Invisible Graffiti on your Buildings: Blind Watermarking of Geographical Databases *

Julien Lafaye[1], Jean Béguec[1,3], David Gross-Amblard[1,2] and Anne Ruas[3]

[1] Laboratoire CEDRIC, Spécialité Informatique – CC 432, Conservatoire national des arts & métiers, 292 rue Saint Martin, 75141 PARIS Cedex 3, France
[2] Laboratoire LE2I, Université de Bourgogne, Faculté des Sciences Mirande, Aile de l'ingénieur, BP 47870 21078 DIJON Cedex, France
[3] Laboratoire COGIT, Institut Géographique National (IGN), 2/4 Avenue Pasteur 94 165 SAINT MANDE Cedex, France

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 transformation, as well as deliberate removal attempts, e.g. by noise addition, cropping or over-watermarking. The impact on the quality of the datasets, 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.

1 Introduction

Geographical Information Systems (GIS) have existed for more than 40 years but their application domain is much wider nowadays, 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 [1]. Most geographical applications rely on an underlying vectorial spatial database (points, polylines and polygons). —longonly—

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. —longonly— Due to the ease of reproduction of digital media, unauthorized copy and use threaten geographical data providers. Hence, protecting the intellectual property (IP) of rights owner is 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 document.

* Work supported by the ACI Sécurité & Informatique TADORNE grant (2004-2007)

Embedded marks must be robust against removal attempts. In this paper, we propose a robust watermarking method for polygonal datasets.

To embed the watermark, the data has to be altered. What might sounds as a drawback is common to most watermarking methods [8]. 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 dataset 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 dataset). 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 legibility. Some others focus on objects like forests, cliffs and shallows for which precise borders can be somewhat difficult to define. But many 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 [17]. Finally, accurate datasets must conform with some standard reference system for interoperability purposes (e.g. the World Geodetic System – WGS84, which is the GPS reference system).

It turns out that most of the vectorial content of geographical databases consists of building polygons (80% on the professional dataset used in the experiments), which constitute the primary focus of this work. Real world requirements entail specific constraints within the dataset, e.g. right angles between walls of buildings. Thus, geographical data is often corrected so that buildings with right angles are mapped to polygons with right angles in the dataset. This correction, called *squaring*, increases the angular quality of the dataset. It is also very invasive since potentially each point of the dataset is moved. Experiments show that it also tends to increase data accuracy. For these reasons, we model the quality of a dataset by means of (1) its accuracy and (2) its angular quality.

To achieve robustness of a watermarking method, one has also to precisely forecast the kind of attacks the data may be subject to. Geographical dataset watermarking is challenging since most users systematically apply treatments to the data. Squaring is such a treatment. Moreover, the huge volume of these datasets and their updates over years motivates the use of blind watermarking algorithms, i.e. methods that do not require the original dataset to perform watermark detection. There exist several recent techniques for databases watermarking. Some of them apply to relational databases [2], others to geographical datasets [15, 13]. None of them takes into account the squaring transformation, which is nearly systematically applied by data users.

It this paper, we propose an effective method for building watermarking that is robust against geographical transformations (squaring, simplification, smoothing) 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 dataset (point accuracy and angular quality) and not introducing topological errors (overlapping polygons). The scheme is blind (the original dataset is not required for

detection) and do not assume that primary keys identifying polygons are available. —longonly—

A classical skeleton [2] 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 well chosen high significant bits 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. This transformation is invariant through squaring. It is also robust against other transformations we present later in the paper. Any naïve user or malicious attacker has to tremendously reduce accuracy and/or angular quality of the dataset to erase the watermark.

Outline After a description of watermarking basics, a simple model for buildings databases and a definition for data quality are presented 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 Section 5 and Section 6 concludes.

2 Preliminaries

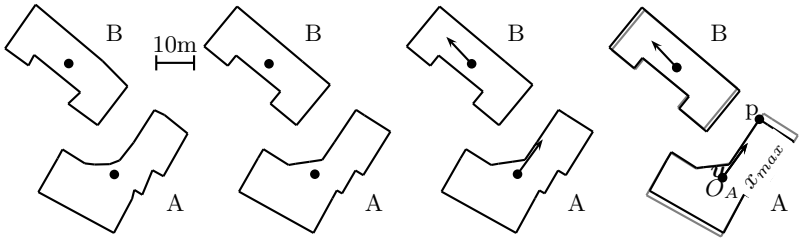
2.1 Quality of Geographical Data

END

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 described by the list of its points .Two polygons taken from a real dataset are shown on Fig. 1(a). 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 P_i. It is always provided with some reference system otherwise it is of no use for automatic operations —longonly—.

We do not rely on the order of polygons within the dataset, 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. —longonly—

The (economical) value of a dataset (R, DB) is correlated with its *mean accuracy*, its *maximum accuracy* and its *angular quality*. The *mean accuracy* mean value of the distance between a point of a building and its corresponding point in the dataset; the *maximum accuracy* is maximum value of the distance between a point of a building and its corresponding point in the dataset. The *angular quality* citeAirault is defined as the opposite of its angular energy. The



(a) Raw data (b) Squared data (c) Orientation (d) Expansion

Fig. 1. Buildings polygons

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 $\pi/4$.

END

2.2 Geographical Filters

Datasets are likely to undergo some transformations by legitimate users before being actually used. These transformations can be:

Squaring: points are moved so that almost-right angles become exact right angles. This transformation increases angular quality and experiments show that it also tends to increase accuracy. An example is given on Fig. 1.

Simplification: points that do not bring extra information to the dataset are pruned. A simple yet often used simplification method is the Douglas-Peucker algorithm [5] which consists of pruning points that are nearly aligned with two other points of the polygon;

Smoothing: moving points of the polylines so that they do not present sharp shapes (mainly used for roads); —longonly—

END

2.3 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 dataset (R, DB) , a secret key $\mathcal{K}$ and some tuning parameters, and produces a watermarked dataset $(R, \mathcal{DB}_{\mathcal{K}})$. The aim of the detector is, given a suspect dataset (R', DB') and the secret key $\mathcal{K}$, to decide whether this dataset holds a watermark or not. A watermarking procedure is said to be blind if the original dataset is not needed by the detector $\mathcal{D}$. It is said to be robust if it detects marks in altered watermarked datasets. It is well known that any robust

watermarking method must alter the data. 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 datasets, 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 dataset must adopt a common reference system and limit the quality loss so that profit can still be made from the attacked database.

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 dataset, 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 modification 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. 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(i.e. expand with a ratio < 1) along its orientation. This orientation is computed relatively to the centroid (see Fig. 1(c)), and represents the majority weighted angle among edges directions. We present its computation in Section 3.3. For instance, for a rectangular shape, the 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. 1(b)): 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. —longonly—

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 [2] 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 secure 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. But it appears purely random to anyone who does not possess this seed (an attacker may easily compute id_i, but $\mathcal{K}$ remains unknown).

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.

Example 1. An example of our watermarking method applied on polygons A and B is shown on Fig. 1(d). Original shapes are shown in black and watermarked ones in grey. First, we compute the centroid of A and B, obtaining $O_A = (293, 155)$ and $O_B = (171, 447)$. To form unique identifiers id_A and id_B, we concatenate the two high significant digits of each coordinate, obtaining $id_A = 2915$ and $id_B = 1744$. Choosing these two digits is correct under the hypothesis that any reasonable alteration is below 10 meters and that the typical distance between any two buildings is more than 10 meters (this example considers decimal base while our algorithm considers binary). 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 $\boldsymbol{u}$ of A and find the vertex p such that $\boldsymbol{u}.Op$ 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.

END

3.2 Computing Robust Identifiers

END

As a robust identifier, we use the highest significant bits of the centroid of the polygon. The centroid is the center of mass of the polygon and is computed using the formulae from [3]. Centroids of polygons A and B are represented as black dots on Fig. 1(a) and 1(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 it. On the other hand, h must be small enough so that two adjacent polygons do not share the same identifier. For space reasons, we omit the discussion on a proper choice of h. We present a systematic way to compute h in [**?**]. —longonly—

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.

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

3.3 Computing Polygon Orientation

We define the main orientation $\boldsymbol{u}$ of a polygon as the maximum weighted orientation of its edges. For instance, if only e_1 and e_2 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 dataset. 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 $2.\pi/k < \varepsilon$. 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.

Example 2. We illustrate the orientation computation algorithm on polygon A of Fig. 1(b). The number of classes is set up to 10. We got the following repartition of edges: $\{b_0 : 3, b_3 : 8, b_8 : 6\}$ and the corresponding weights: $\{b_0 : 9.02, b_3 : 62.4, b_8 : 45.2\}$. —longonly— The highest cost bucket is bucket 3, 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 3 gives 0.96 rad. END

END

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} along the orientation $\boldsymbol{u}$ 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 after this transformation). Hence, we choose to alter all lengths along the orientation $\boldsymbol{u}$ so that most angles are preserved. Defining by $\boldsymbol{v}$ the unary vector such that $(0, \boldsymbol{u}, \boldsymbol{v})$ 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 the basis $(0, \boldsymbol{u}, \boldsymbol{v})$;
– compute the main length $x_{max} = \max_i |x_i|$;
– expand all points coordinates along direction $\boldsymbol{u}$ so that x_{max} becomes one of the values $\{x^0_{max}, x^1_{max}\}$ coding a watermark bit 0 or 1. This later operation on x_{max} is known as quantization (see below).

This paragraph details 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 and define the quantized version of x_{max} as $x'_{max} = k_0.d$. To use 1-quantizers, one should choose $k_r = x_{max}/d - 1/2$ and $x'_{max} = k_0.d + d/2$. The quantization process is illustrated on Fig. 2. END

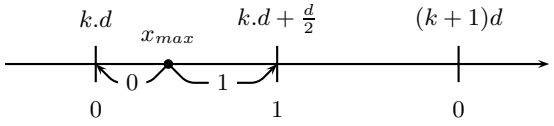


Fig. 2. Encoding 0 or 1 into the main length x_{max} using quantization

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

The expansion is such that the maximum distortion on a vertex of a polygon is at most d. Remark that this distortion can be reached only for the vertices that are the farthest to the centroid along $\boldsymbol{u}$. On the average, and for these points, the actual distortion is $d/2$.

3.5 Watermarking Algorithm

Watermarking The complete algorithm is presented in Alg. 1. Let $1/\gamma$ be the target ratio of watermarked polygon. It is a parameter of the algorithm. For each polygon of the dataset, 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.

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 dataset, 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 [**?**]. —longonly—

END

Discussion Using this method, the watermark is spread almost uniformly over the dataset. This process being controlled by a secret key, it is impossible to find the exact locations of polygon expansions, assuming PRNG are 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 dataset. They cannot be fixed arbitrarily for all applications but the following rules are always valid:

- there is an unavoidable trade-off between quality alteration ($\gamma \uparrow, d_{min} \downarrow, d_{max} \downarrow$) 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 dataset 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 dataset is sold under the agreement of an accuracy β_2, then d_{max} must be chosen so that $d_{max} < 2(\beta_2 - \beta_1)$;
- the allowed alteration on the building, i.e. d_{max} must be higher than the typing accuracy of the dataset. Below this value, alterations can be considered as noise and rounded by a malicious user without altering the quality of the dataset at all. For instance, a 1mm alteration is meaningless in a dataset of accuracy 1m.

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 `testCollisions`). Such a strategy is valid as soon as a few errors occurs. By choosing $d_{max} = 4$ meters, the alteration on each point of a polygon is at most 2 meters. Usually, even in urban areas, polygons are spaced by a distance superior to 2 meters. 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. —longonly—

3.7 Detection

Outline Given a suspect dataset (R', DB'), we first translate it into the original reference system R_0, obtaining (R_0, DB'). The detection algorithm is very

Algorithm 1: Watermarking algorithm

Input: secret key $\mathcal{K}$, watermarking ratio $1/\gamma$, h,quantization step interval $D = [d_{min}, d_{max}]$
Data: (R_0, DB): original dataset
Output: $(R_0, DB_{\mathcal{K}})$: watermarked dataset
foreach *building* P *in* DB **do**
  $O \leftarrow$ `centroid`(P);
  $id \leftarrow$ `hsb`(O, h); /** */
  [f]robust identifier id
  `seed`$(G, \mathcal{K} \cdot id)$; /** */
  [f]seed the PRNG G with $\mathcal{K} \cdot id$
  if *nextInteger(G)* $\mod \gamma = 0$ **then**
    //Watermark this building
    $\boldsymbol{u} \leftarrow$ `orientation`(P); /** */
    [f]orientation
    $x_{max} \leftarrow \max\{p \in P | \mathbf{Op} \cdot \boldsymbol{u}\}$; /** */
    [f]main length
    $d \leftarrow d_{min} +$ `nextFloat(`G`).`$(d_{max} - d_{min})$; /** */
    [f]quantization step
    $b \leftarrow$ `nextInteger(`G`)` $\mod 2$; /** */
    [f]watermark bit b
    $x'_{max} \leftarrow$ `quantize` (x_{max}, d, b); /** */
    [f]quantize x_{max}
    $\sigma \leftarrow x'_{max}/x_{max}$; /** */
    [f]expansion factor
    `expand`$(P, O, \boldsymbol{u}, \sigma)$;
    if *testCollision ()* **then**
      `rollback` () ;

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 of the polygon we seed a random generator with $\mathcal{K}$ concatenated with its identifier. If the polygon satisfies the watermarking condition (i.e. `nextInteger`(G) $\mod \gamma = 0$), we compute the expected bit value b as `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 enables to optimize the trade-off between average distortion (here, $d/2$) and the minimum alteration leading to a decoding error (here, $d/4$).

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

It is easy to see that on a third party dataset, 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 value of α. From [10], setting $\alpha = -\log(\delta/2)/2t$ achieves a false positive occurrence probability $f_p \leq \delta$. We use this formula in our experiments to keep false positives occurrence probability under $\delta_0 = 10^{-4}$. —longonly— END

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**®[7], 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 4278 polygons (35565 vertices), representing dense build areas —longonly— as well as sparse ones —longonly—, and has an accuracy of 1 meter. END

END

4.2 Watermarking Impact on Quality

To evaluate the impact of our watermarking algorithm on the Pamiers dataset, 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 meter) to the maximum reasonable alteration (6 meters).

Mean Accuracy Alteration The impact on the mean accuracy is displayed on Fig. 3(a). Alteration increases when quantization steps increase and when γ decreases. We observe that the mean alteration is proportional to $(d_{min} + d_{max})/\gamma$. For instance, when $[d_{min}, d_{max}] = [3, 4]$, a good approximation of the mean accuracy

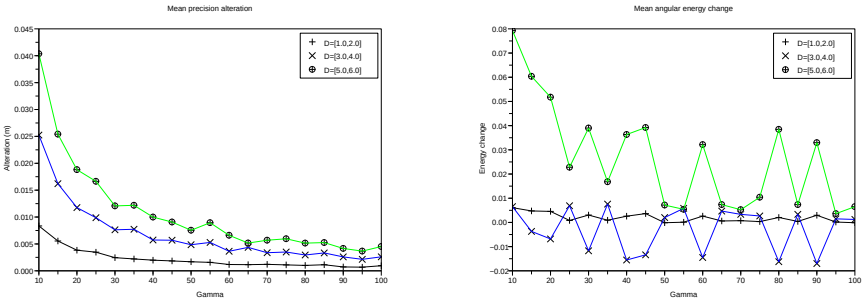
Algorithm 2: Detection algorithm

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
Data: (R', DB'), a suspect dataset
Output: MARK or NO_MARK
foreach *building* P *in* DB **do**
  $O \leftarrow \texttt{centroid}(P)$;
  $id \leftarrow \texttt{hsb}(O, h)$;
  $\texttt{seed}(G, \mathcal{K} \cdot id)$;
  if $\texttt{nextInteger}(G) \mod \gamma = 0$ **then**
    t++; /** */
    [f]increment total count
    $\boldsymbol{u} \leftarrow \texttt{orientation}(P)$;
    $x_{max} \leftarrow \max\{p \in P | \mathbf{Op} \cdot \boldsymbol{u}\}$;
    $d \leftarrow d_{min} + \texttt{nextFloat}(G).(d_{max} - d_{min})$;
    $b \leftarrow \texttt{nextInteger}(G) \mod 2$; /** */
    [f]expected bit b ;
    $x'_0 \leftarrow \texttt{quantize}(x_{max}, d, 0)$; /** */
    [f]closest 0-quantizer
    $x'_1 \leftarrow \texttt{quantize}(x_{max}, d, 1)$; /** */
    [f]closest 1-quantizer
    if $|x_{max} - x'_0| > |x_{max} - x'_1|$ **then**
      $b' \leftarrow 1$; /** */
      [f]found bit is $b' = 1$
    else
      $b' \leftarrow 0$; /** */
      [f]found bit is $b' = 0$
    if $b = b'$ **then** $m++$; /** */
    [f]increment match count
  $\alpha \leftarrow \texttt{threshold}(f_p, t)$;
  if $|m/t - 1/2| > \alpha$ **then return** MARK; **else return** NO_MARK;

alteration 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 $(d_{min} + d_{max})/2$. 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 mean alteration is fixed, one can easily compute γ.

END

Angular Quality We verify that the variation of angular quality introduced by our method is negligible on Fig. 3(b). Even for the highest quantization steps, $[d_{min}, d_{max}] = [5, 6]$, the highest angular energy variation is at most $+0.08$. As



(a) Mean accuracy alteration (b) Average angular energy alteration

Fig. 3. Impact of watermarking

a comparison, a weak gaussian noise (deviation $d = 0.2m$) increases the angular energy by +6.19.

END
END
END

4.3 Quality Impact

END

Filters We test the quality impact of the following filters and algorithms:

SQ Squaring filter. The strength of the squaring is controlled by the maximum allowed alteration d on coordinates.

DP Douglas-Peucker simplification algorithm. Artifacts under some distance threshold d are removed from the edges of the polygon. —longonly—

AKH Agrawal, Kiernan and Haas based algorithm (the original algorithm for relational databases is described in [2]) applied on the coordinates of the vertices. Its strength is controlled by the least power of two $lpow2$ considered as unalterable —longonly—. In this extension, we use our robust polygon identifiers and move only one point per polygon for a fair comparison.

GN Gaussian noise: 0-mean random noise added to each point of the database. The deviation of the noise is d.
—longonly—

WM Our watermarking algorithm.

As watermarking introduces a quality loss in the dataset, we use this quality loss to calibrate our benchmarks. 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.

Impact Table 1 gives a summary of the quality loss introduced by the filters presented above. In terms of accuracy alteration, our algorithm is comparable to a gaussian noise of deviation 20cm or an AKH watermarking algorithm with parameter $lpow2 = 0$ (point accuracy modifications are strictly less than $2^0 = 1m$). A gaussian noise with deviation 1m is far more destructive than a squaring filtering with $d = 1m$. In terms of angular quality, our algorithm achieves the smallest angular alteration. Notice also that squaring significantly improves angular quality (angular energy decreases). The values presented in Table 1 must be kept in mind while analyzing experimental results presented in the following.

Table 1. Quality Modification of Geographical Filters

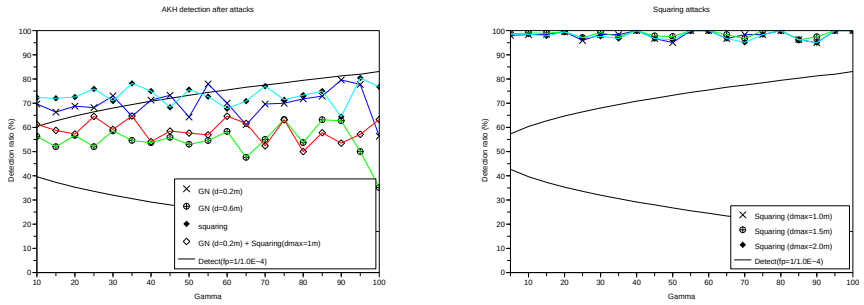
No.	Filter	Precision alt.	Angular energy
1	CA	0	0
2	GN (d=0.2m)	0.18	6.19
3	GN (d=0.6m)	0.53	40.00
4	SQ (d=1m)	0.19	- 14.11
5	DP (d=2m)	N/A	- 0.18
6	GN (d=1m)	0.89	78.52
7	DP (d=5m)	N/A	110.63
8	**WM**$_{(\mathbf{dmin=3,dmax=4})}$	**0.013**	**0.02**
9	AKH (lpow2=3)	0.14	26.73
10	AKH (lpow2=2)	0.07	13.01
11	AKH (lpow2=1)	0.04	4.93
12	AKH (lpow2=0)	0.02	1.58

—longonly—

4.4 Robustness of AKH-based Schemes

The straightforward extension of the classical AKH scheme [2] to our context does not lead to a robust watermarking algorithm. The point is that the alterations introduced by this algorithm can be seen as random noise. This kind of noise is removed by a squaring filter. We implemented and tested such an extension. Our implementation of the AKH algorithm consists of looping over the polygons and for each of them seeding a PRNG with the most significant bits of its centroid. Drawings from the PRNG are used to select which coordinate of which vertex of the polygon is going to be used for watermarking. The selected value is then modified according to the classical AKH method. On Fig. 4(a), we display the detection ratios obtained after the attacks of a watermarked dataset obtained by the AKH-based algorithm: squaring, gaussian noise (with deviation 0.2 and 0.6), and gaussian noise (deviation 0.2) followed by a squaring. We also display the detection bounds for a false positive occurrence probability under 10^{-4}. These bounds are depicted as full black lines. Above the upper one

and under the lower one, watermarks are detected. Detection ratios on a watermarked dataset are considerably lowered after squaring. If $\gamma > 50$, squaring even makes the watermark unreadable.



(a) AKH-based scheme (b) Our scheme

Fig. 4. Robustness against squaring

Another competitive approach is to apply the AKH method on the least significant bits of the centroid: a translation can be computed so that the correct bit of the centroid is changed into the intended watermark bit. This method is more simple than ours and respects angular quality, but does not resist to small random translations or rotations.

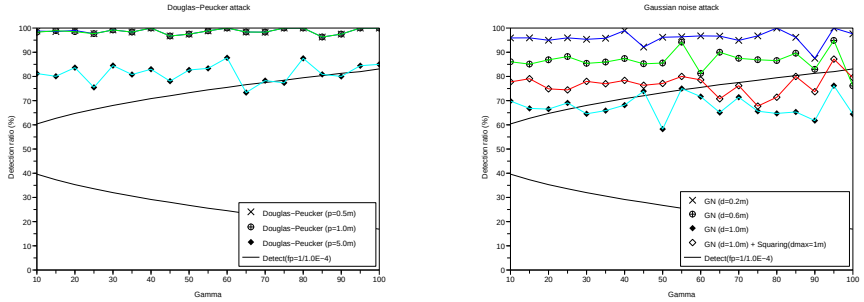
4.5 Robustness to Geographical Filters

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}).

Squaring We present in Fig. 4(b), the detection ratios observed 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 squaring, used here only for validation). 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.

Douglas & Peucker Simplification The Douglas-Peucker algorithm [5] is a polyline simplification algorithm. It is systematically used by geographic data users with a 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. We tested the robustness of our algorithm against Douglas-Peucker filtering for different

threshold values. —longonly— The results are displayed on Fig. 5(a). Note that our algorithm performs very well when the Douglas-Peucker filtering distance is 1 or 2 meters. The detection ratio nevers falls below the detection threshold. The reason is that the shapes of the polygons are regular enough so that they are invariant to these filterings. For a filtering distance of 5 meters, the mark is removed in some situations. But such a filtering is very aggressive: it can remove a $4m \times 4m$ room in a building.



(a) Douglas-Peucker Simplification (b) Gaussian Noise

—longonly—

Fig. 5. Robustness against attacks

4.6 Robustness to Attacks

END
END

Gaussian Noise, Smoothing, Coordinates Rounding All these attacks can be modeled by Gaussian Noise so we do not present experimental results for smoothing and coordinates rounding. Gaussian noises with deviations of $20cm$,$60cm$ and $1m$ were tested and the results displayed in Fig. 5(b). For reasonable noises ($d = 20cm$ or $d = 60cm$), marks are still detected. When $d = 1m$, the watermark is removed for a large interval of γ.

Interestingly, the application of a squaring algorithm on a noised dataset increases the detection ratio. It even permits the recovery of the watermark for higher values of γ.

END
END
END

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

Since watermark resides in the length of polygons, an attacker may try to randomly expand polygons of a watermarked dataset. This is quite an effective technique as soon as quality is not taken into account. To be sure that a great number of watermarked bits are erased, the attacker has to alter far more polygons.

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 angle tolerance ratio.

Note also that our method is invariant to polygons rotation since the expansion coefficient is defined relatively to polygon orientation and not to a particular reference system.

We also observe that squaring the data prior to the watermarking process, beside increasing its value, enhances slightly the detection ratio.

Due to space reasons, we are unable to present all experimental results. The interested readers may refer to [9] for presentation and/ discussion of other attackes. In particular, we show in [9] that our scheme is also robust against over-watermarking, cropping and missing reference system.

END

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 [2] 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 datasets in use in GPS viewers). Methods described in [16, 6] allow for the description of usability queries to be preserved by watermarking. But they either focus on basic numerical aggregates like `SUM` queries [6], which are not rich enough to represent angular constraints, or based on a trial and error method to handle generic black-box queries [16]. Using the latter method, it might not be possible to reach a valid set of alteration since no search strategy is defined.

Despite the fact that state-of-the art is very rich on watermarking still images which can be easily applied on image maps, only a few works were carried on on watermarking vector maps. In [13], a watermarking algorithm for 2D vector maps based on Delaunay triangulation and a decomposition in the mesh spectra

domain is introduced. The method is robust against a broad set of attacks but requires the original database for detection. The presented algorithm involves computationnaly expensive steps and its applicability in the context of Gigabytes databases is not discussed. In [15], a high watermarking capacity algorithm for vector maps is introduced. 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 document can be easily planned, as the method requires known synchronization points.

Another interesting approach is to add extra points to the shapes of the polygons. This approach, taken in [14, 12], was not really usable in our context where extra points in the envelope of a polygon can easily be detected and thus, removed, e.g. by a Douglas-Peucker fitering.

Quantization techniques were introduced by Chen and Wornell [4] as a way to optimize rate/robustness/distortion trade-offs in watermarking. Practical implementations of quantization, including dither modulation, achieve information theoretical optima of these tradeoffs.

6 Conclusion

In this paper we presented a blind watermarking algorithm for polygonal datasets. It is well suited to building layers of geographical datasets since watermarks are invariant through aggressive geographical filters applied by data users. We experimentally showed that it is difficult for an attacker to erase the watermark without paying an extra quality fee, compared to watermarking. Indeed, the possibility that an attacker destroys the watermark must be put into balance with the quality loss it introduces. The algorithm has been implemented into an open database watermarking framework [11]. We are currently working on designing algorithms for other layers of geographical datasets. A real challenge we are faced with is to deal with the interactions between the differents layers. 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 datasets remain consistent.

References

1. GéoPortail (visited 20/10/2006). `http://www.geoportail.fr`.
2. R. Agrawal, P. J. Haas, and J. Kiernan. Watermarking relational data: framework, algorithms and analysis. *VLDB J.*, 12(2):157–169, 2003.
3. P. Bourke. Calculating the area and centroid of a polygon, July 1988. `http://local.wasp.uwa.edu.au/~pbourke/geometry/polyarea/`.
4. B. Chen and G. W. Wornell. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, 47(4):1423–1443, 2001.

5. D. Douglas and T. Peucker. Algorithms for the reduction of the number of points required for represent a digitized line or its caricature. *Canadian Cartographer*, 10(2):112–122, 1973.
6. D. Gross-Amblard. Query-preserving watermarking of relational databases and XML documents. In *PODS '03: Proceedings of the Twenty-Second Symposium on Principles of Database Systems*, pages 191–201. ACM, 2003.
7. Institut Géographique National. BD TOPO - descriptif technique (in french), december 2002. `http://www.ign.fr/telechargement/MPro/produit/BD_TOPO/JT_Agglo/DT_BDTOPOPays_1_2.pdf`.
8. S. Katzenbeisser and F. A. Petitcolas. *Information hiding, techniques for steganography and digital watermarking*. Artech house, 2000.
9. J. Lafaye, J. Béguec, D. Gross-Amblard, and A. Ruas. Blind watermarking of geographical databases by polygon expansion. Technical report, Cnam-CEDRIC, Feb. 2007.
10. J. Lafaye and D. Gross-Amblard. Xml streams watermarking. In *20th Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSEC 2006)*, 2006.
11. J. Lafaye, D. Gross-Amblard, M. Guerrouani, and C. Constantin. Watermill: an optimized fingerprinting system for databases under constraints. *submitted to TKDE*.
12. C. M. Lopez Vazquez. Method of inserting hidden data into digital archives comprising polygons and detection methods, November 2003. US Patent no. 20030208679.
13. R. Ohbuchi, H. Ueda, and S. Endoh. Watermarking 2D vector maps in the mesh-spectral domain. In *Shape Modeling International*, pages 216–228. IEEE Computer Society, 2003.
14. K. T. Park, K. I. Kim, H. I. Kang, and S.-S. Han. Digital geographical map watermarking using polyline interpolation. In *PCM '02: Proceedings of the Third IEEE Pacific Rim Conference on Multimedia*, pages 58–65, London, UK, 2002. Springer-Verlag.
15. G. Schulz and M. Voigt. A high capacity watermarking system for digital maps. In *MM&Sec '04: Proceedings of the 2004 workshop on Multimedia and security*, pages 180–186, New York, NY, USA, 2004. ACM Press.
16. R. Sion, M. J. Atallah, and S. Prabhakar. Rights protection for relational data. *IEEE Trans. Knowl. Data Eng.*, 16(12):1509–1525, 2004.
17. The International Hydrographic Organization (IHO). *Specifications for Chart Content and Display Aspects of ECDIS*. IHO, 5th edition, december 2001.