computers

MDPI

Article

ApproxGeoMap: An Efficient System for Generating Approximate Geo-Maps from Big Geospatial Data with Quality of Service Guarantees

Reem Abdelaziz Alshamsi [1], Isam Mashhour Al Jawarneh [1,*], Luca Foschini [2] and Antonio Corradi [2]

[1] Department of Computer Science, University of Sharjah, Sharjah P.O. Box 27272, United Arab Emirates; u23102393@sharjah.ac.ae
[2] Dipartimento di Informatica—Scienza e Ingegneria, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy; luca.foschini@unibo.it (L.F.); antonio.corradi@unibo.it (A.C.)
* Correspondence: ijawarneh@sharjah.ac.ae

Abstract: Timely, region-based geo-maps like choropleths are essential for smart city applications like traffic monitoring and urban planning because they can reveal statistical patterns in geotagged data. However, because data overloading is brought on by the quick inflow of massive geospatial data, creating these visualizations in real time presents serious difficulties. This paper introduces ApproxGeoMap, a novel system designed to efficiently generate approximate geo-maps from fast-arriving georeferenced data streams. ApproxGeoMap employs a stratified spatial sampling method, leveraging geohash tessellation and Earth Mover's Distance (EMD) to maintain both accuracy and processing speed. We developed a prototype system and tested it on real-world smart city datasets, demonstrating that ApproxGeoMap meets time-based and accuracy-based quality of service (QoS) constraints. Results indicate that ApproxGeoMap significantly enhances efficiency in both running time and map accuracy, offering a reliable solution for high-speed data environments where traditional methods fall short.

Keywords: heatmaps and choropleth; earth mover's distance; geospatial visualization; spatial data sampling; geohash; approximate query processing

check for updates

Academic Editor: Wenbing Zhao

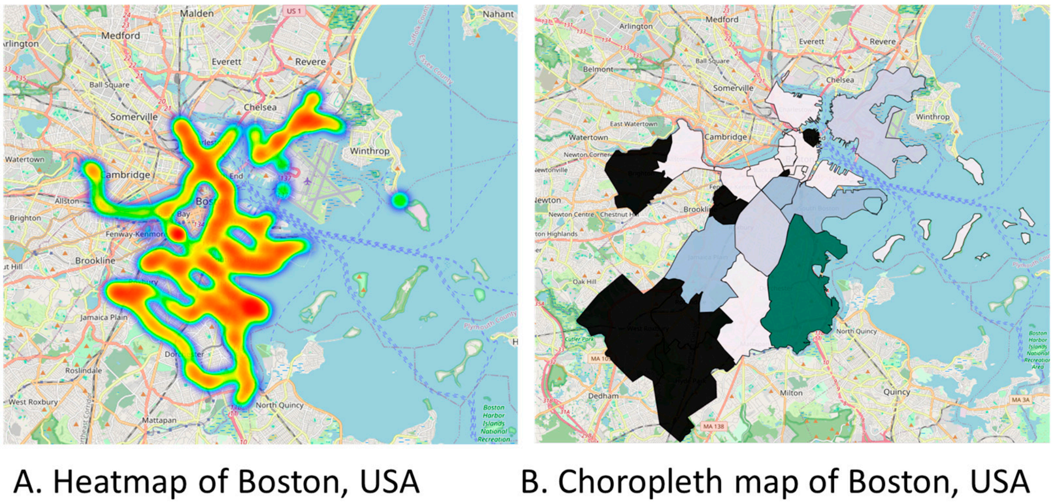
Received: 31 October 2024
Revised: 12 December 2024
Accepted: 9 January 2025
Published: 23 January 2025

Citation: Alshamsi, R.A.; Al Jawarneh, I.M.; Foschini, L.; Corradi, A. ApproxGeoMap: An Efficient System for Generating Approximate Geo-Maps from Big Geospatial Data with Quality of Service Guarantees. *Computers* **2025**, *14*, 35. https://doi.org/10.3390/computers14020035

1. Introduction

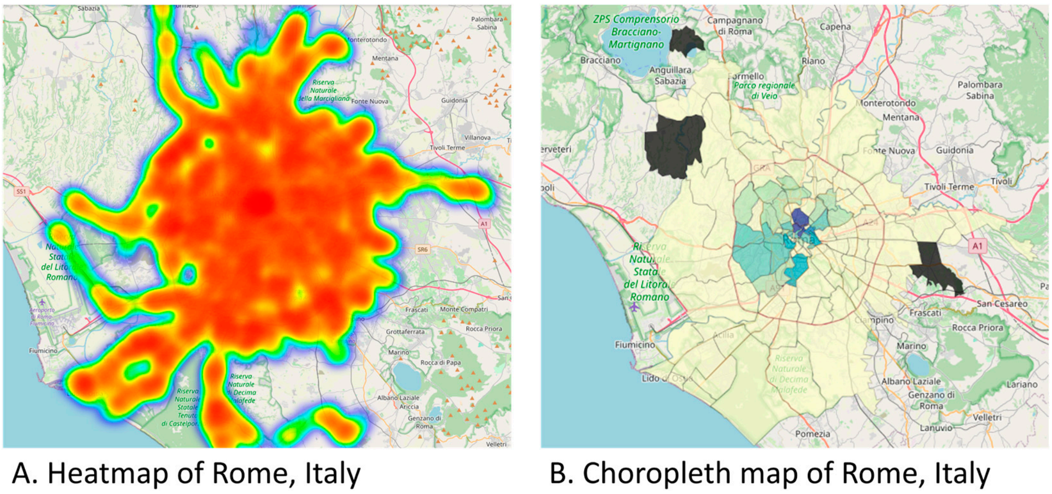
Every hour, immense amounts of georeferenced data streams are collected from smart cities around the world in various forms: pollution data, mobility data, and other geotagged data [1,2]. These streams serve as essential inputs for various spatiotemporal data science applications, such as Exploratory Spatial Data Analytics (ESDA). A key component of ESDA is the regular creation of geo-maps (e.g., region-based maps such as choropleths), which play a crucial role in urban planning for smart cities [3,4]. Figure 1A,B illustrate this process, with Figure 1A presenting a heatmap of air quality data density over 3 months in Boston, USA, and Figure 1B showing a choropleth map of the same data. Similarly, Figure 2A,B depict heatmaps and choropleths of taxi pickup patterns in Rome, Italy. These examples highlight how geospatial visualizations reveal critical patterns for improved decision making.

The visualization of large-scale geospatial data serves as a critical tool for an exploratory analysis, revealing patterns that can drive improved decision making in various areas of life. Nevertheless, the immense volume of georeferenced data introduces additional challenges, primarily due to the high cost of communication requirements and system scalability limitations. In real-time applications, the influx of fast-arriving geospatial data can overwhelm processing systems, creating significant bottlenecks [5]. For instance, spatiotemporal visualization enables us to recognize specific spatial areas, temporal segments, or a blend of both—often referred to as the spatiotemporal scope of interest. Pinpointing those scopes helps practitioners to refine and target their modeling efforts with greater precision [3].


A. Heatmap of Boston, USA

B. Choropleth map of Boston, USA

Figure 1. Region-based geo-maps: Boston, USA. (**A**) Heatmap of air quality data in Boston, USA. The intensity of the colors (from yellow to red) represents the density of records collected, with red areas indicating the highest density. (**B**) Choropleth map of air quality data in Boston, USA. Neighborhoods or polygons shaded in darker green represent areas with the highest density of records collected, while lighter colors indicate lower densities.


A. Heatmap of Rome, Italy

B. Choropleth map of Rome, Italy

Figure 2. Region-based geo-maps: Rome, Italy. (**A**) Heatmap depicting taxi pickup patterns over one day in Rome, Italy. The color intensity (from yellow to red) indicates the density of taxi pickups, with red areas representing the highest concentration. (**B**) Choropleth map illustrating the same taxi pickup data in Rome, Italy. The polygons are shaded to reflect the density of taxi pickups, with darker colors representing higher densities and lighter colors indicating lower densities.

It is becoming increasingly difficult and expensive to create region-based maps on a regular basis (e.g., every few seconds) due to fast-arriving, fluctuating-in-nature big geo-referenced data streams. In fact, during severe spikes in the arrival rate, the system may become overloaded and easily come to a halt. This not only affects system stability but also disrupts smooth user interactions, particularly in environments where decision making depends on timely insights [1,6]. Spatial approximate query processing (SAQP) is becoming crucial to preventing such situations that result in the system being out of service [4,6]. In this work, we present the design and implementation of a novel system for producing high-quality approximate region-based geo-maps. We focus on ensuring smooth user interactions and minimizing computational costs, even under data overload conditions. Our primary emphasis is on choropleth maps, which are frequently used in urban planning and smart city scenarios.

The creation of region-based geo-maps involves two primary tasks. The first task involves preparing geospatial data, which involves retrieving the data and running spatial queries (such as group-by-neighborhood aggregation queries) to produce the geographical tuples that will be geo-visualized. The second task, geo-map visualization, applies a geo-map visualization effect, such as choropleth color encoding, to the geospatial tuples that were produced in the initial step [7,8]. However, as the data volume grows exponentially, these tasks become computationally expensive, leading to latency and degraded performance. Addressing these challenges requires optimization techniques for geo-map generation [9]. In order to create the related image, the geo-visualization procedure is usually divided further into two steps: first, converting geographical coordinates into screen pixels, and second, rendering the values of features associated with those pixels [10,11]. The simple vector data size and the pixel data rendering cost are well correlated, with larger data sizes implying greater rendering costs, as the literature has well corroborated. During strong spikes in multidimensional data arrival rates, the two primary stages may become too expensive to operate in the event of information overload. An example is tweets with geotagging during the US presidential election. Our work contributes to this domain by proposing ApproxGeoMap (AGM), which incorporates a stratified sampling mechanism to ensure efficiency and scalability.

Furthermore, there are two types of methods for creating region-based geo-maps: those that use all of the data that arrive to create precise maps, and those that rely on approximation, sampling, sketching, or any other legitimate method of data size reduction or compression. Although more accurate, the former is more computationally costly and might be unaffordable or unfeasible in environments with limited IT infrastructure or during high-data-inflow situations where the rate of data arrival surpasses the processing and display capabilities of the underlying geospatial processing system. Conversely, the latter aims to obtain timely updates of data patterns and achieve considerable improvement in running times at the expense of a slight loss in accuracy.

The design and implementation of our new system, which we call ApproxGeoMap (for approximation geo-map mapper) or AGM, are presented in this work. AGM efficiently minimizes the volume of geospatial data to be visualized by employing a stratified spatial sampling technique based on geohash tessellation. This preprocessing step reduces data size before sending it to the geo-visualizer, which produces high-quality region-based geo-maps. A controller in our system detects arrival rates and adjusts the sampling fraction dynamically based on time-based and accuracy-based quality of service (QoS) criteria, ensuring smooth system operation. Earth Mover's Distance (EMD) serves as the foundation for this controller [12]. As will be covered in the next section, our geospatial sampling method is specifically based on tessellation, which involves dividing the geographic area (for which the region-based map is to be generated) into equal-sized rectangles using a dimensionality reduction approach based on z-order curves. To achieve this, we specifically use geohash encoding. This approach not only ensures spatial locality but also enables scalable and efficient processing, even under constrained computational environments [13].

This paper's remaining sections are structured as follows. In Section 2, we explore the basic theoretical background necessary to understand the methodologies and approaches used in this study. Section 3 provides a comprehensive literature review, focusing on advancements in geospatial data processing, approximate query techniques, and innovations in map rendering and interaction. Section 4 introduces the concept and prototype of our proposed system, ApproxGeoMap, designed for the visualization of approximate region-based geo-maps. Section 5 discusses the experimental evaluation and presents the findings on the system's performance across various datasets and configurations. Finally, Section 6 concludes the paper with key insights, practical implications, and recommendations for future research directions.

2. Preliminaries and Theoretical Background

This section offers a brief overview of the preliminary concepts and theoretical background necessary to grasp the design and implementation details of our novel system, which will be addressed later in the paper.

2.1. Geo-Visualization

Geo-visualization can be broadly defined as the process of creating geo-maps from georeferenced data, which involves two main stages. The first stage is geospatial data processing, where queries are applied to incoming georeferenced tuples based on the geo-visualization query. For instance, when generating a choropleth map, data must be aggregated into clusters (either ad hoc or pre-selected), requiring the use of stateful geospatial aggregation queries, such as grouping and counting tuples or determining the 'average' scalar value, such as the 'average' taxi speed in mobility data. The output is a geospatial vector dataset, which is then converted to a raster format for the next stage. The second stage is geospatial data visualization, where the rasterized vector data are rendered into geo-maps for user display. Rasterization involves determining the correct pixel location within the map's grid, corresponding to the geographical location of the real-world tuple [8,14].

Approaches for visualizing georeferenced data are typically divided into three main categories: point-based, line-based, and region-based techniques [15,16]. Point-based methods plot individual points on maps, such as Point-of-Interest (POI) maps, allowing users to observe spatial objects or events directly [17–19]. When multiple points overlap, data aggregation methods, such as KDE-based heatmaps, can resolve the overlaps and reveal meaningful patterns [20–22]. Line-based methods, on the other hand, are used for visualizing time-series trajectory data, illustrating the movement of objects over time using lines and curves [23–27].

Region-based approaches, which are the most resource-intensive due to their reliance on dividing geographic areas into grid cells, aggregate data into predefined spatial regions [28]. These techniques often require extensive geospatial data processing, including grouping by regions and applying aggregation operations like counting or averaging scalar values (e.g., calculating the average speed of vehicles). This step is crucial for generating visual outputs like choropleth maps, commonly used for urban planning and spatial analyses [18,19,29]. In this study, point-based techniques are primarily leveraged to visualize alternatives in spatial contexts. One popular and typical example of visualizing georeferenced data using region-based methodologies is the generation of choropleth maps. The process entails creating a map using the already-established tessellation of a specific geographic area, after which each region is given a color density according to color coding and the density in each tessellation tile based on geo-statistics or geospatial aggregations calculated during the geospatial data processing stage. It should be noted that regional divisions of a study area (also known as administrative regions) are the level at which geospatial aggregations for the purpose of creating choropleth maps are carried out. An example is some areas or communities inside a large city. For example, Figure 1B shows a choropleth map of air quality data in the city of Boston, USA. Neighborhoods or polygons that have darker green colors have the highest density of records collected. Another typical example of a region-based geo-map that is seen in smart cities is a heatmap. All types of region-based geo-maps, despite some differences, need stateful data aggregation, which is known to be computationally costly in real data stream settings and has the potential to quickly bring down systems in the event of severe spikes in data arrival rates. This challenge is particularly critical in real-world urban environments with limited IT infrastructure, where resource constraints exacerbate the difficulties of handling fluctuating data volumes. Our system, ApproxGeoMap, addresses this issue through stratified sampling and load-shedding techniques that balance efficiency and accuracy while minimizing computational resource usage. Georeferenced data are typically encoded as coordinate pairs (longitude and latitude) to minimize network congestion during transmission. However, this parameterization strips the data of its true geometrical form.

Bringing those parametrized points back to their original forms is necessary to perform geospatial aggregation and produce region-based maps. This is a computationally expensive type of a geospatial join in data stream settings since it specifies which areas in real geometries the points belong to [20]. This computational overhead is exacerbated during spikes in data arrival rates, potentially affecting system responsiveness and user interactions. Having stated that, it is evident that geospatial data preprocessing plays a major role in the generation of geospatial region-based maps. In this regard, using geographic SAQP techniques like load shedding and geospatial sampling is a last resort if preprocessing is taking longer than it should.

However, it is important to note that while stratified sampling can improve accuracy by reducing within-stratum variance, its efficiency depends on the spatial correlation of the data. When spatial correlation is weak, stratified sampling may not outperform simple random sampling, and in some cases, it may result in a “stratification trap” if the designed strata differ significantly from the true strata of geospatial objects [30]. Addressing this challenge requires the careful consideration of spatial correlation characteristics when designing sampling methods.

A lot of research has been conducted regarding geo-visualization and its uses in multiple fields all over the world. For example, in the domain of social media, ref. [31] introduces a visual analytics approach to analyze tweet topic popularity across locations and time, offering insights into public interests and social trends. It focuses on a spatiotemporal analysis to track how topics gain traction in various cities and detect significant events like political movements through burst detection. Interactive tools allow users to explore topic evolution and regional differences in sentiment. The approach aggregates tweets by hashtags and time intervals for better topic modeling, using techniques like Latent Dirichlet Allocation (LDA). Visualization tools help analyze trends, bursts, and spatial distributions. Future research aims at real-time monitoring, dynamic time windows, and scalability to handle larger datasets, enhancing the understanding of social media’s spatial and temporal patterns.

Contrastingly, refs. [32–36] collectively highlight geo-visualization’s role in optimizing transportation systems, urban mobility, and sustainability by providing insights into traffic patterns and mobility behaviors. On one hand, refs. [32,33] discuss techniques like a trajectory analysis, origin–destination mapping, a semantic zoom, and 3D visualizations for analyzing mobility patterns, congestion, and travel efficiency, with future research focusing on real-time analytics, multimodal integration, and predictive modeling. Then again, refs. [34,35] address public transportation and smart city optimization using heatmaps, space–time cubes, and multi-objective optimization for traffic management, planning, and incident response. Future work emphasizes real-time data integration, machine learning, and user-friendly design. Additionally, ref. [36] extends to urban mobility and environmental monitoring, stressing the importance of real-time data, multi-source fusion, and predictive analytics for effective urban management.

On the other hand, the authors of [26,37,38] emphasize the role of geo-visualization in a trajectory analysis for urban mobility, traffic management, and real-time monitoring. Techniques such as graph-based modeling (e.g., TrajGraph), hot route discovery, and GPU-based visualizations help identify congestion, optimize infrastructure, and respond quickly to traffic changes. Ref. [38] further covers applications like emergency response and environmental monitoring, using methods such as space–time cubes and heatmaps to reveal movement patterns. Key research focuses on adding contextual data, enhancing real-time interaction, and exploring multiscale analyses. Together, [26,37,38] show how trajectory-based geo-visualization improves traffic flow, supports emergency management, and provides real-time insights, with future work aimed at predictive modeling and scalable solutions.

Geo-visualization is essential in urban planning, offering insights into mobility patterns and spatial interactions to guide decision making. The authors of [38] highlight tools like Location2vec for a dynamic analysis of human activity, aiding infrastructure development. On the other hand, ref. [39] discusses identifying functional areas using movement data, such as taxi trips, to optimize traffic and accessibility. Both papers emphasize real-time monitoring of urban changes, using techniques like clustering and visual encodings to assess policies and infrastructure. Future research aims at integrating semantic data and enhancing scalability for large datasets. Overall, geo-visualization supports data-driven planning by revealing trends and optimizing city growth.

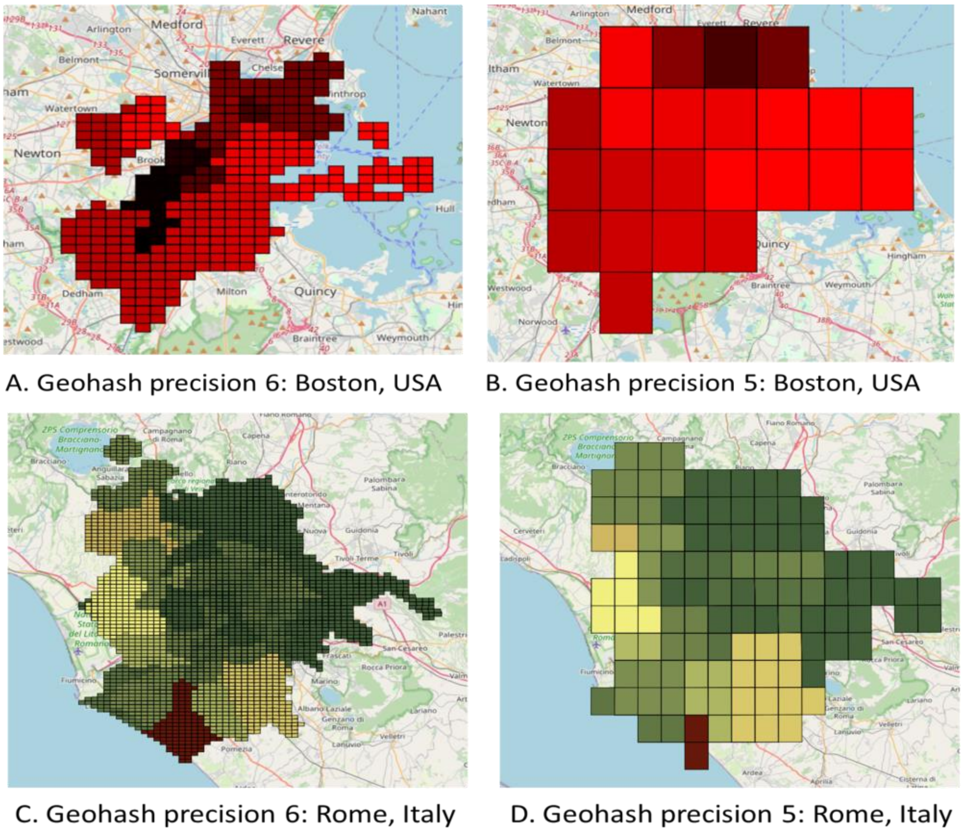
The authors of papers [40–43] emphasize geo-visualization's key role in traffic management, providing insights for congestion management, accident analyses, and route optimization. For instance, ref. [40] focuses on using techniques like heatmaps, flow maps, and 3D visualizations to identify traffic hotspots and optimize routes, supporting real-time monitoring and safer road design. On the other hand, ref. [41] discusses clustering techniques, such as k-means, for mapping accident hotspots and improving congestion control through predictive modeling. Moreover, ref. [42] highlights the benefits of tools like Kepler.gl for visualizing railway and road traffic data to manage congestion, optimize public transport, and improve incident response. Furthermore, ref. [43] addresses congestion forecasting, employing multi-period hotspot clustering and map-matching algorithms to predict traffic patterns and inform infrastructure planning. Together, these papers underscore geo-visualization's impact on improving urban mobility, enhancing safety, and enabling proactive traffic management through real-time data integration and predictive modeling.

2.2. Geohash as a Dimensionality Reduction Approach

Performing extensive geospatial analytics on vast amounts of georeferenced data involves two primary phases: data representation and access data structure. The data representation, which can be either space-driven or data-driven, forms the foundation for the analysis. The embedding space, from which the data are extracted, can either be represented as regularly shaped grids of uniform size or arbitrary shapes. Access data structures are then imposed on these representations to facilitate efficient retrieval of data for spatial queries. These structures enable faster, targeted scans, ensuring that the system can process geospatial data effectively, especially in large-scale settings.

In grid decomposition, cells are often assigned an ordering, which is then subjected to a tree-based access structure (such as a B+-tree) [5]. This process reduces the dimensionality of the data by projecting multidimensional cells into a one-dimensional space. Among the different types of orderings, this study focuses on z-order curves, which offer a structured and computationally efficient approach to representing geospatial data.

Geohash [2] is a unique use of z-order curves, where a z-shaped ordering is applied to the grid space and geocodes are strings with a shared prefix that indicates geometrically nearby spatial coordinates; the greater the shared prefix, the closer the items involved are in real geometries. This characteristic makes geohash encoding highly effective for applications such as proximity searches and region-based data aggregation. The dimensionality reduction provided by geohash encoding enables the efficient representation of geographic areas while maintaining their spatial relationships. An example of a quick-and-dirty proximity search that functions as a quick-and-dirty sieve is geohash encoding. For example, a geohash string's length determines the level of precision: longer geohashes represent smaller, more granular areas, while shorter geohashes correspond to larger, coarser regions. Figure 3 shows the geohash covering generated for A, Boston, USA, with a precision of 6; B shows the geohash covering the same city, but with a lower precision, 5. Precision means the number of characters in the geohash value, in string representation. For instance, 'sr2yk0' represents one of the boxes covering Rome, Italy, in Figure 3C, while 'sr2yk' is the value of one of the rectangles covering the city of Rome at precision 5 as shown in Figure 3D. Precision 6 offers finer granularity to precision 5 as the lower precision value represents a coarser resolution.



A. Geohash precision 6: Boston, USA

B. Geohash precision 5: Boston, USA

C. Geohash precision 6: Rome, Italy

D. Geohash precision 5: Rome, Italy

Figure 3. (**A**,**B**) Geohash tessellation for Boston, USA. (**C**,**D**) Geohash tessellation for Rome, Italy. Geohash tessellation for Boston, USA (**A**,**B**), and Rome, Italy (**C**,**D**). (**A**,**C**) show geohash coverings at precision level 6, characterized by smaller, more granular grid cells, while (**B**,**D**) illustrate coverings at precision level 5, with larger, coarser grid cells. Colors represent distinct geohash values within each precision level, highlighting the spatial distribution of data points. Precision corresponds to the number of characters in the geohash value (e.g., sr2yk0 at precision 6 in (**C**), and sr2yk at precision 5 in (**D**)), where higher precision offers finer spatial detail, and lower precision aggregates larger spatial areas.

Geohash-based tessellation serves as the foundation for our stratified sampling approach in ApproxGeoMap (AGM). Stratified sampling was chosen due to its potential to improve accuracy by reducing variance within strata when spatial correlation is strong. By treating each geohash grid cell as a stratum, this method ensures spatially localized sampling, which is particularly advantageous for maintaining accuracy in the generation of region-based geo-maps [2].

However, we acknowledge that stratified sampling is not universally superior to random sampling. Its efficiency depends significantly on the spatial correlation of the data. When spatial correlation is weak, stratified sampling may result in a "stratification trap", where predefined strata diverge from the true distribution of geospatial objects, leading to inefficiencies. This risk must be mitigated by carefully analyzing the spatial correlation characteristics of the dataset prior to designing strata [44].

For example, strong spatial correlation enables stratified sampling to enhance accuracy by aligning sample distributions with real-world patterns, thereby reducing within-stratum variance. In contrast, when spatial correlation is weak, stratified sampling may perform no better than random sampling. To avoid this issue, AGM incorporates a feedback mechanism that evaluates sampling efficiency and adjusts strata dynamically based on the observed spatial correlation, as discussed in Section 4.

The decision to use geohash-based stratified sampling stems from its ability to partition geospatial data into well-defined grid cells, ensuring computational efficiency and spatial locality. Moreover, this approach minimizes the risk of data overloading during spikes in arrival rates, providing AGM with the scalability needed for real-world urban applications.

2.3. Earth Mover's Distance (EMD)

A distance-based metric called Earth Mover's Distance (EMD) (http://infolab.stanford.edu/pub/cstr/reports/cs/tr/99/1620/CS-TR-99-1620.ch4.pdf 13 September 2024) can be used to compare the (dis)similarity of two frequency distributions, measurements, or densities over a given area. It is often used in the context of comparing weighted point sets or probability distributions. Known also as the Wasserstein metric, EMD is a reliable technique that is typically applied to distribution comparisons. It was initially coined in 1781 by Gaspard Monge, in the context of transportation theory (https://en.wikipedia.org/wiki/Earth_mover's_distance 13 September 2024).

It represents the minimum amount of "work" required to transform one distribution into another, where "work" is defined as the cost of moving "weight" or "mass" from one point to another. The cost is proportional to the amount of weight moved and the distance over which it is moved.

This heuristic overview is comparable to calculating EMD (Earth Mover's Distance). We envision a two-dimensional representation of the earth that is flattened down and has two perspectives of the identical area of the earth: one with holes and the other with heaps/piles of dust that are piling/building up. The least amount of cost needed to cover the holes with dust from piles is then captured by EMD. It is the same as multiplying the quantity of dust transferred from piles to holes by the ground distance that the dust is transferred over to. Converting one distribution into another (and hence solving the optimal transport problem) is comparable.

In EMD terms, the two distributions are represented by what are called signatures. Assume that any distribution can be divided into groups or clusters. Each distribution's signature is the set of all pairs that include a single representative point from each cluster (such as the center) and the proportion of the distribution that is present in that cluster (also called the weight). EMD can then be computed with (1).

$$EMD(P,Q) = \frac{min}{F} = \sum_{i=1}^{m} \sum_{j=1}^{n} fi, j \cdot di, j \tag{1}$$

It is subject to a few constraints defined by Equations (2) to (4):

$$\sum_{j=1}^{n} f(i,j) \leq Wpi, 1 \leq i \leq m \tag{2}$$

$$\sum_{i=1}^{m} f(i,j) \leq Wqj, 1 \leq j \leq n \tag{3}$$

$$1 \leq i \leq m, 1 \leq j \leq n \Longrightarrow f(i,j) \geq 0 \tag{4}$$

(2) ensures that the outflow from every point in distribution P to fill every point in Q is at most equal to the cluster's weight in p. Next, (3) ensures that the inflow to a particular cluster j in Q from all P clusters is at most equal to the weight of that Q cluster.

We now give a hypothetical example to illustrate the EMD. We create a grid of equal-sized tiles (16 tiles in total) out of the embedding space. Positions in those dimensions are represented by monotonically increasing numbers on the x and y axes. Each grid cell's (tile's) weights are expressed as a percentage, and in each distribution, the total of all grid cells equals 1.

We utilize the Manhattan distance to calculate the ground distance for the EMD. For instance, cells (2,2) from distribution P and (3,4) from distribution Q are separated by a distance. The visulization of this example is shown in Figure 4. The Manhattan distance between two corresponding tiles, (x1,y1) in P and (x2,y2) in Q, is defined by Formula (5). The cost of the transfer from P to Q is the product of the flow and the Manhattan distance. For example, if 0.1 units of mass are moved from tile (2,2) in P to tile (3,4) in Q with a Manhattan distance of 3 (as defined in Formula (6)), the resulting cost would be 0.3 as shown in Formula (7). Such an application of EMD is so common for comparing raster data (pixelized data) [44].

$$^{\mathrm{d}}\mathrm{Manhattan} = |\mathrm{x2} - \mathrm{x1}| + |\mathrm{y2} - \mathrm{y1}| \tag{5}$$

$$\text{Substituting values: } ^{\mathrm{d}}\mathrm{Manhattan} = |3 - 2| + |4 - 2| = 1 + 2 = 3 \tag{6}$$

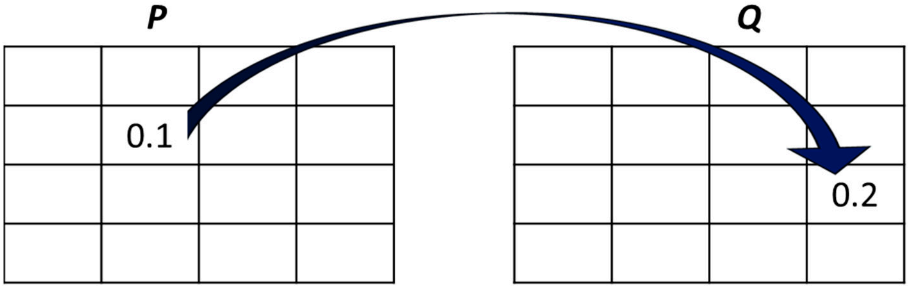


Figure 4. Earth Movers' Distance explanation. Visualization of Earth Mover's Distance (EMD), illustrating the "work" required to transform one distribution into another by moving mass (dust) from piles to fill holes, with cost proportional to the amount moved and the distance covered.

The cost of the transfer from P to Q is the product of the flow and the Manhattan distance:

$$\mathrm{Cost} = 0.1 \times 3 = 0.3 \tag{7}$$

To minimize the EMD, the challenge is to identify a flow F that minimizes the overall cost. This requires calculating the flow and cost for all tile pairs, then finding the combination of flows that results in the lowest total cost, representing the Earth Mover's Distance between P and Q. The total cost represents the dissimilarity between the distributions.

2.4. Geospatial Data Modeling, Reduction, and Sampling

Two requirements must be met when working with massive multidimensional data streams in order to ensure that data processing stays under QoS guarantee criteria. These are representations of geospatial data that are successful by enforcing access structures on them [13,45]. There are two primary categories of geospatial data representation models: data-driven and space-driven. The embedding space is the typical term for the data withdrawal space, which is where the data are extracted. Similarly to order-preserving hash functions, space-driven algorithms divide the embedding space (the geographic region from which data are extracted) using grid files and quadtrees, for instance.

However, data-driven approaches rely on splitting the data items themselves using tools like R-trees and KD-trees. Space-driven grids can be further divided into equal-sized grids and grids with arbitrary sizes. Spatial data structures, which help to expedite access to target data in accordance with the spatial representation and distribution of data, usually follow this data representation [5]. In other words, it is essential to first represent the data using a suitable multidimensional model, such as a tree-based model, before applying a spatial data structure to the model representation to enable speedy access to the data, such as B+-trees, in order to work with large amounts of geospatial data efficiently.

Modern Geographic Information Systems (GISs) use a technique called "ordering", which reduces multidimensional data to single dimensions; z-order curves are one example, to further improve the modeling and accessibility of geographical data. A tree-based spatial access, like B+-trees overlaying the ordering, is applied after the embedding space representation (such as a grid representation) has been ordered. This allows for faster and more efficient access to the data. The geohash encoding is a crucial illustration from the family of applications involving z-order curves. The cells that represent the adjacent grid cell decomposition of an embedding space are essentially represented by a string. The greater the shared prefix, the closer the spatial objects are in real geometries; geospatial objects with equal geohash prefixes typically belong to the same grid cell. To expedite the processing of massive volumes of geographic tuples, geohash encoding and other dimensionality reduction z-order-based techniques, like Google's S2 (http://s2geometry.io/ 1 October 2024) and Uber's H3 (https://www.uber.com/en-AE/blog/h3/ 1 October 2024), are crucial tools. In this context, geohash is used as a fast and accurate filter for spatial proximity queries. Figure 3 shows the geohash covering generated for A, Boston, USA, with a precision of 6; B shows the geohash covering the same city, but with a lower precision, 5. Precision means the number of characters in the geohash value, in string representation. Geohash strings are usually 1–12 characters long. The higher the number of characters, the higher the granularity or resolution. For instance, 'sr2yk0' represents one of the boxes covering Rome, Italy, in Figure 3C, while 'sr2yk' is the value of one of the rectangles covering the city of Rome at precision 5 as shown in Figure 3D.

Sampling is one of the most crucial elements of SAQP techniques. The process of choosing a representative subset of a population to estimate an unknown population parameter value, like a "mean" or "count", is known as sampling in statistics. The method used to choose a sample of units or locations is known as the sampling design. Nonetheless, there is consensus that the sample needs to be representative of the population it is chosen from. To put it another way, a sample is a smaller portion of the population that accurately captures and reflects the traits of the population it represents.

It is impossible to achieve the goal of a completely representative sample that is perfect. But typically, we look for a sample that reflects reality in a way that helps convey the characteristics of the study variables to a believable degree of accuracy and confidence. One of the common issues that leads to some sampling methods being deemed poor is "selection bias", which occurs when a sampling technique intentionally ignores a small portion of the population [46].

Simple random sampling, or SRS for short, and stratified sampling, or SS for short, are the two main sample designs found in the literature. Every unit in a population is given an equal selection probability by SRS, which then labels each unit and chooses labels at random until a predefined number of unique units—equal to the sample size—are collected. From each separate category in the data, stratified-based sample designs choose equal or non-equal parts, such as 50% male and 50% female students from a school's student body [46], or for instance 25% from each geohash value in the data. Because stratified-like sample techniques are known to produce superior estimations than their random-based counterparts, they are favored above their counterparts for the overall characteristics they provide [46].

Stratified sampling is particularly effective in geospatial contexts when spatial correlation within strata is strong. In such cases, it reduces within-stratum variance and improves sampling efficiency. However, this approach is not inherently superior to random sampling in all situations. When the strata in the sample differ significantly from the true strata of the geospatial objects, it might lead to lower efficiency than random sampling. Therefore, it is vital to consider the spatial correlation characteristics of the data when designing stratified sampling strategies [30]. For instance, strong spatial correlation allows stratified sampling based on prior knowledge to improve accuracy and reduce variance. In contrast, weak spatial correlation makes stratified sampling no more effective than random sampling.

Seeking predictable solutions for complex spatial queries and geo-map display in real time is getting less convenient due to the rapidly incoming floods of massive, overwhelming geo-referenced data streams. The issue is made worse by the fact that geographic data are multidimensional, have intricate data structures, and exhibit skewness and oscillation in data arrival rates. SAQP solutions that capture approximations with error bounds are highly valued in the literature in the fields of geo-statistics and geo-visualization [47]. Stratified sampling was chosen for ApproxGeoMap (AGM) because it provides predictable solutions for complex spatial queries and real-time geo-map rendering. SAQP's ability to balance accuracy and processing efficiency makes it a robust solution for creating region-based approximate geo-maps from large-scale geotagged data. Moreover, stratified sampling effectively addresses the practical constraints of observing entire populations, such as tracking migratory birds across vast areas.

That said, the design of stratified sampling in ApproxGeoMap explicitly considers spatial correlation characteristics to minimize stratification traps. This ensures that within-stratum variance is reduced, and sampling efficiency is maximized, particularly for datasets with strong spatial dependencies. Future work could explore adaptive sampling designs that dynamically adjust to varying spatial correlation strengths, further enhancing the robustness of the system.

2.5. Challenges Associated with Geo-Visualization of Big Data

Supporting query assessments for large-scale explorative visualization has significant difficulties, particularly when working with huge spatiotemporal datasets. The amount of data points that correspond to a user's region of interest frequently surpasses the perceptual scalability limit at the sizes taken into consideration, which is one of the most urgent issues [48]. Because individual points grow too small to distinguish, consumers find it challenging to interpret visual data in a meaningful way. Therefore, in order to decrease the data to acceptable levels and improve the user's interaction with the display, an efficient and adaptable aggregation strategy is essential. The proliferation of spatiotemporal datasets from social sensors, IoT devices, and urban environments is another major problem. These datasets are not only vast but also exhibit high complexity, often requiring resource-intensive spatial and temporal query processing. Traditional analyses tend to focus on precise, confirmatory questions due to the computational cost, limiting the potential for interactive exploration [49]. However, exploratory analyses demand real-time, interactive response times, particularly for operations like drill-downs, zooming, and panning across multiple regions. High query latency disrupts the user experience, slowing down observations and impeding the generation of insights. To ensure that users may constantly develop insights without disrupting their train of thought during interactions like drill-downs or panning over multiple regions, exploratory analyses, on the other hand, require interactive reaction times. Users' ability to make observations and draw generalizations is slowed down by high query latency [49]. A number of technological issues need to be resolved for visualization systems to function well in these kinds of situations, one of which is preserving interactivity in spite of the massive amount of data being processed. Ensuring interactivity under these conditions is critical for effective geo-visualization systems.

Dynamic user interaction is another core challenge in geo-visualization. Users must be able to engage with data in a dynamic manner, allowing them to pan, zoom, and change parameters without experiencing any noticeable lag. However, as the dataset size grows, these interactions become increasingly demanding. The rendering process is further complicated by the fact that selectable overlays, which enable the visualization of numerous datasets concurrently, are frequently required to highlight particular patterns [50].

Data management at scale is a key technical issue. Visualization systems need to effectively handle memory usage, computation, network transfers, and disk accesses. Both the client-side and server-side infrastructures may be strained as a result of these operations rapidly surpassing resource limits as data volumes rise. Visualization systems must reduce client–server interactions and shift more work to the client in order to scale efficiently, particularly when there are numerous users interacting at once [50]. However, considering the limitations of memory hierarchy, striking this balance while maintaining real-time interactions is not simple. Although datasets are frequently stored on disks, some must be made memory-resident for quick access, and delays may result from the sharp latency and bandwidth disparities between disks and memory as does the management of limited client-side resources [51].

Furthermore, maintaining quick response times during query evaluations is challenging due to the growing number of data points, whether from social media, urban sensors, or other sources [51]. The intricacy of spatial queries, particularly those involving point-in-polygon checks, which become computationally costly when working with complex-shaped polygons in the real world, exacerbates this difficulty [49]. Finally, network I/O introduces an additional level of complexity because it frequently necessitates the transmission of huge datasets over the network, which lengthens response times. The challenges of transporting, processing, and presenting the data required for real-time visualization increase with data volumes [51]. In conclusion, a significant problem is presented by the combination of increasing data volumes, intricate queries, and the requirement for interactive visualization. Perceptual scalability, computational efficiency, memory hierarchy management, and network limitations must all be addressed in solutions while maintaining the capacity for users to perform insightful, real-time exploratory research. Potential solutions to these problems include aggregation techniques, GPU acceleration, and sophisticated caching schemes; nonetheless, striking a balance between accuracy and response time is still a complex matter that needs constant attention [49]. By addressing these challenges, geo-visualization systems can better support an insightful, real-time exploratory analysis in the era of big data.

2.6. Problem Formulation for ApproxGeoMap

To provide a clearer understanding of how the key components of our unique system, ApproxGeoMap, operate, we present a series of formal definitions in this section as a foundation.

Definition 1. *Geospatial Data. A spatial dataset consists of several georeferenced tuples in the form of (long, lat, [values]), where long and lat represent the coordinates (longitude and latitude), and the dataset is represented as*

$$\mathrm{D} = [(\mathrm{long}_1, \mathrm{lat}_1, \mathrm{values}_1), (\mathrm{x}_2, \mathrm{y}_2, \mathrm{values}_2), \ldots, (\mathrm{x}_\mathrm{n}, \mathrm{y}_\mathrm{n}, \mathrm{values}_\mathrm{n})] \tag{8}$$

where $|D| = n$ is the number of data tuples in the dataset. The geospatial data can be encoded using geohashes, resulting in

$$\mathrm{D} = [(\mathrm{long}_1, \mathrm{lat}_1, \mathrm{values}_1, \mathrm{geo}_1), (\mathrm{x}_2, \mathrm{y}_2, \mathrm{values}_2, \mathrm{geo}_2), \ldots, (\mathrm{x}_\mathrm{n}, \mathrm{y}_\mathrm{n}, \mathrm{values}_\mathrm{n}, \mathrm{geo}_\mathrm{n})] \tag{9}$$

Definition 2. *Geospatial Sampling. A geospatial sample is a subset of the geospatial dataset, such that*

$$\mathrm{S} \subset \mathrm{D} = \{(\mathrm{long}_1, \mathrm{lat}_1, \mathrm{values}_1, \mathrm{geo}_1), (\mathrm{long}_2, \mathrm{lat}_2, \mathrm{values}_2, \mathrm{geo}_2), \ldots, (\mathrm{long}_\mathrm{m}, \mathrm{lat}_\mathrm{m}, \mathrm{values}_\mathrm{m}, \mathrm{geo}_\mathrm{m})\} \tag{10}$$

where $|S| = m$ is the size of the sample, with $m \leq n$. The sample is selected based on stratified sampling over the geohashes covering the area, ensuring representative data selection from each geohash region.

Definition 3. *Geohash Cover. A geohash cover is the list of all geohashes that cover the spatial polygons of the study area, represented as*

$$\mathrm{cover} = [\mathrm{g}_1, \mathrm{g}_2, \ldots, \mathrm{g}_\mathrm{n}] \tag{11}$$

If the geohash cover is reduced using some optimization method (e.g., reducing the number of geohashes while retaining spatial coverage), the reduced geohash cover is

$$\mathrm{reduced\ Cover} = [\mathrm{g}_1, \mathrm{g}_2, \ldots, \mathrm{g}_\mathrm{m}], \text{ where reduced Cover} \subseteq \mathrm{cover}, \mathrm{m} \leq \mathrm{n} \tag{12}$$

Definition 4. *Proxy Aggregation. For each geohash g_i in the geohash cover or reduced geohash cover, the aggregated data A can be defined as a summarized representation of the sampled data:*

$$\mathrm{A} = \{(\mathrm{g}_1, \mathrm{aggValues}_1), (\mathrm{g}_2, \mathrm{aggValues}_2), \ldots, (\mathrm{g}_\mathrm{m}, \mathrm{aggValues}_\mathrm{m})\} \tag{13}$$

where $aggValues_i$ represents the aggregation function applied to the values within each geohash region (e.g., summing counts or averaging values).

Definition 5. *Thematic Map. A thematic map (such as a choropleth) can be generated based on the aggregated geohash data. Let M be the matrix that represents the spatial distribution of the aggregated values. The thematic map function F takes the matrix as input and outputs a rendered map:*

$$\mathrm{thematicMap} = \mathrm{F}(\mathrm{M}) \tag{14}$$

where M is generated based on the geohashes and aggregated data:

$$\mathrm{M} = \mathrm{generateMatrix}(\mathrm{A}) \tag{15}$$

Definition 6. *Error Estimation. Let E be the error function that measures the difference between the original geospatial dataset D and the aggregated dataset A. The error estimation function can be defined as*

$$\text{errorEstimate} = \text{E(D,A)} \tag{16}$$

This function computes the error metric, which, in this paper, is the Earth Mover's Distance (EMD).

3. Literature Review

The rapid growth of geospatial big data has necessitated the development of efficient processing techniques to manage and analyze these datasets effectively. Therefore, this literature review focuses on key advancements in approximate query processing, spatial join optimization, spatial sampling, geohash encoding, spatial partitioning, enhancing map rendering and interaction experiences, and spatial query optimization. By examining recent contributions in these areas, we can better understand the challenges and solutions proposed to enhance the performance and scalability of geospatial big data analytics. Spatial approximate query processing (SAQP) has seen significant advancements in the context of geospatial big data analytics. Several studies have developed systems and techniques aimed at reducing computational costs while maintaining acceptable accuracy. For instance, in [52], the authors introduced GeoMapComp, which focuses on enabling fast and approximate comparisons of satellite remote sensing products. This is achieved through the conversion of raster geo-maps into vectorized representations and the use of geohash encoding, ensuring efficient spatial data processing. Similarly, in [53], the focus is on approximate query processing by employing polygon simplification to reduce data complexity while ensuring fast geospatial aggregation queries. Moreover, in [9], approximate query processing is used to optimize spatial joins, thus improving real-time processing efficiency. Additionally, the study in [4] contributes by simplifying polygon shapes using the Ramer–Douglas–Peucker algorithm, enabling faster approximate analytics for large-scale geospatial data. Furthermore, the EMDI system described in [1] leverages approximate methods to handle heterogeneous geospatial data integration efficiently. Likewise, in [6], ApproxSSPS efficiently processes geospatial data streams using approximation techniques to balance query precision with system performance. Finally, the authors of [45] introduce the SAOS algorithm, which uses spatially aware sampling to facilitate efficient approximate query processing in real-time geospatial data streams. Recent advancements in enhancing map rendering and interaction complement these efforts by integrating simplification techniques like stylized hierarchical symbol models, which prioritize the visualization of critical information while reducing computational overhead [54,55].

A spatial join has been extensively explored in geospatial data analytics, with several studies focusing on optimizing the process of combining geospatial datasets. For instance, ref. [53] employs a filter-and-refine approach to spatial joins, using geohash encoding to filter candidate points before applying exact point-in-polygon operations. Similarly, in [9], the system performs stream-static spatial joins, enabling the combination of real-time geospatial data streams with static geographic datasets. Moreover, the EMDI system described in [1] focuses on integrating mobility and pollution datasets through spatial join processing, ensuring efficient and accurate analytics. Lastly, in [56], spatial join optimization is integrated within a distributed Spark framework, benefiting from load balancing and geospatial indexing to enhance query performance.

Spatial sampling is another critical area in geospatial analytics. Several studies have focused on reducing computational load while preserving the accuracy of spatial queries through sampling techniques. For example, in [52], stratified sampling is utilized to generate comparable samples from geospatial data, ensuring efficient processing of large datasets.

Similarly, in [4], spatial sampling is incorporated to maintain geographical distribution while reducing data size. Furthermore, in [13], the ex-SAOS system is introduced to ensure fair representation across regions during sampling, minimizing errors in geospatial queries. On the other hand, in [6], spatial-aware sampling techniques like SAOS are employed to balance query efficiency with accuracy in real-time data processing. Finally, ref. [45] uses stratified sampling through the SAOS algorithm to ensure efficient real-time spatial processing by selecting representative data points from various regions. Stylized hierarchical symbol models (SHS) contribute to improving spatial data visualization under constrained conditions by allowing selective rendering of features based on priority, thus enhancing the interpretability of sampled data [54].

Geohash encoding is a widely used technique to efficiently organize and index geospatial data, and its application is evident across multiple studies. For instance, in [4], geohash encoding is used to simplify polygons and group spatial objects efficiently, thus improving computational performance. Similarly, the EMDI system in [1] applies geohash encoding to spatially organize mobility and pollution data, enabling quick proximity searches and optimized spatial joins. Furthermore, in [2], geohash encoding helps reduce spatial data dimensionality, thereby improving the efficiency of spatial joins. Likewise, in [13], geohash encoding enables the efficient organization of geospatial data into grid-based representations for real-time processing. Additionally, in [6], geohash encoding is used to group spatial data points into grid cells, aiding efficient spatial indexing. Lastly, in [56], geohash encoding is used to optimize spatial indexing in a distributed Spark environment, improving the execution of spatial queries.

Spatial partitioning is essential for efficiently distributing geospatial data across multiple computing nodes, particularly in distributed environments. For example, in [2], the SCAP system is introduced to minimize data shuffling by preserving spatial locality, thus improving the performance of spatial queries. Similarly, in [56], advanced spatial partitioning techniques are implemented to evenly distribute workloads across nodes, optimizing query performance in large-scale geospatial data analytics.

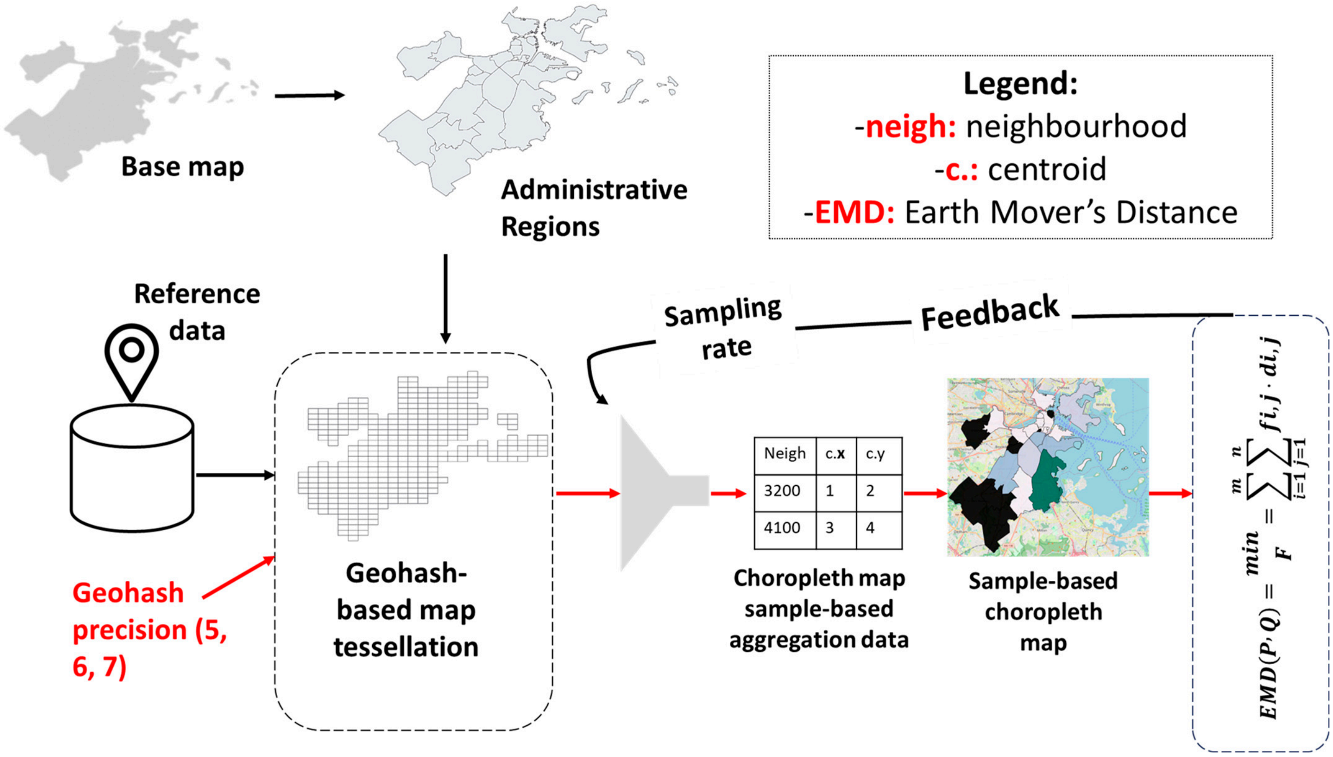
Enhancing map rendering and interaction experiences is a growing area of interest in geospatial visualization research. For example, the stylized hierarchical symbol (SHS) model discussed in [54] introduces a weight-based prioritization mechanism to improve rendering performance, ensuring that critical map features are highlighted while non-essential details are omitted during constrained operations. Similarly, the Geographic Feature Color-weighted Rendering (GFCR) technique [54] enhances user interaction by dynamically adjusting feature importance based on user input, allowing for smoother transitions between zoom levels. Additionally, adaptive caching methods like Catalan Number-based Caching Access (CatNCa) [54] optimize resource usage, reducing latency during frequent map interactions. Together, these methods address challenges like symbol clutter, interaction latency, and resource constraints, ensuring a seamless user experience even with large geospatial datasets.

Spatial query optimization focuses on improving the performance of complex spatial queries such as proximity detection and spatial joins. In [2], the SCAP system is integrated with a query optimizer to enhance the performance of spatial queries like k-nearest neighbor (kNN) and density-based clustering, reducing computational overhead and improving processing efficiency in distributed systems. Recent advancements in caching and indexing mechanisms, such as grid-based dynamic caching, complement these efforts by accelerating spatial query response times in interactive environments [54,55].

4. ApproxGeoMap Geospatial Visualization at Scale with QoS Guarantees

4.1. Architecture Design and System Operation

In this section, we present the design and implementation of our novel system, ApproxGeoMap (AGM), which is designed for the efficient generation of region-based geo-maps from large-scale geotagged locational data, with a focus on choropleth maps. The system consists of five main components that work sequentially, forming a pipeline: geospatial data modeling and representation, stratified-like geospatial sampling, region-based geo-map proxy generation, geo-map rendering, and a quality of service (QoS) controller. These components work together as a pipeline, as shown in Figure 5, to ensure the efficient processing and visualization of geospatial data. ApproxGeoMap begins by processing two types of input data: raw geotagged tuples, which may number in the millions or billions, and a polygon file representing the study area's spatial boundaries, typically provided in GeoJSON or shapefile format. The geospatial modeling and representation module partitions the study area into a uniform grid using geohash encoding at a predefined precision (e.g., levels 5, 6, or 7). Sometimes, the use of higher precisions is critical; for instance, emergency response systems and environmental hazard monitoring systems require a more specific, granular representation. However, in this study, geohash precisions of 5 and 6 were used to compare the performance of the system with the varying resolutions. This geohash encoding ensures spatial locality, meaning that geographically proximate objects share the same geohash values. This representation is critical to the stratified-like sampling approach used in ApproxGeoMap, where each geohash value is treated as a stratum.


Base map
Administrative Regions
Legend:
-neigh: neighbourhood
-c.: centroid
-EMD: Earth Mover's Distance
Reference data
Sampling rate
Feedback
Geohash precision (5, 6, 7)
Geohash-based map tessellation
Choropleth map sample-based aggregation data
Sample-based choropleth map
$EMD(P \cdot Q) = \underset{F}{\min} = \sum_{i=1}^{m}\sum_{j=1}^{n} fi,j \cdot di,j$

Neigh	c.x	c.y
3200	1	2
4100	3	4

Figure 5. ApproxGeoMap architecture. This figure illustrates the main components and workflow of the ApproxGeoMap system. The process begins with geohash-based tessellation of input data, followed by stratified-like sampling using geohash regions. The system then aggregates geospatial statistics and visualizes the data through choropleth or heatmaps. The colors in the sample-based choropleth map represent different data densities or attribute values, with darker colors indicating higher values and lighter colors representing lower values. A feedback loop evaluates the error using Earth Mover's Distance (EMD) to adjust sampling rates and ensure accuracy.

The data modeling process encodes both the study area polygons and the raw data tuples using geohashes, with the same level of precision applied to both datasets. This results in two intermediate geohash-encoded datasets: one representing the polygons and the other representing the raw locational tuples. Since each polygon in the study area is covered by multiple geohash values, the system can perform stratified sampling based on these geohash codes. The ApproxGeoMap system then proceeds through its workflow, as described by Algorithm 1, ApproxGeoMap Workflow1, below:

Algorithm 1: ApproxGeoMap Workflow

```
    Input: refData, geoPrec, mapRenderType, sampFraction, seed
    // Step 1: Geohash Tessellation
    geoHashMap ← ∅  // Initialize an empty map for geohashes
    For each dataPoint in refData do
        geoHash ← geohashEncode(dataPoint.lat, dataPoint.long, geoPrec)  // Encode
geohash
        geoHashMap[geoHash].add(dataPoint)  // Group data points by geohash
    End
    // Step 2: Stratified Sampling
    sampledData ← AGMSampler(geoHashMap, sampFraction, seed)  // Sample data from
each geohash
    / / Step 3: Aggregation
    aggregatedData ← proxyAggregate(sampledData)  // Aggregate the sampled data
    / / Step 4: Map Rendering
    thematicMap ← (mapRenderType == "choropleth") ? choroplethRender(generate
Matrix(aggregatedData)): heatmapRender(generateMatrix(aggregatedData))  // Render the
map
    // Step 5: Error Estimation
    errorEstimate ← calculateError(refData, aggregatedData)  // Estimate error between
original and aggregated data
    // Output
    output thematicMap, errorEstimate, adjustSamplingRate(errorEstimate)  // Output
map, error, and sampling feedback
                            End
```

In Step 1 of Algorithm 1, ApproxGeoMap Workflow, geohash tessellation encodes the geographic space and splits it into geohash tiles. Each geotagged tuple from the raw dataset is grouped according to its geohash code, creating a mapping of geohash values to data points. In Step 2, the system applies stratified sampling through the AGMSampler function, as shown below in Algorithm 2, which selects a representative subset of data points from each geohash region based on a predefined sampling fraction. The sampled data are then aggregated in Step 3, where geospatial statistics (such as averages or counts) are computed for each geohash region. Step 4 renders the final map, generating either a choropleth map or a heatmap depending on the user’s specification. Finally, in Step 5, the system estimates the error between the original dataset and the sampled approximation, providing a feedback mechanism to adjust the sampling rate if needed.

The stratified-like sampling process is implemented using Algorithm 2, AGMSampler, described below. This algorithm selects data points from each geohash region based on a random sampling fraction.

Algorithm 2: AGMSampler

```
Input: geoHashMap, sampFraction, seed
r = random(seed)   // Initialize random number generator with seed
sampledData ← ∅   // Initialize empty set for sampled data
For each geoHash in geoHashMap do
    tuples ← geoHashMap[geoHash]   // Retrieve data for current geohash
    fraction ← sampFraction       // Set sampling fraction (can be adjusted per geohash)
    For each tuple in tuples do
        If (r < fraction) then    // Randomly sample each tuple based on fraction
            sampledData.add(tuple)
        End
    End
End
return sampledData
End
Output: sampledData
```

The AGMSampler works by iterating through each geohash region and applying a random sampling process. For each data tuple within a geohash, the algorithm compares a random value (generated using the seed) with the predefined sampling fraction. If the random value is less than the sampling fraction, the tuple is included in the sample. This ensures that data from each geohash region are sampled independently and proportionally, making the approach akin to stratified sampling. The sampler architecture is shown in Figure 6.

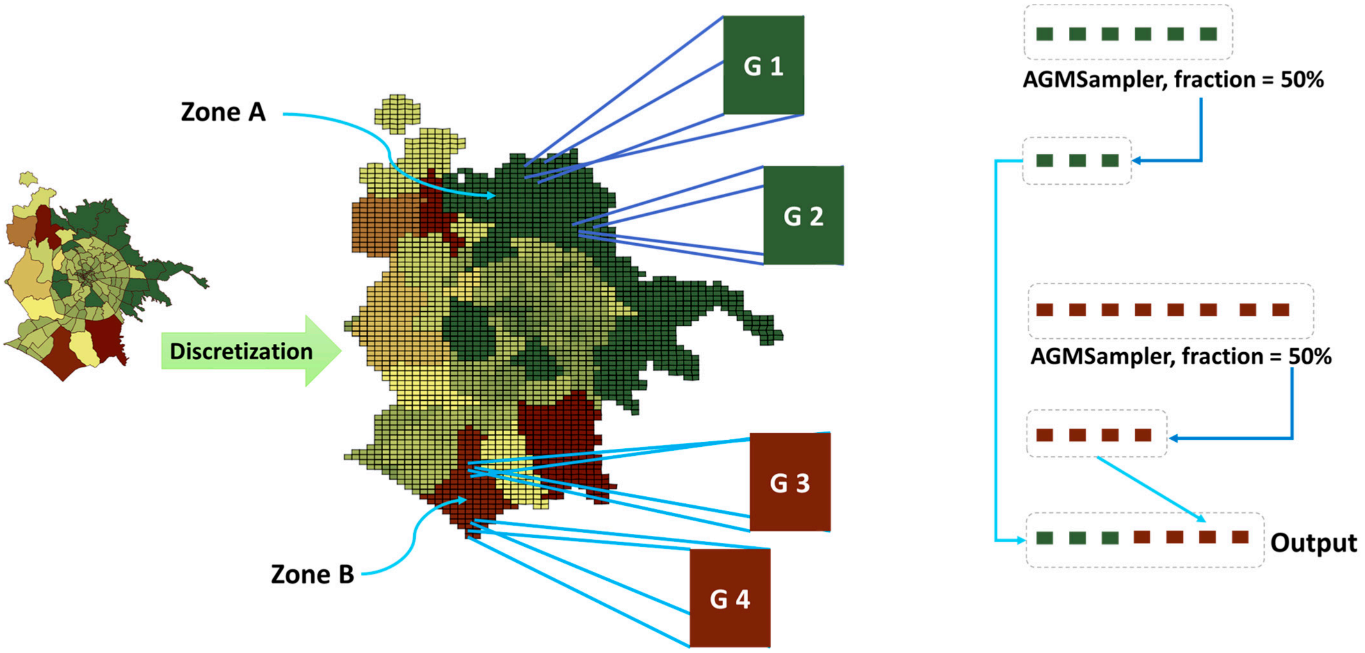


Figure 6. AGMSampler architecture. The AGMSampler implements stratified-like sampling by iterating through geohash regions within distinct zones (Zone A and Zone B) and selecting data points based on a predefined sampling fraction. Discretization divides these zones into grid cells, where G1 and G2 belong to Zone A but have different geohash values, and G3 and G4 belong to Zone B with distinct geohash values. This process ensures proportional sampling across regions, facilitating accurate and efficient proxy-based aggregation for geospatial data.

Once the sampling is complete, the system proceeds to the area-based geo-map proxy generation step. In this stage, the sampled data are aggregated into a compact matrix format, where each row corresponds to a geohash value, along with the aggregated geospatial statistics and the geohash centroid. This matrix can be used to represent data at both fine-grained (geohash-level) and coarse-grained (polygon-level) resolutions, depending on the level of aggregation.

The Geo-Map Renderer then visualizes the aggregated data by generating a choropleth map or heatmap. If the data are provided at a coarse-grained level (e.g., polygon-level), the renderer directly produces the choropleth map. For finer-grained data (geohash-level), the renderer first aggregates the data covering each polygon before generating the map.

Finally, the QoS Controller evaluates the quality of the rendered map by estimating the error between the original geospatial data and the sampled data used for map generation. The error estimation uses Earth Mover's Distance (EMD) to measure the similarity between the two distributions, providing error bounds that are displayed alongside the map to inform the user of the map's accuracy.

By employing geohash-based tessellation, stratified-like sampling, and proxy-based aggregation, AGM provides an efficient pipeline for generating large-scale choropleth maps while maintaining a balance between performance and accuracy. The integration of error estimation using EMD allows users to receive meaningful feedback on the quality of the generated visualizations. The EMD metric quantifies the dissimilarity between the original dataset and the approximated sampled data, ensuring adherence to QoS constraints. Specifically, we calculate EMD using the formula defined in Equation (1), where $flow_{i,j}$ is the flow between P_i and Q_j that minimizes the total cost. $dist_{i,j}$ is the distance between P_i and Q_j. For our implementation, Manhattan distance is used as the ground distance metric, aligning with the spatial data structure of our geohash-based approach. This ensures that the system achieves efficient and reliable results while minimizing computational overhead.

4.2. System Scope of Operation

In this research, we focus on middleware software systems that serve as a bridge between the presentation layer and the data sources. Since our primary focus is on value-by-area maps, our survey is not able to cover many other themed maps, including choropleth maps, graded circle maps, and travel-distance maps [57].

A key component of a buffer-overlay analysis is buffer and overlay generation. Many approaches have been created to address generating issues, and they can be broadly divided into two categories according to the output they produce: raster-based and vector-based techniques. While raster-based approaches use pixels to represent geographical data, which can lead to sawtooth distortions, vector-based approaches use polygons to represent geographic characteristics, allowing for in-depth spatial analyses without sacrificing resolution when zooming in. A raster-based analysis has the advantage of being computationally simpler, but it also uses more storage space and frequently distorts results when zoomed in [44].

Our study focuses on a vector data analysis, which performs spatial computations using geometric objects such as polygons, lines, and points. Although vector-based techniques, such as edge-constrained triangulation, provide great geometric representation accuracy, they are more computationally costly. Despite requiring more storage, raster-based techniques are typically avoided for large-scale spatial data because of their inefficiency. The majority of this field's research focuses on serial computing models, especially when it comes to raster buffers [44].

5. Experimental Evaluation

In this section, we summarize the experimental setup including the datasets and the test settings, in addition to the baseline methods and evaluation metrics used in this study. Furthermore, the experimental results on the datasets are also discussed.

5.1. Experimental Setup

5.1.1. Datasets

The Boston (dataset link: https://zenodo.org/records/7961851), USA, dataset focuses on hyperlocal air quality, collected using mobile sensing platforms from February to April 2022, in a neighborhood near Boston Logan International Airport. The data include measurements of particulate matter (PM2.5, PM10), nitrogen dioxide (NO_2), temperature, and humidity. Collected with a mobile environmental lab, the data offer high spatial and temporal resolution, crucial for understanding air pollution variability in urban areas. Calibration was performed using machine learning models to ensure accuracy. This dataset supports studies on urban air pollution and can help inform environmental policy decisions.

The Rome dataset (dataset link: https://ieee-dataport.org/open-access/crawdad-romataxi) consists of mobility traces collected from approximately 320 taxi cabs over 30 days, between 1 February 2014 and 2 March 2014. The dataset captures GPS coordinates from taxis operating in central Rome, with data points recorded every 7 s. The purpose of the dataset is to analyze user mobility, network performance, human behavior, and opportunistic connectivity. The data were sanitized by replacing driver names with IDs and include the position and timestamp of each taxi, providing insights into urban mobility patterns. The precision of the GPS data is filtered to maintain accuracy within 20 m. This dataset is valuable for studying mobility, communication networks, and urban transportation systems.

In comparing the Boston air quality data with the Rome mobility data, the distinctions in their distributions are apparent, illustrating the use of two different types of datasets for our experiment. The Boston dataset focuses on environmental factors such as temperature, humidity, and pollutant concentrations (NO_2 and PM2.5), which are essential for understanding variations in air quality, as shown in Figure 7. In contrast, the Rome dataset revolves around human mobility patterns, including driver activity, distribution by the day of the week, and the hour of the day, giving insights into urban traffic behaviors as shown in Figure 8. These distinct focuses lead to different data distributions and structures.

The temporal distribution in the two datasets highlights significant differences. For the air quality data in Boston, the records span multiple months, with the majority of data points collected during March and February, and a smaller portion from April. This spread across late winter and early spring allows for an analysis that incorporates seasonal changes. On the other hand, the Rome mobility data are confined to a single month, providing a more concentrated snapshot of mobility patterns without seasonal variations. This distinction in time frames emphasizes how each dataset approaches the temporal analysis from a different perspective.

When analyzing daily and hourly patterns, the mobility data from Rome display clear trends. There are noticeable peaks in activity during morning and evening rush hours, correlating with commuter behavior, and higher activity levels are seen on Fridays and Saturdays. This reflects typical urban mobility patterns, driven by workweek rhythms. In contrast, the air quality data do not exhibit such clear daily or hourly peaks, as they capture continuous environmental variables that change more gradually over time, such as temperature and pollution levels.

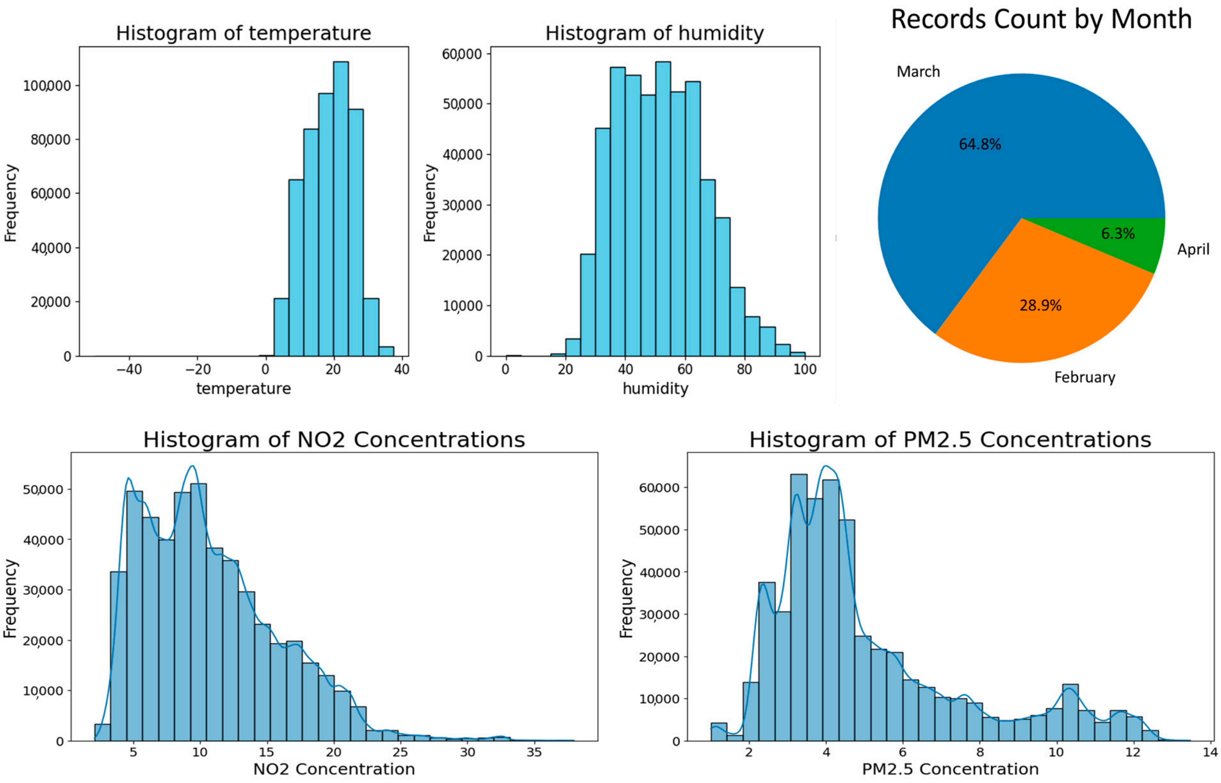


Figure 7. Exploratory Data Analysis: Boston, USA, air quality dataset. The histograms depict temperature, humidity, and pollutant concentrations (NO_2 and PM2.5), highlighting their distributions and frequency across the dataset. The pie chart illustrates the record counts by month, showing the temporal distribution of data collection.

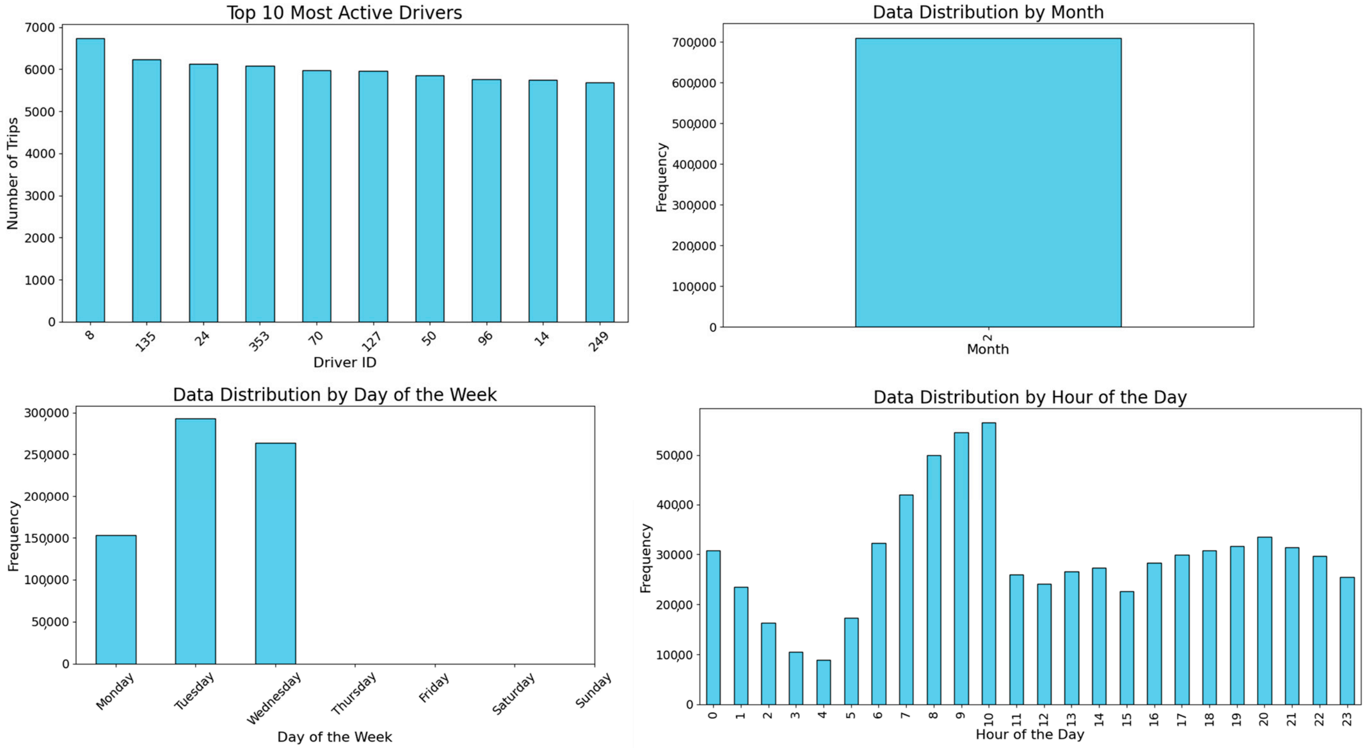


Figure 8. Exploratory Data Analysis: Rome, Italy, mobility dataset. The bar charts represent driver activity (top 10 most active drivers), data distribution by the month, day of the week, and hour of the day, providing insights into urban mobility trends and temporal activity patterns.

The underlying distributions of variables further emphasize the distinction between the two datasets. The Boston air quality data show Gaussian-like distributions for temperature and humidity, with a clear peak around certain values, while NO_2 and PM2.5 concentrations display multimodal distributions, reflecting variability in pollution levels throughout the city. The Rome mobility data, however, focus on discrete variables. For example, the most active drivers exhibit uniform participation, and there are clear patterns in the distribution of activity across days and hours, with peaks corresponding to human behaviors rather than environmental factors.

In summary, the Boston air quality and Rome mobility datasets represent two fundamentally different distributions that were used for the experiment. The Boston data follow a continuous environmental distribution with seasonal elements, while the Rome dataset is discrete, reflecting human mobility patterns. These differences in distribution and data structure ensure that the experiment involves a variety of data types, enriching the analysis with diverse perspectives on environmental conditions and urban mobility.

5.1.2. Deployment

We used the following resources to set up our tests on a Microsoft Azure virtual computer: 4 E8 v3 computers with 64 GB of RAM and eight cores. We used geo-packages like Geopandas to implement our system's standard-compliant prototype in Python.

Evaluation metrics: We use Earth Mover's Distance (EMD) to measure the distance between distributions, and then apply RMSE (Root Mean Squared Error) to quantify the error in our predictions, helping us assess the model's accuracy after calculating the distributional differences.

5.2. Experimental Results and Discussion

This section explores the performance of ApproxGeoMap (AGM) in generating region-based maps (specifically choropleth maps) from big geo-referenced data. We evaluated its efficiency by varying geohash precision levels (5 and 6) and sampling rates, comparing stratified-like sampling (AGM) with simple random sampling as a baseline. Then, we use the EMD measurement to test the performance of the system by applying both samplers. It is worth noting that AGM refers to the ApproxGeoMap system and is introduced here as an abbreviation for clarity.

In Figure 9, which uses geohash precision level 5 (labeled as A), we compare the Earth Mover's Distance (EMD) across different sample sizes for the Boston dataset. At a small sample size of 0.2, random sampling has a significantly higher EMD (0.0012) compared to AGM (0.00005), demonstrating a 96% reduction in error when using AGM. As the sample size increases to 0.5, random sampling's EMD decreases to 0.0003, but AGM still outperforms it with an EMD of 0.00002, achieving a 93% reduction in error. At the largest sample size of 0.9, random sampling reaches an EMD of 0.00005, while AGM's EMD is negligible, showing almost 100% improvement. This result suggests that AGM provides a more accurate representation of the data distribution across all sample sizes, particularly at smaller sizes.

In Figure 9B, which uses geohash precision level 6, similar trends are observed. At the smallest sample size of 0.2, random sampling has an EMD of 0.00065, whereas AGM achieves a much lower EMD of 0.00005, representing a 92% reduction in error. As the sample size increases to 0.5, random sampling's EMD drops to 0.0003, but AGM achieves a near-zero EMD, demonstrating almost 100% improvement. At the largest sample size of 0.9, both methods converge toward similar low EMD values, but AGM maintains consistently better performance.

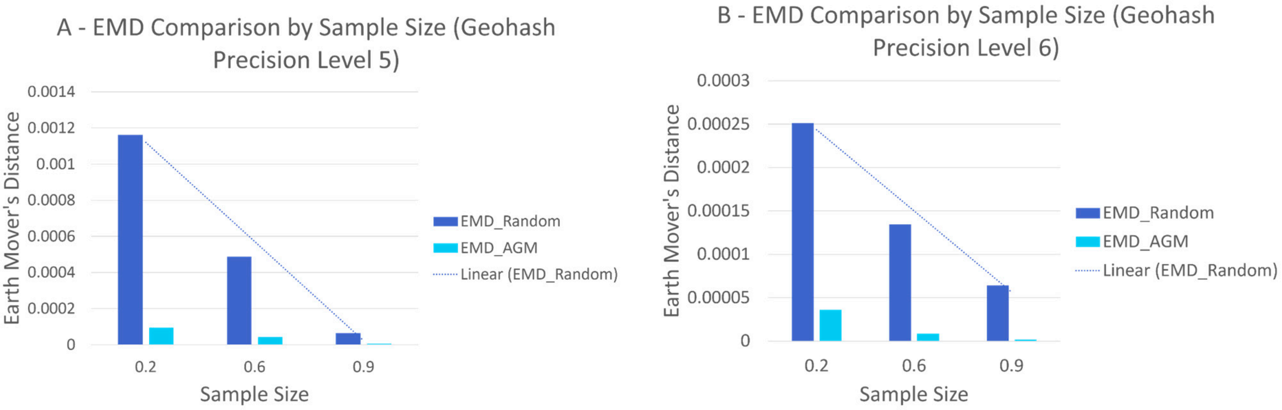


Figure 9. Comparison of EMD for stratified and random sampling across different sample sizes in Boston, USA. Comparison of Earth Mover's Distance (lower better) across varying sample sizes for Random Sampling vs. AGM (Stratified Sampling). (**A**) Results with Geohash Precision Level 5, (**B**) Results with Geohash Precision Level 6.

In Figure 10A, using geohash precision level 5, the Root Mean Squared Error (RMSE) for the Boston dataset is compared between random sampling and AGM. At the smallest sample size of 0.2, random sampling's RMSE is 0.0016, while AGM achieves a much lower RMSE of 0.00005, indicating a 96% reduction in prediction error. At the medium sample size of 0.5, random sampling's RMSE drops to 0.0009, but AGM still outperforms it with 0.00004, reflecting a 95% reduction in error. At the largest sample size of 0.9, random sampling reaches 0.0004, while AGM approaches an RMSE of 0.00001, showing an impressive 99% improvement.

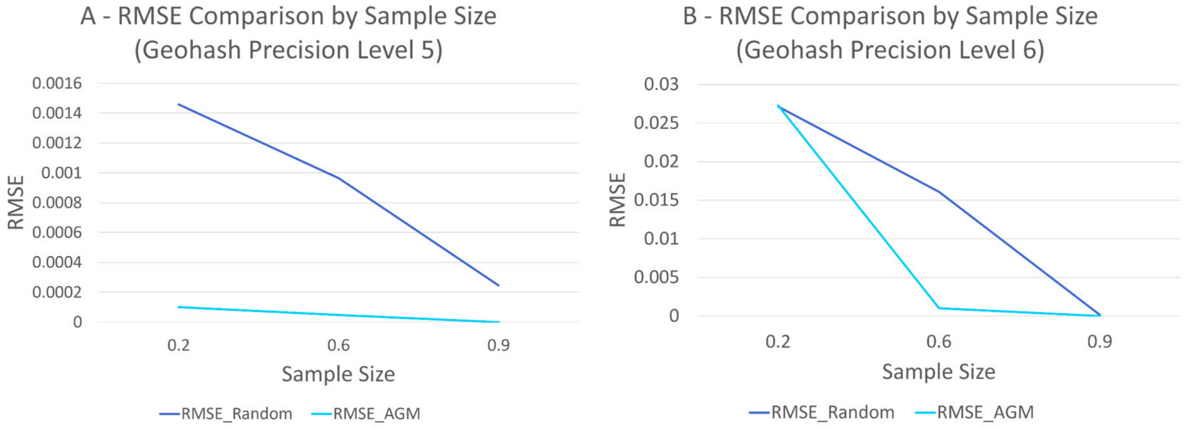


Figure 10. Comparison of RMSE for stratified and random sampling across different sample sizes in Boston, USA. Comparison of RMSE across varying sample sizes for Random Sampling vs. AGM (Stratified Sampling). (**A**) Results with Geohash Precision Level 5, (**B**) Results with Geohash Precision Level 6.

In Figure 10B, with geohash precision level 6, the RMSE analysis shows similar patterns for the Boston dataset. At the smallest sample size of 0.2, random sampling has an RMSE of 0.08, while AGM reduces it to 0.0002, showing a dramatic 99.75% reduction in error. At the sample size of 0.5, random sampling's RMSE remains relatively high at 0.06, while AGM approaches zero, reflecting an almost 100% reduction in error. At the largest sample size of 0.9, random sampling still shows higher RMSE values than AGM, underscoring AGM's superior performance.

For the Rome dataset, in Figure 11A, using geohash precision level 5, the EMD analysis reveals similar trends. At the smallest sample size of 0.2, random sampling has an EMD of 0.00065, while AGM reduces it to 0.00005, indicating a 92% reduction in error. As the sample size increases to 0.5, random sampling's EMD drops to 0.0003, but AGM achieves a near-zero EMD, showing almost 100% improvement. At the largest sample size of 0.9, both methods perform similarly, but AGM consistently maintains better accuracy.

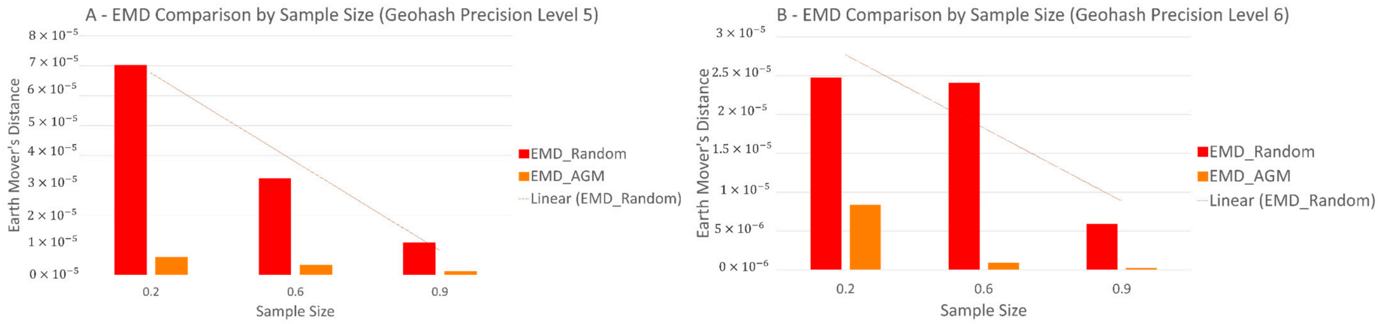


Figure 11. Comparison of EMD for stratified and random sampling across different sample sizes in Rome, Italy. Rome Data: Comparison of Earth Mover's Distance (lower better) across sample sizes for Random Sampling vs. AGM (Stratified Sampling). (**A**) Results with Geohash Precision Level 5, (**B**) Results with Geohash Precision Level 6.

In Figure 11B, using geohash precision level 6, similar results are observed. At a sample size of 0.2, random sampling's EMD is 0.00065, while AGM shows 0.00005, indicating a 92% reduction. As the sample size grows to 0.5, AGM's EMD remains near zero, with random sampling decreasing but still underperforming. AGM consistently achieves better performance across all sample sizes.

Finally, in Figure 12A, using geohash precision level 5, we compare RMSE for the Rome dataset. At the smallest sample size of 0.2, random sampling has an RMSE of 0.08, while AGM drastically reduces it to 0.0002, achieving a 99.75% reduction in prediction error. At the sample size of 0.5, random sampling's RMSE stays at 0.06, while AGM approaches zero, demonstrating almost 100% improvement. At the largest sample size of 0.9, random sampling remains higher in RMSE compared to AGM.

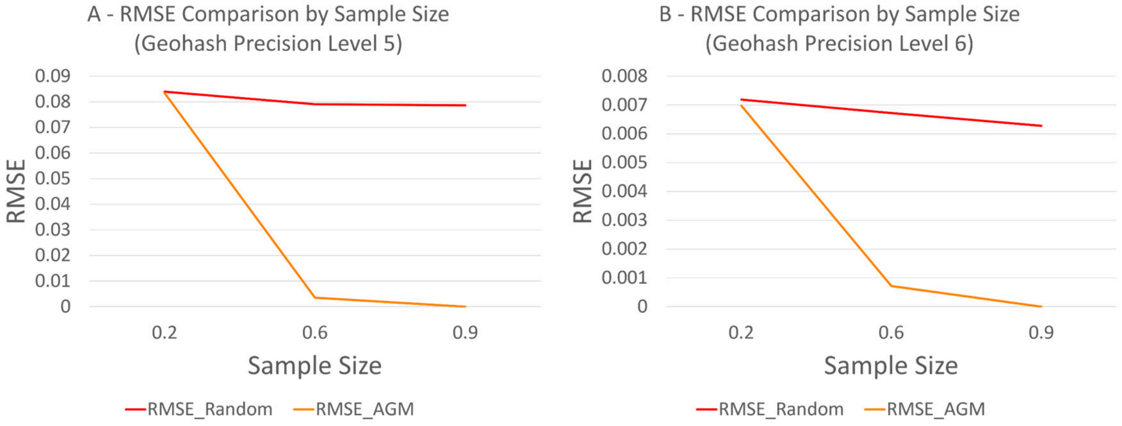
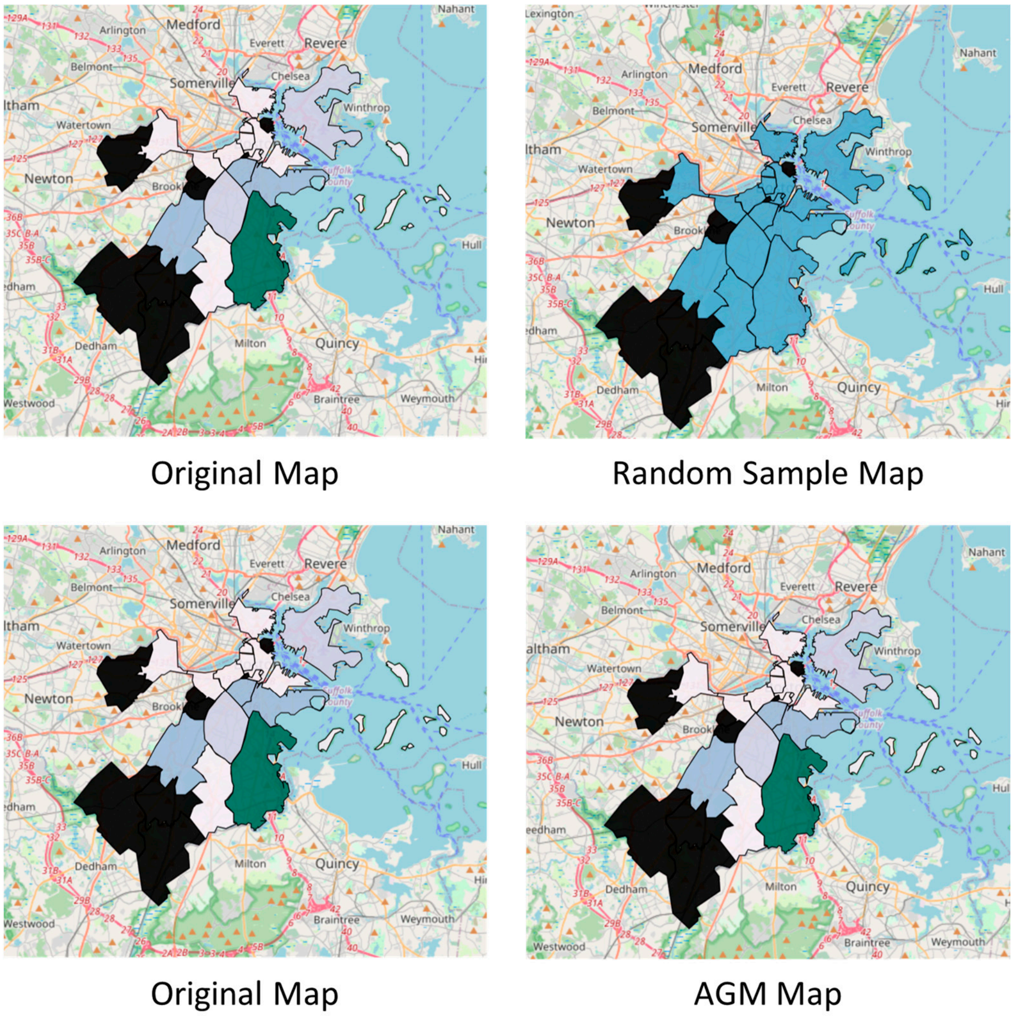


Figure 12. Comparison of RMSE for stratified and random sampling across different sample sizes in Rome, Italy. Rome Data: Comparison of RMSE across varying sample sizes for Random Sampling vs. AGM (Stratified Sampling). (**A**) Results with Geohash Precision Level 5, (**B**) Results with Geohash Precision Level 6.

In Figure 12B, with geohash precision level 6, similar RMSE trends are seen for the Rome dataset. Random sampling starts with an RMSE of 0.08 at a sample size of 0.2, while AGM reduces it significantly to 0.0002, demonstrating a 99.75% reduction in error. At larger sample sizes, AGM continues to outperform random sampling.

In conclusion, across both datasets of Boston and Rome, and geohash precision levels 5 and 6, AGM consistently provides 92% to 99.75% reductions in both EMD and RMSE compared to random sampling, with the improvement being most significant at smaller sample sizes. AGM proves to be a far more efficient and accurate sampling method in both data distributions and prediction accuracy, making it highly reliable for generating maps and region-based analyses.

On the other hand, Figure 13 demonstrates the visual differences between the original dataset and the maps generated using random and AGM sampling techniques with a geohash precision level of 6. The comparison reveals critical insights into the efficacy of AGM sampling in preserving spatial distributions. The original maps serve as the baseline, showcasing the full detail and accuracy of the dataset. These maps capture the true variability and density patterns across regions, providing a high-resolution reference for evaluating sampling methods.


Original Map

Random Sample Map

Original Map

AGM Map

Figure 13. Comparison of original, random sampling, and AGM sampling maps for Boston using geohash precision level 6. The figure showcases density variations in Boston using geohash precision level 6. Black regions indicate the highest density of data points, followed by dark blue (moderately high), light blue (moderate), green (low), and white (lowest or no data). The top left map represents the original Boston dataset, duplicated in the bottom left for consistency. The top right map applies random sampling, which introduces disruptions in spatial patterns, while the bottom right map employs ApproxGeoMap (AGM) sampling, demonstrating superior preservation of density patterns and spatial coherence.

In contrast, the random sampling map introduces significant discrepancies. Certain regions are underrepresented, and abrupt transitions appear in density values, disrupting the spatial coherence of the map. These inaccuracies stem from the random selection process, which fails to account for spatial correlations. The result is a visualization that may mislead decision-making processes, particularly in applications requiring high precision, such as urban planning or environmental monitoring.

Furthermore, the AGM sampling map (bottom-right) closely aligns with the original dataset. By leveraging geohash-based tessellation and stratified-like sampling, AGM preserves both the density gradients and region-specific patterns. This alignment is particularly evident in regions with high variability, where AGM successfully retains the spatial coherence lost in random sampling. The smoother transitions and faithful representation of density distributions make AGM a more reliable method for generating region-based maps under constrained data conditions.

The importance of maintaining precision is further highlighted by the granularity provided at geohash precision level 6. At this level, the maps capture finer details of smaller regions, amplifying the visual discrepancies caused by random sampling. AGM's ability to preserve these details while reducing computational complexity underscores its value for real-time applications, where both speed and accuracy are critical.

In summary, Figure 13 illustrates the limitations of random sampling and the strengths of AGM in retaining the fidelity of spatial patterns. These results emphasize the necessity of choosing sampling methods that account for spatial correlations, particularly in scenarios where even minor errors could lead to significant consequences. The findings further demonstrate the trade-off between accuracy and performance, solidifying AGM's role as an optimal solution for geospatial data visualization.

6. Conclusions

In this paper, we presented ApproxGeoMap, a novel system for efficiently generating approximate aggregate-based geo-maps from fast-arriving georeferenced data streams, addressing the urgent need for responsive visualizations in smart city applications. ApproxGeoMap employs a stratified-like sampling method at the front stage, acting as an intelligent filter that discards excess data loads when data arrival rates exceed processing capabilities. The system's quality of service (QoS) controller dynamically adjusts the sampling rate through a feedback loop mechanism, ensuring that the geo-visualizer receives a manageable volume of data aligned with system capacity. Experimental results confirm that ApproxGeoMap significantly reduces error metrics, such as Earth Mover's Distance (EMD) and Root Mean Squared Error (RMSE), particularly when compared to random sampling methods at smaller sample sizes, thus balancing accuracy with efficiency and overcoming traditional processing limitations for large-scale geospatial data.

The practical implementation of ApproxGeoMap in real-world smart city contexts requires addressing several technical and logistical considerations. For instance, deploying ApproxGeoMap in existing urban infrastructures may necessitate integration with legacy systems, hardware upgrades, and ensuring data privacy compliance. Future research could explore how ApproxGeoMap can be fine-tuned for diverse use cases, such as real-time traffic management, disaster response, and urban planning, where the trade-off between accuracy and speed may vary significantly.

Additionally, future work will focus on developing a distributed computing version atop frameworks like Apache Spark to enhance scalability for even larger datasets. While the current similarity values for error estimation are expert-guided, we aim to develop a mathematically principled algorithm to dynamically set these values based on data stream characteristics, further refining ApproxGeoMap's effectiveness. This will extend its application to broader urban planning, environmental monitoring, and other high-demand geospatial contexts.

Author Contributions: Conceptualization, R.A.A. and I.M.A.J.; methodology, R.A.A. and I.M.A.J.; software, R.A.A. and I.M.A.J.; validation, R.A.A. and L.F.; formal analysis, R.A.A., I.M.A.J. and L.F.; investigation, R.A.A., I.M.A.J. and A.C.; resources, A.C. and L.F.; data curation, R.A.A. and I.M.A.J.; writing—original draft preparation, R.A.A., I.M.A.J., L.F. and A.C.; writing—review and editing, R.A.A., I.M.A.J., L.F. and A.C.; visualization, R.A.A.; supervision, I.M.A.J.; project administration, I.M.A.J., L.F. and A.C.; funding acquisition, A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the European Union's NextGenerationEU instrument, under the Italian National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.3, enlarged partnership "Telecommunications of the Future" (PE0000001), program "RESTART".

Data Availability Statement: Publicly available datasets were analyzed in this study. The Boston, USA, air quality dataset, focusing on hyperlocal pollution data collected using mobile sensing platforms from February to April 2022, can be accessed at https://zenodo.org/records/7961851, accessed on 4 September 2024. The Rome taxi mobility dataset, consisting of GPS traces collected from approximately 320 taxis between 1 February 2014, and 2 March 2014, is available at https://ieee-dataport.org/open-access/crawdad-romataxi, accessed on 4 September 2024.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Al Jawarneh, I.M.; Foschini, L.; Bellavista, P. Efficient Integration of Heterogeneous Mobility-Pollution Big Data for Joint Analytics at Scale with QoS Guarantees. *Future Internet* **2023**, *15*, 263. [CrossRef]
2. Al Jawarneh, I.M.; Bellavista, P.; Corradi, A.; Foschini, L.; Montanari, R. Locality-Preserving Spatial Partitioning for Geo Big Data Analytics in Main Memory Frameworks. In Proceedings of the GLOBECOM 2020–2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6.
3. Hassan, A.; Vijayaraghavan, J. *Geospatial Data Science Quick Start Guide: Effective Techniques for Performing Smarter Geospatial Analysis Using Location Intelligence*; Packt Publishing Ltd.: Birmingham, UK, 2019; ISBN 978-1-78980-933-6.
4. Al Jawarneh, I.M.; Foschini, L.; Bellavista, P. Polygon Simplification for the Efficient Approximate Analytics of Georeferenced Big Data. *Sensors* **2023**, *23*, 8178. [CrossRef] [PubMed]
5. Al Jawarneh, I.M.; Bellavista, P.; Corradi, A.; Foschini, L.; Montanari, R. Big Spatial Data Management for the Internet of Things: A Survey. *J. Netw. Syst. Manag.* **2020**, *28*, 990–1035. [CrossRef]
6. Al Jawarneh, I.M.; Bellavista, P.; Corradi, A.; Foschini, L.; Montanari, R. QoS-Aware Approximate Query Processing for Smart Cities Spatial Data Streams. *Sensors* **2021**, *21*, 4160. [CrossRef] [PubMed]
7. Yu, J.; Tahir, A.; Sarwat, M. GeoSparkViz in Action: A Data System with Built-in Support for Geospatial Visualization. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, China, 8–11 April 2019; pp. 1992–1995.
8. Yu, J. SRC: Geospatial visual analytics belongs to database systems: The BABYLON approach. *SIGSPATIAL Spec.* **2018**, *9*, 2–3. [CrossRef]
9. Al Jawarneh, I.M.; Bellavista, P.; Corradi, A.; Foschini, L.; Montanari, R. SpatialSSJP: QoS-Aware Adaptive Approximate Stream-Static Spatial Join Processor. *IEEE Trans. Parallel Distrib. Syst.* **2024**, *35*, 73–88. [CrossRef]
10. Guo, M.; Huang, Y.; Guan, Q.; Xie, Z.; Wu, L. An efficient data organization and scheduling strategy for accelerating large vector data rendering. *Trans. GIS* **2017**, *21*, 1217–1236. [CrossRef]
11. Guo, M.; Guan, Q.; Xie, Z.; Wu, L.; Luo, X.; Huang, Y. A spatially adaptive decomposition approach for parallel vector data visualization of polylines and polygons. *Int. J. Geogr. Inf. Sci.* **2015**, *29*, 1419–1440. [CrossRef]
12. Rachev, S.T. The Monge–Kantorovich Mass Transference Problem and Its Stochastic Applications. *Theory Probab. Appl.* **1985**, *29*, 647–676. [CrossRef]
13. Al Jawarneh, I.M.; Bellavista, P.; Corradi, A.; Foschini, L.; Montanari, R. Spatially Representative Online Big Data Sampling for Smart Cities. In Proceedings of the 2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Pisa, Italy, 14–16 September 2020; pp. 1–6.

14. Ma, M.; Wu, Y.; Ouyang, X.; Chen, L.; Li, J.; Jing, N. HiVision: Rapid visualization of large-scale spatial vector data. *Comput. Geosci.* **2021**, *147*, 104665. [CrossRef]
15. Chen, H.; Chen, W.; Mei, H.; Liu, Z.; Zhou, K.; Chen, W.; Gu, W.; Ma, K.-L. Visual Abstraction and Exploration of Multi-class Scatterplots. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 1683–1692. [CrossRef] [PubMed]
16. Zheng, Y.; Wu, W.; Chen, Y.; Qu, H.; Ni, L.M. Visual Analytics in Urban Computing: An Overview. *IEEE Trans. Big Data* **2016**, *2*, 276–296. [CrossRef]
17. Chung, D.H.S.; Parry, M.L.; Griffiths, I.W.; Laramee, R.S.; Bown, R.; Legg, P.A.; Chen, M. Knowledge-Assisted Ranking: A Visual Analytic Application for Sports Event Data. *IEEE Comput. Graph. Appl.* **2016**, *36*, 72–82. [CrossRef]
18. Andrienko, N.; Andrienko, G.; Barrett, L.; Dostie, M.; Henzi, P. Space Transformation for Understanding Group Movement. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 2169–2178. [CrossRef]
19. Krüger, R.; Thom, D.; Wörner, M.; Bosch, H.; Ertl, T. TrajectoryLenses—A Set-based Filtering and Exploration Technique for Long-term Trajectory Data. *Comput. Graph. Forum* **2013**, *32*, 451–460. [CrossRef]
20. Liu, D.; Weng, D.; Li, Y.; Bao, J.; Zheng, Y.; Qu, H.; Wu, Y. SmartAdP: Visual Analytics of Large-scale Taxi Trajectories for Selecting Billboard Locations. *IEEE Trans. Vis. Comput. Graph.* **2017**, *23*, 1–10. [CrossRef]
21. Liu, H.; Gao, Y.; Lu, L.; Liu, S.; Qu, H.; Ni, L.M. Visual analysis of route diversity. In Proceedings of the 2011 IEEE Conference on Visual Analytics Science and Technology (VAST), Providence, RI, USA, 23–28 October 2011; pp. 171–180.
22. Liu, S.; Pu, J.; Luo, Q.; Qu, H.; Ni, L.M.; Krishnan, R. VAIT: A Visual Analytics System for Metropolitan Transportation. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1586–1596. [CrossRef]
23. Di Bartolomeo, M.; Hu, Y. There is More to Streamgraphs than Movies: Better Aesthetics via Ordering and Lassoing. *Comput. Graph. Forum* **2016**, *35*, 341–350. [CrossRef]
24. Dougenik, J.A.; Chrisman, N.R.; Niemeyer, D.R. An Algorithm to Construct Continuous Area Cartograms*. *Prof. Geogr.* **1985**, *37*, 75–81. [CrossRef]
25. Al-Dohuki, S.; Wu, Y.; Kamw, F.; Yang, J.; Li, X.; Zhao, Y.; Ye, X.; Chen, W.; Ma, C.; Wang, F. SemanticTraj: A New Approach to Interacting with Massive Taxi Trajectories. *IEEE Trans. Vis. Comput. Graph.* **2017**, *23*, 11–20. [CrossRef]
26. Huang, X.; Zhao, Y.; Yang, J.; Zhang, C.; Ma, C.; Ye, X. TrajGraph: A Graph-Based Visual Analytics Approach to Studying Urban Network Centralities Using Taxi Trajectory Data. *IEEE Trans. Vis. Comput. Graph.* **2016**, *22*, 160–169. [CrossRef] [PubMed]
27. Scheepens, R.; Hurter, C.; Van De Wetering, H.; Van Wijk, J.J. Visualization, Selection, and Analysis of Traffic Flows. *IEEE Trans. Vis. Comput. Graph.* **2016**, *22*, 379–388. [CrossRef] [PubMed]
28. Andrienko, G.L.; Andrienko, N.V. Interactive maps for visual data exploration. *Int. J. Geogr. Inf. Sci.* **1999**, *13*, 355–374. [CrossRef]
29. Liu, D.; Xu, P.; Ren, L. TPFlow: Progressive Partition and Multidimensional Pattern Extraction for Large-Scale Spatio-Temporal Data Analysis. *IEEE Trans. Vis. Comput. Graph.* **2019**, *25*, 1–11. [CrossRef]
30. Guo, J.; Wang, J.; Xu, C.; Song, Y. Modeling of spatial stratified heterogeneity. *GISci. Remote Sens.* **2022**, *59*, 1660–1677. [CrossRef]
31. Li, J.; Chen, S.; Andrienko, G.; Andrienko, N. Visual exploration of spatial and temporal variations of tweet topic popularity. In Proceedings of the EuroVis Workshop on Visual Analytics, Brno, Czech Republic, 4 June 2018; Eurographics Association: Goslar, Germany, 2018; pp. 7–11.
32. Andrienko, G.; Andrienko, N.; Chen, W.; Maciejewski, R.; Zhao, Y. Visual Analytics of Mobility and Transportation: State of the Art and Further Research Directions. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2232–2249. [CrossRef]
33. Sobral, T.; Galvão, T.; Borges, J. Visualization of Urban Mobility Data from Intelligent Transportation Systems. *Sensors* **2019**, *19*, 332. [CrossRef]
34. Lock, O.; Bednarz, T.; Pettit, C. The visual analytics of big, open public transport data—A framework and pipeline for monitoring system performance in Greater Sydney. *Big Earth Data* **2021**, *5*, 134–159. [CrossRef]
35. Kalamaras, I.; Zamichos, A.; Salamanis, A.; Drosou, A.; Kehagias, D.D.; Margaritis, G.; Papadopoulos, S.; Tzovaras, D. An Interactive Visual Analytics Platform for Smart Intelligent Transportation Systems Management. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 487–496. [CrossRef]
36. Silva, C.T.; Freire, J.; Miranda, F.; Lage, M.; Doraiswamy, H.; Hosseini, M.; Tokuda, E.; Ferreira, G.; Cesar, R.M., Jr. *Integrated Analytics and Visualization for Multi-Modality Transportation Data*; Connected Cities for Smart Mobility toward Accessible and Resilient Transportation Center (C2SMART): New York, NY, USA, 2019.
37. Gomes, G.A.M.; Santos, E.; Vidal, C.A.; Coelho da Silva, T.L.; Macedo, J.A.F. Real-time discovery of hot routes on trajectory data streams using interactive visualization based on GPU. *Comput. Graph.* **2018**, *76*, 129–141. [CrossRef]
38. He, J.; Chen, H.; Chen, Y.; Tang, X.; Zou, Y. Diverse Visualization Techniques and Methods of Moving-Object-Trajectory Data: A Review. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 63. [CrossRef]
39. Zhou, Z.; Yu, J.; Guo, Z.; Liu, Y. Visual exploration of urban functions via spatio-temporal taxi OD data. *J. Vis. Lang. Comput.* **2018**, *48*, 169–177. [CrossRef]
40. Clarinval, A.; Dumas, B. Intra-City Traffic Data Visualization: A Systematic Literature Review. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 6298–6315. [CrossRef]

41. Sinclair, C.; Das, S. Traffic Accidents Analytics in UK Urban Areas using k-means Clustering for Geospatial Mapping. In Proceedings of the 2021 International Conference on Sustainable Energy and Future Electric Transportation (SEFET), Hyderabad, India, 21–23 January 2021; pp. 1–7.
42. Медведенко, С.А.; Намиот, Д.Е. Визуальный анализ данных пассажиропотока железнодорожного транспорта. *Int. J. Open Inf. Technol.* **2021**, *9*, 51–60.
43. Xu, C.; Zhang, A.; Chen, Y. Traffic Congestion Forecasting in Shanghai Based on Multi-Period Hotspot Clustering. *IEEE Access* **2020**, *8*, 63255–63269. [CrossRef]
44. Ma, M.; Wu, Y.; Chen, L.; Li, J.; Jing, N. Interactive and Online Buffer-Overlay Analytics of Large-Scale Spatial Data. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 21. [CrossRef]
45. Al Jawarneh, I.M.; Bellavista, P.; Foschini, L.; Montanari, R. Spatial-Aware Approximate Big Data Stream Processing. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Big Island, HI, USA, 9–13 December 2019; pp. 1–6.
46. Lohr, S.L. *Sampling: Design and Analysis*, 2nd ed.; Chapman and Hall/CRC: New York, NY, USA, 2019; ISBN 978-0-429-29628-4.
47. Stoehr, N.; Meyer, J.; Markl, V.; Bai, Q.; Kim, T.; Chen, D.-Y.; Li, C. Heatflip: Temporal-Spatial Sampling for Progressive Heat Maps on Social Media Data. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 3723–3732.
48. Mitra, S.; Khandelwal, P.; Pallickara, S.; Pallickara, S.L. STASH: Fast Hierarchical Aggregation Queries for Effective Visual Spatiotemporal Explorations. In Proceedings of the 2019 IEEE International Conference on Cluster Computing (CLUSTER), Albuquerque, NM, USA, 23–26 September 2019; pp. 1–11.
49. Zacharatou, E.T.; Doraiswamy, H.; Ailamaki, A.; Silva, C.T.; Freire, J. GPU rasterization for real-time spatial aggregation over arbitrary polygons. *Proc. VLDB Endow.* **2017**, *11*, 352–365. [CrossRef]
50. Bruhwiler, K.; Buddhika, T.; Pallickara, S.; Pallickara, S.L. Iris: Amortized, Resource Efficient Visualizations of Voluminous Spatiotemporal Datasets. In Proceedings of the 2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT), Leicester, UK, 7–10 December 2020; pp. 47–56.
51. Bruhwiler, K.; Pallickara, S. Aperture: Fast Visualizations Over Spatiotemporal Datasets. In Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing, Auckland, New Zealand, 2–5 December 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 31–40.
52. Alsalama, A.; Kubba, A.; Alsmirat, M.; Al Jawarneh, I.M. A Novel Approximate Computing Method for Efficient Search in Satellite Remote Sensing Products. In Proceedings of the 2024 International Conference on Multimedia Computing, Networking and Applications (MCNA), Valencia, Spain, 17–20 September 2024; pp. 21–27.
53. Al Jawarneh, I.M.; Montanari, R.; Corradi, A. Cost-Effective Approximate Aggregation Queries on Geospatial Big Data. In Proceedings of the 2023 IEEE Globecom Workshops (GC Wkshps), Kuala Lumpur, Malaysia, 4–8 December 2023; pp. 1313–1318.
54. Huang, K.; Liu, D.; Chen, T.; Wang, Y.; Wang, C.; Shi, W. Real-time map rendering and interaction: A stylized hierarchical symbol model. *Int. J. Digit. Earth* **2024**, *17*, 2367728. [CrossRef]
55. Huang, K.; Wang, C.; Wang, S.; Liu, R.; Chen, G.; Li, X. An Efficient, Platform-Independent Map Rendering Framework for Mobile Augmented Reality. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 593. [CrossRef]
56. Aljawarneh, I.M.; Bellavista, P.; Corradi, A.; Montanari, R.; Foschini, L.; Zanotti, A. Efficient spark-based framework for big geospatial data query processing and analysis. In Proceedings of the 2017 IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece, 3–6 July 2017; pp. 851–856.
57. Nusrat, S.; Kobourov, S. The State of the Art in Cartograms. *Comput. Graph. Forum* **2016**, *35*, 619–642. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.