OntarioTech
Engineering
& Applied Science

AID: An Adaptive Image Data Index for Interactive Multilevel Visualization

Presented By:

- Dipkumar Patel
- Noopa Jagadeesh
- Prasanth Varma

Agenda

- Introduction
- Related work
- Proposed Index: AID Index
- Visualization Query
- Results
- Conclusion
- Future work

Introduction

In recent years, there has been an explosion in the amounts of spatial data

*"Over **2.5 quintillion bytes** of data are created every single day, and it's only going to grow from there. By 2020, it's estimated that 1.7MB of data will be created every second for every person on earth."* - sixth edition of DOMO's report

An applications capable enough to visualize big spatial data

- Scatter plot of billions of tweets worldwide
- A frequency heat map for Twitter data showing the hot spots of generated tweets
- A road network for the whole world

Map Index

- The number of tiles increases exponentially with each zoom level, an efficient indexing technique is required to store these tiles
- Most web maps provide 18 zoom levels with up-to 90 billion tiles
- Users should be able to zoom in and out in the generated image to get different resolutions of the whole data set

Figure1 : Map zoom levels

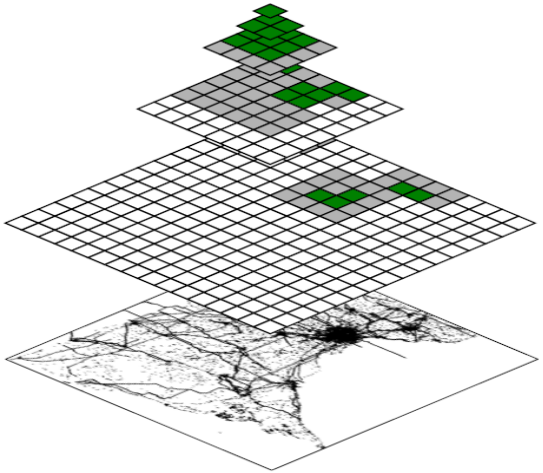


Figure 2: Map indexes tiles

One dimensional indexes

- One dimensional ordering of key values, such as B-trees and ISAM indexes do not work because the search space is multi-dimensional
- Structures based on exact matchmg of values, such as hash tables, are not useful because a range search is required

- **Visualization index**
  - Image index
  - Data index

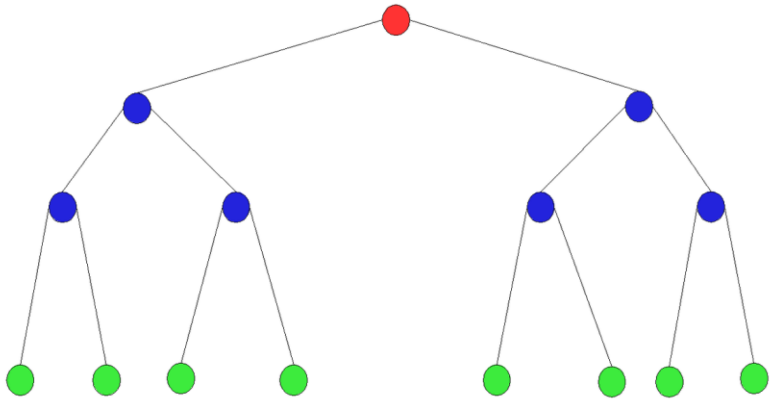


Figure 3: Binary Tree

Image Indexes

- All non empty tiles are **pre-generated** in a **preprocessing phase** and are stored in a simple hash index by their **tile ID**
- The visualization process becomes almost constant time as it just fetches tiles from the image index
- This technique is helpful for highly **reusable** visualizations which are visualized by millions of users
- It comes at a very **expensive** preprocessing phase to build the index

Data indexes

- A traditional spatial index (R-tree) is constructed and used to retrieve the desired data and visualize it upon user request
- These indexes are designed mainly to answer range queries
- They are only useful when the query result is small enough to be visualized on the fly

R-Tree

- It groups nearby objects and represent them with their minimum bounding rectangle in the next higher level of the tree
- The "R" in R-tree is for rectangle
- Root node: Holds pointer reference to largest region
- Internal nodes: Store pointers to childs
- Leaf nodes: It stores actual data
- For CRUD operations, Learn More: Reference

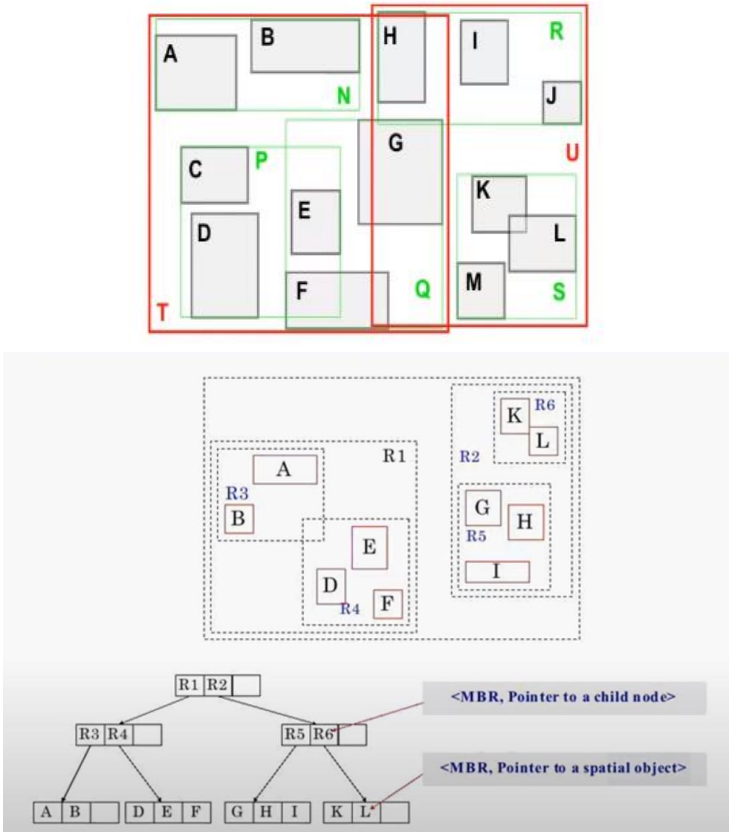


Figure 4: R-Tree

Data index vs Image index

- Image indexes provide an interactive visualization but require a long indexing time
- data indexes are fast to build but are not interactive for big data.

AID: An adaptive image data index

- **Key idea:** idea is to identify the regions that are costly to visualize

The Proposed Index: AID Index

•The proposed index uses a mix of image and data tiles.

•The image tiles are only pregenerated for the regions that are dense while data tiles cover all the remaining regions that can be visualized on demand.

•The proposed index follows a multilevel pyramid layout.

AID Index(contd…)

•In a traditional image index, a **quad-tree** structure is implemented to generate image tiles at different levels.

•For example, level 0 which is the topmost level has one single image of a particular resolution .

•Level 1, will have four image tiles generated from the parent tile at level 0, representing the same image.

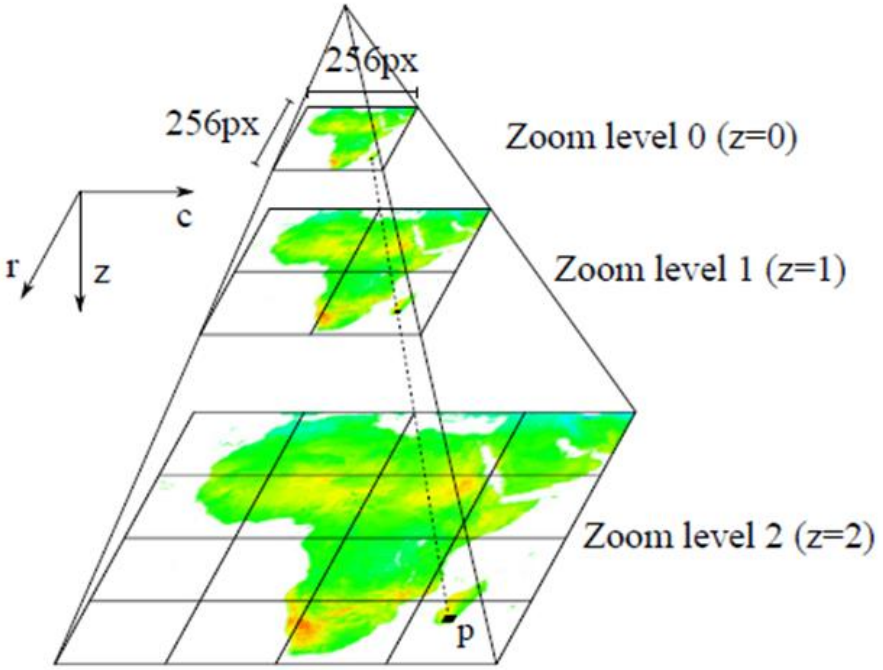


Figure 5 : Multilevel Image Visualization for a traditional image index[2]

AID Index(contd…)

- The proposed AID index focuses on reducing the number of tiles.

- A size **threshold θ_N** is defined where tiles with a data size larger than θ_N are considered expensive and tiles with a data size that is less than or equal to θ_N are considered cheap.

- These expensive tiles are pregenerated and materialized as an image.

- The inexpensive tiles cover a small amount of data and can be generated in real-time.

Visualization Query

•For executing visualization query we need visualization server.

•We define a single query, **GetTile,** that the visualization server processes to fetch an image tile or generate an image from a data tile to support different kinds of interactivity.

•The input is a tile ID and the output is an image that represents this tile.

Visualization Query(contd...)

•When a user requests a tile, the server looks into the index for an image tile by searching for an image that corresponds to the requested tile.

• If found, it returns that pre-generated image.

•If an image tile is not found for the corresponding requested image, the server then checks for a data tile with the same ID.

•If found, it reads the data tile and generates the image on the fly before returning it to the server.

Results

We measure the results based on-

1.Index Construction Time

2.Index Size

3.Query Processing Time

Results (cont)

1. The proposed AID index is compared to two baselines, an image index built using HadoopViz and an R-tree data index using SpatialHadoop .

1. The datasets are extracted US Census Bureau TIGER files and from Open- StreetMap.

1. DataSets Used:-

Dataset	Size	Records
Linear Water	6 GB	5.3 M
Roads	7.7 GB	20 M
All Nodes	96 GB	2.7 B

1. Index Construction Time

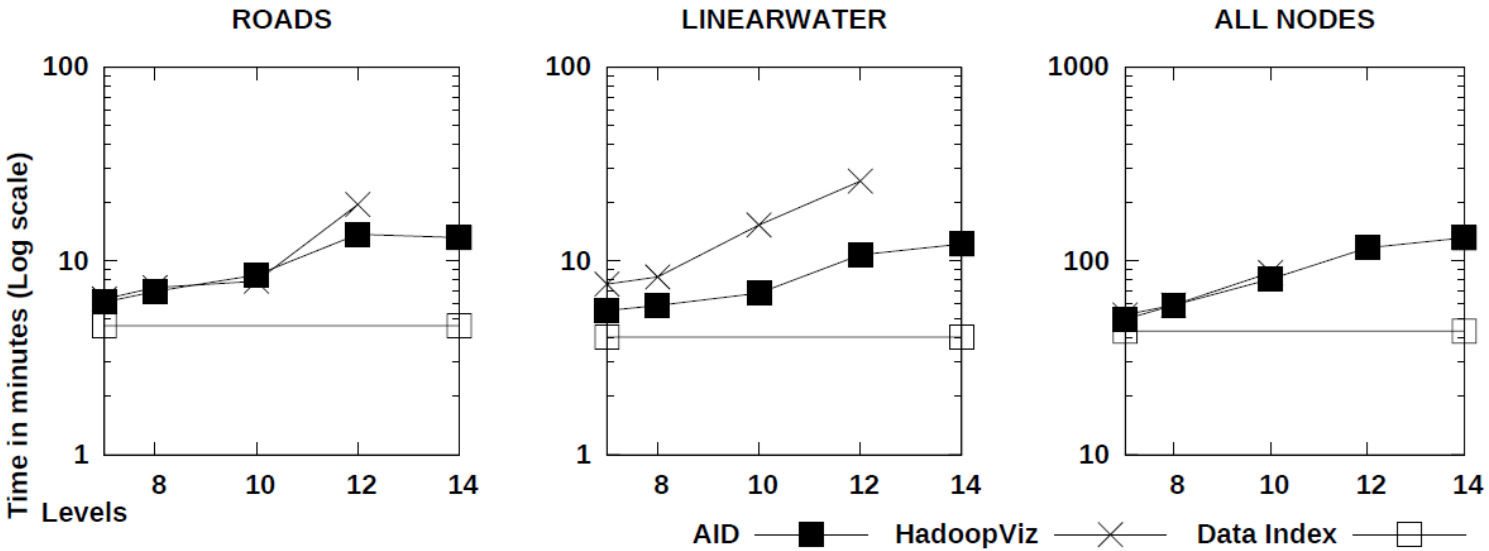


Fig. 2. Index construction time

2. Index Size

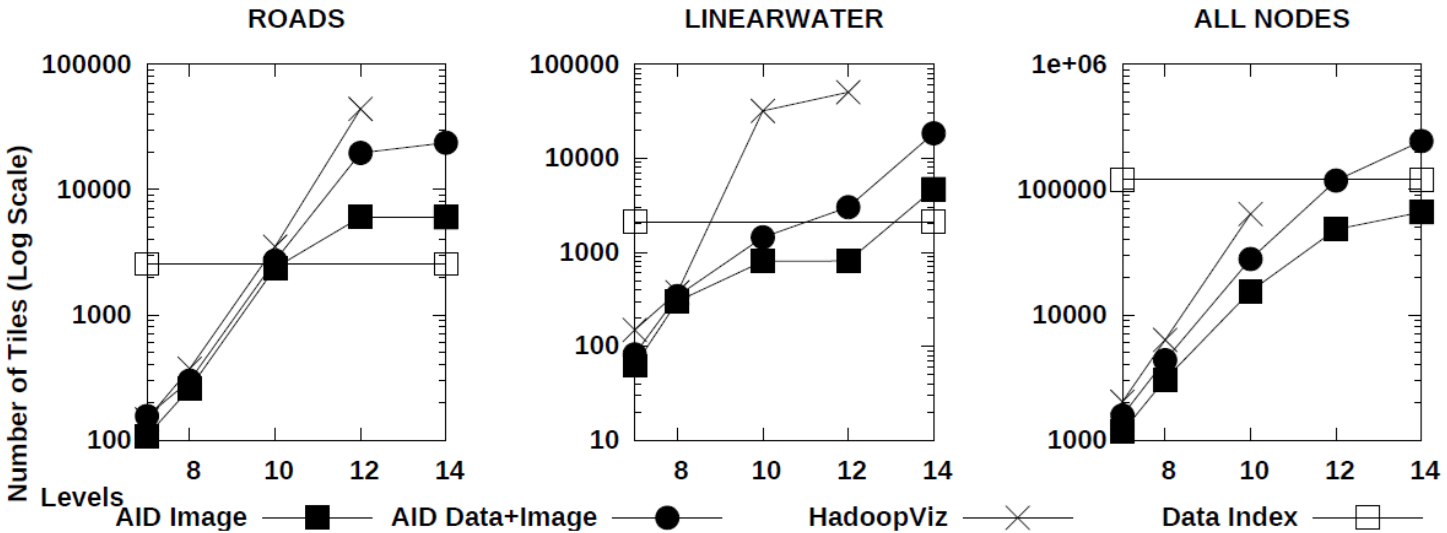


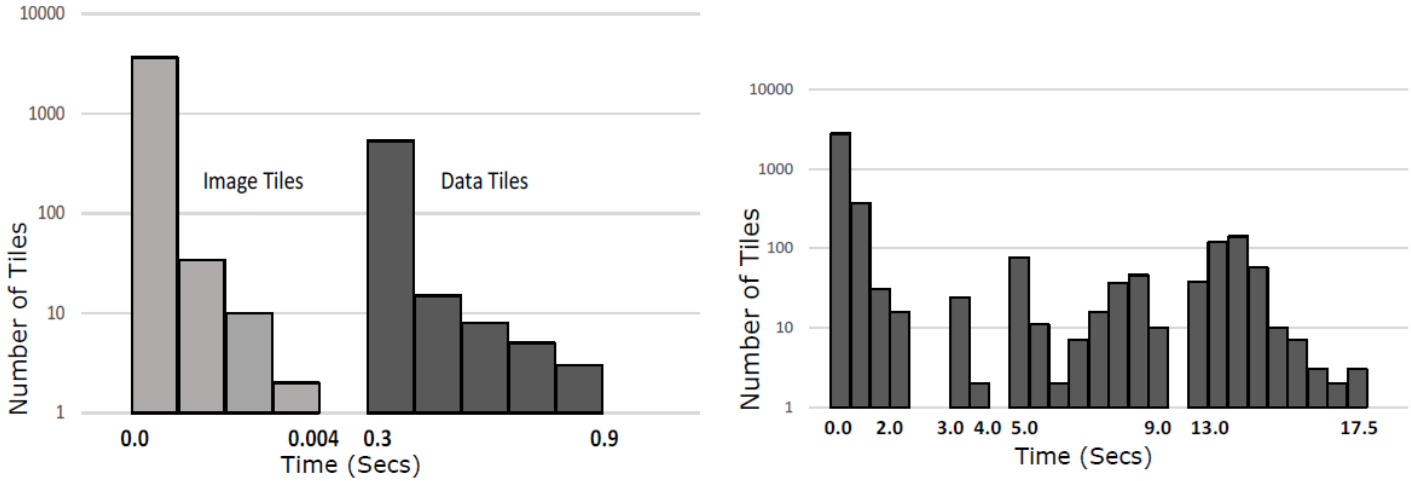
Fig. 3. Index size in terms of number of tiles

3. Query Processing Time

- We generate a benchmark that comprises a set of random tile positions. We choose 1000 tiles at random at level 9 and add all of them to the benchmark.

- Additionally, we add all the ancestors of the generated tiles, up-to the root tile, to the benchmark.

- This benchmark simulates the real workload of users zooming in from the root tile to a chosen location of the image.



(a) AID index on SPORTS

(b) Data index on SPORTS

Fig. 4. A histogram of the running time of the visualization query

Conclusion

- The proposed index requires much less time in index creation.
- The index size is smaller than the existing image index such as in HadoopViz
- The key finding in this experiment is that AID serves the majority of the queries in less than 500 milliseconds and the entire set of requested tiles in less than a second which makes it very interactive to end users.

Future work

The key idea is to identify the regions that are expensive to visualize and store them as pregenerated images while storing the remaining regions as raw data and produce the visualizations on the fly.

1) The current index uses only the data size to classify tiles. We can extend this by employing other parameters such as a desired index size, indexing time, or query time.

2) A more sophisticated index layout that can combine many small files into a few big files so as to further minimize the indexing time and index size.

Reference

[1] S. Ghosh et al., A. Eldawy et al., S. Jais., “AID: An Adaptive Image Data Index for Interactive Multilevel Visualization” , in IEEE, 2019.

[2] A. Eldawy et al., “SpatialHadoop: A MapReduce framework for spatial data,” in *ICDE*, 2015.

[3] G. Planthaber et al., “Earthdb: scalable analysis of MODIS data using scidb,” in *BigSpatial*, 2012.

Thank You!

Any Questions?