Material Interface Reconstruction using Machine Learning

1st Daniel Fenn
Computing Directorate
Lawrence Livermore National Lab
Livermore, USA
fenn1@llnl.gov

2nd Katie Lewis
Computing Directorate
Lawrence Livermore National Lab
Livermore, USA
lewis66@llnl.gov

3rd Charles Doutriaux
Computing Directorate
Lawrence Livermore National Lab
Livermore, USA
doutriaux1@llnl.gov

I. Introduction

When simulating multiple materials, the ability to accurately track the interfaces between the materials is of special importance. On discretized domains where material boundaries are not explicitly tracked, these interfaces must be locally constructed from the volume-averaged information provided by computational zones. This process of material interface reconstruction (MIR) is highly underconstrained, leading to difficulties in developing generalized algorithms for simulation and visualization codes [1]–[3].

Various approaches have been used to address this need. Popular methods such as Simple Linear Interface Construction (SLIC) [4] and Piecewise Linear Interface Construction (PLIC) [5], conserve volume fraction, but the linear, zone-wise nature of the reconstruction results in interfaces that are discontinuous across zone boundaries. Other methods [3], [6] enforce interface continuity, but sacrifice conservation.

Variations of these schemes are in common use, but encounter documented shortcomings [7]. Methods that conserve volume fraction sacrifice interface continuity, and vice versa [3]. Additionally, PLIC-type methods may encounter difficulty in zones with more than two materials, in which interfaces are misplaced. Since the interface locations are used in flux calculations to preferentially advect the correct materials into or out of the zone, these shortcomings have deleterious effects on simulation accuracy.

Advancements in image-based machine-learning (ML) and computer vision have yielded promising results regarding image segmentation [8], denoising [9], [10], and generation and reconstruction [11]. These topics bear conceptual similarity to the process of material interface reconstruction, and may have application as a complementary solution method.

In this work, we give an overview of our investigation of ML methods to MIR. We describe our computational model, along with methods for generating synthetic training data. We show some preliminary results, and highlight model features that contribute to successful outcomes.

II. Models and Methods

Many ML methods exist with potential applications to MIR. For this investigative study, we utilized image-based supervised-learning methods [12]. Supervised learning is a widely used, well-developed branch of ML, with many established techniques and extensive, mature libraries. Likewise, image-based ML has been broadly studied, and is supported by a robust software and hardware (including GPU) ecosystem. Given that we are not studying ML per se, but rather its application to an existing field, the ability to utilize pre-existing ML resources was an important factor.

A. Overview

The MIR problem is fundamentally mesh-based. In order to utilize image-based methods, a mesh→image mapping must be performed. Suppose we are provided with a mesh containing zones with mixed materials in which we would like to reconstruct the material interfaces. The basis of our approach was to perform a grid-like sample over all or part of the domain. At each sampled point, the corresponding material composition was encoded as a pixel (or voxel, for 3D) value. For example, in the simple case of two materials, each pixel would take a value in the range $[0, 1]$, where the value would correspond to the volume fraction of one of the materials. If more than two materials were present, additional image channels (RGB) could be utilized. The resultant pixels were then used to construct an image, which was fed to an ML network. The network used the image to infer the most likely interface morphology, which it returned as another image. An inverse mapping was then performed from the image back onto the mesh to construct the interface. The process may be summarized:

mesh→image→network→image→mesh

We limited this study to meshes containing two materials. Image pixel values are determined by material composition, which must be calculated at each pixel location. In preliminary evaluations, we determined that performing PLIC reconstruction on the mesh prior to generating the image offered a small advantage over using volume fractions directly. This resulted in images where pixels represented pure material values, and were either black or white.

B. Network architecture

We used an autoencoder [13] design with symmetric layers of convolution and deconvolution. The network took an image as input, and returned an inferred image as output. At the center lay three fully-connected layers. Our network used batch normalization and no pooling. Experimentation showed that outcomes were sensitive to the number of neurons in the middle layer, with optimal results achieved with 1024 for this application. We implemented the network using the Julia library Flux [14]–[16]. Note that this design was not explicitly conservative.

C. Data generation

Training in supervised learning requires that each sample fed to the network be comprised of two parts: input that the network uses to perform inference, and the optimal solution corresponding to that input–the "ground truth." In our case, the input is the image generated from the mesh data, and the ground truth is the image representing the corresponding correct interface morphology.

For the purposes of simplicity and controlled experimentation, we generated synthetic data for training, rather than using simulation data. This allowed us to create exact ground truth representations for the input images, and enabled the straightforward generation of arbitrary quantities of training data. It also allowed us to control the characteristics of the training data to gain insight into network performance.

We constructed a database of 15000 pairs of images with resolution 256×256 for training and testing. For each pair, we constructed the ground truth image first by randomly superimposing ellipses and polygons of varying sizes, orientations, and colors (black/white) to create a source mesh $\mathcal{M}_s$ with exact, analytically-defined interfaces, and exporting $\mathcal{M}_s$ as an image. To create an image representing a simulation mesh with undefined interfaces, we generated a target mesh $\mathcal{M}_t$ with a randomly-chosen mesh topology. We then mapped the mesh fields (specifically, those indicating material composition) from $\mathcal{M}_s$ to $\mathcal{M}_t$, thereby losing the interface information. We used the mapped mesh to perform a PLIC reconstruction, from which the network input image was obtained.

D. Loss functions

We additionally analyzed the performance of two different loss functions; results will be discussed in Section III. In both versions, the loss was given by the pixelwise MSE when compared against the ground-truth image. In the baseline version, the error from all pixels was weighted uniformly. For most images, the majority of the pixels belonged to pure zones, so the error was driven principally by zones containing no interfaces. In the modified version of the loss function, pixels belonging to zones containing mixed materials were weighted more heavily than those in pure zones, so that the error was largely driven by regions containing interfaces.

III. Results and Discussion

This investigative study sought to determine the feasibility of applying ML to MIR; the preliminary results show promise. Our results indicated that the modified loss function described in Section II-D performed notably better than the baseline. This was true particularly in regard to narrow material structures, where the total contribution to the error is small due to the small number of pixels comprising these areas. We additionally observed notable improvement in the interfaces themselves, which were less diffused in the modified network.

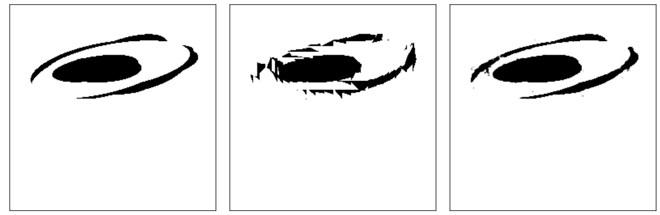

Fig. 1. A selected sample demonstrating network output. Left: ground truth image, constructed by superimposing randomized shapes; middle: input image to network, created by mapping the ground-truth data to a mesh and performing a PLIC reconstruction; right: network reconstruction.

A result obtained using the modified loss function is shown in Fig. 1, which compares the ground truth, input, and network output images for a selected sample. This sample is representative of typical results obtained from our network. Here, as in most cases, the network successfully reconstructs large and small-scale morphological features, with typical error of less than 1%. The network is highly successful in providing continuous reconstructions. Conservation is not explicitly enforced, but given the high fidelity of the reconstructions, is typically well-preserved.

These results come with some caveats, which will be addressed in future work. The network sometimes performs poorly for long thin structures, for which the PLIC representations are substantially different than the correct morphology. Furthermore, our synthetic data may not be representative of actual simulation data in some ways. Particularly, the synthetic data is comprised of a superposition of shapes with well-defined characteristics. It is possible that the network is able to learn those characteristics and is overfit to this specific form of interface morphology. Additionally, we have limited our attention to two-material problems; it is as-yet unclear how well the network would perform with more materials. Future work will use real simulation data representative of problems of interest to train and assess network performance. In addition, we will incorporate physics variables from the mesh into the training to enforce conservation and aid in inference. Our encouraging preliminary results demonstrate a promising approach, and we expect future work to yield positive results.

References

[1] J. C. Anderson, C. Garth, M. A. Duchaineau, and K. I. Joy, "Discrete multi-material interface reconstruction for volume fraction data," *Computer Graphics Forum*, vol. 27, no. 3, pp. 1015–1022, 2008. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2008.01237.x

[2] L. Hüttenberger, H. Obermaier, C. Garth, K. I. Joy, and H. Hagen, "Topological aspects of material interface reconstruction: Challenges and perspectives," 2013.

[3] J. S. Meredith and H. Childs, "Visualization and analysis-oriented reconstruction of material interfaces," *Computer Graphics Forum*, vol. 29, no. 3, pp. 1241–1250, 2010. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2009.01671.x
[4] W. F. Noh and P. Woodward, *SLIC (Simple Line Interface Calculation)*, 1976, vol. 59, p. 330.
[5] D. Youngs, *Time-Dependent Multi-material Flow with Large Fluid Distortion*, 01 1982, vol. 24, pp. 273–285.
[6] K. S. Bonnell, D. R. Schikore, K. I. Joy, M. Duchaineau, and B. Hamann, "Constructing material interfaces from data sets with volume-fraction information," in *Proceedings Visualization 2000. VIS 2000 (Cat. No.00CH37145)*, 2000, pp. 367–372.
[7] M. Kucharik, R. V. Garimella, S. P. Schofield, and M. J. Shashkov, "A comparative study of interface reconstruction methods for multi-material ale simulations," *Journal of Computational Physics*, vol. 229, no. 7, pp. 2432 – 2452, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999109003891
[8] S. Minaee, Y. Boykov, F. M. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *ArXiv*, vol. abs/2001.05566, 2020.
[9] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, "Deep learning on image denoising: An overview," *ArXiv*, vol. abs/1912.13171, 2019.
[10] X.-J. Mao, C. Shen, and Y.-B. Yang, "Image restoration using convolutional auto-encoders with symmetric skip connections," 2016.
[11] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.
[12] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, http://www.deeplearningbook.org.
[13] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, vol. 37, no. 2, pp. 233–243, 1991. [Online]. Available: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690370209
[14] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM review*, vol. 59, no. 1, pp. 65–98, 2017. [Online]. Available: https://doi.org/10.1137/141000671
[15] M. Innes, "Flux: Elegant machine learning with julia," *Journal of Open Source Software*, 2018.
[16] M. Innes, E. Saba, K. Fischer, D. Gandhi, M. C. Rudilosso, N. M. Joy, T. Karmali, A. Pal, and V. Shah, "Fashionable modelling with flux," *CoRR*, vol. abs/1811.01457, 2018. [Online]. Available: https://arxiv.org/abs/1811.01457

Disclaimer

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.