PAVE

Samuel Leventhal
samlev@cs.utah.edu
University of Utah School of Computing: Scientific Computing and Imaging Institute

Mark Kim
kimmb@ornl.gov
Oak Ridge National Lab

Dave Pugmire*
pugmire@ornl.gov
Oak Ridge National Lab

Figure 1: Rendered Conditional Images

ABSTRACT

In this work we offer an approachable platform for visualization tasks by employing a neural network for real time rendering and accurate light transport simulation within the framework of Python made compatible for distributed systems and high performance computing (HPC). The provided model is a coalescence of VTK-m, a visualization tookit fit for massively threaded architectures, PyTorch, an increasingly popular language within machine learning due to robust libraries for neural networks, and Adios, an adaptable unified IO framework for data managment at scale. The resulting work accomplishes this combination by utilizing VTK-m to construct a path trace renderer able to fluidly and efficiently communicate to a conditional Generative Advisarial Network (cGAN) by means of Adios during training cullminating in a generative model which serves as a filter for rendered images and visual simulations capable of approximating indirect illumination and soft shadows at real-time rates while maintaining quality comparable to offline approaches.

CCS CONCEPTS

• **Theory of computation** → **Parallel computing models**; **Distributed computing models**; **Structured prediction**; **Adversarial learning**; **Data structures and algorithms for data management**; *Probabilistic computation*; *Database query languages (principles)*; • **Applied computing** → **Computer-aided design**.

KEYWORDS

VTKm, neural networks, generative adversarial network, Adios, PyTorch, path tracing

ACM Reference Format:
Samuel Leventhal, Mark Kim, and Dave Pugmire. 2019. PAVE. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Unpublished working draft. Not for distribution.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference'17, July 2017, Washington, DC, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 APPLICABLE "AREA OF INTERESTS" TARGETS

(1) In situ data management and infrastructures Current Systems: production quality, research prototypes , Opportunities ,Gaps
Current Systems: integration of VTKm, Adios2 and Python (PyTorch). Prototype being a conditional generative adversarial network (cGAN) designed to use a VTKm based pathtracer applied but not limited to learning global illumination and light behavior in rendering tasks. Opportunities: Introducing a framework allowing researchers easy access to python on HPC systems as well as machine learning aided technique to treat and study experimental data used in scientific simulations as learnable probability distributions with derived conditional dependencies of interest.
(2) System resources, hardware, and emerging architectures. Enabling Hardware, Hardware and architectures that provide opportunities for In situ processing, such as burst buffers, staging computations on I/O nodes, sharing cores within a node for both simulation and in situ processing
Enabling Hardware: By constructing an architecture allowing for Python to interface with VTKm data management controlled by Adios2 the proposed software

allows for a well distributed simulation task among cores.

(3) Methods and algorithms: Analysis: feature detection, statistical methods, temporal methods, geometric and topological methods Visualization: information visualization, scientific visualization, time-varying methods
(4) Case Studies and Data Sources In situ methods/systems applied to data from simulations and/or experiments / observations
(5) Simulation and Workflows: Integration:data modeling, software-engineering, Workflows for supporting complex in situ processing pipelines
(6) Requirements, Usability: Reproducibility, provenance and metadata

2 INTRODUCTION

3 RELATED WORK

Tomas and Forbes Deep Illumination: [5]

VTKm [3]

Reinforced learning for light transport simulation [1]

4 TECHNIQUE OVERVIEW

Utilization of PAVE consists of three consecutive phases: rendering phase of conditional training images, training phase of the generative neural network, and execution phase of the trained network. Three core components, VTK-m, PyTorch, and Adios2 fufill a unique functional requirement during each stage. In this section we describe the independent design and global role of each system plays.

4.1 System Overview

To achieve our goal of a conditional generative neural network capable of rendering geometric dependent object path simulations we begin by rendering informative conditional image buffers along with ground truth scene renderings. For this purpose the VTK-m was chosen due to its scalability and robust capability for HPC visualization tasks. Provided the training set of conditional and ground truth images two neural networks, one convolutional and one generative, play a zero-sum game common to training GANs. To segue data managment of training images the path tracer saves the training set in a distributed setting with the use of Adios2. During training PyTorch is then able to retrieve needed image data through the use of the adaptable IO provided by Adios's Python high-level APIs.

4.2 Path Tracer Design

4.3 Neural Network Design

The cGAN used closely follows that introduced by Thomas and Forbes with Deep Illumunation [5]. Both the desriminator and generator network are deep convolutional neural networks implemented in PyTorch using training data retrieved from Adios files formated and stored by the VTK-m path tracer. The training stage relies on four conditional buffers depth, albedo, normals and direct lighting along with an associated ground truth image of high light sample count and ray depth. Given the four conditional buffers the generator attempts to construct the ground truth image from noise. The discriminator is then fed both the generated and ground truth image. The loss used for the gradient backpropagation update of both networks is based on the quality of the descriminators ability to classify the artifical and true image in which the generator is greater penalized when the discriminator accurately differentiates the two images, and similarly, the discriminator has a larger loss when incorrectly identifying real from fabricated images. The generator is then considered to have converged when the descriminator predicts both generated and true images with equal probability.

4.3.1 Descriminator Network. For descriminating between artificial and ground truth image renderings a deep convolutional patchGAN network is used motivated by the added advantage of providing a patch-wise probability of an image in question as being real or fake. The benefit of a patch-wise probability allows for higher regional accuracy within an image as well as applicable for image-to-image tasks as introduced by Isola et. al. [2].

4.3.2 Generator Network. The generative network used is a deep convolutional network consisting of an encoder and decoder with skip connections concatonating equal depth layers of the encoding and decoding stages. Due to the illustrative 'shape' of this design the network is denoted a U-Net as introduced by Ronneberger et. al. for medical segmentation [4]. The motivation for utilizing a U-Net is due to success of the skip connections linking the decoded convolutional process to the encoded deconvolutional in capturing geometric and spatial attributes.

5 EXPERIMENTS

5.1 Cornell Box

5.2 Streamline Simulation

6 RESULTS

7 CONCLUSIONS

ACKNOWLEDGMENTS

Identification of funding sources and other support, and thanks to individuals and groups that assisted in the research and the preparation of the work should be included in an acknowledgment section, which is placed just before the reference section in your document.

REFERENCES

[1] Ken Dahm and Alexander Keller. 2017. Learning light transport the reinforced way. *arXiv preprint arXiv:1701.07403* (2017).
[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 1125–1134.
[3] Kenneth Moreland, Christopher Sewell, William Usher, Li-ta Lo, Jeremy Meredith, David Pugmire, James Kress, Hendrik Schroots, Kwan-Liu Ma, Hank Childs, et al. 2016. Vtk-m: Accelerating the visualization toolkit for massively threaded architectures. *IEEE computer graphics and applications* 36, 3 (2016), 48–58.

Unpublished working draft. Not for distribution.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.
[5] Manu Mathew Thomas and Angus G Forbes. 2017. Deep Illumination: Approximating Dynamic Global Illumination with Generative Adversarial Network. *arXiv preprint arXiv:1710.09834* (2017).

A APPENDIX

Unpublished working draft.
Not for distribution.