

ERDC/ITLMP-21-4

Information Technology Laboratory



**US Army Corps
of Engineers®**
Engineer Research and
Development Center



In Situ and Post-Processing Volume Rendering with Cinema

Andrew C. Bauer, Jennifer Abras, and Nathan Hariharan

April 2021

The U.S. Army Engineer Research and Development Center (ERDC) solves the nation's toughest engineering and environmental challenges. ERDC develops innovative solutions in civil and military engineering, geospatial sciences, water resources, and environmental sciences for the Army, the Department of Defense, civilian agencies, and our nation's public good. Find out more at www.erdclibrary.on.worldcat.org/discovery.

To search for other technical reports published by ERDC, visit the ERDC online library at <https://erdclibrary.on.worldcat.org/discovery>.

In Situ and Post-Processing Volume Rendering with Cinema

Andrew C. Bauer

*Information Technology Laboratory
U.S. Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180*

Jennifer Abras and Nathan Hariharan

*HPCMP CREATE Quality Assurance
10501 Furnace Road
Lorton, VA 39762*

Final report

Approved for public release; distribution is unlimited.

Prepared for U.S. Army Corps of Engineers
Washington, DC 20314

Under PE 633461DS700

Preface

This study was conducted for the U.S. Army Corps of Engineers (USACE) and the workshop presentation was made possible with outreach funding under Program Element 633461DS700. Material presented was a product of the Computational Research and Engineering for Acquisition Tools and Environments (CREATE) element of the Department of Defense (DoD) High Performance Computing (HPC) Modernization Program. The technical monitor was Dr. Michael M. Stephens.

The work was performed by the Data Analysis and Assessment Center Branch of the Supercomputing Research Center Division, U.S. Army Engineer Research and Development Center, Information Technology Laboratory (ERDC-ITL). At the time of publication, Dr. Michael M. Stephens was Branch Chief, and Mr. Robert M. Hunter was Division Chief. The Deputy Director of ERDC-ITL was Ms. Patti S. Duett, and the Director was Dr. David A. Horner.

This paper was presented at the ISAV 2020 Workshop and was published in ISAV 2020 Workshop proceedings November 16, 2020.

The Commander of ERDC was COL Teresa A. Schlosser and the Director was Dr. David W. Pittman.

DISCLAIMER: The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. All product names and trademarks cited are the property of their respective owners. The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

DESTROY THIS REPORT WHEN NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

In Situ and Post-Processing Volume Rendering with Cinema

ABSTRACT

We present a new batch volume rendering technique which alleviates the time and expertise needed by the domain scientist in order to produce quality volume rendered results. This process can be done both *in situ* and as a post-processing step. The advantage of performing this as an *in situ* process is that the user is not required to have *a priori* knowledge of the exact physics and how best to create a transfer function to volume render that physics during the *in situ* run. For the post-processing use case, the user has the ability to easily examine a wide variety of transfer functions without the tedious work of manually generating each one.

CCS CONCEPTS

- **Computing methodologies** → **Massively parallel algorithms;**
- **Human-centered computing** → **Scientific visualization; Visualization techniques.**

1 INTRODUCTION

Direct volume rendering is a powerful visualization technique that can reveal the intricacies of complex scientific data. Unfortunately though, generating a quality opacity transfer function in order to reveal these intricacies is still as much of an art form as it is a technical process. This can be a time consuming manual process and the larger the dataset is, the more time consuming the process can be. Additionally, when *in situ* analysis and visualization is added to the workflow the burden on the domain expert can become too large to overcome. If the data produced during the *in situ* instrumented simulation run is not familiar enough to the domain expert then this poses extra difficulties in trying to set up appropriate transfer functions for volume rendering. To overcome these issues, we utilize

the concepts of Cinema [2] which generate a number of images over a specified parameter space. Our parameter space is based on creating a number of opacity transfer functions. In this paper we demonstrate how volume rendering with Cinema can be used *in situ* and *post hoc* to effectively visualize results.

2 BACKGROUND

The earliest published works for volume rendering [7] and *in situ* processing [14] go back to 1984 and 1964, respectively, with a significant amount of work in both fields since then. Because of this we refer readers to the recent survey papers on transfer function design for volume rendering [9] and *in situ* analysis and visualization [4] for more complete backgrounds in these subjects.

2.1 Transfer Function Design

In 2001 "The Transfer Function Bake-Off" [10] performed comparisons of four general approaches to transfer function design in which they listed the following general methods:

- trial and error, with minimum computer aid;
- data-centric, with no underlying assumed model;
- data-centric, using an underlying data model; and
- image-centric, using organized sampling.

Since then transfer function generation has become more complex with improved semi-automatic methods for users as well as dealing with multi-dimensional data. Still though, many general use post-processing tools currently utilize the trial and error method with the GUI providing a simplified means of setting both the opacity and color transfer functions for one-dimensional data.

2.2 *In Situ* Analysis and Visualization

While much of the early *in situ* work was focused on researching the subject, of late a significant amount of work has also included development of production level tools that are being used in a variety of simulation codes. While tools like ParaView Catalyst [5], VisIt LibSim [13] and Galaxy [1] are all able to do volume rendering *in situ*, they make the assumption that setting the opacity transfer function is a relatively straightforward task in setting up the *in situ* visualization. While this may be the case where the domain expert is very familiar with the problem set up being run, for use-cases where the user has only a vague idea of the what the results may look like, setting up an opacity transfer function before the simulation is executed may produce undesirable results.

3 VOLUME RENDERING WITH CINEMA

We utilize Cinema techniques to simplify creating an appropriate opacity transfer function for volume rendering. The concept is to explore a variety of opacity transfer functions in a methodical

manner. This is done by dividing up the volume rendered field range into a number of subranges, N . Then, a set of functions are selected to represent the transfer function inside each of these subranges. We use a continuous piecewise linear function form, shown in Figure 1a. The simplest version would allow only one of these functions to be non-zero per Cinema image. For each image the function would vary between fully transparent and the user specified maximum opacity. This results in $N+1$ output Cinema images per camera view per output time step. In many situations though, this would not lead to useful output, especially *in situ* since it would only be able to display one portion of the full field range and everything else would be transparent.

The next step in complexity is allowing multiple functions to be simultaneously non-zero. If we allow all combinations except for the situations where all functions are enabled or all functions are disabled, i.e. rendering all field values at the specified opacity level or nothing rendered, respectively, the number of output Cinema images per camera view per output time step becomes $\sum_{i=1}^N \frac{(N+1)!}{i!(N+1-i)!}$. An example of this is shown in Figure 1b. This is still problematic though in that there is still no proper control over the opacity level. This can be improved by allowing each piecewise linear function to have a maximum value that is less than the user specified maximum opacity level. For this we add an extra Cinema parameter corresponding to the number of non-zero opacity levels, M , that each piecewise linear function will reach at its maximum value. Figure 1c shows a transfer function constructed with M set to 2 with the third and sixth functions attaining a maximum opacity of 0.5 while the first function attains the user specified maximum opacity of 1. This results in $\sum_{i=1}^N M^i \frac{(N+1)!}{i!(N+1-i)!}$ images per camera view per output time step, ignoring the two extreme cases.

Unfortunately, for even moderate values of M and N , the number of images that are generated becomes unmanageable. For many situations users would want few of the subregion functions to be simultaneously active. In this case the number of simultaneously active functions can be limited to be less than a given parameter L . This limits the number of Cinema images created to $\sum_{i=1}^L M^i \frac{N+1}{i!(N+1-i)!}$. Additional Cinema controls include specifying the maximum opacity level, limiting the range of the subregion division space to less than the full field range and whether or not to export the transfer functions to a JSON file for later use.

4 RESULTS

4.1 *In Situ* Volume Rendering Results

We utilized SENSEI's [3] Oscillator mini-app as the driver for testing the *in situ* Cinema volume rendering technique. Our algorithms were built in a custom version of ParaView Catalyst [5]. The SENSEI Oscillator was run with 8 oscillations that varied the field between values of 0 and 1. The Cinema volume rendering output parameters had 5 subregions, N , over the entire field range with the number of non-zero opacity levels, M , set to 2 and the maximum number active piecewise linear functions, L , set to 2. This resulted in 72 images per output time step with only one camera angle used. Sample Cinema output is shown in Figure 2 for simulation time 9.5. See the

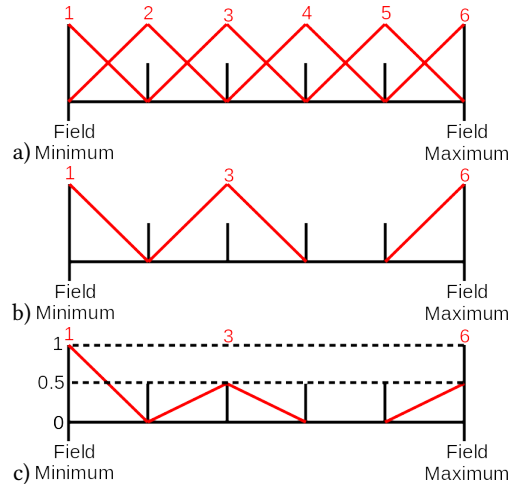


Figure 1: Examples of piecewise linear opacity transfer functions: a) Six continuous piecewise linear functions shown together; b) three piecewise linear function simultaneously active; and, c) three piecewise linear functions with different maximum opacity levels simultaneously active.

reproducibility information in Appendix A for details on replicating the results¹.

4.2 Post-Processing Volume Rendering Results

For our post-processing use case we utilize hovering rotor simulation results from the HPCMP CREATETM-AV Helios rotorcraft simulation code [11]. The data is from a simulation of a hovering rotor. A rotor consists of a collection of lifting surfaces distributed radially about a common rotation axis. The aerodynamic wake of a hovering rotor primarily consists of helical tip vortices and helical wake sheets with additional vortex dynamics and secondary structures included which provides a challenging application for volume rendering techniques. A review of the current state-of-the-art for hover predictions is provided in [6]. Multiple results have been manually volume rendered with an example being shown in Figure 3a. While this image visualizes the tip wake well, the wake sheets are not fully shown. To visualize the wake sheets, we utilized the Cinema volume rendering method with the following parameters:

N=10,20,30 We refined the subregion value to properly resolve the wake sheets.

M=2 We used 2 for the maximum number of simultaneous functions since we knew that the wake sheets would be in a continuous range and that we may want a slightly wider, non-translucent portion of the opacity transfer function to visualize the wake sheet.

L=1 Since we were only visualizing the wake sheets we only needed one maximum opacity value. We set the maximum

¹The reproducibility appendix is the standard form that the authors found for Supercomputing which describe the system and conditions under which the results were produced. The website that contains the software is at <https://zenodo.org/record/3904437#.X2zCXXZKqSF>.

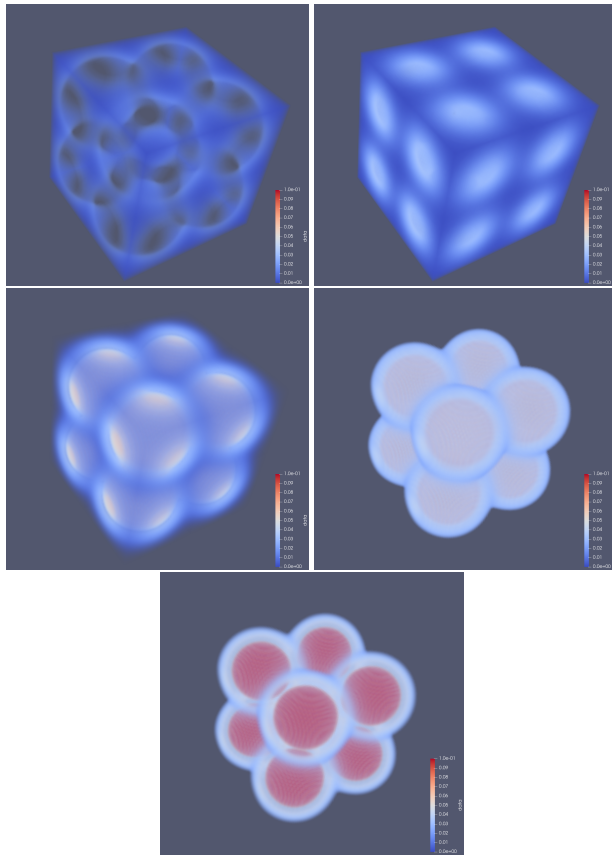


Figure 2: Images for the 0th, 15th, 30th, 45th and 60th opacity transfer functions from the SENSEI Oscillator.

opacity value to 0.5 because of the multiple wake sheets present.

Field range We initially used the original image's field range of $[-0.2, 0.4]$. In the final run the field range was set to $[-0.01, 0.0005]$ to better capture the small wake sheet phenomena.

One of the Cinema output images from the post-processing runs is shown in Figure 3b. With the aid of the Cinema images and their corresponding transfer functions, we were able to construct an appropriate opacity transfer function that combined the qualities of the original opacity transfer function, shown in Figure 4a, that was able to capture rotor tip vortices along with the Cinema opacity transfer function that captured the wake sheets. The final volume rendered image is shown in Figure 3c along with its combined transfer functions in Figure 4b. The thin bump in Figure 4b is the portion of the opacity transfer function that corresponds to visualizing the wake sheet.

5 CONCLUSION

We demonstrated initial work on using Cinema techniques with volume rendering for both *in situ* and *post hoc* use. For *in situ*, the technique provided a way to use volume rendering without requiring specification of an appropriate opacity transfer function *a priori*. For post-processing, the Cinema technique was helpful in

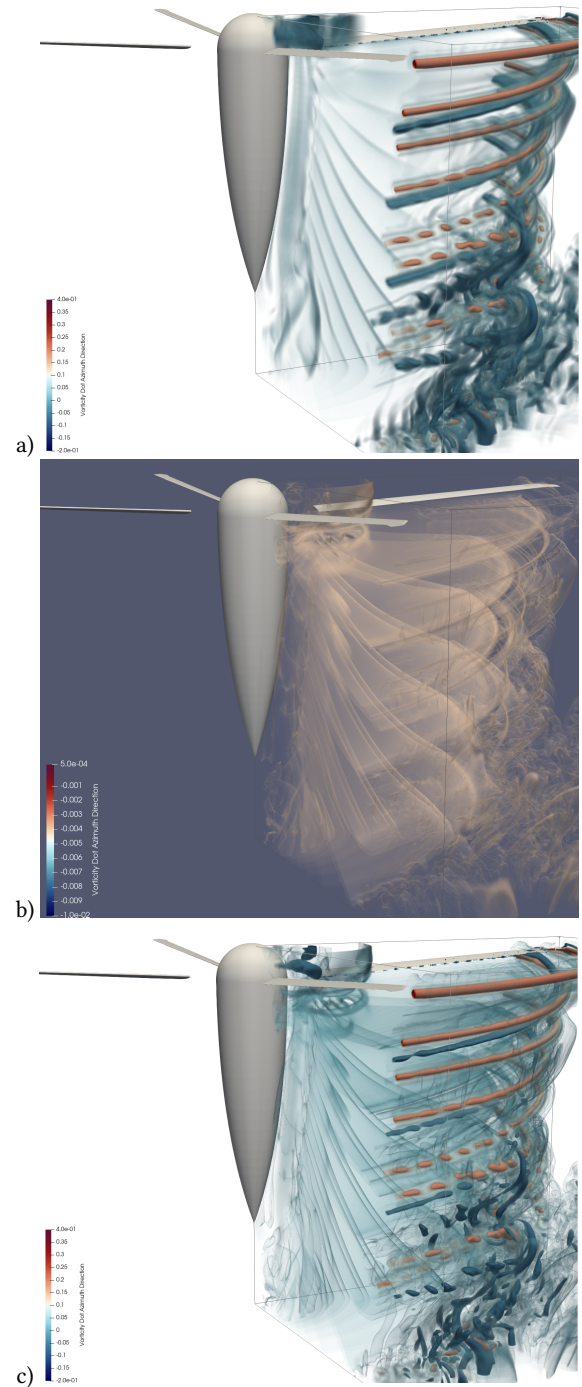


Figure 3: Comparison of volume rendered images: original image (a); Cinema image targeting wake sheet resolution (b); and, image using combination of original and Cinema generated opacity transfer functions (c).

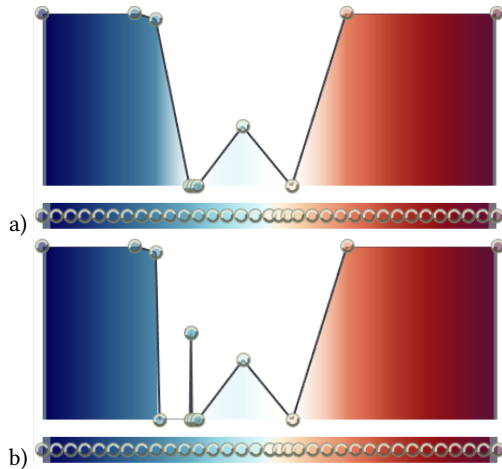


Figure 4: Original (a) and manually modified (b) transfer functions from using Cinema for post-processing.

exploring a number of opacity transfer functions efficiently to find a small scale phenomena in the simulation results, which was not found manually.

Future work includes proper integration into ParaView and methods to reduce the number of images generated even further. We expect that using histogram information to eliminate ranges of the field that are either too densely or sparsely populated with data could significantly reduce the number of images generated when either the number of opacity levels, L , is greater than 1 or when the maximum number of simultaneously active piecewise functions, M , is large. In addition to reducing the number of images produced, there is such work as explorable images [12] and the various Cinema viewers [8] that can be used to reduce the number of useful images the domain scientist needs to view in order to glean insight into their problem at hand.

ACKNOWLEDGMENTS

Material presented is a product of the CREATE (Computational Research and Engineering for Acquisition Tools and Environments) element of the U.S. Department of Defense HPC Modernization Program.

REFERENCES

- [1] G. Abram, P. Navrátil, P. Grosset, D. Rogers, and J. Ahrens. 2018. Galaxy: Asynchronous Ray Tracing for Large High-Fidelity Visualization. In *2018 IEEE 8th Symposium on Large Data Analysis and Visualization (LDAV)*. 72–76.
- [2] J. Ahrens, S. Jourdain, P. O’Leary, J. Patchett, D. H. Rogers, and M. Petersen. 2014. An Image-Based Approach to Extreme Scale In Situ Visualization and Analysis. In *SC’14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 424–434.
- [3] U. Ayachit, B. Whitlock, M. Wolf, B. Loring, B. Geveci, D. Lonie, and E. W. Bethel. 2016. The SENSEI Generic In Situ Interface. In *2016 Second Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization (ISAV)*. 40–44.
- [4] Andrew C. Bauer, Hasan Abbasi, James Ahrens, Hank Childs, Berk Geveci, Scott Klasky, Kenneth Moreland, Patrick O’Leary, Venkatram Vishwanath, Brad Whitlock, and E. Wes Bethel. 2016. In Situ Methods, Infrastructures, and Applications on High Performance Computing Platforms. *Computer Graphics Forum* 35, 3 (June 2016), 577–597. <https://doi.org/10.1111/cgf.12930>
- [5] Nathan Fabian, Kenneth Moreland, David Thompson, Andrew C. Bauer, Pat Marion, Berk Geveci, Michel Rasquin, and Kenneth E. Jansen. 2011. The ParaView Coprocessing Library: A Scalable, General Purpose In Situ Visualization Library. In *Proceedings of the IEEE Symposium on Large-Scale Data Analysis and Visualization*. 89–96. <https://doi.org/10.1109/LDAV.2011.6092322>
- [6] Narducci R. Reed E. Hariharan, N. and A. Egolf. [n.d.]. *AIAA Standardized Hover Simulation: Hover Performance Prediction Status and Outstanding Issues*.
- [7] James T. Kajiya and Brian P Von Herzen. 1984. Ray Tracing Volume Densities. *SIGGRAPH Comput. Graph.* 18, 3 (Jan. 1984), 165–174. <https://doi.org/10.1145/964965.808594>
- [8] LANL. 2020. *Cinema Science*. <https://cinemascience.github.io/downloads.html>
- [9] Patric Ljung, Jens Krüger, Eduard Groller, Markus Hadwiger, Charles D. Hansen, and Anders Ynnerman. 2016. State of the Art in Transfer Functions for Direct Volume Rendering. *Computer Graphics Forum* 35, 3 (2016), 669–691. <https://doi.org/10.1111/cgf.12934> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12934>
- [10] H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. S. Avila, K. M. Raghu, R. Machiraju, and Jinho Lee. 2001. The transfer function bake-off. *IEEE Computer Graphics and Applications* 21, 3 (2001), 16–22.
- [11] Venkateswaran Sankaran, Jayanarayanan Sitaraman, Andrew Wissink, Anubhav Datta, Buvana Jayaraman, Mark Potsdam, Dimitri Mavriplis, Zhi Yang, David O’Brien, Hossein Saberi, Rui Cheng, Nathan Hariharan, and Roger Strawn. [n.d.]. *Application of the Helios Computational Platform to Rotorcraft Flowfields*. <https://doi.org/10.2514/6.2010-1230> arXiv:<https://arc.aiaa.org/doi/pdf/10.2514/6.2010-1230>
- [12] A. Tikhonova, C. D. Correa, and K. Ma. 2010. Explorable images for visualizing volume data. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*. 177–184.
- [13] Brad Whitlock, Jean M. Favre, and Jeremy S. Meredith. 2011. Parallel In Situ Coupling of Simulation with a Fully Featured Visualization System. In *Eurographics Symposium on Parallel Graphics and Visualization*. <https://doi.org/10.2312/EGPGV/EGPGV11/101-109>
- [14] E. E. Zajac. 1964. Computer-Made Perspective Movies as a Scientific and Communication Tool. *Commun. ACM* 7, 3 (March 1964), 169–170. <https://doi.org/10.1145/363958.363993>

Artifact Description / Article Evaluation (AD/AE)

Appendix Form

SC19 submitting authors must complete this form to describe all computational artifacts their results rely on: software, data, and/or hardware. If your paper used no computational artifacts, respond "No" to the first question and you're done.

Please familiarize yourself with the SC Reproducibility Initiative by reading the webpage at:
— <https://sc18.supercomputing.org/submit/sc-reproducibility-initiative/>

Find author resources at:
— <https://github.com/SC-Tech-Program>

* Required

1. **Are there computational artifacts such as datasets, software, or hardware associated with this paper? ***

Mark only one oval.

- Yes — continue to form
- No — end form (auto-generate Appendix).

2. **Summarize the experiments reported in the paper and how they were run. (Example: "We ran the NAS Parallel Benchmarks v3.3.1 on NERSC's Cori supercomputer with both Cray's version of MPICH 3.2.1 and with our SuperPGAS communication layer (v0.2), as described in the paper.").** *

We ran the SENSEI Oscillator mini-app on

a RHEL7 workstation with OpenMPI v1.10.7

with 20 processes. This was run to test our

algorithms for in situ use. Our reproducibility

artifacts are only available for our in situ results.

Artifacts Available (AA)

This section of the form determines eligibility for the Artifacts Available badge. Three outcomes are possible for your paper: (1) it is eligible for both the AA badge and the Student Cluster Competition Reproducibility Challenge; (2) it is eligible for the AA badge, but not the Reproducibility Challenge; (3) it is ineligible for both the badge and the challenge.

AA badge — "This badge is applied to papers in which associated artifacts have been made permanently available for retrieval. Author-created artifacts relevant to this paper have been placed on a publicly accessible archival repository. A DOI or link to this repository along with a unique identifier for the object is provided."

<https://www.acm.org/publications/policies/artifact-review-badging>

3. **Software Artifact Availability:** see <https://opensource.org/licenses/alphabetical> *

Mark only one oval.

All author-created software artifacts are maintained in a public repository under an OSI-approved license.

Some author-created software artifacts are NOT maintained in a public repository or are NOT available under an OSI-approved license.

There are no author-created software artifacts.

4. **Hardware Artifact Availability:** see <https://www.oshwa.org/definition/> *

Mark only one oval.

All author-created hardware artifacts are available and comply with the Open Source Hardware Definition.

Some author-created hardware artifacts are NOT available or do NOT comply with the Open Source Hardware Definition.

There are no author-created hardware artifacts.

5. **Data Artifact Availability** *

Mark only one oval.

All author-created data artifacts are maintained in a public repository with a stable identifier, such as a DOI.

Some author-created data artifacts are NOT maintained in a public repository or do NOT have a stable identifier, such as a DOI.

There are no author-created data artifacts.

6. **Proprietary Artifacts:** see <http://www.linfo.org/proprietary.html> *

Mark only one oval.

None of the associated artifacts, author-created or otherwise, are proprietary.

No author-created artifacts are proprietary. There are associated proprietary artifacts that are not created by the authors.

Some author-created artifacts are proprietary.

7. **List of URLs and/or DOIs where artifacts are available (this information will be unavailable to reviewers, but will be available to the AD/AE Appendices Committee).** *

<https://zenodo.org/record/3904437#.XwXlfHVKgSE>

8. **Do you wish your paper to be considered for the Student Cluster Competition Reproducibility Challenge in SC20? ***

Mark only one oval.

Yes

No

Baseline experimental setup, and modifications made for the paper

Fill in whatever is relevant to your paper and leave the rest blank.

9. **Relevant hardware details, e.g., system names, makes, models, and key components such as CPUs, accelerators, and filesystems.**

10. **Operating systems and versions (e.g., "Ubuntu 17.10 running Linux kernel 4.13.0")**

RHEL7 Linux Kernel 3.10.0

11. **Compilers and versions (e.g., "Clang++ v6.0")**

GCC 7.3.0

12. **Applications and versions (e.g., "NAMD v2.13" or "SPEC CPU2017")**

13. **Libraries and versions (e.g., "OpenMPI v3.1.0")**

OpenMPI v1.10.7

14. **Key algorithms (e.g., "conjugate gradient")**

15. **Input datasets and versions (e.g., "Berkeley Segmentation Dataset: Test Image #296059 [color]")**

16. **Modifications made for the paper: describe how the hardware and software listed in the previous section was improved or otherwise altered.**

ParaView was modified to include our new

Cinema volume rendering method.

SENSEI was modified to build with our version

of ParaView.

17. **Output from scripts that gathers execution environment information — see example scripts at <https://github.com/SC-Tech-Program>**

This was too long to include here. It is included

in the artifacts at <https://zenodo.org/record/3904437#.XwXlfHVKgSE>

in the collect_environment.out file.

Artifact Evaluation

Discuss the steps taken to help ensure the computational artifacts and results are trustworthy.

Describe controls your team put in place, statistics gathered, or other measures to make the measurements and analyses robust to variability and unknowns in the system. E.g., validation of accuracy and precision of timings, use of manufactured solutions or spectral properties, accounting for aleatoric and epistemic uncertainties, sensitivity of results to initial conditions, sensitivity to parameters and computational environment. Did you perform verification and validation studies?

18. **Are you completing an Artifact Evaluation (AE) Appendix? ***

Mark only one oval.

- Yes — continue to next section.
- No — end form (auto-generate Appendix)

19. **Artifacts Evaluation — Describe if and how you: (a) performed verification and validation studies; (b) validated the accuracy and precision of timings; (c) used manufactured solutions or spectral properties; (d) accounted for aleatoric and epistemic uncertainties; (e) quantified the sensitivity of your results to initial conditions and/or parameters of the computational environment. Also describe controls, statistics, or other steps taken to make the measurements and analyses robust to variability and unknowns in the system.**
-

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD-MM-YYYY) April 2021			2. REPORT TYPE Final		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE In Situ and Post-Processing Volume Rendering with Cinema					5a. CONTRACT NUMBER	
					5b. GRANT NUMBER	
					5c. PROGRAM ELEMENT NUMBER 633461DS700A	
6. AUTHOR(S) Andrew C. Bauer, Jennifer Abras, and Nathan Hariharan					5d. PROJECT NUMBER	
					5e. TASK NUMBER	
					5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Engineer Research and Development Center Information Technology Laboratory 3909 Halls Ferry Road Vicksburg, MS 39180					8. PERFORMING ORGANIZATION REPORT NUMBER ERDC/ITLMP-21-4	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Corps of Engineers Washington, DC 20314					10. SPONSOR/MONITOR'S ACRONYM(S)	
					11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.						
13. SUPPLEMENTARY NOTES This paper was originally published in ISAV 2020 Workshop proceedings on November 16, 2020.						
14. ABSTRACT We present a new batch volume rendering technique which alleviates the time and expertise needed by the domain scientist in order to produce quality volume rendered results. This process can be done both in situ and as a post-processing step. The advantage of performing this as an <i>in situ</i> process is that the user is not required to have a <i>priori</i> knowledge of the exact physics and how best to create a transfer function to volume render that physics during the in situ run. For the post-processing use case, the user has the ability to easily examine a wide variety of transfer functions without the tedious work of manually generating each one.						
15. SUBJECT TERMS In situ analysis and visualization, volume rendering, HPC						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT SAR	18. NUMBER OF PAGES 11	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (include area code)	