Dynamically-Spaced Geo-Grid Segmentation for Weighted Point Sampling on a Polygon Map Layer

Kelly Sims
Oak Ridge National Laboratory, One Bethel Valley Rd, Oak Ridge, TN 37831, U.S.A.
simskm@ornl.gov

Gautam Thakur
Oak Ridge National Laboratory, One Bethel Valley Rd, Oak Ridge, TN 37831, U.S.A.

Kevin Sparks
Oak Ridge National Laboratory, One Bethel Valley Rd, Oak Ridge, TN 37831, U.S.A.

Marie Urban
Oak Ridge National Laboratory, One Bethel Valley Rd, Oak Ridge, TN 37831, U.S.A.

Amy Rose
Oak Ridge National Laboratory, One Bethel Valley Rd, Oak Ridge, TN 37831, U.S.A.

Robert Stewart
Oak Ridge National Laboratory, One Bethel Valley Rd, Oak Ridge, TN 37831, U.S.A.

Abstract

Geo-grid algorithms divide a large polygon area into several smaller polygons, which are important for studying or executing a set of operations on underlying topological features of a map. The current geo-grid algorithms divide a large polygon in to a set of smaller but equal size polygons only (e.g. is ArcMaps Fishnet). The time to create a geo-grid is typically proportional to number of smaller polygons created. This raises two problems - (i) They cannot skip unwanted areas (such as water bodies, given about 71% percent of the Earth's surface is water-covered); (ii) They are incognizant to any underlying feature set that requires more deliberation. In this work, we propose a novel dynamically spaced geo-grid segmentation algorithm that overcomes these challenges and provides a computationally optimal output for borderline cases of an uneven polygon. Our method uses an underlying topological feature of population distributions, from the LandScan Global 2016 dataset, for creating grids as a function of these weighted features. We benchmark our results against available algorithms and found our approach improves geo-grid creation. Later on, we demonstrate the proposed approach is more effective in harvesting Points of Interest data from a crowd-sourced platform.

2012 ACM Subject Classification Theory of computation → Divide and conquer

Keywords and phrases geofence, geo-grid, quadtree, points of interest (POI), volunteered geographic information (VGI)

Digital Object Identifier 10.4230/LIPIcs.GIScience.2018.58

Category Short Paper

1 Introduction

Obtaining land use data at a global scale can be arduous with respect to data availability, coverage, resolution, accuracy, computational power and storage. The emergence of Volunteered Geographic Information (VGI) exploited from open sourced platforms, however,

© Kelly Sims, Gautam Thakur, Kevin Sparks, Marie Urban, Amy Rose, and Robert Stewart;
licensed under Creative Commons License CC-BY

10th International Conference on Geographic Information Science (GIScience 2018).
Editors: Stephan Winter, Amy Griffin, and Monika Sester; Article No. 58; pp. 58:1–58:7
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

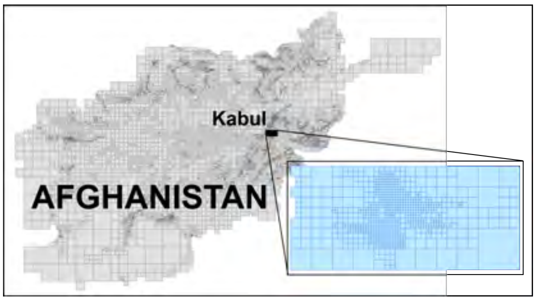


Figure 1 An example of dynamically-spaced geo-grid segmentation for weighted point sampling on a polygon map layer.

can provide rich attributes world-wide from Points Of Interest (POI) data. This data can facilitate many mapping and modeling techniques related to human dynamics, which is vital to emergency management and response, urban planning, and energy use. In order to optimize the collection of such available data, it is a recommended practice to segment a given geographical region into smaller grid cells for a more focused collection effort. ArcMap's fishnet[5], and other spatial indexing techniques that implement triangular mesh gridding are commonly used geoprocessing tools for this geo-grid segmentation. However, these tools are agnostic to the underlying topology and evenly segments areas into equally sized cells and/or triangular facets. Such approaches cannot skip unwanted regions (such as oceans) and their computational time is proportional to the size of individual cells, eventually, it consumes needless time and computational resources.

These challenges led us to develop a new approach for geo-grid segmentation that make use of underlying topology such as population distribution, building settlement extractions, etc. The proposed method creates dynamically spaced geo-grids as a function of underlying topological data. For example, as shown in Figure-1, using population distribution data to influence the location for request calls, the proposed method generates bigger cells for sparsely populated areas, or several smaller cells for densely populated regions.

2 Related Work

Multiple algorithms have been developed to generate a geocoded index to describe an exact or general location on Earth. In fact, many of these methods implement a hierarchical dissecting system that follows a tree structure concept to index places [7], or use a reverse geocoding practice of interpreting actual latitude/longitude coordinates to a single array [1], or develop a continuous map at a specific spatial unit labeled with new and random naming conventions [6] [8] [10]. Other approaches use a triangular mesh grid for analyzing geographic data within equally sized facets with minimal distortion [4] [3], or to identify coverage areas for database retrieval purposes [9] [11].

While the previously mentioned practices influenced this research, our approach is not to create a universal addressing system or a new projected referencing system. Instead, we propose a gridding system partitioned by a given topological requirement to help collect data from open source platform graph APIs that require a lat/long and search radius. Specifically, we produced a hierarchically gridded map of the world based on the underlying population within each grid to maximize our retrieval of Points of Interest (POI) data from VGI. By considering population distributions to subdivide a grid, we produce a map that can allocate additional computational resources where higher populations reside.

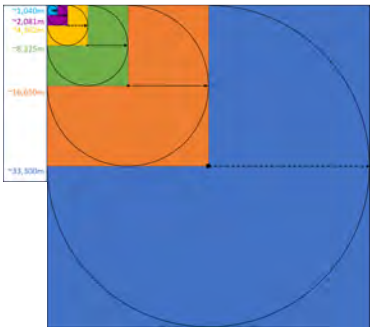


Figure 2 GeoHashed Grid Sizes with their respective search radii.

3 Methodology

In this section, we discuss the proposed method for creating dynamically-spaced geo-grid segmentation for weighted point sampling on a polygon map layer. And with our efforts focused on optimizing the number of POIs fundamental to human dynamics, we assume population is an indicator of places. We use the LandScan Global population dataset [2], developed at Oak Ridge National Laboratory (ORNL), for weighted point sampling purposes. This model depicts an "ambient" population distribution (average over 24 hours) at 30 arc-seconds resolution (roughly 1km at the equator).

3.1 Spatial Analysis

We began by gridding the world in ".6 x .6" degree increments for two reasons. The first reason was to begin with a small enough grid size so that the largest possible search radius (from the center of a grid to the nearest edge) would be less than 50,000 meters. With that said, a .6 degree grid at the equator has a radius of roughly 33,300 meters. The rationale for not using a search radius just under 50,000m (which would actually be a .9 degree grid at the equator), leads us to our second reasoning. Our geohashing technique splits each grid into 4 equal parts, until the last grid becomes smaller than a LandScan Global cell size, which is ~1km at the equator. When a geohashed grid is smaller than a LandScan Global cell, we can no longer sum the population within the single grid without missing at least one LandScan Global cell centroid. By using a .6 degree grid initially, we can geohash until the second to last geohashed grid is roughly 1% bigger than a LandScan Global population grid, allowing us to sum the population one last time. If the threshold is still not met, we can then geohash one final time to make the smallest request grid possible based on population (see Figure 2).

The population threshold for this study was 5,000. Therefore, when a grid's underlying population sum was above 5,000 people, subdividing took place into 4 new quadrants. From the top-left quadrant clockwise, we labeled the four quadrants A, B, C, and D, respectively. These quadrant labels were then appended to the original ID of the grid being GeoHashed. For example, Figure 3 represents the iterations of geohashing grids and the product of each new quadrant's ID. Grid 1 has a population over 5,000 and is replaced with four new grids labeled Grid 1a, Grid 1b, Grid 1c, and Grid 1d. When necessary, these divisions continue through each new quadrant(s) (Grid 1a, 1b, and 1c) constantly replacing the previous grid with new appropriately sized grids. If a grid's summed population is under 5,000, the iterations end and no additional divisions occur (Grid 1d). The rest of the section discusses the algorithms aspect of our method.

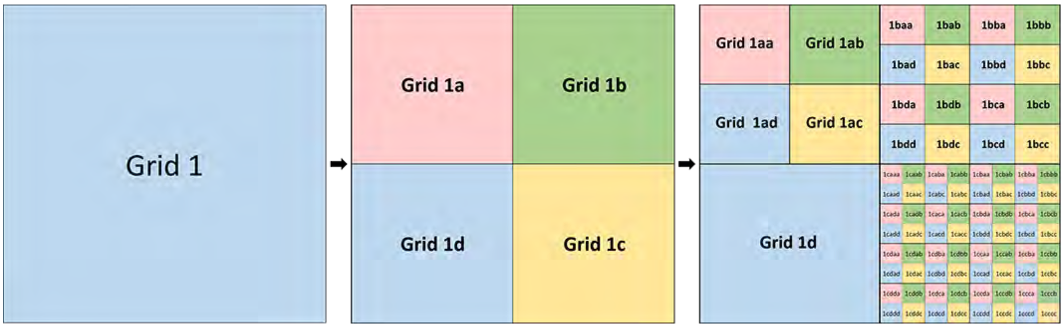


Figure 3 An example of how Grids are geohashed and labeled.

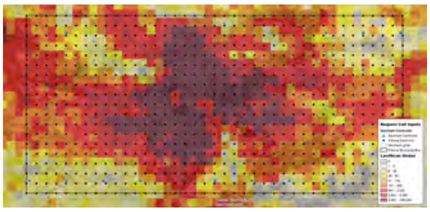


Figure 4 Request points calculated through proposed algorithms and a standard 1km fishnet grid over Kabul, Afghanistan. The background is LandScan Global's population distribution.

3.2 Algorithm

The spatial analysis algorithm performs two important tasks that results in deciding whether to split the current grid cell in equal size small sub-grid cells (or move to next grid cell. This decision is made based on the population count in the current grid cell. If the population count is higher than the threshold, the algorithm (Algorithm:1) segments the grid cell. In this algorithm, first we calculate the extent of the current grid cell. An extent defines the geographic boundaries that contains a population data frame. These boundaries contain top, bottom, left, and right coordinates, which are the edges of the map extent. For the purpose of this work, we rely on fixed extent calculation of the cell. Later, we calculate the total grid cell population count from the raster centroid of this grid cell (Algorithm: 2). The algorithm returns the value of population count that main algorithm use to decide whether to segment or maintain the current extent of the cell. In this algorithm, parameters are passed as reference to calculate the values of grid cell extent and cell population count.

4 Application and Summary of Results

To showcase the usability of our proposed method, we curated Points of Interest (POI) data over Kabul, Afghanistan. Influenced by population distributions from LandScan Global, we generated dynamically-spaced geo-grid cells for weighted point sampling on Kabul's polygon map layer. We then benchmarked this method against the traditional 1-km fishnet generated from ESRI ArcGIS and quantified the overall performance on three different measures. These measures included - (i) total number of POIs curated; (ii) total number of requests made to collect the POIs; and (iii) duration to collect.

4.1 Approach

A Point of Interest (POI) is a feature on a map that has a unique latitude and longitude coordinate. Some examples include - church, school, and hospital. Several mapping sites, such as OpenStreetMap, provide APIs to search POIs in a vicinity. These APIs input a

Algorithm 1: Algorithm to calculate grids using LandScan Global.

```
Function CalculateGeoHash(inputGrid, landScan_global, threshold)
    Data: LandScan Global raster population layer, grid, and threshold values
    Result: grid cells
    /* Split the original grid in 0.6° grid blocks.The radius from
       center is 33,000 mtrs at the equator.API threshold 50,000 mtrs
       */
    intpuGrid ⟵ calculate_fishnet(inputGrid, 0.6°) ;
    overlay_landscan_global();
    while true do
        /* Traverse each grid block in sequential fashion              */
        RunSpatialAnalysis(gridId, GeoPoint geoPoint[4], long populationCount) ;
        if populationCount > 5000 then
            Split the cell in to four equal parts;
            replace old cell with four new cells ;
            if current_grid == len(grid) then
                break();
            end
        end
        else
            continue; /* move to next grid cell                         */
        end
    end
end
```

geocoordinate with a radius, and returns POIs in that extent. Mapping websites impose a limit on the number of POIs returned, so its vital to keep a small radius. However, that increases the scanning time and number of API calls needed to maximize POIs collection. We assume population distribution dictates the distribution of POIs. A densely populated area may have more POIs than a sparsely populated area. In this experiment, we attempt to collect POIs for Kabul, Afghanistan. As shown in Figure-4, we have generated a set of 740 request points and distance between adjacent points using the proposed method and 790 request points at 1-km distance using the ArcMap fishnet algorithm. Next, we will make API calls using these two request points dataset and compare the results of total POIs collected.

4.2 Results Discussion

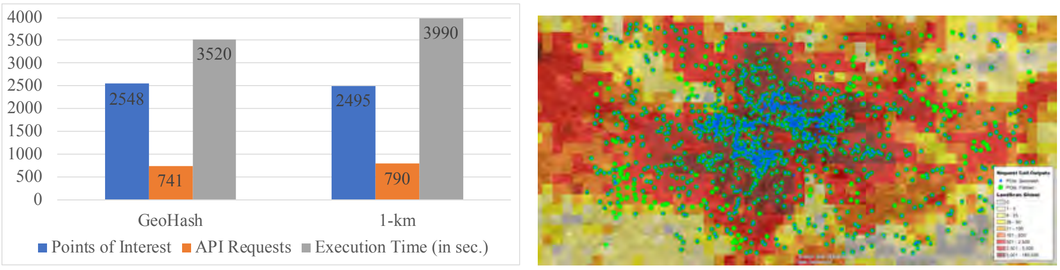
The proposed geohash approach outperforms the traditional 1-km fishnet approach on all three measures. As shown in Figure-5a, when the proposed approach reqeust points dataset was used, it took us 3520 seconds to collect 2548 POIs using 741 API call. When the 1-k dataset was used, it took us 3990 seconds to collect only 2495 POIs using 790 calls. These numbers become significant for large scanning area such as at country scale. In Figure-5b, we have shown the distribution of collected POIs from the two datasets.

4.3 Limitations and Future Work

One limitation to our proposed method is our use of LandScan Global, which restricts our segmentation abilities considering its resolution of 30 arc-seconds. In the future, we plan to implement LandScan HD, also developed at ORNL, because of its resolution at 3 arc-seconds

Algorithm 2: Algorithm to calculate grid cell extent and population.

```
Function RunSpatialAnalysis(gridId, GeoPoint geoPoint[4], long
  populationCount)
    Data: LandScan Global raster population layer, grid, and threshold values
    Result: grid cell extent, populationCount of the grid cell
    if cell !=null then
        /* Genrate coordinate values for top_left, top_right,
           bottom_left, bottom_right corners of the grid cell          */
        geoPoint ⟵ calculate_boundaries_spatial_extent();
        /* Calculate population count for this grid cell              */
        populationCount ⟵ extract_landscan_spatial_statistics(geoPoint) ;
    end
end
```



(a) The histogram shows the distribution of Points of Interest curated using the two methods, time, and the total number of request made to the server.

(b) Curated Points of Interest from two different methods, overlaying LandScan Global's population distribution.

Figure 5 Results of Points of Interest curation.

(~90m at the equator). With a more spatially refined population distribution, our model can be partitioned further with an ending grid size just shy of 90 square meters, instead of the current 600 meters. Furthermore, the processing time to develop our dynamically spaced geo-grids will also need to be improved. Countries with wide-spread population distributions, like India and Pakistan, produced hundreds of thousands of grids that were appended after each iteration, thus slowing down the overall processing time. Future work will explore how to improve this workflow as it will be necessary when we refine our input measurements to analyze population distributions at 3 arc-seconds.

5 Conclusion

In this paper, we proposed a method for gridding the world into varying geofenced grids based on a given measurement threshold to optimize search requests against social media APIs. While our research used ORNL's LandScan Global population dataset to designate the requirements, this algorithm can be augmented for other geospatial analysis and with other datasets. In fact, raster and point data are best suited with this method for generalizing or identifying areas of interest in vector data at multiple spatial representations. For example, crime data can be used to provide emergency personnel a map for allocating resource coverage. While, we recognize this method of geohashing is time consuming on the front end, we have observed an increase in the amount of data exploited, thus validating the necessity up front.

Copyright. This manuscript has been authored by employees of UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy. Accordingly, the United States Government retains, and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

References

1 Ziyad S. AL-Salloum. What Is Your Makaney Code?, 2011. URL: `http://www.makaney.net/`.

2 Budhendra Bhaduri, Edward Bright, Phillip Coleman, and Jerome Dobson. Landscan. *Geoinformatics*, 5(2):34–37, 2002.

3 Jun Chen, Xuesheng Zhao, and Zhilin Li. An algorithm for the generation of voronoi diagrams on the sphere based on qtm. *Photogrammetric Engineering & Remote Sensing*, 69(1):79–89, 2003.

4 Geoffrey Dutton. Encoding and handling geospatial data with hierarchical triangular meshes. In *Proceeding of 7th International symposium on spatial data handling*, volume 43. Citeseer, 1996.

5 ESRI. Create Fishnet—Data Management toolbox | ArcGIS Desktop. URL: `http://pro.arcgis.com/en/pro-app/tool-reference/data-management/create-fishnet.htm`.

6 Google. Google Plus Codes. URL: `https://plus.codes/`.

7 Gustavo Niemeyer. Geohash - Wikipedia, 2008. URL: `https://en.wikipedia.org/wiki/Geohash`.

8 Open Street Maps. DE:Browsing - OpenStreetMap Wiki. URL: `https://wiki.openstreetmap.org/wiki/QuadTileshttp://wiki.openstreetmap.org/wiki/DE:Browsing`.

9 Patrik Ottoson and Hans Hauska. Ellipsoidal quadtrees for indexing of global geographical data. *International Journal of Geographical Information Science*, 16(3):213–226, 2002.

10 Chris Sheldrick, Jack Waley-Cohen, Mohan Ganesalingam, and Michael Dent. what3words. URL: `http://what3words.com/https://map.what3words.com/palace.things.talking`.

11 Kentaro Toyama, Ron Logan, and Asta Roseway. Geographic location tags on digital images. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 156–166. ACM, 2003.