LIGHTHOUSE: FAST AND PRECISE DISTANCE TO SHORELINE CALCULATIONS FROM ANYWHERE ON EARTH

Patrick Beukema, Henry Herzog, Yawen Zheng, Favyen Bastani
Allen Institute for Artificial Intelligence
correspondence: patrickb@allenai.org

ABSTRACT

We present a new dataset and algorithm for fast and efficient coastal distance calculations from anywhere on Earth (AoE). Previous global coastal distance datasets have been generated at relatively coarse resolution (e.g., 4 km), limiting their utility in many real-world contexts. Publicly available satellite imagery combined with computer vision enable much higher precision. We provide a global coastline dataset at 10 meter resolution, a 400-fold improvement in precision over existing data. To handle the computational challenge of querying at such an increased scale, we introduce *Lighthouse* (Layered Iterative Geospatial Hierarchical Terrain-Oriented Unified Search Engine). Our method is both extremely fast and efficient making it well-suited for real-time inference in resource-constrained environments.

1 INTRODUCTION

Regularly updated and precise sea-land demarcations are essential for many applications such as environmental monitoring, maritime intelligence, and infrastructure planning. For example, in environmental monitoring, accurate shoreline data is crucial for tracking coastal erosion, habitat changes, and the impacts of climate change. Across a wide variety of satellite imagery based computer vision tasks, knowing that an object is on water (versus land) improves precision and recall. The distance to the nearest coastal point can also be used as a feature in maritime GPS behavioral classification models. For some applications, the value of this data scales proportionally with its resolution, particularly for activities nearshore or inland (see figure: 1). Our contributions are twofold:

1. We release a $\sim$10 meter coastal dataset including inland bodies of water.
2. We provide a library that efficiently generates the nearest coastal point from (AoE).

There are several publicly available distance-to-coast datasets and tools (table 1). To our knowledge, the only option that provides distances from all points on earth is a 4 km resolution (resampled to 1 km) dataset produced by NASA Ocean Color (2009).

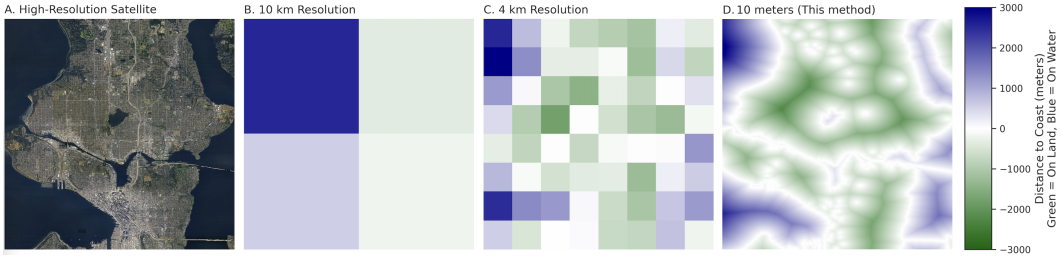

Figure 1: Comparison of distance to coast mapping at progressively higher spatial resolution

Table 1: Existing publicly available coastline datasets

Source	Resolution (m)	Coverage	Additional Notes
Lighthouse	~ 10	Everywhere	resolution is approximate; see sec. 4
NASA	4000	High seas only	also available at 1 km (interpolated)
ArcGIS/ESRI	200000	Land masses only	land to major oceans only
Bing Maps	unavailable	High seas only	see GitHub

We develop an improved global land-sea coastal dataset by selectively merging the European Space Agency's (ESA) WorldCover 2022 (Zanaga et al., 2022), a 10-meter resolution land-cover map derived from Sentinel-2 satellite images, with crowdsourced coastline annotations from OpenStreetMap. WorldCover exhibits high overall accuracy, but misses hundreds of islands in Micronesia and omits Antarctica. OpenStreetMap exhibits lower spatial resolution globally but it covers all islands and includes recent annotations for Antarctica. Thus, merging yields a high-accuracy global map.

However, querying this dataset with the same methods used for 4-km resolution data is cost-prohibitive in both compute and storage. We develop a highly optimized hierarchical search algorithm to reduce these costs. Our approach requires only modest hardware, and processes distance-to-shoreline queries in less than 10 ms.

2 What we did

2.1 Dataset concatenation

High resolution satellite imagery provides a means to generate high resolution coastlines, as the boundary between land and sea is a straightforward application of segmentation via computer vision (supervision on permanent water labels). A variety of global land cover maps have been created in the last several years including ESA's WorldCover (Zanaga et al., 2022), Google's Dynamic World (Brown et al., 2022), and ESRI, Impact Observatory, and Microsoft's LULC map (Karra et al., 2021). We selected ESA's WorldCover V2 because it has been shown to exhibit the highest accuracy for permanent water bodies (Xu et al., 2024). However, ESA omitted several key areas including Antarctica (unavailable at the time of their publication), along with hundreds of islands in Micronesia. We filled in the blanks with Open Street Map's crowdsourced annotations of land-sea labels (OpenStreetMap contributors, 2024). We concatenated these two datasets to complete a map of the planet, resampled both datasets into 1x1 degree tiles, and saved these files to disk or the purpose of parallelization and caching (see algorithm: 2). Supplemental figure 6 shows the distribution of resulting resampled tiles from both sources.

2.2 Coastal point generation

To extract the coastal points from the joint dataset, we binarized the labels (water vs. rest), ran Sobel edge detection over the resulting binary mask, and then constructed balltrees for each tile using the Haversine[1] metric over the coastal points (see algorithm:2.2) [2]. This process was identical for both the Open Street Map tiles and ESA tiles (fig:3).

[1]Note that Vincenty's formula (Vincenty, 1975) is more accurate than Haversine, especially over long distances, but it comes with much greater computational complexity.

[2]Throughout the codebase, we chose options that minimized latency at the expense of an increase in storage. For example, we do not compress the balltrees at all. Doing so would significantly increase the latency for a relatively modest reduction in required disk space. Small increases in latency can kill real-time applications. Disk space is cheap compared to RAM and CPU. We chose h5 for the geotiffs because of the ability to query the land cover class of a single pixel, i.e. without needing to load the entire file into memory.

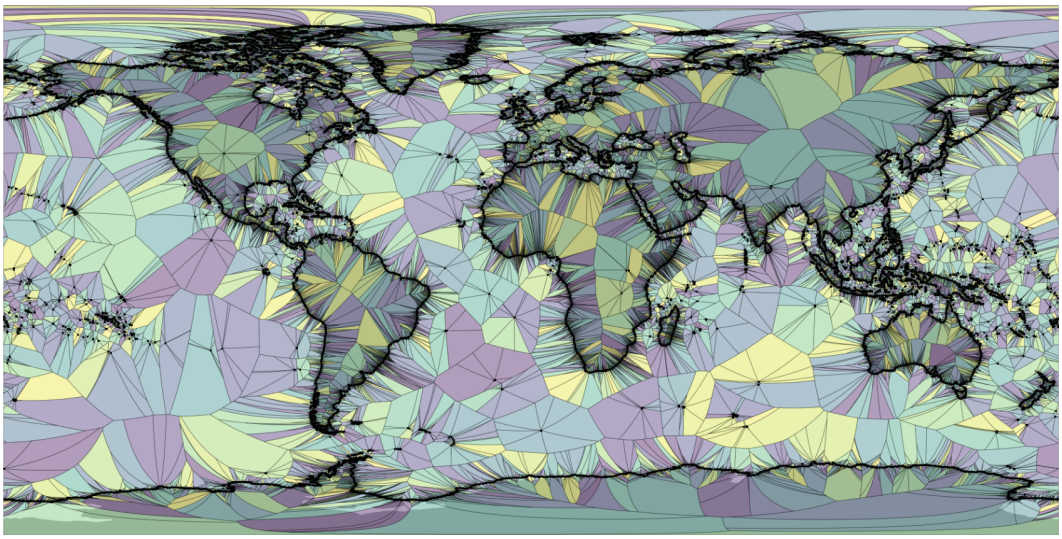

Figure 2: Spherical Voronoi tesselation of the planet based on coastal points.

Algorithm 1 Generate BallTrees of Land-Sea Edges

Data: Open Street Map's land polygons and worldcover data
Result: BallTrees of land-sea edges for each 1x1 degree land tile, saved to disk
Divide the world into 1x1 degree tiles that contain land
for *each land tile* T **do**
    Generate a binary land-sea map for T
    Identify land-sea boundaries via sobel edge detection
    Extract edge coordinates, $edge\_coords$
    Build balltree on $edge\_coords$ using the Haversine metric
    Save balltree (without compression)
end

3 Querying the Data

The nice thing about low resolution data is that you can precompute the distance to every point, store that data in memory, and then retrieve any point on earth in $\mathcal{O}(1)$. But that method doesn't scale to high resolution data. For example, at 10 meter resolution, one would need to store approximately 100 TB of data in RAM (float, float, int, int) which is impractical unless you really don't care about money at all and have a fairly large and idling computer. The solution is to yield the distances at runtime, rather than caching them. To do so efficiently, for real-time applications, you need to search a large space extremely quickly.

Recall that we have high resolution ball trees for every coastal tile (fig. 6), and those can be queried very rapidly for the nearest coastal point. In addition to the nearest coastal point, one must also lookup the location's class label (land vs. water). Doing so rapidly requires storing the land cover maps as h5 files and retrieving just a single point's class, rather than reading the entire tile. With the ball tress and h5 files, one can immediately return the desired distance and class for any point that is contained within a tile (i.e. not on the high seas). If the point lies outside every tile, how do we determine which tile contains the nearest coastal point? The answer is via a spherical Voronoi tesselation, precomputed for the whole planet, which is also loaded in memory at runtime (Tyler Reddy; Van Oosterom & Strackee, 1983; Caroli et al., 2010). We generated this Voronoi tesselation (see fig. (2)) carefully because we could not include every point (temporal complexity scales quadratically) and if any critical point is omitted (such as an island) then the resulting tesselation and resulting distances could be incorrect. Therefore, we down-sampled the coastal points subject to the constraints that 1) every line segment in the original dataset had to be represented by at least one point in the resulting (post-resampled) dataset and 2) that the distance between connected points never exceeded a distance threshold.

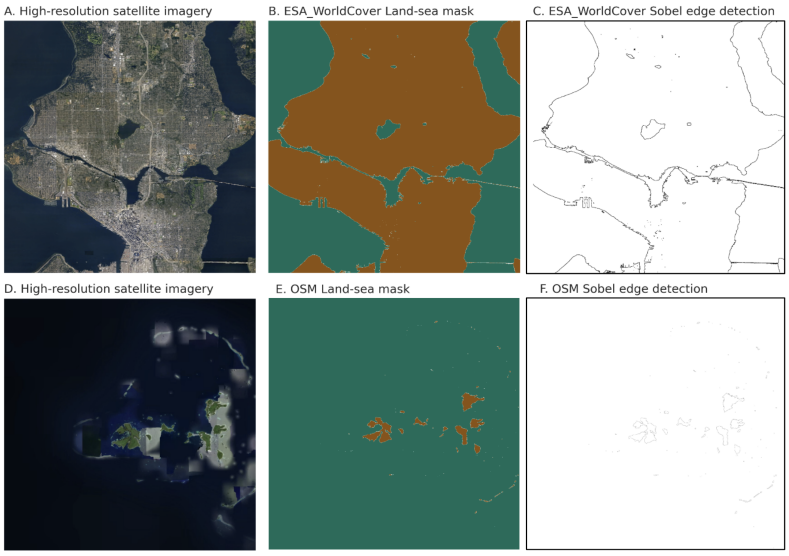

Figure 3: Visual depiction of the method. A, D: High resolution satellite imagery centered on Seattle (A) and a chain of islands in Micronesia (D). B.E: Binary mask constructed from land (vs. sea) labels from ESA and OSM respectively. C, F: result of applying Sobel edge detection to the binary mask. Note faint borders in F.

Algorithm 2 Find Nearest Land Point and Land Cover Class

Initialize an empty tile cache
for *each query point* (lat, lon) **do**
  if (lat, lon) *is contained within one of the existing tile's bounds* **then**
    Load BallTree T and H5 T Land Cover tile
    Query and return distance and land cover class
  if (lat, lon) *is not in any tile's bounds* **then**
    Find the tile T containing (lat, lon) using the Voronoi diagram
    Load BallTree T and H5 T Land Cover tile
    Query and return distance and land cover class
return Results

This method will generate distances on the fly on modest hardware at millisecond timescales. It is not quite $\mathcal{O}(1)$ like a dictionary lookup, but the resulting times are good enough for streaming/real-time inference without breaking the bank.

4 Some caveats

High resolution satellite imagery – the basis of both sets of annotations used here – can facilitate high accuracy but it does not guarantee it. Hybridizing labels from crowd sourced maps (OSM) alongside computer vision from satellite imagery (ESA) will naturally result in a mixture of errors due to human mislabeling and model misclassifications. Consider the challenges of annotating sea-land boundaries, and the complexity of cliffs, beaches, harbors, bays, wetlands, islands, etc.

What is the definition of the coastline anyways? Nearly 50 years ago, Mandlebrot invented a new branch of mathematics, fractal geometry, in part to discuss the complexity of coastlines and the fact that their lengths are effectively infinite (Mandelbrot, 1982). Even today, we hotly debate the true length of a coastline. How far inland should the coastline extend? On top of the challenge of defining it, the earth is changing rapidly. Coastlines change, people build things, sea levels are rising and islands are disappearing, glaciers calve, Antarctica sea ice expands and contracts. The higher the resolution, the greater the opportunity for error.

The 10-meter resolution should be considered an estimate, not a definitive upper bound. The vast majority of the base annotations come from ESA's WorldCover V2, which is derived from Sentinel-2 satellite imagery with a 10-meter pixel resolution. However, there is no single spatial resolution for the OSM labels, as they are generated from a variety of sources using a mix of crowd-sourced and machine annotations. Given this variability, we estimated the inter-label segment distance between neighboring annotations in Antarctica, an area known for limited data, yielding a median of 35 meters (see supplemental figure A). Inter-label segment distance is not a direct measurement of spatial resolution, but it can serve as a useful proxy for estimating effective resolution.

The true resolution is a function of several factors including the base resolution of the satellite imagery, the human and computer vision annotation accuracy, and the complexity of the shoreline (which we do not know at scale). We refer the interested reader to Hormann (2013) and Topf (2013) for more nuanced analyses of the quality of OSM's coastal data and its effective resolution.

5 Conclusion

The coastline is complex, perhaps infinitely so (Mandelbrot, 1982). If even higher precision is desired, this method should scale favorably up to the limit of commercially available satellite imagery (15 cm as of early 2025). Not everyone needs high-resolution coastal data, but for those who do, we have open sourced both our dataset and method under permissive licenses.

1. Dataset: gs://ai2-coastlines/v1/data; (GCP bucket, ODbL)
2. Code: https://github.com/allenai/lighthouse (Apache 2.0)

References

C.F. Brown, S.P. Brumby, B. Guzder-Williams, and et al. Dynamic world, near real-time global 10m land use land cover mapping. *Scientific Data*, 9:251, 2022. doi: 10.1038/s41597-022-01307-4. URL https://doi.org/10.1038/s41597-022-01307-4.

Manuel Caroli, Pedro MM de Castro, Sébastien Loriot, Olivier Rouiller, Monique Teillaud, and Camille Wormser. Robust and efficient delaunay triangulations of points on or close to a sphere. In *Experimental Algorithms: 9th International Symposium, SEA 2010, Ischia Island, Naples, Italy, May 20-22, 2010. Proceedings 9*, pp. 462–473. Springer, 2010.

Christoph Hormann. Assessing the openstreetmap coastline data quality. https://www.imagico.de/map/coastline_quality_en.php, 2013. Accessed: 2025-02-10.

Kontgis Karra et al. Global land use/land cover with sentinel-2 and deep learning. In *IGARSS 2021-2021 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2021.

Benoit B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman, New York, 1982.

NASA Ocean Color. Distance from coastline data documentation, 2009. URL https://oceancolor.gsfc.nasa.gov/resources/docs/distfromcoast/. Accessed: 2024-11-02.

OpenStreetMap contributors. Openstreetmap: Freely Editable Map of the World, 2024. URL https://www.openstreetmap.org. Accessed: 2025-01-31.

Jochen Topf. State of the osm coastline. https://blog.jochentopf.com/2013-03-11-state-of-the-osm-coastline.html, 2013. Accessed: 2025-02-10.

Edd Edmondson Nikolai Nowaczyk Joe Pitt-Francis Tyler Reddy, Ross Hemsley. scipy.spatial.sphericalvoronoi. https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.SphericalVoronoi.html. Accessed: 2025-02-10.

A. Van Oosterom and J. Strackee. The solid angle of a plane triangle. *IEEE Transactions on Biomedical Engineering*, 2(1):125–126, 1983.

Thaddeus Vincenty. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey review*, 23(176):88–93, 1975.

Panpan Xu, Nandin-Erdene Tsendbazar, Martin Herold, Sytze de Bruin, Myke Koopmans, Tanya Birch, Sarah Carter, Steffen Fritz, Myroslava Lesiv, Elise Mazur, et al. Comparative validation of recent 10 m-resolution global land cover maps. *Remote Sensing of Environment*, 311:114316, 2024.

D. Zanaga, R. Van De Kerchove, D. Daems, W. De Keersmaecker, C. Brockmann, G. Kirches, J. Wevers, O. Cartus, M. Santoro, S. Fritz, M. Lesiv, M. Herold, N.E. Tsendbazar, P. Xu, F. Ramoino, and O. Arino. ESA WorldCover 10 m 2021 v200, 2022. URL `https://doi.org/10.5281/zenodo.7254221`.

A APPENDIX

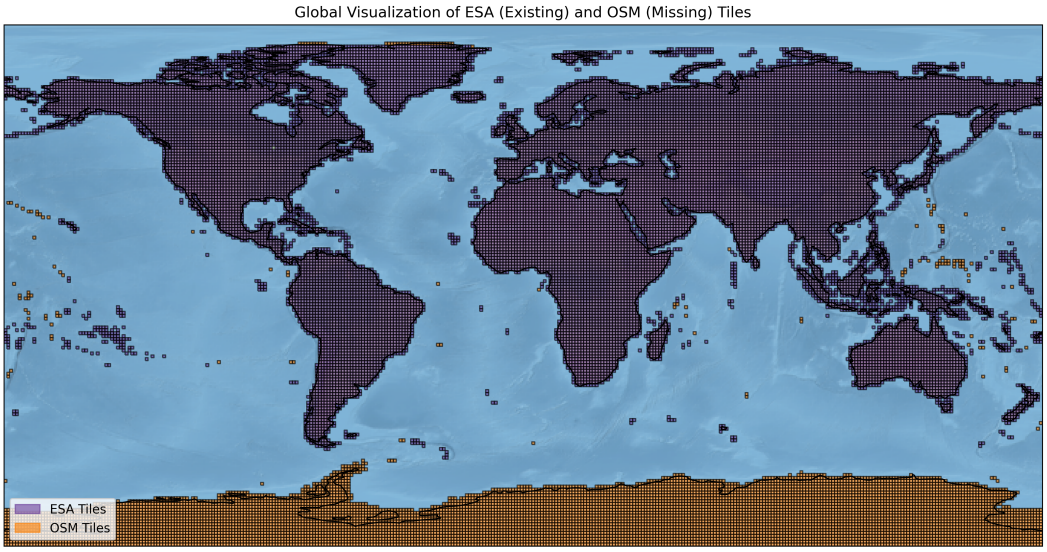

Figure 4: Distribution of 1x1 degree tiles from each data source. Note islands in Micronesia, Hawaii, South Atlantic, and Northern Greenland

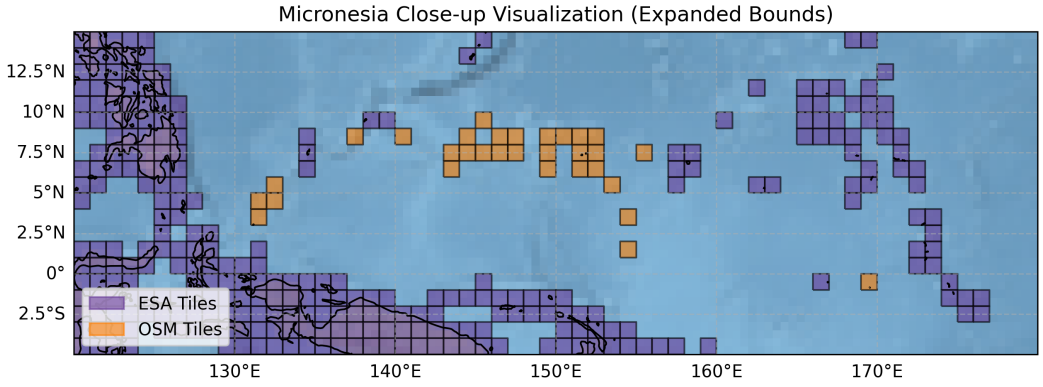

Figure 5: Close up of missing tiles from Micronesia

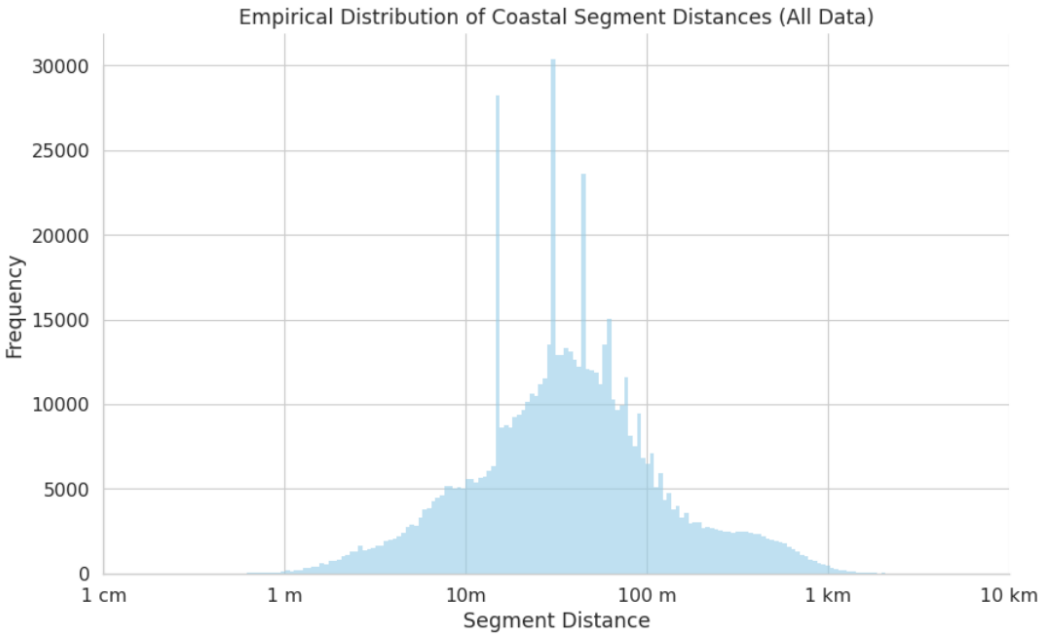

Figure 6: Empirical distribution of coastal segment distances (neighboring points) from Antarctica Open Street Map annotations