

FastGeodis: Fast Generalised Geodesic Distance Transform[★]

Muhammad Asad^{1[0000–0002–3672–2414]}, Reuben Dorent^{1[0000–0002–7530–0644]},
and Tom Vercauteren^{1[0000–0003–1794–0456]}

School of Biomedical Engineering & Imaging Sciences, King’s College London, UK
`muhammad.asad@kcl.ac.uk`

Abstract. The *FastGeodis* package provides an efficient implementation for computing Geodesic and Euclidean distance transforms (or a mixture of both) targeting efficient utilisation of CPU and GPU hardwares. In particular, it implements parallelisable raster scan method from [4], where elements in row (2D) or plane (3D) can be computed with parallel threads. This package is able to handle 2D as well as 3D data where it achieves up to 15x speed-up on CPU and up to 60x speed-up on GPU as compared to existing open-source libraries [9], which uses non-parallelisable single-thread CPU implementation. The performance speed-ups reported here were evaluated using 3D volume data on Nvidia GeForce Titan X (12 GB) with 6-Core Intel Xeon E5-1650 CPU. This package is available at: <https://github.com/masadcv/FastGeodis>

Summary

Geodesic and Euclidean distance transforms have been widely used in a number of applications, where distance from a set of reference points is computed. Methods from recent years have shown effectiveness in applying Geodesic distance transform to interactively annotate 3D medical imaging data [10,3], where it enables providing segmentation labels, i.e. voxel-wise labels, for different objects of interests. Despite existing methods for efficient computation of Geodesic distance transform on GPU and CPU devices [4,3,11,8], an open-source implementation of such methods does not exist. On the contrary, efficient methods for Euclidean distance transform [5] have open-source implementations [1,7]. Existing libraries, e.g. [9], provide C++ implementations of Geodesic distance transform, however they lack efficient utilisation of the underlying hardware and hence result in significant computation time especially when applying them on 3D medical imaging volumes.

The *FastGeodis* package provides an efficient implementation for computing Geodesic and Euclidean distance transforms (or a mixture of both) targeting efficient utilisation of CPU and GPU hardwares. In particular, it implements parallelisable raster scan method from [4], where elements in row (2D) or plane (3D) can be computed with parallel threads. This package is able to handle 2D

[★] Under-review at Journal of Open Source Software (JOSS)

as well as 3D data where it achieves up to 15x speed-up on CPU and up to 60x speed-up on GPU as compared to existing open-source libraries [9], which uses non-parallelisable single-thread CPU implementation. The performance speed-ups reported here were evaluated using 3D volume data on Nvidia GeForce Titan X (12 GB) with 6-Core Intel Xeon E5-1650 CPU. Further in-depth comparison of performance improvements is discussed in *FastGeodis* documentation.

Statement of Need

Despite existing open-source implementation of distance transforms [1,7,9], open-source implementations of efficient Geodesic distance transform algorithms [4,11] on CPU and GPU do not exist. However, for Euclidean distance efficient CPU [7] and GPU [1] implementations exist. To the best of our knowledge, *FastGeodis* is the first open-source implementation of efficient Geodesic distance transform [4], achieving up to 15x speed-up on CPU and up to 60x speed-up on GPU as compared to existing open-source libraries [9]. It also provides efficient implementation of Euclidean distance transform. In addition, it is the first open-source implementation of generalised Geodesic distance transform and Geodesic Symmetric Filtering (GSF) proposed in [3].

The ability to efficiently compute Geodesic and Euclidean distance transforms can significantly enhance distance transform applications especially for training deep learning models that utilise distance transforms [10]. It will improve prototyping, experimentation, and deployment of such methods, where efficient computation of distance transforms has been a limiting factor. In 3D medical imaging problems, efficient computation of distance transforms will lead to significant speed-ups, enabling online learning applications for better processing/labelling/inference from volumetric datasets [2]. In addition, *FastGeodis* provides efficient implementation for both CPUs and GPUs hardware and hence will enable efficient use of a wide range of hardware devices.

Methodology

FastGeodis implements an efficient distance transform algorithm from [4], which provides parallelisable raster scans to compute distance transform. The implementation consists of data propagation passes parallelised using threads for elements across a line (2D) or plane (3D). Figure 1 shows these data propagation passes, where each pass consists of computing distance values for next row (2D) or plane (3D) by utilising parallel threads and data from previous row/plane, hence resulting in propagating distance values along the direction of pass. For 2D data, four distance propagation passes are required, top-bottom, bottom-top, left-right and right-left, whereas for 3D data six passes are required, front-back, back-front, top-bottom, bottom-top, left-right and right-left. The algorithm can be applied to efficiently compute both Geodesic and Euclidean distance transforms. In addition to this, *FastGeodis* also provides non-parallelisable raster scan

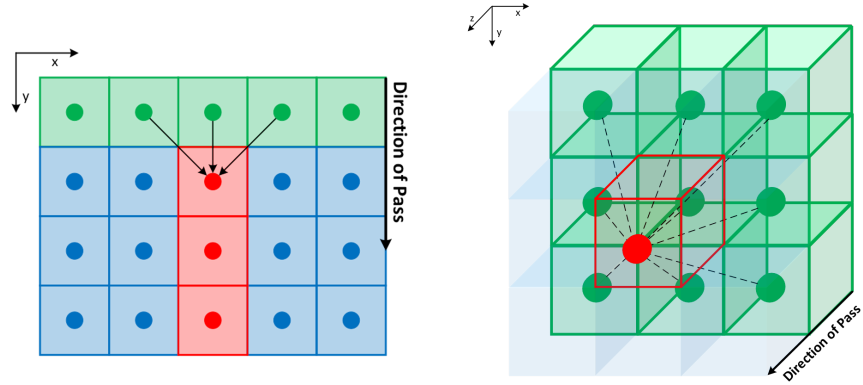


Fig. 1. Raster scan data propagation passes in FastGeodis.

based distance transform method from [8], which is implemented using single-thread CPU and used for comparison.

FastGeodis package is implemented using *PyTorch* [6] utilising OpenMP for CPU and CUDA for GPU parallelisation of the algorithm. It is accessible as a python package, that can be installed across different operating systems and devices. A comprehensive documentation and a range of examples are provided for understanding the usage of the package on 2D and 3D data using CPU or GPU. The provided examples include 2D/3D examples for Geodesic, Euclidean, Signed Geodesic distance transform as well as computing Geodesic symmetric filtering (GSF) that is essential first step in implementing interactive segmentation method from [3]. Further in-depth overview of implemented algorithm, along with evaluation on common 2D/3D data input sizes is provided in *FastGeodis* documentation.

Acknowledgements

This research was supported by the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101016131.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), <https://www.tensorflow.org/>, software available from tensorflow.org

2. Asad, M., Fidon, L., Vercauteren, T.: Econet: Efficient convolutional online likelihood network for scribble-based interactive segmentation. arXiv preprint arXiv:2201.04584 (2022)
3. Criminisi, A., Sharp, T., Blake, A.: Geos: Geodesic image segmentation. In: European Conference on Computer Vision. pp. 99–112. Springer (2008)
4. Criminisi, A., Sharp, T., Siddiqui, K.: Interactive geodesic segmentation of n-dimensional medical images on the graphics processor. In: Radiological Society of North America (RSNA) (December 2009)
5. Felzenszwalb, P.F., Huttenlocher, D.P.: Distance transforms of sampled functions. *Theory of computing* **8**(1), 415–428 (2012)
6. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) *Advances in Neural Information Processing Systems* 32, pp. 8024–8035. Curran Associates, Inc. (2019), <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
7. Seung-Lab: Multi-label anisotropic 3d euclidean distance transform (mlaedt-3d) (2018), <https://github.com/seung-lab/euclidean-distance-transform-3d>
8. Toivanen, P.J.: New geodesic distance transforms for gray-scale images. *Pattern Recognition Letters* **17**(5), 437–450 (1996). [https://doi.org/10.1016/0167-8655\(96\)00010-4](https://doi.org/10.1016/0167-8655(96)00010-4)
9. Wang, G.: Geodistk: Geodesic distance transform toolkit for 2d and 3d images (2020), <https://github.com/taigw/GeodisTK>
10. Wang, G., Zuluaga, M.A., Li, W., Pratt, R., Patel, P.A., Aertsen, M., Doel, T., David, A.L., Deprest, J., Ourselin, S., et al.: Deepigeos: a deep interactive geodesic framework for medical image segmentation. *IEEE transactions on pattern analysis and machine intelligence* **41**(7), 1559–1572 (2018)
11. Weber, O., Devir, Y.S., Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Parallel algorithms for approximation of distance maps on parametric surfaces. *ACM Transactions on Graphics (TOG)* **27**(4), 1–16 (2008). <https://doi.org/10.1145/1409625.1409626>