Approximating Optimal Sets of Views in Virtual Scenes

Sebastian Freitag* Clemens Löbbert † Benjamin Weyers* Torsten W. Kuhlen*

Visual Computing Institute, RWTH Aachen University, Germany — JARA-HPC, Aachen, Germany

Abstract

Viewpoint quality estimation methods allow the determination of the most informative position in a scene. However, a single position usually cannot represent an entire scene, requiring instead a set of several viewpoints. Measuring the quality of such a set of views, however, is not trivial, and the computation of an optimal set of views is an NP-hard problem. Therefore, in this work, we propose three methods to estimate the quality of a set of views. Furthermore, we evaluate three approaches for computing an approximation to the optimal set (two of them new) regarding effectiveness and efficiency.

Index Terms: I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing algorithms

1 Introduction

The concept of viewpoint quality assigns each position in a virtual scene a scalar quality value. For simple scenes, this can be used to choose the best location out of a set of candidates (e.g., a regular sampling) as representative viewpoint for the scene. However, for more complex scenes, a single viewpoint cannot represent the complete scene due to occlusion, necessitating *sets of views* as representatives instead. Such sets can be used, e.g., to generate static overview images, as input to path finding or virtual tour algorithms, or for point-of-interest-based navigation techniques. However, while there are different methods to estimate the quality of a single viewpoint [1], there are very few approaches to judge the quality of a set of views. Furthermore, finding the *best* set of views is an NP-hard problem (related to the Art Gallery Problem [4]).

A straightforward approach that can be computed without an explicit set quality measure is a greedy method, where iteratively the viewpoint with the best quality is added to the set and all parts of the scene visible from that point are subsequently ignored [2, 6]. However, this means that the location with the best individual quality is always part of the resulting set, which is not necessarily optimal. A different greedy approach seeks to minimize the difference between the two probability distributions representing the projected areas of polygons for the viewpoints in the set and the actual polygon areas [5]. Similar problems may arise with this method, and it may perform worse if the scene contains large polygons [1]. The problem has also been formulated as an optimization problem, explicitly modeling set quality [3], and approached with simulated annealing. However, only a small number of candidate positions was considered as possible viewpoints.

2 Measuring Set Quality

The quality of a set of views cannot simply be defined as the sum of the qualities of its constituent viewpoints due to redundancies/overlaps between these views. Therefore, we propose and evaluate three different approaches to measure set quality, based on a viewpoint quality metric. Two of them, *AvoidOverlap* and *BestObserver*, try to minimize redundancies between viewpoints, while the third one, *OptimalOverlap* tries to achieve an optimal overlap between views.

AvoidOverlap The viewpoints in the set are processed successively. For each, its viewpoint quality is computed, whereupon all scene entities visible from this viewpoint are ignored for all following computations. Although the result depends on the order in which the viewpoints are iterated, a natural order exists, always processing the viewpoint with the currently highest quality contribution first.

BestObserver To avoid order effects, the *best observer* for each scene entity is computed, i.e., the viewpoint from which the visible size of that entity is maximal. The contribution of a viewpoint to the set quality is determined by its viewpoint quality computed while only considering scene entities for which it is the best observer.

OptimalOverlap This method seeks to achieve an optimal overlap between viewpoints, as a certain redundancy may be desired to provide context. Therefore, the contribution of each viewpoint to the set quality is its viewpoint quality, reduced by a penalty depending on its overlap with other viewpoints in the set. For each viewpoint, the relative visible size of each scene entity can be represented by a normalized visibility histogram. The overlap of two viewpoints can then be determined from the overlap of the corresponding visibility histograms. To compute the overlap between the histograms a and b, we use the Bhattacharyya coefficient $BC(a,b) = \sum_i \sqrt{a_i \cdot b_i} \in [0,1]$, where $BC(a,b) = 0$ if the views do not overlap, and $BC(a,b) = 1$ if all entities are seen equally well. For each viewpoint, based on the largest overlap with any other viewpoint γ and an optimal overlap θ, a penalty is computed. This penalty should be low for overlaps around θ, rise to 1 (reducing the quality contribution to 0) if the overlap becomes too large, and reaches a medium value for an overlap of 0, to allow viewpoints without any overlap in the set if no better option is available. From these requirements, we chose the penalty function $pen(x) = \frac{2^x \cdot x^3}{2}$ (see Fig. 1), where

$$x = \begin{cases} 2 \cdot \frac{\gamma - \theta}{1 - \theta}, & \text{if } \gamma > \theta \\ 0.9 \cdot \frac{\theta - \gamma}{\theta}, & \text{otherwise} \end{cases}$$

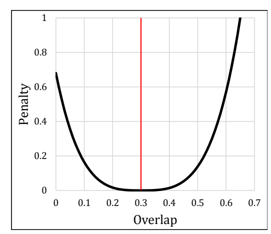

Figure 1: Overlap penalty.

3 Computing Best Sets of Views

We compared three different approaches to approximating the best set of n views, based on any set quality measure (section 2).

Greedy Based on previous work [2, 6], this method iteratively selects the viewpoint increasing the set quality most, until n are selected.

Genetic Algorithm This approach uses the set quality as fitness and the set of n viewpoint positions as an individual's genetic information. The population is modified between generations by applying the *crossover* and *mutation* operators, before performing *selection*. The *crossover* operator combines the viewpoints of two "parents" (sets of views) to create an offspring set. To ensure that the offspring set is similar to its parents, first n pairs of similar viewpoints between parents are formed, before one viewpoint of each pair is chosen at random for the offspring set. The pairs are selected by computing a *minimum weight maximum matching* between the viewpoints of each set, where the edge weight corresponds to the distance between viewpoints. A *mutation* is performed by moving a random viewpoint of the set a random distance in a random direction. Finally, the population is pruned by iteratively selecting n individuals to survive. Each individual is chosen with a probability proportional to its fitness.

*e-mail: {freitag | weyers | kuhlen}@vr.rwth-aachen.de
†e-mail: clemens.loebbert@rwth-aachen.de

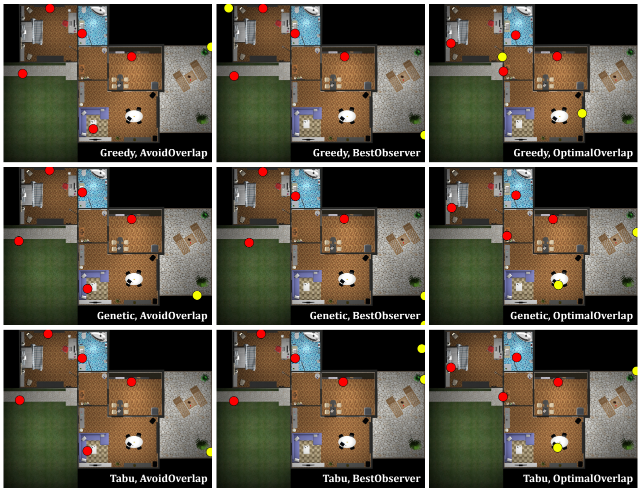

Figure 2: Sets of $n = 6$ views selected on the *house* scene by the three algorithms, using the three set quality methods. Viewpoints marked in red are approximately identical across algorithms for the same measure, yellow viewpoints are different. The position with the highest viewpoint quality in the kitchen (in the center) is part of all results.

Tabu search The tabu search starts with a random set of views, and iteratively tries to improve it by performing the best step (i.e., the one that leads to the highest set quality) out of a discrete set of possible actions, avoiding all steps on the *tabu list* to escape local optima. In our case, the possible actions consist of moving any one of the viewpoints in one of 8 directions by one of 10 distances. To avoid repeating moves, the inverse of each step is recorded in the *tabu list*. Furthermore, the viewpoint moved in the last step cannot be moved again in the next step, to avoid that single viewpoints are moved in a circle back to where they started, circumventing the tabu list. In addition, an *aspiration criterion* is used. If no new best solution is found after a number of iterations—indicating that the search has probably gone into the wrong direction—the search is reset to the currently best solution.

4 Evaluation

We tested each of the algorithms from section 3 in combination with each of the set quality measures from section 2 on three different scenes: a *house*, an *office*, and a *bookstore* scene. However, due to space limitations, we will only present the results on the *house* scene in detail, as they are representative for all three scenes.

To compute viewpoint quality, we used the exponentiated *object area entropy* measure [1]. We always generated sets of size $n = 6$, and used a regular 2D sampling with a spacing of .05 m at 1.65 m height above the floor for candidate viewpoints (125,763 points in total). For the genetic algorithm and tabu search, we interpolated quality and visibility histograms between points, if necessary. The genetic algorithm used a population size of 1800, a mutation probability of .02 and a maximum mutation distance of 15 m. For the tabu search, we used an aspiration criterion of 50 steps, and a move distance per step between .25 m and 5.25 m. We chose $\theta = 0.3$ as optimal overlap. We ran each algorithm for 60 seconds on an AMD Phenom II X6 1100T (3.3 GHz) running Windows 7. The computation of visibilities was performed before as a precomputation step, by rendering a cube map from all viewpoints into an item buffer, as in [1, 6]. As fundamental scene entity, we used objects as defined in [1].

Representative results on the *house* scene are visualized in Figure 2. Interestingly, for each set quality measure, the results of all three algorithms are very similar, placing many points at almost the same locations. Furthermore, the location with the highest viewpoint quality is always in the set, suggesting that this property of the greedy algorithm might not be a strong disadvantage in practice. While both *AvoidOverlap* and *OptimalOverlap* produce useful results, *BestObserver* leads to the placing of one of the six points outside the scene, evidently to keep it from becoming the best observer for any scene entity, which would reduce the quality of other viewpoints. Furthermore, *BestObserver* does not place a point in the living room of the house, leading to arguably worse results than both of the other measures. Overall, the results for *OptimalOverlap* seem to be slightly better than *AvoidOverlap* with the genetic algorithm and tabu search, which we also observed on the *office* and *bookstore* scenes. However, they depend on a good choice of optimal overlap, which may differ for other scenes.

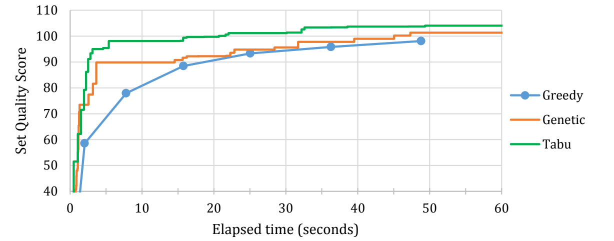

Figure 3: Set quality (*AvoidOverlap* measure) as a function of the runtime on the *house* scene for the three algorithms. The blue dots indicate the selection of an additional viewpoint of the greedy algorithm, both other algorithms start with a complete set.

The achieved set quality score as a function of the runtime is illustrated exemplarily in Figure 3. The genetic algorithm and tabu search reach a high score very quickly, only marginally improving afterwards, while the greedy algorithm provided its result only after considerable time. However, this is mainly due to the fact that it has to take into account all candidate points in each iteration, while both other algorithms only sample the space. As the runtime of the greedy algorithm depends linearly on the number of candidate points, it can be vastly faster if a low number of candidates is acceptable.

5 Conclusion

We proposed and compared three approaches to measure the quality of a set of viewpoints, and three algorithms to compute best sets of views (two of them new). All algorithms produced usable results for the *AvoidOverlap* and *OptimalOverlap* measures on our test scenes. Interestingly, the greedy algorithm performed surprisingly well, which means that this simple approach can be used to compute good sets of views in practice, if the number of candidate points is low. If a finely resolved grid is used instead, we recommend the tabu search approach, as it provides interim results instantly and converges quickly.

In future work, we will investigate approaches to automatically determine the optimal size n of the set. Furthermore, we want to extend our analysis to views of limited field of view, which significantly enlarges the search space due to the added degrees of freedom.

References

[1] S. Freitag, B. Weyers, A. Bönsch, and T. W. Kuhlen. Comparison and Evaluation of Viewpoint Quality Estimation Algorithms for Immersive Virtual Environments. In *ICAT-EGVE 2015*, pages 53–60, 2015.

[2] B. Jaubert, K. Tamine, and D. Plemenos. Techniques for Off-line Scene Exploration Using a Virtual Camera. In *Int. Conf. 3IA*, volume 6, 2006.

[3] P. Moreira, L. Reis, and A. De Sousa. Best Multiple-View Selection for the Visualization of Urban Rescue Simulations. *International Journal of Simulation Modelling*, 5(4):167–173, 2006.

[4] J. O'Rourke. *Art Gallery Theorems and Algorithms*, volume 57. Oxford University Press, 1987.

[5] M. Sbert, D. Plemenos, M. Feixas, and F. González. Viewpoint Quality: Measures and Applications. In *Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging*, pages 185–192, 2005.

[6] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Automatic View Selection Using Viewpoint Entropy and its Application to Image-Based Modelling. In *Computer Graphics Forum*, volume 22, pages 689–700, 2003.