

Storage Feature-Based Watermarking Algorithm with Coordinate Values Preservation for Vector Line Data

Qifei Zhou^{1,2,3}, Na Ren^{1,2,3}, Changqing Zhu^{1,2,3}, Deyu Tong^{1,2,3}

¹ Key Laboratory of Virtual Geographic Environment (Nanjing Normal University), Ministry of Education
Nanjing 210023, China
[e-mail: renna1026@163.com]

² State Key Laboratory Cultivation Base of Geographical Environment Evolution (Jiangsu Province)
Nanjing 210023, China

³ Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and
Application
Nanjing 210023, China

*Corresponding author: Na Ren

*Received December 11, 2017; revised February 11, 2018; accepted March 6, 2018;
published July 31, 2018*

Abstract

Most of current watermarking algorithms for GIS vector data embed copyright information by means of modifying the coordinate values, which will do harm to its quality and accuracy. To preserve the fidelity of vector line data and protect its copyright at the same time, a lossless watermarking algorithm is proposed based on storage feature in this paper. Firstly, the superiority of embedding watermark based on storage feature is demonstrated theoretically and technically. Then, the basic concepts and operations on storage feature have been defined including length and angle of the polyline feature. In the process of embedding watermark, the watermark information is embedded into directions of polyline feature by the quantitative mechanism, while the positions of embedding watermark are determined by the feature length. Hence, the watermark can be extracted by the same geometric features without original data or watermark. Finally, experiments have been conducted to show that coordinate values remain unchanged after embedding watermark. Moreover, experimental results are presented to illustrate the effectiveness of the method.

Keywords: Lossless watermarking, storage feature-based, vector line data, robustness

1. Introduction

With the rapid development of the information industry, data security is facing the unprecedented challenges. A lot of data security frameworks and techniques have been proposed [1-5]. As one of the advanced technologies, watermarking is the process of embedding and detecting the secret information in host data, which can be applied for copyright protection, user tracking, forgery prevention and many other purposes. Accordingly, the copyright protection of digital multimedia has attracted a lot of research focus due to the increasing security issues raised. In recent years, lots excellent researchers have worked in the field of watermarking for the multimedia data, image, video, etc. [6-8] In this paper, we focus on the digital watermarking for the vector line data that has become a valuable resource because of the high processing cost in the acquisition.

Many researchers have presented various methods for watermarking vector data. According to the watermark information embedding locations, the algorithms for vector data can be classified into spatial domain [9-23] and frequency domain [24-29]. Commonly, the algorithms based on spatial domain are summarized as follows: the algorithms that embed watermarks based on blocks division, such as regular grids[11, 13, 21] and MQUAD[17], the algorithms that embed watermarks in the feature vertices[20, 22], the algorithms that embed watermarks according to the constructing function based on the spatial topological relation[9, 16, 17], the algorithm that works on the newly added points[19], the algorithms that embed watermarks according to the different embedding rules[14, 18], and the coordinate mapping is used to embed watermark[23]. Existing frequency domain algorithms embed the watermarks into the frequency domains after discrete Fourier transform [24-26], discrete cosine transform [28, 29], and discrete wavelet transform [27].

Generally, the existing watermark schemes for the vector data mainly embed watermark information into the data by modifying the host data in either spatial domain or transform domain. Unfortunately, there is a very fatal disadvantage of those robust watermark algorithms that those methods will cause irreversible distortions to the vector data and thus destroy the accuracy of the data, which requires further improvement.

Recently, zero-watermark schemes have the potentiality to solve the problem of the existing watermark schemes [30-35]. These schemes can ensure the watermark robustness without introducing any data distortion. Instead, the watermark is not embedded into but built or picked up from the original vector data, which is completely different from the traditional watermark methods, by mapping the most two important characters the length ratio and angle to a dot picture which can also be saved in a vector format [31] or constructing the zero-watermark by the other feature information [32-35].

However, all the zero-watermark schemes need an authoritative third party and thus the data can be considered as watermarked despite the fact that there is no any additional watermark information embedded into the data. Therefore, they still cannot protect the copyrights of the vector data simultaneously and independently.

To address the above methods issues, a novel lossless watermarking algorithm based on the storage feature of polyline is proposed in this paper, which is benefit for high precision and copyright protection. The watermarking algorithm realizes the watermarking embedding and extraction full using of the character of polyline feature without modifying the coordinate value with high robustness.

The rest of this paper is organized as follows. Section 2 presents the important feature of vector line data which is the basis of the proposed algorithm. Section 3 describes the lossless watermarking algorithm. The performance of the proposed algorithm is developed in Section 4, before giving the final conclusions of the paper.

2. Storage Feature of Vector Line Data

Vector line data is composed of polyline features, and the polyline feature is composed of vertices. A vertex contains a coordinate like (x,y) and other auxiliary information such as elevation, descriptions. All vertices of a polyline feature are connected in turn to render the polyline feature. If the storage order of vertices in a polyline is reversed, there is no difference in the rendering results. There is the road data of a part of a city. The rendering result of the roads is shown in **Fig. 1 (a)** and after reversing the storage order of vertices of every road, which is shown in **Fig. 1 (b)**.



(a) Original storage order



(b) Reversed storage order

Fig. 1. The rendering results

As can be seen from the visual effects, the results in **Fig. 1 (a)** and **Fig. 1 (b)** are the same. However, the judgment from the visual effects alone is far from enough. Compared to **Fig. 1 (a)** and **Fig. 1 (b)**, the number of the point features with changed coordinates is zero. It illustrates that reversing the storage order of vertices in every polyline for vector line data has no effect on the count of polyline features and point features. Meanwhile, the accuracy of the data is kept effectively.

In order to facilitate the subsequent algorithm proposed and described, the following three definitions are given.

(1) The length of the polyline feature

It refers to the length of the polyline feature between the starting vertex and the end vertex, which are two adjacent vertices in the polyline data.

It is shown in **Fig. 2** where *Line* means the polyline feature, P_i represents the vertex in the polyline, n is the number of the points and *Len* is the length of polyline feature. (x_i, y_i) represents the coordinate of the vertex and the length of the polyline feature can be expressed by the formula (1).

$$Len = \sqrt{(x_1 - x_n)^2 + (y_1 - y_n)^2} \quad (1)$$

where, (x_1, y_1) and (x_n, y_n) are the starting and ending vertex of the polyline feature, respectively.

(2) The polyline angle

The polyline angle is the angle between the connection between the start and end points and the x-axis, which is called θ , $\theta \in [0, 360]$. As drawn in **Fig. 3**.

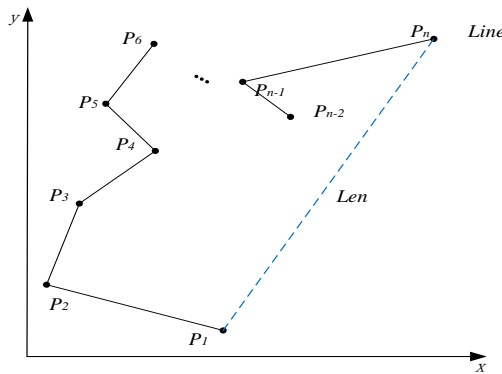


Fig. 2. The Length of polyline feature

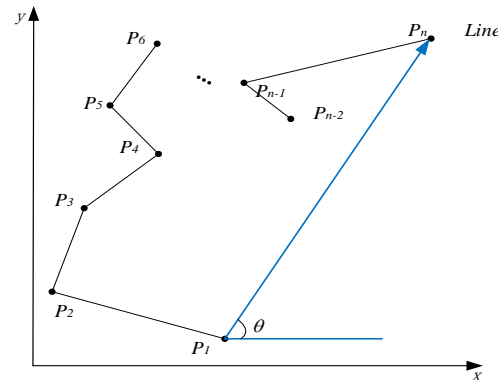


Fig. 3. The direction of polyline feature

The direction value θ of the polyline feature *Line* is calculated as follows.

$$\theta = \begin{cases} \arctan \frac{y_n - y_1}{x_n - x_1}, x_1 < x_n, y_1 \leq y_n \\ 90^\circ + \arctan \frac{x_1 - x_n}{y_n - y_1}, x_1 \geq x_n, y_1 > y_n \\ 180^\circ + \arctan \frac{y_n - y_1}{x_n - x_1}, x_1 > x_n, y_1 \geq y_n \\ 270^\circ + \arctan \frac{x_1 - x_n}{y_n - y_1}, x_1 \leq x_n, y_1 > y_n \end{cases} \quad (2)$$

(3) Reverse the polyline feature

It means reversing the storage order of vertices of the polyline feature. The polyline feature in **Fig. 3** is represented by an ordered set $l, l = \{(x_i, y_i) | i = 1, 2, \dots, n\}$. Reverse l and get l' whose ordered set is stored as $l'_i = reverse(l) = \{(x_i, y_i) | i = n, n-1, \dots, 1\}$.

3. The lossless watermarking algorithm

As mentioned above, we propose a novel lossless watermarking scheme as shown in **Fig. 4**. In the process of watermarking generation, a secret key is used to generate a meaningless watermarking information. In particular, the watermark embedding bit is determined by the length of the polyline feature and the embedding form is achieved by reversing the polyline feature or not judged by whether the embedded watermark information is consistent with the quantization value of the direction. Therefore, the key idea of the embedding rule is to reverse the storage order of all the elements, which has no effect on the data accuracy and rendering results in the terminal. Therefore, we can see that the proposed method doesn't modify the coordinate value in the process of embedding watermark.

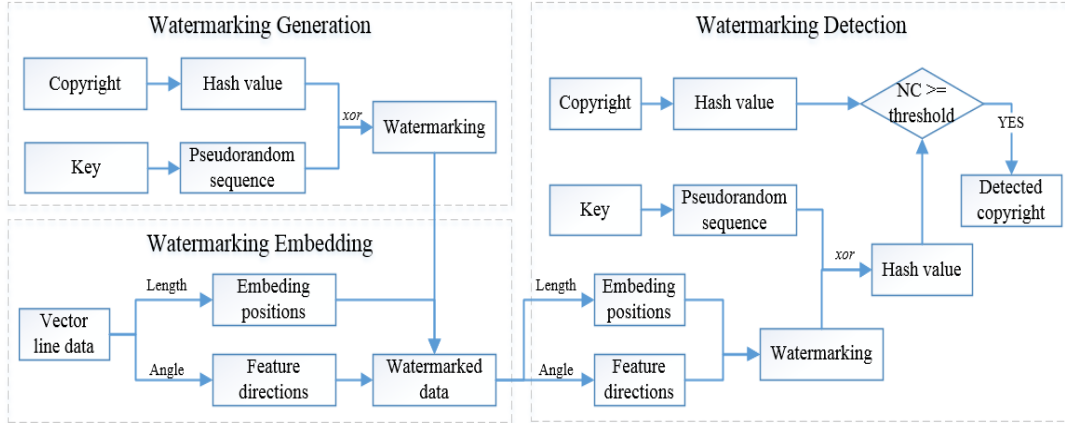


Fig. 4. The modules of the proposed algorithm

3.1 Watermarking generation

In this paper, the meaningless watermarking information is chosen. The generation steps of the watermarking information are as follows:

- (1) Assume that the copyright information is C , and use C to generate a hash value H , $H = \{h_i \mid i = 1, 2, \dots, N\}$ with the length N , and $h_i = \{0, 1\}$.
- (2) Use a key to generate a pseudorandom sequence R with the length N .
- (3) Do xor operation for the index bits of H and R , to get the scrambling watermarking information, as described,

$$W = H \wedge R \quad (3)$$

where, \wedge is xor operation, $W = \{w_i \mid i = 1, 2, \dots, N\}$, and $w_i = \{0, 1\}$.

3.2 Watermarking embedding

The proposed algorithm is not based on the sequential embedding mode, but the order of the watermark is determined according to the index value of each vector line feature. Moreover, the watermark embedding rules are according to the watermark bit obtained by the watermark information embedded in the vector data.

For more clarity, the embedding process is described in more details in the following.

- (1) Get the set L of all the polyline features from the original vector line data. $L = \{l_i \mid i = 1, 2, \dots, m\}$, where, m is the number of polyline features.

- (2) For each line feature, the watermark embedding bit X_i is generated by the length of the line. It can be calculated by the formula (4).

$$X_i = \text{round}(|\sin(\text{Len}_i)| \times (N - 1) + 1) \quad (4)$$

where, *round* is the rounding operation, and $X_i \in [1, N]$. The set of the watermark embedding bits of all the polyline features is X , $X = \{X_i | i = 1, 2, \dots, m\}$. Therefore, the watermark that will be embedded in each linear feature is W_{X_i} .

(3) For each line element, its direction is quantified by the following expression (5).

$$d_i = \begin{cases} 1, \sin \theta_i \geq 0 \\ 0, \sin \theta_i < 0 \end{cases} \quad (5)$$

The direction quantization value of all the polyline features is $D = \{d_i | i = 1, 2, \dots, m\}$.

(4) To realize embedding the watermark information, the embedding form is judged by whether the embedded watermark information is consistent with the quantization value of the direction. If they are the same, the storage order of the line remains unchanged, otherwise, reverses the polyline feature. It can be computed as follows:

$$l'_i = \begin{cases} l_i, & W_{X_i} = d_i \\ \text{reverse}(l_i), & W_{X_i} \neq d_i \end{cases} \quad (6)$$

The study of this scenario shows that the all the watermarked value of the line feature remains the same as the original one although the order of storage may be changed.

Once watermarked all polyline features are available by using the above steps, the last step of the watermarking process is the creation of the certain data format file.

3.5 Watermarking detection

The original data is not needed in the watermark detection processes, so this is a blind watermarking approach. The steps of watermark detection are illustrated as follows.

(1) Get the set L' of all the polyline feature $L' = \{l'_i | i = 1, 2, \dots, m'\}$ from the watermarked vector line data, where m' is the number of the polyline features.

(2) Calculate the watermarking embedded bit of each polyline feature $X' = \{X'_i | i = 1, 2, \dots, m'\}$ using the formula (3).

(3) Calculate the direction quantization value of each polyline feature $D' = \{d'_i | i = 1, 2, \dots, m'\}$ by the formula (4).

(4) For each line element, the watermarking information is the direction quantization value, so it can be obtained by formula (7).

$$W'_{x_i} = d'_i \quad (7)$$

The watermarking information in X'_i position may be detected more than one time, so it obeys the principle of majority. Therefore, the watermarking information detected in maximum times is the final watermarking information.

(5) The detected watermark W' and pseudorandom sequence R do *xor* operation can get a result, as shown,

$$H' = W' \wedge R \quad (8)$$

where, $H' = \{h'_i | i = 1, 2, \dots, N\}$ and $h'_i = \{0, 1\}$.

(6) NC can be calculated with H' and the original hash value H , as calculated,

$$NC = 1 - \left(\sum_{i=1}^N (h_i \wedge h'_i) \right) / N \quad (9)$$

where $NC \in [0, 1]$. If the NC value exceeds the threshold value, the copyright will be extracted. Otherwise, there is no watermarking.

4. Results and analysis

In order to verify the performance of the proposed algorithm, we use three *ESRI* format 2D vector maps “road”, “river” and “railway” as the original maps to test the performance of the proposed scheme. **Table 1** lists some basic properties of the three vector maps, including data format, the feature type, the number of vertices, the number of features, and the scale. **Fig. 5(a), (b)** and **(c)** show the original vector data. The watermark information in this test is “Copyright Protection”, and the length of the watermark information is 128. The detection threshold is set as 0.65.

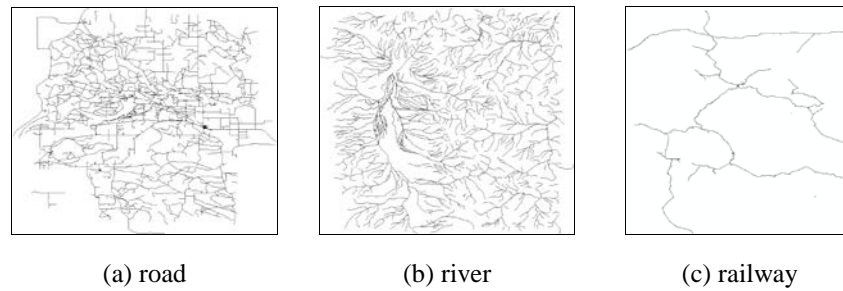


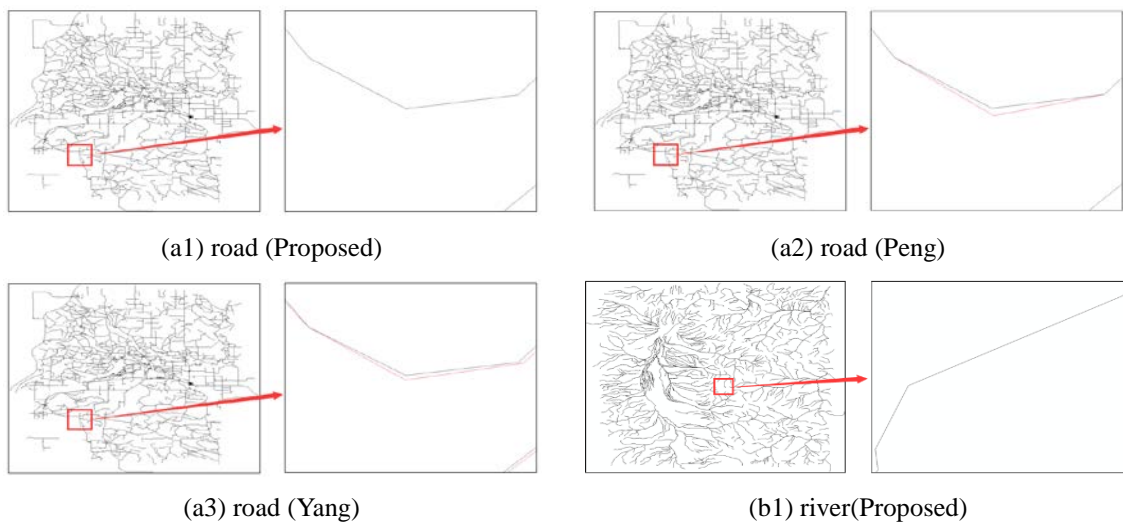
Fig. 5. Original vector maps

Table 1. Properties of original vector maps

| Vector maps | Format | Feature type | Vertices | Features | Scale |
|-------------|-----------|--------------|----------|----------|----------|
| Road | shapefile | polyline | 23627 | 1478 | 1:200000 |
| River | shapefile | polyline | 36060 | 1695 | 1:250000 |
| Railway | shapefile | polyline | 23101 | 1617 | 1:200000 |

4.1 Imperceptibility

Our algorithm has the higher fidelity than the traditional method under the premise of high robustness. Therefore, we evaluated the imperceptibility of our scheme and the conventional blind and robust geospatial watermarking schemes. An experiment is carried out to investigate the imperceptibility after embedding the watermark without degrading the quality. The conventional schemes of Peng[20], and Yang[23] were tested using the same watermark length as our scheme. The scheme of Yang is a very typical traditional algorithm. Fig. 6 presents the watermarked data which is overlapped the original data, then the corresponding part is enlarged. The watermarked data is black and the original data is red. Table 2 shows the extracted watermark information and the NC value.



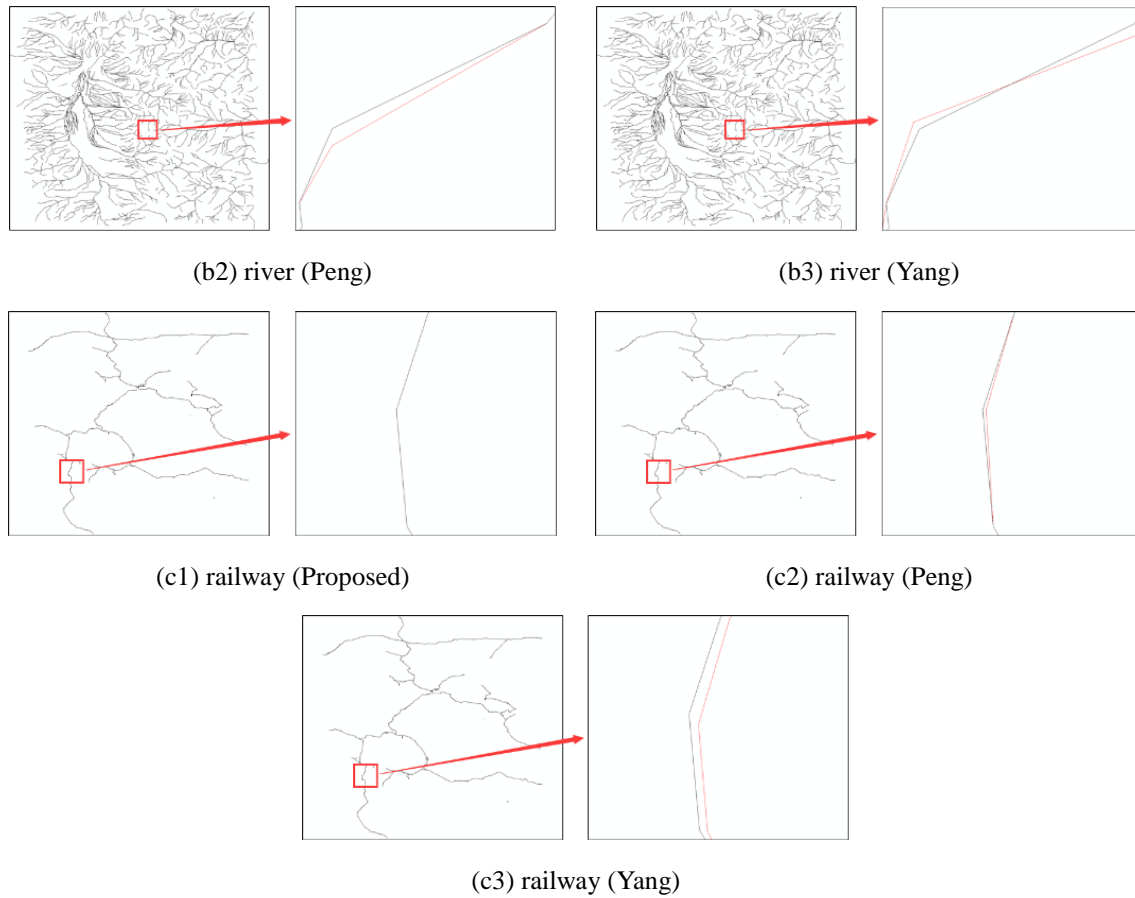


Fig. 6. Overlap original data and watermarked data and then zoom in the part

Table 2. The extracted watermark and the *NC* value

| Vector maps | The extracted watermark | <i>NC</i> value | | |
|-------------|-------------------------|-----------------|------|------|
| | | Proposed | Peng | Yang |
| Road | Copyright Protection | 1 | 1 | 1 |
| River | Copyright Protection | 1 | 1 | 1 |
| Railway | Copyright Protection | 1 | 1 | 1 |

It can be seen from **Fig. 6** that the watermarked data which used the proposed algorithm is same as the original data. However, the watermarked data which used traditional algorithms is easy to find the differences from the original data. **Table 2** shows the extracted watermark with the three *NC* value of 1. It can be said that the watermarked quality is acceptable and the scheme has a good imperceptibility.

4.2 Analysis the destruction of the coordinate point

In the first test case, the invisibility of the scheme is demonstrated. For evaluating the quality of the embedded vector data, we compared the original vector map and the watermarked one. The change rate of coordinate value r_c is calculated,

$$r_c = \frac{\text{num}\left(\sqrt{(x'_i - x_i)^2 + (y'_i - y_i)^2} \neq 0\right)}{N_l} \quad (10)$$

Where, N_l denotes the total number of vertices in the vector data and $\text{num}(\)$ represents the number of changed coordinate values. The experimental results are listed in [Table 3](#).

Table 3. Results of the distortion

| Distortion | | Road | River | Railway |
|---------------------|----------|-------|-------|---------|
| Numbers of features | | 1478 | 1695 | 1617 |
| Numbers of vertices | | 23627 | 36060 | 23101 |
| r_c | Proposed | 0 | 0 | 0 |
| | Peng | 0.60 | 0.31 | 2.01 |
| | Yang | 96.85 | 98.82 | 94.07 |

Based on the experimental results illustrated in [Table 3](#), it can be seen that there is no change in the coordinates of the vector data after embedding the watermark. However, there are changes for the other two algorithms. Besides, the changes of Peng's algorithm are only the feature vertices, which is much less than the changes of Yang's algorithm. The results show that the traditional watermarking algorithm often embedded watermark in each point. That leads to the result that the majority coordinates have been changed.

That is to say, for the proposed algorithm, the operation of the embedding watermark has no effect on the accuracy of the vector data. The watermarked data does not change the coordinate errors of vertices and does not lead to distortion of the shape or topology of map objects. This also verifies our analysis above that the proposed algorithm is lossless to the watermarked vector data.

4.3 Analysis of translation attacks

There are two translation attacks, global translation and local translation, respectively. Global translation is measured by the translation distance, and the ratio of the polylines is to measure the local translation.

The data in X-axis and Y-axis respectively are translated by a random relative translational distance to test the global translation attack. Another one is the data randomly translate different proportions of polyline elements to test the local translation attack.

Table 4 and **Table 5** show the translation experimental results. The degree of global translation is measured by the ratio of the maximum width relative to the data, and the degree of local translation is measured by the ratio of translational polyline features.

Table 4. Results of the global translation attacks

| Distance ratio(%) / NC | Road | River | Railway |
|---------------------------|------|-------|---------|
| 10 | 1 | 1 | 1 |
| 20 | 1 | 1 | 1 |
| 30 | 1 | 1 | 1 |
| 40 | 1 | 1 | 1 |
| 50 | 1 | 1 | 1 |
| 60 | 1 | 1 | 1 |
| 70 | 1 | 1 | 1 |
| 80 | 1 | 1 | 1 |
| 90 | 1 | 1 | 1 |
| 100 | 1 | 1 | 1 |

Table 5. Results of the local translation attacks

| Translating ratio(%) / NC | Road | River | Railway |
|------------------------------|------|-------|---------|
| 10 | 1 | 1 | 1 |
| 20 | 1 | 1 | 1 |
| 30 | 1 | 1 | 1 |
| 40 | 1 | 1 | 1 |
| 50 | 1 | 1 | 1 |
| 60 | 1 | 1 | 1 |
| 70 | 1 | 1 | 1 |
| 80 | 1 | 1 | 1 |
| 90 | 1 | 1 | 1 |
| 100 | 1 | 1 | 1 |

As shown in **Table 4** and **Table 5**, no matter the translation mode and how much translation, the *NC* value is always equal to 1. The reason is that the polyline direction and length will never change with the translation. That is to say, the translation attack has no effect on the proposed scheme.

4.4 Analysis of rotating attacks

We consider here global rotation of the whole vector data. Considering that if the rotation angle is too large, the order spatial relationship of the data will change and the data will lose the proper use value. So we rotated the watermarked data around the origin (0,0) by a random angle ranging from -40 to 40 degree every 5 degrees. [Table 6](#) and [Fig. 7](#) display the experimental results.

Table 6. Results of rotating attacks

| Degree of rotation(°) /NC | Road | River | Railway |
|------------------------------|-------|-------|---------|
| -40 | 0.953 | 0.906 | 0.898 |
| -35 | 0.969 | 0.93 | 0.938 |
| -30 | 0.977 | 0.938 | 0.961 |
| -25 | 0.977 | 0.953 | 0.984 |
| -20 | 0.977 | 0.984 | 1 |
| -15 | 0.977 | 0.992 | 1 |
| -10 | 0.984 | 1 | 1 |
| -5 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 |
| 15 | 0.977 | 0.992 | 0.984 |
| 20 | 0.969 | 0.992 | 0.922 |
| 25 | 0.953 | 0.992 | 0.883 |
| 30 | 0.922 | 0.977 | 0.836 |
| 35 | 0.898 | 0.969 | 0.758 |
| 40 | 0.867 | 0.961 | 0.695 |

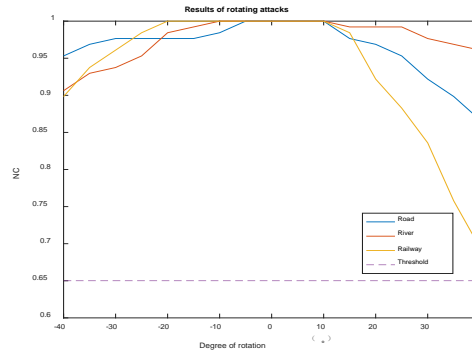


Fig. 7. Results of rotating attacks

It can be seen from **Table 6** and **Fig. 7** that the *NC* value increases gradually from -40° to 0° and decreases gradually from 0° to 40° after the rotation of the origin of different degrees of rotation. Moreover, the minimum *NC* value is higher than the threshold. Therefore, the proposed algorithm is resistance to a normal range of rotation.

4.5 Analysis of compression attack

Compression the polylines approximate the polyline curves by removing some vertices in the objects while preserving the geometrical property. Douglas–Peucker algorithm is a kind of compression algorithm used for the purpose of this test. Different quality factors were applied to the watermarked vector data. **Table 7** summarizes the results.

It can be clearly seen that all the *NC* values are always equal to 1 even if the compression ratio of up to 80%. This is because the compression algorithm does not remove the beginning and end points of the polyline elements, and the calculation of watermark embedded bit only uses the first and last point in this algorithm. This demonstrates that the proposed algorithm is completely resistant to Douglas–Peucker compression.

Table 7. Results of compression attacks

| Compression ratio(%) / <i>NC</i> | Road | River | Railway |
|----------------------------------|------|-------|---------|
| 10 | 1 | 1 | 1 |
| 20 | 1 | 1 | 1 |
| 30 | 1 | 1 | 1 |
| 40 | 1 | 1 | 1 |
| 50 | 1 | 1 | 1 |
| 60 | 1 | 1 | 1 |
| 70 | 1 | 1 | 1 |
| 80 | 1 | 1 | 1 |

4.6 Analysis of data adding

Any unauthorized user can modify polylines by adding some vertices on the basis of an interpolation or a random position or can modify a map by adding some vertices to them. The new adding polylines can lead to a confusion or a noise to the extracted watermark information. In the adding experiment, we add approximately from 10% to 200% vertices into randomly selected polylines. The experimental results for data adding are shown in [Fig. 8](#) and [Table 8](#).

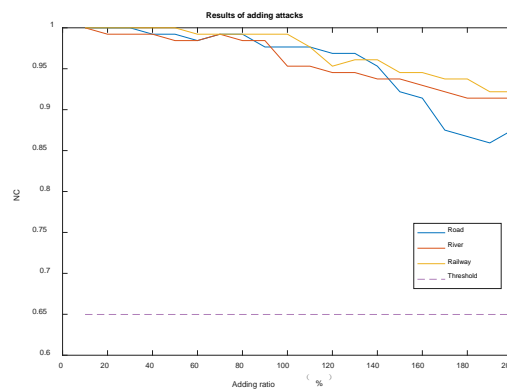


Fig. 8. Results of adding attacks

Table 8. Results of adding attacks

| Adding ratio(%)/NC | Road | River | Railway |
|--------------------|-------|-------|---------|
| 10 | 1 | 1 | 1 |
| 20 | 1 | 0.992 | 1 |
| 30 | 1 | 0.992 | 1 |
| 40 | 0.992 | 0.992 | 1 |
| 50 | 0.992 | 0.984 | 1 |
| 60 | 0.984 | 0.984 | 0.992 |
| 70 | 0.992 | 0.992 | 0.992 |
| 80 | 0.992 | 0.984 | 0.992 |
| 90 | 0.977 | 0.984 | 0.992 |
| 100 | 0.977 | 0.953 | 0.992 |
| 110 | 0.977 | 0.953 | 0.977 |
| 120 | 0.969 | 0.945 | 0.953 |
| 130 | 0.969 | 0.945 | 0.961 |
| 140 | 0.953 | 0.938 | 0.961 |
| 150 | 0.922 | 0.938 | 0.945 |
| 160 | 0.914 | 0.93 | 0.945 |

| | | | |
|-----|-------|-------|-------|
| 170 | 0.875 | 0.922 | 0.938 |
| 180 | 0.867 | 0.914 | 0.938 |
| 190 | 0.859 | 0.914 | 0.922 |
| 200 | 0.875 | 0.914 | 0.922 |

As can be seen from [Table 8](#) and [Fig. 8](#), the *NC* value never falls below the detection threshold, and decreases slowly as the proportion of the increased polyline element becomes larger. The *NC* value is still very high, even in a worst-case scenario, the number of added polyline elements is twice the original polyline feature. Therefore, the algorithm is resistance to adding attacks.

4.7 Analysis of polyline cropping

Any unauthorized user can crop any of the regions of or any data in a vector map. We crop a region occupying an area of approximately from 10% to 90% of the vertices that are randomly selected in the polylines. The experimental results for polylines cropping are shown in [Table 9](#) and [Fig. 9](#).

Table 9. Results of cropping attacks

| Cropping ratio(%)/ <i>NC</i> | Road | River | Railway |
|------------------------------|-------|-------|---------|
| 10 | 1 | 1 | 1 |
| 20 | 1 | 1 | 1 |
| 30 | 1 | 1 | 1 |
| 40 | 1 | 1 | 1 |
| 50 | 1 | 1 | 1 |
| 60 | 0.995 | 0.992 | 1 |
| 70 | 0.992 | 0.969 | 0.984 |
| 80 | 0.945 | 0.942 | 0.922 |
| 90 | 0.844 | 0.836 | 0.854 |

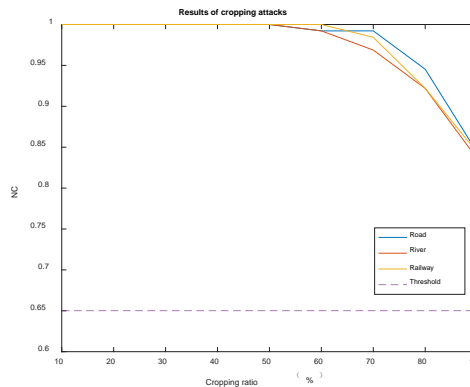


Fig. 9. Results of cropping attacks

Results in [Table 9](#) and [Fig. 9](#) demonstrate that deleting any area of the vector data will remove any watermark bits present. However, the watermark can be extracted in the remaining polylines. The results also show that as long as the proportion of the cropped line elements becomes larger, the *NC* value begins to drop until the line elements of halves and above are deleted. This is because the watermark is spread across the entire vector data, it is very likely to survive deleting attacks, unless a very large portion of the watermarked data is deleted, in which case, its original commercial value would be lost. These results show that the proposed scheme is robust to data cropping.

4.8 Analysis of format conversion

Because of the existence of various GIS platforms, different platforms have different data formats and even on a platform, there are a variety of data formats. So the format conversion is inevitable in the use of vector geography data to meet the format requirements of the target GIS platform. At the same time, the format conversion also makes a high demand for the robustness of the digital watermarking. The following tests the line data of the watermarked “*shapefile*” format for various format conversion, as shown in [Table 10](#).

Table 10. Analysis of format conversion

| Target format / <i>NC</i> | Road | River | Railway |
|---------------------------|------|-------|---------|
| AutoCAD DXF | 1 | 1 | 1 |
| AutoCAD DWG | 1 | 1 | 1 |
| ESRI File Geodatabase | 1 | 1 | 1 |
| ESRI Personal Geodatabase | 1 | 1 | 1 |
| GML | 1 | 1 | 1 |
| KML | 1 | 1 | 1 |

From the results of [Table 10](#), we can see that the *NC* value is always 1 after the format conversion. The reason for this is that the polyline in the format conversion usually does not change the order of the points in the polyline feature. Therefore, the proposed algorithm is resistance to the format conversion.

In summary, in order to verify the robustness to geometrical attacks, we performed translation, rotation, compression, addition, cropping and format conversion on the watermarked vector data respectively. In these experiments, the watermark can be extracted successfully. The scheme is not dependent on the coordinates directly, thus translation, compression cannot affect the watermark detection. Furthermore, the scheme is robust to rotation, addition and cropping data.

5. Conclusion

In this paper, a lossless watermarking algorithm based on storage feature of GIS vector line data is investigated. Unlike other schemes, the watermark embedding just changes the storage order of the data, which makes our algorithm useful for avoiding any influence on coordinate value. In addition, watermark embedding is realized by quantization of polyline angles and reversing the polyline feature, which helps the algorithm resisting geometry attacks. Experiments have been performed to validate the robustness against different types of attack. Overall, the proposed algorithm proves its effectiveness in security and copyright protection of polyline data. Besides, the scheme can be applied to other kinds of vector data like CAD drawing, which have similar data structures. In the future, we will test more types of vector data, which may have different characteristics.

Acknowledgement

Funding: This work was supported by a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions; Six Talent Peaks Project in Jiangsu Province.

Thanks for the data provider, Yangtze River Delta Science Data Center, National Earth System Science Data Sharing Infrastructure, National Science & Technology Infrastructure of China.

References

- [1] X. Niu, C. Shao, and X. Wang, "A survey of digital vector map watermarking," *International Journal of Innovative Computing Information & Control Ijicic*, vol. 2, no. 6, Dec. 2006. [Article \(CrossRef Link\)](#).
- [2] L. Zhou, D. Wu, B. Zheng, and M. Guizani, "Joint physical-application layer security for wireless multimedia delivery," *IEEE Communications Magazine*, vol. 52, no. 3, pp. 66-72, Mar. 2014. [Article \(CrossRef Link\)](#).
- [3] L. Zhou, H. C. Chao, "Multimedia traffic security architecture for the internet of things," *Network IEEE*, vol. 25, no. 3, pp. 35-40, May. 2011. [Article \(CrossRef Link\)](#).
- [4] J. Shen, D. Liu, J. Shen, Q. Liu, and X. Sun, "A secure cloud-assisted urban data sharing framework for ubiquitous-cities," *Pervasive & Mobile Computing*, vol. 41, pp. 219-230, Oct. 2017. [Article \(CrossRef Link\)](#).
- [5] H. Yan, L. Zhang, and W. Yang, "A normalization-based watermarking scheme for 2D vector map data," *Earth Science Informatics*, vol. 10, no. 4, pp. 471-481, Dec. 2017. [Article \(CrossRef Link\)](#).

- [6] Z. Bahrami, F. A. Tab, "A new robust video watermarking algorithm based on SURF features and block classification," *Multimedia Tools & Applications*, vol. 77, no. 1, pp. 327-345, Jan. 2017. [Article \(CrossRef Link\)](#).
- [7] Y. Erfani, R. Pichevar, and J. Rouat, "Audio watermarking using spikegram and a two-dictionary approach," *IEEE Transactions on Information Forensics & Security*, vol. 12, no. 4, pp. 840-852, Apr. 2017. [Article \(CrossRef Link\)](#).
- [8] E. Etemad, S. Samavi, S. M. R. Soroushmehr, N. Karimi, M. Etemad, S. Shirani, and K. Najarian, "Robust image watermarking scheme using bit-plane of hadamard coefficients," *Multimedia Tools & Applications*, vol. 77, no. 2, pp. 2033-2055, Jan. 2017. [Article \(CrossRef Link\)](#).
- [9] C. Wang, Z. Peng, Y. Peng, L. Yu, J. Wang, and Q. Zhao, "Watermarking geographical data on spatial topological relations," *Multimedia Tools & Applications*, vol. 57, no. 1, pp. 67-89, Mar. 2012. [Article \(CrossRef Link\)](#).
- [10] R. Ohbuchi, H. Ueda, and S. Endoh, "Robust watermarking of vector digital maps," in *Proc. of IEEE International Conference on Multimedia and Expo*, pp. 577-580, Aug 26-29, 2002. [Article \(CrossRef Link\)](#).
- [11] R. Ohbuchi, H. Ueda, and S. Endoh, "Watermarking 2D vector maps in the mesh-spectral domain," *Shape Modeling International*, pp. 216-225, May 12-15, 2003. [Article \(CrossRef Link\)](#).
- [12] P. Jia, Y. Chen, M. A. Jinsong, and D. Zhu, "Digital watermark-based security technology for geo-spatial graphics data," *Chinese Geographical Science*, vol. 16, no. 3, pp. 276-281, Aug. 2006. [Article \(CrossRef Link\)](#).
- [13] D. Zhang, D. Qian, and P. Han, "A new attributes-priority matching watermarking algorithm satisfying topological conformance for vector map," in *Proc. of International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 469-472, Nov 26-28, 2007. [Article \(CrossRef Link\)](#).
- [14] S. Deng, L. Liang, S. Deng, and S. Che, "Research on a digital watermarking algorithm suitable to vector map," in *Proc. of IEEE International Conference on Automation and Logistics*, pp. 1236-1240, Aug 18-21, 2007. [Article \(CrossRef Link\)](#).
- [15] Y. Dakroury, I. A. El-Ghafar, and A. Tammam, "Protecting GIS data using cryptography and digital watermarking," *International Journal of Computer Science & Network Security*, vol. 10, no. 1, Jan. 2010. [Article \(CrossRef Link\)](#).
- [16] M. Z. Lai, G. Y. Zhang, "Watermarking scheme of vector maps based on spatial relationship," in *Proc. of International Conference on Computer Science and Software Engineering*, Oct 18-19, 2014. [Article \(CrossRef Link\)](#).
- [17] E. Horness, N. Nikolaidis, and I. Pitas, "Blind city maps watermarking utilizing road width information," in *Proc. of Signal Processing Conference*, pp. 2291-2295, Sept 3-7, 2007. [Article \(CrossRef Link\)](#).

- [18] H. Yan, J. Li, and H. Wen, "A key points-based blind watermarking approach for vector geo-spatial data," *Computers Environment & Urban Systems*, vol. 35, no. 6, pp. 485-492, 2011. [Article \(CrossRef Link\)](#).
- [19] K. T. Park, K. I. Kim, H. I. Kang, and S. S. Han, "Digital geographical map watermarking using polyline interpolation," in *Proc. of IEEE Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing*, pp. 58-65. Dec 16, 2002. [Article \(CrossRef Link\)](#).
- [20] Z. Peng, M. Yue, X. Wu, and Y. Peng, "Blind watermarking scheme for polylines in vector geo-spatial data," *Multimedia Tools & Applications*, vol. 74, no. 24, pp. 11721-11739, Dec. 2015. [Article \(CrossRef Link\)](#).
- [21] M. Sakamoto, Y. Matsuura, and Y. Takashima, "A scheme of digital watermarking for geographical map data," in *Proc. of Symposium on cryptography and information security*, Jan 26, 2000. [Article \(CrossRef Link\)](#).
- [22] C. Shao, H. Wang, X. Niu, and X. Wang, "A shape-preserving method for watermarking 2D vector maps based on statistic detection," *IEICE TRANSACTIONS on Information and Systems*, vol. 89, no. 3, pp. 1290-1293, Mar. 2006. [Article \(CrossRef Link\)](#).
- [23] C. S. Yang, C. Q. Zhu, and D. X. Tao, "A blind watermarking algorithm for vector geo-spatial data based on coordinate mapping," *Journal of Image and Graphics*, vol. 15, no. 4, pp. 684-688, Apr. 2010. [Article \(CrossRef Link\)](#).
- [24] V. R. Doncel, N. Nikolaidis, and I. Pitas, "An optimal detector structure for the Fourier descriptors domain watermarking of 2D vector graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 5, pp. 851-863, Jul. 2007. [Article \(CrossRef Link\)](#).
- [25] D. H. Xu, Q. S. Wang, "The study of watermarking algorithm for vector geospatial data based on the phase of DFT," in *Proc. of Wireless Communications, Networking and Information Security (WCNIS)*, pp. 625-629. Jun 25-27, 2010. [Article \(CrossRef Link\)](#).
- [26] I. Kitamura, S. Kanai, and T. Kishinami, "Copyright protection of vector map using digital watermarking method based on discrete Fourier transform," in *Proc. of Geoscience and Remote Sensing Symposium*, pp. 1191-1193. Jul 9-13, 2001. [Article \(CrossRef Link\)](#).
- [27] C. Q. Zhu, C. S. Yang, and Q. S. Wang, "A watermarking algorithm for vector geo-spatial data based on integer wavelet transform," *Int Arch Photogramm Remote Sens Spat Inf Sci*, vol. 37, no. B4, pp. 15-18, 2008. [Article \(CrossRef Link\)](#).
- [28] X. Wang, D. J. Huang, and Z. Y. Zhang, "A DCT-based blind watermarking algorithm for vector digital maps," *Advanced Materials Research*, pp. 1053-1058. Jan 1, 2011. [Article \(CrossRef Link\)](#).
- [29] B. Liang, J. Rong, and C. Wang, "A vector maps watermarking algorithm based on dct domain," *ISPRS Congr*, vol. 38, no. 3, 2010. [Article \(CrossRef Link\)](#).

- [30] 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, Feb. 2003. [Article \(CrossRef Link\)](#).
- [31] H. Zhao, S. Du, and D. Zhang, "Zero-Watermark scheme for 2d vector drawings based on mapping," *Information Science and Cloud Computing Companion (ISCC-C)*, pp. 601-605. Dec 7-8, 2013. [Article \(CrossRef Link\)](#).
- [32] W. Xun, H. Ding-Jun, and Z. Zhi-Yong, "A robust zero-watermarking algorithm for 2D vector digital maps," *Computer, Informatics, Cybernetics and Applications*, pp. 533-541: Springer, 2012. [Article \(CrossRef Link\)](#).
- [33] Y. Peng, M. Yue, "A zero-watermarking scheme for vector map based on feature vertex distance ratio," *Journal of Electrical and Computer Engineering*, vol. 2015, pp. 35, Jun. 2015. [Article \(CrossRef Link\)](#).
- [34] A. Li, B. Lin, Y. Chen, and G. Lü, "Study on copyright authentication of GIS vector data based on Zero-watermarking," *The international archives of the photogrammetry, remote sensing and spatial information sciences*, vol. 37, no. B4, pp. 1783-1786, 2008. [Article \(CrossRef Link\)](#).
- [35] Y. Liu, F. Yang, K. Gao, W. Dong, and J. Song, "A zero-watermarking scheme with embedding timestamp in vector maps for Big Data computing," *Cluster Computing*, vol. 20, no. 4, pp. 3667-3675, Dec. 2017. [Article \(CrossRef Link\)](#).



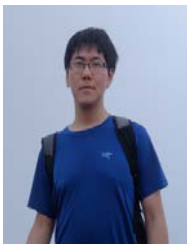
Qifei Zhou is currently pursuing a Ph.D. in Nanjing Normal University. He received the B.E. degree in Nanjing Normal University, Nanjing, China, in 2015. His research interests include digital watermarking and transparent encryption.



Na Ren is an associate professor in Nanjing Normal University, Nanjing, China. She received the Ph.D. from Nanjing Normal University in 2011. Her research interests include digital watermarking, data security and GIS.



Changqing Zhu is a professor in Nanjing Normal University, Nanjing, China. He received the Ph.D. from PLA Information Engineering University in 1997. His research interests include digital watermarking, data security and GIS.



Deyu Tong is now a Ph.D. candidate in Nanjing Normal University. He received the B.E. degree from Nanjing University of Information Science & Technology, Nanjing, China, in 2012. His research interests include digital watermarking, Commutative Encryption and Watermarking.