2013 IEEE 14th International Conference on Mobile Data Management

Gonio, *Aequus* and *Incognitus*: three spatial granularities for privacy-aware systems

Letizia Bertolaja, Dragan Ahmetovic and Sergio Mascetti
Department of Computer Science, University of Milan
{letizia.bertolaja, dragan.ahmetovic, sergio.mascetti}@unimi.it

***Abstract*—Many solutions proposed in the literature to enforce privacy in presence of location information use, implicitly or explicitly, spatial granularities. However, most of the contributions do not describe the formal and computational properties of this tool in details. In this paper we propose three families of spatial granularities, specifically designed for privacy-aware systems. We mathematically characterize them and prove that all of them have two important formal properties. Then, for each of them, we show how to efficiently compute two fundamental operations.**

I. Introduction

Spatial granularity has often been used in the literature as formal tool to partition a spatial domain into a set of sub-regions called "granules" [13], [2], [4]. In recent years spatial granularities have been extensively used, implicitly or explicitly, in several papers in the field of privacy preservation in presence of location information. A comprehensive survey on privacy preserving techniques for private queries is presented in [6]. In privacy techniques, a granule can model an approximate user location, preventing the disclosure of the exact user position, therefore guaranteeing a form of privacy.

Considering the use of spatial granularities for privacy preservation, we can identify two different approaches in the literature. Some papers uses granules as "minimum uncertainty regions" (MUR) [10], in the sense that the users is known to be located in that granule, but the position of the user within the granules should not be disclosed [7], [9], [14], [1]. In this case, the size of the granule represents the level of privacy protection, in the sense that larger area correspond to higher privacy protection. We call this the "MUR approach". The second approach, that we call "k-objects approach", consists in using granules that contain a sufficiently large number of objects. For example, in the approaches extending the notion of k-anonymity, the considered objects are persons, and the intuition is that it is harder to re-identify an anonymous user if she reports to be located in a granule where other $k-1$ persons are located [8], [5], [15]. In this approach, the level of privacy depends on the number of objects in the reported area and not on the size of the disclosed granule. Finally there are contributions in which a "mixed approach" is used in the sense that granules both have a minimum dimension and contain a minimum number of objects [11], [12].

A common problem in the contributions mentioned above is that spatial granularities are used, often implicitly, as a tool, without extensively describing their properties and basic operations.

In this paper we provide a general definition of spatial granularities and we identify two basic operations and two properties, "non-overlapping" and "space-covering", that are particular relevant when using spatial granularities as a tool to protect privacy. We also introduce the concept of "family of granularities", representing set of granularities with similar characteristics but different size of the granules. The main contribution of this paper consists in the definition of three family of granularities, specifically designed to be used in privacy preserving applications. The first two families "Gonio" (labeled $\mathcal{G}$) and "Aequus" (labeled $\mathcal{A}$) are designed for the "MUR approach", while the "Incognitus" (labeled $\mathcal{I}$) family of granularities is designed for the "k-objects approach" and can be easily extended to support the "mixed approach". For each family of granularities we show how to efficiently compute the two basic operations and we prove that they respect the "non-overlapping" and "space-covering" properties.

II. Problem Formalization

In this paper we consider the entire world as the reference spatial domain S. We use standard latitude-longitude coordinates, hence: $S = (-90, 90) \times [-180, 180)$. We define a *spatial granularity* (or simply "granularity") G as a mapping from a subset of $\mathbb{N}$ to S. For each $G(i)$ we call i an *index* and, if $G(i)$ is defined, we call $G(i)$ a *granule*, denoted as g. If g is a granule of a granularity G, we denote it with $g \in G$.

In this paper we design granularities for privacy-aware applications and we identify two relevant properties for this type of applications. First, privacy protection should be provided everywhere, hence we need granularities that are "space-covering" meaning that any point of the spatial domain should be covered by the granules, as formalized in Definition 1.

Definition 1. *A granularity G is* space-covering *if, for every point $p \in S$, there exists one granule $g \in G$ such that $p \in g$.*

The "non-overlapping" property, formalized in Definition 2, is related to the fact that it should be possible to uniquely identify a granule covering a point in the space. In other words, there should be no intersection between the granules of a given granularity. The lack of the "non-overlapping" property has been noted to negatively impact the privacy guarantees of the "MUR approach" [3].

Definition 2. *A granularity G is* non-overlapping *if, for every pair of $g_1, g_2 \in G$, with $g_1 \neq g_2$, $g_1 \cap g_2 = \emptyset$*

In this paper we are interested in defining families of granularities. The intuition is that all the granularities belonging to a family share the same characteristics, but have different "size" of the granules. Formally, a *family of granularities* is

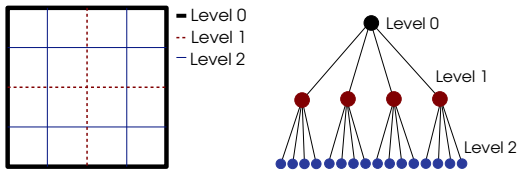
978-0-7695-4973-6/13 $26.00 © 2013 IEEE
DOI 10.1109/MDM.2013.70

a mapping associating a granularity to an element of $\mathbb{N}$ that we call "level". Given a family of granularities $\mathcal{F}$, we denote with $\mathcal{F}^l$ the granularity G having level l in the family $\mathcal{F}$.

In the following we characterize each granularity G by defining $G(i)$ for each index $i \in \mathbb{N}$ and we show how to efficiently compute this function. The "inverse" function $G[p]$ returns, for each $p \in S$, the index i such that $p \in G(i)$. Note that, since we consider "non-overlapping" and "space-covering" granularities, $G[p]$ is always defined, in the sense that for each point $p \in S$ there exists exactly one granule g of G such that $p \in g$. Once it is known how to compute $G(i)$ for every index i, it is possible, in theory, to compute $G[p]$ by computing $G(i)$ for each index i and choosing the one such that $p \in G(i)$. However, this is impractical and in the following of this paper we describe how to efficiently compute $G[p]$.

III. The $\mathcal{G}$onio granularities

The $\mathcal{G}$ granularity family partitions the spatial domain S into rectangular granules all sharing the same angular dimension. The level of the granularity represent the number of granules, where $\mathcal{G}^0$ has only one granule which comprises the entire spatial domain while a granularity of level l has 4^l granules, organized in a grid of 2^l granules along each dimension. This is intuitively analogous to a balanced quadtree as shown in Figure 1. In the following we will refer to this hierarchical structure as the "granularity tree".



(a) Projection plane (b) granularity tree

Fig. 1. Different granularity levels with $\mathcal{G}$ granularity family on the plane and their representation in terms of granularity tree

As shown in Figure 2, the granule with index zero has the bottom-left corner in $90°S, 180°W$, while the others granules are enumerated from left to right and from bottom to top. Since a granularity $\mathcal{G}^l$ has 2^l granules along each dimension, the angular size of each granule is $180°/2^l$ on the vertical dimension and $360°/2^l$ on the horizontal dimension.

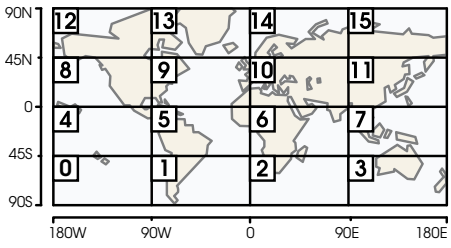


Fig. 2. $\mathcal{G}^2$ shown on the world

The regular structure of $\mathcal{G}$ granularities allows to define the position of each granule in terms of its column X_i and row Y_i. This is an alternative format to index granules that will be useful in the following of this presentation. Clearly, it is possible to convert from the "single-index" format to the "double-index" format in constant time using these formulas:

$$X_i = i \mod 2^l \qquad Y_i = \lfloor \frac{i}{2^l} \rfloor$$

The inverse conversion (i.e., from $\langle X_i, Y_i \rangle$) to the "single-index" is the following:

$$i = X_i + 2^l \cdot Y_i \tag{1}$$

We now formally characterize a granularity $\mathcal{G}^l$ through the specification of $\mathcal{G}^l(i)$ for a generic index i.

Definition 3. *Let l and i be two non-negative integers. $\mathcal{G}^l(i)$ is undefined for $i \geq 4^l$. Otherwise*

$$\mathcal{G}^l(i) = [minLon, maxLon) \times [minLat, maxLat)$$

where:

$$minLon = X_i \cdot \left(\frac{360}{2^l}\right) - 180 \tag{2}$$

$$maxLon = (X_i + 1) \cdot \left(\frac{360}{2^l}\right) - 180 \tag{3}$$

$$minLat = Y_i \cdot \left(\frac{180}{2^l}\right) - 90 \tag{4}$$

$$maxLat = (Y_i + 1) \cdot \left(\frac{180}{2^l}\right) - 90 \tag{5}$$

Note that, in order to guarantee the "space-covering" and "non-overlapping' properties, Definition 3 specifies that any point on the minimum coordinates (left-bottom of the granule) is part of the granule, while any point on the maximum coordinates (right-top of the granule) is not part of the granule.

We now show how to compute $\mathcal{G}^l[p]$ in constant time. That is, how to find the index i of the granule containing the point p (expressed in the $\langle lat, lon \rangle$ format). The column and row positions are computed as the proportion of longitude and latitude (respectively) with respect to the size of the horizontal and vertical domain (respectively). Formally:

$$X_i = \left\lfloor \frac{2^l \cdot (lon + 180)}{360} \right\rfloor \tag{6}$$

$$Y_i = \left\lfloor \frac{2^l \cdot (lat + 90)}{180} \right\rfloor \tag{7}$$

Consequently the $\mathcal{G}^l[p]$ can be defined as follows:

$$\mathcal{G}^l[p] = \left\lfloor \frac{2^l \cdot (lon + 180)}{360} \right\rfloor + 2^l \cdot \left\lfloor \frac{2^l \cdot (lat + 90)}{180} \right\rfloor \tag{8}$$

Given the above definitions, the functions $\mathcal{G}^l(i)$ and $\mathcal{G}^l[p]$ can be clearly computed in constant time.

It is also easily seen that $\mathcal{G}$ granularities are "space-filling" and "non-overlapping", as shown in Properties 1 and 2.

Property 1. *For every level l, $\mathcal{G}^l$ is space-covering.*

Proof: By Definition 1, we need to prove that index $i = \mathcal{G}^l[p]$ is defined, for any $l \in \mathbb{N}$ and for any $p =$

$\langle lat, lon \rangle \in S$. Since $lon \in [-180, 180)$, by Equation 6, $X_i \in [0, 2^l) = [0, 2^l - 1]$. Since $lat \in (-90, 90)$, by Equation 7, $Y_i \in [0, 2^l) = [0, 2^l - 1]$. Consequently, by Equation 1, $i \in [0, 4^l)$. By Definition 3, $\mathcal{G}^l(i)$ is defined. ■

Property 2. *For any level l, $\mathcal{G}^l$ is non-overlapping.*

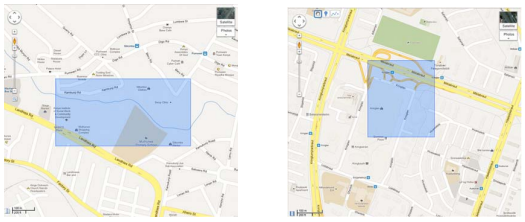
Proof: By Definition 2, we need to prove that, for any level l, given $i, j \in [0, 4^l)$, with $i \neq j$, $\mathcal{G}^l(i) \cap \mathcal{G}^l(j) = \emptyset$. Since $i \neq j$, then $X_i \neq X_j$ or $Y_i \neq Y_j$ or both. We now show that if $X_i \neq X_j$ then $\mathcal{G}^l(i) \cap \mathcal{G}^l(j) = \emptyset$. The proof is analogous for the case $Y_i \neq Y_j$. Without loss of generality, let $X_i < X_j$. By Definition 3, $maxLon$ of $\mathcal{G}^l(i)$ is less than or equal to $minLon$ of $\mathcal{G}^l(j)$ and hence, by Definition 3, $\mathcal{G}^l(i) \cap \mathcal{G}^l(j) = \emptyset$.

■

IV. The $\mathcal{A}$equus granularities

As specified in Section III, the granules of a $\mathcal{G}$ granularity have the same angular dimension. Thus, granules at different latitudes covers areas of different size. Consider Example 1.

Example 1. *We consider two granules of $\mathcal{G}^{16}$, one located in Nairobi, Kenya ($-1.2877, 36.8372$) and the other in Reykjavik, Iceland ($64.1324, -21.8934$). The granule positioned in Nairobi has an area more than two times the area of the granule calculated in Reykjavik, as shown in Figure 3.*



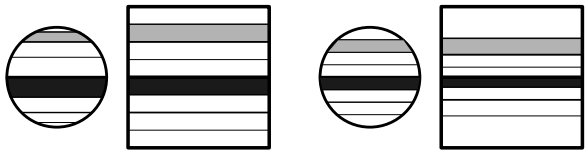
(a) Nairobi-186314.2m^2 (b) Reykjavik-81281.7m^2

Fig. 3. Example of granules of $\mathcal{G}^{16}$ with different areas. Images retrieved on Feb 15, 2013 from http://maps.google.com

Different dimensions of the granules are undesirable in some applications. For example, in the "MUR approach" to privacy preservation, a $\mathcal{G}$ granularity yields different level of privacy in different regions of the world. In this section we present the $\mathcal{A}$ granularities, an extension of $\mathcal{G}$ granularities, with the following property: all granules of $\mathcal{A}^l$ have the same area, independently from their latitude.

Technically, $\mathcal{A}$ granularities are specified like $\mathcal{G}$ granularities; the only difference is in the computation of the horizontal boundaries of the granules. To understand the intuition behind our solution, we suggest to think about a $\mathcal{G}$ granularity as a set of horizontal stripes that are vertically "cut" to create the granules. In this view, each stripe is divided into the same number of granules all having the same width. Since we are considering Earth as our reference space, these "stripes" are actually spherical segments[1].

In a $\mathcal{G}$ granularity the spherical segments have the same angular height, so they appear as having the same area in the Equirectangular projection, but they actually have different areas (see Figure 4(a)). This clearly leads the granules to have different areas. Vice versa, we define $\mathcal{A}$ granularities with spherical segments having different angular height but with the same area (see Figure 4(b)). Consequently, the granules result to have the same size, as we will prove.



(a) Spherical segments in $\mathcal{G}^3$ (b) Spherical segments in $\mathcal{A}^3$

Fig. 4. Intuitive view of $\mathcal{G}$ e $\mathcal{A}$ granularities

We now define the family of $\mathcal{A}$ granularities by specifying $\mathcal{A}^l(i)$ for a generic index i.

Definition 4. *Let l and i be two non-negative integers. $\mathcal{A}^(i)$ is undefined for $i \geq 4^l$. Otherwise*

$$\mathcal{A}^l(i) = [minLon, maxLon) \times [minLat, maxLat)$$

where $minLon$ and $maxLon$ are defined as in Equations 2 and 3 and:

$$minLat = \arcsin\left(1 - \frac{2Y_i - 2}{2^l}\right) \quad (9)$$

$$maxLat = \arcsin\left(1 - 2\,\frac{Y_i}{2^l}\right) \quad (10)$$

To understand the idea behind the above definition, we first recall that the area of a spherical segment with height h is equal to $2\pi\mathcal{R}h$ where $\mathcal{R}$ is the radius of the sphere. Using the above definitions of $minLat$ and $maxLat$, the height of a spherical segment containing a generic granule $\mathcal{A}^l(i)$ is equal to

$$h = \mathcal{R} \cdot [sin(maxLat) - sin(minLat)] \quad (11)$$

$$= \mathcal{R} \cdot \left[1 - 2 \cdot \frac{Y_i}{2^l} - \left(1 - 2 \cdot \frac{Y_i}{2^l} - \frac{2}{2^l}\right)\right] = \frac{2\mathcal{R}}{2^l} \quad (12)$$

This spherical segment has consequently an area A_{seg} that is independent from i and that is actually equal to $A_{seg} = \frac{4\pi\mathcal{R}^2}{2^l}$.

Property 3 follows from the above results and it shows that all granules in a given granularity $\mathcal{A}^l$ have the same area.

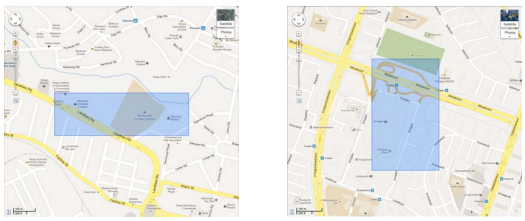
Property 3. *Let G be the granularity $\mathcal{A}^l$. Then each granule g of G has an area equal to:*

$$\frac{A_{seg}}{2^l} = \frac{4\pi\mathcal{R}^2}{4^l}$$

Example 2. *Any granule of $\mathcal{A}^{12}$ has an area of about 30km^2, while granules of $\mathcal{A}^{16}$ have an area of about 0.12km^2. Indeed, compare Figure 3 with Figure 5 that shows two granules of $\mathcal{A}^{16}$ in Nairobi, Kenya and Reykjavik, Iceland, respectively. Differently from what observed in Example 1, in this case the two granules have the same area.*

The $\mathcal{A}^l[p]$ operation (for a given point $p = \langle lat, lon \rangle$) is analogous to the same operation for $\mathcal{G}$ granularities, with the

[1]Here and in the following we approximate Earth with a sphere.



(a) Nairobi, Kenya (b) Reykjavik, Iceland

Fig. 5. Example of granules of $\mathcal{A}^{16}$ with same areas. Images retrieved on Feb 15, 2013 from http://maps.google.com

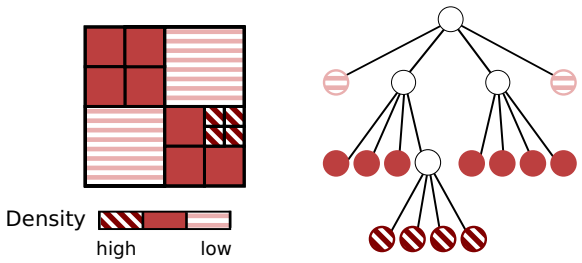


Fig. 6. The subdivision of the space in $\mathcal{I}$ according to objects density and the corresponding unbalanced quadtree

only difference in the computation of Y_i that is defined as:

$$Y_i = \left\lfloor \frac{2^l \cdot (1 - \sin(lat))}{2} \right\rfloor$$

Consequently:

$$\mathcal{A}^l[p] = \left\lfloor \frac{2^l \cdot (lon + 180)}{360} \right\rfloor + 2^l \cdot \left\lfloor \frac{2^l \cdot (1 - \sin(lat))}{2} \right\rfloor$$

Similarly to $\mathcal{G}$ granularities, also $\mathcal{A}$ granularities are "space-filling" and "non-overlapping". Proofs are analogous to the case of $\mathcal{G}$ and are omitted.

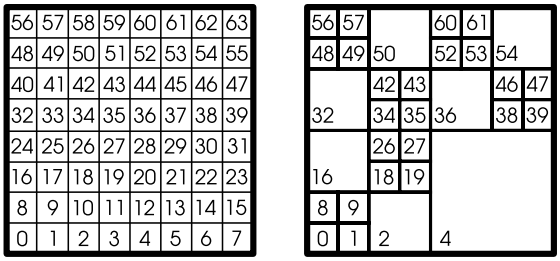
V. The $\mathcal{I}$ncognitus granularities

$\mathcal{G}$ and $\mathcal{A}$ guarantee that, given a granularity, its granules have the same size (in terms of angular dimension or actual area). In this section we tackle the problem of granularity specification from a different point of view: each granule should contain at least a given number of objects that can represents persons, points of interests, etc. The level l of the granularity determines the minimum quantity of objects in each granule. In Section V-A we characterize this family of granularities, while in Section V-B we show how to efficiently compute the $\mathcal{I}^l(i)$ and $\mathcal{I}^l[p]$ operations.

A. Specification of $\mathcal{I}$ granularities

A granularity $\mathcal{I}^l$ is composed by a combination of granules of $\mathcal{A}$ granularities at different levels. Intuitively, in the regions of the world where the density of objects is higher, $\mathcal{I}$ granules are smaller, which means that they correspond to granules of a granularity of $\mathcal{A}$ with high level. Vice versa, where the density is low, it is necessary to aggregate the granules, so that the result contains at least l objects. This form of aggregation is achieved by using granules of $\mathcal{A}$ with lower level. For example, consider Figure 6 showing a $\mathcal{I}$ granularity and the corresponding unbalanced quadtree structure. Each granule corresponds to a leaf of the tree and smaller granules correspond to leaf farther from the root of the tree.

The specification of a family of $\mathcal{I}$ granularities depends on a given $\mathcal{A}$ granularity with level m whose granules represent the smallest granules of all the granularities of $\mathcal{I}$. Indeed, the area of the smallest granules also guarantees a MUR constraint for the $\mathcal{I}$ granularities. A "mixed approach" extension of the $\mathcal{I}$ granularities is easily obtained by enforcing, not only the level l of the granularity, but also the level m of the underlying $\mathcal{A}$ granularity. The choice of $\mathcal{A}^m$ also affects the indexes of $\mathcal{I}$ granularities. Indeed, as shown in Figure 7, when granules are aggregated, the index of the "aggregated granule" is the smallest among the indexes of $\mathcal{A}^m$ granules composing it.



(a) $\mathcal{A}^3$ (b) $\mathcal{I}$ based on $\mathcal{A}^3$

Fig. 7. Example of indexing with $\mathcal{I}$

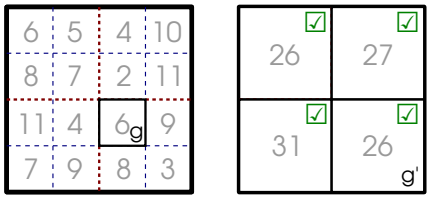
The specification of $\mathcal{I}$ granularities also depends on a function $c()$ that counts how many objects are contained in a given area. Since we want to define granularities having at least l objects in each granule, a granularity $\mathcal{I}^l$ is undefined if $l < c(S)$. In other words, we disregard the case of granularities in which the expected number of objects in each granule is larger than the number of objects in the entire world.

We now mathematically characterize $\mathcal{I}^l$ by specifying $\mathcal{I}^l(i)$ for a given integer i. Lets first define the "good ancestor". Intuitively, a granule g' is a good ancestor of another granule $g \in \mathcal{A}^m$, if g' is an ancestor of g in the corresponding $\mathcal{A}$ granularity tree and if g and each of its siblings in the granularity tree contain at least l objects. Formally:

Definition 5. *Given $g = \mathcal{A}^m(i)$ a granule $g' = \mathcal{A}^n(j)$ for $n > 0$ is a "good ancestor" of g if*
(a) $g \subseteq g'$ and
(b) given $g_f \in \mathcal{A}^{n-1}$ such that $g_f \supset g'$, $\forall g_s \in \mathcal{A}^n$ such that $g_s \subset g_f$, it holds that $c(g_s) \geq l$.

Example 3. *Let's consider granularity $\mathcal{A}^2$ and the granule g shown in Figure 8(a). The number of elements calculated by $c()$ for each granule is specified inside the granules. Let $l = 5$ be the number of objects required inside each granule. While the granule g satisfies the constraint $c(g) \geq l$, one of its siblings does not, as it contains 3 objects only. Consequently, g is not a good ancestor of itself. Instead, if we consider granularity $\mathcal{A}^1$, the ancestor of g is the granule denoted by g' in Figure 8(b). In this case g' and all its siblings contain more than l objects and hence g' is a good ancestor of g.*

Note that a granule $g = \mathcal{A}^m(i)$ can have more than one



(a) g is not a good ancestor of itself

(b) g' is a good ancestor of g

Fig. 8. Example of good ancestor with $m = 2$ and $l = 5$

good ancestor. We are interested in the smallest one (i.e., the one with lower level of granularity), called "min-good ancestor" and formally defined as follows.

Definition 6. *Given $g = \mathcal{A}^m(i)$, the* min-good ancestor *of g is S if g does not have any good ancestor, otherwise it is the smallest one among all the good ancestors of g.*

We are now ready to define $\mathcal{I}^l(i)$.

Definition 7. *Given $m \in \mathbb{N}$, a count function $c()$ and $l \in \mathbb{N}$, let i be an integer value and g the min-good ancestor of $\mathcal{A}^m(i)$. Then, $\mathcal{I}^l(i)$ is equal to*

$$\begin{cases} \textbf{\textit{undefined}} \text{ if } i \geq 4^m \\ \textbf{\textit{undefined}} \text{ if } i \neq \min_{j \in \mathbb{N}}\{j | \mathcal{I}^m(j) \subseteq g\} \\ g \text{ otherwise} \end{cases}$$

The first condition of Definition 7 states that, since we use indexes of $\mathcal{A}^m$ to index the granules of $\mathcal{I}^l$, if $i \geq 4^m$ then $\mathcal{I}^l(i)$ is undefined. The second condition is due to the fact that not all granules are defined for indexes in $[0, 4^m)$. For example, in Figure 7, $\mathcal{I}^l(21)$ is not defined because the granule of $\mathcal{I}$ containing $\mathcal{A}^3(21)$ has index 4. If the first two conditions are not met, then $\mathcal{I}^l(i)$ is equal to the min-good ancestor $\mathcal{A}^m(i)$.

We now show that $\mathcal{I}$ is both space-covering and non-overlapping.

Property 4. *For a level $l \leq c(S)$, $\mathcal{I}^l$ is space-covering.*

Proof: We show how to construct, for every $p \in S$ the granule $g' = \mathcal{I}^l[p]$. By Property 1, granule $g = \mathcal{A}^m[p]$ is defined. By Definition 6, the min-good ancestor g' of g exists. By Definition 5 and 6, $p \in g'$. Finally, since by Definition 7 $g' \in \mathcal{I}^l$, then $\mathcal{I}^l[p] = g'$. ■

Property 5. *For a level $l \leq c(S)$, $\mathcal{I}^l$ is non-overlapping.*

Proof: Let $g_1, g_2 \in \mathcal{I}^l$ with $g_1 \neq g_2$. We now prove that $g_1 \cap g_2 = \emptyset$. Since $g_1, g_2 \in \mathcal{I}^l$, by Definition 7 $\exists n_1, n_2, j_1, j_2$ such that $g_1 = \mathcal{A}^{n_1}(j_1)$, $g_2 = \mathcal{A}^{n_2}(j_2)$. If $n_1 = n_2$, then thesis follows from Property 2. Otherwise, without loss of generality, let $n_1 < n_2$ and let g' be the ancestor of g_1 at level n_2 i.e., $g' \in \mathcal{A}^{n_2}$ and $g_1 \subset g'$. If $g' \neq g_2$ then, from Property 2 it follows that $g' \cup g_2 = \emptyset$, hence $g_1 \cup g_2 = \emptyset$. The other case, namely $g' \neq g_2$ is impossible. We prove this by contradiction. Let $g' = g_2$. Then g' is the min-good ancestor of $\mathcal{A}^m(j_2)$, but this is absurd, since $g' \supset g_1$ and g_1 is min-good ancestor of $\mathcal{A}^m(j_1)$. ■

The $\mathcal{I}$ granularities are defined with the objective of supporting privacy preserving solutions adopting a "k-object approach". However this family of granularities can be easily extended to support the "mixed approach". The intuition is to have two levels characterizing each granularity, one represents the minimum number of objects within each granule, the other the minimum "size" of the granule, expressed in terms of the level of $\mathcal{A}$. To extend the formalism, it is sufficient to specify that a min-good ancestor must be at least at a given level of $\mathcal{A}$. As will be clear in the following, the extension of the algorithmic part is straightforward as well, and it is omitted here.

B. Computation of $\mathcal{I}^l(i)$ and $\mathcal{I}^l[p]$

The computation of $\mathcal{I}^l(i)$ is presented in Algorithm 1. This computation relies on the recursive procedure $minGoodAncestor$ that, starting from the granule $\mathcal{A}^m(i)$, traverses the granularity tree towards the root, until it finds the min-good ancestor of $\mathcal{A}^m(i)$. The output of the $minGoodAncestor$ is a pair of integers, one representing the level of the $\mathcal{A}$ granularity, the other the index. Lines 1 and 4 check the validity of the index, according to Definition 7. If the index is not valid, **null** is returned, meaning that the granule is undefined. Otherwise the algorithm returns the min-good ancestor of $\mathcal{A}^m(i)$. Note that Line 3 can be computed in constant time as follows:

$$i_m = 2^{m-n}(j \mod 2^n) + 2^m \cdot 2^{m-n} \cdot \left\lfloor \frac{j}{2^n} \right\rfloor \qquad (13)$$

Algorithm 1 $\mathcal{I}^l(i)$

Input: The level l of the granularity, the index i of the granule, the level m of the bottom granularity, the function $c()$.
Output: **null** if $\mathcal{I}^l(i)$ is undefined, the region covered by $\mathcal{I}^l(i)$, otherwise.
Procedure:

1: **if** $(i \geq 4^m)$ **return null**
2: $\langle j, n \rangle = minGoodAncestor(i, m, l, c())$
3: $i_m = min\{i' \in \mathbb{N} | \mathcal{A}^m(i') \subset \mathcal{A}^n(j)\}$
4: **if** $(i \neq i_m)$ **return null** {undefined granule}
5: **else return** $\mathcal{A}^n(j)$;

The min-good ancestor is computed recursively (see Procedure 2). At each recursion step, the algorithm moves one level up in the granularity tree (Line 8). There are two termination conditions. First, when we the procedure reaches level 0, which means that S is returned (Line 1). Second, when the the procedure finds the min-good ancestor (Line 5).

We now show how it is possible to compute the "parent" of a given granule $g = \mathcal{A}^n(i)$, i.e., how to compute $g_p = \mathcal{A}^{n-1}(j)$ such that $g \subset g_p$ (Line 2). The idea is to first compute the granule $g_m = \mathcal{A}^m(i)$. Let $X = i \mod 2^m$ be the column number of g_m. Then, the column number X_p of g_p is

$$X_p = \left\lfloor \frac{X}{2^{m-n+1}} \right\rfloor$$

Analogously, the row number Y of g_m is $\lfloor \frac{i}{2^m} \rfloor$ and the row number Y_p of g_p is:

$$Y_p = \left\lfloor \frac{Y}{2^{m-n+1}} \right\rfloor$$

Procedure 2 $minGoodAncestor(i, n, l, c)$

Input: the index i of the granule, the level n of $\mathcal{A}$, the level l of $\mathcal{I}$ and function $c()$.
Output: a pair $\langle j, n\rangle$ where j is an index and n the level of $\mathcal{A}$.
Procedure:

1: **if** $(n = 0)$ return S {entire world}
2: $p \in \mathbb{N}$ s.t. $\mathcal{A}^n(j) \subset \mathcal{A}^{n-1}(p)$ {parent of j}
3: $C = \{g_c \in \mathcal{A}^n \mid g_c \subset \mathcal{A}^{n-1}(p))\}$ {children of j}
4: **if** $(\forall g_c \in C\ c(g_c) \geq l)$ **then**
5: **return** $\langle j, h\rangle$ {min-good ancestor is found}
6: **else**
7: {recursively move up in the tree}
8: **return** $minGoodAncestor(p, n-1, l, c())$
9: **end if**

Consequently,

$$j = \left\lfloor \frac{i \mod 2^m}{2^{m-n+1}} \right\rfloor + 2^{n-1} \left\lfloor \frac{\lfloor \frac{i}{2^m} \rfloor}{2^{m-n+1}} \right\rfloor$$

Analogously to the "parent" (Line 2), it is possible to compute set C (Line 3) in constant time as follows:

$$C = \{j, j + 2^{m-n}, j + 2^{2m-n}, j + 2^{m-n} + 2^{2m-n}\}$$

Overall, each execution of the $minGoodAncestor$ takes constant time and this procedure is run at most m times (since m is the height of the granularity tree). Consequently, the worst case time complexity of Algorithm 1 is $O(m)$.

Algorithm 3 $\mathcal{I}^l[p]$

Input: The level l of the granularity, a point p, the level m of the bottom granularity, the function $c()$.
Output: the index i such that $p \in \mathcal{I}^l(i)$.
Procedure:

1: $i = \mathcal{A}^m[p]$
2: $\langle j, n\rangle = minGoodAncestor(i, m, l, c())$;
3: **return** $\min\{i' \in \mathbb{N} \mid \mathcal{A}^m(i') \subset \mathcal{A}^n(j)\}$

Once the $minGoodAncestor$ procedure is defined, it is simple to show how to compute $\mathcal{I}^l[p]$ for a give point p. Indeed, as shown in Algorithm 3, it is sufficient to compute the granule g of $\mathcal{A}^m$ that contains p (as explained in Section IV) and, after running $minGoodAncestor$, it is only necessary to compute the smallest index of granules of $\mathcal{A}^m$ composing $\mathcal{A}^n$ (this can be obtained in constant time with Equation 13). Consequently, the worst case time complexity of Algorithm 3 is $O(m)$.

Since both $\mathcal{I}^l(i)$ and $\mathcal{I}^l[p]$ can be computed in time $O(m)$ our solution if efficient also for fine underlying granularities $\mathcal{A}^m$. On the other hand, it should be observed that Incognitus is not optimal in the sense that granules of a granularity $\mathcal{I}^l$ can actually contain more than l objects. For example, assuming a uniform distribution of the objects, in the worst case we can have $l-1$ objects in each granule of $\mathcal{A}^m$, which means that the granules of $\mathcal{I}^l$ contain $4(l-1)$ objects. We leave as a future work the empirical analysis of the average number of objects in each granule of $\mathcal{I}^l$.

VI. Conclusions and future works

In this paper we described the formal and computational characteristics of three families of spatial granularities specifically designed for privacy-aware systems. As a future work, we intend to investigate new families of spatial granularities, and to show how to compute other basic operations. Most importantly, we intend to extend this line of research to spatio-temporal granularities that presents new theoretical and computational challenges.

Acknowledgment

The authors would like to thank Claudio Bettini for preliminary discussion about this paper. This work was partially supported by Italian MIUR under grants FIRB-RBFR081L58_002.

References

[1] Osman Abul, Francesco Bonchi, and Mirco Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *24th ICDE Int. Conf. on Data Engineering*. IEEE, 2008.

[2] Thomas Bittner and Barry Smith. A taxonomy of granular partitions. In *Spatial Information Theory*, L.N.C.S. Springer, 2001.

[3] Maria Luisa Damiani, Elisa Bertino, and Claudio Silvestri. The probe framework for the personalized cloaking of private locations. *Trans. on Data Privacy*, 2010.

[4] Iginia De Fent, Donatella Gubiani, and Angelo Montanari. Granular geograph: a multi-granular conceptual model for spatial data. In *Symposium on Advanced Database Systems*, 2005.

[5] Bugra Gedik and Ling Liu. Location privacy in mobile systems: A personalized anonymization model. In *Proc. of 25th ICDCS Int. Conf. on Distributed Computing Systems*, 2005.

[6] Gabriel Ghinita. Private queries and trajectory anonymization: a dual perspective on location privacy. *Tran. on Data Privacy*, 2009.

[7] Gabriel Ghinita, Maria Luisa Damiani, Claudio Silvestri, and Elisa Bertino. Preventing velocity-based linkage attacks in location-aware applications. In *Proc. of the 17th SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems*. ACM, 2009.

[8] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. of the 1st MobiSys Int. Conf. on Mobile Systems, Applications and Services*. The USENIX Association, 2003.

[9] Sergio Mascetti, Claudio Bettini, Dario Freni, X. Sean Wang, and Sushil Jajodia. Privacy-aware proximity based services. In *Proc. of the 10th Int. Conf. on Mobile Data Management*, MDM. IEEE, 2009.

[10] Sergio Mascetti, Dario Freni, Claudio Bettini, X. Sean Wang, and Sushil Jajodia. Privacy in geo-social networks: proximity notification with untrusted service providers and curious buddies. *VLDB Journal*, 2011.

[11] Mohamed F. Mokbel, Chi-Yin Chow, and Walid G. Aref. The new casper: Query processing for location services without compromising privacy. In *Proc. of the 32nd VLDB international conference on Very large data bases*. VLDB Endowment, 2006.

[12] Anna Monreale, Gennady Andrienko, Natalia Andrienko, Fosca Giannotti, Dino Pedreschi, Salvatore Rinzivillo, and Stefan Wrobel. Movement data anonymity through generalization. *Transactions on Data Privacy*, 2010.

[13] Dimitris Papadias, Panos Kalnis, Jun Zhang, and Yufei Tao. Efficient olap operations in spatial data warehouses. In *Proc. of the 7th SSTD Int. Symposium on Advances in Spatial and Temporal Databases*. Springer-Verlag, 2001.

[14] Laurynas Šikšnys, Jeppe Rishede Thomsen, Simonas Šaltenis, and Man Lung Yiu. Private and flexible proximity detection in mobile social networks. In *Proc. of the 11th MDM Int. Conf. on Mobile Data Management*. IEEE, 2010.

[15] Roman Yarovoy and Francesco Bonchi. Anonymizing moving objects: how to hide a MOB in a crowd? In *Proc. of the 12th Int. Conf. on Extending Database Technology*. ACM, 2009.