PAPER

Vector Watermarking Method for Digital Map Protection Using Arc Length Distribution*

Suk-Hwan LEE[†a)], ***Member***, **Xiao-Jiao HUO**[††], ***and*** **Ki-Ryong KWON**[††b)], ***Nonmembers***

SUMMARY With the increasing demand for geographic information and position information, the geographic information system (GIS) has come to be widely used in city planning, utilities management, natural resource environments, land surveying, etc. While most GIS maps use vector data to represent geographic information more easily and in greater detail, a GIS vector map can be easily copied, edited, and illegally distributed, like most digital data. This paper presents an invisible, blind, secure, and robust watermarking method that provides copyright protection of GIS vector digital maps by means of arc length distribution. In our method, we calculate the arc lengths of all the polylines/polygons in a map and cluster these arc lengths into a number of groups. We then embed a watermark bit by changing the arc length distribution of a suitable group. For greater security and robustness, we use a pseudo-random number sequence for processing the watermark and embed the watermark multiple times in all maps. Experimental results verify that our method has good invisibility, security, and robustness against various geometric attacks and that the original map is not needed in the watermark extraction process.

key words: *vector watermarking, GIS security, digital map protection, Arc length*

1. Introduction

A geographic information system (GIS) is a computer-based information system that is used to digitally represent and analyze geographic features on the Earth's surface, as well as events (non-spatial attributes linked to the geography being studied) that take place on it. The feature or event that is to be represented digitally is converted from an analog (smooth line) into a digital form. GIS digital maps employ geometrical primitives such as points, lines, polylines, and polygons to represent objects such as building outlines, roads, rivers, and contour lines. However, since a GIS digital map can be easily updated, duplicated, and distributed by any user, there must be a way to protect it. Digital watermarking, a core technology in this regard, provides an effective means of countering potential abuse. Since a raster map represents a map as raster image data, most of the watermarking algorithms developed for digital images can be applied to raster maps. However, since they cannot be applied to vector maps, many researchers have presented various methods for watermarking vector maps.

Ohbuchi et al. [1] proposed the Uniform (UNIF), Quadtree (QUAD), and Modified quadtree (MQUAD) methods, which are the earliest watermarking techniques intended for vector digital maps. In these methods, however, the original map is needed for extraction of the watermark, and the result has weak robustness against attacks such as the addition of noise, swapping, etc. In another approach, Ohbuchi et al. presented a method for the watermarking of 2D vector maps in the mesh-spectral domain [2], in which a 2D mesh is generated by establishing connectivity among the vertices of a map using Delaunay triangulation. Then, the mesh is transformed into the frequency domain using mesh-spectral analysis. However, this method is sensitive to the mesh triangulation process and fragile to object rearrangement and layer attacks, in which a layer is cut and edited. Vogit et al. presented a high-capacity watermarking scheme for digital maps [3] that embeds the watermark by changing the x, y coordinates of the data within the tolerance of those data [4] and uses a block code for reconstructing missing data in the detection process. They also presented a feature-based watermarking method [5] that chooses two disjunctive sets of grid elements, divides them into two sub-patches, and then changes the distances of points to a reference line within each sub-patch, according to the watermark. But this method has the non-synchronization of the watermark arises when a number of vertices are moved to other positions. In addition to the aforementioned approaches, a number of researchers have presented methods of vector map watermarking [6], [7] that have the weakness of being vulnerable to geometric attacks.

Recently, Zhang et al. [8] divided a vector map into regular blocks and changed the vertex coordinates considering the topology of the vertex. However, this method can extract the watermark only if the topology of the vertex is preserved. Shujun et al. [9] presented a double-grid-based watermarking method that hides the watermark information separately in the least significant bits of the node coordinates. Their method, however, has a number of problems related to its robustness to data reordering, layer editing, and noise. Dakroury et al. [10] introduced a mechanism to protect GIS data, which they termed the distributed transparent extensive data protection mechanism (DTEDPM). This

Manuscript received February 18, 2013.
Manuscript revised June 11, 2013.

†The author is with the Tongmyong University, 535, Yongdang, Namgu, Busan, Korea.
††The authors are with Pukyong National University, 599–1, Daeyeon-dong, Namgu, Busan, Korea.

*This work was supported under IT/SW Creative research program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2013-H0502-13-1023) and 5th. BB (Brain Busan) 21 project and also the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the CPRC (Communications Policy Research Center) support program supervised by the KCA (Korea Communications Agency) (KCA-2013-003).

a) E-mail: skylee@tu.ac.kr
b) E-mail: krkwon@pknu.ac.kr (Corresponding author)

DOI: 10.1587/transinf.E97.D.34

worked by combining AES and RSA cryptography algorithms with watermarking techniques. The focus of their work, however, was more on the encryption and decryption time obtained using shapefiles than it was on watermarking performance. H. Chang [11], [12] proposed a watermarking method using the areas of a polygon. However, this method processes only the polygon object, and the vertices must be changed repeatedly to satisfy the embedded condition. J. Kim [13] proposed a watermarking method for DXF format maps using polyline length. Beyond these methods, many researchers have presented watermarking methods that employ the geometric properties of vectors [14], [15]. Because GIS vector maps mainly consist of polylines and polygons, a vector watermarking method can consider the geometric properties of these two objects as the embedded feature to achieve robustness.

We previously presented a robust watermarking method based on polylines and polygons [16]. This method clusters all polylines and polygons in the feature layers and embeds the watermark into the polyline distance and the polygon area in each cluster. The polyline distance is defined as the sum of the distance between the vertices and an external reference point. Thus, a watermark is embedded in the distribution of the polyline distance and polygon area in each cluster by moving all the vertices in the direction of a reference point. However, the external reference point's movement means that the shape of the polyline and polygon cannot be preserved. Consequently, this method needs the specified vertex movement tolerance. Furthermore, this method does not ensure watermark security.

In this paper, we propose a secure arc-length-based watermarking method that is robust to geometric transformation and preserves object shape. Our method is based on the polyline/polygon-type data of the ESRI shapefile, which is a popular geospatial vector data format for GIS. In our method, we calculate the arc length of all polylines/polygons in a map and cluster them into a number of groups based on the Gaussian Mixture Model (GMM) using the Expectation-Maximization (EM) algorithm. Next, we input the watermarking message and use a pseudo-random number sequence (PRNS) for processing the watermark bit in order to achieve improved security. We then embed a processed watermark bit by changing the local mean arc length of the polylines/polygons of a suitable group. To obtain greater robustness, we also embed the watermark bits multiple times in all maps. In the process of embedding the watermark, we produce a number of keys that are needed for extracting the watermark, such as a PRNS. Finally, we use the saved keys to extract the watermark bits in the subgroups, and use a checkout method to obtain the optimized watermark message. In our method, the original map is not needed in the watermark extraction process, while a condition of invisibility is satisfied. From our experimental results, we confirm that the artifacts produced by the embedded watermark cannot be detected by users, and that the watermark is not damaged or detected by other designers. In addition, the watermark exhibits robustness against geometric attacks such as translation, rotation, random noise, insertion and deletion of vertices, scrambling of the order of the geometric primitives in a data file, and cropping.

The remainder of this paper is structured as follows. In the next section, we introduce the GIS data system and its structure. In Sect. 3, we describe the existing watermarking algorithm, which is based on vector data. Further, we present our watermarking algorithm in Sect. 4, and the results of our experiments in Sect. 5. Finally, in Sect. 6, we offer a summary and conclusion.

2. GIS Data Structure

GIS data consists of spatial data and attribute components. In a spatial component, the observations have two aspects in terms of their localization: absolute localization based on a coordinate system, and a topological relationship, that refers to other observations. This section briefly explains the properties of vector data and the Shapefile format.

2.1 Vector Data

Vector data is defined by the vectorial representation of its geographic data; it is represented in the form of coordinates. In vector data, the basic units of spatial information are points, lines (arcs), and polygons. Each of these units is simply composed of a series of one or more coordinate points; for example, a line is a collection of related points, and a polygon is a collection of related lines. Vector data provides significantly greater analysis capability in applications such as roads, power lines, railways, and telecommunications (for example, it can be used determine the best route, largest port, airfields connected to two-lane highways, and so on).

GIS data consists of several layers that correspond to the data characteristics of a given railroad, river, street, building, topography, etc., as shown in Fig. 1. The expression of each layer comprises four types of data, i.e., points, lines (polylines), surfaces (polygons), and characters, apart from arcs, circles, or ellipses.

· *Point*: Expresses altitude values and symbols of features such as telegraph poles, manholes, rice fields, and farms.

· *Line (Polyline)*: Expresses data for features such as roads, road centerlines, and railroads. According to a re-

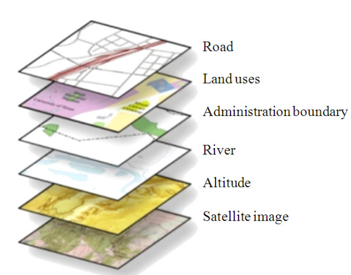


Fig. 1 An example of GIS vector map organized by 6 layers (http://gis.seoul.go.kr).

duced scale, line data can be expressed as a surface.

· *Surface (Polygon)*: Expresses data for features such as borders, buildings, and roads.

· *Character*: Expresses the names of buildings and roads, and the numerical values of contour lines.

Since the vector data of polylines and polygons are considered to be a very important component of the geographical features, the watermarking for a GIS vector map should use these objects in the embedding target.

2.2 Shapefile

The shapefile is a popular geospatial vector data format for geographic information systems software [17], [18]. It is developed and regulated by the ESRI as an open specification for data interoperability among ESRI and other software products. A shapefile stores non- topological geometry and attribute information of the spatial features in a dataset. The geometry of a feature is stored as a shape comprising a set of vector coordinates. Shapefiles can support point, line, and area features. Area features are represented as closed loop, double-digitized polygons. Attributes are held in a dBASE format file. Each attribute record has a one-to-one relationship with its associated shape record.

Because shapefiles do not entail the processing overhead of a topological data structure, they offer advantages over other data sources, such as a faster drawing speed and edit ability. Shapefiles can also handle single features that overlap or are noncontiguous. They also typically require less disk space and are easier to read and write. In addition, there is much software that supports shapefile formats, such as ARC/INFO, PC ARC/INFO, Spatial Database Engine (SDE), ArcViewGIS, and BusinessMap. Further, we can create or modify a shapefile directly by creating a program according to the shapefile specifications. Considering these reasons, we chose the shapefile format as the vector map for watermarking.

3. Proposed Vector Map Watermarking

Our method is based on ESRI shapefile, which is a popular geospatial vector data format for GIS maps, even though GIS data can be structured in different formats. The main features of our method are (1) the use the arc length of polylines and polygons in a map for embedding the geometric properties, and (2) the clustering of their lengths into groups using GMM clustering. The watermark is embedded by changing the mean of the arc lengths in a group. Figure 2 shows the process for watermark embedding and extraction.

3.1 Watermark Embedding

The embedding process consists of three steps: use of the arc length of polylines and polygons, GMM-based clustering, and watermark embedding, as shown in Fig. 2. We explain these steps as follows. Let $\mathbf{P}$ be the set of polylines in a map, $\mathbf{P} = (\mathbf{p}_i)_{i\in[1,N]}$ where N is the number of polylines.

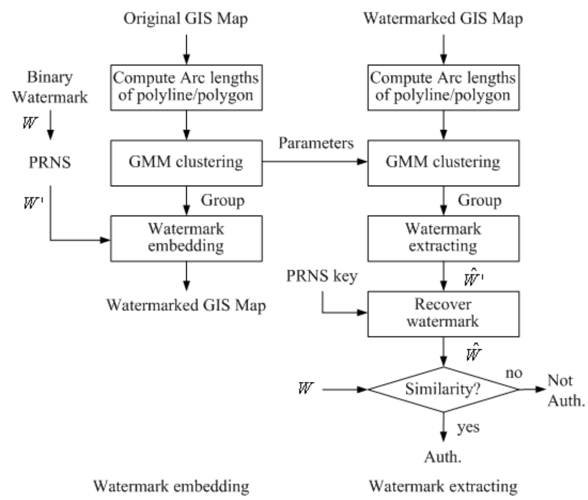


Fig. 2 The process of watermark embedding and extracting in our method.

Let $\mathbf{p}_i$ be ith polyline that includes N_i consecutive vertices, so $\mathbf{p}_i = (\mathbf{v}_{ij})_{j\in[1,N_i]}$. Since a polygon is a closed polyline, we will hereafter refer to a polygon as a polyline.

3.1.1 Arc Length

The arc length is invariant under the isometries of $\mathbf{R}^n$. An isometry of $\mathbf{R}^n$ is a distance-preserving map, i.e., a map $f : \mathbf{R}^n \to \mathbf{R}^n$ with the property that $\|f(x) - f(y)\| = \|x - y\|$. Let C be a curve in Euclidean space $\mathbf{X} \in \mathbf{R}^n$, so that C is the image of a continuous function $f : [a, b] \to \mathbf{X}$ of the interval $[a, b]$ into $\mathbf{X}$.

Polylines $\mathbf{p}_i$ are considered as the approximation of any curve C_i in Euclidean space $\mathbf{R}^n$. From a partition $a = \mathbf{v}_{i1} < \mathbf{v}_{i2} < \cdots < \mathbf{v}_{iN_i-1} < \mathbf{v}_{iN_i} = b$ of the interval $[a, b]$, we obtain a finite collection of vertices $(f(\mathbf{v}_{ij}))_{j\in[1,N_i]}$ on the curve C_i. The arc length $L(\mathbf{p}_i)$ of the approximation polyline $\mathbf{p}_i$ on the curve C_i is defined as

$$L(\mathbf{p}_i) = \sup_{a=\mathbf{v}_{i1}<\cdots<\mathbf{v}_{iN_i}=b} \sum_{i=1}^{N_i-1} \|f(\mathbf{v}_{ij}) - f(\mathbf{v}_{ij+1})\| \tag{1}$$

where the supremum is taken over all partitions of $[a, b]$.

We normalize all polylines $\mathbf{P} = (\mathbf{p}_i)_{i\in[1,N]}$ using the average of the polyline lengths and obtain the arc lengths of all normalized polylines using the above equation.

3.1.2 Polyline Clustering

We cluster the polylines of the arc lengths using GMM-based EM clustering [19]. The sensitivity to the initial parameters obtained makes it possible for the security to be improved.

Let $\mathbf{G}$ be N_G groups of polylines, $\mathbf{G} = \{\boldsymbol{G}_k | k \in [1, N_G]\}$, meaning that N_G GMMs exist. Given arc lengths $(L(\mathbf{p}_i))_{i\in[1,N]}$ of all polylines $\mathbf{P}$ in a map, we find the parameters $\Psi = (\psi_k = (\omega_k, \mu_k, \sum_k))_{k\in[1,N_G]}$ that maximize the log likelihood. Let the distribution of $L(\mathbf{p}_i)$ be the sum of the Gaussian distributions $\{\phi(\mu_k, \sum_k)\}_{k=1}^{N_G}$ weighted by ω_k for N_G

GMMs. The weight ω_k satisfies the conditions $\omega_k > 0$ and $\sum_{k=1}^{N_G} \omega_k = 1$. First, we randomly select N_G initial parameters $(\phi_k^{(0)} = (\omega_k^{(0)}, \mu_k^{(0)}, \Sigma_k^{(0)}))_{k\in[1,N_G]}$ and calculate the initial log-likelihood $l_k^{(0)}$.

$$l_k^{(0)} = \frac{1}{N}\sum_{i=1}^{N} \log\left(\sum_{k=1}^{N_G} \omega_k^{(0)} \phi\left(L(\mathbf{p}_i)|\mu_k^{(0)}, \sum_k^{(0)}\right)\right) \tag{2}$$

In the E-step, we calculate the probability $pr_{ik}^{(t)}$ that $L(\mathbf{p}_i)$ is included in the kth Gaussian for t iterations.

$$pr_{ik}^{(t)} = \frac{\omega_k^{(t)}\phi(L(\mathbf{p}_i)|\mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_{k=1}^{N_G} \omega_k^{(t)}\phi(L(\mathbf{p}_i)|\mu_k^{(t)}, \Sigma_k^{(t)})} \tag{3}$$

where $\sum_{k=1}^{N_G} pr_{ik}^{(t)} = 1$. Next, in the M-step, we estimate the parameters $(\phi_k^{(t)} = (\omega_k^{(t+1)}, \mu_k^{(t+1)}, \Sigma_k^{(t+1)}))_{k\in[1,N_G]}$.

$$\omega_k^{(t+1)} = \frac{1}{N_G}\sum_{k=1}^{N_G} pr_{ik}^{(t)} \tag{4}$$

$$\mu_k^{(t+1)} = \frac{1}{\sum_{k=}^{N_G} pr_{ik}^{(t)}}\sum_{i=1}^{N} L(\mathbf{p}_i)pr_{ik}^{(t)} \tag{5}$$

$$\sum_k^{(t+1)} = \frac{1}{\sum_{k=1}^{N_G} pr_{ik}^{(t)}}\sum_{i=1}^{N} pr_{ik}^{(t)} A_{ik}^{(t)} (A_{ik}^{(t)})^T \tag{6}$$
$$\text{where } A_{ik}^{(t)} = L(\mathbf{p}_i) - \mu_k^{(t)}$$

We then calculate the log-likelihood $l_k^{(t+1)}$ for $t+1$ iterations and obtain the final parameters until $l_k^{(t+1)} \simeq l_k^{(t)}$. From the above process, we obtain N_G GMM-based groups by finding the set of $(\phi_k^{(t)})_{k\in[1,N_G]}$ that maximizes the log-likelihood of the arc length.

$$\boldsymbol{G}_k = \{\mathbf{p}_i : pr(L(\mathbf{p}_i)|\phi_k^{(t)}) > pr(L(\mathbf{p}_i)|\phi_n^{(t)}), \quad \forall k \neq n, (k,n) \in [1, N_G]\} \tag{7}$$

We store the set of parameters Ψ for the watermark extraction.

3.1.3 Bit Embedding

Given a watermark $\mathbf{w} = (w_k)_{k\in N_W}$ of N_W bits $(mN_W < N_G)$ and m repetitions, we embed a bit w_i into the arc lengths of m groups to increase the robustness of the watermark against attacks. The more the m repetitions increase, the more the robustness increases, and the less the length of the watermark decreases. N_G and m depend on the number of polylines. If we assume that the watermark is a character message of 8 bits, the message length will be $N_G/8m$. We set m according to the length of the watermark message and the degree of robustness desired.

For purposes of security, we change the watermark bits using a PRNS generated from a non-uniform generator [20]. The sequence can be generated using a uniform distribution pseudo-random number generator (PRNG) and a function that relates the target distribution and uniform distribution.

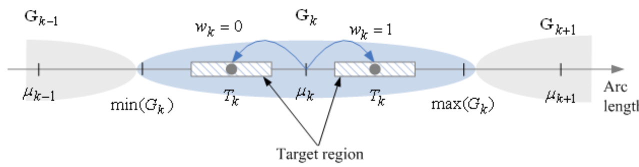


Fig. 3 Embedding a bit w_k by changing the mean arc length of a group $\boldsymbol{G}_k$.

It is very difficult to estimate the PRNS without knowledge of the target distribution. We use the inverse of a cumulative Gaussian distribution $erf^{-1}(x)$ of the target distribution with an ideal uniform PRNG with a range of $[0, 1]$ as an input x and generate a sequence of binary values in a Gaussian distribution. We denote the PRNS with N_W binary values as $\mathbf{c} = (c_k)_{k\in N_W}$. We encrypt the watermark by PRNS using a simple XOR cipher: $\mathbf{w}' = \mathbf{w} \oplus \mathbf{c}$.

Let us look at the process of embedding a watermark bit w_k in a group $\boldsymbol{G}_k$. First, we set the target region $\mathbf{T}_k$ within the range of an arc length of $\boldsymbol{G}_k$ according to w_k as follows.

$$\mathbf{T}_k = \begin{cases} \lfloor T_k - \epsilon_{k0}, T_k + \epsilon_{k0} \rfloor, & \text{if } w_k = 0 \\ \lfloor T_k - \epsilon_{k1}, T_k + \epsilon_{k1} \rfloor, & \text{if } w_k = 1 \end{cases} \tag{8}$$
$$\text{where } \epsilon_{k0} = \frac{1}{2}|\mu_k - min(\boldsymbol{G}_k)|, \epsilon_{k1} = \frac{1}{2}|\mu_k - max(\boldsymbol{G}_k)|$$

T_k is the target value to which the mean arc length μ_k moves by w_k. We set the target value as the center point of the left or right section, as shown in Fig. 3,

$$T_k = \begin{cases} \frac{1}{2}(\mu_k + min(\boldsymbol{G}_k)), & \text{if } w_k = 0 \\ \frac{1}{2}(\mu_k + max(\boldsymbol{G}_k)), & \text{if } w_k = 1 \end{cases} \tag{9}$$

where $min(\boldsymbol{G}_k)$ and $max(\boldsymbol{G}_k)$ are the minimum and maximum of the arc lengths in a group $\boldsymbol{G}_k$.

The mean arc length μ_k moves to the target region $\mathbf{T}_k$ according to a bit w_k. To achieve this, we change the arc lengths that are not in the target region so that they approach the target value by linear interpolation.

$$L'(\mathbf{p}_i) = \alpha_k L(\mathbf{p}_i) + (1-\alpha_k)T_k, \quad \text{if } L(\mathbf{p}_i) \neq \mathbf{T}_k, \forall i \in [1, N_G] \tag{10}$$

α_k is the scaling factor, so that $L(\mathbf{p}_i)$ approaches $L'(\mathbf{p}_i)$ below the quality preservation. Vertices in any polyline $\mathbf{p}_i = (\mathbf{v}_{ij})_{j\in[1,N_i]}$ must be changed so that $L(\mathbf{p}_i)$ moves to $L'(\mathbf{p}_i)$.

We segment a polyline $\mathbf{p}_i$ into two parts, which are to the right and left of the center point $\mathbf{v}_{i\lceil N_i/2\rceil}$, and we expand or compress the segment lines in two parts, as shown in Fig. 4. Given a segment line l_{ij} with two vertices $(\mathbf{v}_{ij}, \mathbf{v}_{ij+1})$ in $\mathbf{p}_i$, this line moves to the target line l'_{ij}.

$$l'_{ij} = \alpha_{ij} l_{ij} + (1-\alpha_{ij})\frac{T_k}{L(\mathbf{p}_i)}, \quad \text{if } 0 \le \alpha_{ij} \le 1, \forall j \in [1, N_i - 1] \tag{11}$$

$\frac{T_k}{L(\mathbf{p}_i)}$ is the target line for l_{ij}, and α_{ij} is the scaling factor for

l_{ij}. If l_{ij} is greater than $\frac{T_k}{L(\mathbf{p}_i)}$, it is compressed to l'_{ij}; otherwise, it is expanded to l'_{ij}. The expansion or compression of a segment line l_{ij} can be accomplished by changing two vertices $(\mathbf{v}_{ij}, \mathbf{v}_{ij+1})$ as follows.

$$\mathbf{v}'_{ij} = \mathbf{v}_{ij} + sgn(l_{ij} - \frac{T_k}{L(\mathbf{p}_i)})\beta_{ij}\hat{\mathbf{u}}_{ij} \tag{12}$$

$$\mathbf{v}'_{ij+1} = \mathbf{v}_{ij+1} - sgn(l_{ij} - \frac{T_k}{L(\mathbf{p}_i)})\beta_{ij}\hat{\mathbf{u}}_{ij} \tag{13}$$

$\hat{\mathbf{u}}_{ij}$ is the unit vector of a segment line l_{ij}, and $sgn()$ is the sign function. β_{ij} is the scaling factor for a vertex $\mathbf{v}_{ij}$. It is selected so that l_{ij} is expanded or compressed to l'_{ij} in the direction of $\hat{\mathbf{u}}_{ij}$. If $sgn(l_{ij} - \frac{T_k}{L(\mathbf{p}_i)})$ is negative, it indicates expansion; otherwise, it indicates compression. Given a line l'_{ij} that is changed by $\mathbf{v}'_{ij}$ or $\mathbf{v}'_{ij+1}$ according to Eqs. (12), (13), β_{ij} can be solved as follows.

$$\beta_{ij} = sgn(l_{ij} - \frac{T_k}{L(\mathbf{p}_i)})(l_{ij} - \sqrt{4l'^2_{ij} - 3l^2_{ij}}) \tag{14}$$

In the case of expansion, β_{ij} is $\sqrt{4l'^2_{ij} - 3l^2_{ij}} - l_{ij}$, regardless of α_{ij} in Eq. (11), because l'_{ij} is always higher than $\sqrt{3}l_{ij}/2$. On the other hand, in the case of compression, α_{ij} must be determined so that $l'_{ij} > \sqrt{3}l_{ij}/2$ as follows: $\alpha_{ij} > |\frac{T_k}{L(\mathbf{p}_i)} - \frac{\sqrt{3}l_{ij}}{2}|/|\frac{T_k}{L(\mathbf{p}_i)} - l_{ij}|$. In order to change l_{ij} to l'_{ij} progressively, we set two scaling factors for α_{ij}, β_{ij}, as follows.

$$\beta_{ij} = sgn(l_{ij} - \frac{T_k}{L(\mathbf{p}_i)})(l_{ij} - \sqrt{4l'^2_{ij} - 3l^2_{ij}})/C_\beta \tag{15}$$

$$\alpha_{ij} = \begin{cases} C_\alpha, & \text{if } sgn(l_{ij} - \frac{T_k}{L(\mathbf{p}_i)}) = -1 \\ 1.5C_\alpha, & \text{if } sgn(l_{ij} - \frac{T_k}{L(\mathbf{p}_i)}) = 1 \text{ and } \delta_{ij} > 1.5C_\alpha \\ \delta_{ij}, & \text{if } sgn(l_{ij} - \frac{T_k}{L(\mathbf{p}_i)}) = 1 \text{ and } \delta_{ij} < 1.5C_\alpha \end{cases} \tag{16}$$

$$\text{where } \delta_{ij} = 1.2(|\frac{T_k}{L(\mathbf{p}_i)} - \frac{\sqrt{3}l_{ij}}{2}|)/(|\frac{T_k}{L(\mathbf{p}_i)} - l_{ij}|)$$

In our paper, the parameter C_β for β_{ij} is 5 and the parameter C_α for α_{ij} is 0.6. Given segment lines in a polyline $\mathbf{p}_i$, we change the first line $l_{i,1}$ using the first vertex $\mathbf{v}_{i1}$ and the last line l_{i,N_G-1} using the last vertex $\mathbf{v}_{iN_G}$ by either expansion or compression, and then change the remaining segment lines sequentially.

Using the above method, we change all the polylines in a group so that the mean arc length μ_k moves to the target region $\mathbf{T}_k$ according to a bit w_k.

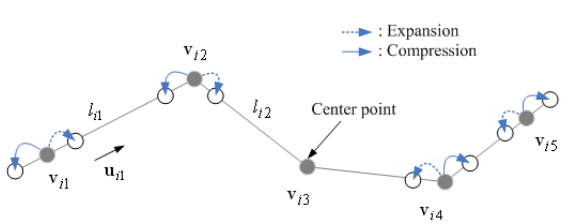


Fig. 4 Expansion or compression of lines on the center point in any polyline.

3.2 Watermark Extracting

Given any pirated map, our method first computes the arc lengths of all the normalized polylines in its map. Our method then clusters them into N_G groups using the parameters $\Psi = (\psi_k = (\omega_k, \mu_k, \sum_k))_{k\in[1,N_G]}$ that are stored in the embedding process. The clustering is performed as a one-time process using Ψ and N_G so that $\Psi' = (\psi'_k = (\omega'_k, \mu'_k, \sum'_k))_{k\in[1,N_G]}$ are generated. We obtain a sequence of N_G bits by comparing μ_k with μ'_k; $(b_k = sgn(\mu'_k - \mu_k))_{k\in[1,N_G]}$ and then obtain the encrypted watermark of $\lfloor N_G/m \rfloor$ bits using a unit of m bits, $\mathbf{w}' = (w'_k = \sum_{j=1}^{m} b_{3k+j}/m)_{k\in[1,\lfloor N_G/m\rfloor]}$. Finally, we generate the watermark $\mathbf{w}^*$ from the XOR operator of $\mathbf{w}'$ and the PRNS sequence $\mathbf{c}$: $\mathbf{w}^* = \mathbf{w}' \oplus \mathbf{c}$.

4. Experimental Results

Our experiment used 10 maps with the ESRI shape file format and repeated the watermarking process 100 times on each map. We compared our method's invisibility and robustness to those of the conventional methods of Ohbuchi [1], Chang [12], and Lee [16]. Typical vector-map attacks that preserve the object shape include the similarity transformation of RST, reordering (re-parameterization) of objects or vertices, and object simplification. Anyone can easily apply these attacks to a vector map using popular editing tools. The conventional methods of Ohbuchi [1], Chang [12], and Lee [16] are robust to similarity transformation, reordering, and simplification. Therefore, our experiment chose these methods for the comparison. We applied our method, Ohbuchi's method, and Lee's method to all polyline and polygon objects and applied Chang's method to polygon objects. Each of maps consists of only polyline or polygon. Table 1 shows the number of polyline and polygon of test maps, N, the number of groups, N_G, and the length of watermark, N_W.

4.1 Invisibility

The test maps contained various numbers of polylines or

Table 1 The number of polylines or polygons, the number of groups, the length of watermark, and vPSNR in each of test maps.

Type	Map (Scale)	Number N	Number of group N_G	Watermark length N_W	vPSNR [dB]			
					Proposed	Ohbuchi	Chang	Lee
Polyline	1:1,000	2,831	566	113	60.14	51.66	-	59.58
	1:5,000	3,563	712	142	69.50	60.47	-	67.09
	1:10,000	3,734	746	149	66.00	58.60	-	63.20
	1:50,000	10,622	2,124	424	66.27	60.42	-	62.46
	1:100,000	46,911	9,382	1,876	82.03	75.07	-	81.88
Polygon	1:1,000	10,346	2,069	413	64.42	55.02	57.16	58.39
	1:5,000	13,486	2,697	539	80.78	71.84	66.84	79.89
	1:10,000	15,260	3,052	610	74.48	62.23	62.37	72.07
	1:50,000	8,003	1,600	320	68.19	67.27	60.57	62.28
	1:100,000	46,106	9,221	1,844	83.01	74.84	67.76	79.07

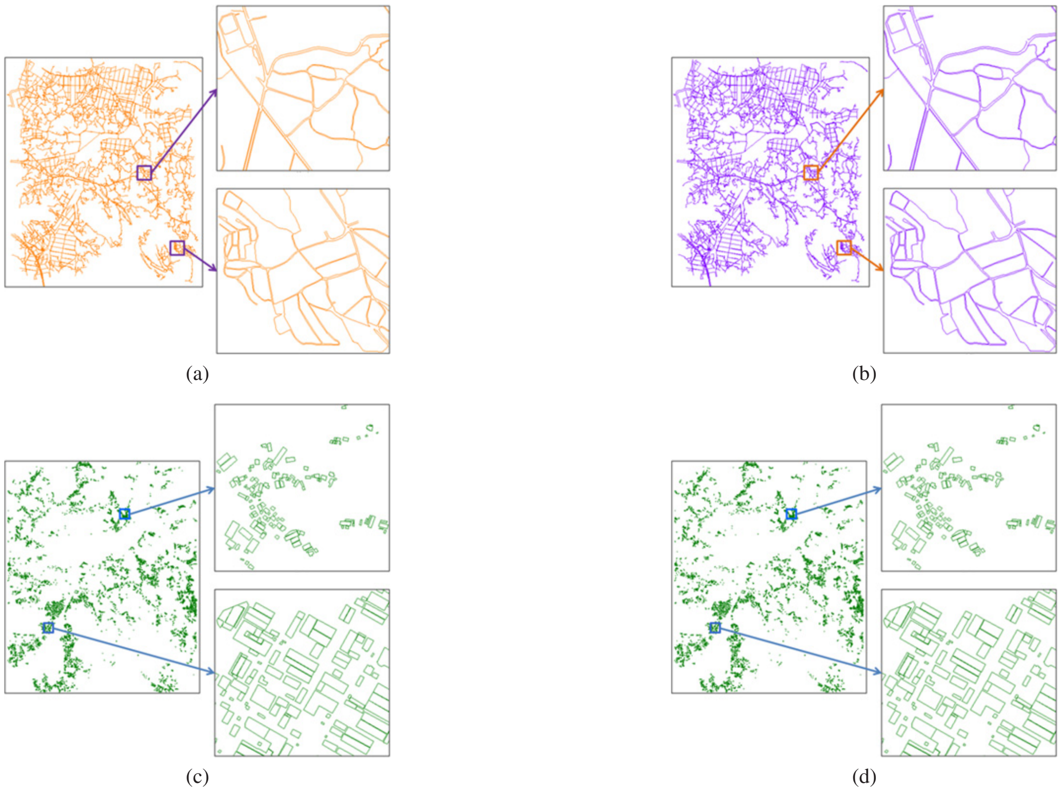


Fig. 5 (a) Original polyline map (1;5,000), (b) watermarked polyline map, (c) original polygon map (1;5,000), and (d) watermarked polygon map.

polygons. We determined the length of the watermark according to the number of polylines or polygons in the test maps. We denote the ratio r as $r = N_G/N$, which indicates the number of groups per number of polylines. Given the repetitions of m and the ratio r, the length of watermark is

$$N_W = \lfloor \lfloor rN \rfloor / m \rfloor \tag{17}$$

It is directly proportional to r and inversely proportional to m. We experimentally set r to 0.2 and m to 5 considering the factor of robustness. Therefore, the data capacity is constant, as follows.

$$C = \frac{N_W}{N} \approx \frac{r}{m} = 0.04[\text{bit/polyline (polygon)}]] \tag{18}$$

To evaluate the objective invisibility, we used the vertex PSNR as defined by

$$vPSNR = 20 \log_{10} \frac{max(\|\mathbf{v}\|)}{RMSE} [\text{dB}] \tag{19}$$

where $max(\|\mathbf{v}\|)$ is the maximum distance between an origin and any vertex, and RMSE is the root-mean-square error between the original vertices $\mathbf{v}_{ij}$ and the watermarked vertices $\mathbf{v}'_{ij}$.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \sum_{j=1}^{N_{ij}} \|\mathbf{v}_{ij} - \mathbf{v}'_{ij}\|^2}{N \times N_{ij}}} \tag{20}$$

The results of vPSNR are shown in Table 1. As can be seen in the table, all vPSNRs are above 60 [dB], which means that the distance errors are within the maximum allowable error stated by NGII [21] and GITTA [22]. The vPSNRs of our method are 0.92–17.40 [dB] higher than those of Obhuchi's method, 7.26–13.94 [dB] higher than those of Chang's method, and 0.15–12.80 [dB] higher than those of Lee's method. Figures 5 (a) and 5 (b) show a polyline map and a watermarked polyline map, respectively. In addition, Figs. 5 (c) and 5 (d) show a polygon map and a watermarked polygon map, respectively. These results verify that it is very difficult for the difference between an original map and a watermarked map to be detected perceptually.

4.2 Robustness

A watermarked map can be attacked intentionally or unintentionally using various editing tools. The common attacks are the similarity transform of RST (Rotation, Scale and Translation), order scrambling, swapping, random noise, addition or deletion of vertices or objects, and cropping. In our experiment on robustness, we used various editing tools in AutoCAD Map 3D 2009 software and Arcview software for the attacks, and used the bit error rate (BER) for comparing our method with the conventional methods. The BERs for each of the attacks are shown in Fig. 6 and Fig. 7.

For the similarity transform, we rotated the watermarked maps to any angle, translated them to any position, and scaled them by a scale factor, which ranged from 0.5 to

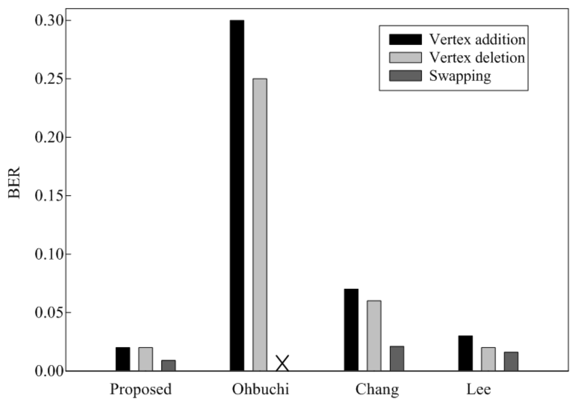
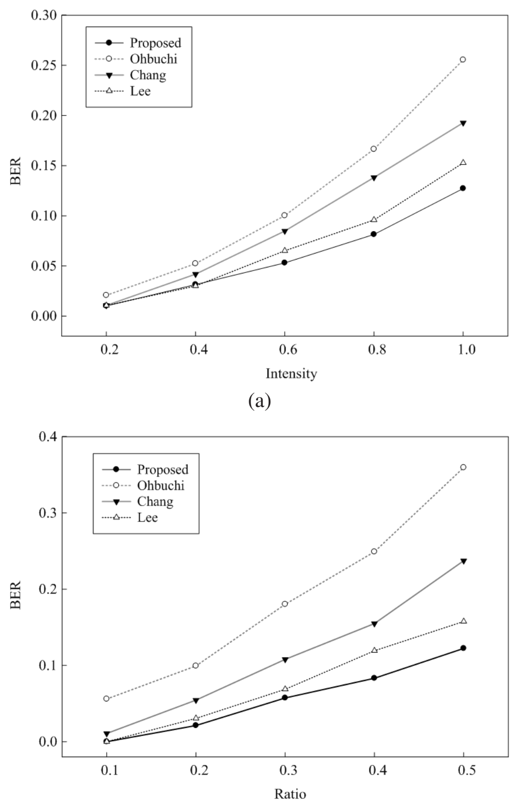


Fig. 6 BERs in vertex addition, vertex deletion and swapping.



(a)

(b)

Fig. 7 BERs in (a) random noise and (b) cutting.

3.0. Neither our method nor the conventional methods were affected by the rotation and translation. In addition, our method, as well as Chang's and Lee's methods, were robust to scaling because of their normalization process, without knowledge of the scaling factor. With Ohbuchi's method, however, we must know the scaling factor for watermark detection.

Objects of polylines and polygons, vertices in objects, and layers in a map are ordered in a way that is user-defined. Therefore, their order is scrambled randomly without degrading the map quality. All the methods tested were not affected by order scrambling and were able to detect all watermarks without error.

Vertex addition is similar to re-parameterization, and vertex deletion is similar to polyline and polygon simplification. We added or deleted 20% of the vertices in all the polylines or polygons, without preserving the shape of the polylines and polygons. The BERs obtained using our method were from 0.01 to 0.05 less than those obtained using Chang's and Lee's method, and from 0.23 to 0.28 less than that obtained using Ohbuchi's method. The reason for this lower rate is that the arc length distribution and the repetitive embedding of our method are more robust than the geometrical properties of conventional methods.

We segmented a map into two parts and switched them such that the positions of all the polylines and polygons were changed, while their shapes were preserved. Figures 8 (a) and 8 (b) show the switched polyline and polygon maps. With Ohbuchi's method, the positions of the two parts are important, so this method could not extract the watermark in a switched map. Using our method, the arc length was not affected much by the switch, and using Chang's and Lee's methods, the geometrical properties were not much affected. The BERs obtained with all three methods were very low.

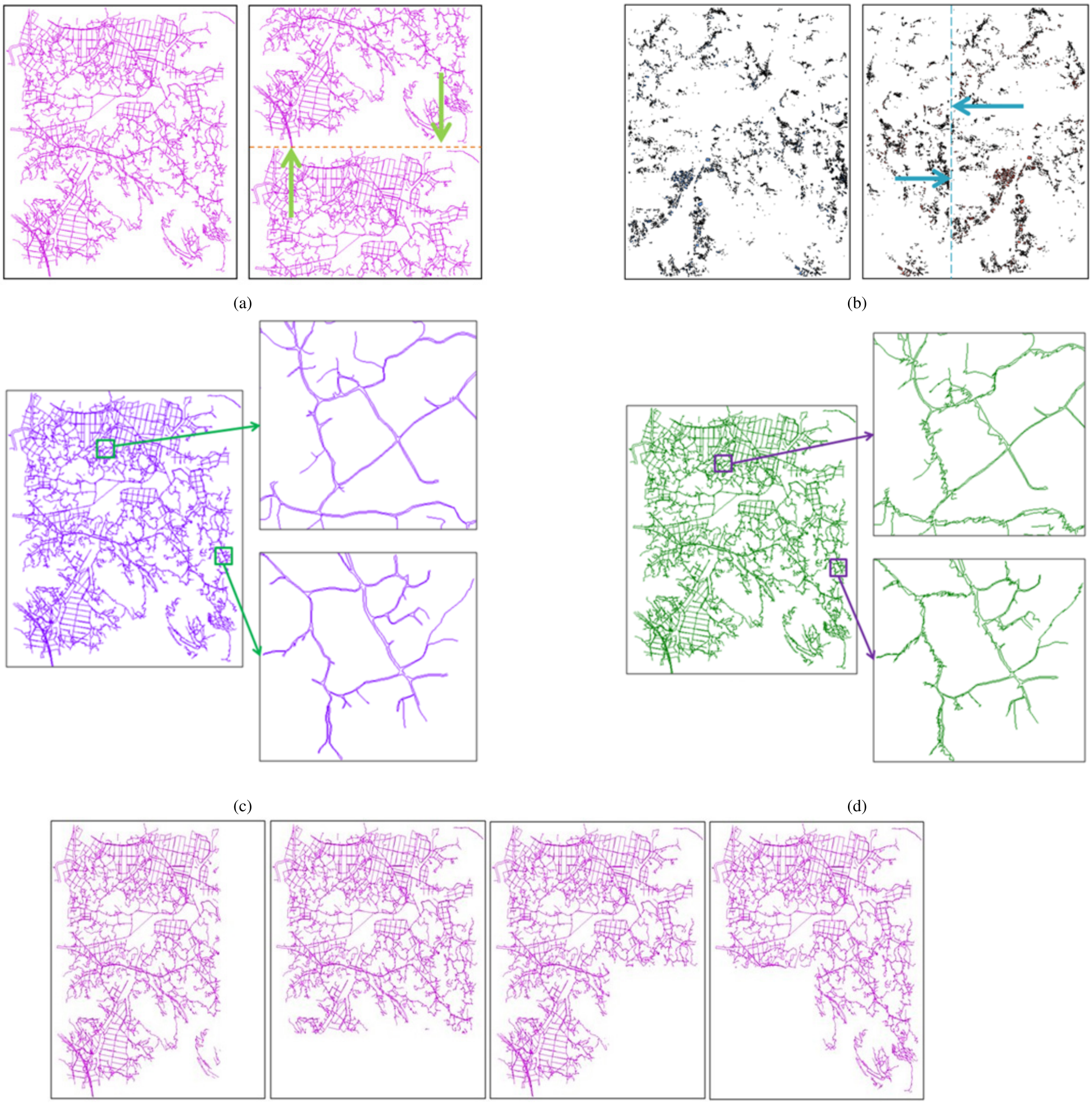
We added random noise to all the vertices by varying the intensity. Given a vertex $\mathbf{v}$ and a map scale s, the noised vertex is $\mathbf{v}' = \mathbf{v} + (10^{-4} s \times rand() \times \alpha)\hat{\mathbf{u}}$, where $\hat{\mathbf{u}}$ is the unit vector of a random direction, and $rand()$ is the uniform distribution on $[0, 1]$. This means that $\mathbf{v}$ is changed to any point in a random direction within $[0, 10^{-4} s\alpha]$ meters of scale s. We varied the intensity α from 0.1 to 1. Figures 8 (c) and 8 (d) show the polyline map in which random noise with $\alpha = 0.8$ is added. The BERs are shown in Fig. 7 (a). As can be seen in the figure, the BERs obtained using our method are from 4.8×10^{-4} to 1.28×10^{-1} less than those obtained using conventional methods.

We cut arbitrary regions from a map that ranged from 10% to 50% of its area. Though the map was cut by this attack, our method was able to detect the watermark from the polylines or polygons in the remaining area. Figure 8 (e) shows polyline maps from which 25% of the area in four directions was cut. The BERs are shown in Fig. 7 (b). As can be seen in the figure, the BERs obtained using our method are from 9.46×10^{-3} to 2.37×10^{-1} less than those obtained using conventional methods.

From the above results, we verified that our method is more robust to attacks than conventional methods.

5. Conclusions

This paper proposed an invisible, blind, and robust watermarking method for the copyright protection of GIS vector digital maps using the arc length distribution. Our method computes the arc lengths of the polylines or polygons in a map and clusters the arc lengths into a number of groups using the GMM-EM clustering algorithm. Our method then encrypts the watermark simply using PRNS and changes the arc length distribution in each group according to the watermark. From our experimental results, we verified that our



(a) (b) (c) (d) (e)

Fig. 8 (a) Watermarked polyline map (1;5,000) and swapped polyline map, (b) watermarked polygon map (1;5,000) and swapped polygon map, (c) Watermarked polyline map and (d) random noised map with intensity 0.8, (e) polyline maps whose 25% regions are cut.

method is invisible, secure, and robust, and thus serves well for the copyright protection of GIS vector maps. Our future works include security analysis based on differential entropy, and a GIS watermarking system that combines a satellite image map and a vector map.

References

[1] R. Ohbuchi, "Robust watermarking of vector digital maps," Proc. IEEE International Conference on Multimedia and Exp., vol.1, pp.577–580, Lausanne, Switzerland, 2002.

[2] R. Ohbuchi, H. Ueda, and S. Endoh, "Watermarking 2D vector maps in the mesh-spectral domain," Shape Modeling International, pp.216–228, Seoul, Korea, 2003.

[3] G. Schulz and M. Vogit, "A high capacity watermarking system for digital maps," Proc. 200 Multimedia and Security Workshop on Multimedia and Security, Magdeburg, pp.180–186, Germany, 2004.

[4] M. Vogit and C. Busch, "Watermarking 2D-Vector data for geographical information systems," Proc. SPIE, Security and Watermarking of Multimedia Content, vol.4675, pp.621–628, San Jose, USA, 2002.

[5] M. Vogit and C. Busch, "Feature-based watermarking of 2D-vector data," Proc. SPIE, Security and Watermarking of multimedia Content, vol.5020, pp.359–366, SantaClara, USA, 2003.

[6] K.T. Park, K.I. Kim, H.I. Kang, and S.S. Han, "Digital geographical map watermarking using polyline interpolation," PCM '02: Proc. Third IEEE Pacific Rim Conference on Multimedia, pp.58–65, London, UK, 2002.

[7] H.I. Kang, K.I. Kim, and S.S. Han, "Watermarking techniques using

the Drawing Exchange Format (DXF) file," Proc. 2001 Multimedia and Security Workshop on Multimedia and Security, pp.19–22, Ottawa Canada, 2001.
[8] D. Zhang, D. Qian, and P. Han, "A new attributes-priority matching watermarking algorithm satisfying topological conformance for vector map," Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIHMSP 2007, vol.2, pp.469–472, Nov. 2007.
[9] D. Shujun, L. Liang, and C. Sen, "Research on a digital watermarking algorithm suitable to vector map," IEEE International Conference on Automation and Logistics, pp.1236–1240, Aug. 2007.
[10] Y. Dakroury, I.A. El-ghafar, and A. Tammam, "Protecting GIS data using cryptography and digital watermarking," Int. J. Computer Science and Network Security, vol.10, no.1, pp.75–84, Jan. 2010.
[11] H.-J. Chang, J.-H. Kim, S.-H. Lee, K.-S. Moon, B.-K. Cha, and K.-R. Kwon, "A digital watermarking of GIS data using the polyline characteristics," Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV08), pp.8–13, Jan. 2008.
[12] H.-J. Chang, B.-J. Jang, S.-H. Lee, S.-S. Park, and S.-G. Kwon, "3D GIS vector map watermarking using geometric distribution," Proc. IEEE International Conference on Multimedia and Expo, pp.1014–1017, NewYork, June-July 2009.
[13] J.-H. Kim, S.-H. Lee, S.-G. Kwon, S.-S. Park, and K.-R. Kwon, "A polyline watermarking scheme for GIS vector map," J. Korea Multimedia Society, vol.13, no.4, pp.582–593, April 2010.
[14] W. Xun, H. Ding-jun, and Z. Zhi-yong, "A robust zero-watermarking algorithm for 2D vector digital maps," Computer, Informatics, Cybernetics and Applications, Lecture Notes in Electrical Engineering, vol.107, pp.533–541, 2012.
[15] C.Y. Shao, H.L. Wang, X.M. Niu, and X.T. Wang, "A shape-preserving method for watermarking 2D vector maps based on statistic detection," IEICE Trans. Inf. & Syst., vol.E89-D, no.3, pp.1290–1293, March 2006.
[16] S.-H. Lee and K.-R. Kwon, "Vector watermarking scheme for GIS vector map management," Multimedia Tools and Applications, vol.63, issue 3, pp.757–790, April 2013.
[17] Shapefile, http://en.wikipedia.org/wiki/Shapefile, Accessed to Nov. 2012.
[18] ESRI, ESRI shapefile technical description, An ESRI White paper, July 1998.
[19] M.R. Gupta and Y. Chen, "Theory and use of the EM algorithm," Foundations and Trends in Signal Processing, vol.4, no.3, pp.223–296, 2010.
[20] L. Devroye, Non-Uniform Random Variate Generation, Springer, 1986.
[21] National Geographic Information Institute (NGII), http://www.ngii.go.kr/eng/index.do [Accessed Oct. 2012].
[22] Geographic Information Technology Training Alliance (GITTA), Position Accuracy, http://www.gitta.info/MetaDataQual/en/multimedia/PositAccuracy.pdf, [Accessed Oct. 2012].

Suk-Hwan Lee received a B.S., a M.S., and a Ph.D. degree in Electrical Engineering from Kyungpook National University, Korea in 1999, 2001, and 2004 respectively. He worked at Electronics and Telecommunications Research Institute in 2005. He is currently an associate professor in Department of Information Security at Tongmyong University, which he started in 2005. He works as an editor of Korea multimedia society journal and is a member of IEEE, IEEK, IEICE and also is an officer of IEEE R10 Changwon section. His research interests include multimedia signal processing, multimedia security, digital signal processing, bio security, and computer graphics.

Xiao-Jiao Huo received a B.S. in Dalian Polytechnic University, China in 2008 and a M.S. in Pukyong National University, Korea in 2011. Her research interests are GIS security, digital signal processing, image processing, and digital watermarking.

Ki-Ryong Kwon received the B.S., M.S., and Ph.D. degrees in Electronics Engineering from Kyungpook National University in 1986, 1990, and 1994 respectively. He worked at Hyundai Motor Company from 1986-1988 and at Pusan University of Foreign Language form 1996–2006. He is currently a professor in Department of IT Convergence and Application Engineering at the Pukyong National University. He visited University of Minnesota at 2000–2001 and Colorado State University at 2011–2012 with visiting professor in USA. He is currently the General Affair and Organizing Committee/Vice President in Journal of Korea Multimedia Society. also is a director of IEEE R10 Changwon section. His current research interests are in the area of digital image processing, multimedia security and digital watermarking, wavelet transform.