A VECTOR MAPS WATERMARKING ALGORITHM BASED ON DCT DOMAIN

Bin Liang, Jiang Rong, Chuanjian Wang*

Shihezi University, College of Information Science and Technology, 832003, Shihezi, China – (binl_inf, rj_inf)@shzu.edu.cn, wangchj@gmail.com

KEY WORDS: Vector Maps, Watermarking, Discrete Cosine Transform, Minimum Bounding Rectangle, Simplification

ABSTRACT:

The acquisition of vector maps is a costly process in which large amount of labour and material resources are needed for acquiring original vector map. It is very important to protect intellectual properties of vector maps under complicated network environment and prevent vector maps from being illegally commercialized abused. In this paper, we propose a watermarking algorithm for vector map based on discrete cosine transform. Watermark is hidden DCT coefficients with middle frequency. Experiments show that our watermarking algorithm is robust against many attacks such as rotation, scaling, translation, and simplification.

1. INTRODUCTION

The acquisition of vector maps is a costly process in which large amount of labor and material resources are needed for acquiring original vector map. However, the rapid development of communication and internet techniques makes it very easy to exchange vector map via networks. It is very important to protect intellectual properties of vector maps under complicated network environment and prevent vector maps from being illegally commercialized abused. Recently, watermarking techniques are exploited to protect copyright of vector maps.

One spatial domain watermarking scheme which embeds watermark by moving the vertices or modifying vertices' coordinates. The basic idea of (Sakamoto, 2000; Kang, 2001; Schulz, 2004) is to construct a predefined structure (masks in ((Sakamoto, 2000; Kang, 2001)), strips in (Schulz, 2004)), shift the vertices within the predefined structure and make the location relationship follows a specific pattern representing '0' or '1'. The algorithm in (Ohbuchi, 2002) embeds watermark bits into a map by displacing a set of vertices in different rectangles which are generated by adaptively subdividing the map according to the density of vertices in the region. Huang proposed a database watermarking algorithm which inserts and detects mark bits with the ability of error correcting in the coordinates of the vertices in geographical objects using the methods of classifying and twice majority-voting. Furthermore, watermarking algorithm is integrated as a function model into the open source DBMS PostgreSQL(Huang, 2007). Some other spatial algorithms watermark curves using curve modeling methods such as chain code, Fourier descriptors and B-splines. The watermarking algorithm in (Solachidis, 2000; Doncel, 2007) exploits Fourier descriptors' rotation, scaling, translation invariant properties, and embeds copyright information in the magnitude of Fourier descriptors. Watermark is achieved by imperceptibly modifying the coordinates of vertices that define a polygonal line through the modification of the magnitude of Fourier descriptors of the line. Li and Xu proposed a blind scheme, in which the amplitude of the coordinates' DWT coefficients are modified to specific modes for representing the watermark bits(Li, 2003).

In this paper, we propose a watermarking algorithm for vector map based on discrete cosine transform. The Minimum Bounding Rectangle (MBR) of a vector map to be watermarked is computed firstly when the feature domain is extracted. Based on the MBR, we divide the map into cells, construct a *grid weight array*. We apply DCT transform to *grid weight array* and embed the watermark into DCT coefficients with middle frequency, and then modify the distribution of the map to derive a watermarked map.

The remainder of this paper is organized as follows: Section 2 discusses the feature domain extraction process and presents watermark embedding and detection process in details. We also discuss how to identify feature vertices and select candidate cells in order to improve the fidelity of the watermarked map. Experimental results are presented in Section 3 to demonstrate the robustness of our method against geometry transformation and simplification attacks. Our paper is concluded in Section 4.

2. THE PROPOSED ALGORITHM

2.1 Feature Domain Extraction

In our proposed algorithm, the feature domain is extracted based on MBR. The extraction process is described as follows:

(1) Computing the MBR of a vector map. Given a vector map, the MBR is the rectangle which encloses the set of vertices and is oriented toward x and y axes.

(2) Gridding vector map. Taking the MBR as boundary, we divide the vector map into $j*k$ cells with same size. Two parameters j and k are saved as the user key which will be used to grid the vector map when watermark is detected. By gridding the vector map, the vertices in the map are distributed into different cells.

(3) Constructing grid weight array. Count the vertices in every cell. Due to irregularity of vector map, some cells do not contain vertices. We cannot embed watermark into empty cells. Therefore, we store the vertices' number in all cells which is

* Corresponding author. wangchj@gmail.com

not empty to a grid weight array A. Let A = [$a(n)$], n = 0,1,…N, where $a(i)$ is the number of vertices in ith cell.

(4) Transforming from spatial to frequency domain. Generally speaking, the spatial watermarking scheme is fragile to geometry attacks. We transform the grid weight array to frequency domain. The change of a frequency coefficients influences several cells, and the change of one cell coefficient spreads to several frequency coefficients. Therefore, one frequency coefficient is not largely influenced although a cell coefficient, the number of vertices, is changed by the attack such as elimination of objects. Let $y(k)$ be the unitary discrete cosine transformation representation of grid weight array A

$$y(k) = w(k)\sum_{n=1}^{N} a(n)\cos\frac{\pi(2n-1)(k-1)}{2N}, \qquad k = 1,2,..., N$$

where

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}} & k = 1 \\ \sqrt{\frac{2}{N}} & 2 \le k \le N \end{cases}$$

Here, the DCT coefficients y(k), k=1,2,…,N, are feature domain on which watermark is embedded and detected.

2.2 Watermark Construction

In this paper, we select binary BMP image whose size is $W\times W$ as watermark data. Even if a small part of pixels are destroyed, binary BMP image can declare the copyright of the map easily. To keep copyright information private, the binary image will be permutated before it is embedded. There are several image permutation techniques such as Arnold permutation, Hilbert permutation and Magic permutation, etc. We scramble the watermark image with Arnold permutation. The time of the permutation is saved as the user key which will be used to detect watermark. Arnold permutation increases the security of watermark. After permutation, pixels in watermark image are translated to a binary bit vector $\boldsymbol{b} = (b_1, b_2, \ldots, b_{m=W*W})$ in which each bit takes value {0,1}. Each bit b_i is duplicated by *chip rate c*, producing a watermark bit vector $\boldsymbol{w} = (w_1, w_2, \ldots, w_{mc})$. Repeatedly embedding the same bit c several times can increase the robustness of the method.

2.3 Watermarking Algorithm

DCT coefficients are used as the feature domain for watermarking. Watermark is embedded by slightly modifying the DCT coefficients. DCT coefficients can be categorized into three ranges: low frequency, middle frequency and high frequency. According to the characteristic of DCT transformation, most of energy of DCT coefficients concentrates in the low frequency. If watermark is embedded in DCT coefficients with low frequency, the watermarking algorithm is more robust, whereas the distortions caused by watermark embedding become more severe. If watermark is embedded in DCT coefficients with high frequency, the distortion caused by watermark embedding is slight, whereas the robustness of the watermarking algorithm becomes bad. To get the tradeoff of the robustness of the proposed watermarking algorithm and data fidelity, we embed watermark in DCT coefficients with middle frequency $y(k)$, $k \in$ [*StartIndex*, *EndIndex*] , 0<*StartIndex* < *EndIndex* <*N*, *EndIndex* = *StartIndex* + *mc*, where parameters *StartIndex* and *EndIndex* control the frequency range that the watermark affects. Two parameters *StartIndex* and *EndIndex* are kept as the user key.

Watermark is embedded by modifying the sign of the DCT coefficients with frequency. Watermark modulation process can be expressed as

$$y(StartIndex+i)' = \begin{cases} |y(StartIndex+i)| & w(i)=1 \\ -|y(StartIndex+i)| & w(i)=0 \end{cases} ,1 \le i \le mc$$

where *StartIndex* is the first DCT coefficient with middle frequency, $w(i)$ is the ith watermark bit, mc is the length of watermark string. If $w(i)$ is '1', the sign of the coefficient is modified to positive, and if $w(i)$ is '0', the sign of the coefficient is modified to negative.

After modification, the inverse DCT transformation is applied to the coefficients $y(k)'$, k = 0,1,…,N, to generate a modified *grid weight array A'*

$$a(n)' = \sum_{k=1}^{N} w(k)y(k)'\cos\frac{\pi(2n-1)(k-1)}{2N} \qquad n = 1,2..., N$$

where

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}} & k = 1 \\ \sqrt{\frac{2}{N}} & 2 \le k \le N \end{cases}$$

We should modify the distribution of vertices in cells based on the *grid weight array A*' to derive a watermarked vector map. The distribution of vertices in all cells is modified according to *A*'. If $a(i)' > a(i)$, the vertices in ith candidate cell must be increased. There are two methods to insert vertices into one cell: duplication or interpolation. Duplication of vertices is easy to realized, but is fragile to simplification attack, because duplicate vertices will be deleted directly when the watermarked map is simplified. So curve interpolation is used to insert vertices into candidate cells.

If $a(i)' < a(i)$, some vertices in ith candidate cell must be removed. A common principle followed by all watermarking schemes is that the embedding of a hidden message should not degrade the validity of the cover data. Removal of *feature vertices* will cause high distortion and degrade the quality of the map. Therefore, we only delete *non-feature vertices* and distortion caused by removal of *non-feature vertices* cannot exceed the precision tolerance of the map. The Douglas-Peuker algorithm is used to identify *non-feature vertices* based on a threshold T_{DP}.

2.4 Watermark Detection

Watermark detection process is similar to watermark embedding process. Given a watermarked map $\mathcal{M}_0$, firstly compute its MBR. Secondly, partition the vector map to be detected into cells with the same size based on its MBR, count the vertices in each cell, and construct *grid weight array A*. Thirdly, apply DCT transformation to *grid weight array A* and derive the DCT coefficients $y(k)$, k=1,2,…,N. two parameters *StartIndex* and *EndIndex* are needed to specify the range of the middle-frequency DCT coefficients in which watermark will be embedded. Watermark vector $\boldsymbol{w}$ is estimated as

$$w(i) = \begin{cases} 1 & if \quad y(StartIndex+i) > 0 \\ 0 & if \quad y(StartIndex+i) < 0 \end{cases} \qquad i = 0,1,..., mc$$

As the original binary bit vector $\boldsymbol{b}$ is duplicated c times, a majority voting schema is employed to decide what the original binary sequence $\boldsymbol{b}$ is from watermark vector $\boldsymbol{w}$. After the bit vector ***b*** is translated to BMP image and then Arnold permutation, it is converted into the original binary BMP image for watermarking.

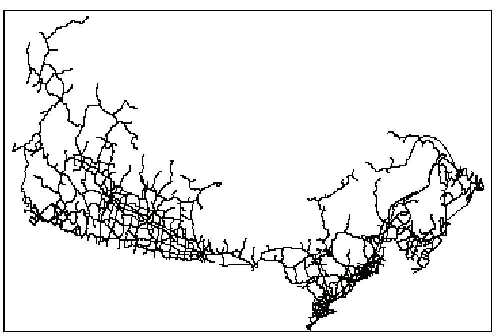
3. EXPERIMENTS

In this section, we examine the fidelity of the watermarked map, evaluate the robustness of the watermarking algorithm against rotation, scaling, translation and simplification.

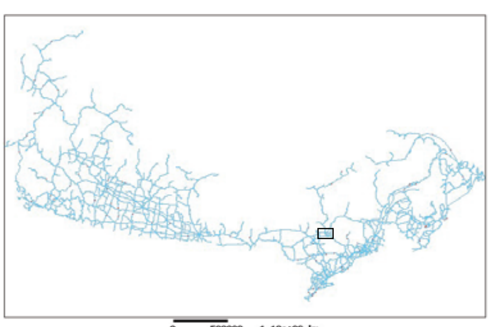
In experiments, PostgreSQL and PostGIS are used to store and manage vector maps, and MapServer is used to visualize vector maps. The algorithm is implemented in C++. Test data is Canada map consisting of layers of road, railway, provincial district, etc. The watermark is embedded in road layer in our experiment. The original map shown in Figure 1(a) consists of 1548 polylines and the number of vertices is 42489. The initial watermark parameters are set as follow: the map is partitioned into 256*256 cells, watermark image is 32*32 binary BMP file, the number of Arnold permutation time is 24, *chip rate c* is 1, and *non-feature vertices* identification threshold T_{DP} is 2m.

(1) Fidelity of the Watermarked Map
We watermark the map. In Figure 1(b), we overlay the original map and the corresponding watermarked map using red lines and blue lines respectively. To verify the fidelity of the watermarked map, we enlarge a portion of the overlaid image in Figure 1(c).

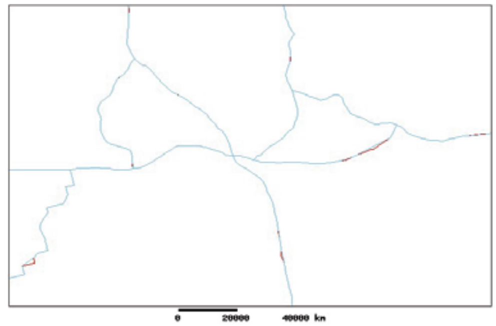
We can see that the distortion caused by watermark embedding is small and the watermarked map preserves the geospatial information in the original map with high precision.



(a)



(b)



(c)

Figure 1. Watermarking vector map. (a) Original map. (b) Original and watermarked map overlaid with each other. (c) Zoom-in view of (b).

(2) Resilience to Affine Transformation
To demonstrate the resilience of our method to geometry transformation, we take the watermarked map and apply a combination of translation, rotation and scaling. The attack sequence and corresponding attack magnitudes are list in Table 1. Every attack in the sequence is based on the adjacent previous one in the sequence. Experiments show that our method has a good robustness and is resilient to geometry transformation with arbitrary magnitudes. As shown in Table 1, the normalized correlation (*NC*) value between the detected watermark image and the original watermark image doesn't change with different attacks.

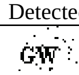
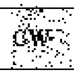
	Attack Magnitude	Watermark Detected	Normalized Correlation
attack	none		0.8528
Rotation (°)	-133°		0.8528
	-58°		0.8528
	290°		0.8528
	355°		0.8528
Scaling (%)	33%		0.8528
	130%		0.8528
	202%		0.8528
	305%		0.8528
Translation (*x*,*y*)	(-110,35)		0.8528
	(660,194)		0.8528
	(590,-353)		0.8528
	(-99,-101)		0.8528

Table 1. Resilience to geometrical attacks

(3) Resilience to Simplification

To demonstrate the resilience of our method to simplification attack, we apply Douglas-Peuker algorithm to simplify vector map. Table 2 shows the resilience of our method to simplification attack.

Table 2. Resilience to simplification attack.

Simplification Distance	Watermark Detected	Normalized Correlation
none		0.8528
0.0001		0.7862
0.01		0.7438
0.1		0.6687

4. CONCLUSIONS

In this paper, we have presented a new watermarking algorithm for vector maps based on DCT. The MBR of a vector map is computed before watermark embedding and detection. Watermark feature domain is extracted based on MBR. In order to grant the fidelity and validity of the watermarked map, we use Douglas-Peuker algorithm to identify the feature vertices so that only non-feature vertices of the map will be modified. Experiments have verified the fidelity of the watermarked map and the robustness of the watermarking algorithm.

References from Journals:

Doncel V.R., Nikolaidis N. and Pitas I., 2007. An optimal detector structure for the fourier descriptors domain watermarking of 2D vector graphics. IEEE Transaction on Visualization and Computer Graphics,13(5), pp.851-863.

References from Other Literature:

Sakamoto, M., Matsuura Y., Takashima Y., 2000. A scheme of digital watermarking for geographical map data. In: Proc. of the Symposium on Cryptography and Information Security, Okinama, Japan, pp.26-28

Kang, H., 2001. A vector watermarking using the generalized square mask. In: Proc. of the International Conference on Information Technology: Coding and Computing, Las Vegas, NV, USA, pp.234-236

Gerrit S., Michael V., 2004. A high capacity watermarking system for digital maps. In: Proceedings of the 2004 workshop on Multimedia and security, September 20-21, 2004, Magdeburg, Germany. pp.180-186

Ohbuchi R., Ueda H., Endoh S., 2002. Robust watermarking of vector digital maps. In: Proceedings of IEEE Conference on Multimedia and Expo 2002 (ICME 2002), Lausanne, Switzerland, pp.1-4

Huang M., Zhou X., Cao JH. et al. 2007. Geo-WDBMS: An Improved DBMS with the Function of Watermarking Geographical Data. In: DASFAA 2007, LNCS 4443, pp.1098–1109.

Solachidis, V., Nikolaidis N., Pitas I., 2000. Watermarking polygonal lines using Fourier descriptors. In: Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing, Istanbul,Turkey, vol.IV, pp.1955-1958

Li, Y. and Xu L., 2003. A blind watermarking of vector graphics images. In: Proc. the Fifth International Conference on Computational Intelligence and Multimedia Applications, pp.27-30.

5. ACKNOWLEDGEMENTS

This research is funded by the Key Technologies Science and Technology Program (Project No.2007BAH12B07)