FastGeodis: Fast Generalised Geodesic Distance Transform

Muhammad Asad [1]¶, **Reuben Dorent** [1], **and Tom Vercauteren** [1]

1 School of Biomedical Engineering & Imaging Sciences, King's College London, UK ¶ Corresponding author

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 the Geodesic distance transform to interactively annotate 3D medical imaging data (Criminisi et al., 2008; Wang et al., 2018). The Geodesic distance transform enables providing segmentation labels, i.e., voxel-wise labels, for different objects of interests. Despite existing methods for efficient computation of the Geodesic distance transform on GPU and CPU devices (Criminisi et al., 2008, 2009; Toivanen, 1996; Weber et al., 2008), an open-source implementation of such methods on the GPU does not exist. On the contrary, efficient methods for the computation of the Euclidean distance transform (Felzenszwalb & Huttenlocher, 2012) have open-source implementations (Abadi et al., 2015; Seung-Lab, 2018). Existing libraries, e.g., Wang (2020), provide C++ implementations of the 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 hardware. In particular, it implements the paralellisable raster scan method from Criminisi et al. (2009), where elements in a 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 a 20x speedup on a CPU and up to a 74x speedup on a GPU as compared to an existing open-source library (Wang, 2020) that uses a non-parallelisable single-thread CPU implementation. The performance speedups reported here were evaluated using 3D volume data on an Nvidia GeForce Titan X (12 GB) with a 6-Core Intel Xeon E5-1650 CPU. Further in-depth comparison of performance improvements is discussed in the `FastGeodis` documentation.

Statement of need

Despite existing open-source implementation of distance transforms (Abadi et al., 2015; Seung-Lab, 2018; Wang, 2020), open-source implementations of efficient Geodesic distance transform algorithms (Criminisi et al., 2009; Weber et al., 2008) on CPUs and GPUs do not exist. However, efficient CPU (Seung-Lab, 2018) and GPU (Abadi et al., 2015) implementations exist for Euclidean distance transform. To the best of our knowledge, `FastGeodis` is the first open-source implementation of efficient the Geodesic distance transform (Criminisi et al., 2009), achieving up to a 20x speedup on a CPU and up to a 74x speedup on a GPU as compared to existing open-source libraries (Wang, 2020). It also provides an efficient implementation of the Euclidean distance transform. In addition, it is the first open-source implementation of generalised Geodesic distance transform and Geodesic Symmetric Filtering (GSF) as proposed

in Criminisi et al. ([2008](#)). Apart from a method from Criminisi et al. ([2009](#)), Weber et al. ([2008](#)) present a further optimised approach for computing Geodesic distance transforms on GPUs. However, this method is protected by multiple patents (Bronstein et al., [2013](#), [2015](#), [2016](#)) and hence is not suitable for open-source implementation in the **FastGeodis** package.

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 ([Wang et al., 2018](#)). 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 ([Asad et al., 2022](#)). In addition, FastGeodis provides an efficient implementation for both CPUs and GPUs and hence will enable efficient use of a wide range of hardware devices.

Implementation

FastGeodis implements an efficient distance transform algorithm from Criminisi et al. ([2009](#)), which provides parallelisable raster scans to compute distance transform. The implementation consists of data propagation passes that are 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 the next row (2D) or plane (3D) by utilising parallel threads and data from the previous row/plane, hence resulting in propagating distance values along the direction of the 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 the non-parallelisable raster scan based distance transform method from Toivanen ([1996](#)), which is implemented using a single CPU thread and used for comparison.

The FastGeodis package is implemented using PyTorch ([Paszke et al., 2019](#)), 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. Comprehensive documentation and a range of examples are provided for understanding the usage of the package on 2D and 3D data using CPUs or GPUs. Two- and three-dimensional examples are provided for Geodesic, Euclidean, and Signed Geodesic distance transforms as well as for computing Geodesic Symmetric Filtering (GSF), the essential first step in implementing the interactive segmentation method described in Criminisi et al. ([2008](#)). A further in-depth overview of the implemented algorithm, along with evaluation on common 2D/3D data input sizes, is provided in the FastGeodis [documentation](#).
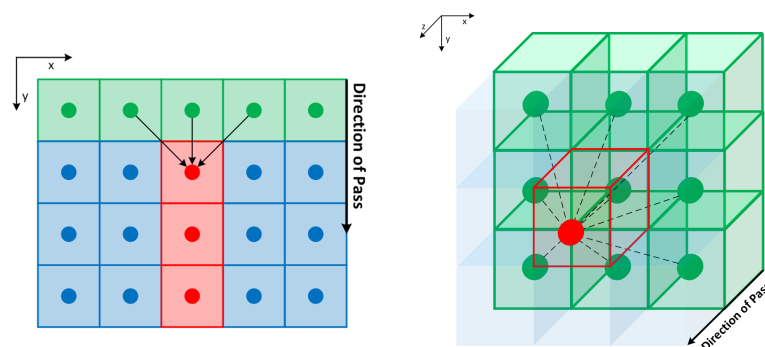

Figure 1: Raster scan data propagation passes in FastGeodis.

Asad et al. (2022). FastGeodis: Fast Generalised Geodesic Distance Transform. *Journal of Open Source Software*, *7*(79), 4532. [https://doi.org/10.21105/joss.04532](#).

Acknowledgements

References

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., … Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems*. https://www.tensorflow.org/

Asad, M., Fidon, L., & Vercauteren, T. (2022). ECONet: Efficient convolutional online likelihood network for scribble-based interactive segmentation. *arXiv Preprint arXiv:2201.04584*.

Bronstein, A., Bronstein, M., Devir, Y., Weber, O., & Kimmel, R. (2013). *Parallel approximation of distance maps (US patent 8,373,716)*. Google Patents.

Bronstein, A., Bronstein, M., Kimmel, R., Devir, Y., & Weber, O. (2015). *Parallel approximation of distance maps (US patent 8,982,142)*. Google Patents.

Bronstein, A., Bronstein, M., Kimmel, R., Devir, Y., & Weber, O. (2016). *Parallel approximation of distance maps (US patent 9,489,708)*. Google Patents.

Criminisi, A., Sharp, T., & Blake, A. (2008). Geos: Geodesic image segmentation. *European Conference on Computer Vision*, 99–112. https://doi.org/10.1007/978-3-540-88682-2_9

Criminisi, A., Sharp, T., & Siddiqui, K. (2009). Interactive Geodesic segmentation of n-dimensional medical images on the graphics processor. *Radiological Society of North America (RSNA)*. https://www.microsoft.com/en-us/research/publication/interactive-geodesic-segmentation-of-n-dimensional-medical-images-on-the-graphics-processor/

Felzenszwalb, P. F., & Huttenlocher, D. P. (2012). Distance transforms of sampled functions. *Theory of Computing*, *8*(1), 415–428.

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., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Seung-Lab. (2018). Multi-label anisotropic 3D Euclidean distance transform (MLAEDT-3D). In *GitHub repository*. GitHub. https://github.com/seung-lab/euclidean-distance-transform-3d

Toivanen, P. J. (1996). New geodosic distance transforms for gray-scale images. *Pattern Recognition Letters*, *17*(5), 437–450. https://doi.org/10.1016/0167-8655(96)00010-4

Wang, G. (2020). GeodisTK: Geodesic distance transform toolkit for 2D and 3D images. In *GitHub repository*. GitHub. https://github.com/taigw/GeodisTK

Wang, G., Zuluaga, M. A., Li, W., Pratt, R., Patel, P. A., Aertsen, M., Doel, T., David, A. L., Deprest, J., Ourselin, S., & others. (2018). DeepIGeoS: A deep interactive geodesic framework for medical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(7), 1559–1572.

Weber, O., Devir, Y. S., Bronstein, A. M., Bronstein, M. M., & Kimmel, R. (2008). Parallel algorithms for approximation of distance maps on parametric surfaces. *ACM Transactions on Graphics (TOG)*, *27*(4), 1–16. https://doi.org/10.1145/1409625.1409626