A Demonstration of Interactive Exploration of Big Geospatial Data on UCR-Star*

Saheli Ghosh, Akil Sevim, Ahmed Eldawy
sghos006,asevi006,eldawy@ucr.edu
University of California Riverside
Riverside, California

ABSTRACT

The ever rising volume of geospatial data is undeniable. So is the need to explore and analyze these datasets. However, these datasets vary widely in their size, coverage, and accuracy. Therefore, users need to assess these aspects of the data to choose the right dataset to use in their analysis. Unfortunately, all the publicly available repositories for geospatial datasets provide a list of datasets with some information about them with no way to explore the datasets beforehand. Through this demonstration, we propose the repository, UCR-Star, that is capable of hosting hundreds of thousands of geospatial datasets that a user can explore visually to judge their quality before even downloading them. This demo provides a deeper dive into the core engine behind UCR-Star. It provides a web interface geared towards database researchers to understand how the index internally works. It provides a comparison interface where the attendees can see side-by-side how two versions of the system work with the ability to customize each of them separately. Finally, the interface reports the response time of the indexes for a quantitative comparison.

CCS CONCEPTS

• **Information systems → Location based services**.

KEYWORDS

datasets, big data, geospatial data

ACM Reference Format:
Saheli Ghosh, Akil Sevim, Ahmed Eldawy. 2020. A Demonstration of Interactive Exploration of Big Geospatial Data on UCR-Star. In *28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '20), November 3–6, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397536.3422334

*This work is supported in part by the National Science Foundation (NSF) under grants IIS-1954644, IIS-1838222 and CNS-1924694 and by Agriculture and Food Research Initiative Competitive Grant no. 2019-67022-29696 from the USDA National Institute of Food and Agriculture

SIGSPATIAL '20, November 3–6, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-8019-5/20/11.
https://doi.org/10.1145/3397536.3422334

1 INTRODUCTION

In the past decade, there has been a significant increase in geospatial data. From satellite data to daily weather reports, from earthquake prone zones to social media, location-based data are needed and used by scientists from various spectres. Many of these data are open sourced and made publicly available by governments, non-profits, and other industries. For example, US Government [2], UK Government [1], World Bank [8], and United Nations [7]. With hundreds of thousands of geospatial datasets available, students and researchers struggle to select the datasets that fit the projects they work on. To choose an appropriate dataset, researchers have to assess the **coverage**, **quality** and **accuracy** of these datasets among other features. Unfortunately, all existing big data repositories provide no means of assessing these. Instead, they provide a boring list of dataset names and descriptions. This leads to, users downloading these datasets, analyzing them, only to find a few dataset that can be used, while discarding majority of them. This is a waste of precious time as many of these datasets are hundreds of gigabytes in size.

One easy way to determine the coverage, quality and accuracy of a geospatial dataset is to visualize it on an interactive map. In a few seconds, users can zoom out to see the coverage of the data or zoom in to measure the quality and accuracy. The main challenge in this approach is **to provide interactive visualization of hundreds of thousands of datasets in terabyte scale through a simple web interface without requiring users to download any data beforehand.**

At UCR, we have built the Spatio-Temporal Active Repository, **UCR-Star** [https://star.cs.ucr.edu] which assists users in the dataset selection problem. Figure 1 provides a high-level overview of UCR-Star. The data portrayed on the map is the bird's eye view of Chicago crime dataset [3]. UCR-Star currently hosts **152 datasets** with nearly a **terabyte** in size and it is capable of hosting hundreds of thousands of such datasets. This work demonstrates the **visualization engine behind UCR-Star**. Internally, UCR-Star indexed each dataset using the adaptive image-data index (AID) [4] which arranges pregenerated image and data tiles in a pyramid structure that enables the web server of UCR-Star to provide the desired real-time visualization.

The contribution of this demonstration can be summarized as follows:

1. It introduces the open sourced repository, UCR-Star, that allows users to visually explore various publicly available big geospatial datasets.
2. Demonstrates the internal design of the visualization engine behind UCR-Star which provides extremely interactive visualization of terabytes of data on a single machine.

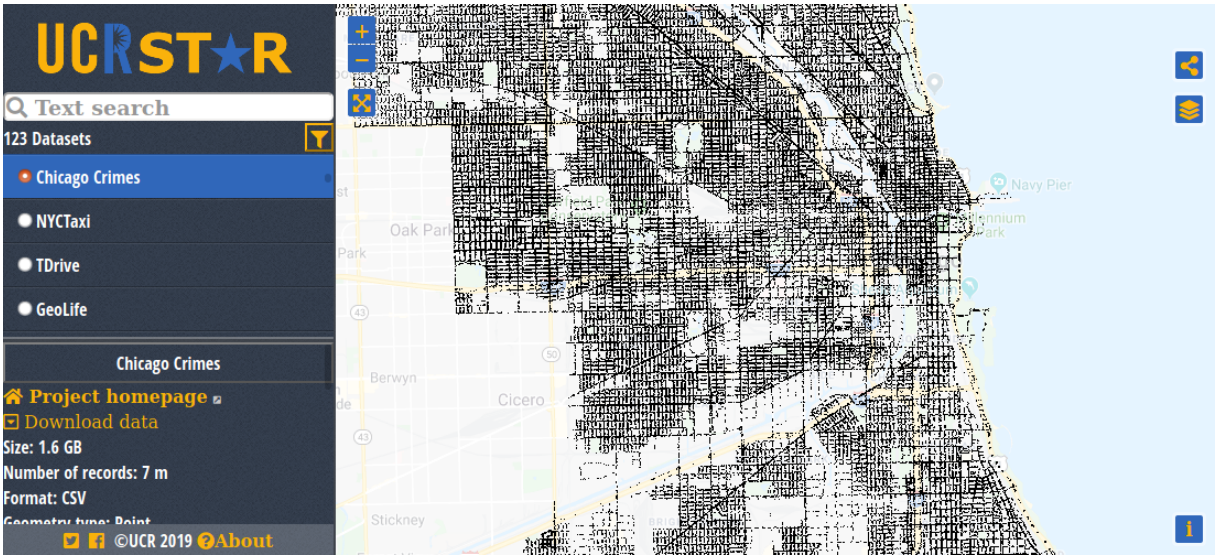

Figure 1: The main screen of UCR-Star [https://star.cs.ucr.edu/] [5] displaying the Chicago Crime dataset

3. It provides a comparison interface between AID and AID* indexes, which are the key components of this visualization engine. This interface allows the audience to change the system parameters and observe the performance.

It should be noted that the focus of this demonstration is *not* on the graphical interface of the repository. The novelty lies in building **one single open sourced system that can host hundreds and thousands of big geospatial data on a single machine for interactive exploration**. This serves the database, geospatial data, and data science communities, who are in a constant search for datasets, to process and analyze these spatial datasets without downloading them.

2 BACKGROUND AND METHODS

This section provides an overview of the core visualization engine and how it achieves a scalable and interactive performance. First, we describe the standard tile-based visualization used in web maps. Then, we describe the two phases of UCR-Star visualization, namely, data preprocessing and visualization query processing.

2.1 Web-map Visualization

The standard and most widely used method for map visualization is through the various JavaScript map libraries including Google Maps and Open Layers. This technique relies on shipping the entire data to the web browser which handles all visualization. Unfortunately, this technique is not suitable for visualizing big data due to the limited capabilities of most web browsers. The alternative method that we use in this demo is the tile-based visualization. This method uses fixed-size image tiles that are organized in a pyramid structure as shown in Figure 2. This pyramid structure enables the visualization of arbitrarily large datasets since the number of tiles that need to be placed on the screen is limited by the screen size not the data size. Unfortunately, this method is not widely used due

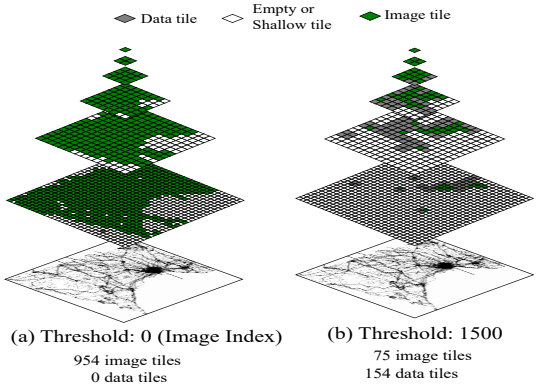

Figure 2: The multilevel AID index structure

to the huge overhead in building and maintaining the tiles which can grow to billions of tiles for 20 zoom levels [6]. Since UCR-Star aimed to host hundreds of thousands of datasets, we use a smart solution to store these pyramids as explained below.

2.2 Data Preprocessing using AID/AID*

In order to prepare the datasets for visualization, UCR-Star builds a partial pyramid structure, called **Adaptive Image-Data (AID)** index [4], that stores way fewer tiles as compared to regular web-map visualization. AID leverages the sparsity of geospatial data in order to pregenerate and materialize selected tiles beforehand. The remaining tiles are generated on-demand upon user request. This results in a very small-sized disk-resident index which makes it possible to host hundreds of thousands of such indexes in a single-machine system for the final visualization. Figure 2 gives an example that reduces the number of tiles in one index from 954 tiles down to only 75 tiles in the proposed index.

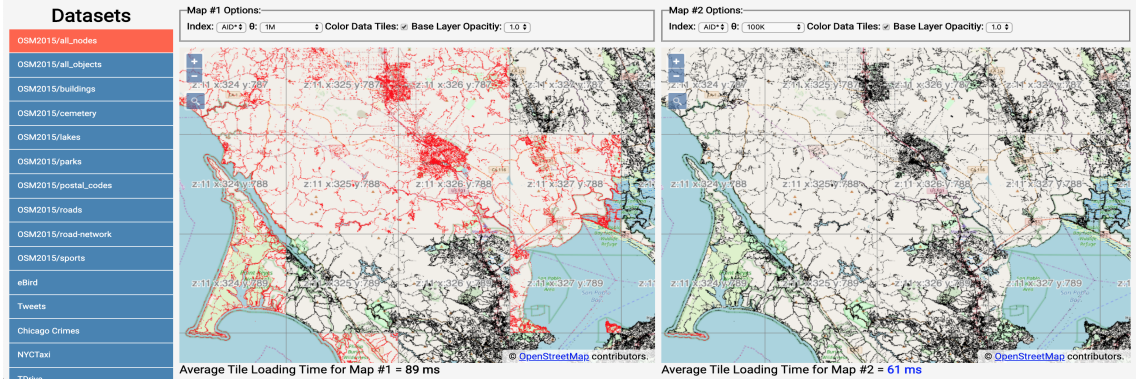

Figure 3: UCR-Star visualization engine showcasing the web interface and the effects of different threshold

AID controls the number of tiles to generate using a **threshold (θ)** which represents the largest size of data that can be generated on-demand without sacrificing the interactivity of the system. This threshold is used to classify tiles into four classes:

***Image*:** If a tile size is larger than θ, it is classified as an image tile.

***Data*:** If a tile size less than or equal to θ and has an image tile as a parent, then it is classified as a data tile.

***Shallow*:** If a tile size is less than θ, but the parent is a data tile, then it is classified as a shallow tile.

***Empty*:** When a tile size is zero, it is classified as an empty tile.

The details of how the tiles are classified and generated at scale can be found in [4].

AID: The AID index significantly reduces the index size by generating and storing only image and data tiles. Image tiles are stored as pregenerated image files. Data tiles are stored in a geospatial data format, e.g., CSV or Geojson. When a user requests a data or a shallow tile, it can be generated on-the-fly by reading and processing exactly one data tile which sets an upper limit on the time needed to generate any tile.

AID*: The AID* index further improves over the AID index by storing only image tiles. Instead of storing data tiles, the AID* index uses a separate data index, e.g., R-tree or R*-tree, that can be utilized with a range query to generate data and shallow tiles on the fly.

The example in Figure 2 illustrates the saving in the storage of the index which also reflects in the time needed to construct the index. For this example, if all tiles are stored, 954 tiles will need to be generated and stored. If an AID index is built, only 229 tiles will be stored. In AID*, only 75 image tiles will be generated and stored. This example is for only six levels. In the demo, we generate 20 levels and the reduction in index size and construction time for AID and AID* is several orders of magnitude [4].

2.3 Visual Exploration

Visual exploration consists of a single visualization query that retrieves an image for a single tile in the pyramid structure [4]. This visualization query is integrated into OpenLayers to provide the interactive visual interface for end users. This visualization query consists of a dataset ID and a tile ID (z, x, y), where z is the zoom level, and (x, y) is the position of the tile in the grid at level z. The result of the query is an image that represents the requested tile. If the requested tile is pregenerated (Image tile), then it can be fetched and returned immediately. Otherwise, it needs to be generated on-the-fly using the AID or AID* index. For AID, the tiles generated on-the-fly are either data tiles or shallow tiles. For the data tiles, the images are simply generated from the information within the data tiles. However for the shallow tiles, the parent of the tile needs to be traced up the pyramid to generate the image for the tile. In case of AID*, a simple range query on the pre-indexed input data is used to generate the tiles on-the-fly. It should be noted that the tiles generated on-the-fly should be small enough to not affect the interactivity of UCR-Star. The threshold (θ) is hence an important component in the design and making it too big or too small can affect the real-time interaction of UCR-Star.

3 DEMO SCENARIO

UCR-Star is already publicly available at https://star.cs.ucr.edu for everyone [5]. Besides the primary UCR-Star web interface, we will also provide an exclusive version of the visualization engine for Sigspatial attendees that allows database researchers to observe how the system internally works. A video of this demonstration is also available at https://www.youtube.com/watch?v=PJEMwrLipVk. In the following part, we first describe the proposed web interface for this demonstration. Then, we give three suggested scenarios for users to interact with this demo to see 1) the distribution of on-demand tiles, 2) the effect of the threshold (θ), and 3) the difference between AID and AID* indexes.

3.1 Web Interface

Figure 3 represents **the web interface of this demonstration.** On the left side of the page all the datasets are listed. Attendees can choose the dataset they want explore from the list. Attendees are able to zoom-in deeper into the dataset for detailed view or zoom-out for bird's eyes view, similar to the interaction of a standard webmap. UCR-Star provides 20 zoom levels for all generated visualization. It should also be noted that UCR-Star does not aggregate or sample the data which means that all individual records for any dataset are visible at deeper zoom levels. The figure also shows that the datasets are realized on top of OpenStreetMap (OSM), but there are options of switching the base layer from OSM to Google Satellite or Google Maps. The page is divided into two panels, so

that attendees can compare and contrast various features side-by-side. On each of the two sides, they have the option of choosing between the two indexing techniques (AID and AID*), different thresholds (θ), base layer opacity, and an option of distinguishing between pregenerated image tiles and tiles generated on-the-fly by color coding them. Both maps are geosynchronized which means that an interaction with one map will always reflect on the other so that they show the same region. Users can also search for a specific location on the map by entering any textual query, e.g., country or city name. At the bottom left corner of both maps, the average tile loading time is provided for the users to identify the efficiency of the features chosen by them. The average time is updated as users interact with the map. The back-end is hosted on two identical AWS servers, one for each map. This ensures a fair comparison between the two techniques since there will be no contention between them on the same set of resources.

3.2 Distribution of On-demand Tiles

The two proposed indexes, AID and AID*, pregenerate and materialize only a few image tiles while the remaining tiles are generated on-demand. The first demo scenario **helps attendees to understand the ratio of pregenerated image tiles and tiles on-the-fly in UCR-Star at different levels and how non-static tiles increase with increasing zoom levels.**

When the users select the 'Color Data Tiles' checkbox, tiles that are generated on the fly will be colored in red, as seen in Figure 4. As they zoom in deeper, the users will see a surge in the red tiles denoting an increase of tiles generated on-the-fly. This is because with deeper zoom levels data coverage by each tile decreases causing the size of the tiles to fall below the threshold (θ). From the figure itself we can see that the black tiles are more dense, containing more records which explains the reason for them being bigger in size. Whereas the red ones are comparatively sparse and contains lesser records. The tile classifications are controlled by threshold (θ), defined in the index construction phase and the details of this can be obtained from [4].

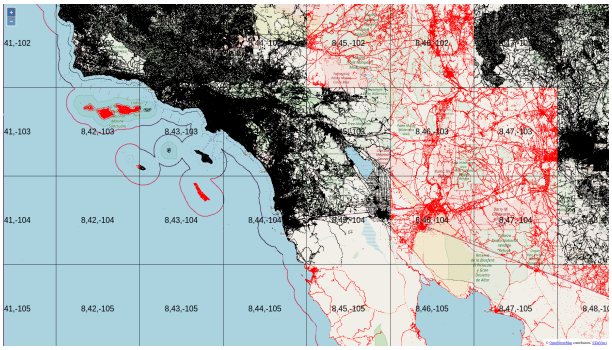

Figure 4: UCR-Star showcasing static image tiles (black) and tiles generated on-the fly (red)

3.3 Threshold (θ)

The key parameter for configuring both AID and AID* is the threshold θ. In this part of the demo, the users can see the effect of changing the threshold θ on indexes. The threshold θ controls the trade-off between the index size and the response time of the visualization query. As θ increases, fewer tiles will be generated by both AID and AID* while the number of on-demand tiles will increase and the server will take more time to generate them. The attendees **can see how increasing threshold θ hinders the interactivity of the system and takes more time in generating the tiles on-the-fly.** Similarly extremely small threshold results in the indexes having almost all static tiles, increasing the index overhead, which makes it impossible to host multiple datasets within UCR-Star.

In Figure 3, we see two different thresholds (θ=1 Mb and θ=100 KB) have been used for visualizing the same dataset. To represent how much tiles are generated on-the-fly, we have kept the color distinction on. As we can see higher threshold (left panel), results in more tiles to be generated on-the-fly, resulting in longer time for average tile loading.

3.4 AID and AID*

Through this demonstration, attendees **will be able to see a side by side comparison of AID and AID* in UCR-Star's visualisation engine.** For the same threshold both AID and AID* have the same number of static image tiles (tiles that are bigger than the threshold (θ) and are pregenerated as image tiles). However, AID* works on a previously indexed dataset when generating images on the fly, while AID generates it from data files (e.g., CSV). This affects the time taken to generate the image tiles on-the-fly. For a data tile, only one file needs to be read from disk and processed but the entire file needs to be read since it is not internally indexed. For example in AID, if a shallow tile is processed, only a small portion of the data tile is needed but since it is stored as a non-indexed file, it need to be processed in its entirety. On the other hand in AID*, the associated R*-tree index needs to be used which means that the index structure needs to be traversed before the data is located. However, since the R*-tree node is much smaller than a data file, the amount of data that needs to be processed can be further limited than AID.

This phenomenon can be experienced by the attendees, by selecting AID on one panel and AID* on the other panel of UCR-Star. As the attendees will zoom in or out or pan across their desired dataset they can see the average tile loading time for both AID and AID* to verify the efficiency of both methods.

REFERENCES

[1] UK Government Data Project, 2010. https://www.data.gov.uk/.
[2] Open Data Repository Maintained by US General Service Administration, 2009. https://www.data.gov/.
[3] C. P. Department. Reported incidents in the city of chicago, 2019. Retrieved from UCR-STAR https://star.cs.ucr.edu/?ChicagoCrimes&d.
[4] S. Ghosh, A. Eldawy, and S. Jais. AID: An Adaptive Image Data Index for Interactive Multilevel Visualization (Poster). In *ICDE*, 2019.
[5] S. Ghosh, T. Vu, M. A. Eskandari, and A. Eldawy. UCR-STAR: The UCR Spatio-Temporal Active Repository. *SIGSPATIAL Special*, 11(2):34–40, Dec. 2019.
[6] OpenStreetMap disk usage, 2019. https://wiki.openstreetmap.org/wiki/Tile_disk_usage.
[7] United nations open data, 2019. http://data.un.org/.
[8] World bank open data, 2019. https://data.worldbank.org/.