A Fragile Watermarking Scheme of Anti-deleting Features for 2D Vector Map

Guoyin Zhang, Qingan Da, Liguo Zhang, Jianguo Sun(✉), Qilong Han, Liang Kou, and WenShan Wang

Harbin Engineering University, Harbin, China
sunjianguo@hrbeu.edu.cn

Abstract. This paper proposes a fragile watermarking scheme of anti-deleting features for 2D vector map. The features in vector map are first divided into disjoint groups to ensure the accuracy of tamper localization. In order to locate the batch features deletion attack, we design a feature group correlation technique based on vertex insertion. And a watermark is generated by folding the hash results of the differences of the log-radiuses, which is robust to resist rotation, uniform scaling and translation (RST) operations. And we embed the watermark with a RST invariant watermarking method. Two datasets are constructed for experimentation and the results compared with previous methods indicate that the proposed scheme has good invisibility and high tampering localization accuracy on the feature addition and deletion attack.

Keywords: Fragile watermarking · Tamper localization
2D vector map · Batch features deletion

1 Introduction

During the past decade, the advent of digital maps has had a significant impact on the GPS navigation, digital city, smart transportation and other fields. Unfortunately, data security issues such as malicious tampering and illegal copying have not been well resolved. Then, fragile watermarking technology provides a new way to solve these problems [1,2]. According to the embedding position of the watermark, the fragile watermarking algorithm can be classified into two categories, one is frequency-based method and the other is spatial-based method.

Some algorithms are embedding the fragile watermark in the frequency domain. In [3], the perceived hash value was embedded in the wavelet subband of the carrier data. In [4], a semi-fragile watermarking algorithm based on frequency domain transform embedded the authentication information into high frequency region. These two watermarking strategies can accomplish the purpose of tamper detection, but these algorithms always have high complexity.

There are lots of spatial-based fragile watermarking strategies in previous studies. In [5], for each object in the map, the robust watermark was embedded into its feature points and the fragile watermark was embedded into its

G. Sun and S. Liu (Eds.): ADHIP 2017, LNICST 219, pp. 360–368, 2018.
https://doi.org/10.1007/978-3-319-73317-3_42

non-feature points. This method implemented the copyright protection and the content authentication for vector maps. In [6], a fragile watermarking scheme was proposed by expanding the Manhattan distances, witch located tampered data with high accuracy [6]. However these two schemes provide less embedded space for the watermark. To solve this problem, Neyman *et al.* created additional vertices for each feature to embed watermarks, they achieved the purpose of locating geometric attacks on received vector map [7]. Nevertheless, the feature rearrangement or vertex reversing operation may disturb the localization ability. In [8], the Douglas algorithm was used to simplify the map before the watermark embedding phase. This method allows users to compress the map, but the contents of the map are damaged to a certain degree. Wang *et al.* used a watermark embedding strategy proposed by Chou and Tseng [9], and designed a signature technique to enhance the localization accuracy [10]. However, these schemes may not be able to detect the batch features deletion attack and then result in passing a dummy authentication.

To solve these problems, we propose a feature group correlation technique to detect the missing group, apply it to the fragile watermarking scheme for 2D geographic data. In this scheme, we divide the spatial features into groups and apply the marking method to each feature. Then we use the correlation mark to mark each feature group. After that, we generate a RST invariant watermark and embed it with the method proposed in [9]. In the watermark authentication phase, we can identify the partial data of the missing group by the correlation mark of the feature group. In order to detect the exact location of the tampered content, the system will compare the extracted watermark with the reproduced watermark. Besides, our watermarking scheme inherits the RST invariance.

2 The Proposed Watermarking Method

Since the polygon feature in the 2D vector map can be seen as a closed polyline, our watermark embedding scheme is designed for polylines. Figure 1 shows the implementation model of our watermarking scheme.

2.1 Pretreatment for Vector Map

To begin with we will provide a brief introduction on the RST invariant fragile watermark embedding method [9]. There are three vertices V_w, T_c and V_n, called the watermark-embedding vertex, the neighboring center and the normalization vertex, respectively. Let $w(0 \leq w < S_w, S_w = 1, 2, 3, ...)$ be the watermark, S_w be an embedding parameter, K_w be a parameter to control the maximum distortion. First, we can obtain the standard quantization $Q_w = \|V_n T^c\| / K_w$. Second, V_w is moved to a new location V_w^e due to quantization operation.

$$V_w^e = V_w - \frac{V_w - T^c}{\|V_w T^c\|} \cdot (\|V_w T^c\| \bmod Q_w). \tag{1}$$

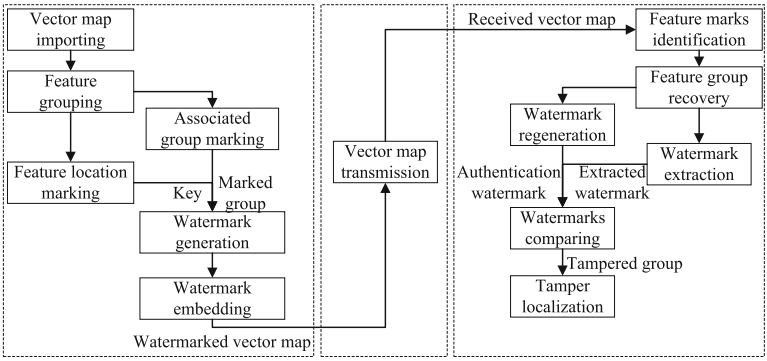


Fig. 1. The implementation model of the proposed fragile watermark algorithm

Third, w is embedded into V_w^e and the watermarked vertex $V_w{}'$ is obtained,

$$V_w{}' = V_w^e + \frac{V_w^e - T^c}{\|V_w^e T^c\|} \cdot \frac{Q_w}{S_w} \cdot w. \tag{2}$$

We assume that the length of watermark is L and a vertex carries c watermark bits. The polylines in the map are first divided into disjoint groups. The location ID [10] is used to indicate its group number and the position in the group. The vertices used to indicate the location ID are called mark vertices. Then we assign several extra marks for each group, called correlation mark, to record the information of the adjacent polyline group. The vertices used to indicate the correlation mark are called synergy vertices. For each polyline, we need two mark vertices, two synergy vertices, a normalization vertex and a neighboring center. Since these six vertices can no longer be used to carry the watermark, the total number of vertices on the polyline in which the watermark can be embedded should be at least $\lceil L/c \rceil + 6$. Therefore, the polyline which contains at least $\lceil L/c \rceil + 6$ vertices is an eligible polyline.

Given a vector map with Z polylines, we divide the polyline list into disjoint groups with the grouping method of [10]. Each group has $n (n \geq 1)$ polylines and contains at least one eligible polyline. The number of groups is $N_g = \lceil Z/n \rceil$. The first polyline in each group is an eligible polyline. We call this polyline as a watermark polyline, the second vertex of it as a reference1 vertex and the penultimate vertex of it as a reference2 vertex. We calculate the location ID of the $q^{th} (1 \leq q \leq n)$ polyline in the $p^{th} (0 \leq p \leq N_g - 1)$ group by $m_{p,q} = p \times n + q$.

In order to mark the synergy vertices, we denote the reference1 vertex of the watermark polyline in $G_p (0 \leq p \leq N_g - 1)$ as $v_{1,w}^p (v_{1,w}^{p,x}, v_{1,w}^{p,y})$, the reference1 vertex in the $G_q (q = (p+1) \bmod N_g)$ as $v_{1,w}^q (v_{1,w}^{q,x}, v_{1,w}^{q,y})$. Let s_1^x and s_1^y denote the sign bit of the difference of vertical coordinates and horizontal coordinates between $v_{1,w}^q$ and $v_{1,w}^p$, respectively. When the subtraction result is negative, the sign bit is 1, otherwise, the sign bit is 0. The offset caused by the vertical or horizontal coordinates is divided several times by 2 until the

result is less than 1. The times to do the division are denoted as $c_{1,x}$ and $c_{1,y}$. For example, if $\left|v_{1,w}^{q,x} - v_{1,w}^{p,x}\right| < 1$, $c_{1,x}$ is set as 0, otherwise, it is calculated by $\left\lfloor \log_2 \left|v_{1,w}^{q,x} - v_{1,w}^{p,x}\right| \right\rfloor + 1$. The offset values are denoted as Δx_1 and Δy_1,

$$\begin{cases} \Delta x_1 = c_{1,x} \times 10 + s_1^x + \left|v_{1,w}^{q,x} - v_{1,w}^{p,x}\right| / 2^{c_{1,x}} \\ \Delta y_1 = c_{1,y} \times 10 + s_1^y + \left|v_{1,w}^{q,y} - v_{1,w}^{p,y}\right| / 2^{c_{1,y}} \end{cases} . \tag{3}$$

Similarly, we denote the offset values between the reference2 vertex in G_q and the one in G_p as Δx_2 and Δy_2. In the subsequent design, we unified use Δx and Δy to represent the correlation marks. For each group, we hide the reference1 vertex's marks of the adjacent group in the watermark polyline, hide the reference2 vertex's marks of the adjacent group in the non-watermark polyline.

2.2 Watermark Embedding

Then, we divide the polyline into five categories: one is composed of more than five vertices (normal), one is composed of five vertices (complex1), one is composed of four vertices (complex2), one is composed of three vertices (complex3) and the other is composed of two vertices (complex4). The embedding results are illustrated in Fig. 2 by way of example. We use $2m_{i,j}$ to indicate the vertex order. The main emphasis is placed on the hidden methods of correlation mark.

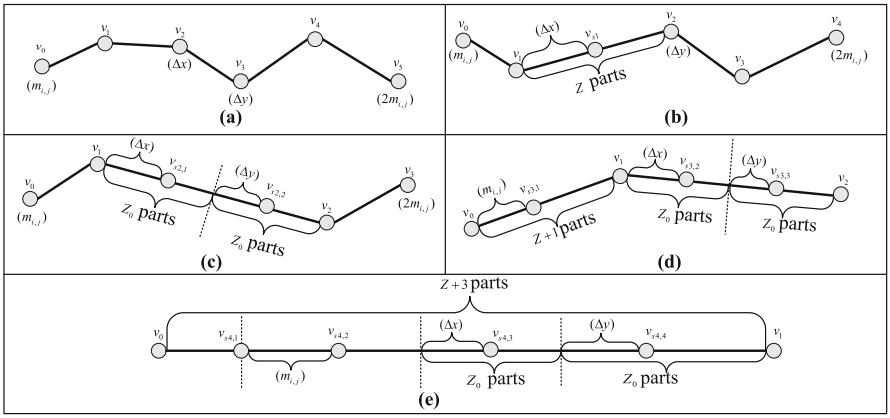


Fig. 2. Method of marking the location for different types of polylines

To embed the correlation mark into the vertex, such as the case in Fig. 2(b), according to Eqs. (1)–(2), we denote the reference vertices' maximum distance between the current group and its correlate group as $dst_{\max}$, define a parameter as $S_w = c_{dst} \times 10 + 2$. The c_{dst} is set as 0 when $dst_{\max}$ is less than 1, otherwise, it is set as $\lfloor \log_2(dst_{\max}) \rfloor + 1$. The parameter S_w is the higher limit of the processed offset values. And then a parameter $K_w = len_{\max}/\tau$ is defined, where $len_{\max}$ is

the maximum length of the polylines in the vector map M, and τ is the accuracy tolerance of M. We embed Δy into v_2 by regarding the vertices $v_1$and v_3 as the normalization vertex and the neighboring center, respectively.

If there is no free vertex to embed the correlation mark, we increase an extra vertex and express the correlation mark through the distance between the vertices. Such as the case in Fig. 2(b), a vertex v_{s1} are inserted between v_1 and v_2. The Euclidean distance between v_1 and v_2 is divided into Z_0 intervals, the number of intervals between v_1 and v_{s1} is equal to Δx, where Z_0 is equal to S_w which is calculated before. After that, Δx is hidden into the polyline.

For a marked group G_i^m with the watermark polyline Pl^m, we see Pl^m's $p(p = \lceil L/c \rceil)$ vertices from v_3 to v_{p+2} as the watermark vertices which is used to embed the watermark, use the rest of the vertices to generate the watermark to obtain a watermark H_i with the method in [10]. According to Eqs. (1)–(2), the reference2 vertex and the reference1 vertex of G_i^m, H_i is embedded in the watermark vertices. Finally, a watermarked vector map M^w is obtained.

2.3 Watermark Authentication

For a polyline Pl^w in received vector map, if the number of vertices on Pl^w is fewer than 6, it is detected as tampered directly. If the number of vertices on Pl^w is greater than 6, we see it as a possible marked normal polyline. If Pl^w has only six vertices, we identify its type with the following rules. First, check if the six vertices of Pl^w are on the same line, if so, see it as a possible marked complex4 polyline. Second, check if the first 3 adjacent vertices starting at one end and the first 4 adjacent vertices starting at the other end of Pl^w are collinear, respectively, if so, see it as a possible marked complex3 polyline. Third, check if the remaining 4 vertices after ignoring the ends of Pl^w are collinear. If so, see it as a possible marked complex2 polyline. Fourth, check if there are three adjacent vertices that are collinear when the ends of Pl^w are ignored. If so, see it as a possible marked complex1 polyline; otherwise, see it as a possible marked normal polyline. Then, it is easy to derive the extraction method from Sect. 2.1 to obtain the vertex order, location ID and correlation mark of each polyline. Assuming that a marked polyline's location ID is m, we can get its group number i and its inner position j in the corresponding group.

Afterwards, we can recovery the original group and derive the distance between the reference vertices of any two correlate groups. For a watermarked group G_i^w, we can obtain its watermark vertex list $V_i^{w\prime}$ and the parameter 1 vertex $v_{r1}{}'$ and the parameter2 vertex $v_{r2}{}'$. We use the input parameter K_w and set the parameter S_w as 2^c. For any watermark vertex $v_j{}'$, the watermark fragment $w_{i,j}{}'$ can be extract from $v_j{}'$. Then according to G_i^w's vertex order, watermark fragments can be connected to obtain the watermark W_i of the current group. Finally, we regenerate the watermark of G_i^w. Comparing the extracted watermark with newly generated one, we can judge whether G_i^w has been tampered.

3 Experiments and Results

We run experiments on a PC with 2.80 GHz, RAM 4.00 GB, Win7 Ultimate, ArcGIS Engine 10.2 and Visual C++6.0. We construct two datasets: one contains 50 maps and the other contains 30 maps. These maps are taken from the resources of ArcGIS. The inputs are set as follows: the number of watermark bit a vertex carries $c = 8$, group size $n = 3$ and the watermark length $L = 128$.

3.1 Verification of Invisibility

Four vector maps of the first dataset are used to show the invisibility of our scheme. They are a British expressway map, a railway map of Taiwan, a lake map of south part of China and an American expressway map. The precision tolerance τ of them are 1300, 200, 500 and 2500, respectively. They are watermarked by the proposed algorithm, the watermarked versions are shown in Fig. 3.

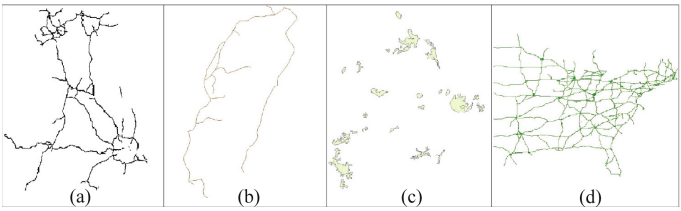


Fig. 3. The watermarked 2D vector maps

We use the average distortion d and the maximum distortion $Maxd$ [10] to measure the objective quality of the received vector map. Table 1 lists the results of three contrast algorithms for the invisibility of each test case and indicates that the introduced distortions do not exceed the tolerance.

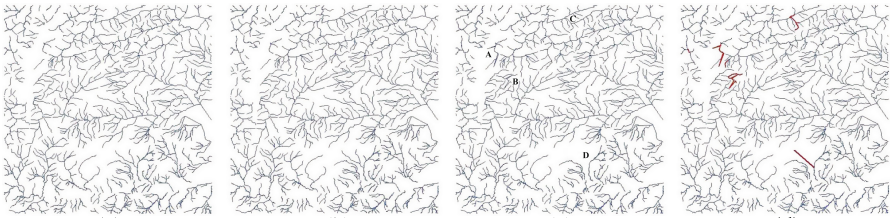
3.2 Discussion of Localization Accuracy

We choose a river map from the second dataset to test the tamper localization ability. Figure 4 shows the changes of this original map at different stages of watermarking. The original map in Fig. 4(a) is watermarked by our scheme yielding the watermarked map shown in Fig. 4(b). Afterwards, the watermarked map is manipulated to yield the map in Fig. 4(c). Expressly, we added 3 vertices to region 'A', modified 3 vertices in region 'B', deleted 5 vertices from region 'C' and deleted 3 polylines from region 'D'. The result of authentication can be seen from Fig. 4(d) which used red marks to indicate the located suspicious groups.

In order to test the tamper localization ability of our scheme, we applied the metrics β [2] which expresses the number of polylines detected as tampered after illegal attack. The expectation of β is denoted as $E(\beta)$, which is calculated

Table 1. The objective quality of the received vector map

2D vector map	The method in [7]		The method in [10]		The proposed method	
	$Maxd(m)$	$d(m)$	$Maxd(m)$	$d(m)$	$Maxd(m)$	$d(m)$
British expressway map	783.704	19.380	612.051	1.940	584.446	2.147
Railway map of China Taiwan	121.763	1.509	77.656	0.297	82.392	0.236
Lake map of south part of China	278.748	1.764	152.005	0.282	147.564	0.299
American expressway map	2231.491	52.441	1852.323	21.206	1981.193	25.191



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

Fig. 4. The changes of a river map at different watermarked stages (Color figure online)

to compare the localization accuracy of our algorithm with the ones proposed in [7] and [10]. In [7], Neyman *et al.* divide the polylines into disjoint groups based on the number of vertices. But it is hard to evaluate the number of vertices within each polylines. For simplicity, we assume a vector map with 100 polylines is divided into 10 groups, each group has 10 features, the probability of adding/deleting operation of the features in i^{th} group is 1/10. In particular, the probability of the case that a whole group is deleted after removing a small number of features is 0. We assume 10 polylines are missing after the batch features deletion attack. These polylines are in the same group or in two different groups. The probability of these two cases is equal. When we calculate $E(\beta)$ for the method reported in [10], we assume that the probability that the added feature is regarded as a valid feature is 1/2.

Results of the first three attack types in Fig. 5 shows that for the method reported in [7], the feature addition/deletion/rearrangement attacks may cause a different grouping result and a wrong tamper localization. From the performance of vertex reversing, feature rearrangement and RST attacks, we can find that our watermarking strategy is robust to resist these kinds of editing operations. From the comparison of the localization accuracy after the batch features deletion attack, we can see that our scheme can locate the missing group.

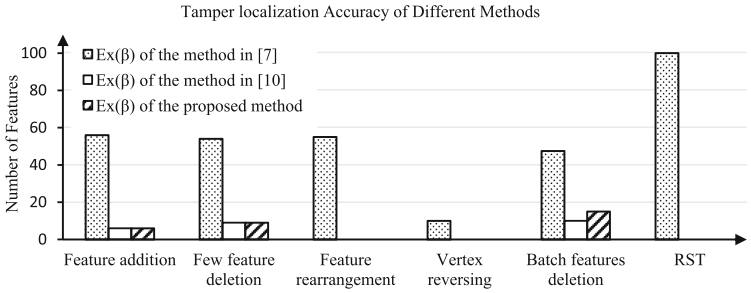


Fig. 5. Tamper localization accuracy of different methods

4 Conclusions

In this paper, we design a digital watermarking method for vector geographic data authentication based on the RST invariant fragile watermark embedding strategy. A grouping method and a feature location marking method are used to ensure the tamper localization accuracy. We design a feature group correlation technique to resist the batch features deletion attack which may lead to passing a dummy authentication. By folding the hash results of the differences of the log-radiuses, our scheme can resist the RST transformations. Furthermore, this watermarking algorithm is robust to resist the feature addition, deletion, rearrangement and vertex reversing attacks.

Acknowledgments. This work was supported by project of NSFC of China (61202455, 61472096, 61501132).

References

1. Wang, N.N., Zhao, X., Xie, C.: RST invariant reversible watermarking for 2D vector map. Int. J. Multimed. Ubiquit. Eng. **11**(2), 265–276 (2016)
2. Wang, N.N., Men, C.G.: Reversible fragile watermarking for 2-D vector map authentication with localization. Comput. Aided Des. **44**(4), 320–330 (2012)
3. Weng, L., Darazi, R., Preneel, B., Macq, B., Dooms, A.: Robust image content authentication using perceptual hashing and watermarking. In: Lin, W., Xu, D., Ho, A., Wu, J., He, Y., Cai, J., Kankanhalli, M., Sun, M.-T. (eds.) PCM 2012. LNCS, vol. 7674, pp. 315–326. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34778-8_29
4. Haojun, F.U., Zhu, C., Jian, M.: Multipurpose watermarking algorithm for digital raster map based on wavelet transformation. Acta Geod. Et Cartogr. Sin. **40**(3), 397–400 (2011)
5. Peng, Y., Lan, H., Yue, M.: Multipurpose watermarking for vector map protection and authentication. Multimed. Tools Appl. 1–21 (2017)
6. Neyman, S.N., Sitohang, B., Sutisna, S.: Reversible fragile watermarking based on difference expansion using manhattan distances for 2D vector map. Procedia Technol. **11**(1), 614–620 (2013)

7. Neyman, S.N., Wijaya, Y.H., Sitohang, B.: A new scheme to hide the data integrity marker on vector maps using a feature-based fragile watermarking algorithm. In: International Conference on Data and Software Engineering (ICODSE) (2014)
8. Ren, N., Wang, Q., Zhu, C.: Selective authentication algorithm based on semi-fragile watermarking for vector geographical data. In: 22nd International Conference on GeoInformatics (2014)
9. Chou, C.M., Tseng, D.C.: Affine-transformation-invariant public fragile watermarking for 3D model authentication. IEEE Comput. Graph. Appl. **29**(2), 72–79 (2009)
10. Wang, N.N., Bian, J., Zhang, H.: RST invariant fragile watermarking for 2D vector map authentication. Int. J. Multimed. Ubiquit. Eng. **10**(4), 155–172 (2015)