Learning to generate Signed Distance Fields from Voxels

Yihuan Mao
Tsinghua University
maoyh16@mails.tsinghua.edu.cn

Amlan Kar
University of Toronto
amlan@cs.toronto.edu

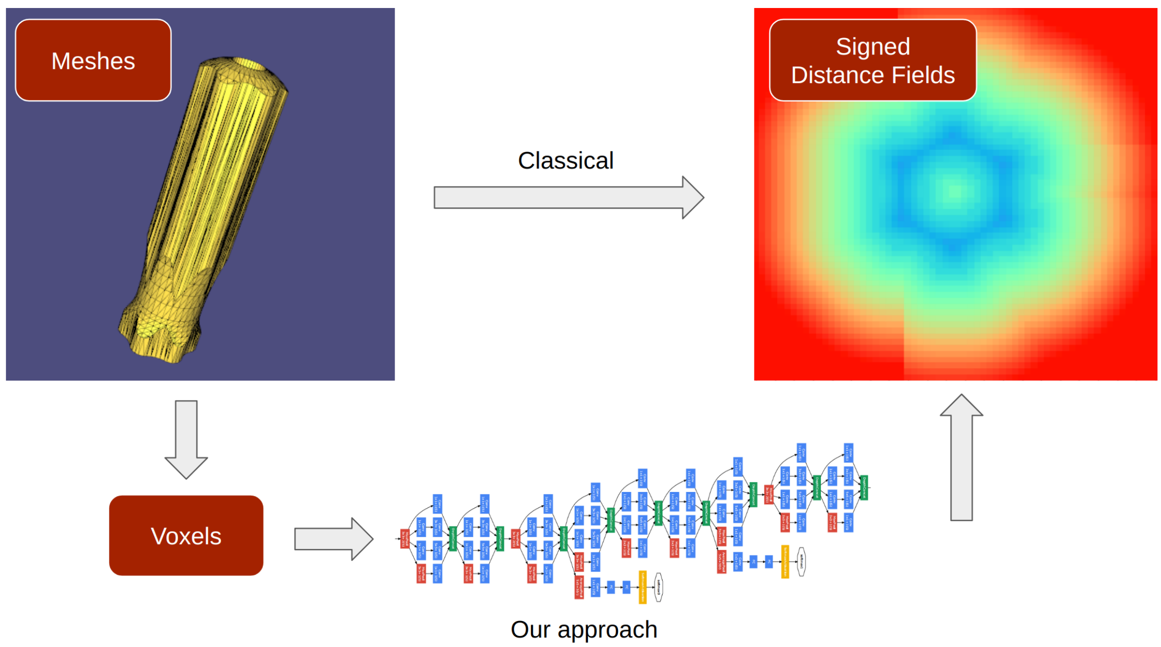

Figure 1: Overview of the task and approach

ABSTRACT

Signed Distance Fields (SDFs) are an essential intermediate representation in a lot of Geometry Processing applications. We explore the usage of Deep Learning techniques to predict a signed (or truncated signed) distance field of a 3d object from its voxelized representation and discuss results.

KEYWORDS

signed distance fields, voxels, deep learning

ACM Reference Format:
Yihuan Mao and Amlan Kar. 2018. Learning to generate Signed Distance Fields from Voxels. In *Proceedings of CSC2520: Geometry Processing*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM ISBN 978-1-4503-1234-5/17/07.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Signed distance fields (SDFs) are used in geometry processing applications such as generating offset surfaces, sculpting, CNC milling, etc. [Frisken and Perry 2006]. SDFs are a particular kind of implicit surface function [Thorpe 2012], which is why they have also been used widely with the Level Set evolution technique [Osher and Sethian 1988] (the 0-level of the SDF exactly represents the surface). Algorithms used to compute the SDF from a 3D mesh are often slow, or require the mesh to have certain restrictive properties. Deep Learning methods have recently shown promise in solving 3D-to-3D semantic tasks such as segmentation [Çiçek et al. 2016] etc. The task of generating a SDF from an imperfect/incomplete mesh also requires a certain degree of semantic reasoning. Motivated by the ability of deep learning methods to 'hallucinate' outputs semantically from incomplete/imperfect inputs, we attempt to use a deep learning based approach for the task.

2 GETTING DATA

Voxelization of meshes can be done using the Raycasting method. This method works for meshes without self-intersections, but usually requires a lot of rays to generate correct results. The winding number is an attractive option to compute whether a point is inside or outside a mesh. However, the winding number method works only for closed meshes. The generalized winding number [Jacobson et al. 2013] is a relaxation of the winding number which is exact for closed meshes and provides a good approximation otherwise. Since we aim to work with imperfect meshes, we use the generalized winding number to obtain a voxelization of a given mesh. To improve computational efficiency, we also utilize a flood-fill algorithm and only use the generalized winding number near the boundaries of the object. For the data, we use the Thingi10K dataset [Zhou and Jacobson 2016], which is a collection of 10K 3D-printing modelour use. s openly available for research. It contains many meshes with imperfections including holes, self-intersections etc. making it ideal for our use. **Yihuan: talk about filtering**. We then split the dataset into 90% training and 10% testing.

3 3D-UNET

We propose to use a neural network to learn the transformation from a (possibly incomplete) voxel representation of an object to its signed distance field (implicit) representation. Specifically, we use the 3D-Unet [Çiçek et al. 2016] architecture, that has been shown to work very well in the medical image segmentation domain. The network consists of 3D convolutional layers, which first encode the input volume, dubbed the 'context' pathway. Next, the 'localization' pathway uses transposed convolutional layers to upsample the encoded volume back into its original dimensions as the output. Additionally, intermediate features from the context pathway are concatenated to the intermediate features in the localization pathway (as context). Fig. 4 shows the exact architecture used. Notice the U-shape, which is where the name U-Net comes from.

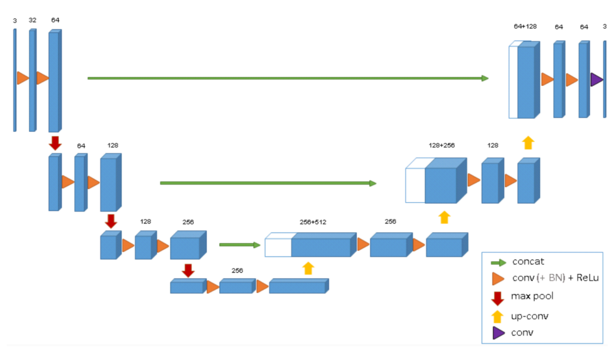

Figure 2: 3D-Unet architecture

3.1 Losses and Training

Our model is trained to regress to the signed distance field. Recent work has shown that binning final values and treating it as a classification produces better results sometimes, we decide to stick with regression, since downstream tasks using the signed distance field would usually require more precision than what one would obtain after binning. The model is trained to minimize the mean-squared-error between the predicted and ground truth SDFs. In our experiment, we just skip the meshes with extremely large number of triangles for efficiency. We also experiment with training with a truncated SDF (TSDF), since SDFs beyond a certain distance aren't useful in general, while the network spends resources trying to get those values right. We see this results in better final error in 4.1.

4 RESULTS

In this section, we describe the results obtained in our project. First, we show quantitative results, specifically, the Mean Squared Error (MSE) on the test set of models trained using various schemes. Next, we show qualitative results and finally provide an analysis on the run-times of various methods.

I order to obtain better performance on the boundary of objects, we define TSDF(Truncated Sigened Distance Field), which clamp the original SDF between a smaller interval. In the visualization part, we'll see the difference.

4.1 Loss table

Method	MSE Training Loss	MSE Testing Loss
SDF	7.3	9.6
TSDF	2.6	3.7

Table 1: Mean Squared Error (MSE) on the testing set of various models

In Table 1, we get a fairly small MSE loss. And it's natural that the loss of TSDF is smaller, because the values become smaller.

4.2 Visualization

Both the SDF and TSDF obtained from neural network are similar to the ground truth, but TSDF shows a nicer boundary in our experiment.

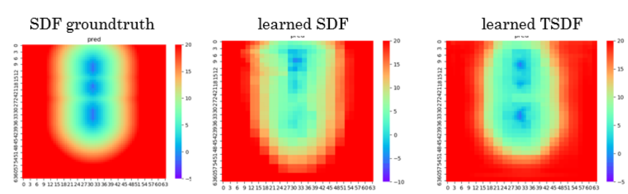

Figure 3: Visualization of results

4.3 Runtime

We compare runtimes of using the pseudonormal test vs using voxelization+ML (our approach) using various meshes. The x-axis denotes the number of faces in the mesh. Note that there was no parallelization in the implementations of the pseudonormal test, as well as the voxelization.

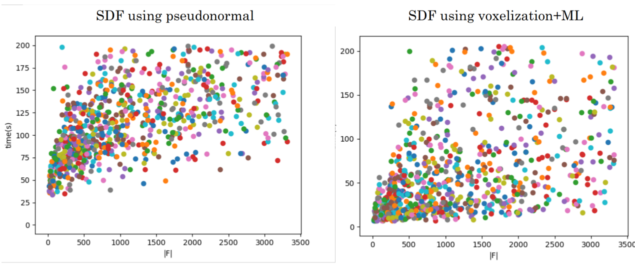

Figure 4: Runtime comparison

ACKNOWLEDGMENTS

The authors would like to thank Gavin Barill for discussions and Alec Jacobson for proposing to try this line of work as a project.

REFERENCES

Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 2016. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 424–432.

Sarah F Frisken and Ronald N Perry. 2006. Designing with distance fields. In *ACM SIGGRAPH 2006 Courses*. ACM, 60–66.

Alec Jacobson, Ladislav Kavan, and Olga Sorkine-Hornung. 2013. Robust Inside-Outside Segmentation using Generalized Winding Numbers. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)* 32, 4 (2013), 33:1–33:12.

Stanley Osher and James A Sethian. 1988. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of computational physics* 79, 1 (1988), 12–49.

John A Thorpe. 2012. *Elementary topics in differential geometry*. Springer Science & Business Media.

Qingnan Zhou and Alec Jacobson. 2016. Thingi10K: A Dataset of 10,000 3D-Printing Models. *arXiv preprint arXiv:1605.04797* (2016).