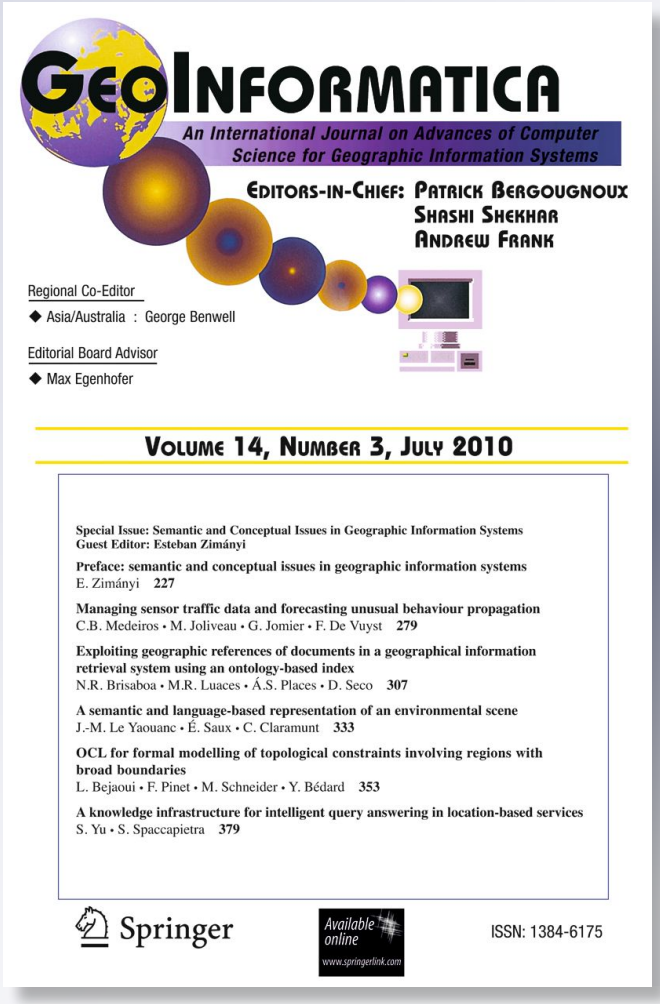
Blind and squaring-resistant watermarking of vectorial building layers

Julien Lafaye, Jean Béguec, David Gross-Amblard & Anne Ruas

GeoInformatica
An International Journal on Advances of Computer Science for Geographic Information Systems

ISSN 1384-6175

Geoinformatica
DOI 10.1007/
s10707-011-0133-8



Springer

Your article is protected by copyright and all rights are held exclusively by Springer Science+Business Media, LLC. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your work, please use the accepted author's version for posting to your own website or your institution's repository. You may further deposit the accepted author's version on a funder's repository at a funder's request, provided it is not made publicly available until 12 months after publication.



Author's personal copy

Geoinformatica
DOI 10.1007/s10707-011-0133-8

Blind and squaring-resistant watermarking of vectorial building layers

Julien Lafaye · Jean Béguec · David Gross-Amblard · Anne Ruas

Received: 24 February 2009 / Revised: 20 March 2011 /
Accepted: 9 June 2011
© Springer Science+Business Media, LLC 2011

Abstract Due to the ease of digital copy, watermarking is crucial to protect the intellectual property of rights owners. We propose an effective watermarking method for vectorial geographical databases, with the focus on the buildings layer. Embedded watermarks survive common geographical filters, including the essential squaring and simplification transformations, as well as deliberate removal attempts, e.g. by noise addition, cropping or over-watermarking. Robustness against the squaring transformation is not addressed by existing approaches. The impact on the quality of the data sets, defined as a composition of point accuracy and angular quality, is assessed through an extensive series of experiments. Our method is based on a quantization of the distance between the centroid of the building and its extremal vertex according to its orientation.

Keywords Watermarking · Blind · Squaring · Buildings

Work partially supported by the ACI Sécurité & Informatique TADORNE grant (2004–2007) and the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC grant Webdam, agreement 226513. (http://webdam.inria.fr/).

J. Lafaye
Laboratoire CEDRIC, Spécialité Informatique – CC 432,
Conservatoire national des arts & métiers, 292 rue Saint Martin,
75141 Paris Cedex 3, France

D. Gross-Amblard
INRIA Saclay, Paris, France

D. Gross-Amblard (✉)
Laboratoire Le2i UMR CNRS 5158, Faculté des Sciences,
Université de Bourgogne, BP 47870, 21078 Dijon Cedex, France
e-mail: david.gross-amblard@u-bourgogne.fr

J. Béguec · A. Ruas
Laboratoire COGIT, Institut Géographique National (IGN),
2/4 Avenue Pasteur, 94 165 Saint Mandé Cedex, France

Published online: 06 July 2011

Author's personal copy

1 Introduction

Geographical Information Systems (GIS) have existed for more than 40 years, and their application domain is now wide, ranging from environmental surveillance by country agencies to localization-aware services for individual mobile users. This phenomenon is stressed for the general public by the increasing availability of GPS devices (e.g. car navigation) and the recent development of Google Earth™ and GeoPortail™ [12]. Most of these geographical applications rely on an underlying vectorial spatial database (points, polylines and polygons). Even in Google Earth™ or GeoPortail™, where the user interface is image-oriented, the provided satellite images are semantically underlined with polygonal structures representing points of interest (viewpoint, services), buildings or road networks.

Gathering such accurate information is an onerous task for the data owner. Hence, huge and detailed vectorial databases carry a high scientific and/or economical value. For example, 50 USD is the usual fee to be licensed to use polygons from a narrow one square kilometer area. This price is 10 times higher for a reproduction license and far much higher for a full commercial license. Due to the ease of reproduction of digital media, unauthorized copy and use threaten geographical data providers. Protecting the intellectual property (IP) of rights owner is then a requirement.

On the legal side, data providers restrict the way buyers are allowed to use their data. On the technical side, robust watermarking is a known technique for IP protection. It consists of hiding a copyright mark within the data set. Embedded marks must be robust against removal attempts to be useful. In this paper, we propose a robust watermarking method for polygonal data sets.

To embed the watermark, the data has to be altered. What might sounds as a drawback is common to most watermarking methods [14]. There is a trade-off between watermark robustness and data alteration: the more alterations are allowed, the more robust the embedded watermark is. So, defining precisely what makes the value of a data set is a prerequisite for watermarking.

Some applications do not rely only on spatial accuracy (i.e. the distance between a point in the real world and this point in the data set). For example, spatial accuracy is not crucial for tourist city maps designers who apply strong transformations to road polylines and building polygons in order to increase readability. Some others focus on objects like forests, cliffs and shallows for which precise borders can be difficult to define. But most applications rely on accurate data for automatic operations (e.g. service proximity search, GPS navigation, spatial analysis of risks, etc.). Accuracy can even be mandatory, e.g. for reefs locations on IHO/SHOM boat maps [27]. Finally, accurate data sets must conform with some standard reference system for interoperability purposes (e.g. the World Geodetic System—WGS84, which is the GPS reference system). So any watermarking method must respect data accuracy.

Among geographical objects, the building layer is of utmost importance. For example, it turns out that most of the vectorial content of geographical databases bundles consists of building polygons (80% on the professional data set used in our experiments). Another example is the increasing market of Building Information Modeling (BIM) solutions, where vectorial data is central [3]. In turn, we focus in this paper on watermarking solutions for the building layer of SIG.

Author's personal copy

Beside accuracy, real world requirements entail specific constraints within such data. Building polygons are under the scope of the *squaring* constraint: data is systematically corrected so that buildings with right angles are mapped to polygons with right angles in the data set. This *squaring* operation, available in any GIS, is systematically performed by data owners *and* data users, and increases the angular quality of the data set. It is also very invasive since potentially each point of the data set is moved. Experiments show that it also tends to increase data accuracy. But surprisingly, squaring impact has never been taken into account by existing watermarking proposals e.g. [20, 24]:

- On the owner side, watermarking is likely to turn squared shapes into skewed ones, reducing the data quality;
- On the user side, squaring is likely to wipe out the watermark of the owner, lowering its security. This natural transformation, along with other common filters, can be interpreted as an attack on the watermark.

To take these effects into account, we model the quality of a data set by means of (1) its accuracy and (2) its angular quality. This choice is motivated in several works [7, 18], where it is deeply discussed that quality encompasses accuracy and must take into account shapes and topology.

In this paper, we propose an effective method for watermarking the building layer of a GIS. This watermarking is robust against geographical transformations (including squaring and simplification) and attacks by malicious users. As far as we know, this is the first method which takes into account the essential squaring transformation. It provides a high level of security while controlling the impact on the quality of the data set (point accuracy and angular quality) and not introducing topological errors (overlapping polygons). An extended version of our method can even resists the MBR attack, which replaces each building by its minimum bounding rectangle. Moreover the scheme is blind: the original data set is not required for detection, definitely an important property for huge data sets.

A classical skeleton [1] of databases watermarking algorithms is to create a secret dependency between (1) a robust identifier of the data and (2) one of its characteristics, e.g. between the primary key of a tuple and one of its numerical attributes. Revealing this dependency acts as a proof of ownership. In our approach, we get rid of the primary key by constructing a robust identifier for each building using a well chosen portion of the highest significant bits of the coordinates of its centroid. Then, we rely on the observation that buildings have an intrinsic orientation and that most of their edges are parallel or perpendicular to this orientation. To hide a watermark bit, we expand or shrink buildings along their orientation. The expansion ratio is deterministically chosen among a set of quantized values according to the robust identifier of the polygon, the secret key of the owner and the bit to be embedded. By embedding the watermark within the shapes of a building rather that within the coordinates of its vertices, we achieve robustness of watermarks against squaring. Our scheme is also robust against other transformations we present later in the paper. Any malicious attacker has to tremendously reduce accuracy and/or angular quality of the data set to erase the watermark.

Author's personal copy

Outline After describing a simple model for the quality of a buildings database, we introduce watermarking basics and common geographical filters in Section 2. Our watermarking procedure is described in Section 3. Correction, efficiency and robustness of the method are assessed in Section 4, through an extensive series of experiments. Related work is exposed in Sections 5 and 6 concludes.

2 Preliminaries

2.1 Quality of geographical data

We suppose, as in any geographical application, that a reference system R_0 has been chosen and that all spatial coordinates are expressed in R_0 (e.g. cartesian coordinates on the World Geodetic System, or WGS84).

A point $p = (x, y)$ is defined by its 2-dimension coordinates (x, y) in some reference system R_0. A simple polygon $P = (p_1, \ldots, p_n, p_{n+1} = p_1)$ is represented by the list of its points. Two polygons taken from a real data set are shown on Fig. 1a. A geographical database instance is defined by (R, DB) where R is a reference system and $DB = \{P_i\}, i \in \{1, \ldots, N\}$ is a set of N polygons. It is always provided with some reference system otherwise it is of no use for automatic operations (nevertheless, we discuss in Section 4.4 the problem of missing reference system).

We do not rely on the order of polygons within the data set, nor on the order of points within a polygon. Furthermore, there is no primary key identifying these polygons. Polygons are supposed non-overlapping as in many geographical applications. They can have holes. In that case, we process them as the full polygon having the same envelope. More sophisticated models of spatial data expressing topology exist, but we omit these enhancements for the sake of simplicity.

The (economical) value of a data set (R, DB) is correlated with its *mean accuracy*, its *maximum accuracy* and its *angular quality*. The *mean accuracy* (resp. *maximum accuracy*) is the mean (resp. maximum) value of the distance between a point of a (real) building and its corresponding point in the data set. The *angular quality* [2] is defined as the opposite of its *angular energy* (Fig. 2). The energy of an angle is a continuous piecewise quadratic whose minima are reached for multiples of $\pi/4$. The angular energy of a polygon is the sum of the energies of its angles. The intuition is that angles of real-world buildings are mostly right, or at least multiples of $\pi/4$. So, regular buildings have lower energy levels. Table 1 sums up our quality setting.

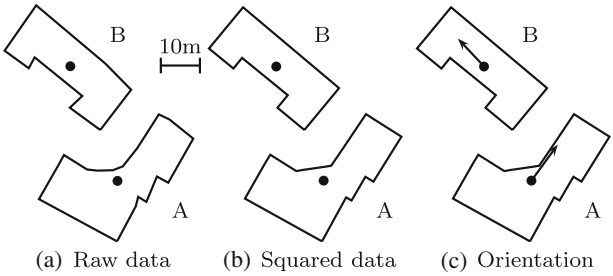


Fig. 1 Buildings polygons

Author's personal copy

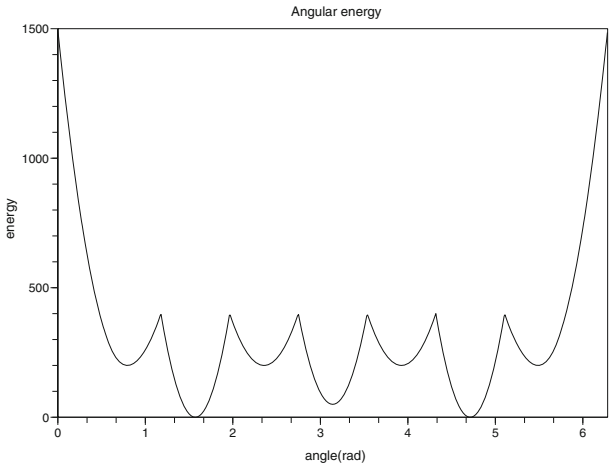


Fig. 2 Minima of angular energy ($k\pi/2$)

2.2 Watermarking

A watermarking procedure is defined as a pair of algorithms $(\mathcal{W}, \mathcal{D})$, where $\mathcal{W}$ is the watermarking algorithm, and $\mathcal{D}$ is the detection algorithm. Algorithm $\mathcal{W}$ takes as inputs a data set (R, DB), a secret key $\mathcal{K}$, tuning parameters, and produces a watermarked data set $(R, DB_{\mathcal{K}})$. The aim of the detector is, given a suspect data set (R', DB') and the secret key $\mathcal{K}$, to decide whether this data set holds a watermark or not. A watermarking procedure is said to be *blind* if the original data set is not needed by the detector $\mathcal{D}$. It is said to be *robust* if it detects marks in altered watermarked data sets. It is well known that any robust watermarking method must alter the data [14]. Hence, there is a trade-off between the allowed alteration, i.e. the allowed impact on the quality, and the robustness of the algorithm.

To evade detection, an attacker may use one of the following attacks: random alteration of point positions, mixing polygons from various data sets, applying the same or another watermarking algorithm, and specific attacks like polygon-wise rotations. We discuss these attacks in Section 4. Of course, the attacker, who still wants to re-sell a valuable data set must adopt a common reference system and limit the quality loss so that profit can still be made from the attacked database.

2.3 Geographical filters & attacks

Geographical data sets are likely to undergo transformations by legitimate or malicious users. A broad collection of such transformations is presented below. They can

Table 1 Quality criteria for building data sets

Mean accuracy	Mean distance between a (real) building point and the data set
Maximum accuracy	Maximum distance between a (real) building point and the data set
Angular quality	Inverse of angular energy, a continuous piecewise quadratic whose minima are multiples of $\pi/4$
Use of a public reference system	e.g. WGS84

Author's personal copy

be divided into correction filters (SQ, DP), readability improvements (ETR, MBR, CE) and malicious attacks (GN, OW, CA). Nevertheless, this taxonomy is not fixed as a malicious user might apply correction filters and a legitimate one might crop a large data set to keep only the part useful to him. A robust enough watermarking algorithm should resist all of them:

- **Squaring (SQ)** For each polygon, its vertices are moved so that its angular energy is lowered. The strength of the squaring is controlled by the maximum allowed alteration d on coordinates. Figure 1b shows a squared building, to be compared with its original counterpart depicted in Fig. 1a.
- **Douglas–Peucker simplification (DP)** The Douglas–Peucker simplification algorithm [6] is a polyline simplification algorithm. It works by removing the vertices of polygons that draw small artifacts on the edges of this polygon. Its strength is controlled by a threshold distance d. The higher d, the larger the removed artifacts are.
- **Cropping (CA)** Polygons not contained within a given rectangle are discarded.
- **Gaussian noise (GN)** A random noise is added to each point of the database. The distribution of the noise has mean 0 and a variable deviation d.
- **Over-watermarking (OW)** Applying the watermarking algorithm with a different key on an already marked data set.
- **Enlarge to rectangle (ETR)** This filter replaces buildings by their bounding rectangle. Two modes are available. The first one replaces each building with a rectangle having the same surface. The second one takes as input a target scale and replaces the buildings that are too small (for a legally fixed threshold value) to be legible on a map at that scale.
- **Change elongation (CE)** Applies a fixed ratio elongation along their orientation on all buildings of the data set.
- **Minimum bounding rectangle (MBR)** Replaces each building by its minimal bounding rectangle.

Translating or rotating the whole data set are not an issue, since we focus on data sets with a reference system: such transformations can be easily reversed. Observe that printing/scanning a map drawn from the data set is out of the scope of our approach, since such paper maps are of no use for automatic operations.

3 Building watermarking

3.1 Outline of the algorithm

The rationale for many watermarking algorithms is to hide a secret dependency between (1) a robust part of the data set, that will survive most alterations, and (2) one of its characteristics, whose alteration is allowed up to a reasonable limit. Revealing this secret dependency acts as a proof of ownership. We build a robust identifier id_i for each polygon P_i by using the highest significant bits of the coordinates of its centroid, expressed in the predefined reference system R_0. This identifier is robust since it is invariant through the modifications of vertex coordinates, involving only least significant bits. High amplitude modifications are likely to break the identifiers but also to lead to visible shapes alterations and/or polygon overlappings.

Author's personal copy

Furthermore, if the coordinates of the polygon of the centroid are expressed in a reference system R', different from R_0, it is easy to convert them back into R_0. Indeed, no geographical data comes without a reference system.

In order to hide a bit of information in polygon P_i, we expand or shrink it along its orientation. This orientation is computed relatively to the centroid (see Fig. 1c), and represents the majority weighted angle among edges directions. We present its computation in Section 3.3. For a rectangular shape, this orientation is parallel to the longest edge. Choosing to expand along this orientation offers several advantages. First, we observed that most edges of a polygon are parallel or perpendicular to this orientation. For example, there are 3 directions in polygon A (Fig. 1c): SW-NE, SE-NW and W-E. The main direction, i.e. the orientation is clearly SW-NE since the longest edges are heading this direction. Other directions are perpendicular or make a $\pi/4$ angle with the orientation. When a polygon is expanded along its orientation, geometrical relations between directions do not change. Second, an expansion along the orientation can still be detected if the polygon is rotated. Finally, the impact on polygon surfaces has tighter bounds compared to the case where shrinking or expanding along several directions is allowed.

It remains to compute the expansion factor to apply, and to choose which polygons are going to be altered. These operations must be done so that any attacker, aware of the watermarking method, is unable to guess on which polygons they were actually applied. A classical method to achieve this [1] is the following: use the concatenation of the given identifier id_i of a polygon and the secret key $\mathcal{K}$ of the owner to seed a pseudo-random number generator (PRNG). Use pseudo-random drawings from the generator to determine whether the current polygon is modified and, eventually, with which expansion factor. The sequence of numbers produced by the generator is predictable if and only if $id_i.\mathcal{K}$ is known. It appears purely random to anyone who does not possess this seed (an attacker may easily compute id_i, but $\mathcal{K}$ remains unknown).

Example 1 An example of our watermarking method applied on polygons A and B is shown on Fig. 3. Original shapes are shown in black and watermarked ones in gray. First, we compute the centroid of A and B, obtaining for example $O_A = (293{,}155)$ and $O_B = (171{,}447)$. To form unique identifiers id_A and id_B, we concatenate the two highest significant digits of each coordinate, obtaining $id_A = 2{,}915$ and $id_B = 1{,}744$. Choosing these two digits is correct under the hypothesis that any reasonable alteration is below 10 m and that the typical distance between any two buildings is more than 10 m (this example considers decimal base while our algorithm considers binary base). Second, based on the pseudo-random choices of a generator seeded with id_A and the secret key $\mathcal{K}$, we decide that A must be watermarked with a mark bit 0. We compute the main orientation $\mathbf{u}$ of A and find the vertex p such that $\mathbf{u}.\mathbf{Op}$ is maximal. Let $x_{\max}$ denote this value. Finally, we expand the building along its main orientation so that $x_{\max}$ becomes a predefined value $x^0_{\max}$, encoding bit 0. Polygon B is processed identically. Remark that A has been expanded whereas polygon B has been shrinked, and that most angles are invariant under this transformation.

In the following, we detail the three consecutive steps of our algorithm: (1) computation of polygon identifiers and orientations, (2) computation of expansion factors and (3) watermarking by expansion.

Author's personal copy

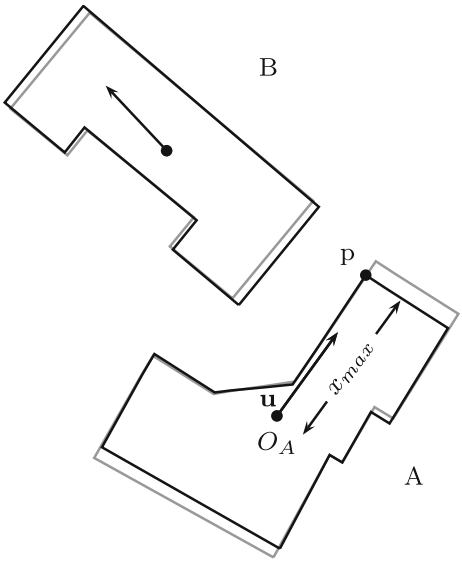


Fig. 3 Bit embedding by expansion

3.2 Computing robust identifiers

As a robust identifier, we use the highest significant bits of the centroid of the polygon. If P is a polygon with n points $p_1, \ldots, p_n$, $p_{n+1} = p_1$, its area A and its centroid $O = (x_0, y_0)$ can be computed using the following common formulae:

$$A = \frac{1}{2} \sum_{1 \le i \le n} (x_i y_{i+1} - x_{i+1} y_i),$$

$$x_O = \frac{1}{6A} \sum_{1 \le i \le n} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i),$$

$$y_O = \frac{1}{6A} \sum_{1 \le i \le n} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i).$$

Centroids of polygons A and B are represented as black dots on Fig. 1a and b.

We need to ensure that the chosen highest significant bits are significant enough. Suppose that the h-th bit is the least highest significant bit. On the one hand, h must be high enough to that small modifications of the polygon do not change the identifier. On the other hand, h must be small enough so that two adjacent polygons do not share the same identifier. The identifier of a polygon P is computed by pruning in the binary representations of its x and y coordinates the bits that represent powers of two at most $h - 1$ and concatenating them. We denote by $hsb(O, h)$ this operation.

$$id = hsb(O, h) = concat(hsb(x_O, h), hsb(y_O, h)).$$

As we will see later, two watermarked polygons sharing an identifier will be expanded using the same quantization step. Such an information may be used by an attacker to break the watermark. Therefore, shared identifiers must be avoided.

Author's personal copy

A good choice is to select the smallest value h for which the identifiers vary from one polygon to the other. We illustrate how we compute h by means of an example. For our real-life data set (detailed in Section 4), we measured, for each polygon, the distances between the centroid and the farthest vertex from the centroid. We obtain a mean of 12 m and a standard deviation of 10.5 m, that is, most polygons have an interior maximum distance between $2(12 - 10.5) = 3$ m and $2(12 + 10.5) = 45$ m. Considering 3 m as a minimum size for a polygon, most polygons have their centroid spaced by at least 6 m. Even if there are pathological configurations, it is very likely that, by choosing $h = 2$ m, two polygons do not share the same identifier. When $h = 2$, $l = 2^h = 2^2 = 4$ m is considered as the minimum significant distance. Experimentally, we verified that, by choosing $h = 0, 2, 4$ and 8, we had 0, 0, 40 and 300 cases of identifiers collision out of 4278 polygons. Hence, it seems that a good heuristic is to use $h = \lfloor \log_2(2.(\bar{l} + \delta l)) \rfloor$ if $\bar{l}$ and δl are the mean and the deviation of the main lengths of the polygons in the data set.

3.3 Computing polygon orientation

We define the main orientation $\mathbf{u}$ of a polygon as the maximum weighted orientation of its edges. For instance, if e_1 and e_2 are the only edges to have orientation α, then the weight of angle α is the sum of the lengths of e_1 and e_2. The problem is that parallel walls in the real world are not necessarily mapped to parallel edges in the data set. So, we need to sum the lengths of edges that are almost parallel. We define ε the tolerance angle, that is e_1 and e_2 are considered as having the same orientation if their orientations α_1 and α_2 are such that $|\alpha_1 - \alpha_2| < \varepsilon$. To efficiently compute the orientation, we defined a bucket-based classifying algorithm based on the observation that there is often only a small number of different orientations per polygon. The algorithm consists of the following three steps. First, we create a set of k empty buckets, provided we choose k such that $\pi/k < \varepsilon$. In bucket i, we put all edges having an orientation between $(i-1).\frac{\pi}{k}$ and $i.\frac{\pi}{k}$. Hence, in buckets i and $i+1$ we have all edges that are almost equal to $i.\pi/k$. Then, we aggregate these small buckets into bigger ones by merging two buckets if there is no empty bucket between them. The main orientation of a building is computed as the mean value of the bucket having the highest cost (the cost of a bucket being defined as the sum of the lengths of the edges in that bucket). It can happen that three or more buckets need to be aggregated, leading to consider orientations as equal when their difference is greater than angle tolerance. This is very unlikely. Indeed, we observed that on buildings, there is only a few directions per polygon (2, 3 in most cases) which are clearly separated. A similar approach was followed in [7] with the main difference that this latter method requires to compute the weight of all π/k orientations and select the one with the highest weight. Our method is more efficient but may be less accurate in a restricted number of situations.

Example 2 We illustrate the orientation computation algorithm on polygon A of Fig. 1b. The number of classes is set up to 10. Table 2 contains the number of edges and the weight of each bucket. The highest cost bucket is bucket 4, i.e. the orientation is between $3\pi/10$ and $4\pi/10 = 2\pi/5$. The computation of the weighted mean angle of bucket 4 gives 0.96 rad.

Author's personal copy

Table 2 Angles buckets

Bucket	0	1	2	3	4	5	6	7	8	9
# edges	3	0	0	8	0	0	0	0	6	0
Weight	9.02	0	0	62.4	0	0	0	0	45.2	0

For some specific shapes, e.g. perfect squares and circles, our definition of orientation is ambiguous. Instead of arbitrarily choosing an orientation, we simple ignore these unusual cases for polygon expansion. On the test sample of our experiments, we had to ignore 1 polygon out of 4278.

3.4 Expansion as a bit embedding method

In this subsection we show how to embed a single watermark bit b into a polygon P. To ensure that the watermark is robust enough, we alter the overall shape of the polygon. More precisely, we alter the longest distance $x_{\max}$ (see Fig. 3) along the orientation $\mathbf{u}$ from the centroid O to a vertex p. For a rectangular polygon, this length is half the length of the longest edge. We name *main length* this longest distance. But only altering the coordinates of p is not sufficient because it may lower angular quality (right angles may be flattened by this transformation). Hence, we choose to alter all lengths along the orientation $\mathbf{u}$ so that most angles are preserved. Let $\mathbf{v}$ be the unary vector such that $(0, \mathbf{u}, \mathbf{v})$ is a direct orthonormal basis. Watermarking is done as follows:

- compute the x coordinate x_i of each point p_i of the polygon in $(0, \mathbf{u}, \mathbf{v})$;
- compute the main length $x_{\max} = \max_i\{|x_i|\}$;
- expand all points coordinates along direction $\mathbf{u}$ so that $x_{\max}$ becomes one of the values $\{x^0_{\max}, x^1_{\max}\}$ coding a watermark bit 0 or 1.

This latter operation on $x_{\max}$ is known as quantization. Given a quantization step d, we define 0-quantizers (resp. 1-quantizers) as $q^k_0 = k.d$ (resp. $q^k_1 = k.d + d/2$), $k \in \mathbb{Z}$. Intuitively, 0-quantizers (resp. 1-quantizers) are used to code a bit 0 (resp. a bit 1). To quantize the value $x_{\max}$ using the i-quantizers ($i \in \{0, 1\}$), we look for k_0 such that $|q^i_{k_0} - x_{\max}|$ is minimal. More precisely, this is achieved with the following steps (quantization on 0-quantizers is presented):

- compute $k_r = x_{\max}/d$;
- round k_r to the closest integer k_0;
- define the quantized version of $x_{\max}$ as $x'_{\max} = k_0.d$.

The quantization process is illustrated on Fig. 4.

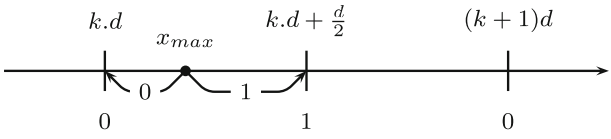


Fig. 4 Encoding 0 or 1 into the main length $x_{\max}$ using quantization

Author's personal copy

The expansion coefficient of the polygon is defined as $\sigma = x'_{\max}/x_{\max}$. We transform each point $p = x.\mathbf{u} + \mathbf{y}.\mathbf{v}$ in the original polygon into a point $p' = \sigma.x.\mathbf{u} + y.\mathbf{v}$ in the watermarked polygon.

$$p = x.\mathbf{u} + y.\mathbf{v} \quad \text{original point } p.$$

$$p' = \sigma.x.\mathbf{u} + y.\mathbf{v} \quad \text{modified point } p'.$$

The expansion is such that the maximum distortion on a vertex of a polygon is at most $d/2$. This distortion is reached only for the vertices that are the farthest from the centroid along $\mathbf{u}$. On the average, and for these points, the actual distortion is $d/4$.

3.5 Watermarking algorithm

Watermarking The complete algorithm is presented in Algorithm 1. Let $1/\gamma$ be the target ratio of watermarked polygons. It is a parameter of the algorithm. For each polygon of the data set, we compute its robust identifier *id*. Then, we seed a pseudo-random number generator G (PRNG) with $\mathcal{K}.id$. If the first integer produced by G modulo γ is 0, we embed a bit in the polygon. The bit is chosen according to the next binary value produced by G and embedded using the previously described expansion method.

Algorithm 1 Watermarking algorithm

```
1: INPUT: secret key K, watermarking ratio 1/γ, h, quantization step interval D = [d_min, d_max]
2: DATA: (R_0, DB): original data set
3: OUTPUT: (R_0, DB_K): watermarked data set
4:
5: for each building P in DB do
6:    O ← Centroid(P)
7:    id ← hsb(O, h)                                   // robust identifier id
8:    seed(G, K · id)                                  // seed the PRNG G with K · id
9:    if NextInteger(G) mod γ = 0 then
10:                                                    // Watermark this building
11:      u ← orientation(P)                            // orientation
12:      x_max ← max{p ∈ P|Op · u}                     // main length
13:      d ← d_min + NextFloat(G).(d_max − d_min)      // quantization step
14:      b ← NextInteger(G) mod 2                      // watermark bit b
15:      x'_max ← quantize(x_max, d, b)                // quantize x_max
16:      σ ← x'_max/x_max                              // expansion ratio
17:      Shrink(P, O, u, σ)
18:
19:      if collisions() then
20:         rollback()
21:      end if
22:   end if
23: end for
```

Author's personal copy

Variable step quantization We do not use a single quantization step d but a quantization interval $[d_{\min}, d_{\max}]$. Indeed, if d is the same for the whole data set, main lengths of all watermarked polygons will be multiples of d. This could be easily detected and used by an attacker to alter the watermark. To prove this, we measured the cumulative distributions of main lengths in three data sets: an unwatermarked one, a watermarked one using a fixed quantization step and another watermarked one using a variable quantization step, randomly chosen. For the two watermarked data sets, all polygons were watermarked ($\gamma = 1$) to emphasize differences. Graphs are shown on Fig. 5. We notice that, when the quantization step is fixed, the cumulative distribution is piecewise constant whereas it is very close to the original one when a variable quantization step is used. Fixed-step quantization is too visible through statistical analysis. So, we use a different quantization step for each watermarked polygon.

Discussion Using this method, the watermark is spread almost uniformly over the data set. This process being controlled by a secret key, it is impossible to find the exact locations of expanded polygons, assuming the PRNG is secure. To alter the watermark, an attacker has to alter much more polygons than the watermarking process did if he wants to be sure to affect all watermarked polygons. The choice of watermarking parameters γ, $d_{\min}$ and $d_{\max}$ depends on the specific usage of the data set. They cannot be fixed arbitrarily for all applications but the following rules are always valid:

- there is an unavoidable trade-off between quality ($\gamma \uparrow, d_{\min} \downarrow, d_{\max} \downarrow$) and robustness of the watermark ($\gamma \downarrow, d_{\min} \uparrow, d_{\max} \uparrow$). Experiments presented in Section 4 give indications on how to choose optimal values;
- if the accuracy (maximum distance between a point in the data set and in the real world) of the unwatermarked database is β_1, then the accuracy of the watermarked one is $\beta_1 + d_{\max}/2$. If the watermarked data set is sold under the agreement of an accuracy β_2, then $d_{\max}$ must be chosen so that $d_{\max} < 2(\beta_2 - \beta_1)$;

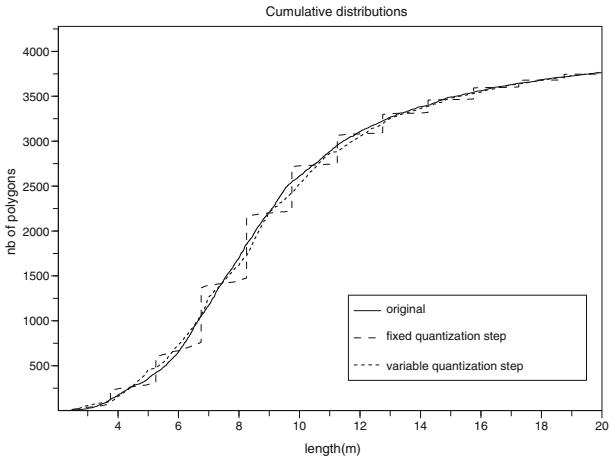


Fig. 5 Statistical invisibility of watermarked data using variable step quantization

Author's personal copy

– the allowed alteration on the building, i.e. $d_{\max}$ must be higher than the typing accuracy of the data set. Below this value, alterations can be considered as noise and rounded by a malicious user without altering the quality of the data set at all. For instance, a 1 mm alteration is meaningless in a data set of accuracy 1 m.

3.6 Handling data constraints

The bit embedding method using expansion does not take into account topological relationships between buildings. We voluntarily chose to ignore them during bit embedding and to detect errors and cancel modifications when needed (function `collisions`). Such a strategy is valid as soon as few errors occur. By choosing $d_{\max} = 4$ m, the alteration on each point of a polygon is at most 2 m. Usually, even in urban areas, polygons are spaced by a distance superior to 2 m. Indeed, with this value, we got only one case of overlapping, even in the worst setting, i.e. when $\gamma = 1$. This validates the detect-and-cancel strategy. Such a post-watermarking filtering enables to handle any kind of errors which can occur sparsely during the watermarking process. Observe that since bit embedding is a local operation, collisions can only occur in a small area around the polygon. This allows the use of classical spatial indexing techniques to check for collisions.

Observe nonetheless that the watermarking operation should not interfere with the original, unchanged data set. Indeed, all watermarking operations should occur only at release time, when data are sold to a user. Otherwise, the watermarking operation may break various integrity constraints used by the storage system.

3.7 Detection

Outline Given a suspect data set (R', DB'), we translate it into the original reference system R_0. Then, we perform the actual detection which is very similar to the watermarking algorithm, with the essential difference that no alteration is performed. It consists of two steps: computing the ratio of matching polygons and comparing this ratio to a predefined threshold value α. The values of $d_{\min}, d_{\max}, h, \gamma$ and $\mathcal{K}$ used for detection must be the same as the ones used for watermarking. So they must be kept as part of the secret. For each polygon we seed a random generator with $\mathcal{K}$ concatenated with its identifier. If the polygon satisfies the watermarking condition (i.e. $\texttt{nextInteger}(G) \mod \gamma = 0$), we compute the expected bit value b as $\texttt{nextInteger}(G) \mod 2$. We also compute the quantization step d between $d_{\min}$ and $d_{\max}$. Then, we decode the bit b' embedded in the main length $x_{\max}$ of the polygon and compare it with b.

Decoding To decode a bit from a quantized value x, we simply check whether it is one of the 1-quantizers or one of the 0-quantizers. If x is none of the i-quantizers, we compute the closest quantized value x'_1 in 1-quantizers and the closest quantized value x'_0 in 0-quantizers. We compare the distance $d_0 = |x'_0 - x|$ and $d_1 = |x'_1 - x|$. If $d_0 < d_1$, we decode a bit 0; if $d_0 > d_1$, we decode a bit 1. If $d_0 = d_1$ no bit can be decoded. Note that a quantized value, with step d, can be altered up to $d/4$ without leading to a decoding error. Quantization has been chosen because it optimizes the trade-off between average distortion (here, $d/2$) and the minimum alteration leading to a decoding error (here, $d/4$).

Author's personal copy

Algorithm 2 Detection algorithm

INPUT: secret key $\mathcal{K}$, watermarking ratio $1/\gamma$, h, quantization step interval $D = [d_{\min}, d_{\max}]$, max. false positive occurrence probability f_p, reference system R_0
DATA: (R', DB'), a suspect data set
OUTPUT: MARK or NO_MARK

Convert DB' into R_0
for each building P in DB **do**
  $O \leftarrow \texttt{Centroid}(P)$
  $id \leftarrow \texttt{hsb}(O, h)$ // robust identifier id
  $\texttt{seed}(G, \mathcal{K} \cdot id)$ // seed the PRNG G with $\mathcal{K} \cdot id$
  if $\texttt{NextInteger}(G) \mod \gamma = 0$ **then**
    t++ // increment total count
    $\mathbf{u} \leftarrow \texttt{orientation}(\mathbf{P})$ // orientation
    $x_{\max} \leftarrow \max\{p \in P | \mathbf{Op} \cdot \mathbf{u}\}$ // main length
    $d \leftarrow d_{\min} + \texttt{NextFloat}(G).(d_{\max} - d_{\min})$
    $b \leftarrow \texttt{NextInteger}(G) \mod 2$ // expected bit b
    $x'_0 \leftarrow \texttt{quantize}(x_{\max}, d, 0)$ // closest 0-quantizer
    $x'_1 \leftarrow \texttt{quantize}(x_{\max}, d, 1)$ // closest 1-quantizer
    if $|x_{\max} - x'_0| > |x_{\max} - x'_1|$ **then**
      $b' \leftarrow 1$ // found bit is $b' = 1$
    else
      $b' \leftarrow 0$ // found bit is $b' = 0$
    end if
    if $b = b'$ **then**
      $m++$ // increment match count
    end if
  end if
end for
$\alpha \leftarrow \texttt{Threshold}(f_p, t)$
if $|m/t - 1/2| > \alpha$ **then**
  Return MARK
else
  Return NO_MARK
end if

If the expected bit b and the decoded bit b' are the same, we say that the polygon matches. We maintain two counters, t (total) and m (match). The first one is incremented each time a polygon satisfying the watermarking condition is found. The second one is incremented each time this polygon carries the expected mark bit. Hence, the detection ratio m/t is the ratio of matching polygons.

It is easy to see that on a third party data set, the probability that each polygon matches is $1/2$. Therefore, the ratio m/t is compared to its expected value $1/2$ to decide whether the mark of the owner is present in the document or not. Practically, a detection threshold α must be set to bound the detection area. We detect a mark when $|m/t - 1/2| \geq \alpha$. The relevance of the detection process highly relies on the

Author's personal copy

value of α. Proposition 1 gives the detection threshold for a maximum false positive occurrence probability f_p.

Proposition 1 (*Direct application of* [9]) *Let p the number of polygons satisfying the watermarking conditions. If each polygon has a probability $1/2$ to match, the probability $\mathcal{P} = Pr\,(m/t - 1/2 \geq \alpha)$ is such that $\mathcal{P} \leq e^{-2\alpha t^2}$.*

Then, the false positive occurrence probability defined as $f = \mathcal{P}\,(|m/t - 1/2| \geq \alpha)$ is such that $f \leq 2e^{-2\alpha t^2}$. Choosing $\alpha = -\log(\delta/2)/2t$ as the detection threshold permits to keep f under δ. We use this formula in our experiments to keep false positives occurrence probability under $\delta_0 = 10^{-4}$.

4 Experiments

4.1 Framework

Data All experiments presented in this paper (except speed ones) were realized on buildings from the French city of Pamiers. The data is part of the **BD TOPO**™ [11], a topological database product from the French National Mapping Agency (IGN), the major maps provider on the French market. The product consists of several coherent layers (hydrographic network, roads, buildings...) from which we extracted only the buildings layer. This layer is composed of 4,278 polygons (35,565 vertices), representing dense build areas (downtown—west side) as well as sparse ones (residential blocks—east side). It has a contractual accuracy of 1 m. A glimpse on the Pamiers data set is given in Fig. 6.

Software We developed a Java version of our algorithm and packaged it as a library for our generic database watermarking framework [16]. The data set was stored in a Postgresql/Postgis database. Advanced geographic functions from GeOxygene [5] were also used. GeOxygene is an open source application developed at IGN. It aims at providing an open framework which implements OGC/ISO specifications for the development and deployment of geographic applications. The core watermarking method is available under the GPL license [15].

Filters/attacks We performed an extensive series of experiments to validate the robustness of our method. All the filters/attacks presented in Section 2.3 were tested (Fig. 7).

Protocol We consider that an attack is successful if it destroys the watermark with high probability while inducing a quality loss comparable to the one introduced by the watermarking process. In a same manner, we consider that a watermarking algorithm A is better than an algorithm B against a specific attack if it is as robust as B while inducing a quality loss significantly smaller than B. To emphasize the benefits of watermarking by expansion, we compare our method with a random noise based method in terms of robustness/distortion trade-offs. We start by quantifying the quality loss of both schemes.

Author's personal copy



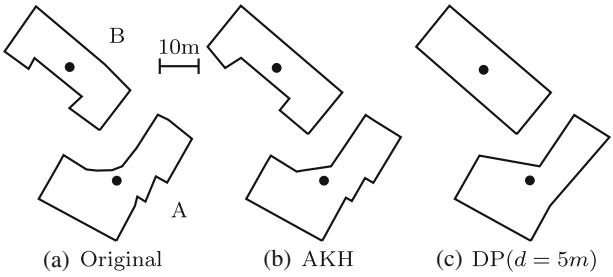
Fig. 6 Pamiers buildings (extract)

4.2 Impact of watermarking on quality

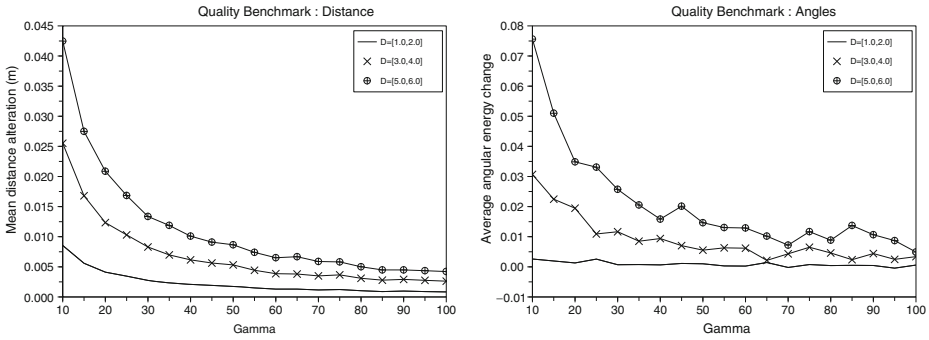
To evaluate the impact of our watermarking algorithm on the Pamiers data set, we applied it with different watermarking ratios $1/\gamma$, with $\gamma \in \{10, 15, \ldots, 100\}$, and different quantization ranges $[d_{\min}, d_{\max}] \in \{[1, 2], [3, 4], [5, 6]\}$. These ranges start from data accuracy (1 m) to the maximum reasonable alteration (6 m).

Average accuracy alteration The impact on the accuracy is displayed on Fig. 8a. Alteration increases when quantization steps increase and when γ decreases. We observe that the average alteration of accuracy is proportional to $(d_{\min} + d_{\max})/\gamma$.

Fig. 7 Examples of filters



Author's personal copy



(a) Mean accuracy (b) Angular energy

Fig. 8 Impact of our watermarking method on quality

For instance, when $[d_{\min}, d_{\max}] = [3, 4]$, a good approximation of the average alteration of accuracy is $0.07.(d_{\min} + d_{\max})/\gamma$. The ratio $1/\gamma$ is not surprising since on the average, $1/\gamma$ polygons are watermarked. Furthermore, the expected alteration for the farthest vertex from the centroid is $(d_{\min} + d_{\max})/4$. The alteration of all the points from a watermarked polygon are proportional to the alteration of this particular vertex. These dependencies can be used by the watermarker to choose the parameters of the marking algorithm: if $d_{\max}$ is obtained as the maximum allowed alteration, and if the target alteration is fixed, one can easily compute γ.

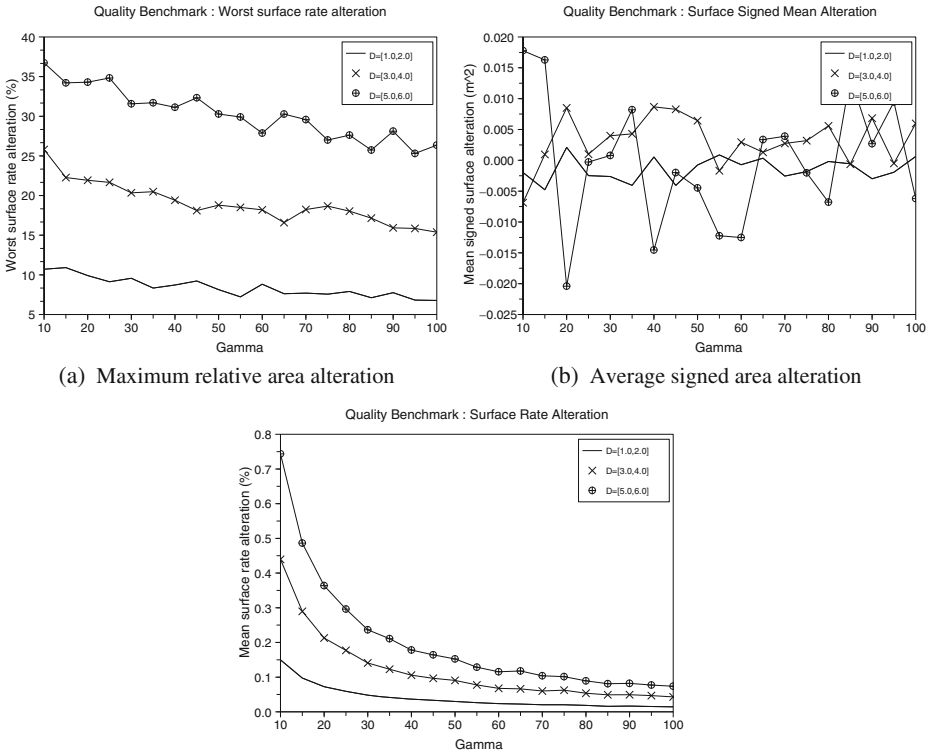
Maximum accuracy alteration The behavior is similar to the one obtained for average accuracy alteration.

Angular quality The variation of angular quality introduced by our method is negligible, as shown in Fig. 8b. Even for the highest quantization steps, $[d_{\min}, d_{\max}] = [5, 6]$, the highest angular energy variation is at most $+0.08$. As a comparison, a weak gaussian noise (deviation $d = 0.2$ m) increases the angular energy by $+6.19$.

Area modification The watermarking impact on polygon areas is displayed on Fig. 9. On Fig. 9a the maximum relative area alteration is shown. Its is proportional to the mean of minimum and maximum quantization steps. On Fig. 9b is displayed the average signed area alteration, which shows that alterations compensate themselves, a good property. On Fig. 9c is displayed the average relative area alteration. Its behavior is similar to average accuracy alteration.

Detection: false positives On Fig. 10 are displayed the detection ratios obtained with different secret keys on a previously watermarked data set. A hundred different keys were tested, among which the one used for watermarking. Detection threshold $\alpha \approx 15\%$ was chosen so that false positive occurrence probability is at most 10^{-4}. It appears that only the secret key $\mathcal{K} = 100$ used in the embedding process leads to a positive detection of mark. This experimentally validates the relevance of the detection algorithm.

Author's personal copy



(a) Maximum relative area alteration

(b) Average signed area alteration

(c) Average relative area alteration

Fig. 9 Impact of our watermarking method on areas

Watermarking speed To evaluate the watermarking rate our implementation can achieve, we used a significantly larger data set, still extracted from the **BD TOPO**™. It consists of the building layer from the Pyrénées Orientales (64), a French department. It contains 243,201 polygons (1,880,405 points and 65 megabytes for the geometry). Using $\gamma = 20$, the whole data set was watermarked in 74 mins 30 s (including database I/O times),i.e. more than 50 polygons per second. The detection process took 1 min 38 s. A simple notebook powered by an Intel Core Duo processor running at 1.6 Ghz with 2 Gb of RAM and a 7,200 rpm hard drive was used for this experiment.

4.3 Comparison with random-noise based watermarking

The random noise scheme is based on the insertion of a pseudo-random noise within the data. It is similar, in most cases, to the schemes that introduce perturbation in the data set without taking into account the shapes of the polygons [18]. The point is that if shapes are expected to be regular, a watermarking algorithm working by altering these shapes is suboptimal in terms of robustness/distortion trade-off. Indeed, most users will correct the shapes so that the regularity aspect is brought back. This operation is very likely to remove the watermark.

Author's personal copy

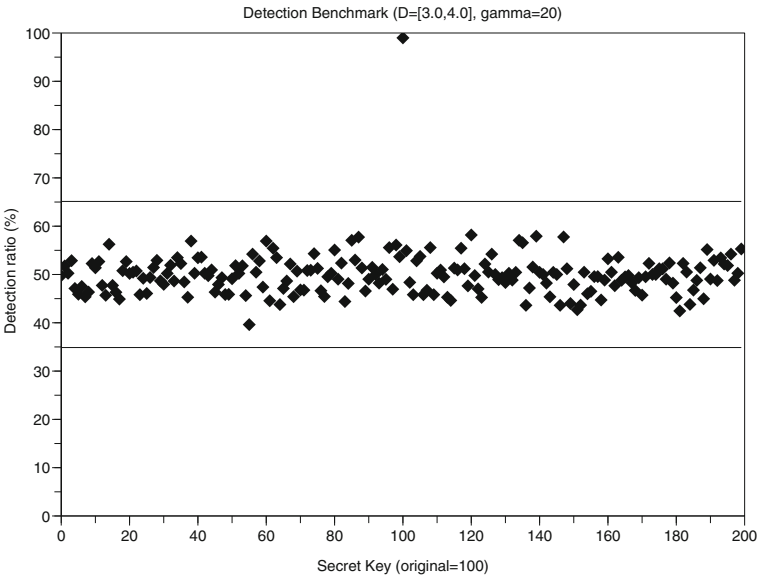


Fig. 10 Undetectability of the mark without the correct key

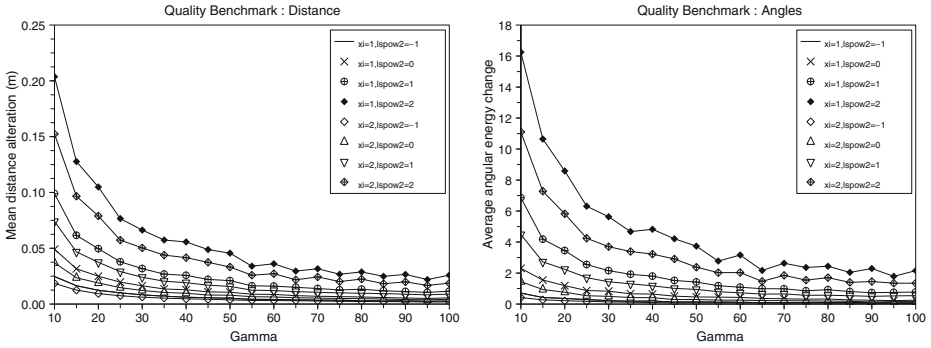
The algorithm consists of looping over all the vertices of the data set. The x and y coordinates of the vertices are watermarked independently. For each coordinate, a PRNG is seeded with its highest significant bits. Only the least significant bits are altered. Their positions and values are determined according to the drawings of the PRNG. This method can be seen as a straightforward extension of the existing AKH [1] scheme in which most significant bits of the coordinates are used as primary keys and least significant bits as alterable attributes.

The quality loss introduced by such a random noise scheme is controlled by three parameters: the watermarking ratio $1/\gamma$, the least highest significant bit $lspow2$ and the number $\xi = 2$ of alterable powers of two. For instance, if $lspow2 = 2$ and $\xi = 2$, the maximum distortion on a coordinate is $2^{lspow2-1} = 2$ and the minimum distortion is $2^{lspow2-\xi} = 1$. Using higher values of ξ enables for extra embedding bandwidth but lead to distortions that are removed by any rounding of the coordinates. In what follows, we compare our scheme (with $d_{\min} = 3$ and $d_{\max} = 4$) with two random noise schemes parameterized with two sets of parameters. For the first one, RNW1, $lspow2 = -1, \xi = 1$; for the second one, RNW2, $lspow2 = 2, \xi = 2$. These values were chosen for the following reasons: RNW1 achieves an average accuracy alteration (compare Fig. 8a and Fig. 11a—both curves decrease as $1/\gamma$ and have a mean accuracy alteration of 0.025 for $\gamma = 10$) very close to the one observed using our method) whereas RWN2 has a maximum alteration on each vertex equals to ours (Fig. 12).

4.4 Robustness against filters and attacks

For all the experiments, we present the detection ratios observed after application of a filter, together with the detection threshold of the mark (computed for a false positive occurrence probability of 10^{-4}). We put side by side the results

Author's personal copy

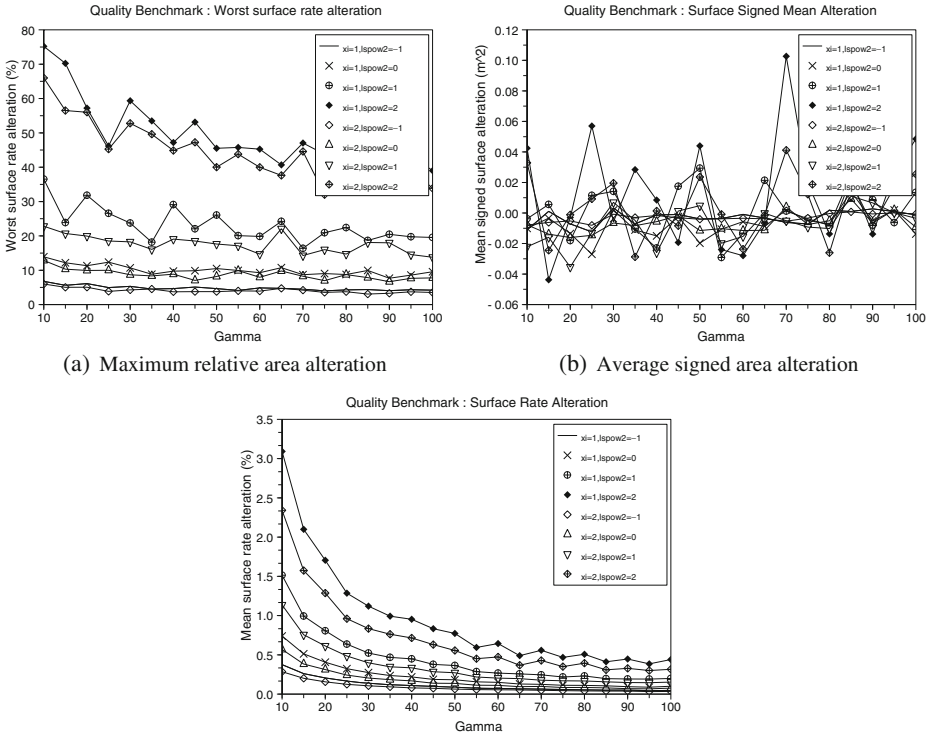


(a) Mean accuracy (b) Angular energy

Fig. 11 Impact of random noise watermarking on quality

obtained using our method, RNW1, RNW2 and the combined scheme presented in Section 4.5.

Squaring Figure 13 shows the detection ratios obtained after squaring. Three different values of the maximum allowed point position alteration were tested, namely $d = 1.0$ (commonly used value), $d = 1.5$ and $d = 2.0$ (very aggressive



(a) Maximum relative area alteration (b) Average signed area alteration

(c) Average relative area alteration

Fig. 12 Impact of random noise watermarking on areas

Author's personal copy

squaring, used here only for validation). For our scheme, squaring has no effect on the detection ratio no matter the watermarking parameters. Even in a worst case scenario, $\gamma = 100$ and $d_{\max} = 2$, the watermark is still detected. For RNW1 and RWN2, the detection ratio is considerably lowered when the data is squared.

Douglas & Peucker simplification The Douglas–Peucker algorithm [6] is a polyline simplification algorithm. It is systematically used by geographic data users with a

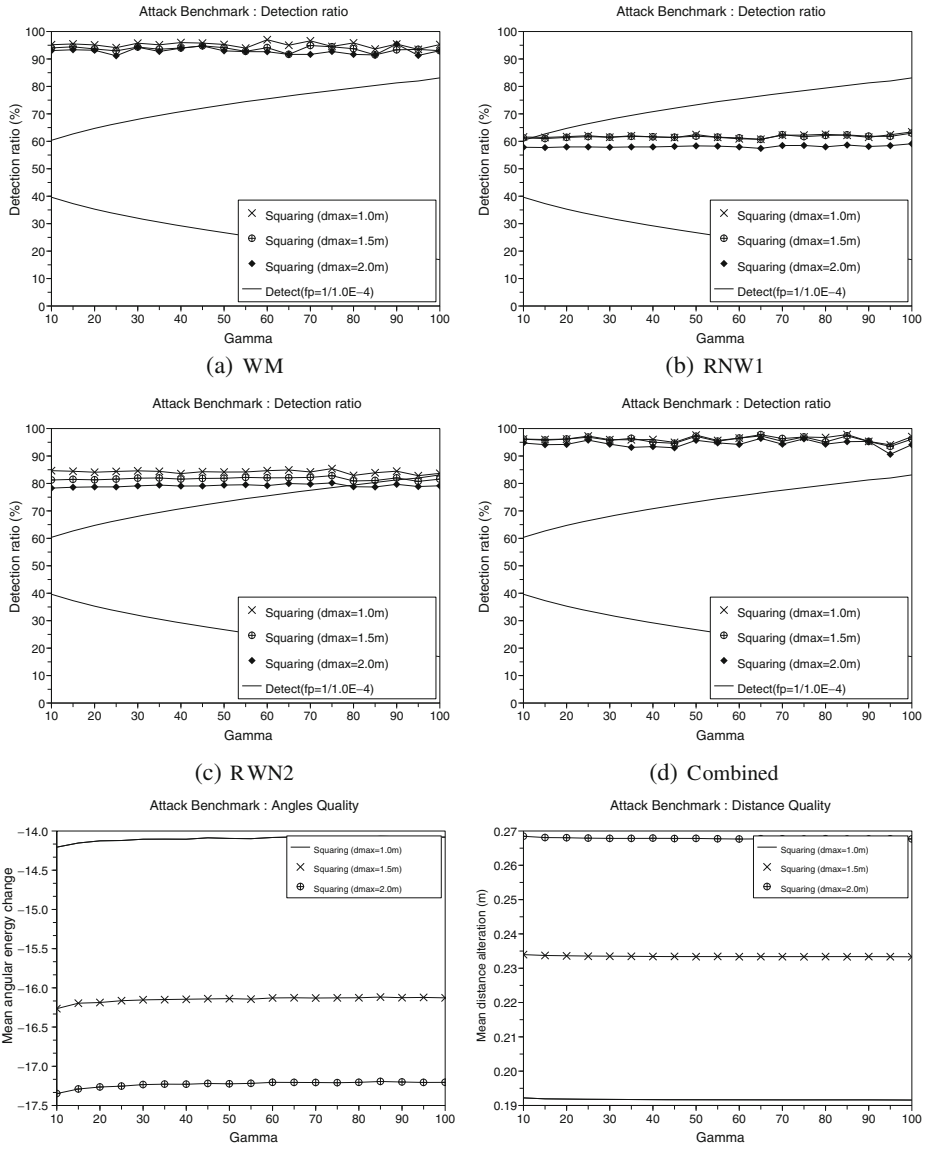


Fig. 13 Squaring filter

Author's personal copy

small factor to filter points of the database. It consists of pruning the points of a polyline whose distance to the line joining the polyline bounds is too small. The distance threshold under which these points are pruned is called d. The higher this threshold, the more aggressive the algorithm is. An example of Douglas–Peucker simplification with $d = 5$ m is given on Fig. 7c. We tested the robustness of our algorithm against Douglas–Peucker filtering for different threshold values. The results are displayed on Fig. 14. Note that our algorithm behaves very well when the

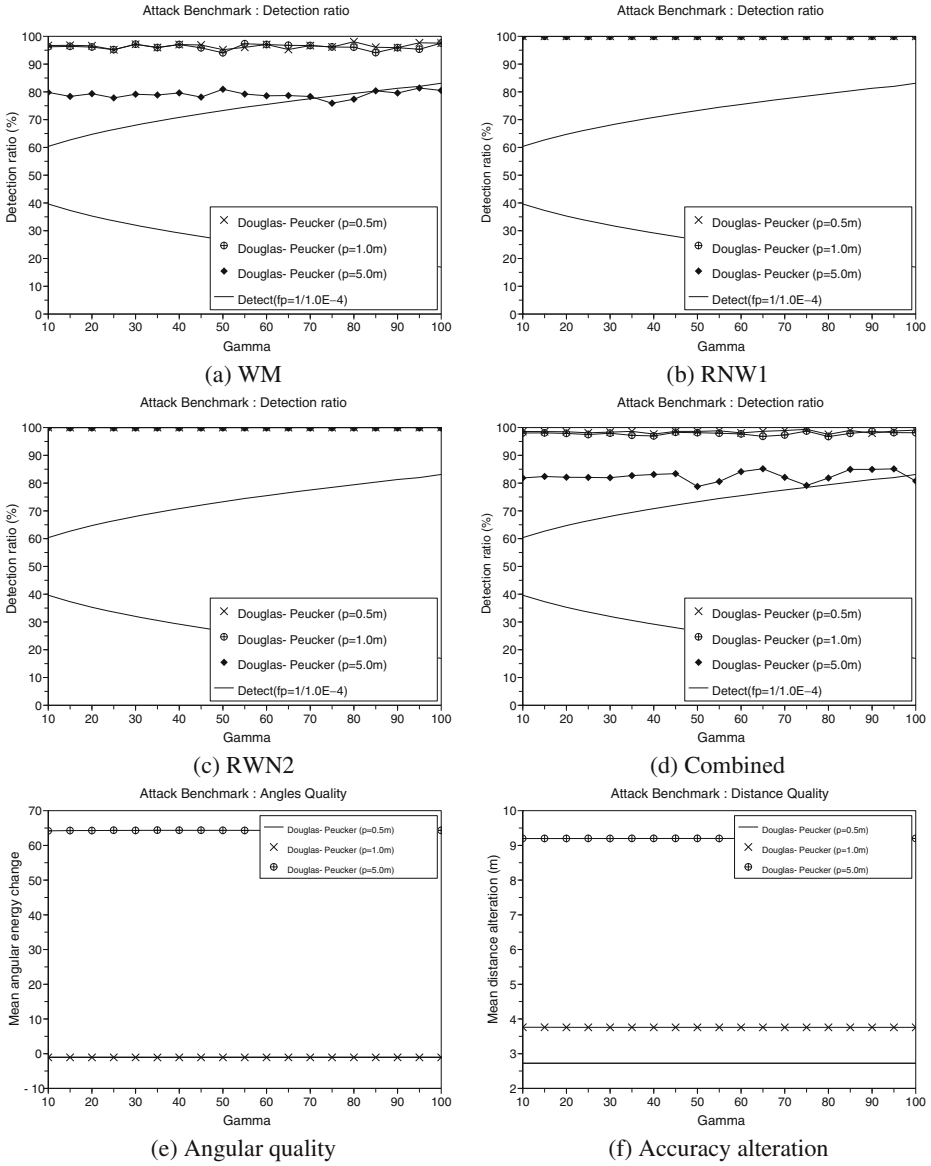


Fig. 14 Douglas–Peucker simplification

Author's personal copy

Douglas–Peucker filtering distance is 1 or 2 m. The detection ratio never falls below the detection threshold. The reason is that the shapes of the polygons are regular enough so that they are invariant to these filters. For a filtering distance of 5 m, the mark is removed in some situations. But such a filtering is very aggressive: it can remove a 4 m $\times$ 4 m room in a building.

Cropping Observe that the whole data set is not necessarily interesting for a malicious user, since profit can still be made from the (illegal) redistribution of a subset. Our method does not require the whole suspect data set to perform detection. This is illustrated on Fig. 15 for our scheme. We randomly generated rectangular subsets of the watermarked data set and performed detection on them. On the x-axis is displayed the number of polygons in a crop while on the y-axis we plotted the number of watermarked and matching polygons against the number of polygons in a crop. We also added the minimum number of matching polygons that must be found to detect the watermark, i.e. the detection threshold. In all cases the watermark is detected. Furthermore, the curves are almost linear after removal of local artifacts. This indicates that the distribution of the watermarked polygons over the data set is nearly uniform. For other schemes, experiments not presented here show that the distribution of watermarked polygons is nearly uniform, so detection is slightly affected by squaring. It is worth noting that single polygons are not protected by our method, and that a set of polygons of reasonable size is required for detection. Indeed, single polygons can only bear one bit of information, as they are made of a few vertices (usually less than 30, conversely to rich BIMs where a single building is made of several thousand of vertices). But in our SIG setting, stealing a single polygon is pointless.

Gaussian noise Gaussian noise was tested with deviations of 20 cm, 60 cm and 1 m. The results are displayed in Fig. 16. For a reasonable noise ($d = 20$ cm or $d = 60$ cm), marks are still detected. When $d = 1$ m, the watermark is removed for a large range of γ. It is worth noting that if a squaring algorithm is applied on a noised data set, the detection ratio of our scheme increases. It even permits the watermark recovery

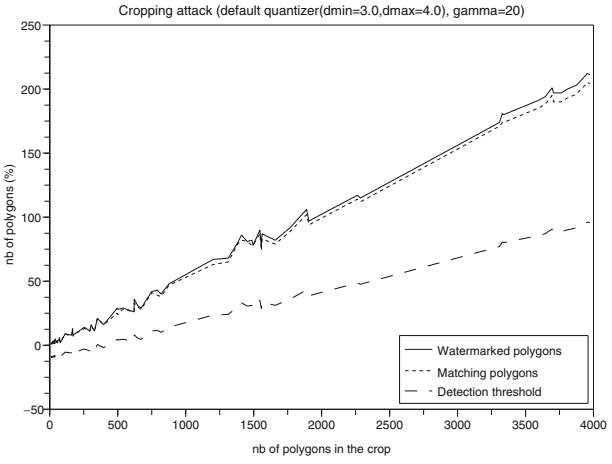


Fig. 15 Cropping attack

Author's personal copy

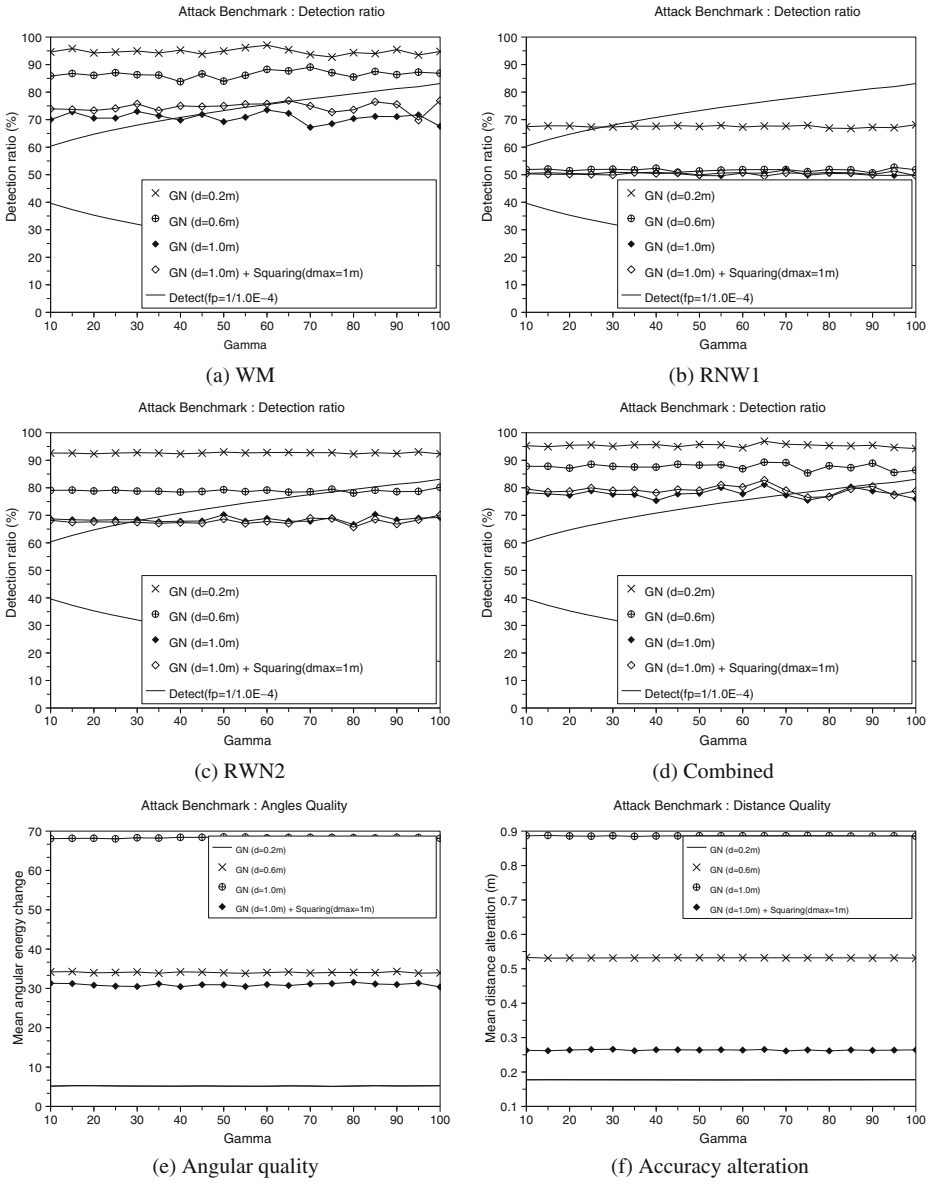


Fig. 16 Gaussian noise

for higher values of γ. RNW1, RNW2 and the combined scheme behave in a similar way.

Applying a new watermark On Fig. 17a, we show how the detection algorithm copes with a new application of the watermarking algorithm with a different key. Different watermarking parameters for the attack were tested: highest quantization steps and lowest γ. For all parameters, the observed detection threshold is such that the first

Author's personal copy

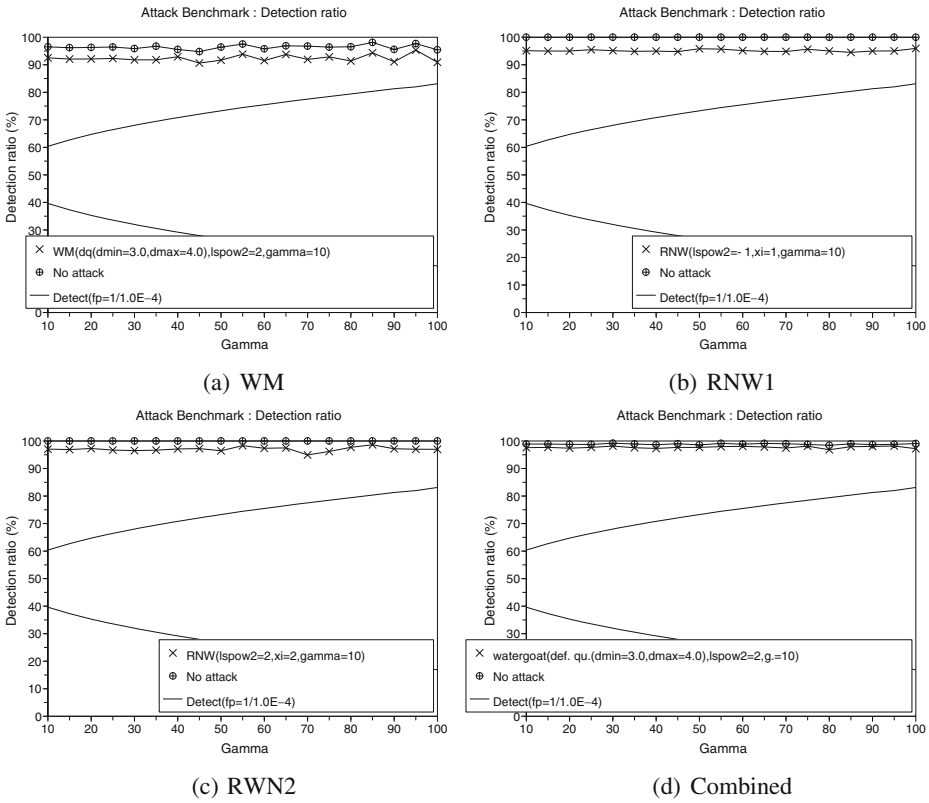


Fig. 17 Over-watermarking

watermark is still detected. Applying another watermark has a limited effect on the first watermark. This is due to the fact that the two watermarks are embedded into distinct polygons. Moreover, on the sparse polygons chosen simultaneously by the two watermark embeddings, the second watermark is the only one to be detected on these polygons. RNW1, RNW2 and the combined scheme show similar behaviors.

Enlarge to rectangle filter For this experiment, we used the Map Generalization Toolbox of OpenJump [21]. The *enlarge to rectangle* filter consists of replacing each building by its minimum bounding rectangle. This filter can be used for drawing a map from the geographical data set, with the desired scale as an argument. In this case, buildings that are invisible for this scale are replaced by their minimum bounding rectangle, enlarged so that they become visible. Larger buildings are not modified. The threshold values are chosen according to the Swiss Society of Cartography [26], e.g. 0.25 mm for width and 0.35 mm $\times$ 0.35 mm for buildings minimum size on maps. The results of the experiments are presented on Fig. 18. They show that replacing all buildings by their bounding rectangle washes out the watermark. When a watermarked data set is used for map generation, the detection ratio decreases as the scale increases. This is not surprising since a high scale implies that many buildings are represented by a minimal area rectangle. So bits embedded

Author's personal copy

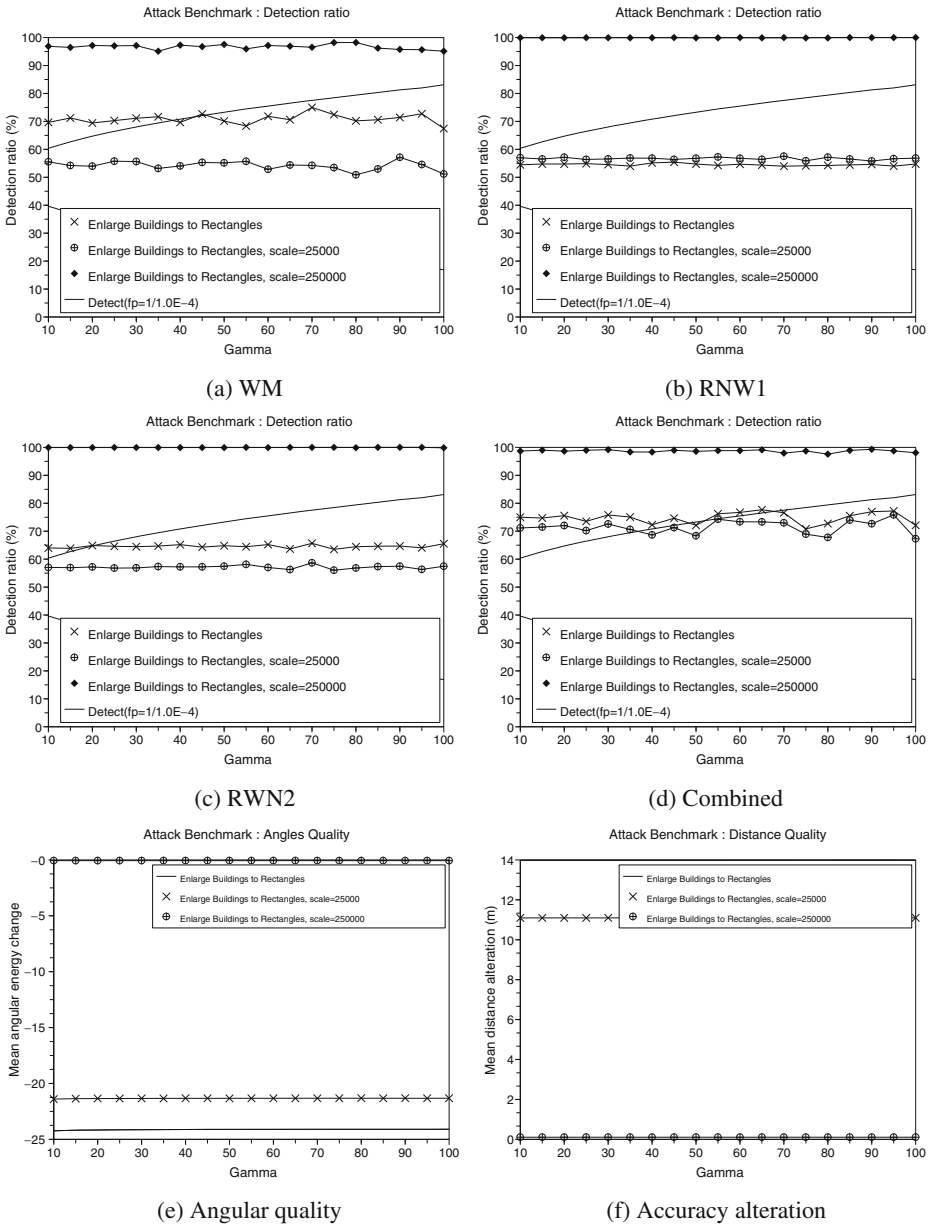


Fig. 18 Enlarge to rectangle filters

in such buildings can not be recovered. Nevertheless, it is interesting to notice that watermarking one polygon out of ten (i.e. $\gamma = 10$) enables to recover marks in a map with scale 25,000. Compared to RNW1 and RNW2, our scheme performs better.

Elongation change filter The elongation change filter consists of multiplying the main length of all polygons with a fixed factor. For experiments presented on Fig. 19,

Author's personal copy

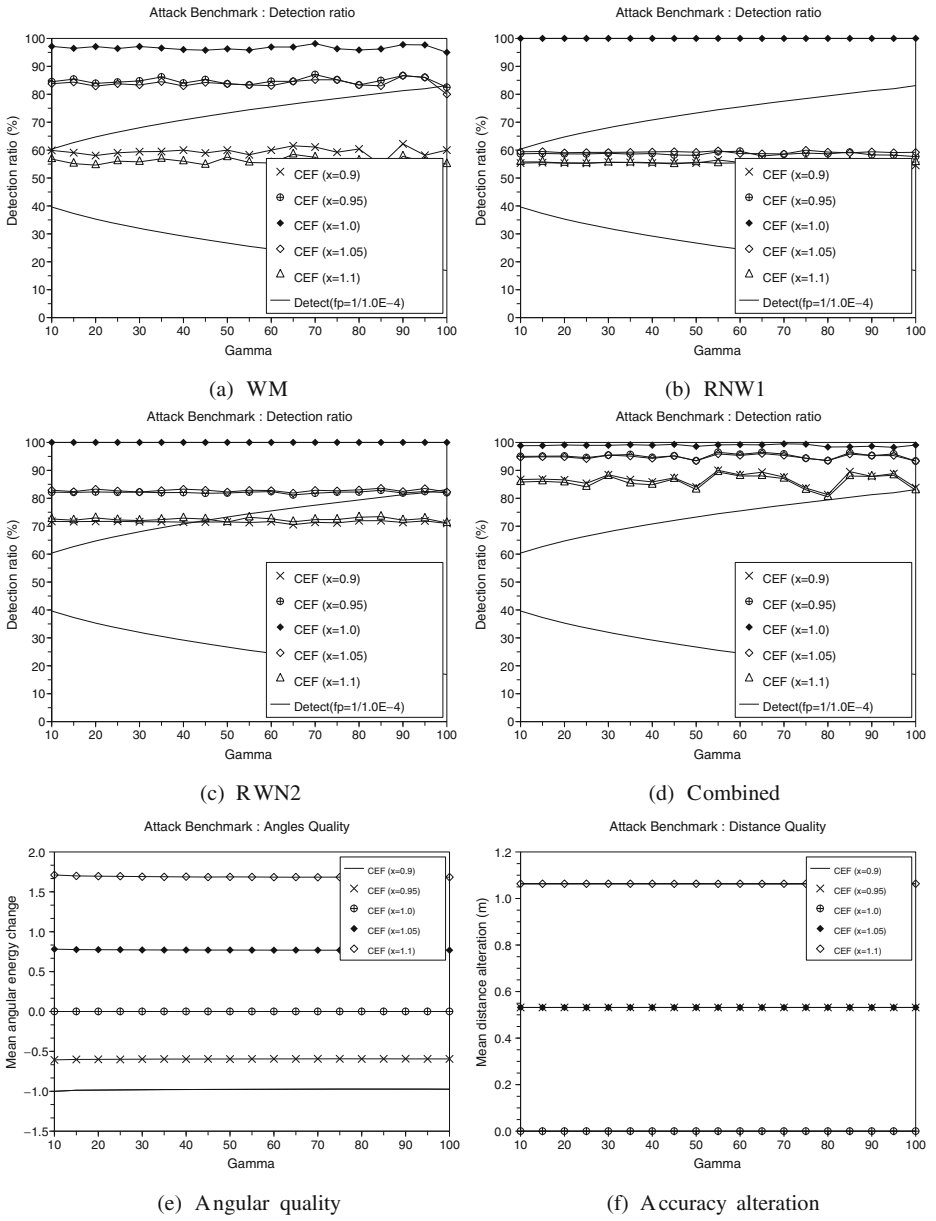


Fig. 19 *Change elongation* filter

ratios 0.85, 0.9, ..., 1.2 were used. Hopefully, when the ratio is 1, the detection ratio is close to 1. If the ratio is between 0.95 and 1.05, the watermark is still detected. This means that an attacker who wants to ensure that the watermark is removed must alter the main length of *all* polygons by more than 5%. Compared to RNW2 and RNW2, our scheme is not as good since elongation ratios 0.9 and 1.1 remove the watermark embedded using our scheme whereas the ones embedded using random

Author's personal copy

noise based scheme are still detectable. Elongation change filter is the worse attack our scheme can encounter since it precisely affects the embedding area used for watermarking. But this attack results in a poor quality data set: its accuracy is lowered on all polygons (see also Section 4.5).

Minimum bounding rectangle filter This filter replaces each polygon by its minimum bounding rectangle. This filter is aggressive for both algorithms, since the watermark is removed on a large range of γ. But, as shown on Fig. 20, our scheme performs slightly better.

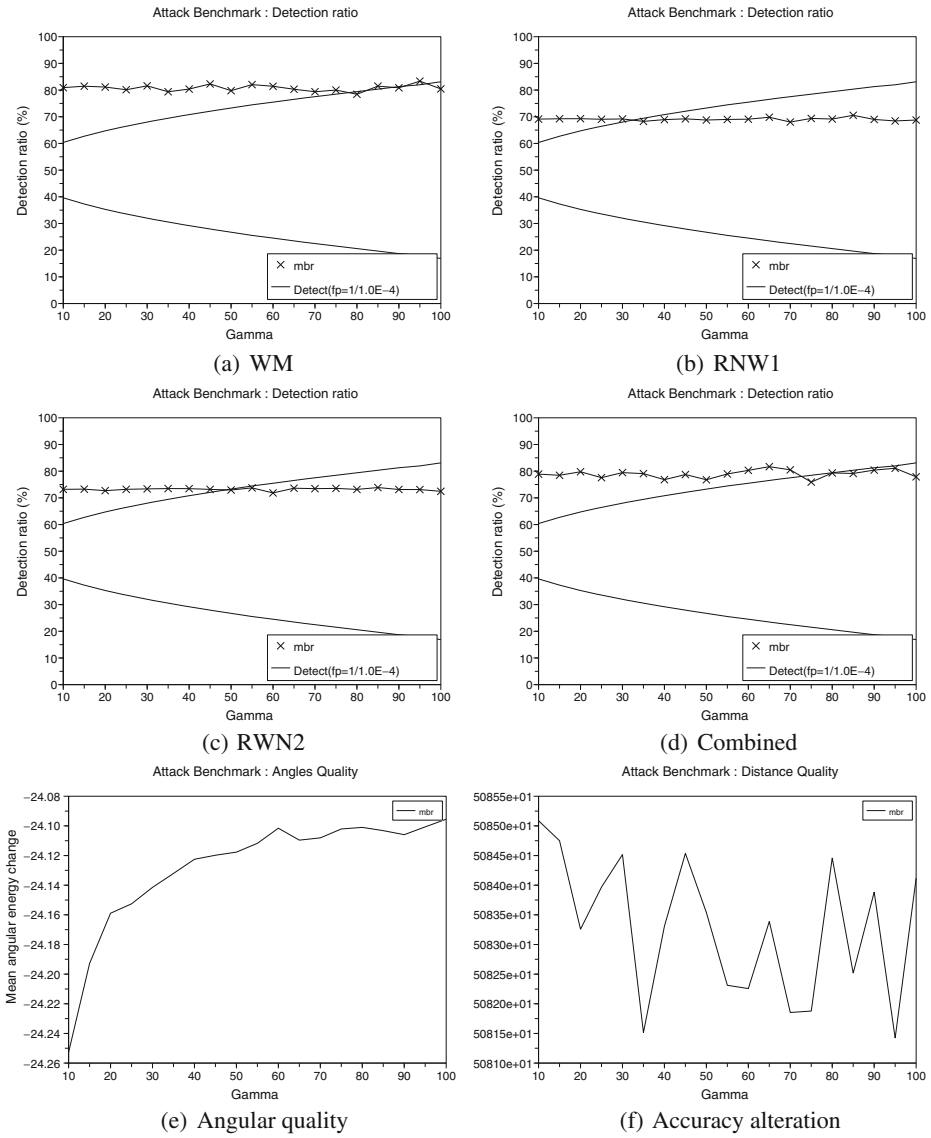


Fig. 20 Minimum bounding rectangle filter

Author's personal copy

Missing reference system Without a reference system, a data set is useless for automatic operation and interoperability. Meanwhile, the suspect data set can be mapped into the original reference system to perform the detection process. This mapping operation is quite common and easy, as soon as (a reasonably large part of) the original data set is available.

Rotation attacks We consider here global rotation of the whole data set, and polygon-wise rotation.

First, if an attacker performs a global rotation on the whole data set without providing a reference system, its topographical data do not correspond any longer to reality, and the data set is useless for a potential buyer (for example, it is now impossible to join this data set with other geolocalized positions used by the buyer). Second, if the suspect data set has a reference system, the performed rotation was just a change of the frame of reference. Since our detection algorithm converts a suspect data set into the reference system chosen at watermarking time (Algorithm 2, line 5), this attack is defeated. Point displacement coming from round-off errors during conversion can be analyzed as a simple Gaussian noise attack with low amplitude, as analyzed in the above paragraph.

Our method is also invariant to polygon-wise rotations since the expansion coefficient is defined relatively to polygon orientation and not to a particular reference system.

Other attacks Many other kinds of attacks may be envisioned. The mixing attack consists of mixing a portion of a watermarked data set with an unwatermarked one. This attack is expected to lower the detection ratio but not render the watermark unreadable.

The aliasing attack consists of switching the edges of the polygon with zig-zag shaped sequences of edges. The goal is to modify the orientation of the polygon. But it implies adding a huge number of extra (fake) points to the database and altering the overall shapes of polygons. Furthermore, the artifact needs to have an amplitude exceeding the angle tolerance ratio.

We also observe that squaring the data before applying our watermarking method enhances slightly the detection ratio, beside increasing the data set quality.

4.5 Discussion

Table 3 sums up robustness experiments. Robustness results for our method are presented in the WM column. A Yes/No status indicates whether a method achieves robustness or not. If robustness is achieved only for a given watermarking period γ, the corresponding threshold is specified. However, depending on the data set, robustness could also be achieved for higher values of γ. In the second and third columns are shown the average quality loss introduced by the attacks. These must be put into balance with the distortion introduced by watermarking algorithms (see Sections 4.2 and 4.3).

As we expected, our method resist all kinds of squaring, including the most destructive ones. On the contrary, watermarks embedded using random-noise based algorithms are washed out. Our method shows also robustness against the majority of attacks. The fact that GN(d = 1 m) and DP(d = 5 m) erase the watermark must be

Author's personal copy

Table 3 Robustness and impact of watermarking

Filter	Precision alt.	Angular energy	WM	RNW1	RNW2	Combined
SQ (d = 1 m)	0.19	−14.1	**Yes**	No	**Yes**	**Yes**
SQ (d = 1.5 m)	0.23	−16.2	**Yes**	No	$\gamma < 90$	**Yes**
SQ (d = 2 m)	0.27	−17.2	**Yes**	No	$\gamma < 80$	**Yes**
DP (d = 1 m)	N/A	−1.3	**Yes**	**Yes**	**Yes**	**Yes**
DP (d = 2 m)	N/A	−0.3	**Yes**	**Yes**	**Yes**	**Yes**
DP (d = 5 m)	N/A	68.1	$\gamma < 70$	**Yes**	**Yes**	$\gamma < 100$
CA	0	0	**Yes**	**Yes**	**Yes**	**Yes**
GN (d = 0.2 m)	0.18	6.19	**Yes**	$\gamma < 30$	**Yes**	**Yes**
GN (d = 0.6 m)	0.53	40.00	**Yes**	No	$\gamma < 75$	**Yes**
GN (d = 1 m)	0.89	78.52	$\gamma < 65$	No	$\gamma < 30$	$\gamma < 75$
OW	–	–	**Yes**	**Yes**	**Yes**	**Yes**
ETR	N/A	−6.53	$\gamma < 40$	No	$\gamma < 25$	$\gamma < 50$
ETR (scale = 25,000)	N/A	−6.06	No	No	No	$\gamma < 40$
ETR (scale = 250,000)	N/A	−0.03	**Yes**	**Yes**	**Yes**	**Yes**
CE (scale = 0.90)	1.06	0.16	No	No	$\gamma < 45$	**Yes**
CE (scale = 0.95)	0.53	0.02	$\gamma < 95$	No	**Yes**	**Yes**
CE (scale = 1.0)	0	0	**Yes**	**Yes**	**Yes**	**Yes**
CE (scale = 1.05)	0.53	0.09	$\gamma < 95$	**Yes**	**Yes**	**Yes**
CE (scale = 1.10)	1.06	0.27	No	No	$\gamma < 50$	**Yes**
MBR	N/A	−6.6	$\gamma < 75$	$\gamma < 35$	$\gamma < 50$	$\gamma < 75$

SQ squaring, *DP* Douglas–Peucker simplification, *CA* cropping attack, *GN* Gaussian noise, *OW* over-watermarking, *ETR* enlarge-to-rectangle, *CE* change elongation, *MBR* minimum bounding rectangle

put into balance with the quality losses these attacks introduce. They tremendously reduce the quality of the data set. Furthermore, it behaves well against *change elongation* filters since the length of polygons must be increased or decreased by more than 10% to ensure watermark removal. But this remains the weak point of our method since the embedding area is directly affected.

Compared to random-based schemes, our algorithm performs better in the majority of cases. Whereas RWN1 and RWN2 are more robust against the *change elongation* filter, they have a significantly higher impact on quality compared to our method. In that sense, the experiments show that our algorithm achieves a very good distortion/robustness trade-off.

In the last column of the table, we combined our scheme and RNW2 into an *expand-and-translate* scheme. First we apply our scheme and then apply RNW2 on the centroid of the polygon (left invariant by the first step). We obtain a modification of the coordinates of the centroid which is used to translate the polygon. Obviously, this combined scheme introduces a larger quality loss but it achieves robustness for all the filters/attacks used in the paper. It can be preferred to our basic scheme if robustness is to supersede quality for a specific application.

5 Related work

In our database approach, polygons are stored in a relational database management systems enriched with geographical features. Since polygons are stored in relational tables, state-of-the art watermarking algorithms for relational databases might have been used. It happens that least significant bits modifications used in previous works

Author's personal copy

[1] can not be mapped onto our geographical setting. It is easy to alter least significant bits of points of the map but the angular quality is not taken into account (on the contrary, least significant bits methods may perform well on simple points databases, like point-of-interest data sets in use in GPS viewers). Methods described in [8, 25] allow for the description of usability queries to be preserved by watermarking. But they either focus on basic numerical aggregates like `SUM` queries [8], which are not rich enough to represent angular constraints, or based on a trial and error method to handle generic black-box queries [25]. Using the latter method, it might not be possible to reach a valid set of alterations since no search strategy is defined.

Despite that state-of-the art is very rich on watermarking still images which can be directly applied to rasterized/bitmap maps, fewer works were carried on watermarking vector maps. A complete study of vector map watermarking [18] has been recently published which divides this field into three categories: spatial domain watermarking, transform domain (DCT, DWT, ...) watermarking and 3D meshes adapted algorithms. Spatial domain watermarking consists of modifying directly the coordinates of the vertices of the data set. Patch-based techniques [13, 19, 23, 29] are based on a decomposition of the data set into patches in which bits are embedded by vertices translation [19] and/or data distribution within a patch [13, 23, 29]. Such schemes are sensitive to patch decomposition that can vary when the data set is cropped. Schemes of [13, 23] create detectable nodes aggregations; [19] is not blind and [29] does not respect the shape of smooth objects (including squared buildings).

Vertices addition based [10, 17, 22] techniques are the best from a quality viewpoint. They consist of interpolating the existing edges of the data set with fake points. In the context of buildings where the shapes are regular, such interpolations are easily detected and removed by simplification algorithms. In our context, these schemes are not robust enough. The least-significant bit watermarking method presented in [28] preserves accuracy but is very sensitive to vertices addition, which destroys synchronization.

In [24], a high watermarking capacity algorithm for vector maps is introduced. It is based on a previous work on 3D meshes watermarking by the same authors. It is robust against common geographical filters like Douglas–Peucker simplification algorithm. It is based on a decomposition of the database into patches and by moving points of a common patch into a subpatch to embed the watermark. Nevertheless, efficient attacks erasing the watermark without destroying the quality of the data set can be easily planned, as the method requires known synchronization points. Another approach followed by [4] synchronizes the watermark based on a triangular mesh decomposition. Within secretely chosen groups of adjacent triangles, a vertex is moved in order to hide a bit of information.

Our method shares a common skeleton with the popular AHK algorithm [1]. But it differs on several aspects: it does not require primary keys, and do not use high significant bits to replace them in a straightforward manner (instead the centroid is used); the bit embedding operation is not a least significant bit embedding, which is very fragile against rounding, but a quantization method.

6 Conclusion

In this paper we presented a blind watermarking algorithm for polygon data sets. It is well suited to building layers of geographical data sets since watermarks are

Author's personal copy

invariant through aggressive geographical filters applied by data users, including squaring and replacement by bounding boxes. We experimentally showed that it is difficult for an attacker to erase the watermark without paying an extra quality fee, compared to watermarking. The algorithm has been implemented into an open database watermarking framework [16] and is available online [15]. We are currently working on designing algorithms for other layers of geographical data sets. A real challenge we are faced with is the interactions between the different layers, for example in a complex BIM. Indeed, watermarking algorithms must be adapted to the data; there is no unique solution. Even if we know how to perform watermarking and detection on a single layer, it is challenging to orchestrate several algorithms on several layers so that resulting watermarked layers remain consistent.

Acknowledgements We would like to thank Eric Grosso for his technical support during our experiments and also for our fruitful discussions.

References

1. Agrawal R, Haas PJ, Kiernan J (2003) Watermarking relational data: framework, algorithms and analysis. VLDB J 12(2):157–169
2. Airault S (1996) De la base de données à la carte: une approche globale pour l'équarrissage de bâtiments (in french). Rev Int Géomatique 6(2–3):203–217
3. Azhar S, Hein M, Sketo B (2008) Building information modeling (bim): benefits, risks and challenges. In: Proceedings of the 44th ASC annual conference (on CD ROM), Auburn, Alabama, 2–5 April 2008
4. Bazin C, Le Bars J-M, Madelaine J (2007) A blind, fast and robust method for geographical data watermarking. In: Proceedings of the 2nd ACM symposium on information, computer and communications security, ASIASCCS'07, pp 265–272
5. COGIT Laboratory (2007) Geoxygene: an open framework for the development and deployment of gis applications
6. Douglas D, Peucker T (1973) Algorithms for the reduction of the number of points required for represent a digitized line or its caricature. Can Cartogr 10(2):112–122
7. Duchêne C, Bard S, Barillot X, Ruas A, Trevisan J, Holzapfel F (2003) Quantitative and qualitative description of building orientation. In: Fifth workshop on progress in automated map generalisation, ICA, commission on map generalisation, April 2003
8. Gross-Amblard D (2003) Query-preserving watermarking of relational databases and XML documents. In: Proceedings of the 22nd symposium on principles of database systems, PODS'03. ACM, pp 191–201
9. Hoeffding W (1963) Probability inequalities for sums of bounded random variables. J Am Stat Assoc 58(301):13–30
10. Huber WA (2002) GIS and steganography - part 3: vector steganography. http://www.directionsmag.com/article.php?article_id=195
11. Institut Géographique National (2002) BD TOPO - descriptif technique (in French). http://www.ign.fr/telechargement/MPro/produit/BD_TOPO/JT_Agglo/DT_BDTOPOPays_1_2.pdf
12. Institut Géographique National (2006) GéoPortail (visited 03/20/2007). http://www.geoportail.fr
13. Kang HI, Kim KI, Cho JU (2001) A vector watermarking using the generalized square mask. In: Proceedings of the international conference on information technology: coding and computing, pp 234–236
14. Katzenbeisser S, Petitcolas FAP (2000) Information hiding, techniques for steganography and digital watermarking. Artech House
15. Lafaye J, Béguec J (2007) Watermarking plugin for OpenJump. http://cedric.cnam.fr/~lafaye_j/index.php?n=Main.WaterGoatOpenJumpPlugin
16. Lafaye J, Gross-Amblard D, Constantin C, Guerrouani M (2008) Watermill: an optimized fingerprinting system for databases under constraints. IEEE Trans Knowl Data Eng (TKDE) 20(4):532–546

Author's personal copy

17. Vazquez CML (2003) Method of inserting hidden data into digital archives comprising polygons and detection methods. US Patent no. 20030208679
18. Niu X, Shao C, Wang X (2006) A survey of digital vector map watermarking. Int J Innov Comput Inform Control 2(6):1301–1316
19. Ohbuchi R, Ueda R, Endoh S (2002) Robust watermarking of vector digital maps. In: Proceedings of the international conference on multimedia and expo, ICME '02, vol 1, pp 577–580
20. Ohbuchi R, Ueda H, Endoh S (2003) Watermarking 2D vector maps in the mesh-spectral domain. In: Proceedings of shape modeling international, SMI'03, pp 216–228
21. OpenJUMP (2007) OpenJUMP—the free, java based and open source geographic information system for the world. http://www.openjump.org
22. Park KT, Kim KI, Kang HI, Han S-S (2002) Digital geographical map watermarking using polyline interpolation. In: PCM '02: proceedings of the third IEEE Pacific rim conference on multimedia. Springer-Verlag, London, UK, pp 58–65
23. Sakamoto M, Matsuura Y, Takashima Y (2000) A scheme of digital watermarking for geographical map data. In: Symposium on cryptography and information security, Okinawa, Japan, January 2000
24. Schulz G, Voigt M (2004) A high capacity watermarking system for digital maps. In: Proceedings of the ACM workshop on multimedia and security, MMSEC'04. ACM Press, New York, NY, pp 180–186
25. Sion R, Atallah MJ, Prabhakar S (2004) Rights protection for relational data. IEEE Trans Knowl Data Eng 16(12):1509–1525
26. Swiss Society of Cartography (2005) Vol. 17: topographic maps—map graphic and generalization
27. The International Hydrographic Organization (IHO) (2001) Specifications for chart content and display aspects of ECDIS, 5th edn. IHO
28. Voigt M, Busch C (2002) Watermarking 2D-vector data for geographical information systems. In: Delp EJ, Wong P (eds) Proceedings SPIE-IS&T electronic imaging, SPIE, vol 4675, pp 621–628
29. Voigt M, Busch C (2003) Feature-based watermarking of 2D vector data. In: Delp EJ, Wong PW (eds) Proceedings SPIE-IS&T electronic imaging, SPIE, vol 5020, pp 359–366



Julien Lafaye graduated from École polytechnique (Paris, France) in 2003. He holds a MSc degree from University Paris XI-Orsay (2004) and a PhD degree from Cnam Paris (2007), both in computer science. His research interests span from databases to security, with a specific focus on watermarking. His PhD dissertation is centered around constraint-preserving watermarking, with applications on relational, geographical and XML databases. He also worked on computational complexity related aspects about watermarking. He is now working for an information technology company as an expert consultant.

Author's personal copy



Jean Béguec holds a MSc degree in Geomatic from University Paris-Est Marne-la-Valléée (France). After his training subject of geographical data watermarking at COGIT Lab, Saint Mandé (France), he worked for an information technology company and now work for the French administration.



David Gross-Amblard is an assistant professor at Cedric Lab., CNAM-Paris and Le2i Lab., university of Bourgogne, France. He received MSc and PhD degree in computer science from the University of Paris XI, Orsay, France in 1996 and 2000 respectively. His research area is the application of finite model theory to various domains, including query approximation in spatial databases and query preservation in the watermarking context.

Author's personal copy



Anne Ruas is a State Engineer of the Bridges, head of the COGIT Lab., Saint Mandé (France). She received a PhD degree in GIS from University Paris-Est Marne-la-Vallée (France) in 1999. Her research area spans from cartography and object oriented GIS to multi-agent systems and spatial analysis.