PROCEEDINGS OF SCIENCE

High Capacity Reversible Watermarking Scheme for 2D Vector Maps

Xiaowei Jing[1]
Information Management Department, China National Petroleum Corporation, Beijing, 100007, China
E-mail: jxw@petrochina.com.cn

Mei Feng
Research Institute of Petroleum Exploration and Development, China National Petroleum Corporation Beijing, 100083, China
E-mail: fm@petrochina.com.cn

Biao Guo
Beijing Petroleum Machinery Corporation, Drilling Research Institute, China National Petroleum Corporation, Beijing, 102206, China
E-mail: guobiaodri@cnpc.com.cn

Liang Chen
Research Institute of Petroleum Exploration and Development, China National Petroleum Corporation Beijing, 100083, China
E-mail: chen_l@petrochina.com.cn

Youjian Zhao
Department of computer science and technology, Tsinghua University, Beijing, 100084, China
E-mail: zhaoyoujian@tsinghua.edu.cn

Mingqin Geng
School of Information Engineering, China University of Geosciences, Beijing, 100083, China
E-mail: 2002011563@cugb.edu.cn

This paper presents a high capacity reversible watermarking scheme for 2D vector maps. This scheme applies an integer transform to a coordinate pair. The integer transform does not change the sum value of the coordinate pair and embeds one watermark bit. Except the first and the last one, each coordinate in an object with its previous and next one will compose two pairs respectively. Namely, each coordinate can embed one watermark bit. A threshold on difference values is predefined to distinguish the embedded pairs. As to one pair, if its absolute value of the difference is less than the threshold, the integer transform will be applied; otherwise, it is shifted. The shifted distance is larger than the embedded one so that the decoder can differentiate the two classes. To assure the distortion introduced by the transform and the shift less than the map precision distortion, the embedding condition is presented. Simulation results are provided to show that the scheme can achieve high capacity while maintaining low distortion value.

CENet2015
12-13 September 2015
Shanghai, China

[1]Speaker
Correspongding Author

PoS(CENet2015)034

1. Introduction

Nowadays, the information security and the copyright protection have become increasingly important. In particular, the acquisition and management of 2D vector map data requires great human, material and financial resources. It would be desirable to have a technique to protect the copyright and the content integrity. Digital watermarking is one of the widely used techniques adopted worldwide in the area of information security because it hides invisible information in the cover data and most of them modify the the covered data permanently. In some critical cases, 2D vector maps can not tolerate any distortions; thus the reversible watermarking, which can recover the original data after extracting the watermark data, is suitable for the critical applications.

There have been fewer reversible watermarking methods for 2D vector maps. The first algorithm in this literature, which was proposed by Voigt et al., modified the highest frequency DCT coefficient to modulate one watermark bit[1]. Shao et al. and Wang et al. employed Tian's difference expansion in 2D vector maps and embedded the watermark data by expanding the differences between two neighbouring coordinates[2, 3, 4]. In order to control the distortion, they presented the embedding condition such differences. They used the location map to record the expansion locations like Tian's algorithm based on difference expansion[4]. The location map consumed large part of the embedding capacity. Zhou's and Wu's algorithms are also based on the difference expansion[5, 6], but the difference histogram shifting technique took the place of location map to distinguish the expanded locations; as a result, the two algorithms increased the capacity. Zhou controlled the capacity by the difference histogram[5].

The algorithms hided data in disjoints pairs. And thus they did not take full advantage of the coordinates[2, 5, 6]; moreover, like Tian's method, these algorithms performed three steps to embed the watermark. To solve the first problem, the proposed scheme consecutively operates every coordinate value except the last one. Hence, more embedding space is achieved. According, the capacity has been increased largely. To solve the second problem, the proposed scheme applied an integer transform on the original coordinate values to embed one watermark bit and calculate the marked coordinate values[7, 8]. The computational complexity has been greatly reduced. The invariability of sum of the coordinate value features the integer transform and it is the key to restore the original data. To control the distortion, not all the coordinate pairs are embedded with some shifted. Whether the embedding or the shifting depends on the threshold which is pre-set. The shift distance is larger than the majority of all the embedded distances. Thus, the embedded differences and the shifted ones lie in different ranges, by which the decoder can distinguish them. What's more, the paper presents the embedding condition in order so that the distortion is less than or equal to the map precision tolerance.

Next, we introduce the integer transform in Section 2. This is followed by the detailed description of the proposed scheme in Section 3. Next come experimental results and the comparison with other schemes in Section 4. Finally, we draw a conclusion in Section 5.

2. The Integer Transform

This Section provides a brief view of the integer transform[8]. In 2D vector maps, two neighboring points constitute one pair (p_1, p_2). Each point contains two coordinates, X and Y. Thus we obtain two coordinate pairs. The X coordinate pair is (x_1, x_2), and the Y (y_1, y_2). Both pairs can hide watermark data in the same way. Then we take X coordinate for instance. We denote the watermark bit as w. Firstly, we calculate the difference and the sum as

$$\begin{cases} h = x_1 - x_2 \\ s = x_1 + x_2 \end{cases} \tag{2.1}$$

Then we apply the integer transform on the coordinate pair to hide one watermark bit as

$$\begin{cases} x_1' = x_1 + (\lfloor \frac{h}{2} \rfloor + w) \\ x_2' = x_2 - (\lfloor \frac{h}{2} \rfloor + w) \end{cases} \tag{2.2}$$

From Equation (2.2), we calculate

$$s' = s \tag{2.3}$$

The transform does not change the sum of the coordinate values. Then we explore the parities of the sum and the difference. We denote the parity of a number n is $\text{Parity}(n)$. If n is even, the function returns 0, otherwise it returns to 1. Since $2x_2$ is an even number and $s = x_1 + x_2 = x_1 - x_2 + 2x_2 = h + 2x_2$, $\text{Parity}(s) = \text{Parity}(h)$. Likewise, $\text{Parity}(s') = \text{Parity}(h')$. Because of (2.3), we have

$$\text{Parity}(h) = \text{Parity}(h') \tag{2.4}$$

The invariant parity of the differences is the key to recover the original vector map.

Now we describe how to extract the watermark bit and restore the original data. By Equations (2.1) and (2.2), the original difference between x_1 *and* x_2 is $h = h' - 2\left\lfloor \frac{h}{2} \right\rfloor - 2w$. Because of $\left\lfloor \frac{h}{2} \right\rfloor = \frac{h - \text{Pariry}(h)}{2}$ and Equation (2.4), we can calculate h by h' as

$$h = \frac{h' + \text{Parity}(h')}{2} - w \tag{2.5}$$

Since h and h' have the same parity, the watermark w can be deduced as

$$w = \begin{cases} 0, \ if \ Parity(h') = Parity\left(\frac{h' + Parity(h')}{2}\right) \\ 1, \ otherwise \end{cases} \tag{2.6}$$

By Equation (2.5) and w, we can calculate the original difference h. By (2.3), we can calculate the original sum s. Then the original coordinate values can be restored as

$$\begin{cases} x_1 = \frac{s+h}{2} \\ x_2 = \frac{s-h}{2} \end{cases} \tag{2.7}$$

3. The Proposed Scheme

Upon embedding, new differences and unchanged differences will overlap. To solve the problem, the un-embedding coordinate values will be shifted. Whether the embedding or the shifting depends on a threshold T, which is determined in turn by the embedding capacity and the distortion. Then we can embed the watermark and in the decoder we extract the hidden information and recover the original data. One of the key steps is pairing the coordinates.

Instead of grouping the coordinate values into disjoint pairs before embedding and extracting in algorithms , we pair the coordinate values jointly on embedding and extracting. Suppose we have n coordinate values $(x_1, x_2, \cdots, x_{n-1}, x_n)$ and n is even. In the methods, the coordinate values are grouped into pairs $(x_1, x_2), (x_3, x_4) \cdots, (x_{n-1}, x_n)$. We operate them from x_1 to x_n as follows. Like the disjoint pair, the first original and operated pair are (x_1, x_2) and (x_1', x_2') respectively. But the following pairs are different. The next pair is composed of the operated coordinate value x_2' and the original coordinate value x_3. We denote it as (x_2'', x_3') after being operated. The others are operated as above and then all the coordinate values become $(x_1', x_2'', \cdots, x_{n-1}'', x_n')$. Clearly, we can hide $(\mathrm{n} - 1)$ bits at most while $(n/2)$ in disjoint pairs.

3.1 Coordinate Value Shifting

We preset a threshold T to determine embedding and shifting. If the difference meets the condition $|h| \leq T$, we embed the watermark in this pair and classify it into the set EMD. Clearly, the range of the original differences H_{EMD} is $[-T, T]$. After embedding, the new difference H'_{EMD} will fall into the range $[-2\mathrm{T} - 1, 2T + 2]$; otherwise, i.e. $|h| \geq T + 1$, we shift the coordinate values and classify it into the set SHT. Obviously, its original range H_{SHT} is $(-\infty, -\mathrm{T} - 1] \cup [\mathrm{T} + 1, +\infty)$. The range of the embedded differences H'_{EMD} and the range of the un-embedded differences H_{SHT} will overlap so as not to distinguish them in the decoder. To

solve the problem, we shift the un-embedded coordinate values. The shifting displacement is denoted as d, which is a positive integer. After shifting, the differences H'_{SHT} must meet the condition $H'_{SHT} \leq -2\mathrm{T}-2$ or $H'_{SHT} \geq 2\mathrm{T}+3$.

If h in set SHT is positive, i.e. $\mathrm{h} \geq \mathrm{T}+1$, the shifting is defined as $x'_1 = x_1 + d$ and $x'_2 = x_2 - d$. The new difference is $h' = \mathrm{h} + 2\mathrm{d}$. That is $\mathrm{h} + 2\mathrm{d} \geq 2\mathrm{T}+3$. So d must meet the condition $\mathrm{d} \geq \lfloor (T+3)/2 \rfloor$. If h in set SHT is negative, i.e. $\mathrm{h} \leq -\mathrm{T}-1$, we define the shifting as $x'_1 = x_1 - d$ and $x'_2 = x_2 + d$. After shifting, the difference is $h' = \mathrm{h} - 2\mathrm{d}$. Then we get the inequality $\mathrm{h} - 2\mathrm{d} \leq -2\mathrm{T}-2$. Thus d must satisfy the condition $\mathrm{d} \geq \lfloor (T+2)/2 \rfloor$. We take the minimum values of d to minimize the distortion.

The shifting for the pairs in set SHT is

$$\begin{cases} x'_1 = x_1 + sign(h) \times \left\lfloor \frac{T+3}{2} \right\rfloor \\ x'_2 = x_2 - sign(h) \times \left\lfloor \frac{T+2}{2} \right\rfloor \end{cases} \tag{3.1}$$

where the function $\mathrm{sign(h)}$ returns 1, 0 and -1 when h is positive, zero, negative, respectively. After shifting, the differences occupy the range $(-\infty, -2\mathrm{T}-2] \cup [2T+3, +\infty)$. The decoder can distinguish the embedded pairs from the shifted ones by their difference ranges. What's more, the decoder can easily recover the original data in set SHT as

$$\begin{cases} x_1 = x'_1 - sign(h') \times \left\lfloor \frac{T+3}{2} \right\rfloor \\ x_2 = x'_2 + sign(h') \times \left\lfloor \frac{T+2}{2} \right\rfloor \end{cases} \tag{3.2}$$

3.2 Embedding Condition

Whether the embedding or the shifting will introduce distortion to the original data. The distortion should less than or equal to the map's precision tolerance. From the above description, in one object, the first coordinate value and the last one are operated once with the others twice. We firstly descript the second case. The two operations are as Table 1.

At the same time, four calculations on the coordinate values are listed in Table 2, where N_n is a negative number and N_p is a positive number.

First operation	Second operation
Embedding	Embedding
Embedding	Shifting
Shifting	Embedding
Shifting	Shifting

Table 1: The Two Operations

Case	First calculation	Second calculation
a	$-N_n$	$+N_n$
b	$-N_n$	$+N_p$
c	$-N_p$	$+N_n$
d	$-N_p$	$+N_p$

Table 2 : The four Calculations

In cases a) and d), the second calculation offsets the first one to some extent. Cases b) and c) will introduce the greatest modification. Thus we analyze the two cases; besides, the modifications by two calculations are greater than one calculation. In summary, the greatest modification must be introduced by the two cases.

N_n and N_p are different in terms of embedding and shifting. If we embed the watermark as Equation (2), the smallest and greatest modifications occur when $\mathrm{h} = 0$ and $\mathrm{w} = 0$, and $\mathrm{h} = \mathrm{T}$ and $\mathrm{w} = 1$, respectively. We take the greatest modification into consideration. It is

$M_E = \lfloor T/2 \rfloor + 1$, where M_E denotes the modification in case of embedding. On the other hand, if we shift the coordinate values as Equation (3.1), there will be only two modifications, i.e. $M_S = \lfloor (T+3)/2 \rfloor$ and $M_S = \lfloor (T+2)/2 \rfloor$, where M_S is the modification when shifting. Clearly, $M_E \leq M_S$. Then the greatest modification when operating twice which is denoted as M_{2max} is $M_{2max} = \lfloor (T+3)/2 \rfloor + \lfloor (T+2)/2 \rfloor$. M_{2max} should be less than or equal to τ. That is $M_{2max} \leq \tau$. Then we obtain the embedding condition on the threshold as

$$\mathrm{T} \leq \tau - 2 \tag{3.3}$$

If T takes the maximum value, we will obtain the greatest capacity. If we embed less information, T will take smaller value and thus the distortion will be decreased.

3.3 Embedding and Extracting Processes

In a vector map, we embed the watermark bits in the polylines and polygones one by one. And in each object, the coordinate values one by one are operated. The embedding progress is shown as follows.

The neighboring coordinate values constitute a pair (x_i, x_{i+1}), where $\mathrm{i} \in \{1, \cdots, \mathrm{n}-1\}$ and n is the total number of the vertices in an object. The left coordinate value in a pair is already operated except for the first coordinate value x_1.

We calculate the difference of the pair by Equation (2.1) and denote it as h_i.

If $-\mathrm{T} \leq h_i \leq T$, we embed one watermark bit in this pair by Equation (2.2). Otherwise, the two coordinate values are shifted as Equation (3.1).

We increase i by one and repeat Step 1-4 till we embed all the watermark bits.

Unlike the embedding process, the extracting process proceeds in the inverse order. It consists of four steps.

The neighboring coordinate values make up of a pair (x'_i, x'_{i+1}), where $\mathrm{i} \in \{1, \cdots, \mathrm{n}-1\}$ and n is the total number of the vertices in an object. The left coordinate value in a pair is operated twice except for the first coordinate value x_1 which is operated once.

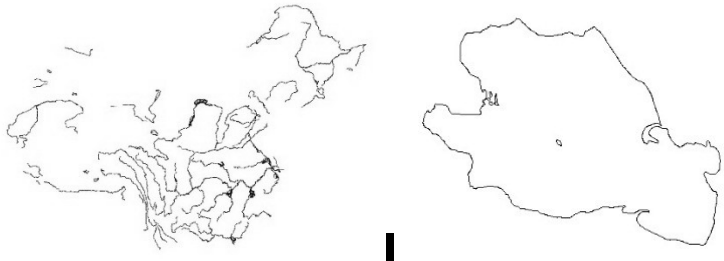
We calculate the difference and the sum values of the pair by Equation (2.1), which are denoted as h'_i and s_i respectively.

If $-2\mathrm{T}-1 \leq h_i \leq 2T+2$, we deduce the watermark bit w and calculate the original difference h_i by Equations (2.5) and (2.6). Then we calculate the original values by Equation (2.7). Otherwise, we recover the original values as Equation (3.1).

We decrease i by one, and repeat Step 1-4 till we extract the watermark bits and recover the original values. Note that the extracted bits are also in the reverse order.

4. Experimental Results

We test our method and compare it with Shao's and Zhou's methods. The tests are performed on two maps, namely a river map and a contour map[2, 5]. There scales are 1:4000000 and 1:10000 respectively. Other parameters are shown in Table 1.3. The original river map is reduced 100 times as shown in fig. 1(a). Note that the dashed box is marked by authors. Fig. 1 (b) is reduced by 10 times as to one part of the original map. One part of the original contour map is magnified by 10000 times as shown in Fig 2.



(a) The River Map (b) One Part of the Original River Map

Figure 1: Original River Map

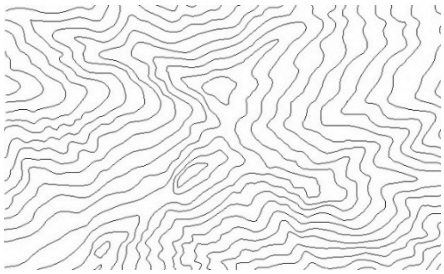
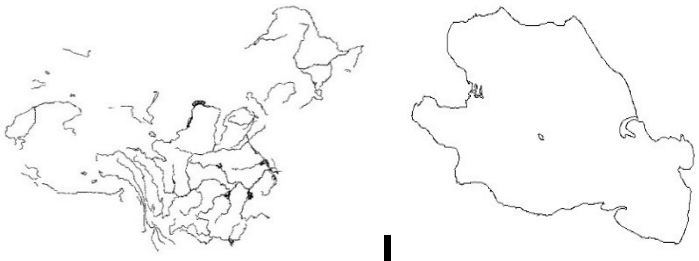


Figure 2: Original Contour Map

Map	Scale	Number of vertices	Map's tolerance precision (meter)	Decimal digits
River	1:4000000	137554	500	6
Contour	1:10000	176526	6	7

Table 3: Parameters of Original Maps

We embed the watermark bits in X and Y coordinate values. And the watermark bits are generated randomly. We experiment the distortions at different capacities including the maximum capacity. Fig. 3 is the marked river map at the maximum capacity $bpc = 0.81$ with $RMSE = 0.0014$. Fig. 4 is a part of the marked contour map at the maximum capacity $bpc = 0.66$ with $RMSE = 0.000016$. We use bpc (bit per coordinate value) to measure the capacity and $RMSE$ to calculate the distortion. The results are shown in Figs. 1.5 and 1.6. Compared to Shao's and Zhou's methods, our method has achieved higher capacity at the same RMSE and introduce lower distortion at the same capacity[2, 5].



(a) The Marked River Map (b) One Part of the Marked River Map

Figure 3: The Marked River Map

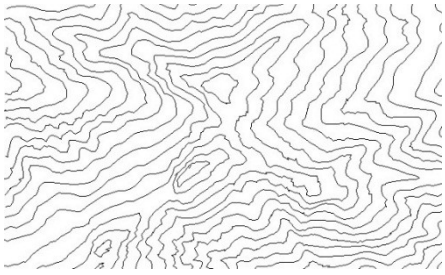


Figure 4: The Marked Contour Map

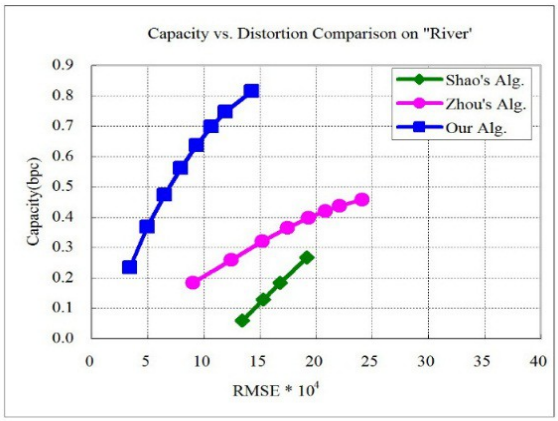


Figure 5: Capacity vs. Distortion Comparison on "River"

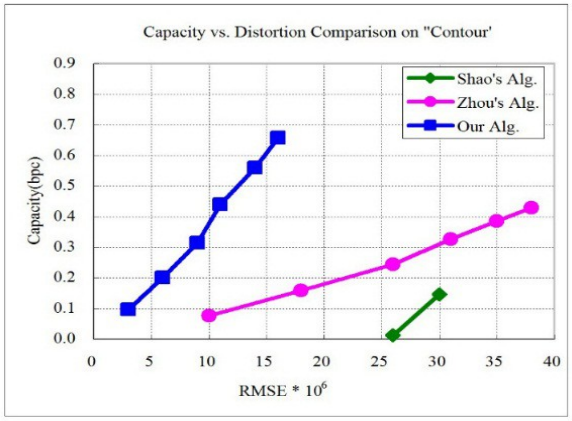


Figure 6: Capacity vs. Distortion Comparison on "Contour"

5. Conclusion

We propose a reversible watermarking method for 2D vector maps. All the coordinate values but two are embedded or shifted twice. Then we obtain higher embedding space; besides, two operations can decrease the distortion at some cases. By the threshold, we embed in a pair or shift the coordinate values. The two operations will make their differences to fall into different ranges. Thus the decoder can distinguish the embedded pairs from the shifted pairs and recover the original data. Experimental results show that our method can achieve high capacity at low distortion. Experimental results indicate the method will increase the capacity.

References

[1] Michael Voigt, Bian Yang, Christoph Busch, *Reversible watermarking of 2d-vector data*, In Proceedings of ACM 2004 Workshop on Multimedia and Security, ACM, New York, USA, pp. 160 ~ 165. 2004.

[2] Chengyong Shao, Xiaotong Wang, Xiaogang Xu, Xiamu Niu, *Study on lossless data hiding algorithm for digital vector maps,* Journal of Image and Graphics, Vol. 12, No. 2, pp. 206 ~ 211, 2007. (In Chinese)

[3] Xiaotong Wang, Chengyong Shao, Xiaogang Xu, Xiamu Niu, *Reversible data-hiding scheme for 2-D vector maps based on difference expansion*, IEEE Transactions on Information Forensics and Security, Vol. 2, No. 3, pp. 311 ~ 320, 2007.

[4] Jun Tian, *Wavelet-based reversible watermarking for authentication*, In Proceedings of SPIE

Security and Watermarking of Multimedia Content IV, SPIE, San Jose, CA, Vol. 4675, No. 74, pp. 679 ~ 690, 2002.

[5] Lu Zhou, Yongjian Hu, Huafei Zeng, *Reversible data hiding algorithm for vector digital maps*, Journal of Computer Applications, Vol. 29, No. 4, pp. 990 ~ 993, 2009. (In Chinese)

[6] Dan Wu, *A reversible watermarking scheme for 2D vector maps*, Software Engineering and Knowledge Engineering: Theory and Practice, Springer, Berlin Heidelberg, Vol. 2, pp. 197 ~ 203, 2012.

[7] Dinu Coltuc, Alain Tremeau, *Simple reversible watermarking schemes*, In Proceedings of SPIE Security, Steganography, and Watermarking of Multimedia Contents VII, SPIE, San Jose, CA, Vol. 5681, pp. 561 ~ 568, 2005.

[8] Shaowei Weng, Yao Zhao, JengShyang Pan, Rongrong Ni, *Reversible watermarking based on invariability and adjustment on pixel pairs*, IEEE Signal Processing Letters, Vol. 15, No.1, pp. 721 ~ 724, 2008.