Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2015, Article ID 421529, 6 pages
http://dx.doi.org/10.1155/2015/421529

Research Article

A Zero-Watermarking Scheme for Vector Map Based on Feature Vertex Distance Ratio

Yuwei Peng and Mingliang Yue

Computer School, Wuhan University, Wuhan 430072, China

Correspondence should be addressed to Yuwei Peng; ywpeng@whu.edu.cn

Received 24 January 2015; Revised 26 May 2015; Accepted 7 June 2015

Academic Editor: D. I. Laurenson

With the rapid development of GIS and computer techniques, vector map data has been widely used in many fields. Since the production of map data is very costly, illegal copying will result in huge loss for data owners. In order to protect the copyright of vector data, digital watermarking has been employed in recent years. In this paper, a zero-watermarking scheme for vector map data is proposed. In the proposed scheme, FVDR (feature vertex distance ratio) is constructed based on the feature vertices of objects. The feature data, FVDR, is combined with watermark to generate the zero-watermark. Due to the specially designed cover data, the proposed scheme is robust to geometrical attacks, vertex attacks, and object attacks. The results of extensive experiments also demonstrate the robustness of the proposed scheme.

1. Introduction

The digital map is one type of important digital resources which has been widely used in navigation, urban planning, and many other areas. Due to high processing cost in the acquisition of digital map data, it becomes a valuable resource to its owner and has high price. Nevertheless, as one kind of digital data, a digital map could be easily copied. Therefore, copyright protection techniques for digital map have received extensive research attention in recent years.

As an effective approach for copyright protection, watermarking has been naturally introduced in the protection of digital map. Although many watermarking schemes have been proposed [1–7], most of them modify the map more or less to embed the watermark in it. For a vector map, which is sensitive to the precision, any form of distortion is insufferable. In order to make up for the shortage, zero-watermarking has been employed in digital map watermarking [8]. Like robust watermarking techniques, zero-watermarking comes from the area of image watermarking. The concept of zero-watermarking was first proposed in [9]. The main idea is generating a watermark from the characteristics of data without any modification on the host data. The generated watermark will be registered in an IPR (Intellectual Property Right) repository to facilitate future watermark detection. However, the scheme proposed in [8] is only robust to geometrical attacks.

In this work, we focus on a watermarking scheme for a particular kind of object in digital map, polyline. Specifically, our goal is to generate a watermark from a set of polylines, without distortion of main characteristics and with resistance to malicious attacks. To summarize, the major contributions of this paper lie in the following aspects:

(1) We construct special cover data, called the feature vertex distance ratio (FVDR), from the characteristics of polylines and employ it to determine the watermark. Also, FVDR has been proven to be invariable against most of malicious attacks.

(2) Based on FVDR, we propose a method to map each bit of the watermark to a polyline in the host data. Due to FVDR's characteristics, the mapping relationship is robust to most of malicious attacks.

(3) A majority voting mechanism is adopted to generate watermark bits from corresponding groups of polylines, which provides robustness to both vertex and object attacks.

(4) Finally, we propose a zero-watermarking scheme for polylines. The scheme employs the aforementioned

mapping method to determine corresponding watermark bit for each polyline. Then FVDRs are calculated for generation of the corresponding bits. As shown by the experimental results, the proposed scheme is robust to geometrical attacks, vertex attacks, and object attacks.

The remainder of this paper is organized as follows. In Section 2, we present the proposed zero-watermarking scheme for polylines. Then, generation and detection algorithms are given in Section 3. In Section 4, we present experimental results and discuss their implications. Finally, the paper is concluded with final remarks and an outlook on future work in Section 5.

2. Proposed Zero-Watermarking Scheme

In the proposed zero-watermarking scheme, the watermark is a sequence of L bits. Each object in the vector map data, that is, a polyline, will be mapped to one bit of the watermark. Usually, since the number of objects is much bigger than L, some objects would be mapped to one same bit. To generate each bit of the watermark, the cover data of the objects mapped to it should be extracted first. Then the bit can be generated from the extracted cover data. Finally, the generated watermark will be registered in an IPR repository.

In this section, we detail three essential steps of the proposed scheme: (1) feature point selection; (2) cover data extraction; (3) watermark generation. Then, we present the watermark generation and detection algorithms based on the proposed scheme in the next section.

2.1. Feature Point Selection. In the proposed scheme, we employ cover data based on feature points. The feature points of a polyline are the vertices that are most likely to represent the objects structure and cannot be arbitrarily added or removed under interpolation or simplification [10, 11]. Methods that extract their cover data based on feature points take full advantage of this property to resist vertex attacks [12, 13]. As a result, the cover data can properly resist interpolation and simplification attacks under the assumption that a successful attack should not violate the map's validity. For a polyline, the most used simplification algorithm is Douglas-Peucker (DP) algorithm. However, an absolute threshold is employed in the DP algorithm to select feature points. While the map has been scaled, the feature points selected by the same threshold would not be the same with those selected before scaling.

To solve this problem, in this paper, we use a relative threshold instead of the absolute one. For a polyline $G_i = \{p_i^1, p_i^2, \ldots, p_i^j\}$, where p_i^k are the vertices of the object, given a relative threshold α, the feature point selection is described as follows.

Step 1. Find the first feature point f_i^1 that is farthest from the line segment (namely, reference line) between p_i^1 and p_i^j. The distance between f_i^1 and the reference line is denoted by $d(f_i^1)$.

Step 2. Apply $\alpha * d(f_i^1)$ as the absolute threshold and select the rest feature points by using DP algorithm.

In the DP algorithm with relative threshold, at least one feature point will be selected for each object. Then the product of its distance to the reference line and α is utilized as the threshold for selection of the other feature points. By this means, although the distances between vertices and their reference lines are scaled, the computed threshold is also scaled with the same magnitude. Thus, the exact same set of feature points can be properly found even when the vector map data has been scaled.

2.2. Cover Data Extraction. Since the set of feature points can always be found as long as the object exists legally, now we can construct our cover data based on the feature points set.

Angle and topological relation have been chosen as cover data for they are insensitive to geometrical attacks [1, 14]. However, a small vertex distortion will contribute to a large variation of the angle when the reference lines are short, while any object attacks will violate the correspondences of the objects used for topological relations extraction. Distance can be extracted from individual objects and is less sensitive to vertex distortion [13], but after scaling watermark covered by distance will inevitably be violated. For the fact that ratio between distances will not change under scaling, in this paper, we extract cover data based on distance ratio.

Given a polyline object G_i, suppose the set of feature points is $F_i = \{f_i^1, f_i^2, \ldots, f_i^n\}$; the cover data, FVDR, is constructed as follows:

$$x_i = \sum_{j=1}^{n} \frac{f_i^j(x)}{n},$$
$$y_i = \sum_{j=1}^{n} \frac{f_i^j(y)}{n}. \quad (1)$$

Step 1. Use all points in F_i to calculate the feature center $C_i(x_i, y_i)$ as shown in (1).

Step 2. Calculate the distance between every feature point f_i^k and C_i, denoted as feature distance d_i^k, and let $D_i = \{d_i^1, d_i^2, \ldots, d_i^n\}$ represent the set of feature distances.

Step 3. Calculate the feature vertex distance ratio (FVDR), R_i, as shown in

$$R_i = \sum_{j=1}^{\lfloor n/2 \rfloor} \frac{d_i^{2j-1}}{d_i^{2j} * \lfloor n/2 \rfloor}. \quad (2)$$

Clearly, FVDR has the property of geometrical invariability and vertex independence. That is, the watermarking scheme using FVDR as cover data can properly resist geometrical attacks as well as interpolation and simplification attacks.

2.3. Watermark Generation. Generally, in a zero-watermarking scheme, the watermark should be extracted from the most stable characteristics of the host data in order to ensure the robustness. In the proposed scheme, we employed FVDR extracted in previous subsections to generate the watermark. Basically, each FVDR is mapped to one bit of the watermark. Generally, the number of FVDRs is larger than the length of the watermark. Therefore, some FVDRs would be mapped to the same bit. After the mapping process, each bit could be generated from the FVDRs mapped to it.

2.3.1. Watermark Bit Mapping. The process of mapping FVDRs to bits in the watermark is called "watermark bit mapping." The kernel idea is a hash function, as shown in

$$S_i = H_m\left(H_b\left(R_i, h\right), L, K\right). \tag{3}$$

In (3), S_i is the index for the corresponding bit of FVDR, R_i; that is R_i is mapped to the S_ith bit in the watermark. H_m is the mapping function which produces S_i according to L, K (secret key), and the highest h bits of R_i. The highest h bits are calculated by function H_b. The reason of using $H_b(R_i, h)$ rather than R_i is the fact that the possible modification on the object G_i might result in slight difference between R_i obtained in watermark generation and detection. This difference would afterwards defeat the watermark detection. However, such difference only exists on the low bits of R_i. With this observation, we employ a truncation function H_b to eliminate the difference. With H_b, the highest h bits of R_i are obtained to ensure the same R_i could be regenerated in watermark detection. To guarantee the usability of the watermarked map, the impact of the modification would be controlled below the precision tolerance. With the precision tolerance λ of the map, the upper and lower bounds of R_i (denoted as R_i^u and R_i^l, resp.) can be calculated by (4). Therefore, h is the largest number by which $H_b(R_i, h)$, $H_b(R_i^u, h)$, and $H_b(R_i^l, h)$ have the same value:

$$\begin{aligned} R_i^u &= \sum_{j=1}^{\lfloor n/2 \rfloor} \frac{d_i^{2j-1} + \lambda}{\left(d_i^{2j} - \lambda\right) * \lfloor n/2 \rfloor}; \\ R_i^l &= \sum_{j=1}^{\lfloor n/2 \rfloor} \frac{d_i^{2j-1} - \lambda}{\left(d_i^{2j} + \lambda\right) * \lfloor n/2 \rfloor}. \end{aligned} \tag{4}$$

The index of corresponding bit for each FVDR, R_i, can be obtained as follows.

(1) Calculate R_i^u and R_i^l based on R_i, and then the appropriate h can be determined with R_i, R_i^u, and R_i^l.

(2) Calculate the index of the corresponding bit by (3) and the h calculated above.

2.3.2. Watermark Bit Generation. For each group, we can extract one watermark bit from it. A majority voting mechanism is employed. The generation of watermark is described as follows.

Step 1. For each group, all hth position of FVDR is calculated.

Step 2. If there are more odd numbers than even numbers, the corresponding watermark bit is "1."

Step 3. If there are more even numbers than odd numbers, the corresponding watermark bit is "0."

After all watermark bits have been decided, they can be formed to the watermark according to their sequence number. And then the watermark can be registered into the IPR repository.

3. Generation and Detection Algorithms

Based on the steps described above, we can give the generation and detection algorithms in this section.

Let $G = \{G_1, G_2, \ldots, G_k\}$ be a set of polylines and let $W = \{w_1, w_2, \ldots, w_L\}$ be the watermark.

The generation algorithm includes the following steps.

Step 1. For every G_i in G, we calculate its FVDR, R_i.

Step 2. Use the highest h positions of R_i to map the object into a set of L mutually exclusive groups ($\{g_1, g_2, \ldots, g_L\}$); g_i corresponds to watermark bit, w_i.

Step 3. Generate the watermark bit, w_i, for group g_{i} and assemble watermark bits into the final watermark.

Step 4. Finally, the watermark generated is registered in an IPR repository.

The detection algorithm is similar to the generation algorithm.

Step 1. For every polyline, after division, a watermark bit is regenerated from each group. All regenerated watermark bits are assembled to a detected watermark.

Step 2. Compare the detected watermark with the one registered in IPR repository.

For every polyline, the algorithms need to invoke the DP algorithm for feature point selection. The time complexity of DP algorithm is $O(C)$, where C is the number of points of the polyline. So, the time complexity of both the generation and detection algorithm is $O(CN)$, where N is the number of polylines in the map.

4. Experiments and Analysis

To evaluate the performance of the proposed scheme, we performed experiments using a map of China which contains 3407 objects and 1128242 vertices.

In the experiments, attacks were applied to the watermarked map with different magnitudes. Then detection ratio was calculated by dividing the watermark length by the number of correct detected bits. Then the detection ratio was used for robust evaluation.

4.1. Geometrical Attacks. To assess the resilience of our algorithm to geometrical attacks, we apply a combination of translation, rotation, and scaling attacks to the watermarked map. Specifically, the magnitudes of the attacks are measured using (relative) coordinate offset, rotation angles, and scaling factors. And the ranges of the magnitudes were set to [−300%, 300%], [0, 360], and [0, 10], respectively. In this experiment, attack magnitudes were gradually increased with certain steps and all the detection ratios were the same and equal to 100%. The experiment result verifies the good performance of our method when facing geometrical attacks.

4.2. Vertex Attacks

4.2.1. Noise Distortion. In this experiment, we evaluate the influence of noise distortion. Attacks were applied by randomly modifying the coordinates with different magnitudes. Here we suppose the noise distortion satisfies uniform distribution and the range of the magnitudes was set to $[0, 1.2\tau]$, where τ is the positional accuracy. As shown in Figure 1, even when the attack magnitude exceeds the positional accuracy, the detection ratio remains acceptable. It verifies the robustness of the proposed method to noise distortion attack.

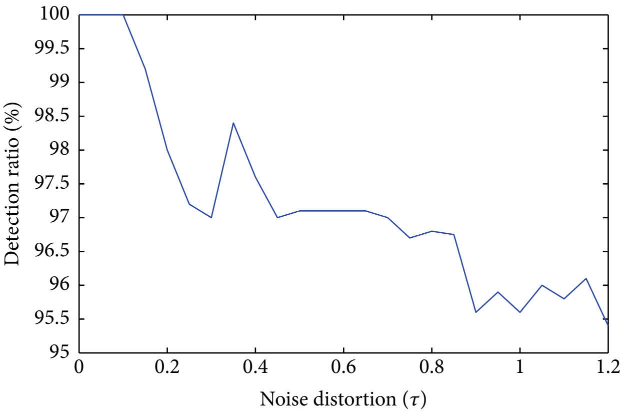

Figure 1: Resistance to noise distortion.

4.2.2. Interpolation and Simplification. In this experiment, interpolation and simplification were performed using Cubic Spline Algorithm [15] and Douglas-Peucker algorithm, respectively. Watermark data was generated using different relative thresholds. To evaluate the strength of the (interpolation and simplification) attack resistance, we gradually increased the attack magnitude and recorded the detection ratio of the corresponding thresholds. The results are presented in Figures 2 and 4, respectively. The attack magnitude was represented by the percentage of the vertices added or deleted due to interpolation or simplification. As we can see, the larger the relative threshold is, the more robust the algorithm is. And, given a properly selected relative threshold, the algorithms detection rate can remain 100% until the attack reaches a certain percentage of vertex addition or deletion (e.g., if the threshold is 0.06, the percentage of addition is 44% and the percentage of deletion is 36%).

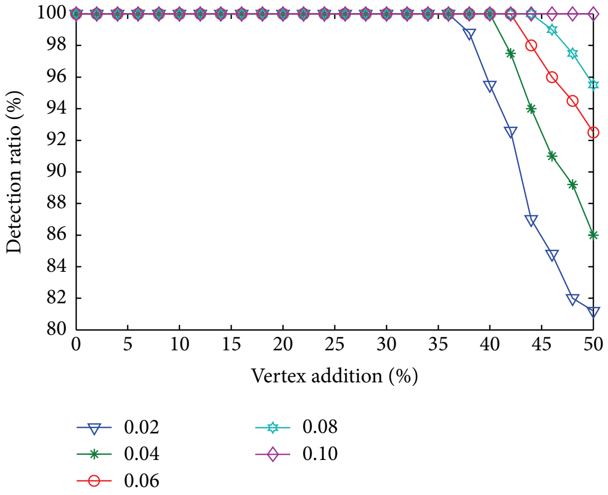

Figure 2: Resistance to interpolation.

4.3. Object Attacks

4.3.1. Reordering. Reordering of objects or their vertices is a special kind of attack; it only affects the artificially assigned identifiers of the objects and vertices. Figure 3 gives an example of reordering attack on vertices. However, neither the identifier of the object nor the one of the vertex is involved in the proposed scheme. Thus, the reordering attack could not affect the FVDR and the generated watermark. In this experiment, objects and their vertices in the watermarked map were reordered randomly for several times. For every reordering attack, the detection ratio is 100%. That is, the proposed scheme is robust to reordering attacks.

4.3.2. Deletion and Addition. The resistance for object deletion and addition relies on the number of objects in each group. Figures 5 and 6 present the experimental results of object deletion and addition, respectively. The results demonstrate that the more objects each group has, the more robustness we get. And when the number of objects in each group is larger than six, the algorithm's detection ratio is as high as 98% even facing a magnitude of 50% object deletion or addition. That is, the proposed scheme is robust to object deletion and addition.

4.4. Assorted Attacks. We also designed an experiment to verify the robustness of the proposed schema against assorted attacks. In this experiment, all aforementioned attacks were applied on the watermarked map. The range of magnitude for every applied attack is listed in Table 1. The experiment has been performed for ten times with different random magnitude taken from the respective range. The detection ratio of these executions is illustrated in Figure 7. In Figure 7, the average and lowest detection ratios are 91.73% and 88.79%, respectively, which demonstrates that the proposed schema is robust against assorted attack.

5. Conclusion

Focusing on polyline objects in vector maps, the Douglas-Peucker algorithm with relative threshold is applied for

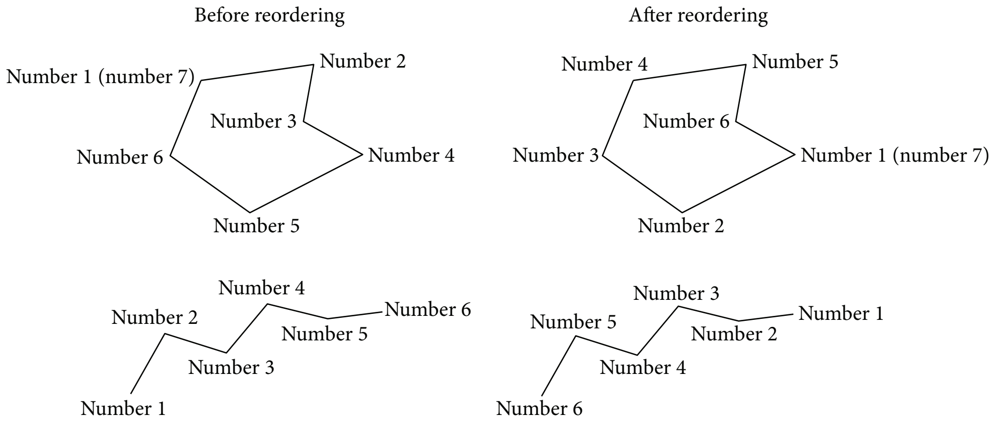

FIGURE 3: Examples of vertex reordering attack.

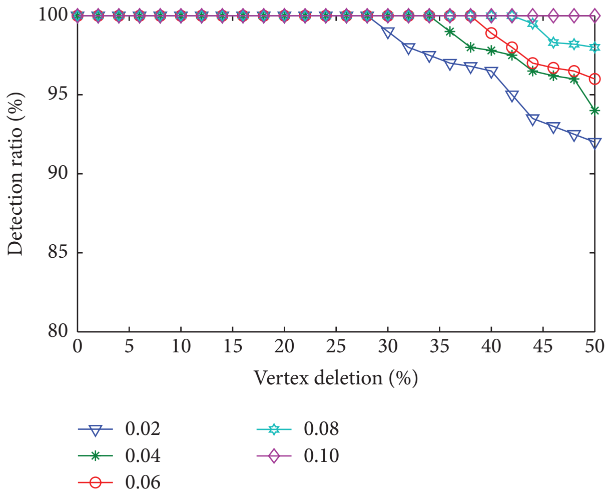

FIGURE 4: Resistance to simplification.

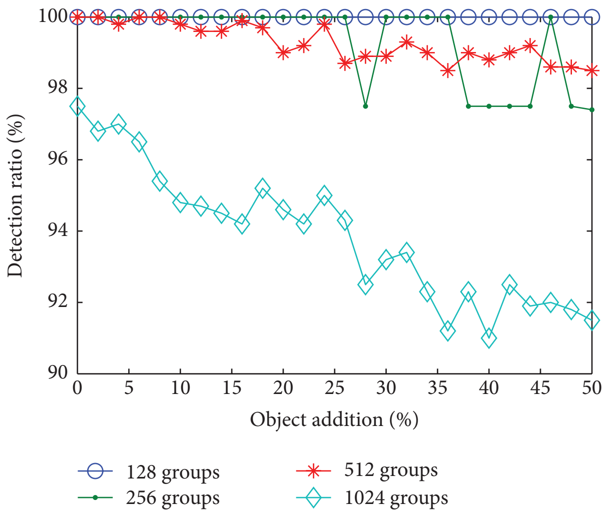

FIGURE 6: Resistance to object addition.

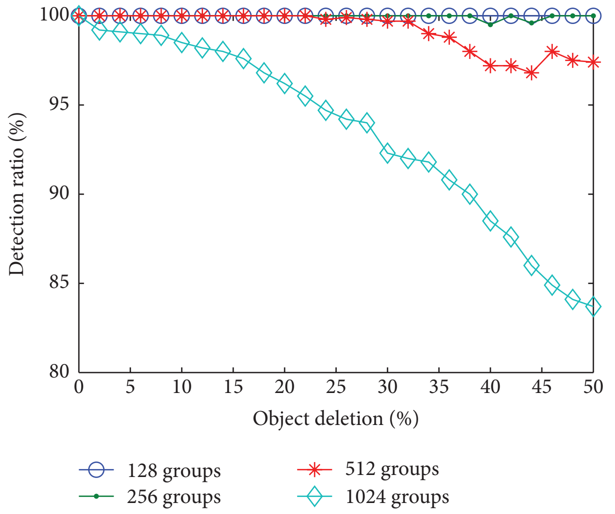

FIGURE 5: Resistance to object deletion.

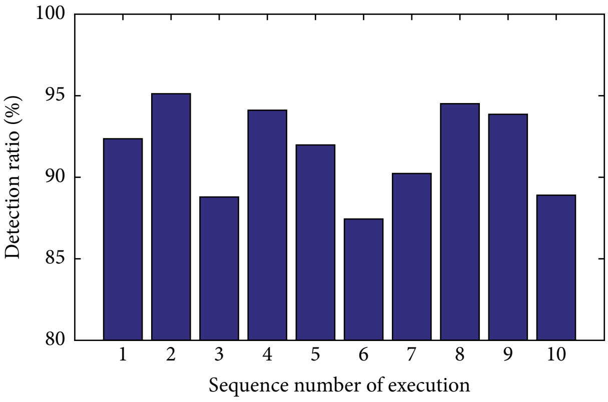

FIGURE 7: Resistance to assorted attack.

Table 1: Magnitudes of attacks.

Attack	Range of magnitude
Translation	−300%–300% on x-axis or y-axis
Rotation	0°–360°
Scaling	0–10
Object insertion	10%–20%
Object deletion	10%–20%
Noise distortion	0–1.2τ
Interpolation	10%–20%
Simplification	10%–20%

feature point extraction, and then cover data is defined based on the ratio between pairs of feature distances; finally, the objects are partitioned into groups according to the cover data and every bit of the watermark data is generated from objects of a corresponding group. The proposed scheme is robust to translation, rotation, scaling, simplification, interpolation, noise addition, object addition, deletion, and reordering attacks. In this paper, point and polygon objects are out of our consideration. Moreover, attacks like collusion attack that can only be resisted using protocols are not considered. These are planned for our future works.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The project was supported by the National Nature Science Foundation of China (nos. 61100019 and 61202033).

References

[1] C. J. Wang, Z. Y. Peng, Y. W. Peng, L. Yu, J. Z. Wang, and Q. Z. Zhao, “Watermarking geographical data on spatial topological relations,” *Multimedia Tools and Applications*, vol. 57, no. 1, pp. 67–89, 2012.

[2] S.-H. Lee and K.-R. Kwon, “Vector watermarking scheme for GIS vector map management,” *Multimedia Tools and Applications*, vol. 63, no. 3, pp. 757–790, 2013.

[3] B. Y. Wu and W. Wang, “Research on applied technology in blind and shape-preserving watermarking of vector map data using variable quantization step,” *Advanced Materials Research*, vol. 886, pp. 706–710, 2014.

[4] H. Yang, L. Min, and X. Hou, “An asymmetrical watermarking algorithm for vector map data,” *WIT Transactions on Information and Communication Technologies*, vol. 51, pp. 125–133, 2014.

[5] S. N. Neyman, I. N. P. Pradnyana, and B. Sitohang, “A new copyright protection for vector map using FFT-based watermarking,” *Telecommunication Computing Electronics and Control*, vol. 12, no. 2, pp. 367–378, 2014.

[6] Y. Chen, L. Zhang, and X. Ji, “A new robust watermarking algorithm for small vector data set,” *Applied Mechanics and Materials*, vol. 263–266, no. 1, pp. 2999–3004, 2013.

[7] J. Sun, G. Zhang, C. Men, Y. Wu, and X. Wang, “Lossless digital watermarking technology for vector maps,” *Applied Mechanics and Materials*, vol. 241–244, pp. 2773–2778, 2013.

[8] X. Wang, D. J. Huang, and Z. Y. Zhang, “A robust zero-watermarking algorithm for 2D vector digital maps,” in *Computer, Informatics, Cybernetics and Applications*, vol. 107 of *Lecture Notes in Electrical Engineering*, pp. 533–541, Springer, Amsterdam, The Netherlands, 2012.

[9] Q. Wen, T.-F. Sun, and S.-X. Wang, “Concept and application of zero-watermark,” *Acta Electronica Sinica*, vol. 31, no. 2, pp. 214–216, 2003.

[10] D. H. Douglas and T. K. Peucker, “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature,” *Canadian Cartographer*, vol. 10, pp. 112–122, 1973.

[11] T. Gökgöz, “Generalization of contours using deviation angles and error bands,” *The Cartographic Journal*, vol. 42, no. 2, pp. 145–156, 2005.

[12] Y.-C. Pu and I.-C. Jou, “Blind and robust watermarking for street-network vector maps,” *Information Technology Journal*, vol. 8, no. 7, pp. 982–989, 2009.

[13] X.-M. Niu, C.-Y. Shao, and X.-T. Wang, “GIS watermarking: hiding data in 2D vector maps,” *Studies in Computational Intelligence*, vol. 58, pp. 123–155, 2007.

[14] J. Kim, “Robust vector digital watermarking using angles and a random table,” *Advances in Information Sciences and Service Sciences*, vol. 2, no. 4, pp. 79–90, 2010.

[15] H. S. Hou and H. C. Andrews, “Cubic splines for image interpolation and digital filtering,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 6, pp. 508–517, 1978.

Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
International Journal of Distributed Sensor Networks
Advances in Civil Engineering
Journal of Control Science and Engineering
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration