Hall of Mirrors: A novel strategy to address locality in geocoded-based Pol private queries

PEDRO WIGHTMAN¹, (Senior Member, IEEE), MAYRA ZURBARAN², AUGUSTO SALAZAR³ AND LORENA GARCIA.⁴(Senior Member, IEEE),
¹Schools of Engineering, Science and Technology, Universidad del Rosario, Bogota D.C. 111711 Colombia (e-mail: pedro.wightman@urosario.edu.co)
²Seidor, Ispra, VA 21027 Italy (e-mail: mayra.zurbaran@seidor.it)
³Department of Systems Engineering, Universidad del Norte, Barranquilla, ATL 081007 Colombia (e-mail: augustosalazar@uninorte.edu.co)
⁴School of Engineering and Basic Sciences, Universidad Central, Bogota D.C., 110311 Colombia (e-mail: lgarciap9@ucentral.edu.co)
Corresponding author: Pedro Wightman (e-mail: pedro.wightman@urosario.edu.co).

ABSTRACT Location privacy techniques try to protect user’s information by altering, aggregating or generalizing it. Geographical codification techniques, like Geohash, can be used to protect individual locations altering the precision of the location so it represents wide area that contains the user’s location but does not give out the exact coordinate. However, this transformation creates some problems when a simple range-based query wants to be performed over coded data: neighbor points may have quite different representations if they fall in different quadrants. This problem, named lack of locality, has been solved by extending the search area of the query by projecting the user’s location in all adjacent grid cells and use the common prefix of the code to identify all the points in the adjacent cells, but the result set increases substantially, creating a problem for the user which needs to filter the useful results from the set returned by the extended query. In this work, the Hall of Mirrors strategy is presented, which creates multiple representations of the points of interest in adjacent quadrants. This allows, not only the execution of the traditional common prefix query, but also distance-based queries from the user’s location, using the numerical code difference, which overcomes the locality problem by obtaining the relevant points of interest - PoI and reduces the number of total results. Four PoI projection techniques are introduced and compared to the AHG technique and the regular geographic query. The results of the experiments performed on a dataset of 827 points of interests in Bogota, Colombia, show that, compared to regular common prefix queries, distance-based HoM generates from 29% up to 91% fewer irrelevant results in the best scenarios. In addition, results show that HoM techniques can find the relevant points faster than the AHG technique, due to the nature of the points projection and better distance correspondence.

INDEX TERMS Geohash, Location based information services, Location coding, Location privacy

I. INTRODUCTION

FOR more than two decades, Location-Based Services, or LBSs, have been gaining traction with users due to the accessibility of GPS-enabled devices, initially in vehicle navigation, but now increasingly integrated to smartphone and IoT device applications.

Service providers now include user’s location as part of the service context, in order to increase the quality and pertinence of their results to their customers. These locations are stored and used to build user profiles, which later can be sold to third-party companies, usually in an aggregated manner to protect the user’s data. However, the exact location of the users remains in the servers and holds information that can be used to calculate the users’ routines and, if used negligently could harm the user’s privacy.

Location privacy has been an active research area, especially since the proliferation of LBSs in the 2000s, in order to protect the location information of the user from undesired and unauthorized access. Location privacy comprises, as defined in [1], "the right of the individuals to decide how, when and under which motives their location information can be shared". Many different techniques can be used for this [2]. One of the techniques in the literature for providing privacy is the use of geographical coding to rewrite the location...
coordinate into a different representation, that should allow querying, neighbor search, clustering and all the operations that the service requires. Location privacy comes in place when the coding strategy allows changing the precision of the representation, generalizing the original location into a coordinate that, based on the scale used, represents the area of a grid cell: the lower the precision, the wider the area of uncertainty around the original location. This allows the execution of secure queries, where users, that want to obtain a list of nearby points of interest (restaurants, drugstores, schools, etc.) from their location, can control the level of location uncertainty that they want to share with the service provider when performing a distance-based query.

The main problem of using the coding techniques to perform range queries is locality: the ability to preserve the sense of vicinity among coded or protected locations. Depending on the algorithm, location codes of nearby points which are in different grid cells, can be very different from each other, which makes it difficult to group the results in a simple query, because the most common way to perform a range query is by using the common prefix idea. This technique consists on comparing the first portion of the code of the points, which if it is the same, implies that the points are contained in the same grid cell of a higher level, depending on the number of digits involved. For example, coded locations d2g66qrpqf and d2g6d8knb share the first 4 digits on their representation, d2g6, thus they can be grouped up to scale 4 in the same grid cell; however, their corresponding level 5 grid cells and all lower level ones, are different. The problem of locality appears when even nearby locations have very distinct codes that do not share a common prefix, like points near the border of a grid cell, and the system is not be able to identify them as neighbors. Most existing techniques that try to solve this problem, usually generate query result sets much larger than the regular geographical queries. This bloated result set creates a problem in terms of computing and communication overhead for the user’s end device, which will receive the query results and then has to filter the relevant points of interest, or PoIs, based on its real location. This extra processing is an undesired consequence and must be reduced as much as possible.

This work introduces a new approach to reduce the problem of locality, called Hall of Mirrors, or HoM. Instead of projecting the user’s location into the adjacent grid cells and querying the complete extended area, this technique proposes a novel approach: projecting each PoI in the adjacent grid cells, so that the projected points will have different code prefixes, making them reachable locally in several grid cells when queried, and, thus, accessible to the user’s coded location. This characteristic allows, not only the implementation of the common prefix query based only on the user’s location, but also the implementation of a distance-based query, using the difference between the numerical code representation of the PoIs and the user’s, as an estimation of the distance; this is, the absolute value of the subtraction of the codes.

The main contribution of the paper is the introduction of a new way to solve the problem of locality, based on PoI projection instead of user projection. The work includes the definition and evaluation of four different projection techniques, which show different benefits from each approach. In addition, this paper explores for the first time, as far as explored in the related work, the use of the difference between numerical representation of the Geohash coded location as an estimation for the distance between points which, when used with PoI projection, produces a smaller query result set, compared to the common prefix-based queries and the AHG technique.

Experiments show that the four HoM techniques are able to gather all the relevant points with a lower distance parameter, and tend to reduce the size of the total query results, compared to the Adaptive Hilbert-Geohash, or AHG [3] algorithm, in both city centric and non-centric scenarios, where the PoI density changes drastically. In addition, the query time is similar between all the HoM and AHG technique and an improvement compared to the standard geographical query, which means that the total number of records, which is increased by the inclusion of the projected points, does not have a great impact on the performance of the queries.

This work is organized as follows: Section 2 is dedicated to present the state of the art in the area la geocoding related to privacy and the solution of the lack of locality problem. Section 3 introduced the Hall of Mirror strategy, and all the 4 projection techniques. Section 4 presents the experimental design and the results of the performance evaluation of the techniques. Section 5 presents a discussion and analysis of the results in a more general manner; and finally, Section 6 presents the conclusions and future work for this work.

II. RELATED WORK

Location privacy has been examined for decades, but it has gained greater importance in the later years due to the amount of location information that is being collected from the users; however, there is a constant fear that this data may have not been treated well enough to protect the user’s privacy when acquired, stored analyzed or shared with third-party companies for profit.

Many techniques has been proposed in the literature [2]: obfuscation using random-noise generation [4]–[8], transformation via matrix multiplication [9], private information retrieval [10], [11], aggregation [12]–[16], k-anonymity [17], cloaking [18], among others. In this last one, cloaking the user’s location consists on generalizing the coordinates in such a way that just an geographical area that contains the location is revealed, hiding the exact location. The size of the cloak can change, depending on the level of desired protection. One way to provide this kind of protection is by using geographical coding.

Geographical coding can be defined as a function to transform a bi-dimensional coordinate into a uni-dimensional value that represents the location and that can later be operated to perform actions like calculating distance, clustering, geofencing, etc. Geocoding has been studied in a large...
number of previous works, with many applications, like data organization in large databases [19], data aggregation [20], and location privacy [16], [21].

There are several well-known algorithms for geographical coding like GeoHash [22] developed by Gustavo Niemeyer, which has several variants, like Adaptive Hilbert-Geohash or AHG [3], which both will be explained in more detail later on. Other techniques are GeoSOT [23], in which a quadrant-based division of the space is proposed, the work in [24] which improves GeoHash by allowing clustering by using B-trees, STGI [25] which is optimized for marine coordinates, Google S2 Geometry coding [26], the work in [24] which is based on Google S2 encoding to include three dimensional coordinates, and the work in [27] where the authors use diamonds as tessellation geometry for the Earth.

Coding techniques, like GeoHash and AHG, allow the representation of individual locations, with different level of precision, reducing the dimensionality of the data from 2 to 1 components, which has benefits in terms of query time and storage [28]. Both techniques use a bisection-based technique, in which the final code is the integration of the binary representation of the coordinates, based on divisors of 90° and 45° for longitude and latitude, respectively, as it can be seen in Fig. II. The code is built by combining the latitude and longitude binary digits in pairs, at each level of precision. Based on the bit representation, a Base32 code is generated, which keeps adding less significant digits to the code, grouping them in 5 binary digits. GeoHash is a widely used geocoding technique, with applications in several areas, like in navigation, big data acceleration, among other application seen in [29], [30], [31], [32], [33], [34], [28], [35], [36].

GeoHash and AHG perform area-based queries by selecting all the grid cells that fall in the search area and using the common prefix of the point’s representation to define the search criteria among points in the same grid cell, ignoring the least significant portion of the code. Now, the locality of the grid cells in GeoHash is not ideal, because adjacent cells can have very different prefixes, since these techniques work mainly in groups of four cells, and then

Example of how Geohash works

jumps geographically following a N or Z-shaped path along the grid cells, or a Hilbert curve sequence, like in AHG [3]. This difference, based on the coding perspective, will show a great distance between PoIs in adjacent cells. For example, as it can be seen in Figure II cells in gray are adjacent in terms of code when following the pattern; however, the cells in white that are directly adjacent, will probably have a very different code, creating a problem when searching for neighbors, because the code does not accurately represents geographical vicinity; thus, a simple code-based query will dismiss geographically adjacent points with large difference in their coded representation, as mentioned in [37].

To solve this problem, different solutions have been proposed: the authors in [38] proposed the use of randomly generated areas to group geographically adjacent points in searches, defining a mathematical function to verify if points belong in that area. In [39], authors explore in detail how the GeoHash bit-level representation can be used to determine adjacent grid cells. In [40], the authors propose a simple approach in which the server decodes the location and uses regular geographical queries, implying extra processing time, both for the decoding and the query time which is longer for traditional geographical queries compared to coded search, which is usually text or number-based. In [41] the authors propose the use of a common prefix among many points but the use of a prefix mask to allow distance calculation between two coded points. In [42], the authors identify adjacent cells for range queries with irregular shapes that overlap with other cells. In [43], the authors propose the use of pairs of bits, instead of the combination of 5 digits in base32 in order to simplify the identification of neighbor cells and improve accuracy when generating the locations. In [44], authors propose the use of Bloom filters in order to faster identify the points inside a grid cell, based on common prefix.

However, the most common way of solving the problem [27], [44]–[47] is to trick the system by querying all adjacent grid cells to the user’s location, so it will include grid cells even with large code difference. In AHG [3], the authors propose the use of Hilbert’s curves in order to improve locality, and, given that adjacent cells will still have different common prefix, they project the user’s location in each adjacent cells, using the relative distance between the point and its centroid on the original cell; this is, applying the vertical and horizontal distance from the original centroid to the new ones, as it can be seen in Figure II. The set of 8 extra
In this paper, a new technique, called Hall of Mirrors, is proposed to overcome the problem of locality in queries over coded locations, based on using multiple mirrored representations of PoIs on adjacent grid cells at different precision levels.

In the HoM technique, instead of changing the user’s location, like in AHG, the locations of the PoIs are the ones projected into adjacent cells, and stored in the database for future searches. In this case, a single query from the user’s coded location will be enough to find the nearby PoIs, both original and projected, because they will share the common prefix of the original user’s grid cell.

In addition, making the PoIs reachable inside the user’s grid cell, makes it possible to calculate the difference between the code numerical representations of points, which will only differ at the less significant portion of the code, and emulate, like this, a distance indicator. This difference can be used to perform more precise queries than just using common prefix, because the relevant points may not be in the complete area of the grid cells but only should include a partial area of the cells, determined by a distance from the user’s location. This delimitation should make the query return less irrelevant results. The main issue to be defined, in order to perform this technique, is how the PoIs will be projected in the adjacent grid cell, in order to improve its reachability and still provide a good estimation of location and distance. In this section, four different projection techniques ways are explored.

A. AHG-BASED MIRRORING

For the sake of comparison, the AHG projection technique will be used to reinterpret the PoIs in the adjacent cells: the relative position of the PoI, compared to its main centroid, will be reproduced in all surrounding cell’s centroids, as shown in 1. The projected point will reproduce the vertical and horizontal distance between the PoI’s location and the original centroid of the grid cell, and apply it to the centroid of the adjacent grid cell. However, it is expected that the projected points may show problems related to losing the level of closeness, as it was mentioned before. In algorithm III-A, it is shown how the AHG-based mirroring technique works:

AHG-based HoM projection algorithm

\[
\text{for eachGeoHashLevel in (5, 6) do}\ \\
\text{for eachPinPoIDatabase do}\ \\
\text{for eachDirection in (N, S, E, W, SE, SW, NE, NW) do}\ \\
\begin{align*}
\text{mainCentroid} & \leftarrow \text{calculateCentroid}(p) \\
\text{nextCentroid} & \leftarrow \text{calculateCentroid}(p, \text{direction}) \\
\text{diff} & \leftarrow \text{calculateDifference}(p, \text{mainCentroid}) \\
\text{newDiff} & \leftarrow \text{addDifference}(p, \text{diff}) \\
\text{PoIDatabase.add(newP, GeoHashLevel)}
\end{align*}
\]

This delimitation should make the query return less irrelevant results. The main issue to be defined, in order to perform this technique, is how the PoIs will be projected in the adjacent grid cell, in order to improve its reachability and still provide a good estimation of location and distance. In this section, four different projection techniques ways are explored.
In order to overcome this problem, three solutions are proposed in this work: centroid mirroring, border mirroring and scale mirroring.

### B. CENTROID MIRRORING

The Centroid Mirroring uses the AHG-based reproduction of the distance between the PoI’s location and its original centroid, but changes the orientation of the PoI’s relative location to its centroid, based on the position of the adjacent cell: if the point is adjacent to the top cell (this is, there is positive difference), then the latitude of the projected point will have a negative difference compared to its centroid, changing the orientation of the latitude. This is an improvement over AHG because the change in orientation makes the projected point more likely to appear closer to the original one and it will increase the possibility of showing up in distance-based query, as shown in Figure 2.

In algorithm III-B, it is shown how the Centroid-based mirroring technique works:

```plaintext
for eachGeoHashLevel in (5, 6) do
  for eachPinPoIDatabase do
    for eachDirection in (N, S, E, W, SE, SW, NE, NW) do
      mainCentroid ← calculateCentroid(p)
      nextCentroid ← calculateCentroid(p, direction)
      diff ← calculateDifference(p, mainCentroid)
      newP ← addDifference(p, diff, direction)
      PoIDatabase.add(newP, GeoHashLevel)
    end for
  end for
end for
```

The `calculateCentroid(p)` function determines the closest point to the grid cell determined by the PoI’s location.

### C. BORDER MIRRORING

In the Border Mirroring, the projection is calculated based on the distance of the PoI to the border of the adjacent cell. A contact point to the border will be selected: in vertical and horizontal cells, it will be the point located in the shortest straight line to the border. The length of this line will be applied to the projected point in the adjacent cell away from the border, also in a straight line. In the case of the adjacent diagonal cells, the contact point is the corner point and the projection point will apply the vertical and horizontal distances to this location, as shown in Figure 3.

In algorithm III-C, it is shown how the Border-based mirroring technique works:

Border HoM projection algorithm

```plaintext
for eachGeoHashLevel in (5, 6) do
  for eachPinPoIDatabase do
    for eachDirection in (N, S, E, W, SE, SW, NE, NW) do
      contactP ← calculateContactPoint(p, direction)
      diff ← calculateDifference(p, contactPoint)
      newP ← addDifference(contactP, diff, direction)
      PoIDatabase.add(newP, GeoHashLevel)
    end for
  end for
end for
```

The `calculateContactPoint(p, direction)` function determines the closest point to the grid cell determined by the PoI’s location.
direction. After the difference between the original point and the contact point is determined, it is applied to the contact point in order to reproduce the location in the neighbor grid cell.

D. SCALE MIRRORING

In the Scale Mirroring, the projection is calculated based on the distance of the PoI to the centroid at the adjacent cell. This approach looks for a projection that lies in the general direction to the adjacent cell from where the PoI is located.

In algorithm III-D, it is shown how the Scale-based mirroring technique works:

Scale HoM projection algorithm

\[
\text{for each } \text{pinPoIDatabase do} \\
\quad \text{for each } \text{direction in } (N, S, E, W, SE, SW, NE, NW) \text{ do} \\
\quad \quad \text{nextCentroid } \leftarrow \text{calculateCentroid}(p, \text{direction}) \\
\quad \quad \text{diff } \leftarrow \text{calculateDifference}(p, \text{nextCentroid}) \\
\quad \quad \text{newP } \leftarrow \text{addDifference}(p, \text{diff } \times 0.7, \text{direction}) \\
\quad \text{PoIDatabase.add(newP, GeoHashLevel)} \\
\quad \text{end for} \\
\text{end for} \\
\text{end for} \quad =0
\]

The \text{addDifference}(p, \text{diff } \times 0.7, \text{direction}) function determines the distance between the original point and the centroid in the neighbor grid cell, and the projects the point 70\% of the distance, in order to guarantee that the point falls inside the grid cell. As shown in Figure 4. The projections follow a star-like geometry that limits the positions in which the projections can be, which could help to group the external points and facilitate its reachability on distance-based queries.

On these last three mirrored projection techniques of HoM, it can be seen that, when a PoI is close to a border, the projection retains that characteristic, making it more susceptible to be included in distance-based queries from the immediate adjacent cell. In the case of non adjacent cells, the projected points either tend to get closer to the real location (Centroid- and Scale-based) or tend to go very far from the actual location (Border-based). However, given that these projections may be too far from the actual PoI, they may not be relevant enough for the user’s context and should not be eligible to appear in a distance-based query, in contrast with the the projections in nearby cells. This behavior could be used in order to privilege short range queries, allowing the reduction of projections, but this will be explored in a future work.

Projections depend on the level of precision of the grid cell, thus each PoI should be projected in more than one level; for example, a GeoHash code with 5 and 6 digits of precision roughly approximate areas of 2.5x2.5 km and 0.6x0.6 km grid cell sizes, which, including the adjacent cells, could represent and large portion of the city area (6.25 km\(^2\)) and a neighborhood area (0.36 km\(^2\)). If only these two levels are used, each PoI would need 17 representations: the original location and 16 projections, 8 per level of precision.

One example of this distribution is shown in Figure 5, where a PoI in Bogota, Colombia, is projected on its immediate adjacent cells using the Centroid Mirroring technique. The coordinated of the projections can be seen in Table 1. The GeoHash column represents the numerical value of the code, given that it is originally coded in Base 32, which is not recognized or operable in traditional databases. In order to calculate this value, it is important to use the same length for the code (even if filling with zeros at the end) to guarantee that the common prefixes are in the same position; otherwise, there would be two very different numerical representations, even if the GeoHash values do have a common
prefix: d2g66s would be 405,248,216, and d2g66sy would be 1,2967,942,960, respectively. This would become a limitation for the code to be used for distance-based queries if a similar code length is not defined. In order to standardize the results in this work, all geohash codes will be stored and interpreted with a 9 digit precision, using the substring function in order to compare common prefixes.

Finally, an empirical comparison of the AHG and the Hall of Mirrors technique, based on a theoretical query scenario, shows a user performing a distance-based query where there are a number of PoI's in its own cell and in adjacent cells with no common code prefix.

Based on these results in Table 2, it can be seen how in the full grid cell search, AHG returns a larger set of PoIs that the geographical query or the HoM technique, including PoIs that are not relevant to the user. Now, in the distance-based query it can be seen that AHG ignores PoIs 1 and 9 which are the closest PoI's to the user. In contrast, the result set of HoM, despite of the extra results compared to the geometrical query, contains the nearby points, which confirms the quality of the results.

IV. PERFORMANCE EVALUATION

A. DATASET

The dataset used for the experiment is a list of points of interest from Bogota, Colombia, with 827 locations and 148 different categories, obtained from Foursquare. The top categories of the subset are presented in Table 3.

The graphical representation of the location of the PoIs and their projections using HoM-Border in the area of Bogota can be seen in Figure 7. In this work, just the location information...
was used. Other criteria, like name of category could be used in a future work to define more specific queries or limit the amount of points in the query definition. The database has 106,683 records, which include all the original points and their 8 projected points, at 4 different scales, for each HoM projection mechanism.

### B. SELECTION OF TECHNIQUES

The set of experiments will compare five different coded search: AHG, HoM-AHG, HoM-Centroid, HoM-Border and HoM-Scale Mirroring, along with the standard geographical range query function from PostGIS. These techniques were selected in order to compare the new strategy, of projecting the PoIs, with the closest one of the existing alternatives, projecting the user’s location, and also in order to determine the best projection techniques among the proposed ones.

The geographical query will obtain all PoIs within a certain distance, depending on the scale. Two distances will be used, associated to the Geohash grid cell size at scale 5 and 6: 0.02197265625 and 0.0054931640625 degrees, which represent a circular area of approximately 2.5 km and 600 mts, respectively.

Five locations will be used for simulating the execution of a query by a user looking for all the PoIs at a certain distance from its locations. The location are shown in Table 4. These locations were selected in order to have three locations in centric areas of the city with a PoI high density, and two less centric areas of the city, with low density of points of interest, in order to compare the behavior of the projections in terms of inclusion and reachability of the points in adjacent grid cells. In the table it can be see how the first three scenarios show a large number of query results from the geographical distance query, compared to the last two, mainly at scale 5.

The AHG technique will be applied in the following way: a single query will search in 9 cells, the user’s and the 8 surrounding ones, based on the projected locations of the original coordinate. This technique will only include the original PoIs, given that it does not depend on projection of the points, at a given scale. The projected user’s locations used in the experiments are defined in Tables 5 and 6.

The HoM-based techniques will only use the coded location of the user, for distance calculation and common prefix experiments, and will look over the original and projected PoIs at the given scale, and for the particular projection technique.

### TABLE 4. User locations for the experiment

| Description          | Longitude | Latitude  | Geo-5 | Geo-6 |
|----------------------|-----------|-----------|-------|-------|
| Chapinero Alto       | -74.401467| 4.655659 | 88    | 9     |
| 93st Park            | -74.052714| 4.6776598| 97    | 8     |
| 72st-7th Ave.        | -74.0641296| 4.6591177| 54    | 3     |
| Bus Station 80st     | -74.1073024| 4.707942 | 5     | 0     |
| Plaza Americas       | -74.137144| 4.617481 | 6     | 2     |

### C. QUERY DEFINITION

Two types of queries will be used for testing: Geohash common prefix, based on the Base32 codification, and Geohash distance, based on the numeric value of the code. The first type will include all points with the same common prefix of the Base32 code of the user’s location or, in the case of the AHG, all the projected locations of the user. The query will return all the points that comply, both originals and projections, when they apply. This experiments serves as a baseline to measure the impact of distance-based query, mainly in terms of total query results, compared to the current solution to the locality problem.

The distance-based query will look for the closest PoIs, original or projected, to the user’s location, based on the difference between the numeric representations of the coded locations. Given that there is not a proper geographical translation of the numeric representation to distance, due to the intertwined nature of the GeoHash code, it was necessary to test several distances and characterize for each technique and scale configurations: 1,000, 3,000, 6,000, 10,000, 30,000, 60,000, 100,000, 300,000, 600,000, 1,000,000, 3,000,000, and 6,000,000, in both scales. However, the figures will only show the distances up to the point in which all techniques have found the relevant points of interest; this is, the ones that appeared in the original range-based query and are contained in the query results including the projections.

### D. PERFORMANCE INDICATORS

Three performance indicators will be evaluated in the experiments: i) Relevant query results, which shows how many of the points of the geographical query are in the complete set of results, depending on the distance; ii) Average point distance, which shows the average distance of the PoIs in the result set from the user’s location when the distance increases, i.e., the average distance between all pairs of nearest neighbor points.
TABLE 7. Query results of geographical and common prefix

| Scenario          | Geo-5 | CP-5 | Geo-6 | CP-6 |
|-------------------|-------|------|-------|------|
| Chapinero Alto    | 88    | 621  | 9     | 79   |
| 93st Park         | 97    | 673  | 8     | 83   |
| 72st-7th Ave.     | 54    | 673  | 3     | 87   |
| Bus Station 80st  | 5     | 326  | 0     | 4    |
| Plaza Americas    | 6     | 157  | 2     | 8    |

compared to the distance of the original range query; and iii) Total query results, which shows how many unique PoIs are returned by the query. The first two are relevant in terms of quality of the results obtained by technique, and the last one is related to the overhead of the techniques.

E. RESULTS - SCALE 5

1) Common prefix query

In this experiment, the query will bring the PoIs with a similar Geohash prefix; this is, with the same first $N$ digits, depending on the search scale. As expected, all the experiments return the same number of results, showing that all the points from the adjacent grid cell will be included, either by querying the cells directly, as in AHG, or by obtaining the projected points from those grids at the user’s cell, using their projected code.

As it can be seen in Figure 7, compared to the geographical query, all the other techniques generate a much larger set of PoIs, around 10 times the number of points in dense scenarios, and between 30 and 60 times in sparse locations. This happens because of the larger search area of the common prefix query, which, especially in scale 5, can become very critical in locations at centric areas in the city, where a large percentage of eligible PoIs will be included. The need for a distance-based query becomes critical to provide more relevant results to the user, so there are less points to filter out in the presentation layer of the user’s search application.

2) Distance-based query

In this set of experiments, the query requests the PoIs that are in certain range from the user’s location. Given that coded coordinates are used, the relative distance is estimated by subtracting the numerical codifications of the user’s and the PoIs’ locations and using that difference as an estimation of the distance. The main goal of these experiments is to evaluate if the distance-based query can be implemented over the coded locations, and to evaluate if there is a reduction of the size of the final result set. For the sake of the experiments, the range being used on the geographical queries is the equivalent to the side of a grid cell. Given that, as mentioned before, Geohash does not specify a proper scale to translate the numeric representation to a distance, the code distance will increase until the result set includes all the points in the geographical query results. A more general evaluation of queries, with arbitrary areas not only within the range determined by the scale, will be studied in a future work.

The first element to evaluate is the quality of the results; this is, if the set of results from the standard geographical query are contained in the query results. This is critical because it will prove if the approximation techniques can overcome the problem of locality by including neighbor points over the borders of the grid cells, finding those PoIs in a searchable distance from the the user’s location. In addition to the relevant results, the size of the total query results is also shown, in order to illustrate the overhead generated by the approximation techniques, in terms of all the PoIs that appear in the query, as the distance from the user’s location grows. Figures 8, 9, 11, 10, 12 show the results for scale 5.

At scale 5, the search area extends for 2.5 km from the user-s location, but with the extended, area of search, the search area extends to 7.5 km, in order to include the adjacent cells. The first three locations are very centric, so a large portion of the PoIs fall in the range. For example, at the 1,000,000 code distance parameter, in Chapinero Alto, the number of results from the HoM techniques is 621 results compared to 706 of the AHG. The number of results from common prefix at this scenario was 621, so at this distance there is no improvement compared to the common prefix.
However, note that the HoM techniques were able to find all the relevant points at a code distance of 600,000 and all the new mirroring techniques were able to find them at 300,000, with results between 431 and 487, and between 541 and 594, respectively to the distances. This means a reduction between 30% and 29% in the number of results, at the 300,000 distance evaluation, for the HoM mirroring techniques. A similar behavior, where all the HoM mirroring techniques were able to retrieve the relevant PoIs long before AHG or HoM-AHG happened at the Business District 72nd St. scenario, where at a distance of 300,000, all 54 were found with only 57 to 98 results, compared to 673 from the common prefix query. This means a reduction of between 91% and 85%.

Now, the Park 93rd scenario showed a very similar behavior among all techniques, where all 5 were only able to find all the relevant points at the 600,000 code distance value, with 676 to 688 results, compared to 673 from the common prefix. In the general progression of relevant point retrieval, HoM-mirroring techniques were reaching more points, but the results suggest that some of the points were at the full range distance from the user’s location, thus the complete 600,000 code distance was required to find them.

Now, in less centric points, like the Bus Portal at 80Th St. and the Plaza Americas, the behavior is similar. At the Bus Portal scenario, with 100,000 code distance parameter, two of the HoM had already found the relevant points, with between 145 and 152 points, compared to the 326 of the common prefix. At 300,000, all 3 HoM mirroring techniques have found the relevant points, with result sets between 194 and 308, while AHG had 200 results but was not able to find all the relevant points yet. In the Plaza America scenario, at 300,000 code distance parameter, all three HoM mirroring techniques had found the relevant points with results sets between 58 and 98, compared to the 157 of the common prefix experiment. At this points AHG has not found all the relevant points, despite the 66 results points on its result set. At 600,000, when AHG finally get all the points, the number of query results of the HoM techniques, increases drastically, but all the PoIs had already been discovered.

Another indicator of quality is how the average distance of the points included in the query results behaves when the distance parameter increases. Ideally, the average distance of the points in the result should not go over the range distance parameter of the standard geographical query; however, as the search area expands, points that are farther away of the user will be included in the query results, increasing this average. To calculate this indicator, the average is calculated using the original location of the points included in the query, and not the one of the projections. This behavior, for scale 5, can be seen in Figures 13, 14, 16, 15, 17.
FIGURE 16. Average distance of point results to user's location in distance query - Bus portal 80th St., scale 5

FIGURE 17. Average distance of point results to user's location in distance query - Plaza Americas, scale 5

It can be seen how in most scenarios, by the 30,000 code distance value, the average distance is still under the range query distance; however, just at most 50% of the relevant points have been found in all experiments. In the Chapinero Alto scenario, at 300,000 code distance, where the HoM mirroring techniques already had retrieved the relevant points, there was an increase of 35% of the average, equivalent to an average distance of 3.3km, compared to the 2.5 km of the range query. At this same point, AHG already has an average distance of almost 3.6 km, which is an increase of 45%, and has not been able to get all relevant points. This can be explained by the fact that AHG starts querying far from the borders, where most near-border relevant points will be located, while the HoM techniques projects the points near the border, making them reachable from the user’s location, in an easier manner, with less distance parameter. This behavior repeats in the other two dense point scenarios.

In the case of the less centric scenarios, the average distance of the points increases faster and reaches around 6.7km in the two scenarios. This can be explained by the fact that most points are in nearby grid cells and not on the user’s. In the Bus Portal scenario, HoM-border and Scale showed the lowest average up to 300,000, where they joined the other techniques at around 5.6k of average distance, already having found all relevant points.

Now, in most scenarios centric and not centric at scale 5, HoM-border tends to show a lower average distance value, and lower number of total results, which makes it a good candidate for a general good performance. This can be explained by the fact that that points of interest are projected very close to the border where they are located, and very far from the border on the cells that are not adjacent from them. This makes that non-relevant points are less likely to appear in a distance-based query because they are projected far from the original PoI in cells that are not likely to perform a near-border distance query. This can be seen in the Business District scenario, Figure 15, where the average distance starts very high, but converges rapidly to lower values. Probably, there was a point projected near the user that was originally located very far away from the query location. This idea of prioritizing the near-border projections could be used to reduce the projections and reduce the final result set. This idea will be explored in a future work.

F. RESULTS - SCALE 6

When working at scale 6, results in Figures 18, 19, 20, 22 show a similar behavior of all techniques, in terms of their ability to retrieve relevant points, and having found them all at a code distance of 30,000. This shows how there is no linearity in terms of coded distance, given that the search range in scale 6 is a quarter of the one in scale 5, but the code difference is just 5% of the 600,000 where most scenarios were covered at scale 5.

Now, in centric scenarios, the total number of results is always larger for the AHG technique, always over a 100 results, compared to values between 79 and 87. The HoM techniques got better results in all scenarios, between 93 and 102 in the Chapinero Alto, compared to 79 in the common prefix, between 87 and 96 in the Park 93rd St. compared to 83, and 92 and 110 in the Business District area, compared to 87 in the common prefix. In this experiments, the common prefix had better results in terms of result set because the coded distance parameter included points whose code may have not shared the same common prefix at a lower significance, but the difference between them was not large enough. At this scale, code distance can be more sensitive to changes and a smaller granularity could have been used.

In the non centric scenarios, each scenario deserves its own analysis. The Bus Portal scenario did not have any PoI in the geographical query; however, all techniques tried to look for points in the nearby cells. Results show that the common prefix query only found 4 points, while the distance queries found between 5 and 7, with HoM-Centroid and HoM-Scale showing the best results. In the Plaza Americas, common prefix found 8 results, while distance queries obtained between 9 and 10 results, in order to find 2 relevant points. Again, HoM-Centroid had the best performance.

In terms of average distance, results can be seen in Figures 23, 24, 26, 25, 27. In centric scenarios, With the exception of Chapinero alto, at 30,000 code distance, the average distance tends to converge for all methods, but way above the expected distance: between 0.008 and 0.01 compared to 0.0054. In the Chapinero Alto scenario, the AHG technique had worst behavior than the HoM techniques. It is important to note that in all centric scenarios, the average distance always starts at the expected distance or above, which means that grid cells at this scale tend to be very small and have very few points,
such that points in nearby cells are distant from the user in comparison with the grid size, which increases the average immediately.

In non-centric scenarios, at 30,000, the average distance of the points from the AHG technique is larger than the HoM techniques, which means that, despite the fact that all the techniques have found the relevant points at that distance, the AHG technique is including more distant points than the HoM techniques and that the points retrieved by the HoM techniques were more relevant to the user’s location than the ones obtained by AHG.
V. DISCUSSION AND ANALYSIS

Further than the quantitative results from the experiments that show the benefits of the HoM techniques, it is very important to note that, in general, this work shows that range-based queries can be performed over coded data.

For example, at scale 5, a distance between 600,000 to 1,000,000 is a conservative estimation of a search area equivalent to 2.5 km, and, with a high probability, the HoM mirroring techniques could even use a distance of 300,000 and still cover a substantial set of relevant points, reducing the total query result set. This relation can be appreciated in Figure 28, where the outreach of the result sets of each distance parameter can be seen, including the projected points retrieved by the query at the different distance parameters (HoM Border Mirroring), and the real Pols that were represented by these projections and that will be sent to the user for filtering (HoM Border Real Pols).

At scale 6, as mentioned previously, 30,000 units of code difference seems like a conservative distance to assume as valid for the given search range, but a finer granularity is recommended.

The behavior is very similar to the one at scale 5, as it can be seen in Figure 29, where the area around the user’s location is clearly denoted by the Pols in the results. The colorful points represent the projected locations of the Pols, while the black and gray ones represent the real location of the Pols that appeared on the query, based on the projection. Note that the colorful points follow a pattern and locate themselves under the user’s location, which, in this scenario, is very close to a border. This behavior can be explained by the fact that the points in the grid cell right above the user’s location do have a great code difference that will not show up in a regular range-based query, illustrating the problem of lack of locality. However, these points were projected in the user’s grid cell and thus, they could be found via code difference very easily; that is the reason why the black and gray points appear above the user’s location but not the colorful ones. This behavior can also be seen at scale 6, where the projection of the points is what allows its reachability.

Now, the concept of distance is not as exact as desired, because, it can be seen how some black Pols, which should be the farthest away points, are sometimes closer than the gray ones. This issue clearly reduces the accuracy of the code distance for range queries, even though it works well enough for a rough estimation.

This work did not include a structured study of total query time because the results of all the techniques was very similar, around 120 ms, with some exceptions for the regular geographical query, which in some scenarios took much longer, sometimes over a second. However, the comparison was not fair because the database was inflated by all the projections of all mirroring techniques, at several scales. When tested in smaller databases, the query times reduces substantially, to around 20 ms. This variable is very important for scalability of the solution and will be addressed with more detail in a future work. A larger dataset of points could be
used to test the real impact of the projections in the query execution time.

Finally, the use of a fixed search range, based on the scale, poses a limitation to the results of this work, because the idea was to limit the number of involved grid cells and not to extend the search area excessively; however, the experiments showed the different behavior of two different distances that may match a good number of search areas of interest for a user.

VI. CONCLUSIONS AND FUTURE WORK

This paper introduces the Hall of Mirrors, a technique to implement coding-based private location query based on GeoHash, that overcomes the problem of border line locations by projecting the PoIs into all the adjacent grid cells, allowing both common prefix and distance-based queries. Four different techniques were proposed to project the PoIs in adjacent cells: HoM-AHG, which used the AHG user’s location projection which mirrors the PoI’s relative position to its own centroid in the adjacent cells; HoM-Centroid works similar to HoM-AHG but changes the orientation of the point to locate it as close as possible to the original PoI; HoM-Border which mirrors the location of the point against the nearest border to the adjacent cell; and HoM-Scale which locates the projected point over a direct line from the PoI to the adjacent centroid.

Results show that the PoI projection technique in all the HoM algorithms is able to acquire all relevant locations, in both common prefix and distance-based queries. In addition, the distance-based queries, the HoM-Centroid, Border and Scale tend to produce a smaller result set in most scenarios, due to the fact that the distance parameter works more precisely compared to the AHG technique. Also, this results proves that the difference between the numerical representation of the GeoHash code can be used as an estimation of distance, with values of 600,000 for scale 5 and 30,000 for scale 6 searches.

HoM Border Mirroring showed the best performance in many of the scenarios, due in part to the fact that projections of the PoIs in non directly adjacent cells were projected very far from the original location, making them not easily reachable in distance-based queries, which in short-range queries can be a desired characteristic to reduce the size of the result set even more. This aspect will be explored in order to avoid including non-relevant projections in the database.

Another important issue to address in future work is to evaluate the use of GeoHash as the coding technique, given that intertwined nature of the construction of the code generates problems of interpretation and does not allow a more precise representation of distance among the coded points. In addition, the non intuitive division of the space makes it difficult for processing the data.

REFERENCES

[1] M. Duckham and L. Kulik, “A formal model of obfuscation and negotiation for location privacy,” in Proceedings of PERVASIVE 2005 - LNCS 3468, pp. 152–170, Springer-Verlag Berlin Heidelberg, 2005.
[2] M. Zurbaran, L. Gonzalez, P. Wightman Rojas, and M. Labrador, “A Survey on Privacy in Location-Based Services,” Ingenieria y Desarrollo, vol. 32, no. 2, pp. 33–51, 2014.
[3] N. Guo, W. Xiong, Y. Wu, L. Chen, and N. Jing, “A Geographical Meshing and Coding Method Based on Adaptive Hilbert-GeoHash,” IEEE Access, vol. 7, pp. 39815–39825, 2019.
[4] J. Krumm, “Inference attacks on location tracks,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4480 LNCS, pp. 127–143, 2007.
[5] P. Wightman, M. Zurbaran, and A. Santander, “High variability geographical obfuscation for location privacy,” in Proceedings - International Carnahan Conference on Security Technology, 2013.
[6] P. Wightman, M. Zurbaran, E. Zurek, A. Salazar, D. Jabba, and M. Jimeno, “θ-Rand: Random noise-based location obfuscation based on circle sectors,” in ISIE 2013 - 2013 IEEE Symposium on Industrial Electronics and Applications, 2013.
[7] P. Wightman, W. Coronell, D. Jabba, M. Jimeno, and M. Labrador, “Evaluation of location obfuscation techniques for privacy in location based information systems,” in 2011 IEEE Latin-American Conference on Communications, LATINCOM 2011 - Conference Proceedings, 2011.
[8] M. Zurbaran, K. Avila, P. Wightman, and M. Fernandez, “Near-Rand: Noise-based location obfuscation based on random neighboring points,” IEEE Latin America Transactions, vol. 13, no. 11, 2015.
[9] P. M. Wightman, M. A. Jimeno, D. Jabba, and M. Labrador, “Matlock: A location obfuscation technique for accuracy-restricted applications,” in 2012 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1829–1834, 2012.
[10] P. M. Wightman, M. Zurbaran, M. Rodriguez, and M. A. Labrador, “MaPIR: Mapping-based private information retrieval for location privacy in LBSs,” in Proceedings - Conference on Local Computer Networks, LCN, pp. 964–971, oct 2013.
[11] R. Al-Dhubhani and J. Cazalas, “Correlation analysis for geodistinguishability based continuous LBS queries,” in 2017 2nd International Conference on Anti-Cyber Crimes (ICACC), pp. 203–208, IEEE, mar 2017.
[12] M. Zurbarán, P. Wightman, M. Brovelli, D. Oxloli, M. Iliffe, M. Jimeno, and A. Salazar, “NRand-K: Minimizing the impact of location obfuscation in spatial analysis,” Transactions in GIS, aug 2018.
[13] M. Zurbaran and P. Wightman, “VoKA: Voronoi K-aggregation mechanism for privacy in location-based information systems,” in Proceedings - International Carnahan Conference on Security Technology, vol. 2017-October, pp. 1–6, Institute of Electrical and Electronics Engineers Inc., dec 2017.
[14] A. Kapadia, N. Triandopoulos, C. Cornelius, D. Peebles, and D. Kotz, “AnonySense: Opportunistic and privacy-preserving context collection,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 5013 LNCS, pp. 280–297, 2008.
[15] W. L. Croft, W. Shi, J.-R. Sack, and J.-P. Corriveau, “Comparison of approaches of geographic partitioning for data anonymization,” Journal of Geographical Systems, vol. 19, pp. 221–248, jul 2017.
[16] W. Xiang, “An Efficient Location Privacy Preserving Model based on GeoHash,” in BESC 2019 - 6th International Conference on Behavioral, Economic and Socio-Cultural Computing, Proceedings, Institute of Electrical and Electronics Engineers Inc., oct 2019.
[17] S. Gao, J. Ma, C. Sun, and X. Li, “Balancing trajectory privacy and data utility using a personalized anonymization model,” Journal of Network and Computer Applications, 2014.
[18] B. Henne, C. Kater, M. Smith, and M. Brenner, “Selective cloaking: Need-to-know for location-based apps,” in 2013 11th Annual Conference on Privacy, Security and Trust, PST 2013, 2013.
[19] I. S. Suwardi, D. Dharma, D. P. Satya, and D. P. Lestari, “GeoHash index based spatial data model for corporate,” in Proceedings - 5th International Conference on Electrical Engineering and Informatics: Bridging the Knowledge between Academic, Industry, and Community, ICEEI 2015, pp. 478–483, Institute of Electrical and Electronics Engineers Inc., dec 2015.
[20] W. Xiong, R. Li, J. Peng, Y. Wu, N. Guo, and N. Jing, “Geo-gap tree: A progressive query and visualization method for massive spatial data,” IEEE Access, vol. 7, pp. 99428–99440, 2019.
[21] H. Li, X. Xue, Z. Li, L. Li, and J. Xiong, “Location privacy protection scheme for lbs in iot,” Wireless Communications and Mobile Computing, vol. 2021, 2021.
[22] G. Niemeyer, “GEOCACHING - geohash.org,” 2008.
[23] C. Cheng, X. Tong, B. Chen, and W. Zhai, “A subdivision method to unify the existing latitude and longitude grids,” ISPRS International Journal of Geo-Information, vol. 5, no. 9, 2016.
[24] Y. Lei, X. Tong, Y. Zhang, C. Qiu, X. Wu, G. Lai, H. Li, C. Guo, and Y. Zhang, “Global multi-scale grid integer coding and spatial indexing: A novel approach for big earth observation data,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 163, pp. 202–213, may 2020.
[25] T. Qu, L. Wang, J. Yu, J. Yan, G. Xu, M. Li, C. Cheng, K. Hou, and B. Chen, “STGIA spatio-temporal grid index model for marine big data,” Big Earth Data, vol. 4, pp. 435–450, oct 2020.
[26] D. G. L. Larikin-York, “S2 Geometry,” 2008.
[27] B. Lin, L. Zhou, D. Xu, A. X. Zhu, and G. Lu, “A discrete global grid system for earth system modeling,” International Journal of Geographical Information Science, vol. 32, pp. 711–737, apr 2018.
[28] X. Sui and W. Zhou, “A position-coding terminal device for ships based on beidou navigation satellite system,” vol. 693, 2021.
[29] J. Verma, S. Mankad, and S. Garg, “Geohash tag based mobility detection and prediction for traffic management,” SN Applied Sciences, vol. 2, no. 8, 2020.
[30] S. Shah, V. Meduri, and M. Sarwat, “Gem: An efficient entity matching framework for geospatial data,” pp. 346–349, 2021.
[31] C. Zhou, H. Lu, Y. Xiang, J. Wu, and F. Wang, “GeoHashTile: Vector geographic data display method based on geohash,” ISPRS International Journal of Geo-Information, vol. 9, no. 7, 2020.
[32] Z. Li and Z. Zhao, “Mgeo: A hash coding technique for geospatial data pre-processing,” pp. 241–246, 2021.
[33] T. Kim, J. Lee, K.-S. Kim, A. Matono, and K.-J. Li, “Utilizing extended geocodes for handling massive three-dimensional point cloud data,” World Wide Web, vol. 24, no. 4, pp. 1321–1344, 2021.
[34] H. Irshaid, M. Hasan, R. Hasan, and J.-S. Oh, “User activity and trip recognition using spatial positionning system data by integrating the geohash and gis approaches,” Transportation Research Record, vol. 2675, no. 4, pp. 391–405, 2021.
[35] F. Baig, D. Teng, J. Kong, and F. Wang, “Spear: Dynamic spatio-temporal query processing over high velocity data streams,” vol. 2021-April, pp. 2279–2284, 2021.
[36] N. Yang and H. Tang, “Semantic segmentation of satellite images: A deep learning approach integrated with geospatial hash codes,” Remote Sensing, vol. 13, no. 14, 2021.
[37] Y. Yan, K. Gai, P. Jiang, L. Xu, and L. Zhi, “Location-based Privacy-preserving Techniques in Connected Environment: A Survey,” in Proceedings - 4th IEEE International Conference on Smart Cloud, SmartCloud 2019 and 3rd International Symposium on Reinforcement Learning, ISRL 2019, pp. 156–162, Institute of Electrical and Electronics Engineers Inc., dec 2019.
[38] X. Lei, A. X. Liu, R. Li, and G. H. Tu, “SeeEQP: A secure and efficient scheme for SKNN query problem over encrypted geodata on cloud,” in Proceedings - International Conference on Data Engineering, vol. 2019-April, pp. 662–673, IEEE Computer Society, apr 2019.
[39] D. Lee, R. Moussalli, S. Asaad, and M. Srivatsa, “Spatial Predicates Evaluation in the Geohash Domain Using Reconfigurable Hardware,” in Proceedings - 24th IEEE International Symposium on Field-Programmable Custom Computing Machines, FCCM 2016, pp. 176–183, Institute of Electrical and Electronics Engineers Inc., aug 2016.
[40] A. Ye, Q. Chen, and L. Xu, “Private and Flexible Proximity Detection Based on GeoHash,” in IEEE Vehicular Technology Conference, vol. 2017-June, Institute of Electrical and Electronics Engineers Inc., nov 2017.
[41] A. Aloufi, P. Hu, H. Liu, S. S. Chow, and K. K. R. Choo, “Universal location referencing and homomorphic evaluation of geospatial query,” Computers and Security, vol. 102, p. 102137, mar 2021.
[42] T. Ningsheng, Y. Chongjun, Y. Liuszhong, and L. Yuan, “An address regional-isation method for spatial subdivision and geocoding in digital earth system,” International Journal of Digital Earth, vol. 8, pp. 825–839, oct 2015.
[43] M. Varalakshmi, A. Kesarkar, and D. Lopez, “High-performance implementation of a two-bit geohash coding technique for nearest neighbor search,” Concurrency and Computation: Practice and Experience, vol. 33, no. 5, 2021.
[44] R. Guo, B. Qin, Y. Wu, R. Liu, H. Chen, and C. Li, “MixGeo: Efficient secure range queries on encrypted dense spatial data in the cloud,” in Proceedings of the International Symposium on Quality of Service, IWQoS 2019, Association for Computing Machinery, Inc., jun 2019.
[45] D. Shen, J. Fang, and Y. Han, “A nearby vehicle search algorithm based on HBase spatial index,” in Proceedings - 2015 12th Web Information System and Application Conference, WISA 2015, pp. 71–74, Institute of Electrical and Electronics Engineers Inc., feb 2016.
[46] S. He, L. Chu, and X. Li, “Spatial query processing for location based application on Hbase,” in 2017 IEEE 2nd International Conference on Big Data Analysis, ICDBA 2017, pp. 110–114, Institute of Electrical and Electronics Engineers Inc., oct 2017.
[47] Q. Liu, X. Tan, F. Huang, C. Peng, Y. Yao, and M. Gao, “GB-Tree: An efficient LBS location data indexing method,” in 2014 The 3rd International Conference on Agro-Geoinformatics, Agro-Geoinformatics 2014, Institute of Electrical and Electronics Engineers Inc., sep 2014.

PEDRO WIGHTMAN B.Sc. in Systems Engineering from the Universidad del Norte. Doctor in Computer Science and Engineering from the University of South Florida. Principal professor at the University of Applied Science and Technology of the Universidad del Rosario, and Director of the Bachelors program of Applied Mathematics and Computer Science. Senior member of IEEE and Senior researcher of the Ministry of Science. His research lines are location-based services, especially focused on location data privacy, blockchain and medical information systems, communication infrastructure for the internet of things and industry 4.0. Author of 3 technical books and several publications in indexed journals and international events.

MAYRA ZURBARAN She received the B.S. degree in Systems Engineering from Universidad del Norte, Barranquilla, in 2011 and the Ph.D. degree in Systems Engineering and Computing from Universidad del Norte, Barranquilla, in 2018. From 2013 to 2015 she was a Research Assistant with the Systems Engineering Department, Universidad del Norte, Barranquilla. From 2015 to 2016 she was a Ph.D. Intern and from 2018 to 2020 she was a Research Fellow with the Environmental and Civil Engineering Department at Politecnico di Milano, Italy. She is currently working as an external consultant for the Joint Research Centre under the company Seidor. She is the author of 2 books and 28 articles. Her research interests include location privacy, geographic information systems, and geospatial data analysis.

AUGUSTO SALAZAR He received his B.S. degree in Systems Engineering from Universidad del Norte, Barranquilla, Colombia, and his M.S. degree in computer science from National Chiao Tung University, Hsinchu, Taiwan, in 2012. He was with the Industry on embedded systems for a period of nine years on companies, such as Ericsson LMF (FI), Hitron Technologies (TW), and Proceed Comunications (TW). Since 2012, he has been an Assistant Professor with the Department of Systems Engineering, Universidad del Norte. His research interests include embedded system, mobile application development and game analytics.
LORENA GARCIA She received her B.S. degree in Electronics Engineering degree from Universidad del Norte (2006) and the M.Sc. in Electronic and Computer Engineering from Universidad de los Andes (2008). She has more than 15 years of experience in academic administration, teaching and research in important institutions. Currently, she is Professor and Director of Laboratories and Infrastructure of the School of Engineering and Basic Sciences of Universidad Central. She has also been consultant on design and certification of internal telecommunications networks. Senior member and currently the Chair of the Pre-University Education Coordinating Committee at the Educational Activities Board of IEEE.