New Triggers for Automatic Camera Placement Over Time

Meghanto Majumder*
University of Oregon

Nicole Marsaglia†
Lawrence Livermore National Laboratory

Hank Childs‡
University of Oregon

Abstract

We propose new trigger variants for in situ automatic camera placement over time (ACPOT). We evaluate the performance of these variants on three data sets from two simulation codes. We find that our approach has two major benefits: (1) it mitigates a problem where the camera "stagnates" on a view occluded from interesting phenomena and (2) proposes alternate trigger criteria that provides comparable camera placement quality (evaluated using a entropy-based viewpoint quality metric) with reduced computational cost.

1 Introduction

Camera placement is an important aspect of visualizing scientific data. In the post hoc case, i.e., when data is visualized after it is generated, scientists can be part of the visualization process, interactively setting a camera placement themselves. However, the in situ case, i.e., when data is visualized as it is generated, typically does not consist of a human-in-the-loop (HITL) workflow, making it more difficult for the scientist to direct the camera placement process. In this non-HITL case, camera positions can be obtained by referring to predecessor, smaller-scale simulations, by incorporating knowledge from domain scientists, or by applying an automated technique. With this work, we consider the automation of camera placements, building on efforts by Marsaglia et al. [5]. Their work used Viewpoint Quality (VQ) metrics, specifically a metric they referred to as "DDS entropy," to find preferable camera positions. The "DDS" acronym refers to Data, Depth and Shading, with Data evaluating the visible data of a field, Depth evaluating the distance from camera to the visible field data, and Shading evaluating the shading coefficients of the visible geometry associated with the data. The Shannon Entropy is calculated for each of these three components to obtain the average level of information, and then the three entropies are summed. Their user study on isosurface data found this metric predicts expert preference 68% of the time, i.e., when given the choice between two cameras, experts chose the camera with the higher DDS Entropy score more than two thirds of the time. Their subsequent work [3] provided an approach to calculate the DDS Entropy efficiently in a distributed-memory parallel environment and also an approach to quickly search the set of possible camera positions to find a desirable position.

Search time is an important consideration. Evaluating the DDS Entropy of a camera position requires rendering the scene, putting the time to evaluate a position at slightly more than the rendering time. Hence, Marsaglia et al. [5] recommends lower "search budgets" (the number of positions to consider), as marginal improvements in DDS Entropy score does not always translate to views that experts prefer. Marsaglia's doctoral dissertation [4] also expands the idea to a time-varying setting by considering automatic camera placement over time (ACPOT). Their ACPOT algorithm does not do the trivial approach of doing a fresh search each cycle, but instead only "triggers" a search for a new camera position space when the DDS Entropy score of the current best camera position changes by a set percentage threshold. This approach aims to balance between overhead (number of cameras considered) and quality (measured by DDS Entropy score).

This work proposes three variations to the ACPOT approach established by Marsaglia's doctoral dissertation. One of the main intuitions behind these variations is the need to prevent view "stagnation," i.e., if the DDS Entropy score has not changed significantly for many cycles, then to still perform an occasional search, in case interesting phenomena are fully occluded. In our results, we show that these variations outperform the ACPOT algorithm proposed in Marsaglia's doctoral dissertation.

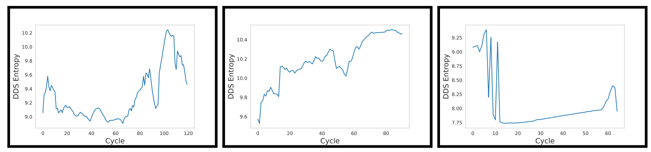

Figure 1: Three figures showing the entropy scores of the best sample at each cycle, with a search budget of 10 positions. From left to right: Jetbox, Ball of Fury, and AMR-Wind.

2 Our Method

We extend the method of a trigger-based ACPOT algorithm by suggesting variants of the trigger described by Marsaglia in her dissertation [4]. We initially run the Marsaglia et al. [3] search algorithm to find the initial camera position $C^{\star}$, and its corresponding DDS Entropy score $DDS^{\star}$. The total number of cycles elapsed since the last searched is also stored, along with the current DDS score of camera position $C^{\star}$, i.e., DDS^{cur}. In the original approach, the trigger would "fire" if DDS^{cur} has changed significantly from $DDS\star$, as determined by a DDS Entropy threshold. In our first proposed variant, **Variant 1**, the trigger only fires if there is a significant decrease in DDS^{cur} from $DDS^{\star}$, and if DDS^{cur} is greater than $DDS^{\star}$, then $DDS^{\star}$ gets updated to the new value of DDS^{cur}. Therefore, $DDS^{\star}$ always refers to the maximum entropy value encountered since last search. In **Variant 2**, along with the original's trigger criteria, a time threshold is considered. Therefore, alongside checking for change in DDS^{cur}, we also check if the number of cycles since the last search is higher than the time threshold. If any of the two conditions are fulfilled, we fire the trigger. **Variant 3** combines the trigger conditions of Variants 1 and 2. The Variant 3 trigger fires if DDS^{cur} has decreased significantly from $DDS^{\star}$, where $DDS^{\star}$ is the maximum entropy value encountered since last search. The Variant 3 trigger also fires if the number of cycles since the last search is higher than the time threshold considered.

3 Background for Experiments

Data Sets:

For our experiments, we considered three data sets coming from two simulation codes: AMR-Wind [1] and CloverLeaf3D [2]. The first data set, "Ball of Fury," came from a Cloverleaf simulation of 9100 simulation cycles on a 64^3 mesh, with visualization occurring every 100 cycles (91 total visualizations). The visualization was of isosurfaces of the energy field. The second data set, Jetbox,

*e-mail: meghanto@uoregon.edu
†e-mail: marsaglia1@llnl.gov
‡e-mail: hank@uoregon.edu

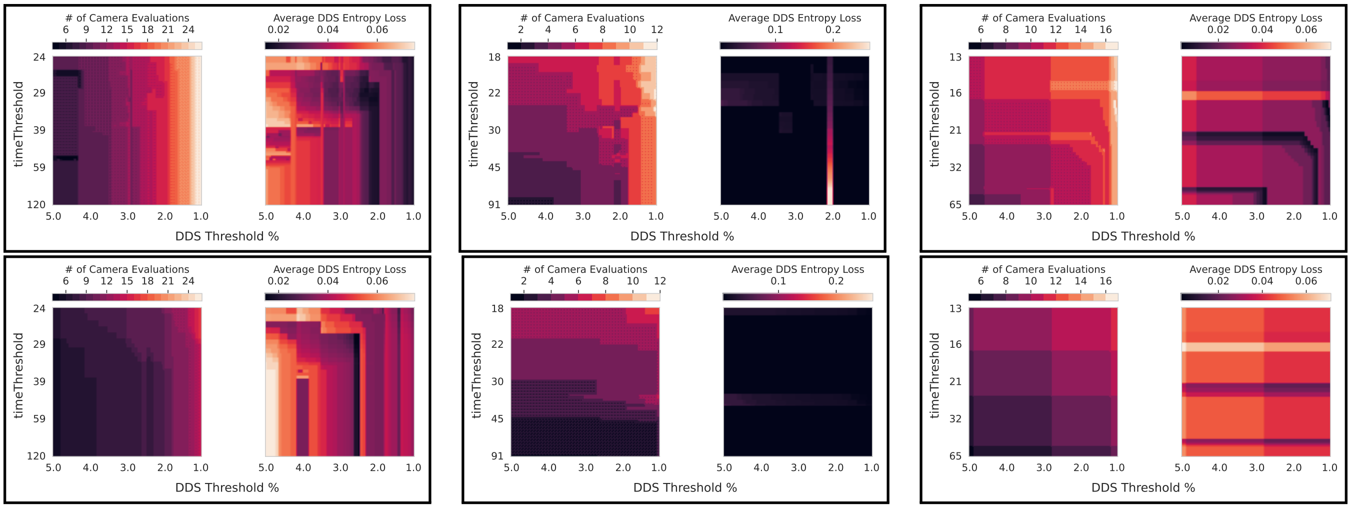

Figure 2: Six pairs of figures, where each pair shows how Number of Camera Positions Evaluated and Average DDS Entropy loss varies for various values of DDS Entropy threshold percentage and time threshold, with a search budget of 10 positions. The columns are organized by data set: Jetbox (left), Ball of Fury (middle), and AMR-Wind (right). The rows are organized by trigger variants: Variant 2 (top) and Variant 3 (bottom). The original approach and Variant 1 are special cases of Variant 2 and 3, where the time threshold equals the number of cycles in the data.

also came from a Clover simulation. This simulation had 1200 simulation cycles on a $256 \times 128 \times 256$ mesh, with visualization occurring every 10 cycles (120 total visualizations). Once again, the visualization was of isosurfaces of the energy field. The third data set, AMR-Wind, was of 65 simulation cycles on a 848^3 mesh, with visualization occurring every cycle (65 total visualizations). This visualization was of isosurfaces of the velocity field. Figure 1 shows the change in DDS Entropy over time for all data sets.

Metrics: The two key elements that determine the performance of an ACPOT algorithm as described above are:

- Computational Work Performed: We use the number of camera searches triggered as the metric of computational work performed. The total time taken to execute the algorithm scales linearly with the number of searches.
- Quality of Chosen Cameras: We use the loss in DDS Entropy as the metric. At each cycle, for a fixed sampling budget, the difference between the DDS entropy score of the "best sample" and the chosen sample is recorded. We average over this metric to assess the quality of the algorithm's full run.

4 Results

We run the experiments by varying both the DDS entropy threshold and the time threshold between the ranges of 1% - 5% and $N_d/5$ - N_d respectively, where N_d is the number of cycles for data set d. We evaluate Variant 2 and Variant 3 triggers on the data sets and report our findings. The original approach and Variant 1 are also evaluated, as they are a special case of Variant 2 and 3 respectively, when the time threshold equals N_d (as the trigger based on number of cycles since last search never reaches the threshold value).

Figure 2 shows the effect on both the number of searches and average DDS entropy loss. The fields are mapped to the same color map across the variants, for easy visual comparison. From the analysis, we find that Variant 3 tends to trigger less searches compared to Variant 2 on average. Variant 3 is comparable to Variant 2 in average DDS Entropy loss, although the average DDS Entropy loss in Variant 3 tends to be higher when DDS Threshold is low. We also find that both Variant 2 and Variant 3 can overcome the issue where the view "stagnates," as the time threshold gets lowered. This result is most notable in Ball of Fury's Variant 2 results, where the average DDS entropy loss is very high around the 2% DDS Threshold region, which gradually fades when we lower the time threshold.

5 Conclusion

We presented new triggers for the ACPOT algorithm proposed in Marsaglia's doctoral dissertation [4]. For the same data sets and search budget, we find that the triggers variants proposed here can outperform the original approach. Variant 2, which adds a time threshold trigger onto the original approach, mitigates the issue where the view "stagnates" and interesting phenomena are occluded. Variant 3 is suggested as it performs comparably with Variant 2 but does not trigger as many searches, making it computationally more efficient at the cost of DDS entropy. Future work includes exploring and analyzing the space of search budget, time threshold, and DDS entropy threshold across a variety of data sets, exploring triggers with variable search budgets, and optimizing the search algorithm for a time-varying setting.

Acknowledgments

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

References

[1] S. Ananthan, M. Brazell, J. Rood, G. Vijayakumar, M. Henry de Frahan, A. Almgren, and W. Zhang. Amr-wind, aug 2020.

[2] A. Mallinson, D. A. Beckingsale, W. Gaudin, J. Herdman, J. Levesque, and S. A. Jarvis. Cloverleaf: Preparing hydrodynamics codes for exascale. *The Cray User Group*, 2013, 2013.

[3] N. Marasaglia, M. Mathai, S. Fields, and H. Childs. Automatic In Situ Camera Placement for Large-Scale Scientific Simulations. In *Eurographics Symposium on Parallel Graphics and Visualization (EGPGV)*, pp. 49–59. Rome, Italy, June 2022.

[4] N. Marsaglia. *Automating Camera Placement for In Situ Visualization*. PhD thesis, University of Oregon, 2022. `https://cdux.cs.uoregon.edu/pubs/MarsagliaDissertation.pdf`.

[5] N. Marsaglia, Y. Kawakami, S. D. Schwartz, S. Fields, and H. Childs. An Entropy-Based Approach for Identifying User-Preferred Camera Positions. In *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 73–83, Oct. 2021.