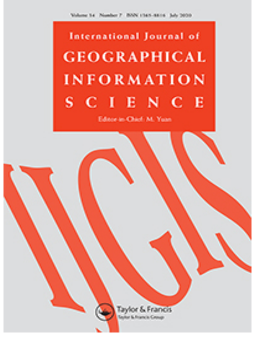

International Journal of Geographical Information Science

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tgis20

Obfuscating spatial point tracks with simulated crowding

Simon Scheider, Jiong Wang, Maarten Mol, Oliver Schmitz & Derek Karssenberg

To cite this article: Simon Scheider, Jiong Wang, Maarten Mol, Oliver Schmitz & Derek Karssenberg (2020) Obfuscating spatial point tracks with simulated crowding, International Journal of Geographical Information Science, 34:7, 1398-1427, DOI: 10.1080/13658816.2020.1712402

To link to this article: https://doi.org/10.1080/13658816.2020.1712402

Published online: 14 Jan 2020.

INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE
2020, VOL. 34, NO. 7, 1398–1427
https://doi.org/10.1080/13658816.2020.1712402

RESEARCH ARTICLE

OPEN ACCESS

Obfuscating spatial point tracks with simulated crowding

Simon Scheider[a], Jiong Wang[b], Maarten Mol[a], Oliver Schmitz[b] and Derek Karssenberg[b]

[a]Human Geography and Spatial Planning, Utrecht University, Utrecht, Netherlands; [b]Physical Geography, Utrecht University, Utrecht, Netherlands

ABSTRACT

Spatial point tracks are of concern for an increasing number of analysts studying spatial behaviour patterns and environmental effects. Take an epidemiologist studying the behaviour of cyclists and how their health is affected by the city's air quality. The accuracy of such analyses critically depends on the positional accuracy of the tracked points. This poses a serious privacy risk. Tracks easily reveal a person's identity since the places visited function as fingerprints. Current obfuscation-based privacy protection methods, however, mostly rely on point quality reduction, such as spatial cloaking, grid masking or random noise, and thus render an obfuscated track less useful for exposure assessment. We introduce *simulated crowding* as a point quality preserving obfuscation principle that is based on adding fake points. We suggest two crowding strategies based on *extending* and *masking* a track to defend against inference attacks. We test them across various attack strategies and compare them to state-of-the-art obfuscation techniques both in terms of information loss and attack resilience. Results indicate that simulated crowding provides high resilience against home attacks under constantly low information loss.

ARTICLE HISTORY
Received 1 May 2019
Accepted 3 January 2020

KEYWORDS
Geoprivacy; obfuscation; point quality preservation; exposure assessment

1. Introduction

Spatio-temporal tracking affords measurements of spatial behaviour patterns on an unprecedented level of detail (Shoval *et al*. 2014). This has recently spurred a wave of geographic health and epidemiological studies, targeting the environment's impact on the health of individuals or monitoring their health status (Curtis *et al*. 2011, Chaix *et al*. 2013). In such studies, there is an increasing need to share tracks on analytic platforms to enrich them with environmental context information.

This is possible since tracks are sequences of spatial points representing a time series of snapshots of a moving object (Hu *et al*. 2013). The locations recorded in such a track, however, serve as fingerprints (Krumm 2007) which can be used against the interest of the tracked person. An adversary can exploit tracks for potentially illegal acts, such as helping burglars to better plan their crimes. Even legal activities can turn out problematic, e.g. when health insurance companies monitor behaviour and adjust their rates to the

CONTACT Simon Scheider s.scheider@uu.nl
Supplementary data for this article can be accessed here.

detriment of the tracked person (Weiser and Scheider 2014). *Geoprivacy* is, therefore, a problem of increasing societal relevance (Keßler and McKenzie 2018).

Current *obfuscation* methods that were designed to anonymize tracks mainly build on the reduction of point quality, for example, by reducing the precision or accuracy of spatial points (cf. Section 3 and Krumm (2009)). Yet, it is precisely these details contained in a track which are required for environmental health analytics. Take for example Figure 1, which shows an environmental air pollution raster used to assess the exposure of cyclists to NO_2 in Oss, Netherlands (Schmitz *et al*. 2019). It can be seen that NO_2 concentration peaks locally and changes over small distances along road features, which requires any reasonable health-related assessment to be of high spatial detail. Practically useful methods for privacy protection of geo-analytic tracks, therefore, need to *retain point quality*.

How could a point quality preserving obfuscation technique look like, and how does it score in comparison to standard methods, both concerning security and usefulness for analytic purposes?

As an analogy, consider how people can easily hide in a crowd because our eyes have difficulties distinguishing many details on a small spot (Figure 2). In this way, we can become anonymous, even though we are not hiding our locational or personal details from an observer. The effectiveness with which we can hide in a crowd, and thus our anonymity

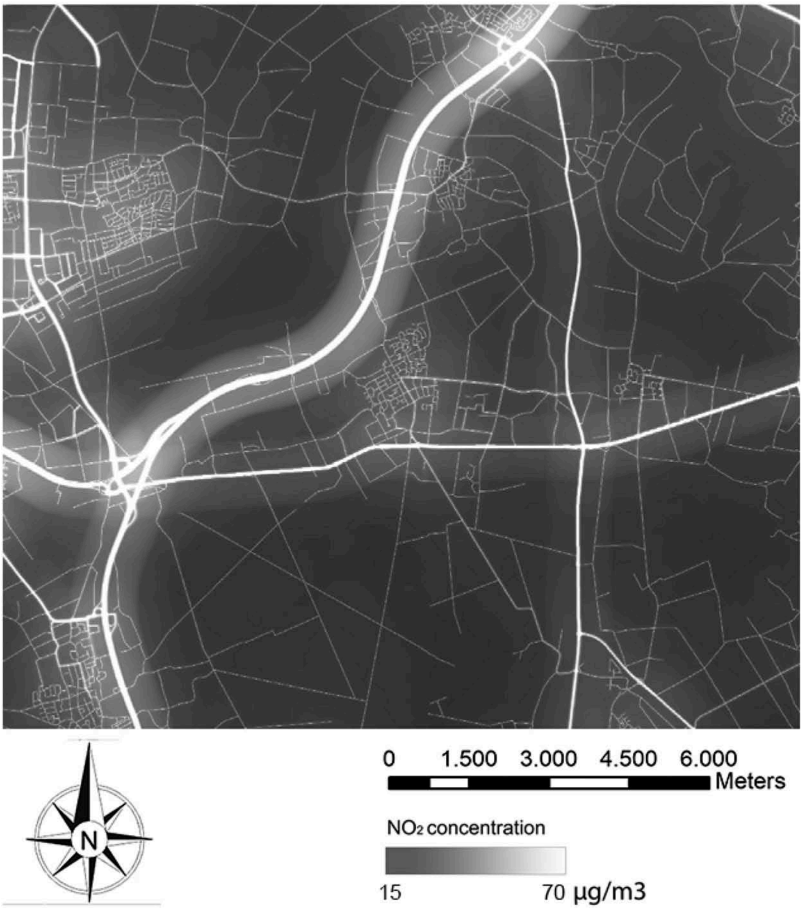

Figure 1. Air quality in Oss, Netherlands.

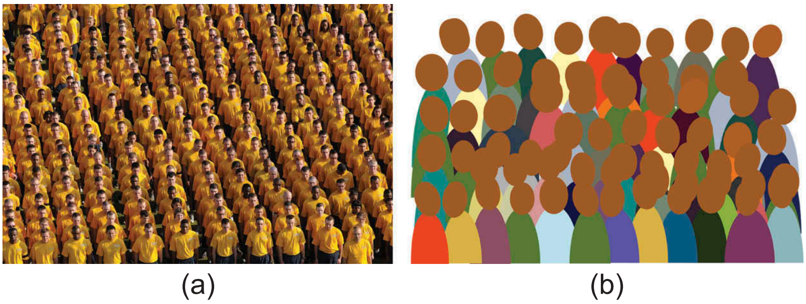

Figure 2. Strategies of hiding in the crowd. In (a), the crowd imitates a person. In (b), the crowd is diverse but lacking a discriminating feature that would allow us to pick out the right person.

level, directly depends on how easily we can be confused with the rest. We explore crowding principles that are based on two approaches: 1) *Mimicking* a person's look or behaviour to become indistinguishable from the target person. In Figure 2(a), this is illustrated by a crowd wearing identical clothes. 2) *Masking* the look or behaviour of a person. Here, looks or behaviours can be different, as long as persons in the crowd are equally credible candidates for an adversary (Figure 2(b)), and thus can be confused with each other.

In this article, we propose *simulated crowding* as an obfuscation technique which differs from existing approaches by hiding tracks in extensions which emulate credible behaviour. In this way, we do not trade point quality for an increased level of protection, but rather increase the *analytic complexity and computational effort* required to filter tracks. As we will illustrate in Section 3, the idea of dummy locations goes back to the early times of research on location privacy but has so far neither been exploited fully nor tested and compared systematically. We will demonstrate in this article that simulated crowding is a very resilient obfuscation strategy capable of retaining the quality of point coordinates yet at the expense of computation time and the quality of summary statistics over tracks. In the remainder, we first explain our motivating scenario in greater detail (Section 2), before we introduce crowding as a particular point quality preserving obfuscation method (Section 3). We then suggest crowding principles as well as simple algorithmic realizations in Section 4. We present our evaluation strategies in Section 5 to compare crowding against standard obfuscation methods on a sample of cycling tracks. Finally, we measure information loss as well as the risk against attacks (Section 6), before we conclude.

2. Exposure services and tracks: use case and requirements

Traditionally, most geoprivacy studies have been focusing on Location-Based Services (LBS), where *online queries* of a user need to be protected every time the client requests information from the LBS and therefore shares its location coordinates (Ghinita 2009). For example, to find nearby restaurants, a smartphone needs to send its current location to a place server. Since LBS are usually less sensitive to location error, point quality reduction strategies such as imprecision or inaccuracy can be used to obfuscate a point (see next section).

In *offline trajectory sharing*, in contrast, entire records of location measurements of a set of persons are submitted to an analytic service, which often requires high-quality data (see Figure 3). Consider an epidemiologist who requires personal exposures to NO_2 for

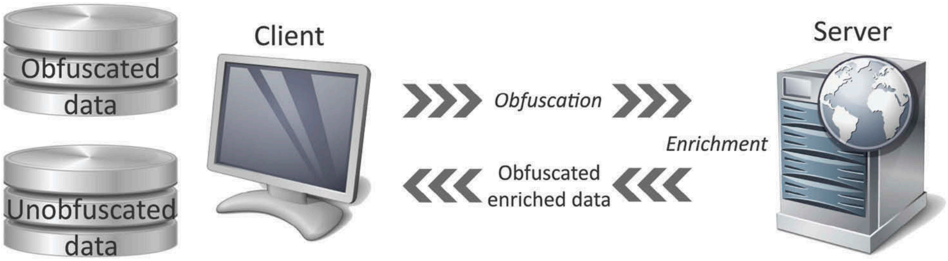

Figure 3. Illustrating the need of obfuscating tracks before sending them to an enrichment server.

each person in a cohort,[1] and who lacks the required analytic capabilities or the necessary environmental data sources. The researcher has collected trajectories of the people in the cohort and wants to make use of an external analytic service which is used by many other researchers as well. Because of privacy regulations,[2] it is not allowed to share these tracks directly with such a service, since the researcher cannot control the conditions of data access on the remote server. While a *secure trusted linkage service* might guarantee location privacy (Rodgers *et al*. 2012), it also restricts access to both the data and the service by first requiring the expensive establishment of a trust relationship between the service and every user individually. Instead, we consider the possibility of obfuscating the information provided by many researchers for their own kinds of data. For this purpose, a researcher locally obfuscates tracks before sending them to the enrichment service (Figure 3), to minimize the danger of data leakage. We assume he or she only shares obfuscated spatial point coordinates with the server, to obtain either of two enriched variants:

(1) Pointwise exposures
(2) Aggregated exposures over a track

During the entire process, the original track stays at the client computer, and only the obfuscated and enriched points are transferred between client and remote servers. Besides, the client stores a mapping from the original track into the obfuscated counterpart for track recovery and further analysis of enriched information (Figure 4).

Obfuscation needs to be fully automatized so that users without a technical background can make use of it. Since enrichment should be as quick as possible, the amount of data sent to the server should be minimized. Also, both the risk of privacy attacks and the error in assessing exposure (information loss) should be minimized. What would be the most suitable obfuscation strategy for this scenario?

3. Strategies for anonymizing point tracks using obfuscation

Anonymity is commonly defined as the state of a person ‘not being identifiable within a set of subjects’ (Pfitzmann and Köhntopp 2001). In the digital world, this effectively means that information about an identifiable person cannot be matched with the unique record of this person in a database that contains privacy-sensitive information. The latter might contain spatial data, e.g. a spatial trajectory which indirectly tells us

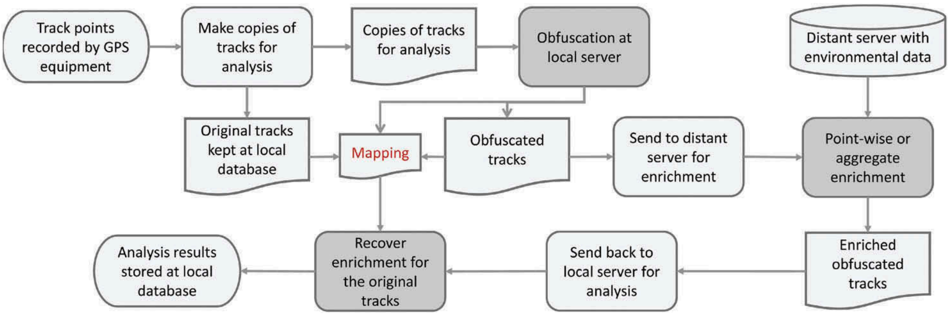
Track points recorded by GPS equipment
Make copies of tracks for analysis
Copies of tracks for analysis
Obfuscation at local server
Distant server with environmental data
Original tracks kept at local database
Mapping
Obfuscated tracks
Send to distant server for enrichment
Point-wise or aggregate enrichment
Analysis results stored at local database
Recover enrichment for the original tracks
Send back to local server for analysis
Enriched obfuscated tracks

Figure 4. Workflow of obfuscation used in remote environmental exposure analysis. Round rectangles are processes, and dark grey ones are automated.

about where the person lives, works and what habits the person has. If this record cannot be associated with the right person, then the tracked person stays 'anonymous'. This state of anonymity is usually called *location privacy* (Krumm 2009, Chow and Mokbel 2011).[3]

The anonymity of a given trajectory may be defended in several ways. The simplest one, stripping away identifiers and replacing names with a pseudonym (called *pseudonymity* by Pfitzmann and Köhntopp (2001)), is known to be very ineffective. This is not only because spatial coordinates alone are easily reverse-geocoded to obtain addresses (Chow and Mokbel 2011, Zandbergen 2014), but also because human space-time trajectories are astonishingly unique. As De Montjoye *et al.* (2013) have shown, only 4 randomly selected points of a trajectory of a person need to be known by an adversary to be able to uniquely identify 95% of all persons amongst 1.5 million users. Furthermore, spatial data can be used to derive all kinds of secondary information that was never intended to be shared (Keßler and McKenzie 2018), and anonymity can be nowadays breached in combination with an unforeseeable amount of external data sources that help de-anonymize a trajectory (Weiser and Scheider 2014). Other measures, such as data encryption, access control and data retention are often impractical and prevent useful services from which a user might benefit (Keßler and McKenzie 2018). All this makes geoprivacy a kind of problem that might not be solvable by technical means alone, requiring also other forms of behavioural and societal control (Weiser and Scheider 2014, Keßler and McKenzie 2018).

To render trajectory data anonymous, researchers have proposed various techniques (see Table 1) which are based on reducing its *geodata quality* in one way or another, and which are summarized under the term *spatial obfuscation* (Duckham and Kulik 2005). These techniques can be ordered along the various dimensions of imperfection within spatial information (c.f. Duckham and Kulik (2005), Veregin (1999)):

- *Imprecision*: By reducing the resolution of data items. This refers to the lack of specificity. An example is generating a 100-meter resolution spatial grid from a 10-meter resolution grid, or clustering points to a container region.
- *Inaccuracy*: By reducing the accuracy of data items. Accuracy refers to the lack of correspondence with some ground truth. This is what we normally call an "error".
- *Vagueness*: By reducing the interpretability of a data item. This refers to the uncertainty of whether a given georeference refers to a given coordinate location or not.

Table 1. Overview of track obfuscation strategies based on dimensions of geodata imperfection. The strategies in bold are more closely studied in this article.

Track obfuscation strategies								
Point quality reduction strategies					Track quality reduction strategies			
Imprecision based		Inaccuracy based		Vagueness	Incompleteness based		Crowding	
Spatial Cloaking	Temporal Cloaking	Translation	Perturbation	Cognitive referencing	Sampling	Splitting	Merging	Simulation

For example, when using "near the city centre" or "downtown" instead of a postcode area to refer to some location (Bennett 2010).

- *Incompleteness*: By reducing the number of items in a dataset representing a specified phenomenon. This refers to the uncertainty of what in some domain is represented in a data set and what is not. For example, when leaving out moments of a trajectory due to decreasing sampling rates, or when mapping not all houses in a city (Veregin 1999).

In contrast to existing obfuscation approaches, crowding is a strategy that is closely related to incompleteness, as it concerns the quality of *entire tracks* and not *individual data points*.

3.1. Point quality reduction strategies

Strategies based on reducing the quality of points are among the most frequent ways to obfuscate a track (Duckham and Kulik 2005, Chow and Mokbel 2011).

Imprecision strategies include aggregation to coarser radio cells (De Montoye *et al.* 2013) unique, as well as spatial and temporal cloaking (Gruteser and Grunwald 2003, Gedik and Liu 2004, Xu and Cai 2007). The spatial or temporal 'cloaks' can be defined based on intervals, by removing the precision of coordinates, or by coarser spatial objects, provided that enough other persons are contained in these regions so that persons are confused with each other (Chow and Mokbel 2011). However, according to the results in De Montjoye *et al.* (2013), Zang and Bolot (2011), such resolution based approaches are not very effective in anonymizing tracks because they still allow identifying users uniquely. For the identification of homes in health science, the user study by Curtis *et al.* (2011) shows that the strategy might be effective, though. However, lower resolution renders geodata often unusable for analysing spatial patterns, such as disease risk (Kwan *et al.* 2004).

Inaccuracy based strategies introduce a limited positional error to each point in a track. When applied to pure point data, this strategy is often called *geographic masking* (Zandbergen 2014), a term introduced by Armstrong *et al.* (1999). Standard methods involve *affine transformation* (Kwan *et al.* 2004), *grid masking* (moving a point to a defined grid) and *Voronoi masking* (to a Voronoi polygon border), as well as (weighted) *random perturbation* (Seidl *et al.* 2015, Kounadi and Leitner 2016). A recent masking method called *location swapping* transforms points to nearby locations with similar geographic characteristics (Zhang *et al.* 2017). Studies on geographic masking usually investigate how useful data remains for analysis under limited distortion.

Strategies based on *vagueness* substitute measurable coordinates with cognitive spatial references (such as 'near' or 'in front') or places whose extension is vague, such as 'mall', 'school', 'home' (Duckham and Kulik 2006, Krumm 2007). This strategy is easily understood because it corresponds to the way how people normally communicate about space, yet difficult to exploit for computation (Scheider *et al*. 2018).

The big disadvantage of all point-based strategies lies in those analytic tasks that require precise/accurate references. While random perturbation may hardly affect statistical summary measures, a point which is moved by only 30-meters beyond a street border will be significantly less exposed to air pollution.

3.2. Track quality reduction strategies

The following strategies do not change the quality of individual point data items but rather affect the quality of entire datasets.

One simple possibility is to *reduce the completeness* of data for a trajectory. For example, Hoh *et al*. (2006) found that reducing the sample interval of a track from one minute to four minutes reduced the home identification rate from 85% to 40%. Similarly, a track can be simply cut in half and may, therefore, hide the home location.

Another option is to *add data instances* beyond the points of a given trajectory, which we suppose to call *crowding*. The principle idea of using a crowd for anonymization was first developed in Web communication with servers (Reiter and Rubin 1999), where a group of users of a server hide behind a crowd protocol which prevents the server from storing their identity. However, the idea is also reflected in some recent geodata anonymization methods.

For example, Kido *et al*. (2005), Kido (2006) included false position data of moving dummy objects in LBS requests, so that the LBS cannot distinguish between true points and the points sent by the user. Similarly, Nussbaum *et al*. (2017) recently proposed *i, j-anonymity*, which assures that through generating fake points, under a realistic movement constraint, every point is sure to be a successor of at least *i* other points and a predecessor of at least *j* other points. In this way, many different realistic tracks are synthesized, among which the user trajectory can effectively hide.

Crowding based anonymization strategies, however, have not been a subject of research as such. The few mentioned studies above have neither investigated the effectiveness of this strategy against various kinds of attacks, nor its effect on the usefulness of data for spatial analysis. While the strategy leaves the quality of individual points intact, it considerably changes the distribution of points. We regard crowding, therefore, as complementary to the commonly used point quality reduction strategies, with clear advantages for track based exposure measurements.

3.3. Privacy metrics, attack strategies and prior knowledge of adversaries

To assess the *amount of privacy* that an obfuscation technique can achieve, it becomes necessary to consider *attack strategies* and the prior knowledge which can be employed by an adversary for hacking an obfuscated track.

Various researchers have proposed privacy metrics which could be used to give *minimum privacy guarantees* for an obfuscation method, based on making assumptions about an adversary's prior knowledge. This includes *k-anonymity* (Sweeney 2002), which aims at assuring that a person can be confused with at least k others, as well as *l-diversity*

(Machanavajjhala *et al*. 2008), which achieves k-anonymity based on spatial imprecision. These latter methods have been rightly criticised because they only incompletely model an adversary's prior information (Shokri *et al*. 2011), and instead rely on the assumption that the database of persons under comparison is somehow known. Shokri's privacy measure instead models an adversary's prior knowledge in terms of the probabilistic behaviour of an obfuscation algorithm, as well as a mobility profile of known users. The obfuscation behaviour is considered a function from tracks into a probability distribution over possible obfuscation results in terms of points or regions. An attack is an estimated inversion of this function, and the amount of privacy is measured as the distance of a corresponding reconstruction to the actual track. Similarly, Andrés *et al*. (2012) introduced *geo-indistinguishability*, which requires from a corresponding obfuscation function (assumed to be known by the adversary), that it produces, when applied to two points, probability distributions which differ only up to a positive factor of the spatial distance between these points.

While we believe that a corresponding privacy measure for crowding would be beneficial to provide minimum privacy guarantees and optimality results, it remains unclear how existing approaches could be employed for our purpose. The main problem is that the mentioned authors focus on obfuscation probability functions which map tracked locations to other locations or regions. Thus, in essence, they assume point quality reduction strategies, such as spatial cloaking, masking or perturbation. Yet, simulated crowding as supposed below is a track quality reduction strategy based on behavioural similarity, and, therefore, rather maps tracks to entire tracks without resorting to a probabilistic map of particular locations. It seems open how a distance metric (and more generally, *differential privacy*) could be employed for measuring privacy under these conditions.

In this article, we took a pragmatic approach and instead incorporated prior knowledge of an adversary into simulated crowding by taking into account *behavioural credibility constraints*. As simulated behaviour becomes indistinguishable from the observed behaviour, prior knowledge about plausible behaviour (concerning movement and environmental context) becomes useless as an attack strategy. We evaluated our simulation based on testing concrete attack strategies on track data, by comparing the error of reconstructing crowded tracks with the reconstruction error under inaccuracy based point quality reduction strategies.

4. Simulated crowding

In this section, we explain basic principles of our approach and introduce simple algorithms which implement them.[4] Note that the discussed principles would allow also for more complex and maybe more effective implementations, which is considered future work.

4.1. Rasterization

We start with a point quality reduction strategy. It consists in reducing the *precision*, and with it also the *concentration* of track points at places by rounding coordinates and snapping them to a regular grid. Although this imprecision strategy is not strictly required for simulated crowding and to some extent contradicts its philosophy, we will see that it makes the realization of crowding much simpler as it can be computed on discrete space. Furthermore, it helps to counter attack-strategies that are based on exploiting *point clusters*, i.e. high concentrations of points that may indicate places of interest, such as *home*.[5]

Algorithm 1: Rasterization algorithm

```
Rasterize (t = track, m = roundingincrement)
  Result: Map track points to points in a regular grid with a fixed cellsize
rastertrack = [];
lookup = [];
for v in t do
    /* For every point v in t, round X and Y coordinates         */
    o.X =Round (v.X/m)*m;
    o.Y =Round (v.Y/m)*m;
    /* List of rounded coordinates 1-to-1 with track              */
    lookup.append(o);
    /*Hashtable with rounded coordinates as keys                  */
    rastertrack[o]=None;
end
return lookup, rastertrack
```

To this end, we select a resolution level (a grid cell size) based on a *rounding increment m*, and subsequently map all points of the track to their nearest grid point, reducing the precision of coordinates accordingly (compare Algorithm 1). Since this mapping is *not injective*, close points will be mapped to identical grid points, and point clusters with high concentrations of points at distances below the cell size will be resolved. The choice of this rounding parameter *m* depends on other parameters of the entire method and will be explained later (see Section 4.4).

To be able to reverse this mapping later after environmental enrichments were done, and thus to recover the location of original points, we use rounded space/time references as keys for retrieval and store them in a list (*lookup*). This list is 1-to-1 and stored together with the original track. To write back enrichment results, the lookup list can be traversed and its gridded elements can be later used to retrieve enrichment results from a hash table with rounded coordinates as keys, which represents the rasterized track (*rastertrack*).

4.2. Track extension strategy

One strategy of crowding is based on mimicking the behaviour of the tracked object, e.g. by extending tracks. In this approach, privacy attacks are misled to fake locations where simulated behaviour is displayed, but no actual behaviour ever occurred in reality. For this purpose, we need a way to capture tracked behaviour and to simulate points accordingly.

4.2.1. Movement and location behaviour

We suggest to measure tracked behaviour with respect to two dimensions. The first is the *behaviour of movement*. Incoherent movement behaviour can be used to attack an obfuscated track by distinguishing different patterns of movement within a track. We, therefore, suggest to capture movement behaviour with a probability distribution over (discrete) *relative vectors* $V = [v_1, \ldots v_n]$ which map the possible space of movement steps. Each relative vector v_i stands for a possible kind of movement as observed in the track, measured from a given point to its successor[6]:

$$getV(track) = \{v \text{ for } v_i, v_{i+1} \text{ in } track \text{ if } v_{i+1} - v_i == v\}$$

Each v is further associated with the *probability that this kind of movement actually occurs* in the track, based on the frequency distribution of relative vectors of direct successors across the track:

$$P_{track}(v) = \frac{\| \{v_i \text{ for } v_i, v_{i+1} \text{ in } track \text{ if } v_{i+1} - v_i == v\} \|}{\| track \| - 1}$$

The denominator denotes the number of successor points in the track, and the numerator counts the number of successor points that correspond to a given type of movement v. The probability that a movement from any point v_1 to another point v_2 corresponds to the movement behaviour observed in a *track* is then $P_{track}(v_2 - v_1)$. An example for a probability over discrete vector space is given in Figure 5, where the track tends to move towards the North East/South West axis.

To compare the behaviour of a simulated track with an original one, and to assure a minimal behaviour congruence, we also measure the *similarity* of tracks concerning movement behaviour. One way of doing this would be to compute the *cosine similarity* over the set of movement probabilities of relative vectors occurring in both tracks. Yet, this simplistic measure does not account for the *sample size* of tracks. More precisely, it does not account for the fact that small sample tracks statistically show a much more diverse behaviour pattern than large sample tracks, and thus deserve a more tolerant similarity judgment. We chose therefore to estimate similarity by a two-sample (Pearson's Chi-squared) χ^2 significance test, where H_0 is the

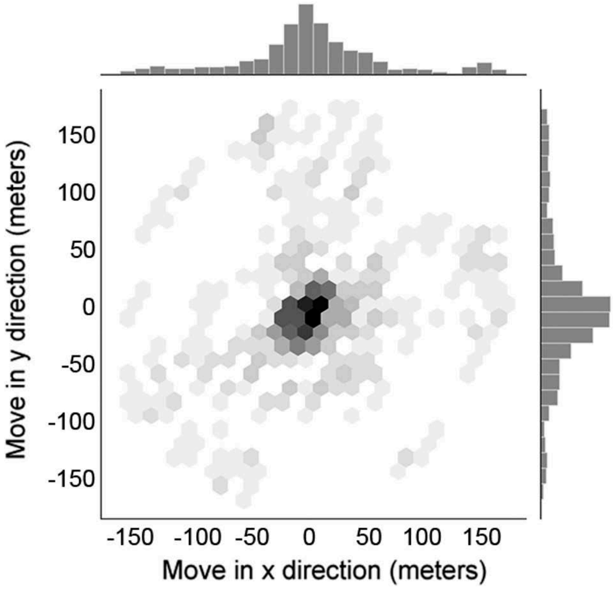

Figure 5. Distribution of movement behaviour in 2-dimensional vector space involving x and y direction components.

hypothesis that both tracks have a similar distribution over relative vectors V (where V is the union of relative vectors occurring in both tracks), and two tracks are considered sufficiently similar as long as the p-value of the test statistic is above a given significance threshold (such as > 0.05). This test statistic depends on a track's size, and p_{χ^2} returns its probability according to the χ^2 distribution:

$$sim^{move}_{\chi^2}(track_A, track_B) = 1 - p_{\chi^2}\left(\| \, track_A \, \| \sum_{v_i}^{\|V\|} \frac{\left(P_{track_A}(v_i) - P_{track_B}(v_i)\right)^2}{P_{track_B}(v_i)} \right)$$

The second behaviour dimension is with respect to the *location* of track points, referring to the types of spatial infrastructure that support movement. For example, when moving with a car, a track is bound to stick with a road, as opposed to the case when a track rather measures a pedestrian, who might not care at all about the surface quality. One might attack an obfuscated track therefore based on incoherent infrastructure categories, excluding points that indicate e.g. a movement with car speed across a lawn. We measure infrastructure with *land use classes*, using a high resolution land use polygon file, buffering the track to account for location errors (using a 50 meter buffer radius), and then spatially intersecting the buffer with land use polygons. The *percentage of the area for each land use class* within this buffer then gives us a probability for encountering a given land use class within the range of the track. Suppose the function *luse* retrieves the land use class for a given location. Then P^{luse}_{track} is the probability for encountering class *i* within the track:

$$P^{luse}_{track}(i) = \frac{\sum area(p) \text{ for } p \text{ in } inters(buffer(track), land\,use) \text{ if } luse(p) == i}{area(buffer(track))}$$

To compare simulated tracks with original ones, we measure location similarity of two tracks using a χ^2 distribution to account for sample size:

$$sim^{luse}_{\chi^2}(track_A, track_B) = 1 - p_{\chi^2}\left(\| \, track_A \, \| \sum_{v_i}^{\|V\|} \frac{\left(P^{luse}_{track_A}(luse(v_i)) - P^{luse}_{track_B}(luse(v_i))\right)^2}{P^{luse}_{track_B}(luse(v_i))} \right)$$

To measure probability over both dimensions for ordering fake point candidates, we treat both kinds of probabilities as if they were independent, combining them by their product $P^{luse}_{track}(luse(v)) * P_{track}(v)$.

4.2.2. Random simulation

The second problem concerns the realistic simulation of a track. We suggest a heuristic search strategy for extending tracks based on 3 iterative steps. For each track:

(1) Select a point of the track and find a set of (rasterized) point candidates in a predefined spatial neighbourhood of this point, defined by the observed movement behaviour.
(2) Add a candidate fake point to the track based on its movement and location probability with respect to the original tracked behaviour, as defined above.
(3) Check whether original track and fake track are sufficiently similar with respect to the proposed probabilistic measures. If not, remove point and go to 2, otherwise go to 1 and start from the newly added point.

We applied the proposed simulation strategy iteratively until a randomly chosen number of fake points was generated (see Algorithm 2, lines 2–5). The random number lies within 0 and a maximum size given as *extension parameter e*. For instance, with e = 0.8, we generate 80% of the number of track points (rdm([0 : 0.8$*$ || *track* ||])). Regarding *step 1* above, we use a neighbourhood defined by the discrete vector space on movement behaviour defined in the last section (*getV*). So, possible candidates are points that are reachable under the behaviour observed in the original track (Algorithm 2, line 3). The selection of candidates (*step 2* above) (Algorithm 2 line 10) is done with a frequency based on the combined location and movement probability of the fake point candidates with respect to the track (Algorithm 2 lines 6–8). In that way, we prefer points showing a behaviour as in the original track. Since this does not guarantee a minimum level of similarity, in the *third step*, we compare the extended track with its non-extended (initial) version (Algorithm 2, line 12). If the algorithm does not find a sufficiently similar candidate, this means that an error is thrown and the search strategy needs to be changed, e.g. by changing the neighbourhood or the significance level.

Algorithm 2: Mimicking algorithm

```
1  Mimic(t = track, p = significance, e = extension)
   Result: Randomly extend track based on movement/location similarity
2  $v_0$ = getEnd(track);
3  V = getV(track);
4  faketrack = track;
5  for i in [0:rdm([0 : $e*$ || track ||])] do
6      for v in V do
7          P($v_0 + v$) = $P_{track}(v_0 + v) * P^{luse}_{track}(luse(v_0 + v))$
           /*construct probability over candidates                    */
8      end
9      for v in V do
10         candidate = randomchoice(P);
11         test = faketrack.append(candidate)
12         if candidate $\notin$ faketrack and $sim^{move}_{\chi^2}(test, track) > (1 - p)$ and
             $sim^{luse}_{\chi^2}(test, track) > (1 - p)$ then
13             faketrack = test;
14             $v_0$ = candidate;
15             error = False;
16             break ;
17         else
18             error = True
19         end
20     end
21     if error then
22         return ERROR: No sufficiently similar candidate found!
23     end
24 end
25 return faketrack
```

4.3. Track masking strategy

Another crowding strategy is called masking. Here, we also simulate fake points, yet not in a way that resembles the original track, and thus adds more variety to the observable spatial patterns. The generated points can fundamentally change the geometric 'face' and spatial pattern of the point cloud. As long as they are as *credible* as the original track points, an adversary has no way of deciding which are the correct ones. A strategy for successful masking, therefore, needs to account for 1) generating a variety of behaviour in a crowd, and for 2) assuring credibility, so that fake points cannot be discovered as such based on implausible behaviour.

The basic idea of generating variety is the use of *templates*. A template is a spatial (or, more generally, spatio-temporal) configuration of point vectors which can be added to a single point, such that new points are generated in a neighbourhood defined by the template. A template thus corresponds to a *reference frame*, in which one of the vectors plays the role of the *origin*, and all other vectors are expressed relative to this origin. The direction of the main axis corresponds to the main axis (Y) in the spatial reference system used for the track. Since we have chosen to rasterize the track, our masking strategy is built on this raster. The units on our axes, therefore, correspond to the rounding increment of our chosen rasterization.[7]

While the form of the template may be arbitrary, candidate forms can be found among common lattice neighbourhoods, such as a *von Neumann* neighbourhood or a *Moore* neighbourhood, see Figure 6.

Suppose a von Neumann neighbourhood template with radius r is given in terms of a list of 2-dim vectors in discrete cell space $\mathbb{Z}$ (corresponding to rounding increment m), where the origin $v_0 = (0, 0)$ lies in the middle (black cells in Figure 6):

$$Template_r^{vN} = \{(x, y) \in \mathbb{Z}^2 | x + y \leq r\}$$

Then the size of this template (number of points minus the origin) depends on the radius, and is given by[8]:

$$size^{vN}(r) = \frac{r(r+1)}{2} * 4$$

However, using such a template in its raw form is not a good masking strategy. The reason is that whenever one uses a fixed template for masking, the rigid spatial configuration of points is repeated across the track and becomes recognizable. Therefore, it becomes possible to guess the template's origin. For example, suppose we mask a track

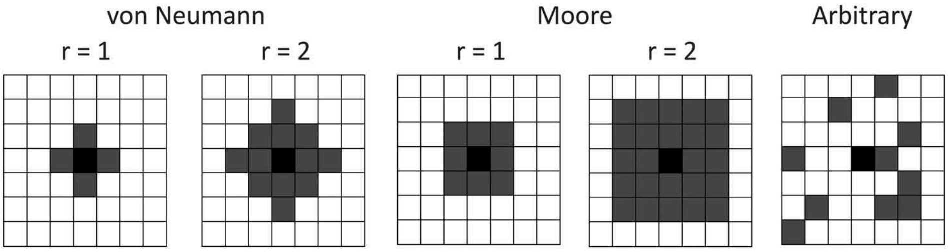

Figure 6. Lattice neighbourhoods.

(Figure 7) with a von Neumann neighbourhhood of radius 2, then the result might look as in Figure 8.

Due to the repeating form and fixed centre of origin of the template, one simply needs to compute the centre of gravity for each recognizable neighbourhood to reconstruct the original track.

To defend against such attacks, we suggest to randomly variate the template's origin. In this way, it becomes difficult to estimate the template origin just based on the form of the point crowd (Figure 9). The centre of mass e.g. yields a largely distorted version of the track (its error depending on the radius of the neighbourhood), and thus would any strategy based on the form of the template. Randomly transforming a template concerning its origin is defined in terms of vector subtraction from its origin, where *rdm* is a random function that selects an element of a template with a linearly increasing probability from its geometric centre:

$$rdmshift(template) = \{(v_i - v_0) \text{ for } v_i \in template \text{ if } v_0 = rdm(template)\}$$

The new template now has its origin at $v_0 = (0, 0)$ which is likely not in the geometric centre (Figure 9). Additionally, we could transform the template by rotating and by scaling

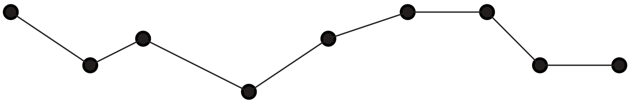

Figure 7. A rasterized sample track.

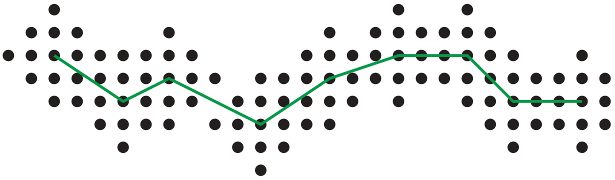

Figure 8. Masking the track with a von Neumann neighbourhood template of radius 2. The centre of mass attack (line) easily yields the original track.

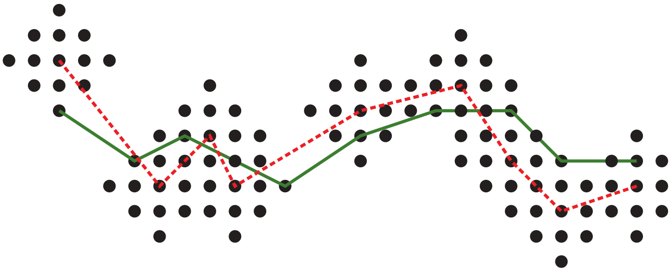

Figure 9. Masking by randomly transforming the origin of the template. The centre of mass now yields a largely wrong track (dotted line).

it with a factor, however, this was not done in this article. *Applying* this template to a given track coordinate point $v_{geo} = (x, y)$ in a georeference system needs to take into account the rounding increment m used in rasterization (to scale the template to the cell size), which can be done by the vector sum:

$$apply(v_{geo}, template, m) = \{(v_{geo} + m * v_i) \text{ for } v_i \in template\}$$

If we do this for each point in a track (see Algorithm 3 lines 5 and 6), then the maximum number of possible alternative tracks (assuming no overlaps) amounts to k^n, where n is the number of track points and k is the number of points of the template.

Algorithm 3 : Masking algorithm

```
1  Masking (t = track, m = increment, r = radius)
     Result: Masking track points with a neighbourhood of credible points
2  out = [];
3  Pred = [];
4  $template_r^{vN}$ = $[v_0, v_1, ...]$;
5  for v in t do
6      temp = apply(v, rdmshift($template_r^{vN}$), m)
       /* Randomly shift template                                   */
7      candidates = [v' for v' in temp if v' $\notin$ Pred]
       /* Select fake point candidates                              */
8      mask=candidates.append(v)
9      out.extend(mask)
10     Pred = mask
11 end
12 return out
```

Neighbourhoods, however, can easily overlap (see Figure 9). Whenever this is the case, they must share points, and thus the number of points that can exclusively be confused with a single track point becomes less than *k*. For this reason, we only consider template points as fake candidates which are not yet in the predecessor mask (*Pred*) (Algorithm 3 line 7).

More sophisticated approaches to ensure credibility of point masks could be based on corresponding *movement or location constraints*. If two points in successive crowding neighbourhoods are located beyond a threshold reasonable for realistic movement, i.e. if one is not 'reachable' from the other, then they can be easily excluded as endpoints of an edge in the track. Solutions to this problem called *(i,j)privacy* were proposed by Nussbaum *et al.* (2017) and can easily be added to our crowding approach, but were not considered in this article.

4.4. *Putting the pieces together*

The different algorithms proposed above need to be integrated into a single crowding solution. Since their parameters depend on each other, we need to find a robust way of integrating them.

For example, rasterization including the rounding increment *m* as well as the template radius *r* and the number *k* of template points should be chosen with respect to a credible distance (*d*) between track points, otherwise the rasterized points, as well as the masks, will become too coarse for representing the track or too detailed to be effective in hiding it. This credible distance should, therefore, be based on the distances between points of the track. We propose to automatically determine the parameters *d, m*, and *r* based on *k* and the maximum distance measured between successor points in a track, as outlined in the appendix. The only remaining input parameter is *k*, plus the significance level *p* and the extension parameter *e* needed for mimicking.

Finally, the order of method applications is important. Track extension changes the available points for masking, and masking would fundamentally change the spatial patterns available to mimick a track. We therefore suggest to apply track extension *before* masking. Consequently, we first rasterize a track, then extend the track, and finally apply the masking strategy (see Algorithm 4).

```
Algorithm 4: Simulated crowding algorithm

Crowding ($t$ = $track$, $k$ = $template\ size$, $p$ = $significance$, $e$ = $extension$)
  Result: Simulated crowding of a track
  $d = \mathrm{param}_d(t)$;
  $m = \mathrm{param}_m(d, k)$;
  $r = \mathrm{param}_r(k)$;
  $rastert$ = Rasterize($t, m$) ;
  $extendedt$ = Mimic ($rastert, p, e$) ;
  $maskedt$ = Masking ($extendedt, m, r$) ;
  return $maskedt$
```

5. Evaluation methods

In this section, we introduce the methods to evaluate simulated crowding. We suggest that a robust privacy evaluation needs to take into account at least two aspects.[9]

One is related to the idea that every obfuscation method negatively affects the quality of the data (cf. Section 3), and therefore also lowers its *usefulness* for a certain *analytic purpose*. For this reason, obfuscation always includes a certain loss of information. We measure this loss by comparing the effect of different obfuscation methods (including crowding) on track analysis results. Furthermore, to assess the increased effort of analysing tracks due to obfuscation, we also compared the computing time needed to enrich a given track with this kind of analysis.

The second aspect is related to the *effectiveness* of obfuscation concerning the resilience against attacks. For this purpose, we will compare obfuscation methods based on how effectively a given attack strategy can reconstruct the original track. We also assessed how k-anonymity is influenced by obfuscation within the test set of tracks.

We start by discussing our benchmark for testing and then introduce the measures we took to address the two quality aspects.

5.1. Comparing crowding against other obfuscation strategies

To benchmark crowding, we compared it against two simple point quality reduction strategies that belong to the standard repertoire for obfuscation of health data (Seidl *et al*. 2015) (see Section 3). One is *Random Perturbation*, where points in a track are redistributed according to some probability distribution over space. We randomly sampled new points from a two-dimensional Gaussian Density Kernel estimated over the 6 nearest neighbours for every point in the track. The second standard method we used is *Voronoi Masking*. For this purpose, we computed the Voronoi polygons for a track and then projected each point onto the nearest Voronoi line (Seidl *et al*. 2015).

Benchmarking was done by computing, for each obfuscation method, the information loss of an obfuscated track, as well as the effectiveness of attacks compared to the original track, and averaging these out over all tracks.

5.2. Measuring information loss

Computational privacy studies usually assess the quality of their methods using theoretical or empirical anonymity measures (cf. Chow and Mokbel (2011)). However, information loss or usefulness of an obfuscated track is commonly not considered. Usefulness depends on the purpose of tracking analysis, which is application dependent and thus cannot be answered in general. However, purpose is central to applied information science, and therefore often appears in corresponding studies about practical accuracy decreasing methods (cf. Section 3). Geographic masking, e.g. is evaluated based on how strongly standard spatial data analysis is affected by added noise (Zandbergen 2014, Seidl *et al*. 2018).

For our purpose, we measured information loss in terms of the error in air pollution exposure. We averaged NO_2 concentration values in a buffer around the track line using zonal aggregation. For each obfuscated track, we then computed the percentage error in average concentration with respect to the original track.

5.3. Effectiveness of privacy attacks

Many studies are centreed around the effectiveness of anonymization. Most often, this is assessed by proving that a method assures a minimal abstract anonymity level (see e.g. Nussbaum *et al*. (2017), Zang and Bolot (2011), Sweeney (2002)). However, as we argued in Section 3, we cannot control the external information environment of an adversary, and therefore such theoretic measures do not tell us much about the true risk. We therefore measured effectiveness empirically, by testing resilience against various kinds of attacks, as was done in Krumm (2007), Curry (1999), Fechner and Kray (2012), Raubal (2011).

In principle, obfuscated tracks can be attacked in several ways. One possibility is to exploit realistic *movement constraints* to detect erroneous points that would require unrealistic movements (Nussbaum *et al*. 2017). Another one is based on exploiting the *geometry* of a track, to distinguish realistic points based on the form of the point cloud. For example, a simple attack strategy exploits the point density to detect *home* (Krumm 2007). Finally, it is also possible to detect erroneous points based on assessing the *spatial context* of a point, using e.g. land use patterns or other geographic data. In this article, we used attacks based on the latter two approaches, because we consider our design less vulnerable to movement constraint attacks, as it takes movement patterns into account. Our implementation of both attack strategies first involved *spatial smoothing* of a track using a moving average window of size 3. Thereafter we proceeded as follows.

5.3.1. Effectiveness of reconstructing homes

First, we implemented two simple attack strategies for reconstructing stopping places and homes from (obfuscated) tracks. Being able to detect homes is among the most vulnerable information contained in a track because it allows opponents to link the track with an address to identify the person. Note that other stop locations might be equally dangerous, as they reveal a person's habits. To reconstruct home, we computed the residential land use density within a 30-meter buffer around each point in a track, as well as the point density. We then selected the point where the product of both was the highest. To measure the effectiveness of these attack strategies, we computed the Euclidean distances between home estimated as above and true home.

5.3.2. Effectiveness of reconstructing tracks

Second, we implemented two different attack strategies to assess the effectiveness of reconstructing entire tracks. One is simply based on smoothing the obfuscated track and measuring the similarity to the original track based on (50 meter-) buffering tracks and computing the *areal overlap* in terms of the *Jaccard index* (which is the ratio of the intersection and the union of both buffered tracks). The other is based on estimating the main directions of a track and the spread along the elongated track using the *Standard Deviational Ellipse* (SDE). SDE determines the 2-dim spatial angle of the major axis (θ) of the shape of a point cloud, as well as the standard deviations along the major (x) and the minor axis (y) of the ellipse.

To measure the effectiveness of these attack strategies, we directly used the Jaccard index for the first strategy and computed the differences of the SDE parameters (θ, x, y) between each obfuscated and the original track for the second one.

5.4. Test set up

The evaluation was computed on a sample of tracks drawn randomly from a large set of bicycle tracks obtained from a panel of 581 smartphone users, gathered in a large bicycle stimulation program throughout the year 2014 ('B-riders'[10]) in Noord Brabant, a southern province of the Netherlands (de Kruijf 2014). The dataset was enriched with information about individual trips and journeys, together with a classification model of types of stop places (home, work, leisure) based on the method in Feng and Timmermans (2017). We focused on trip sections in January, July and December 2014 that lead from or to home locations so that we can examine obfuscation techniques for both track pattern and home location protections. From each record of a person, 4 tracks containing home locations were drawn randomly, resulting in 1745 tracks. We discarded those incompletely recorded (some track sections only contain home locations), which are too short for analysis. During crowding, each of the tracks is extended with a parameter of 0.5 (see Algorithm 2), resulting in obfuscated tracks having 1.5 times the length of the original tracks.

The NO2 concentrations used for the exposure assessment were obtained from datasets developed by Schmitz *et al.* (2019). The raster datasets provide annual average air pollution concentration maps for the year 2009 on a 5 × 5 m grid. The concentrations for nitrogen (di)oxides were calculated using the land-use regression models developed in the European Study of Cohorts for Air Pollution Effects (ESCAPE) project.

The level of exposure along a track was estimated by considering the average pollution concentration within a buffer zone around the track line. Since the exposure level at each GPS track point only depends on the variation of the geographic distribution of pollutants, the size of the buffer zone can be determined based on the variation of pollutants like NO2 around each track point. According to the map of NO2 concentration, this variation stays rather uniform within a 30-meters distance. Also, this distance roughly corresponds to the GPS measurement error, which is why we decided for a buffer zone of 30 m.

The algorithms are implemented through Python 3.5 along with 'geopandas', 'shapely' and 'fiona' as major packages. The experiment is conducted on a 64-bit Operating System with Intel Core i7-6700 CPU 3.40 GHz and a RAM of 28G.

6. Results

In this section, we discuss the evaluation results, including the assessment of information loss and the effectiveness of different kinds of attacks. We start with descriptive statistics about the tracks used in this study.

The distribution of track length shows two peaks, one for very short tracks, and another one for tracks with around 100 points (Figure 10(a)). A large proportion of tracks is thus larger than 100 points, lasting for several minutes. Also, the track complexity is considerable, meaning that the track line is often convoluted instead of being straight, with a large portion of tracks showing a fractal dimension greater than 1 (Figure 10(b)).

In Table 2, we can see how this distribution is influenced by obfuscation. While the line length for random perturbation and Voronoi masking stays equivalent, the tracks become less convoluted and complex. Simulated crowding, however, considerably increases the length of tracks as well as their complexity.

Regarding computational effort, Table 2 shows that simulated crowding correspondingly requires most computing time on average, however only slightly more than Voronoi

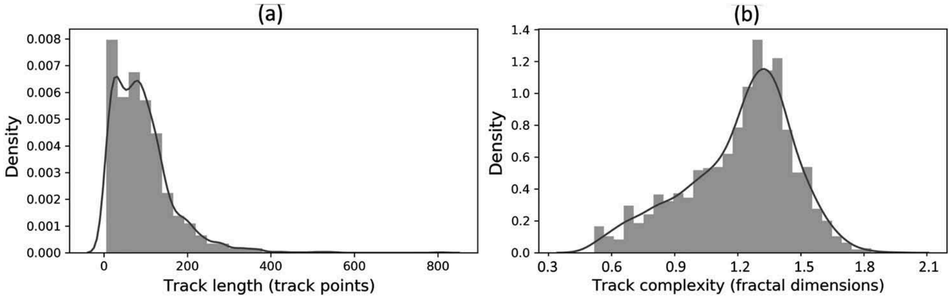

Figure 10. Statistics of original track sample. Distribution of lengths (number of points) and track complexity (fractal dimension).

Table 2. Average line length and fractal dimension of tracks under different forms of obfuscation.

	Average line length	Average fractal dimension	Avg time sec. per point
Original	92.06	1.26	
Simulated Crowding	138.06	1.44	0.1082
Gaussian RP	90.02	1.18	0.0725
Voronoi Masking	89.61	1.14	0.0990

masking. Random perturbation is cheapest in time. However, all obfuscation methods stay within a few centiseconds for either moving an existing or adding a new track point. This means that, for example, masking a track of 100 track points while extending it by 50% of its length would take 15 seconds. Thus for an experiment with a track dataset around 2000 tracks (1754 in this study), obfuscation will take slightly more than 8 hours on a HP EliteDesk 800 G3 TWR PC with a RAM of 28, and Intel Core i7-6700 CPU, 3.40 GHz.

As can be seen in Figure 11, the original track shown in (a) is extended by simulated crowding (c), such that it roughly follows the road network and ends up in a different residential area. The template masking (d) additionally hides the track within von Neumann neighbours. Gaussian perturbation and Voronoi masking, on the other hand, both conserve the general layout of the track.

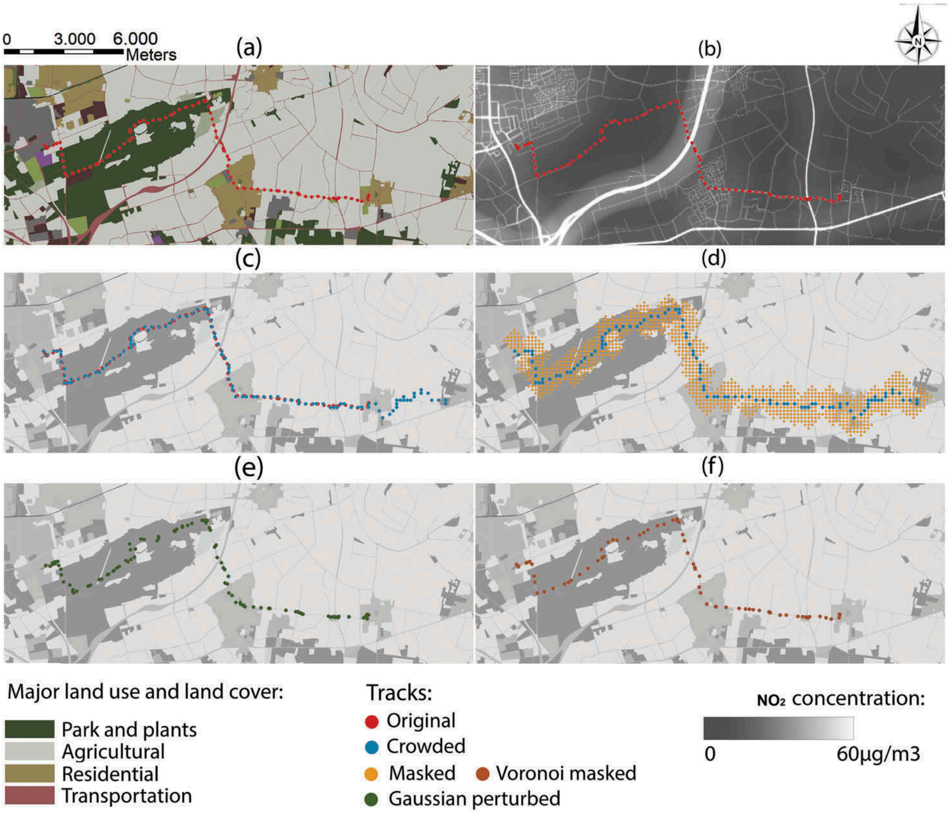

Figure 11. Overview of obfuscation results based on an example. Original track against (a) Land use and land cover, and (b) NO2 concentration. (c) Crowded (extended) track. (d) Crowded (masked) track. Same track with (e) Gaussian perturbation, and (f) Voronoi masking applied.

6.1. Effectiveness of attacks

Table 3 shows results of attacking different kinds of obfuscation with a given attack strategy, averaged over all tracks. This is measured in terms of average spatial distance (of reconstructed from original home, in m), Jaccard similarity (of buffered obfuscated tracks and buffered originals), as well as percentage error in SDE parameters (of obfuscated vs. original), measured in terms of direction angle θ, and in standard deviations along the major (x) and minor axis (y).

We can see that attacking the *home location* is much more difficult in the case of simulated crowding than in the case of the other obfuscation techniques. While the latter allow attacks with estimations of the home that are on average less than 200 meters from the true home location, simulated crowding pushes such estimations away from the true home by more than 1.5 kilometres on average. This is because our crowding method not only removes point clusters which may indicate home but also simulates fake movements far away from the original track in correspondence with residential land use patterns. The vulnerability of a track concerning these two home characteristics, therefore, seems to be effectively diminished by our method. Using the distance tolerance between true home and estimated home for each track, we assessed a corresponding *anonymity level*, counting how many other track homes in our databases would fall within this tolerance. The large reconstruction error due to crowding ensures higher k-anonymity than can be produced by either random perturbation or masking. For instance, an average home location reconstruction error of over 1.5 kilometres would yield an average *k* of around 10. Since on average there are no more than 5 neighbouring track points at the end of all the tracks within 200 meters, the Gaussian random perturbation or Voronoi masking would only ensure a k of around 5.

Similarly, Jaccard similarity of simulated crowding is lower than for all other methods and considerably lower than Voronoi Masking. Simulated crowding also increases the percentage errors in SDE parameters in comparison to the other methods by 2 to 5 times.

In Figure 12, we show the joint distribution of effectiveness of track reconstruction (Jaccard index) versus home attack error. It can be seen that a track obfuscated with simulated crowding is not only most difficult to reconstruct, but at the same time provides the largest error in attacking homes.

These results together indicate that it is going to be significantly more difficult to attack a track that was obfuscated with simulated crowding, as compared to the other two methods, at least when considering attack strategies based on land use patterns, point density and track geometry.

Table 3. Effectiveness of different privacy attacks (columns) on different obfuscation methods (rows) in terms of similarity and anonymity.

	Benchmarks	Home error avg in m	Jaccard similarity	SDE θ diff	SDE x diff	SDE y diff
Simulated	Distance	1519.89	57.18%	93.14%	244.66%	317.51%
Crowding	k-Anonymity	10				
Gaussian	Distance	175.42	60.62%	51.72%	50.73%	89.40%
RP	k-Anonymity	5				
Voronoi	Distance	169.08	72.17%	32.80%	63.08%	80.32%
Masking	k-Anonymity	5				

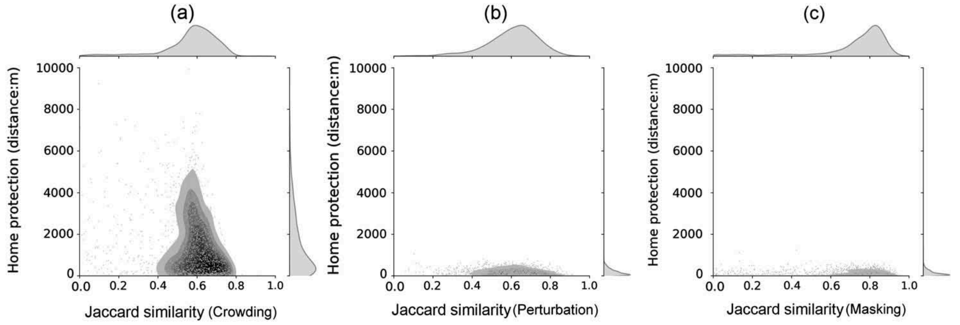

Figure 12. Effectiveness of attacks in terms of reconstruction accuracy (Jaccard index) against home reconstruction, compared for (a) crowded tracks, (b) Gaussian perturbed tracks, and (c) Voronoi masked tracks.

6.2. Information loss

Information loss was measured in terms of a mean percentage error (MPE) of the average, minimum and maximum NO_2 concentration near an obfuscated track as a proxy of estimating the pollution exposure of the tracks. The error was measured with respect to the original track. The mean absolute percentage error (MAPE) (using absolute differences) is given in brackets, see Table 4. For all obfuscation techniques, exposure tends to be underestimated, as shown by negative differences. This is because that track points are moved away from the road networks, where the pollution concentration is high. As can be seen in Table 4, the errors produced by the crowded track (*obfuscated* benchmark) are slightly higher than for the perturbed or Voronoi variants, yet within comparable ranges. Voronoi masking produces the smallest error, also in absolute terms, because Voronoi boundaries between track points stay close to these points. However, since the points of the original track within a crowded track can be *reconstructed* (compare Section 4.1), crowding offers the possibility of reducing the information loss by obfuscation to almost zero ($\approx$ 2% in absolute terms). The remaining error, in this case, is only due to the rasterization within simulated crowding. In the following comparisons, we consider the worst case, assuming that such a reconstruction of track point values is not possible because a user requests a statistics over the entire track from the service (see Section 2).

Figure 13 visualizes the joint effects of obfuscation effectiveness and information loss over all tracks. Jaccard similarity is considered as a proxy of obfuscation effectiveness, where low similarity indicates low attack accuracy and thus high obfuscation effectiveness. Among

Table 4. Information loss of different obfuscation methods (rows) in terms of error of NO_2 concentration exposure. The error was calculated compared to the original track, as a percentage of exposure difference, summarized over all tracks. We computed the mean and the absolute error (the latter in brackets). Exposure was calculated as local NO_2 concentration.

	Benchmarks	Mean (absolute) percentage error (M(A)PE) of track avg	M(A)PE of track min	M(A)PE of track max
Simulated	obfuscated	−5.12% (8.57%)	−3.43% (5.73%)	−1.66% (3.97%)
Crowding	reconstructed	−0.46% (2.06%)	−0.44% (1.49%)	−0.73% (1.52%)
Gaussian RP	obfuscated	−5.21% (5.33%)	−1.68% (2.24%)	−1.09% (2.22%)
Voronoi Masking	obfuscated	−3.18% (3.37%)	−1.96% (2.44%)	1.43% (3.47%)

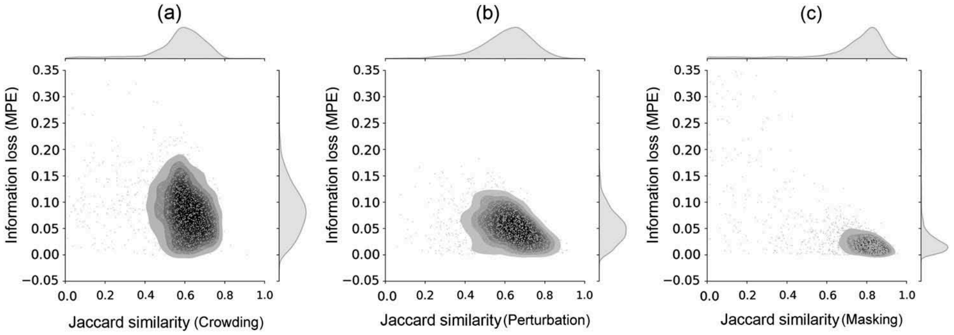

Figure 13. Obfuscation effectiveness in terms of reconstruction accuracy (Jaccard similarity, x axis) against information loss (MPE, y axis) for (a) crowding, (b) Gaussian perturbation and (c) Voronoi masking.

all obfuscation techniques, we can see that attack accuracy and MPE are negatively correlated. This means that increasing the effectiveness of obfuscation in terms of decreasing Jaccard similarity produces growing information loss in estimating pollution exposure, and vice versa. Crowding produces the steepest negative correlation (Figure 13(a)), implying that preserving information sacrifices the least amount of obfuscation effectiveness. In contrast, Gaussian perturbation (b) largely compromises obfuscation effectiveness to preserve the same amount of information. Voronoi masking (c) may preserve the highest amount of information among all the obfuscation techniques, yet is also subject to the highest risk of attack.

To further investigate these joined effects, we varied the parameters of each obfuscation strategy to see how a method can be moved along both quality dimensions. For this purpose, we ran all techniques 5 times over all tracks with increasing obfuscation intensity (and thus decreasing Jaccard similarity): For crowding, the key parameter of track length extension was increased from 0.5 to 3 times of the original length, while the Gaussian perturbation parameter of Gaussian neighbourhood size was increased from 6 to 30-meters with equal intervals. This moves the methods along the x-axis towards decreasing Jaccard similarity. As there are no parameters for the Voronoi masking, the technique was applied identically each time. Figure 14 shows the average quality values on each dimension for each of the 5 runs for these three techniques.

We can see that Voronoi masking shows no preeminent behaviour, other than being very good in preserving information and very bad in defending against attacks. For Gaussian perturbation, information loss seems to increase linearly when obfuscation is intensified. Simulated crowding, in contrast, exhibits stable information loss when increasing the intensity of obfuscation, meaning that information loss, in this case, is least sensitive to increasing effectiveness of privacy protection compared to other techniques. This stability is achieved through mimicking the original track pattern in terms of land use and spatial directions. In this way, the crowded track maintains the pollution exposure pattern along roads. The standard deviation of information loss is shown as error bars, where crowding again displays a stable information loss pattern against obfuscation intensity. Gaussian perturbation suffers from growing information loss variations with more intense obfuscation.

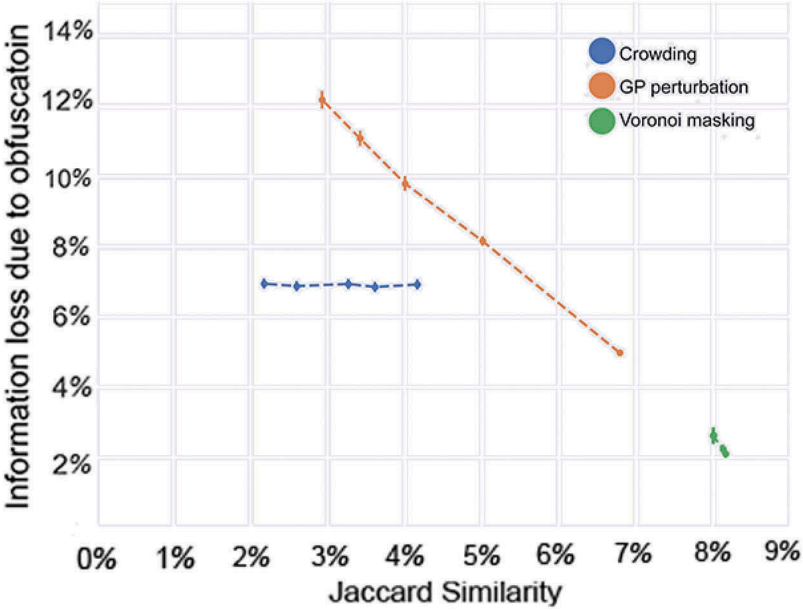

Figure 14. The change of information loss (y axis) against obfuscation effectiveness (decreasing Jaccard similarity) when increasing obfuscation intensity for each technique. This moves results along the x axis towards the origin.

We finally visualized attack accuracy and information loss together with track length to examine whether obfuscation is sensitive to track characteristics. As shown in Figure 15, all obfuscation techniques perform better on short tracks than longer ones. As the tracks become longer, reconstruction becomes less difficult leading to higher attack accuracy. The attack accuracy exceeds 0.8 once the track length is roughly over 100 track points for Gaussian perturbation and 0.9 for Voronoi masking. Even though Voronoi masking can cut down attack accuracy to a value below 0.4 for some tracks longer than 200 track points, the loss of information is large (large blue circles in Figure 15(c)). A comparably large percentage of crowded tracks reach an attack accuracy below 0.2 even though being longer than 200 points (see lower circles in Figure 15(a)). At the same time, the information loss of such tracks is within the range or smaller than Voronoi masking, though larger than Gaussian perturbation.

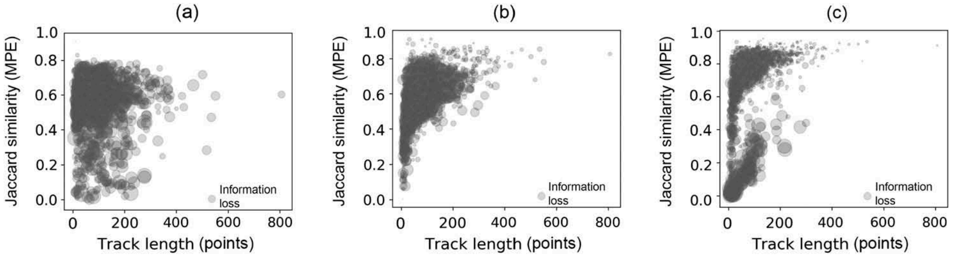

Figure 15. Sensitivity of obfuscation with respect to track length. We compare attack accuracy and information loss between (a) Crowding, (b) Gaussian perturbation and (c) Voronoi masking.

6.3. Discussion and future work

Our results illustrate that simulated crowding provides a valid alternative to state-of-the-art point quality reduction techniques. On the one hand, it is a method that significantly lowers privacy risk in terms of home attacks using point density and residential land use as well as the risk of geometric track reconstruction (using e.g. point cloud-based methods such as SDE). For all of these attack strategies, we could show that simulated crowding performs at least as good as or better than random perturbation and Voronoi masking, and is often a better choice due to the largely increased resilience against home attacks. On the other hand, our method is also capable of minimizing information loss or keeping it constant with increasing obfuscation intensity. This is an important result for analytic purposes requiring the preservation of point quality, such as exposure measurements. When computing analytics over the raw result of simulated crowding, increasing obfuscation intensity exhibits a stable error concerning exposure assessment. Also, since original track points within a crowded and enriched track can be effectively reconstructed on the client-side, we can further reduce this analytic error to the error that is introduced by grid masking.[11] In contrast to other obfuscation techniques, we thus could show that simulated crowding preserves a fixed amount of information while allowing us to largely increase the privacy level. Furthermore, though it takes somewhat more time to analyse than other strategies due to the increased amount of points, the time difference lies within a few centiseconds per track on average.

However, there are also drawbacks of our proposed method. First, for computationally expensive analytic tasks, crowding multiplies the analytic burden, as it multiplies the number of points to be analyzed. On average, simulated crowding by mimesis increases the size of a track by $\approx$ 50%. This means that the runtime of analytic methods of quadratic complexity is more than doubled ($\approx$ 220%). This is further worsened by template masking. And second, *privacy metrics for crowding*, which would be needed for minimum privacy guarantees and optimality results, are yet unknown, since current approaches are focused on point quality reduction.

For these reasons, future research should focus on designing appropriate privacy measures which would allow determining optimal crowding strategies under certain assumptions. *Crowding efficiency*, that is, identifying the least amount of fake points that can be used for crowding a track with a given analytic goal under a given privacy level should be investigated. Furthermore, different crowding strategies (with and without rasterization, with a more diverse set of mimesis and masking algorithms) should be compared against each other. Finally, the usefulness of crowding regarding analytic techniques should be assessed in future studies. It is clear that crowding as a track quality reduction strategy negatively affects those analyses that assess the entirety of a track. The implementation of a plug-in tool for common GIS is planned, which would make our method easy to use. We believe that simulated crowding should not be regarded as a substitute for other obfuscation techniques but rather occupies a particular niche within the set of geoprivacy preserving methods.

7. Conclusion

In this article, we proposed a novel obfuscation technique for spatial point tracks. Simulated crowding is capable of preserving point quality and corresponding analytic accuracy to a large extent. At the same time, it is an effective measure against privacy attacks on home locations and against track reconstruction techniques that exploit point density, track geometry or spatial context such as land use patterns. This makes it a valid alternative for defending geoprivacy, in addition to point quality reduction strategies which dominate the current state-of-the-art. In this article, we could demonstrate that simulated crowding has a higher resilience against home attacks and a stable information loss, and thus is most effective in preserving information when increasing obfuscation intensity compared to state-of-the-art obfuscation techniques. In future work, simulated crowding should be tested under different analytic goals, such as different forms of exposure, focusing on its inherent limits regarding analyses of entire tracks.

Notes

1. A group of people who share a defining characteristic and is studied for a period of time.
2. https://en.wikipedia.org/wiki/General_Data_Protection_Regulation.
3. The notion of location privacy is sometimes also used to refer to the *privacy of locations* visited by non-anonymous users of social networks (Vicente *et al*. 2011). Note that we do not focus on this second meaning.
4. Code is available online https://github.com/simonscheider/CrowdingObfuscation.
5. The high concentration of points around stops makes it possible to detect start or goal locations, at which the speed is bound to decrease.
6. Here we use a notation for *set and list comprehension* similar to Python syntax: $\{$*x for x in S if Cond(x)*$\}$ denotes the set of elements of *S* satisfying the condition *Cond*. For vectors, we use ordinary vector algebra notation.
7. Masking would work also for continuously distributed points according to the same principles outlined below, but using a different kind of template. For example, a template in which points are distributed according to some stochastic process in coordinate space, such as a bivariate normal distribution.
8. In this formula, the first factor is Gauss' *sum formula* for the sum of numbers 1 to r, and the second factor is a constant for van Neumann neighbourhoods.
9. In our study of literature, we noticed that obfuscation studies often left out either one or even both of these aspects.
10. http://www.b-riders.nl/.
11. In case a different crowding strategy is used, which substitutes discrete rasterization with continuous space, this error can even be reduced to zero.
12. To solve this equation, we use the well known zero of a quadratic function.

Acknowledgments

This study received funding from the Global Geo Health Data centre at Utrecht University, the Netherlands (http://www.gghdc.nl). We also would like to thank the anonymous reviewers for their helpful suggestions to improve the article.

Data and codes availability statement

The code used in this study is available at https://figshare.com/ under the identifier https://doi.org/10.23644/uu.11295308.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Dr. Simon Scheider is an assistant professor in Geographic Information Science at the Department of Human Geography and Spatial Planning, University Utrecht. His research lies at the interface between conceptual modeling, geographic data analysis and knowledge extraction.

Dr. Jiong Wang is a post-doc researcher and data engineer at the Department of Physical Geography, University Utrecht. His research interests involve understanding short and long-term dynamics of geographic processes.

Maarten Mol is a MSc student of Geographic Information Management and its Applications (GIMA) at the Department of Human Geography and Spatial Planning, Utrecht University.

Dr. Oliver Schmitz is an ITC developer at the Department of Physical Geography, University Utrecht.

Dr. Derek Karssenberg is a professor of Computational Geography at the Department of Physical Geography, University Utrecht. He is interested in methods and tools for process-based spatio-temporal modelling.

ORCID

Simon Scheider http://orcid.org/0000-0002-2267-4810

References

Andrés, M.E., *et al*., 2012. Geo-indistinguishability: differential privacy for location-based systems. *arXiv Preprint*, arXiv:1212.1984, 1–15.

Armstrong, M.P., Rushton, G., and Zimmerman, D.L., 1999. Geographically masking health data to preserve confidentiality. *Statistics in Medicine*, 18 (5), 497–525.

Bennett, B., 2010. Methods for handling imperfect spatial information. *In*: R. Jeansoulin, *et al*., eds. *Chap. spatial vagueness*. Vol. 256. Berlin Heidelberg: Springer, 15–47.

Chaix, B., *et al*., 2013. GPS tracking in neighborhood and health studies: a step forward for environmental exposure assessment, a step backward for causal inference? *Health & Place*, 21, 46–51. doi:10.1016/j.healthplace.2013.01.003

Chow, C.Y. and Mokbel, M.F., 2011. Privacy of spatial trajectories. *In*: Y. Zheng and X. Zhou, eds. *Computing with spatial trajectories*. New York, NY: Springer, 109–141.

Curry, M.R., 1999. Rethinking privacy in a geocoded world. *Geographical Information Systems*, 2, 757–766.

Curtis, A., *et al*., 2011. Confidentiality risks in fine scale aggregations of health data. *Computers, Environment and Urban Systems*, 35 (1), 57–64. doi:10.1016/j.compenvurbsys.2010.08.002

de Kruijf, W., 2014. Bike print: bike policy renewal and innovation by means of tracking technology. *Fietscongres 2014, Het fietsbeleid van de toekomst academy for urban development, logistics and mobility*.

De Montjoye, Y.A., *et al*., 2013. Unique in the crowd: the privacy bounds of human mobility. *Scientific Reports*, 3, 1376. doi:10.1038/srep01376

Duckham, M. and Kulik, L., 2005. A formal model of obfuscation and negotiation for location privacy. *In*: *International conference on pervasive computing*. Munich, Germany: Springer, 152–170.

Duckham, M. and Kulik, L., 2006. Location privacy and location-aware computing. *Dynamic & Mobile GIS: Investigating Change in Space and Time*, 3, 35–51.

Fechner, T. and Kray, C., 2012. Attacking location privacy: exploring human strategies. *In*: *Proceedings of the 2012 ACM conference on ubiquitous computing*. Pittsburgh Pennsylvania: ACM, 95–98. doi:10.1037/a0030481.

Feng, T. and Timmermans, H.J., 2017. Using recurrent spatio-temporal profiles in GPS panel data for enhancing imputation of activity type. *In*: L. A. Schintler, Z. Chen, eds. *Big Data for Regional Science*. Routledge, 21–130.

Gedik, B. and Liu, L., 2004. A customizable k-anonymity model for protecting location privacy. *In*: *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*. Tokyo, Japan: ACM.

Ghinita, G., 2009. Private queries and trajectory anonymization; a dual perspective on location privacy. *Transactions on Data Privacy*, 2 (1), 3–19.

Gruteser, M. and Grunwald, D., 2003. Anonymous usage of location-based services through spatial and temporal cloaking. *In*: *Proceedings of the 1st international conference on Mobile systems, applications and services*. San Francisco, CA, USA: ACM, 31–42.

Hoh, B., *et al*., 2006. Enhancing security and privacy in traffic-monitoring systems. *IEEE Pervasive Computing*, 5 (4), 38–46. doi:10.1109/MPRV.2006.69

Hu, Y., *et al*., 2013. A geo-ontology design pattern for semantic trajectories. In: T. Tenbrink, J. Stell, A. Galton, Z. Wood, eds. *Spatial Information Theory. COSIT 2013. Lecture Notes in Computer Science*. Vol. 8116. Scarborough, UK: Springer, Cham, 438–456.

Keßler, C. and McKenzie, G., 2018. A geoprivacy manifesto. *Transactions in GIS*, 22 (1), 3–19. doi:10.1111/tgis.2018.22.issue-1

Kido, H., 2006. *Location anonymization for protecting user privacy in location-based services*. Thesis (PhD). Masters thesis, Graduate School of Information Science and Technology.

Kido, H., Yanagisawa, Y., and Satoh, T., 2005. An anonymous communication technique using dummies for location-based services. *In*: *Pervasive Services, 2005. ICPS'05. Proceedings. International Conference on* IEEE, Santorini, Greece. 88–97. doi:10.3171/spi.2005.2.1.0088.

Kounadi, O. and Leitner, M., 2016. Adaptive areal elimination (AAE): a transparent way of disclosing protected spatial datasets. *Computers, Environment and Urban Systems*, 57, 59–67. doi:10.1016/j.compenvurbsys.2016.01.004

Krumm, J., 2007. Inference Attacks on Location Tracks. *In:* A. LaMarca, M. Langheinrich, K. N. Truong, eds. *Pervasive Computing. Pervasive 2007. Lecture Notes in Computer Science*. vol 4480. Toronto, Canada: Springer, Berlin, Heidelberg, 127–143.

Krumm, J., 2009. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13 (6), 391–399. doi:10.1007/s00779-008-0212-5

Kwan, M.P., Casas, I., and Schmitz, B., 2004. Protection of geoprivacy and accuracy of spatial information: how effective are geographical masks? *Cartographica: the International Journal for Geographic Information and Geovisualization*, 39 (2), 15–28. doi:10.3138/X204-4223-57MK-8273

Machanavajjhala, A., *et al*., 2008. Privacy: theory meets practice on the map. *In*: *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on* IEEE, Cancun, Mexico. 277–286.

Nussbaum, D., Omran, M.T., and Sack, J.R., 2017. Maintaining anonymity using (i,j)-privacy. *Journal of Location Based Services*, 11 (1), 1–28. doi:10.1080/17489725.2017.1363419

Pfitzmann, A. and Köhntopp, M., 2001. Anonymity, unobservability, and pseudonymity - a proposal for terminology. *In*: H. Federrath, eds. *Designing privacy enhancing technologies*. Berlin, Heidelberg: Springer, 1–9.

Raubal, M., 2011. Cogito ergo mobilis sum: the impact of location-based services on our mobile lives. *In*: *The SAGE Handbook of GIS and Society*. London: Sage Publishing, 159–173.

Reiter, M.K. and Rubin, A.D., 1999. Anonymous web transactions with crowds. *Communications of the ACM*, 42 (2), 32–48. doi:10.1145/293411.293778

Rodgers, S.E., *et al*., 2012. Protecting health data privacy while using residence-based environment and demographic data. *Health & Place*, 18 (2), 209–217. doi:10.1016/j.healthplace.2011.09.006

Scheider, S., *et al*., 2018. Computing with cognitive spatial frames of reference in GIS. *Transactions in GIS*, 22, 1083–1104. doi:10.1111/tgis.v22.5

Schmitz, O., *et al.*, 2019. High resolution annual average air pollution concentration maps for the Netherlands. *Scientific Data*, 6, 190035. doi:10.1038/sdata.2019.35

Seidl, D.E., *et al.*, 2015. Spatial obfuscation methods for privacy protection of household-level data. *Applied Geography*, 63, 253–263. doi:10.1016/j.apgeog.2015.07.001

Seidl, D.E., Jankowski, P., and Nara, A., 2018. An empirical test of household identification risk in geomasked maps. *Cartography and Geographic Information Science*, 46, 1–14.

Shokri, R., *et al.*, 2011. Quantifying location privacy. *In*: *2011 IEEE symposium on security and privacy* IEEE, Oakland, CA, USA, 247–262. doi:10.1177/1753193410392116.

Shoval, N., *et al.*, 2014. The shoemakers son always goes barefoot: implementations of GPS and other tracking technologies for geographic research. *Geoforum*, 51, 1–5. doi:10.1016/j.geoforum.2013.09.016

Sweeney, L., 2002. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10 (05), 571–588. doi:10.1142/S021848850200165X

Veregin, H., 1999. Data quality parameters. *Geographical Information Systems*, 1, 177–189.

Vicente, C.R., *et al.*, 2011. Location-related privacy in geo-social networks. *IEEE Internet Computing*, 15 (3), 20–27. doi:10.1109/MIC.2011.29

Weiser, P. and Scheider, S., 2014. A civilized cyberspace for geoprivacy. *In*: *Proceedings of the 1st ACM SIGSPATIAL international workshop on privacy in geographic information collection and analysis*. Dallas, Texas, USA: ACM, p. 5.

Xu, T. and Cai, Y., 2007. Location anonymity in continuous location-based services. *In*: *Proceedings of the 15th annual ACM international symposium on advances in geographic information systems*. Seattle, Washington: ACM, p. 39. doi:10.1094/PDIS-91-4-0467B.

Zandbergen, P.A., 2014. Ensuring confidentiality of geocoded health data: assessing geographic masking strategies for individual-level data. *Advances in Medicine*, 2014, 1–14.

Zang, H. and Bolot, J., 2011. Anonymization of location data does not work: a large-scale measurement study. *In*: *Proceedings of the 17th annual international conference on Mobile computing and networking*. Las Vegas, Nevada: ACM, 145–156. doi:10.1177/1753193411417230.

Zhang, S., *et al.*, 2017. The location swapping method for geomasking. *Cartography and Geographic Information Science*, 44 (1), 22–34. doi:10.1080/15230406.2015.1095655

Appendix

(1) A credible d (credible maximum movement distance between points) should depend on the maximum distance measured between points in the track. Here we used a tolerance of 1.3 times this maximum:

$$param_d(track) = 1.3 * max([\| (v_i - v_{i+1}) \| \text{ for } v_i \text{ in } track \text{ if } \mathrm{i} < \| track \|])$$

(2) $m \leq d$, more precisely $\pi \left\lfloor \frac{d}{m} \right\rfloor^2 \geq k$ (at least k points with increment m must fit in a buffer of radius d). If we assume that a single movement should be able to reach at least k points, then the rounding increment m must be:

$$\pi \left\lfloor \left(\frac{d}{m} \right) \right\rfloor^2 = 2k$$

$$\left(\frac{d}{m} \right)^2 \geq \frac{2k}{\pi}$$

$$\frac{d}{m} \geq \sqrt{\frac{2k}{\pi}}$$

$$m \leq \frac{d}{\sqrt{\frac{2k}{\pi}}}$$

$$param_m(d,k) = \left\lfloor \frac{d}{\sqrt{\frac{2k}{\pi}}} \right\rfloor$$

(3) the template radius *r* must provide a search space big enough to find *k* candidates in the template, i.e. assuming a *von Neumann* template, $size^{vN}(r) > k$. For example, if the neighbourhood radius *r* for generating a template should contain double as many points as required by k[12]:

$$\frac{r(r+1)}{2}4 \geq 2k$$

$$r(r+1) \geq k$$

$$r^2 + r \geq k$$

$$r^2 + r + (-k) \geq 0$$

$$param_r(k) = \left\lceil \frac{-1 + \sqrt{1+4k}}{2} \right\rceil$$

So, e.g. when k is 2, r must be 1 (von Neumann neighbourhood template containing 4 points), and if k is 3, r must be 2 (template containing 12 points).