Diversified Virtual Camera Composition

Mike Preuss, Paolo Burelli, Georgios N. Yannakakis

Computational Intelligence Group, Dept. of Computer Science
Technische Universität Dortmund, Germany and
Center For Computer Games Research, IT University of Copenhagen, Denmark
Email: mike.preuss@tu-dortmund.de, pabu@itu.dk, yannakakis@itu.dk

Abstract. The expressive use of virtual cameras and the automatic generation of cinematics within 3D environments shows potential to extend the communicative power of films into games and virtual worlds. In this paper we present a novel solution to the problem of virtual camera composition based on niching and restart evolutionary algorithms that addresses the problem of diversity in shot generation by simultaneously identifying multiple valid camera camera configurations. We asses the performance of the proposed solution against a set of state-of-the-art algorithms in virtual camera optimisation.

1 Introduction

In computer games, as well as in most 3D applications, effective camera placement is fundamental for the user to understand the virtual environment and be able to interact. Camera settings for games are usually directly controlled by the player or statically predefined by designers. Direct control of the camera by the player increases the complexity of the interaction and reduces the designer's ability to control game storytelling (e.g. the player might manually look at an object revealing an unwanted information). Statically defined cameras, on the other hand, release the player from the burden of controlling the point of view, but often fail to correctly frame the game actions. Moreover, when the game content is procedurally generated, the designer might not have the necessary information to define, a priori, the camera positions and movements.

Automatic camera control aims to define an abstraction layer that permits the designers to instruct the camera with high-level and environment-independent rules. The camera controller should dynamically and effectively translate these rules into camera movements. Most researchers model this problem as an optimisation problem [8] in which the search space is the space of all the possible camera configurations and high level properties are modelled as an objective function 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 [7] and the evaluation of such properties is computationally expensive with respect to the time available for computation (16ms for real time applications), 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 [7].

To the authors knowledge, all the research carried out to solve this optimisation problem focuses on providing more accurate, robust and efficient algorithms to find the best possible camera configuration given the objective function defined by the designer's requirements. However, as pointed out by Thawonmas et al. [22], one single best solution is often unsatisfactory. When filming a scene with little movement, such as a dialogue, selecting always the same solution will lead to a repetitive direction. While this might be the explicit will of the designer, it is often an issue for media such as films and comics. Thawonmas et al. address this problem by randomizing the shot definition; such a solution, however, acts on the design of the shot rather than on the implementation, potentially disrupting the intended message. We consider the problem of providing multiple alternative good solutions as largely unsolved, and it naturally calls for application of niching methods because they are designed for providing more than one solution. However, as the *black box optimization benchmark* (BBOB) competitions[1] at GECCO 2009 and 2010 conferences have shown (see [2] for data and [13] for a comprehensive analysis and summary), the CMA-ES also copes well with multimodal functions due to its clever restart mechanisms and naturally, each restart delivers an approximation for a local optimum. Consequently, we intend to pursue the following tasks in this work:

a) Assess if modern evolutionary algorithm approaches as niching and restart based variants of the CMA-ES [10] are capable of reliably providing multiple diverse good solutions to the problem quickly;
b) investigate the trade-off between diversity and quality (in solutions) by setting up specific performance criteria and comparing our suggested methods with different state-of-the-art ones;
c) collect some (experimentally based) knowledge about the landscape structure, following the idea of *exploratory landscape analysis* (ELA) [14], in order to allow for even faster future algorithm implementations.

Our approaches exploit the multi-modal nature of the camera optimisation problem and identify multiple alternative solutions basins, thereby also revealing much information about the fitness landscape of the problem. Each basin contains potentially optimal camera configurations that have comparable fitness, but different visual aspect; such configurations can be used to diversify the shots while maintaining the designers requirements. However, in order to correctly estimate the suitability of the different algorithms, we make several simplifications that have to be rethought when applying them under real-time conditions:

a) We relax the runtime limit by allowing longer runs than would be possible in 16ms. This follows the *make it run first, then make it run fast* principle. Once good methods are found, they can be further adjusted to the problem to increase performance.
b) For now, we ignore the multi-objective nature of the problem as this will most likely make it even harder. This must be considered later on when already challenging single-objective formulation is solved sufficiently.

[1] `http://coco.gforge.inria.fr/doku.php`

In the remaining of the paper we describe the current state-of-the-art in virtual camera composition, we present our algorithmic approaches and showcase their capabilities and performance in a set of test environments.

2 Related Work

Since the introduction of virtual reality, virtual camera control attracted the attention of a large number of researchers [8]. Early studies on virtual camera [23] 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 [9] leading researchers to investigate for the automation of camera control.

In 1988, Blinn [4] 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. [15] 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 [8]. These approaches offer different performances with respect to computational cost, robustness and accuracy; however, none of them regards diversity of solutions as a key characteristic.

Thawonmas et al. [22] 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 propose the application of niching and restart evolutionary algorithms based on the real-valued blackbox optimization method CMA-ES to the virtual camera composition problem, to find multiple alternative solutions during the optimisation process and we showcase its performance with respect of a selection of state-of-the-art algorithms.

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 [6], the properties that can be imposed are: *Object Visibility*, *Object Projection Size*, *Object View Angle* and *Object Frame Position*. The

(a) Eavesdropping **(b)** Ambush **(c)** Chat

Fig. 1: Test problems' virtual environments.

first property defines whether an object (or a part of it) should be visible in the frame, the second defines the size an object should have in the frame, the third one defines the angle from which the camera should frame the object and the fourth one defines the position that the projected image of the object should have in the frame.

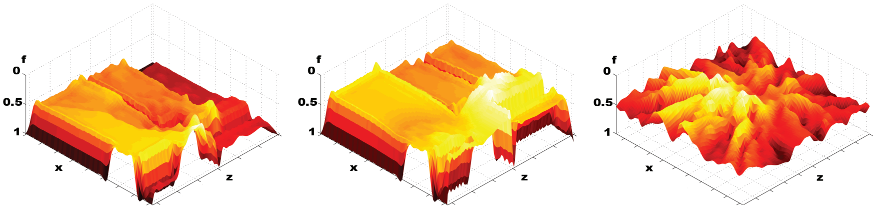
Each composition property corresponds to an objective function which describes the satisfaction of such property. The complete virtual camera composition objective function F is a linear combination of the objective functions corresponding to each property included in the camera profile.

3.1 Test Problems

In order to assess the performance of the proposed solutions we compare their convergence behaviour with a set of state-of-the-art algorithms across three test problems. Each test problem requires the camera to frame a common game situation (e.g. a dialogue between virtual characters) according to a set of standard cinematographic visual properties. The problems are set in a virtual 3D environment including 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 configuration and the virtual environments are designed to include alls the typical optimisation challenges of the virtual camera composition problem such as lack of gradient or multi-modality.

In the first problem (Fig. 1a) the environment includes three characters, with two of them facing each other and ideally chatting, while the third one eavesdropping. The properties for this problem include full visibility of all characters and a projection size equal to one third of the screen for all characters. In the second problem (Fig. 1b) the environment includes two characters on two sides of a wall. The properties for this problem include full visibility of all characters and a projection size equal to half of the screen for all characters and an horizontal angle of 90 degrees to the right of each character. The last problem is based on the chat scene by Thawonmas et al. [22] and it includes three characters with one ideally chatting to the other two. The visibility and projection size properties are equal to the ones in the first problem but the camera is also expected to be on the back of the listening characters.

The first and the second problems are set in an indoor environment with closed spaces separated by solid walls. As described in [7], walls act as large occluders induc-



(a) Eavesdropping **(b)** Ambush **(c)** Chat

Fig. 2: Maximum value of the problems' objective function sampled across the X and Z axis of the virtual test environments.

ing large areas of the objective function landscape to have little or no gradient. Figures 2a and 2b display the aforementioned characteristic which are smoothed by the presence of other properties besides visibility in the problem description. The third problem is set in an outdoor environment composed by a cluster of trees. As displayed in Fig 2c, such environment influences the objective function landscape by increasing the modality.

4 Niching and Restart CMA-ES Variants Under Test

Niching in evolutionary optimization dates back at least to the 1970s with the suggestion of Sharing and Crowding. Its general idea is that by organizing the search process and keeping several populations/local searches separate from each other we can obtain multiple good solutions at once, which is not that far from the scheme of modern real-valued memetic seach algorithms. In the biological protoype (Earth), niching works well because the surface on which most lifeforms move around is only 2 dimensional. However, in optimization, we usually have a larger number of dimensions, so that human intuition can get very wrong about distances, relative positions and volume sizes and the principles of geometry get less and less applicable. The test case we have here is interesting, as its 5 dimensions place it somewhere between 'well suited' (2D) and 'not applicable' ($>$ 20D) with respect to niching algorithms. The number of available niching algorithms is quite large, recent suggestions include e.g. [19], [20], [16], and we by no means claim that we are able to select the most appropriate niching EA (this would hardly be possible without knowing much more about the problem properties).

We therefore resolve to an algorithm that is a further development of [16] which is to date the only niching method with documented results on the BBOB test set. We call the original version (also labelled as NBC-CMA) *niching evolutionary algorithm 1* (NEA1) here to differentiate it from the newer version we term NEA2. NEA1 highly relies on the CMA-ES as local searcher, but uses a much larger starting population ($40\times$D) on which the nearest-better clustering method is run to separate it into populations representing different basins of attraction [18]. This topological clustering method connects every search point in the population to the nearest one that is better and cuts the connections that are longer than $2\times$ the average connection. The remaining connections determine the found clusters by computing the weakly connected components. This works very well for a reasonably large population in two dimensions, but increasingly fails if the number of dimensions increases. Therefore, in NEA2, a second additional cutting rule has been implemented: For all search points that have at least 3 incoming connections (it is the nearest better point for at least 3 others), we divide the length of its own nearest-better connection (in case it has none it is the best point and has surely been treated by the old rule) by the median of its incoming connections. If this is larger than a precomputed correction factor, the outgoing connection is cut (and we have one additional cluster). The correction factor cf has been experimentally derived and depends on D and the population size $\#elems$. This works astonishingly well for up to around 20D and not too complex landscapes.

$$\begin{aligned} cf = &-4.69 * 10^{-4} * D^2 + 0.0263 * D + 3.66/D - 0.457 * log10(\#elems) \\ &+7.51e-4 * D^2 - 0.0421 * D - 2.26/D + 1.83 \end{aligned} \tag{1}$$

As both cutting rules are heuristics that work well in many cases but come without guarantee, the number of resulting clusters had to be limited in NEA1, as it processes all clusters as separate CMA-ES populations in parallel. This can result in very long runtimes in cases where the clustering was not very accurate. NEA2 overcomes this problem by switching from a BFS-like to a DFS-like search in which the clusters are treated sequentially sorted according to their best members (best first, see [17] for details). Should the problem be less multimodal then detected, (e.g. unimodal), NEA2 would perform very similar to the CMA-ES as every start point leads to the same optimum. Although these niching methods are still much simpler than many other ones suggested in literature, they are arguably still much more complex than a restart CMA-ES.

However, there is a much simpler way to cope with organizing the search, and that is by just randomly chosing a new starting position as soon as stagnation is detected. Of course, this does not require to compare positions in search space and should work especially well in higher dimensions, when the geometry-based niching must fail. The CMA-ES does just that and is currently one of the leading algorithms in real-valued black-box optimization. We therefore add it to the algorithm test set, as a reference and reliable default solution. As the problem is highly time-critical and thus only very few evaluations are allowed, the CMA-ES is run without heuristic population enlargement as e.g. proposed with the IPOP- [3] and BIPOP-CMA variants. All parameters are left at their default values with the exception of the `TolFun` stopping criterion which is highly connected to the desired accuracy [11]. 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. As the NEA2 internally also heavily relies on the CMA and its stopping criteria, it is also affected by this change.

5 Experimental Analysis

5.1 Measures

Next to the raw performance (best obtained objective value over time), we measure the diversity target by first defining the properties of one/multiple suitable solutions. A solution is considered good enough if its fitness value is ≤ 0.05 (fitness values range from 0 to 1). This is an ad-hoc definition, but the first test runs told us that this quality can be achieved for all 3 problem instances. The motivation for 0.05 is that for a human, it will be hard to discriminate these solutions from the one with 0 values, thus they can be considered good enough for the practical application.

It is somewhat harder to determine when several good solutions are useful (this would not be the case if they are too similar). For discriminating useful alternatives, we demand a minimal Euclidean distance of at least 1 in the three spatial coordinates, regardless of the camera angles. The expected time to reach the desired quality is computed over several repeated runs after the *expected runtime* (ERT)[2] definition suggested in [3], with $\#fevals$ being the sum of all evaluations that were spend before reaching the target value $f_{target} = 0.05$, and $\#succ$ standing for the number of successful runs:

$$ERT = \frac{\#fevals > f_{target}}{\#succ} \tag{2}$$

As we desire several good solutions, we denote the ERT for the first one by ERT1, and the running times for the next ones (that have to fulfill the distance criterion concerning all the already detected ones) as ERT2 and ERT3, respectively. The diversity of the attained solutions is also of interest and it is measured by taking the average over the distance sums from the first solution to every other solution per run.

5.2 Experiment

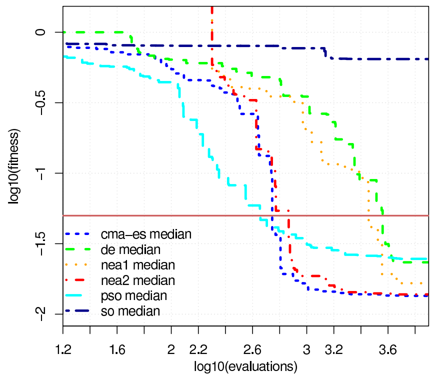
With the following experiment we want to find out which of the suggested algorithms, CMA-ES, NEA2, Particle Swarm Optimisation (PSO) [12], Differential Evolution (DE) [21], or Sliding Octree (SO) [5] is capable of reliably delivering multiple, diverse and good solutions quickly, and to pursue the goals named in the introduction. NEA1 is only added for a performance comparison to NEA2. Standard variants of DE, PSO and SO methods have been included as representatives of previous approaches to the problem.

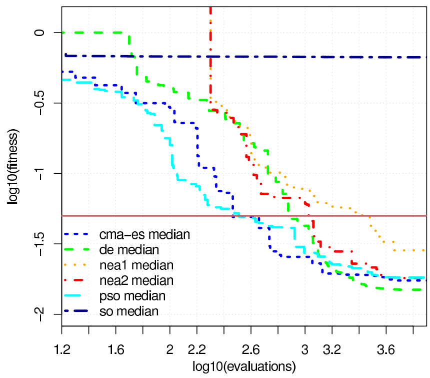
Pre-experimental planning During the first test runs, we found that a run length of 5000 evaluations is usually enough for the algorithms to converge to (best) solutions of around 0.05 or better.

Setup We run each algorithm on each of the three 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 and NEA1) which is set to 10^{-3}. Default values for NEA2 resemble the ones of NEA1, given in [16]. 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 $[0, 1]$.

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

divers.	ERT1	ERT2	ERT3	sd1	alg	inst.
0.151	1580	4710	-	417	pso	1
5.458	5868	7503	8250	1313	de	1
-	-	-	-	-	so	1
7.370	740	1437	2018	524	cma-es	1
2.237	4266	5881	11314	1047	nea1	1
8.968	1031	1444	2286	599	nea2	1
4.568	1095	3004	8509	806	pso	2
44.755	989	1238	1395	526	de	2
0.131	95290	95501	-	-	so	2
15.286	851	1266	2020	917	cma-es	2
7.351	3807	6094	10809	1226	nea1	2
10.648	1338	2509	5276	1051	nea2	2
0.150	5752	8203	-	750	pso	3
4.018	18566	19596	49899	414	de	3
0.141	95354	96265	-	-	so	3
4.650	2433	3937	11109	1069	cma-es	3
0.745	10252	11902	99635	718	nea1	3
4.501	1587	3564	10687	1013	nea2	3





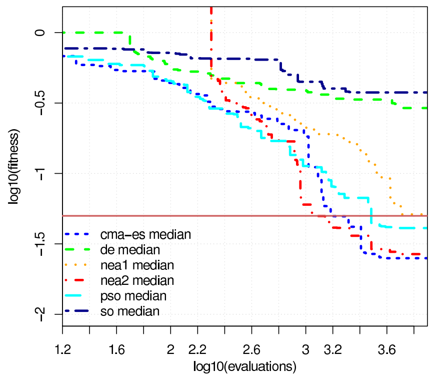


Fig. 3: Table: Diversity based measures for all 6 algorithms on all 3 test problem instances, sd1 resembles the standard deviation over the successful detections of the first solution. Figures: Empirical attainment surface plots of the best obtained solutions over time (only 50% attainment surface), for the three problem instances (first in the upper row). The red line marks the required quality for an applicable solution. Logarithmic scaling on both axes.

Task We do not dare to declare a clearly winning algorithm, instead we demand that the methods find at least 2 sufficiently good and diverse solutions reliably and call these algorithms 'suitable' to the problem, to be considered for further work. However, we take out Wilcoxon rank-sum tests between the time needed to the first optimum as measured in each run (together resembling ERT1), between the different algorithms.

Results/Visualization Figure 3 shows the table of the diversity measures and depicts the median best solutions over the number of spent evaluations for all three instances.

Observations As the variances in the ERT values are quite high (see e.g. the sd1 value), it is dangerous to read too much out of the obtained result. However, from the pictured median performance values, we can clearly see that the third scenario is the hardest, followed by the first one, and the second scenario is the easiest. Concerning the different algorithms, SO does not solve any test case, DE does not solve instance 3 and is very slow on instance 1, and NEA1 is not much better. It is noteworthy that DE is the fastest method to obtain at least 2 or 3 diverse solutions for instance 2. PSO mostly converges quickly to the first solution but needs a lot of time to provide the second one. CMA-ES and NEA2 are both reliable in detecting several solutions, where CMA-ES looks clearly favourable for the simple and the medium instance, and NEA2 a bit better on the hard one. With the noteable exception of DE on instance 2, the diversity values obtained by the best algorithms are comparable. We review the results of our speed based statistical tests only for the leading algorithms: in scenario 1, PSO is significantly worse than CMA-ES and NEA2, but CMA-ES and NEA2 cannot be differentiated. In scenario 2, the leading three (DE, PSO, CMA-ES) are not distinguishable, only between CMA-ES and NEA2 we get significance (at the 5%-level). For scenario 3, the difference of CMA-ES and NEA2 is just significant, while the CMA-ES itself is significantly faster than all others.

Discussion Why DE fails to solve medium or hard instances cannot be easily seen, possibly this is due to premature convergence to a bad local optimum. PSO clearly needs a better restart mechanism as the convergence is often fast but no second best solution can be obtained. However, we would not recommend to use both algorithms for these kind of problems in their current form. More instances would be needed to collect better evidence on the relationship between problem hardness and algorithm performance, but it seems that as a default method, one should employ a CMA-ES unless it is known that the problem is very hard, then niching methods as NEA2 can pay off. At least in the case of given quality and distance requirements, it seems that concentrating on the diversity instead of convergence speeds does not change much, the good algorithms are still the same. This may of course change if no concrete quality and distance tasks are provided.

6 Summary and Conclusions

This paper proposed the application of niching and restart evolutionary algorithms to the problem of diversity of shot generation in virtual camera composition. The suggested algorithms are compared against state-of-the-art algorithms for optimisation of virtual camera composition and have been evaluated in their ability to find up to three different valid solutions on three different problems with varying complexity. Both NEA2 and CMA-ES show at least comparable performance to the standard optimisation algorithms in terms of number of evaluations required to find the first solution; however, for the second and third solution, these two algorithms demonstrated a clear advantage compared to all others included in the experiment.

The actual analysis has been performed using the Euclidean distance as a diversity measure between the solutions. Even though it is effective, this solution does not evaluate accurately how different are the shots generated by two solutions. In the future it is advisable to investigate different objectives next to visibility and also new diversity measurements such as the Euclidean distance in the multi-objective (target) space.

References

1. Arijon, D.: Grammar of the Film Language. Silman-James Press LA (1991)
2. Auger, A., Finck, S., Hansen, N., Ros, R.: BBOB 2010: Comparison Tables of All Algorithms on All Noiseless Functions. Technical Report RT-388, INRIA (Sep 2010)
3. Auger, A., Hansen, N.: 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. pp. 1769–1776. IEEE press (2005)
4. Blinn, J.: Where Am I? What Am I Looking At? IEEE Computer Graphics and Applications 8(4), 76–81 (1988)
5. Bourne, O., Sattar, A., Goodwin, S.: A Constraint-Based Autonomous 3D Camera System. Journal of Constraints 13(1-2), 180–205 (2008)
6. Burelli, P., Yannakakis, G.N.: Combining Local and Global Optimisation for Virtual Camera Control. In: IEEE Conference on Computational Intelligence and Games. p. 403 (2010)
7. Burelli, P., Yannakakis, G.N.: Global Search for Occlusion Minimisation in Virtual Camera Control. In: IEEE Congress on Evolutionary Computation. pp. 1–8. IEEE, Barcelona (2010)
8. Christie, M., Olivier, P., Normand, J.M.: Camera Control in Computer Graphics. In: Computer Graphics Forum. vol. 27, pp. 2197–2218 (2008)
9. Drucker, S.M., Zeltzer, D.: Intelligent camera control in a virtual environment. In: Graphics Interface. pp. 190–199 (1994)
10. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation 9(2), 159–195 (2001)
11. Hansen, N.: The cma evolution strategy: A tutorial, `http://www.lri.fr/~hansen/cmatutorial.pdf`, version of June 28, 2011
12. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: IEEE Conference on Neural Networks. pp. 1942–1948 (1995)
13. Mersmann, O., Preuss, M., Trautmann, H., Bischl, B., Weihs, C.: Analyzing the bbob results by means of benchmarking concepts. Evolutionary Computation (2012), accepted
14. Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., Rudolph, G.: Exploratory landscape analysis. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation. pp. 829–836. GECCO '11, ACM (2011)
15. Olivier, P., Halper, N., Pickering, J., Luna, P.: Visual Composition as Optimisation. In: Artificial Intelligence and Simulation of Behaviour (1999)
16. Preuss, M.: Niching the cma-es via nearest-better clustering. In: Proceedings of the 12th annual conference companion on Genetic and evolutionary computation. pp. 1711–1718. GECCO '10, ACM (2010)
17. Preuss, M.: Improved Topological Niching for Real-Valued Global Optimization. In: Applications of Evolutionary Computation - EvoApplications. Springer (2012), in this volume
18. Preuss, M., Schönemann, L., Emmerich, M.: Counteracting genetic drift and disruptive recombination in $(\mu + /, \lambda)$-EA on multimodal fitness landscapes. In: Proc. Genetic and Evolutionary Computation Conf. (GECCO 2005). vol. 1, pp. 865–872. ACM Press (2005)
19. Shir, O.M., Emmerich, M., Bäck, T.: Adaptive niche radii and niche shapes approaches for niching with the cma-es. Evolutionary Computation 18(1), 97–126 (2010)

20. Stoean, C., Preuss, M., Stoean, R., Dumitrescu, D.: Multimodal optimization by means of a topological species conservation algorithm. IEEE Transactions on Evolutionary Computation 14(6), 842 – 864 (2010)
21. Storn, R., Price, K.: Differential Evolution A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. Journal of Global Optimization 11(4), 341–359 (1997)
22. Thawonmas, R., Oda, K., Shuda, T.: Rule-Based Camerawork Controller for Automatic Comic Generation from Game Log. In: IFIP International Conference on Entertainment Computing. pp. 326–333. Seoul (2010)
23. Ware, C., Osborne, S.: Exploration and virtual camera control in virtual three dimensional environments. ACM SIGGRAPH 24(2), 175–183 (1990)