Watermarking 2D Vector Maps in the Mesh-Spectral Domain

[1]Ryutarou Ohbuchi, [1]Hiroo Ueda, [2]Shuh Endoh

ohbuchi@acm.org, k7026@kki.yamanashi.ac.jp, endoshu@jp.ibm.com

[1]Computer Science Department, University of Yamanashi
4-3-11 Takeda, Kofu-shi, Yamanashi-ken, 400-8511, Japan.

[2] GIS Business Promotion, IBM Japan.

Abstract

This paper proposes a digital watermarking algorithm for 2D vector digital maps. The watermark is a robust, informed-detection watermark to be used to prevent such abuses as an intellectual property rights violation. The algorithm proposed in this paper embeds watermarks in the frequency-domain representation of a 2D vector digital map. Our method treats vertices in the map as a point set, and imposes connectivity among the points by using Delaunay triangulation. The method then computes the mesh-spectral coefficients [Karni00] from the mesh created. Modifications of the coefficients according to the message bits, and inverse transforming the coefficients back into the coordinate domain produces the watermarked map. Our evaluation experiments showed that the watermark produced by the method is resistant against additive random noise, similarity transformation, vertex insertion and removal. It is also resistant, to some extent, against cropping. Compared to our previous algorithm [Ohbuchi02], the algorithm described in this paper showed significantly improved attack resiliency.

1. Introduction

Applications of digital maps have been increasing rapidly. They are used, for example, in car navigation systems, location-based services for cellular phones with GPS (Global Positioning System) capability, web-based map services, and in geographical information systems (GIS) for city planning or disaster management. Digital maps are easy to update, duplicate, and distribute. As a digital data, digital maps are very easy to update, duplicate, and distribute. They are also prone to such abuses as illegal duplication and illegal distribution.

Geographical maps may be published by a government agency and shared (with fees) among map producing companies. Map companies add value to the base maps. For example, a car-navigation map should have up-to-date building positions, road and building outlines, building ownership, road signs, business and shop data (e.g., gas stations, hotels, and convenience stores), all of which are the result of the work which is often human-resource intensive.

Digital watermarking is a possible approach to counter abuses of digital media data, such as texts, audio data, images, movies, as well as 2D digital maps [16, 6, 12]. Digital watermarking adds a structure called watermark to the target data object (mostly) imperceptibly to the users and inseparably from the object. The information encoded in the watermark can be used, for example, to identify the copyright owner or to detect tampering.

Two-dimensional (2D) digital maps can be classified into either *raster*- or *vector*-digital maps (Figure 1). A raster digital map represents a map as raster image data, i.e., an image represented by a 2D array of pixels. A limitation of raster digital maps is quality degradation caused by rotation, scaling, and other geometrical transformations. Many web-based map services uses raster digital maps exactly for this reason; raster digital maps having limited resolution have a limited value for reuse or redistribution. As an image data, most of the watermarking algorithm developed for digital images [16, 6, 12] can be applied to the raster digital map. A vector digital map, on the other hand, employs geometrical primitives such as points, lines, polylines, and polygons to represent objects in the maps, such as building outlines, roads, rivers, reference points for strings, and contour

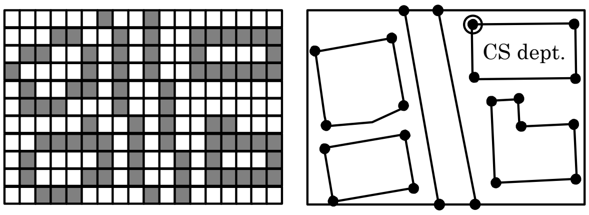


Figure 1. Raster (left) and vector (right) digital maps.

lines. Unlike the raster digital maps, the vector digital maps have an advantage of being able to be scaled and rotated without loss of quality. This advantage, on the other hand, makes a vector digital map a more valuable target for theft than an image digital map.

This paper presents a robust, informed-detection watermarking algorithm for 2D vector digital maps. The watermarking algorithm treats vertices in the map as a point set, and creates a 2D mesh out of the point set by using Delaunay triangulation. The algorithm then transforms the mesh shape into a "frequency" domain representation by using the mesh spectral analysis technique proposed by Karni et al [14, 15]. The algorithm modifies the most significant, that is, the low-frequency coefficients to encode watermark message bits. An inverse transformation back into the spatial domain creates a watermarked map.

Experiments showed that the watermark produced by the method is resistant against (1) additive random noise, (2) global Affine transformation, (3) vertex insertion and deletion, and (4) scrambling of object orders in the file. It is also resistant, to a certain extent, against (5) cropping. The method is mildly resistant to local deformations as well. Compared to our previous algorithm [22], the algorithm described in this paper showed significantly improved attack resiliency.

The rest of the paper is organized as follows. After reviewing previous work in the next section, we will present our watermarking algorithm Section 2. We then describe the results of evaluation experiments in Section 3, followed by a conclusion and future work in Section 4.

1.1. Previous Work

We know of only a few published works on watermarking vector digital maps [18, 17, 10]. Kurihara et al [18] encoded information into individual vertex coordinate, and their watermarks are quite fragile, among others, against additive random noise. Endoh, Masuda, Kanai, and Ohbuchi collaboratively developed nearly a dozen algorithms to watermark vector digital maps [10]. These watermarking methods are available as a part of the GIS map development toolkit. These watermarking methods targeted either vertex coordinate or vertex connectivity for watermarking. Kitamura et al reported, in detail, one of the algorithms developed by the collaboration [17]. In the Kitamura's method, a vector digital map is converted into a 2D array of scalar values, i.e., a "raster image". They subdivided the digital map uniformly into a rectangular grid, and treated each rectangle of the grid as a "pixel" of a raster image. As the pixel value of the images, they used the average of the areas of buildings defined by polygons that fall inside each rectangular "pixel". The image is then watermarked by using a method similar to a wavelet-based image-watermarking algorithm.

We have previously reported a watermarking algorithm for 2D vector digital maps [22]. The algorithm used a simple idea of translating a set of vertices in a uniformly subdivided rectangular region for embedding a message bit. The direction of translation of the set of vertices in a rectangle encoded a bit of the watermark message. The algorithm employed modified quad-tree [27] subdivision to create the rectangles adaptively to the density of vertices. A depth-first traversal of the quad tree created an ordering among the pixels. By averaging the displacement of the vertices upon extraction, and by repeatedly embedding the same message many times over a map, the watermarks produced by the method are resistant against additive random noise,

As a watermarking target, a three-dimensional (3D) polygonal mesh is somewhat similar to a 2D vector map; both are defined as a set of vertices (with either 2D or 3D coordinate values) and their connectivity. There are algorithms for watermarking 3D meshes [19, 20, 13, 1, 29, 24, 28, 32, 30, 5, 23]. These algorithms alter either vertex coordinate or vertex connectivity of the meshes for watermarking. Many of the recent 3D mesh watermarking methods employed transformed-domain approaches to watermarking [13, 24, 21, 30, 5, 23]. A method in this class would transform the mesh into a domain that reflects the notion of "frequency" and modify the most significant, low frequency components of the mesh shape to embed watermark messages. By modifying the low frequency component, the watermarks embedded by using these "frequency-domain" techniques are resistant against additive random noise, mesh smoothing (i.e., low-pass filtering), and other attacks. Our initial ideas question was if we could apply techniques developed for 3D polygonal meshes to watermarking 2D vector digital maps.

Of course, there are differences between 2D vector digital maps and 3D polygonal meshes, one of which is the relative ease of pose normalization. One of the major difficulties in watermarking 3D meshes is that of pose normalization, that is, normalization of the position, size, and orientation of the original and watermarked 3D meshes. Pose normalization of a 3D model is quite difficult if mesh simplification or remeshing has been applied in addition to geometrical transformation. In the case of the 2D vector maps, however, pose normalization is relatively easy. A reference map to align a watermarked map against can be found most of the time, so the scale and the orientation of the watermarked map can be normalized easily.

In the algorithm reported in this paper, we adopted the frequency domain mesh watermarking approach of [21, 23]. We also exploit a characteristic of 2D vector digital maps, namely, the ease of pose normalization, in developing our robust, informed-detection watermarking algorithm for 2D vector digital maps described in this paper.

2. The algorithm

Our watermarking algorithm embeds message bits into a 2D vector digital map by modifying a "frequency" domain representation of the map. Figure 2 shows an overview of the embedding and extraction steps.

To compute the "frequency" domain representation, the algorithm first establishes connectivity among vertices of the map by using *Delaunay* triangulation, creating a 2D mesh that covers every vertex in the map. The mesh is then transformed into a frequency domain representation by using mesh spectral analysis proposed by Karni and others [14, 15]. Modification of the frequency domain coefficients according to the message bits embeds a watermark. Inverse transforming the modified coefficient back into the coordinate domain produces a map with the watermark embedded. The modification of coefficients in the frequency domain ultimately displaces vertex coordinates in the spatial domain.

For computational efficiency and for robustness against cropping, a map is first divided into many rectangular sub-areas. We employed the *k-d* tree subdivision [9, 27] adaptively to the density of vertices in the map so that sub-areas have approximately equal numbers of vertices. Aforementioned mesh spectral analysis and watermark embedding is performed for each of the sub-areas. By embedding the same watermark repeatedly in multiple sub-areas, the watermark becomes resilient against cropping. The watermark is resilient against random noise and other attacks since the watermark is embedded into the low frequency component of the mesh version of the map.

The watermark is an *informed-* (or *non-blind-*) detection watermark. Watermarks are extracted by comparing the *reference map* (the map before watermarking, which may be escrowed) with the watermarked and possibly attacked *watermark map*.

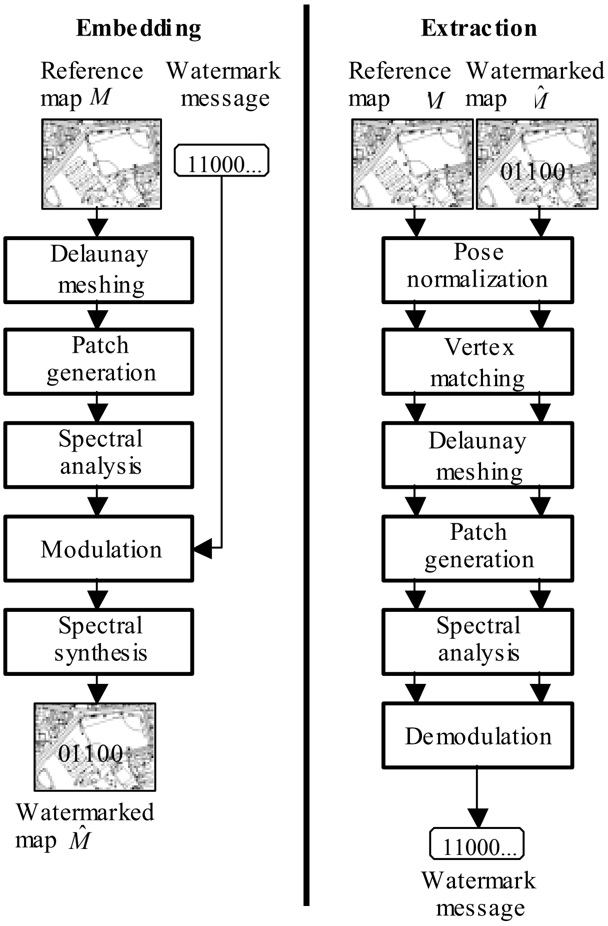


Figure 2. An overview of the watermark embedding and extraction steps.

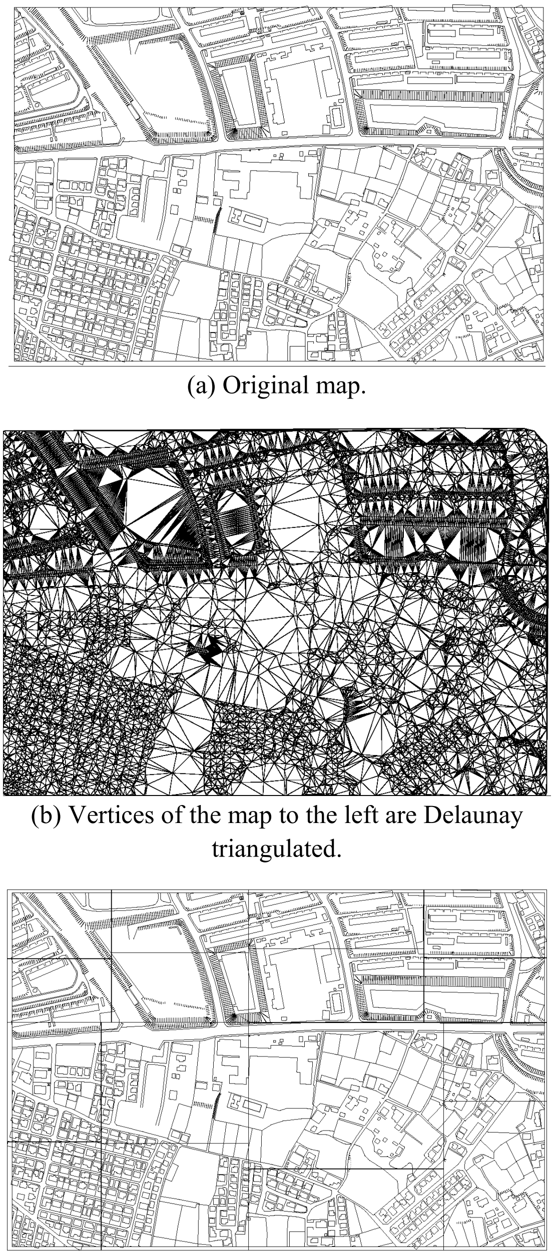
To extract the watermark, the two maps are first geometrically registered by using an iterative optimization process to minimize distance among a set of landmarks. This registration could remove an *Affine* transformation applied to the watermarked map. Then, the area subdivision equal to the one used for the embedding is recreated on the reference map, and the subdivision is transferred to the watermarked map. For each corresponding sub-area, mesh spectral analysis and then comparison of spectral coefficients recovers the embedded watermark.

2.1. Embedding

2.1.1. Meshing and area subdivision

A vector digital map is a collection of polygons and polylines that are not connected to each other. We first connect all the vertices into a single mesh by using Delaunay triangulation [9]. Figure 3a shows a vector digital map and its Delaunay mesh is shown in Figure 3b.

The algorithm then creates multiple rectangular submeshes, called watermarking patches. The purpose of subdivision into watermarking patches are twofold; (1) to increase resiliency against cropping attacks by repeatedly embedding the same watermark into multiple patches of a map, and (2) to reduce computational cost of eigenanalysis, the core of the mesh spectral analysis, by reducing the mesh size. The patches generated should contain roughly equal number of vertices, and that the number of vertices must exceed certain threshold to ensure the payload (i.e., the amount of watermark message bits embeddable.) We employed *k-d* tree [9, 27] to adaptively subdivide the mesh into patches of roughly equal vertex counts. An example of the patches generated by this technique is shown in Figure 3c, overlaid on the original map.


(a) Original map.

(b) Vertices of the map to the left are Delaunay triangulated.

(c) Watermark patches are generated adaptively to the local vertex counts.

Figure 3. Vertices are triangulated to create a mesh, which is then subdivided into watermark patches.

2.1.3. Spectral analysis

Mesh spectral analysis has been introduced by Karni and Gotsman to analyze shapes of 3D polygonal mesh models for compressing their geometry [14, 15]. Ohbuchi et al. [21, 23], and later, Cayre et al [5], applied the technique for watermarking 3D polygonal meshes in the "frequency" or mesh spectral domain. We borrow the technique for watermarking 2D vector digital maps by converting the maps into 2D meshes prior to watermarking.

There are several different mesh Laplacian matrices [4, 3, 8]. We employ Biggs' definition of mesh Laplacian $\mathbf{R}$ for the algorithm described in this paper.

$$\mathbf{R} = \mathbf{I} - \mathbf{HA} \qquad (1)$$

In the formula, $\mathbf{I}$ is the identity matrix and $\mathbf{H}$ is a diagonal matrix whose diagonal element $\mathbf{H}_{ii} = 1/d_i$ is the reciprocal of the degree (or valence) of the vertex *j*. $\mathbf{A}$ is the adjacency matrix whose elements are defined as below;

$$a_{ij} = \begin{cases} 1, & \text{if vertices } i \text{ and } j \text{ are adjacent;} \\ 0, & \textit{otherwise.} \end{cases} \qquad (2)$$

Figure 4a shows a simple mesh and Figure 4b shows its Laplacian matrix.

A polygonal mesh *M* having *n* vertices yields a Laplacian matrix $\mathbf{R}$ of size $n \times n$. Eigenanalysis of $\mathbf{R}$ produces *n* eigenvalues λ_i and *n* *n*-dimensional eigenvectors $\mathbf{w}_i$ $(1 \le i \le n)$. Projecting each component of the vertex coordinate $\mathbf{v}_i = (x_i, y_i)$ $(1 \le i \le n)$ separately onto the *i*-th normalized eigenvectors $\mathbf{e}_i$

$$\mathbf{e}_i = \mathbf{w}_i / \|\mathbf{w}_i\| \qquad (1 \le i \le n) \qquad (3)$$

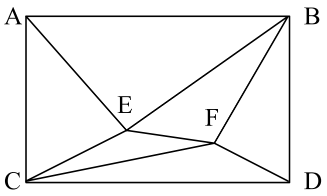
produces n mesh spectral coefficient vectors $\mathbf{r}_i = \left(r_{s,i}, r_{t,i}\right)$ $(1 \le i \le n)$. The subscripts s, and t denote orthogonal coordinate axes in the mesh-spectral domain corresponding to the spatial axes x and y.

We form the matrix $\mathbf{R}$ for a watermark patch using the connectivity within the patch. Edges connecting the patch with other patches are not included in our Laplacian matrix.

The inverse transformation, the mesh-spectral synthesis, is simply a linear combination of the basis $\mathbf{e}_i$ scaled by the mesh spectral coefficients $\mathbf{r}_i$.

$$\begin{aligned} \left(x_1, x_2, ..., x_n\right)^T &= r_{s,1}\mathbf{e}_1 + r_{s,2}\mathbf{e}_2 + \cdots + r_{s,n}\mathbf{e}_n, \\ \left(y_1, y_2, ..., y_n\right)^T &= r_{t,1}\mathbf{e}_1 + r_{t,2}\mathbf{e}_2 + \cdots + r_{t,n}\mathbf{e}_n. \end{aligned} \tag{4}$$

The spectral coefficients represent the notion of frequency (in the sense of the Fourier transformation) of the shape of the mesh, especially if (1) the lengths of edges are uniform over the mesh, and that (2) the degrees of vertices are uniform over the mesh [14, 15]. The mesh produced by the Delaunay triangulation has more uniform edge length than by the other triangulation methods given a set of points. However, as it is obvious from the example shown in Figure 3, the triangles in the mesh have a wide range of size and varying aspect ratio that may interfere with the Frequency decomposition of the mesh shape by using the mesh spectral analysis.



(a) A simple example mesh.

$$\begin{array}{c|cccccc|} & \text{A} & \text{B} & \text{C} & \text{D} & \text{E} & \text{F} \\ \text{A} & 1 & -1/3 & -1/3 & 0 & -1/3 & 0 \\ \text{B} & -1/4 & 1 & 0 & -1/4 & -1/4 & -1/4 \\ \text{C} & -1/4 & 0 & 1 & -1/4 & -1/4 & -1/4 \\ \text{D} & 0 & -1/3 & -1/3 & 1 & 0 & -1/3 \\ \text{E} & -1/4 & -1/4 & -1/4 & 0 & 1 & -1/4 \\ \text{F} & 0 & -1/4 & -1/4 & -1/4 & -1/4 & 1 \end{array}$$

(b) The Laplacian matrix for the mesh.

Figure 4. An example of the mesh Laplacian matrix.

2.1.4. Modulation

After the spectral coefficients are computed, a watermark is embedded into the map by modifying the spectral coefficients according to the message bits. The algorithm employs a simple modulation method similar to Hartung's [11]. The data to be embedded is an m-dimensional bit vector $\mathbf{a} = \left(a_1, a_2, ..., a_m\right)$ in which each bit takes values $\{0,1\}$. Each bit a_j is spread spatially over the map by duplicating each symbol by *chip rate* c, producing a watermark symbol vector $\mathbf{b} = (b_1, b_2, ...b_{mc})$, $b_i \in \{0,1\}$ of length $m \cdot c$. Repeatedly embedding the same bit c times increases resiliency of the watermark against additive random noise. If a watermark patch contains more vertices than the specified minimum d, the maximum value for the repetition is $c = floor(L/n)$ where n is the number of bits of the payload, the watermark message. For example, a mesh contains 480 vertices and the payload is 128 bit, chip rate c can be 1, 2, or 3.

Each element b_i of the symbol vector $\mathbf{b}$ is then repeated or *spread* c times;

$$b_i = a_j, \quad j \cdot c \le i < (j+1) \cdot c \tag{5}$$

After the spreading, the bit vector $\mathbf{b}_i$ is converted to an embedding symbol vector $\mathbf{b}' = (b'_1, b'_2, ...b'_{mc})$, $b'_i \in \{-1,1\}$ by the following mapping to create a zero-mean signal;

$$b'_i = \begin{cases} -1, & \text{if } b_i = 0; \\ 1, & \text{if } b_i = 1. \end{cases} \tag{6}$$

Assume that there are L usable rectangles and that ith rectangle contains M vertices. Let $v_{i,m}$ be the coordinate of mth vertex $(1 < m < M)$ in the ith rectangle prior to the watermarking, $p_i \in \{-1,1\}$ be the *pseudo-random number sequence* (PRNS) generated from a known key k_w, and α $(\alpha > 0)$ be the modulation amplitude. The coordinate $\hat{s}_{i.m}$ of the vertex after watermarking is computed by the following formula;

$$\hat{s}_i = s_i + b'_i \cdot p_i \cdot \alpha \tag{7}$$

The extraction requires the key k_w used for the embedding. Key distribution can be achieved by using a public-key cryptography scheme, for example.

The modulation amplitude α in the mesh spectral domain should be chosen so that the vertex displacement in the spatial domain won't affect visual qualities of maps. The geographical map standard by the *Geographical Survey Institute of Japan* states that the maximal error arrowed in a 1/2500-scale map is 0.3mm, which corresponds to 75 cm in the real world. Perturbation of vertices on the map by 2 or 3 integer coordinate points, that are, 20 or 30 cm in the real world, should be acceptable as long as the discontinuity artifacts introduced by the displacements are unnoticeable.

Since the spatial domain displacement is determined by the formula (4), both the modulation amplitude α as well as the number of coefficients modified, that is, the chip rate *c*, determines the vertex displacement in the spatial domain. The vertex displacement is roughly proportional to the product of $\alpha \cdot c$.

2.2. Extraction

2.2.1. Pose normalization

Prior to the extraction, an affine transformation applied to the watermarked map $\hat{M}$ is removed. This is a rather simple case of so-called *affine matching* problem (e.g., [6, 26]). In our algorithm, a set of corresponding pairs of landmarks are selected in the maps M and $\hat{M}$. Then, the sum of Euclidian distance between the landmark pairs is minimized. This minimization is performed by repetitively applying miniscule rotation, translation, and scaling transformations. Our algorithm uses the coordinate of the lower-left corner of a string that is associated with a building (or any other land object) for the matching. Correspondence between a pair of landmarks in the maps can be established easily by simply comparing the strings attached to them. We typically employ about 30 to 80 landmark pairs for the pose normalization.

In terms of watermarking, the most significant difference between 2D vector digital map and the 3D polygonal meshes lies in the relative difficulty of the pose normalization. For 2D vector digital maps, accurate pose normalization is possible even after affine transformation using (literally) landmarks. For 3D mesh models, even with reference mesh, such normalization can be quite difficult if the mesh had gone through geometrical transformation combined with mesh simplification and other connectivity changes.

2.2.2. Vertex matching

After the pose normalization, vertices that are either inserted or deleted due to attacks are found by comparing the coordinate values of the reference map M and the watermarked map $\hat{M}$.

To deal with the vertex-insertion attack, the algorithm chooses a vertex in $\hat{M}$ that is inside the circle of diameter *t* of the vertex v_r of M . If there is more than one such vertex, the vertex closest to v_r is used. To deal with the vertex-removal attack, if no vertex in $\hat{M}$ is found inside the circle of diameter *t* of the vertex v_r of M , a vertex v_w is inserted into $\hat{M}$. The inserted vertex v_w has the coordinate of v_r. The coordinate value of v_r is of course incorrect; it is simply treated as noise by the extraction algorithm.

The diameter *t* is a user-defined parameter. We chose the value of *t*=100 cm for the experiments described below. The value is chosen based on the maximal error of 75 cm (in the real-world scale) arrowed in a 1/2500-scale Japanese geographical map.

2.2.3. Meshing and patch generation

The reference map M is Delaunay triangulated, and the watermarking patches are created. The triangulation and the patches on M are exactly the same as the ones used for embedding. The triangulation and the patchfication are then transferred to the watermarked map $\hat{M}$. Using the vertex-to-vertex correspondence between the maps M and $\hat{M}$ established above.

2.2.4. Spectral Analysis

The spectral analysis is performed first on the reference mesh, which produces exactly the same set of eigenvectors and mesh spectral coefficients as the ones computed for the M during embedding. Note that the eigenvectors computed for the M can be used to derive mesh spectral coefficients for the $\hat{M}$, for they have the same connectivity. Expensive eigenvalue decomposition computation need to be performed only once per patch for the reference map M .

2.2.5. Demodulation

To extract a message bit, the algorithm compares a mesh spectral coefficient of M with the corresponding coefficients of $\hat{M}$. Let's assume that the *i*th coefficients of $\hat{M}$ and $\hat{M}$ are s_i and $\hat{s}_i$, respectively, and that p_i is the same PRNS as is used for embedding, generated from the same stego-key k_w. Then, the sum of the products q_j can be computed as follows;

$$q_j = \sum_{i=j\cdot c}^{(j+1)\cdot c-1} (\hat{s}_i - s_i)\cdot p_i = \sum_{i=j\cdot c}^{(j+1)\cdot c-1} b'_i \cdot \alpha \cdot p_i^2 \qquad (8)$$

If p_i is the same for embedding and extraction, and if disturbances of the vertex coordinates of $\hat{M}$ (e.g., additive random noise) are negligible,

$$q_j = c \cdot \alpha \cdot b'_i \qquad (9)$$

where q_j takes one of the two values $\{-\alpha c, \alpha c\}$. Since α and c are always positive, simply testing for the signs of q_j recovers the original message bit sequence a_j,

$$a_j = sign(q_j) \tag{10}$$

The string a_j can easily be converted to the original message bit sequence b_i by applying an inverse of the mapping as the embedding.

3. Experiments and results

In all the experiments described below, we used the following parameters.

- Minimum patch size d: $d = 480$ is used. The perceptibility experiment used $d = 128$ as well.
- Payload: A message of size 128 bit is embedded.
- Modulation amplitude α: $\alpha = 1.0$ and $\alpha = 1.5$.
- Chip rate c: c=1 for the cases where $d \geq 128$, and c=2 and c=3 for the cases where $d \geq 480$.

For the experiments, we used the 6 maps listed in Figure 5. As we target maps that mainly represent houses and buildings, we choose the maps from the urban and suburban residential and commercial areas.

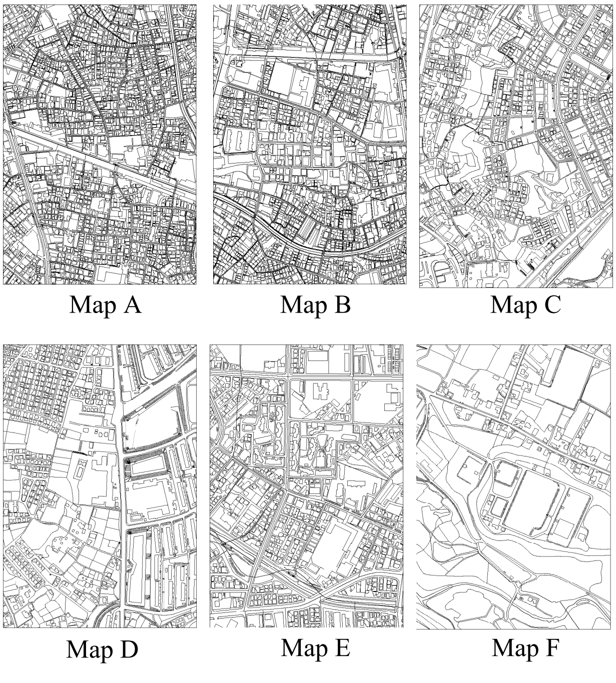

Map A
Map B
Map C
Map D
Map E
Map F

Figure 5. Six maps used for the evaluation experiment.

3.1. Perceptibility

Perceptibility of the watermark depends on various parameters, including the payload *m*, patch size *d*, the chip rate *c*, and the modulation amplitude *α*. An increase in *c* or *α* improves the attack resiliency but it decreases the visual quality of the map. The effect of the patch size *d* is

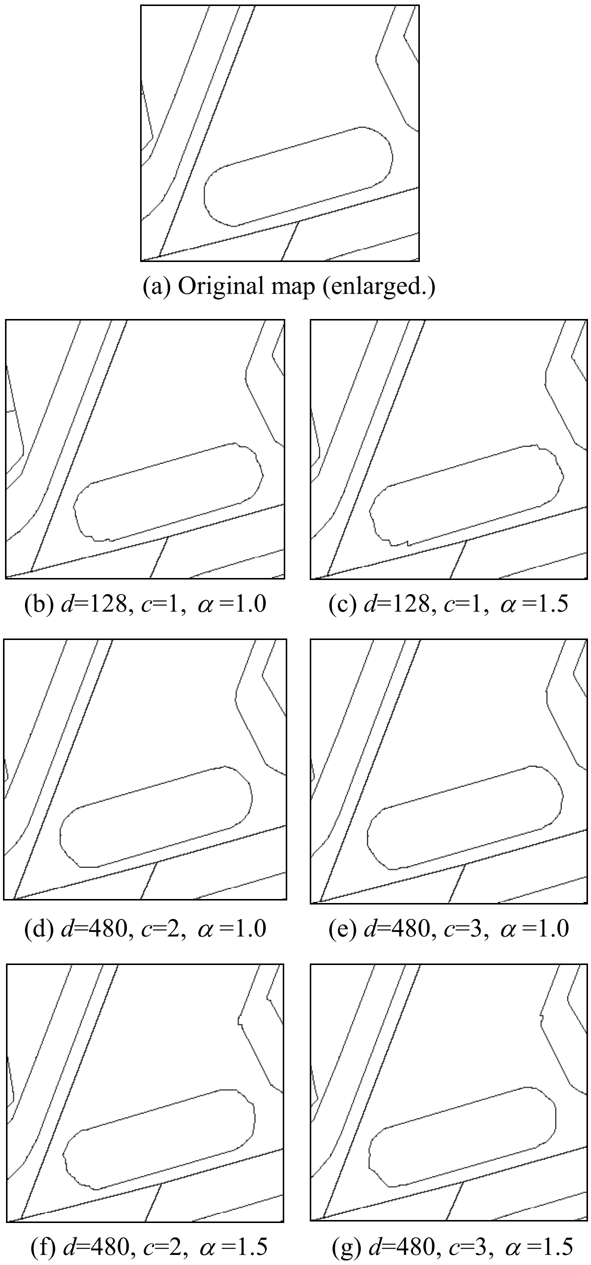

(a) Original map (enlarged.)
(b) d=128, c=1, α=1.0
(c) d=128, c=1, α=1.5
(d) d=480, c=2, α=1.0
(e) d=480, c=3, α=1.0
(f) d=480, c=2, α=1.5
(g) d=480, c=3, α=1.5

Figure 6. Perceptibility of the watermark using various watermarking parameters.

subtler. For example, a larger patch would have a higher resiliency against additive random noise, but the resulting decrease in the number of patches per map reduces the resiliency against cropping attacks.

Figure 6a-6g shows the effect on visual quality of the map of the watermarking, using 6 combinations of watermarking parameters. The figure shows the map area of the size approximately $30\text{m} \times 30\text{m}$ in the real world. A higher chip rate c and the higher amplitude α (of the coefficients modulation) clearly degraded the visual quality of the map. In this example, parameters d=480, c=2, α=1.0 showed the least amount of distortion. It also showed acceptable performance in terms of attack resiliency in the experiments that follows.

3.2. Resiliency against attacks

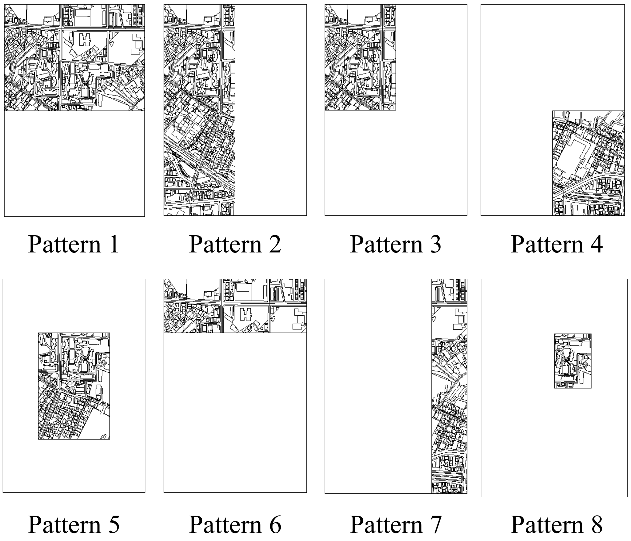
We experimentally evaluated the attack resiliency of the watermark using the following attacks;

1. Translation: Translate all the vertices in the map by 1000 units and 500 units, respectively, toward positive x and y directions.
2. Upscaling: Uniformly enlarge the map by the factor 5.5.
3. Downscaling: Uniformly shrinks the map by 0.3 (attack 3a) and 0.6 (attack 3b) times the original. Coordinate values are rounded to the nearest integers.
4. Rotate: Rotate the map by 45 degree about the upper-left corner (0,0) (attack 4a) or the center (3750, 2500) (attach 4b) of the map.
5. Similarity transformation: The map is rotated by 45 degree about the center of the map (3750, 2500), translated, and then downscaled by the factor of 0.6. Coordinate values are rounded to the nearest integers after the downscaling.
6. Local deformation: The map is subdivided uniformly into rectangular grid of size 10×10, and the vertices inside each rectangle are rotated by 1 degree about the center of the rectangle. Given the vertex coordinate of (x, y), the direction of rotation for the coordinate is clockwise if $(x+y) \bmod 2 = 0$, counterclockwise if otherwise.
7. Object order scrambling: The order of appearance of objects (e.g., polygons of building outlines) in a data file is scrambled.
8. Vertex insertion: Vertices are added to target objects *i.e.*, polygons and polylines, while trying to preserve the appearance of the map. In the experiment, 3000 vertices are added to a map.
9. Additive random noise: Random noise having the amplitudes of either α =10 cm, 30cm, or 50cm (attacks 9a, 9b, and 9c, respectively) is added to the vertex coordinate.
10. Cropping: Each map is cropped according to the 8 cropping patterns shown in Figure 7. Areas of the cropped maps varied from 1/2 to 1/16 of the original.

Table 2 shows the results of the experiments. In Table 2, all the bit error rates are the average over the six maps used for the experiment. Table 2 also includes the results obtained using our previous watermarking method [22] for comparison. Some of the boxes are left vacant, as these figures are not available for our previous method.

Overall, the new frequency domain watermarking method described in this paper significantly outperformed our previous algorithm in terms of attack resiliency. Comparing between the chip rates of 2 and 3, the results produced by the higher chip rate of c=3 produced somewhat less error than the lower chip rate of c=2.

The new watermark is robust against translation, upscaling, or rotation in which case no error occurred. The new watermark is immune to vertex insertion and scrambling of object order in the map data file as expected.


Pattern 1 Pattern 2 Pattern 3 Pattern 4

Pattern 5 Pattern 6 Pattern 7 Pattern 8

Figure 7. Eight cropping patterns used for the experiments.

Downscaling by the factor of 0.3 caused errors due to the round off error of the coordinate values. A combination of rotation, translation, and downscaling also caused errors, most likely due to the round off error due to downscaling.

The watermark showed significant error after the local deformation. Unlike a global geometrical transformation (e.g., similarity transformation), local random deformation can't be compensated for by our pose normalization method. The watermarks also showed significant errors after the cropping that reduced the area of the map down to 1/2~1/16 of the original. In both attacks, the error rates of the new algorithm are again much lower than those using our previous method.

4. Conclusion and future work

In this paper, we presented a frequency-domain watermarking algorithm for vector digital maps. The algorithm embeds bits into a map by modifying "frequency" domain representation of the map. The mesh spectral coefficients are computed by first converting the map into a mesh by using *Delaunay* triangulation, and then applying mesh spectral analysis proposed by Karni et al [14, 15].

Experiments showed that the watermarks produced by our new method described in this paper are more resilient than our previous algorithm [22]. The watermark produced by new algorithm is resilient, to some extent, against such attacks as additive random noise, addition of vertices, rotation, scaling, and cropping of the map.

Our future work includes improvements in attack resiliency. For example, resiliency against additive random noise or other frequency dependent attacks can be improved by computing a better frequency domain representation of the shape.

We also need to find a quantitative measure of distortion that reflects human perception, as well as standard sets of attacks and maps that can be used for objectively evaluating and comparing the proposed map watermarking methods.

Table 2. Bit error rates due to various attacks. The payload was 128 bit. The "N/M" and "N/A" in the boxes indicate, respectively, "not measured" and "not available".

		New method		Previous method
Minimum vertex counts per patch *d*		480	480	
Modulation amplitude α		1.0	1.0	
Chip rate *c*		2	3	
(1) Translation		0.0 %	0.0 %	0.0 %
(2) Upscaling (× 5.5)		0.0 %	0.0 %	0.0 %
(3) Downscaling	(3a) × 0.6	0.0 %	0.0 %	0.7 %
	(3b) × 0.3	0.1 %	0.0 %	---
(4) Rotation	(4a) Center at (0, 0), 45 degree	0.0 %	0.0 %	0.0 %
	(4b) Center at (3750, 2500), 45 degree	0.0 %	0.0 %	0.0 %
(5) Similarity transformation ((4b)+(1)+(3a))		0.1 %	0.0 %	1.0 %
(6) Local deformation		9.2 %	8.1 %	45.2 %
(7) Object order scrambling		0.0 %	0.0 %	0.0 %
(8) Vertex insertion (3000 vertices)		0.0 %	0.0 %	0.0 %
(9) Additive random noise	(9a) α=10cm	0.0 %	0.0 %	0.0 %
	(9b) α=30cm	0.1 %	0.0 %	---
	(9c) α=50cm	5.9 %	3.4 %	8.5 %
(10) Cropping	(10a) Pattern 1	0.0 %	0.0 %	1.4 %
	(10b) Pattern 2	0.1 %	0.4 %	1.8 %
	(10c) Pattern 3	0.7 %	1.3 %	7.0 %
	(10d) Pattern 4	0.4 %	0.0 %	6.8 %
	(10e) Pattern 5	0.8 %	0.0 %	16.7 %
	(10f) Pattern 6	0.4 %	0.4 %	15.0 %
	(10g) Pattern 7	1.4 %	0.4 %	11.3 %
	(10h) Pattern 8	18.8 %	15.1 %	35.9 %

References

[1] O. Benedens, Geometry-Based Watermarking of 3D Models, *IEEE CG&A*, pp. 46-55, January/February 1999.

[2] O. Benedens, C. Busch, Towards Blind Detection of Robust Watermarks in Polygonal Models, Proc. *EUROGRAPHICS 2000* (*Computer Graphics Forum*, Volume 19(2000), No. 3)

[3] N. Biggs, *Algebraic Graph Theory* (2nd Ed.). Cambridge University Press, 1993.

[4] B. Bollobás, *Modern Graph Theory*, Springer, 1998.

[5] F. Cayre, P. Rondao Alface & H. Maître, *Compression and watermarking of 3D triangle meshes*, SPIE 47th Annual Meeting, Jul. 2002, Seattle, WA (2002)

[6] G. S. Cox, G. DeJager, A survey of point pattern matching techniques and a new approach to point pattern recognition, Proc. *South African Symposium on Communications and Signal Processing 1993*, pp. 243-248.

[7] Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, *DIGITAL WATERMARKING*, Morgan Kaufmann Publishers, 2001.

[8] F. R. K. Chung, *Spectral Graph Theory*, Number. 92 in Regional Conference Series in Mathematics, American Mathematical Society, 1997.

[9] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, 2nd edition, Springer, 2000.

[10] Shuh Endoh, Hiroshi Masuda, Ryutarou Ohbuchi, Satoshi Kanai, Development of Digital Watermarking Technology for Vector Digital Maps, *IPA Technology Expo* 2001 Reports, 2001 (Japanese).

[11] F. Hartung, P. Eisert, and B. Girod, Digital Watermarking of MPEG-4 Facial Animation Parameters, *Computers and Graphics*, Vol. 22, No. 4, pp. 425-435, Elsevier, 1998.

[12] Neil F. Johnson, Zoran Duric, and Sushil Jajodia, *Information Hiding: Steganography and Watermarking–Attacks and Countermeasures*, Kluwer Academic Publishers, 2000.

[13] S. Kanai, H. Date, and T. Kishinami, Digital Watermarking for 3D Polygons using Multiresolution Wavelet Decomposition, Proc. *Sixth IFIP WG 5.2 GEO-6,* pp. 296-307, Tokyo, Japan, December 1998. (http://minf.coin.eng.hokudai.ac.jp/members/kanai/wm1-geo6.pdf)

[14] Zachi Karni, Craig Gotsman, Spectral Compression of Mesh Geometry, Proc. *SIGGRAPH 2000*, pp. 279-286, 2000.

[15] Zachi Karni, Craig Gotsman, 3D Mesh Compression Using Fixed Spectral Bases, Proc. *Graphics Interface 2001*, pp. 1-8, 2001.

[16] S. Katzenbeisser, F. A. P. Petitcolas, *Digital Watermarking*, Artech House, London, 2000.

[17] I. Kitamura, S. Kanai, and T. Kishinami, Watermarking Vector Digital Map using Wavelet Transformation, Proc. *Annual Conference of the Geographical Information Systems Association (GISA) 2000*, Vol. 9, pp.417-421, 2000 (Japanese).

[18] M. Kurihara, N. Komatsu, H. Arita, Watermarking Vector Digital Maps, *Special Interest Group Report* Vol. 2000, No. 36, Information Processing Society of Japan (IPSJ), (Computer Security No. 009-1, 2000) (Japanese).

[19] R. Ohbuchi, H. Masuda, and M. Aono, Watermarking Three-Dimensional Polygonal Models, Proc. *ACM Multimedia '97*, pp. 261-272, Seattle, Washington, USA, November 1997.

[20] R. Ohbuchi, H. Masuda, and M. Aono, Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications, pp. 551-560, *IEEE JSAC*, May 1998.

[21] Ryutarou Ohbuchi, Shigeo Takahashi, Takahiko Miyazawa, and Akio Mukaiyama, Watermarking 3D Polygonal Meshes in the Mesh Spectral Domain, in Proc. *Graphics Interface* 2001, pp. 9-17, Ontario, Canada, June, 2001.

[22] Ryutarou Ohbuchi, Hiro Ueda, Shu Endoh, Robust Watermarking of Vector Digital Maps, in Proc. *IEEE Conference on Multimedia and Expo 2002 (ICME 2002)*, Lausanne, Switzerland, August 26-29, 2002.

[23] Ryutarou Ohbuchi, Akio Mukaiyama, Shigeo Takahashi, A Frequency-Domain Approach to Watermarking 3D Shapes, *Computer Graphics Forum* **21**(3), pp. 373-382, 2002 (Proc. *EUROGRAPHICS 2002*)

[24] Emil Praun, Hugues Hoppe, Adam Finkelstein, Robust Mesh Watermarking, Proc. *SIGGRAPH '99*, pp. 49-56, Aug. 1999.

[25] W. H. Press et al., *Numerical Recipes in C-The Art of Scientific Programming*, 2nd Ed., Cambridge University Press, Cambridge, UK, 1992.

[26] I. Rothe, H. Suesse, K. Voss, The method of normalization to determine invariants, *IEEE Trans. on PAMI*, Vol. 18, No. 4 (1996), pp. 366-377.

[27] H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, Reading, MA, 1990.

[28] M. G. Wagner, Robust Watermarking of Polygonal Meshes, Proc. *Geometric Modeling & Processing 2000*, pp. 201-208, Hong Kong, April 10-12, 2000.

[29] B-L. Yeo and M. M. Yeung, Watermarking 3D Objects for Verification, *IEEE CG&A*, pp. 36-45, January/February 1999.

[30] Kangkang Yin, Zhigeng Pan, Jiaoying Shi, David Zhang, Robust mesh watermarking based on multiresolution processing, *Computers & Graphics*, Vol. 25 (2001), pp. 409-420.

Acknowledgements

We thank Zenrin Co. Ltd. for lending us the map data for the experiments. We'd like to thank Prof. Hiroshi Masuda of the University of Tokyo and Prof. Satoshi Kanai of Hokkaido University for valuable discussions and comments. We also thank Prof. Shigeo Takahashi for useful comments. A part of this research is funded by the grant No. 12680432 from the *Ministry of Education, Culture, Sports, Sciences, and Technology of Japan*, as well as the *Okawa Foundation of Information and Telecommunications*.