variants of DE, PSO and SO methods have been included as representatives of previous approaches to the problem.

Setup.

We run each algorithm on each of the nine problem instances (scenarios) 20 times for 5000 evaluations. Performance is measured as given in sec. 5.1. All parameters are kept at default values, except for the `TolFun` stopping criterion (applying to CMA-ES, NEA2) which is set to 10^{-3}. Default values for NEA2 are provided in [12]. The start population is determined randomly for the CMA-ES based methods and the stepsize start value is set to 0.15 in the normalized parameter space $[0, 1]$.

Task.

Taking the obtained measures of the algorithms on the different problems into account, we want to determine which algorithms reliably find good solutions quickly and also provides a second good, diverse solution in time (before the end of the run on average). Additionally, we look for relations between the type of problem (Forest, House, Square, 1, 2, and 3 subjects) and the algorithm performance.

Results/Visualization.

Table 1 shows the averaged results over the different problems, with ERT1 giving the expected runtime for reaching the first good solution, ERT2 the expected runtime for obtaining the most diverse good solution, and $\varnothing$ dist(1,2) standing for the objective space diversity between these two, averaged over all 20 runs (assuming 0.0 where no second solution was found, and an ERT of ∞ meaning that no first or second solution was obtained in all 20 runs, respectively).

Observations.

We first note that the only 2 algorithms that obtain 2 solutions reliably in all cases are CMA-ES and NEA2. DE and PSO do so often, but not always, and SO is always the weakest algorithm and sometimes fails to find a solution in any of the 20 runs. The average distances between the first and the most diverse solution are in all cases the highest for CMA-ES and NEA2 (see Fig. 5 for a sample of two solutions found by NEA2), and usually much smaller for the other methods. When considering the best 2 algorithms, it seems that the 1 subject scenarios are always the easiest to solve. For the forest and the square problems, the 2 subject scenarios are the most difficult, followed by the 3 subject scenarios. This is different for the house scenarios, where the order is 1 subject, 2 subjects, 3 subjects. Overall, the ERT1 and ERT2 values are only slightly increase for the more difficult scenarios (at least for CMA-ES and NEA2).

Discussion.

It is no surprise that CMA-ES and NEA2 do not differ much in performance, so that one may rely on CMA-ES as default method. The only case where NEA2 is clearly better is the square with 2 subjects. Taking the search space structure of the problems as depicted in figure 2 into account, this may stem from the difficulty to separate the two good areas above and below the subjects, which are relatively small and close to each other. However, this is difficult to say as the figures plot the search and not the objective space diversity — these two are presumably related, but the objective space may add some difficulties we cannot see in the pictures. As the objective space distances depend on both points, they cannot be plotted in 2D. The house 2 problem may be easier because it provides two much larger good regions (at least in the search space, see figure 2).

Additonally, we have to take into account that the algorithms are asked to deliver diverse solutions in a space they cannot see, they act only on the single aggregated objective value. This could be cured by either trying a multi-objective approach or with some new technique direct the restarts into regions of high objective space distance to the already found good solution (some sort of objective space niching).

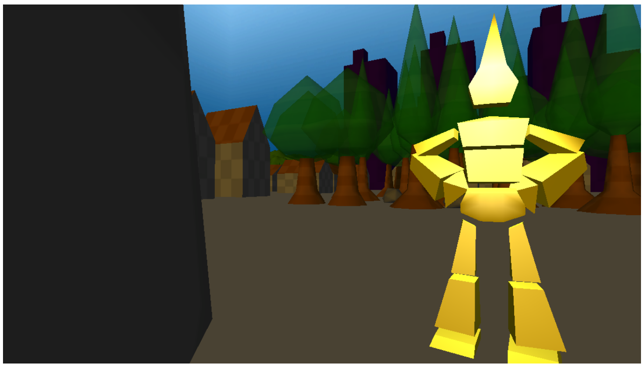
Table 1: Averaged performance of the 5 algorithms

Algorithm	ERT1	ERT2	$\varnothing$ dist(1,2)	Problem
pso	1580	4091	0.16	Forest - 1 Subs.
cma-es	471	2515	0.28	
de	1599	2533	0.07	
so	2929	3176	0.08	
nea2	636	2592	0.28	
pso	211	1925	0.22	Forest - 2 Subs.
cma-es	961	2812	0.24	
de	546	1850	0.20	
so	6356	6858	0.06	
nea2	1152	2663	0.24	
pso	8438	10245	0.08	Forest - 3 Subs.
cma-es	598	2398	0.35	
de	998	1928	0.27	
so	∞	∞	0.00	
nea2	763	2718	0.36	
pso	3973	7374	0.11	House - 1 Subs.
cma-es	498	2514	0.29	
de	20927	23778	0.02	
so	12281	12537	0.00	
nea2	764	2455	0.26	
pso	754	2849	0.25	House - 2 Subs.
cma-es	623	2849	0.28	
de	7033	8590	0.08	
so	∞	∞	0.00	
nea2	819	2465	0.29	
pso	97002	99440	0.01	House - 3 Subs.
cma-es	814	2640	0.23	
de	∞	∞	0.00	
so	∞	∞	0.00	
nea2	1016	2774	0.20	
pso	505	1543	0.30	Square - 1 Subs.
cma-es	476	2464	0.34	
de	470	1685	0.28	
so	2252	2558	0.06	
nea2	562	2646	0.35	
pso	1966	4199	0.14	Square - 2 Subs.
cma-es	1900	3440	0.20	
de	1393	2685	0.18	
so	28518	28730	0.01	
nea2	1392	2460	0.20	
pso	95407	97956	0.01	Square - 3 Subs.
cma-es	778	2650	0.23	
de	1217	1890	0.17	
so	∞	∞	0.00	
nea2	1097	2838	0.20	

6. SUMMARY AND CONCLUSIONS

This paper proposes an approach to shot diversity in virtual camera composition based on the characteristics of composition as an optimisation problem. We show that, in a wide range of examples, covering different composition

(a) First solution



(b) Second solution

Figure 5: An example of two solutions found by NEA2 fo a one subject problem set in the square environment. The solutions have objective space distance equal to 0.28

problems and different types of virtual environments, it is possible to find multiple valid camera configurations optimising the problem. Although, with different magnitude, each problem exhibits exhibits solutions which are different in terms of visual characteristics but yet similar in quality. Moreover, we demonstrate that such solutions can be found using a niching and restart based evolutionary algorithm For this purpose, we present an experiment in which we compare the behaviour of NEA2, a novel niching evolutionary algorithm, against a set of state-of-the-art algorithms for optimisation of virtual camera composition on the task of finding two different valid solutions on nine test problems.

The analysis of the test problems confirms the presence of multiple valid solutions for each problem, while theresults of the comparison reveal that NEA2, as well as a baseline CMA-ES, are eable to find multiple diverse solutions, systematically in all the test problems. Such multiple solution hold a potentially valuable information also in real-time camera control and animation, as these multiple cameras could be used to perform real-time cinematographic cuts or as backup camera solutions. However, to evaluate the applicability of such approach, it would be necessary to evaluate how the optimisation problem changes during time in a dynamic environment.

Finally, we believe that a further investigation of the correlation between the multi-objective space distance and the search space distance, would reveal more valuable information about the topology of the solutions, helping the development of more accurate and efficient algorithms.

7. REFERENCES

[1] Daniel Arijon. *Grammar of the Film Language.* Silman-James Press LA, 1991.

[2] Anne Auger and Nikolaus Hansen. A restart cma evolution strategy with increasing population size. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, 2-4 September 2005, Edinburgh, UK*, pages 1769–1776. IEEE press, 2005.

[3] James Blinn. Where Am I? What Am I Looking At? *IEEE Computer Graphics and Applications*, 8(4):76–81, 1988.

[4] Owen Bourne, Abdul Sattar, and Scott Goodwin. A Constraint-Based Autonomous 3D Camera System. *Journal of Constraints*, 13(1-2):180–205, 2008.

[5] Paolo Burelli and Georgios N. Yannakakis. Combining Local and Global Optimisation for Virtual Camera Control. In *IEEE Conference on Computational Intelligence and Games*, page 403, 2010.

[6] Paolo Burelli and Georgios N. Yannakakis. Global Search for Occlusion Minimisation in Virtual Camera Control. In *IEEE Congress on Evolutionary Computation*, pages 1–8, Barcelona, 2010. IEEE.

[7] Marc Christie, Patrick Olivier, and Jean-Marie Normand. Camera Control in Computer Graphics. In *Computer Graphics Forum*, volume 27, pages 2197–2218, 2008.

[8] Steven M. Drucker and David Zeltzer. Intelligent camera control in a virtual environment. In *Graphics Interface*, pages 190–199, 1994.

[9] Nikolaus Hansen. The cma evolution strategy: A tutorial. version of June 28, 2011.

[10] James Kennedy and Russell C. Eberhart. Particle swarm optimization. In *IEEE Conference on Neural Networks*, pages 1942–1948, 1995.

[11] Patrick Olivier, Nicolas Halper, Jonathan Pickering, and Pamela Luna. Visual Composition as Optimisation. In *Artificial Intelligence and Simulation of Behaviour*, 1999.

[12] Mike Preuss. Niching the cma-es via nearest-better clustering. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, GECCO '10, pages 1711–1718. ACM, 2010.

[13] Burelli Paolo Preuss Mike and Georgios Yannakakis. Diversified Virtual Camera Composition. In *European Conference on the Applications of Evolutionary Computation*, Malaga, 2012.

[14] Rainer Storn and Kenneth Price. Differential Evolution âĂŞ A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[15] Ruck Thawonmas, Ko Oda, and Tomonori Shuda. Rule-Based Camerawork Controller for Automatic Comic Generation from Game Log. In *IFIP International Conference on Entertainment Computing*, pages 326–333, Seoul, 2010.

[16] Colin Ware and Steven Osborne. Exploration and virtual camera control in virtual three dimensional environments. *ACM SIGGRAPH*, 24(2):175–183, 1990.