

Watermarking Geographical Data on Polygon Feature Using Random Table

Zahra Aborawi Aborawi, Atef Zaki Ghalwash, Nahla El_Haggar

Faculty of Computers and Information, Helwan University , Cairo, Egypt
, atef.ghalwash@hq.helwan.edu.eg, ahla_elhaggar@yahoo.com, Zahraaborawi@yahoo.com

Abstract

A novel digital watermarking method is presented, for maps' polygons, for embedding two watermarks in LSB (Least Significant Bit) of an angle and a side of a polygon, using random tables. One of the challenges that faced watermarking technique is that angles and distances are not changed easily in vector data. The proposed method overcomes this challenge since it is robust to scaling. The proposed method doesn't cause any distortion or topological change for maps because of the voting feature in the method. CR (Correspondence Ratio) is used to validate the embedding effectiveness. Recorded test results show that the presented method outperforms methods previously known in the literature. Therefore, the proposed method could be used in various fields to protect the copyright of vector map.

Keywords: *Digital Watermarking, Vector Map Data, GIS, Shape-Preserving, Random Table Angle, Aside*

1. Introduction

Information System (GIS) vector map is a kind of strategic information resource that is broadly used in developing social economy, national defense, and commercial applications. Copyright protection of geospatial data has become a vital issue in the community of geographic information science. Digital watermarking is one of the means which used to protect the copyright of the data.

Geographical data consists of the geographical objects which represent various objects in the real world. Geographical objects can be classified into three basic geometrical elements which are points, polylines and polygons [1].

Digital maps can be categorized into two categories: raster digital maps and vector digital maps [2]. Raster images depend on the resolution of the screen or the display device. They use small square grids, which are called pixels, to express images. Each pixel has its own location and color value. So during processing images, the editions occur over groups of pixels rather than objects or shapes. After scaling an image from this type it will produce saw tooth wave and distortion. The two main advantages of vector representation are: small file size and no saw tooth will appear if it is zoomed in the map. These are important reasons beyond why vector representation method is adopted in digital maps.

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 make 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.

Every watermarking system consists of at least two different units: A watermark embedding unit and a watermark detection/extraction unit, as shown in figures1 & 2. In this paper a new digital watermarking method is presented for maps' polygons by embedding two watermarks in LSB (Least Significant Bit) of an angle and a side of polygon using random table. The paper is organized as follows: the related work is covered in chapter 2. Chapter 3 contains the presented method for watermarking. Chapter 4 explains the different experiments and the recorded results. Finally, chapter 5 has the conclusion.

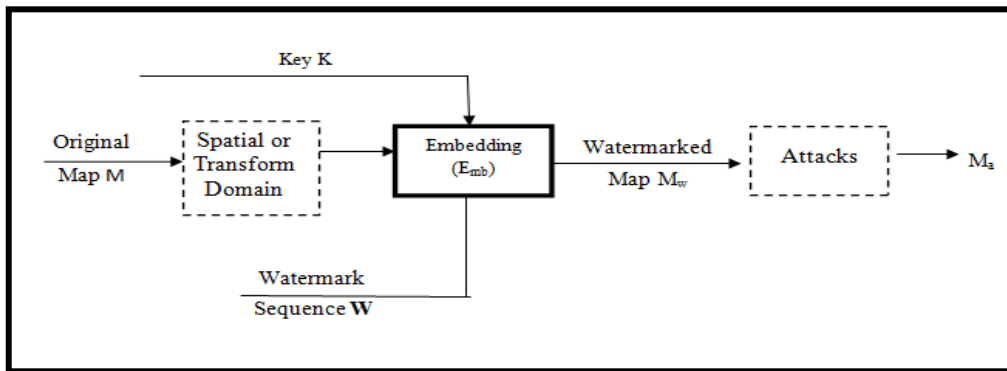


Figure 1: Embedding process

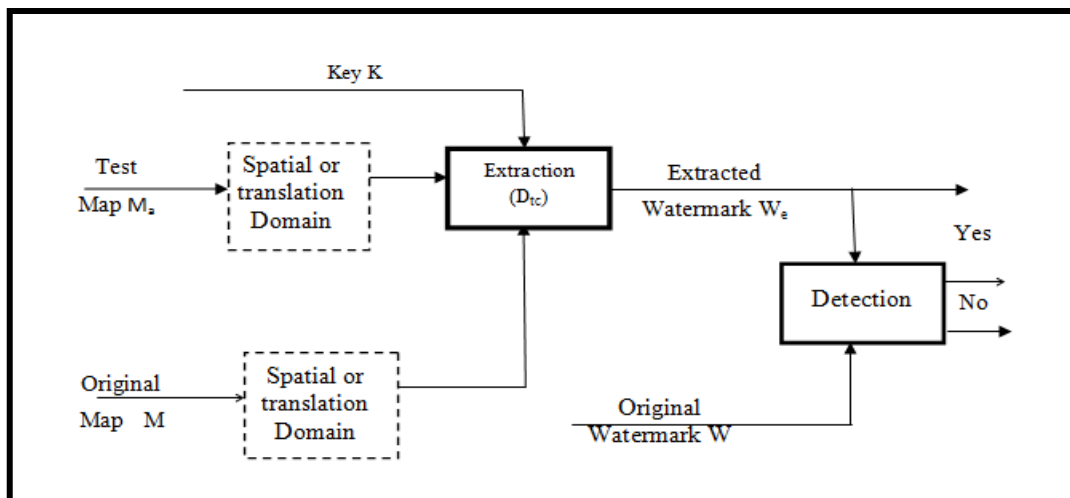


Figure 2: Extraction / Detection process

2. Related Work

In recent years, the study of digital watermarking technology achieves a great progress. Watermarking techniques are divided into spatial domain techniques and transform domain techniques [3]. They are presented as follows:

2.1 Watermarking techniques in spatial domain

One spatial domain watermarking scheme embeds watermark by moving the vertices or modifying vertices' coordinates. However in transform domain, it is not easy to control the distortion after embedding watermarks. Spatial domain can be a good method in vector map data because the distortion and position accuracy are very important in map field [10].

In [4] researchers developed a watermarking system based on modifying the location relationship of vertices, but the system was fragile to noise attack. In [5], they used a square mask of arbitrary sizes rather than of a fixed size. A disadvantage of the method was the selection of an optimal mask size in order to preserve the vector topology for the vector map. In [6] researchers designed a method for digital watermarking using angles. This method suffered from the incapability to extract watermarks if the interior angles were changed. The researchers in [7] used the spatial topological relation between polygons as a cover data, and watermark was embedded by slightly modifying the selected LSB (Least Significant Bit) of the metric measure of the spatial topological relation. The algorithm was not sufficiently robust to cropping attack. The authors in [8] presented a blind watermarking algorithm technique for watermarking 2D vector map. Watermarks were embedded in a spatial topological relation by slightly modifying the metric measure of the spatial topological relation. The algorithm was not sufficiently robust to cropping and simplification attack. In [9] authors designed a watermarking algorithm and shape-preserving for vector maps consisting of polygons was suggested. They exploited highest significant bits of the coordinates of the centers of polygons to identify polygons, selected the mean feature distance as cover data, and embedded watermark using quantization modulation. In [10] authors suggested a novel algorithm to watermark the building layers which took into account the right angles and avoided the dependence on the vertices density by modulating the watermark sequence into the objects' spatial relations, It also had its own weakness since the schema was relatively weak to data cropping attacks. In [11] the authors recommended a digital watermarking algorithm for vector map, which was based on data configuration of vector map, combined with the error correction theory in the field of digital communication, this technique also wasn't robust against cropping.

2.2 Watermarking techniques in transform domain

The transform domain technique is used to perform some kind of transformation to the coordinates, and to embed the watermark information into the transformed coefficients. The typical schemes are implemented in following domains: DFT (Discrete fourier transform) domain, DWT (Discrete wavelet transform) domain and DCT (Discrete cosine transform) domain.

In [12] authors designed DFT (Discrete Fourier Transform) blind watermarking model. The vertex sequence that extracted from vector map was carried on DFT transform. This technique set up a watermarking scheme to embed the watermark into the phase of DFT through quantification. The algorithm wasn't robust against deleting vertices, clipping, and scaling. In [13] researchers presented a watermarking algorithm for copyright protection of geographical maps composed in vector format. A watermark was embedded in the low frequency coefficients of wavelet transform (WT). It does not include improving. However, it didn't improve robustness for some types of attacks like cropping and map simplification vector data. The researchers in [14] proposed an algorithm, the line and surface coordinate points were decomposed by integer wavelet transform. The watermark was embedded in the low frequency coefficients of integer wavelet transform. It was a non-blind system since the original map was needed for watermark detection. Researchers in [15, 16] recommended vector maps for a new watermarking algorithm that used the Minimum Encasing Rectangle of the map. Spatial data was transformed into frequency domain with DCT. Watermarks were embedded in DCT coefficients with middle-frequency. By slightly modifying the distribution of vertices, a watermarked map was generated. Watermark synchronization was achieved by introducing a MER. However, a vector map may have several MERs.

3. The Proposed Watermarking Method

3.1 Overview

The idea of the method is watermarking the polygons vector map by embedding two watermarks in each feature. The first watermark is embedded in the angle by slightly changing the two corners coordinates of the angle. The second watermark is embedded in one side by slightly changing the two coordinates that links between two points in the polygon. The distance watermark is embedded in the length between two points; these two points are not affected by the angle watermark. So, the distance watermarks are embedded only on the polygons which have at least 5 points (6 points are checked because the polygon is a closed feature, the last coordinates is the same as the starting coordinates). The watermark is embedded in the selected LSB (Least Significant Bit). Both watermarks are generated using a random table which is considered as watermarks key, see figure 3.

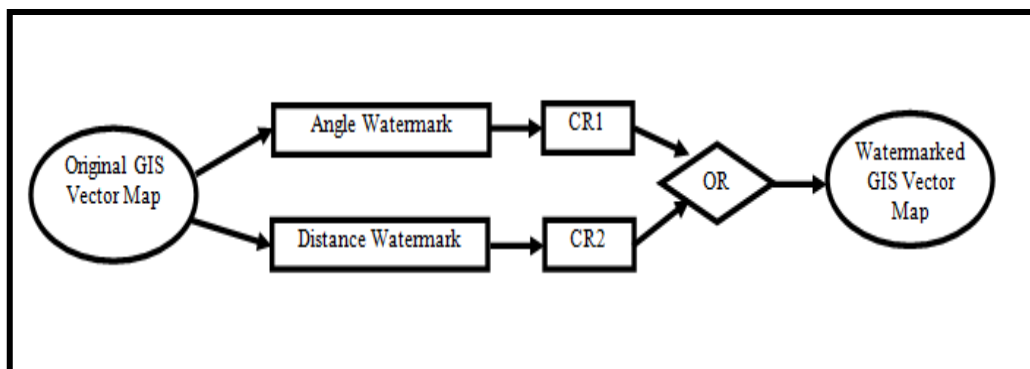


Figure (3): Proposed watermarking method

3.2 Calculating the interior angle

The angle has been calculated using the consecutive three points and the cosine rule as in figure 4. The interior angle at the starting point of the polygon is used. Therefore, one angle per object can be obtained.

$$a^2 = b^2 + c^2 - 2bc \cdot \cos A \tag{1}$$

Where A is the angle to be calculated, a is the side of the triangle that facing the angle A, b and c are the other two sides of the triangle.

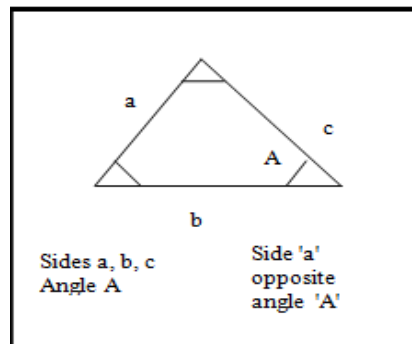


Figure 4: Calculating angles using cosine rule

Since the Cosine rule calculates the smallest angle and the interior angle may be the complementary angle as shown in Figure 5, so it is needed to determine if the calculated angle is the correct interior angle or not in the following steps:

- Calculating the angle Θ using the cosine rules.
- Calculate the center point (M) of the side facing the angle Θ
- Create a side that connects between the point M and the head of the angle point A. Then get the center point of this side (T).
- Check if the center point (T) lies inside polygon or not. If (T) lies inside polygon, then angle Θ is the interior. Otherwise, the interior angle is the complementary angle for the angle Θ .

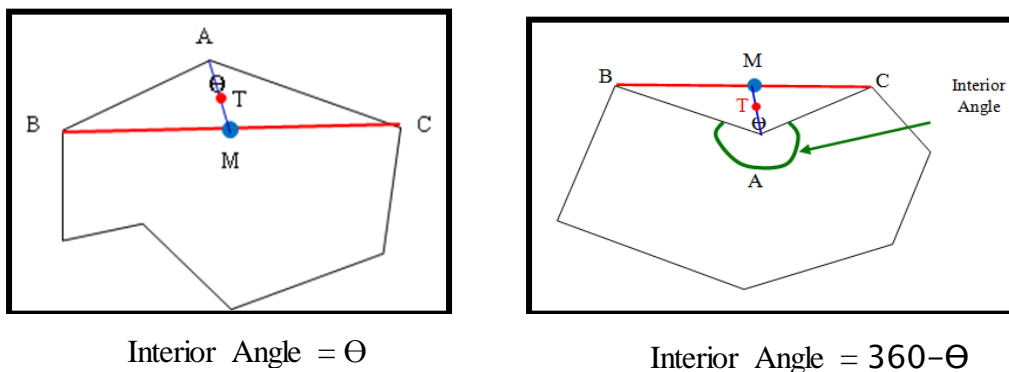


Figure5: alculating Polygon's Interior Angle

3.3 Calculating Distance

After selecting an angle e.g at point A for watermarking (see figure 6). the distance between two points (not on the angle sides) e.g, points D & E in the polygon feature can be calculated as follows:

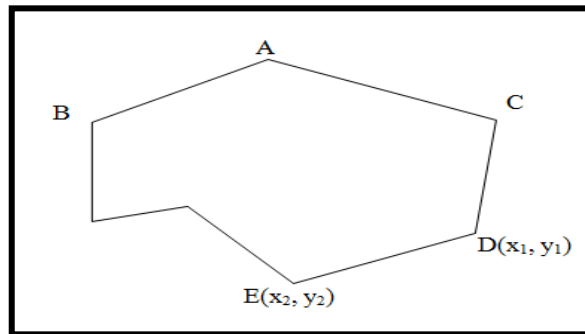


Figure 6: Calculating Distance

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2}$$

Where (x_1, y_1) and (x_2, y_2) are the coordinates of the points D & E in the polygon feature, respectively.

3.4 User Input Keys

User input keys are used to generate the user’s own value, $G(x)$ for user x , during watermarking. The user’s own value, $G(x)$ enhances the watermarking security since $G(x)$ is an essential value when watermarks are embedded and extracted. As an example, if the keys for user x are $k1:Z$, $k2:H$, $k3:R$, then the user’s own value $G(x)$ is the sum of the ASCII value for each key i.e, $G(x) = \text{ASCII}(Z) + \text{ASCII}(H) + \text{ASCII}(R)$, which is $G(x) = 89+71+81 = 241$.

3.5 Random Table

The random table is generated by generating a random number between 0, 20 for every number between 0 & 9 as shown in table 1. The random table can vary and is used as a watermark key that prevent attempting to damage the watermark. The random table is applied to the vector map data, after calculating an angle and a distance between two points for each object and is used for hiding and extracting the original data during watermarking and extracting the watermark, respectively.

Table 1: Random Table

Original Value	0	1	2	3	4	5	6	7	8	9
Changed Value	3	6	17	19	2	11	14	5	0	18

3.6 Intersection test

The vector map data topology is very important, so that a good watermarking algorithm must preserve shape topology. An intersection test is used to check if there is a change in topology before embedding watermark to be sure that the watermarks will preserve the topology. The intersection test checks the intersection between each polygon's sides and all other polygons' sides in the map and returns three values: (Not Intersected: if the two sides are not intersected), (Intersected: if there is an intersection between two sides), (Include End Point: if the two sides are intersected and the intersection point is the side end point) [6], see figure 7. Comparing these values before and after embedding watermarks, the vector map data topology is preserved.

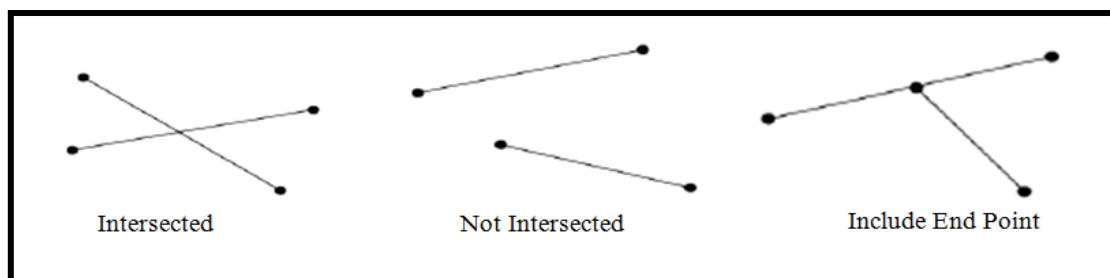


Figure 7: Intersection Test

3.7 Watermark Generation

A watermark is generated for user x , using the random table, for an angle (or a distance between two points) in an object. Using the random table, the corresponding value, $P(x)$, for the integer part of the calculated angle (or the distance bet. Two points) is extracted from the table. The watermark is then calculated using the formula in (3)

$$\text{Watermark} = G(x) - P(x) \% G(x) \tag{3}$$

$G(x)$ is the user's own value. For example, if the angle or the distance value is 96.8104892, the corresponding value $P(x)$ of the integer part 96 equals to 1814 (see table 1, where 9=18 & 6=14). Assume that user input keys are ZHR (as in section 3.4), the value $G(x) = 241$. Therefore, the watermark is calculated from formula (3). $\text{Watermark} = 241 - 1814 \% 241 = 241 - 127 = 114$.

3.8 Embedding the Watermark

To embed the watermark, the LSB (least significant bit) of the angle or the distance value is altered. For an angle Θ , ξ least significant bit is defined as ξ th digit after decimal point of the angle Θ , denoted as $\text{LSB}_\xi(\Theta)$ [7]. For example, 1st, 2nd, 5th least significant bit of decimal number 0.8104892 is 8, 1, 8, respectively. Considering the 5th LSB, the angle is slightly modified by replacing the 5th LSB by the watermark value. So, the watermarked angle is 96.8104114. This watermarked angle causes the coordinates of the angle sides to be changed where the difference between the original angle and the watermarked angle is bisected.

Change = watermarked angle – original angle = 96.8104114 - 96.8104892 = -0.0000778
 Bisected change = change/2 = -0.0000778/2 = -0.0000389 = α

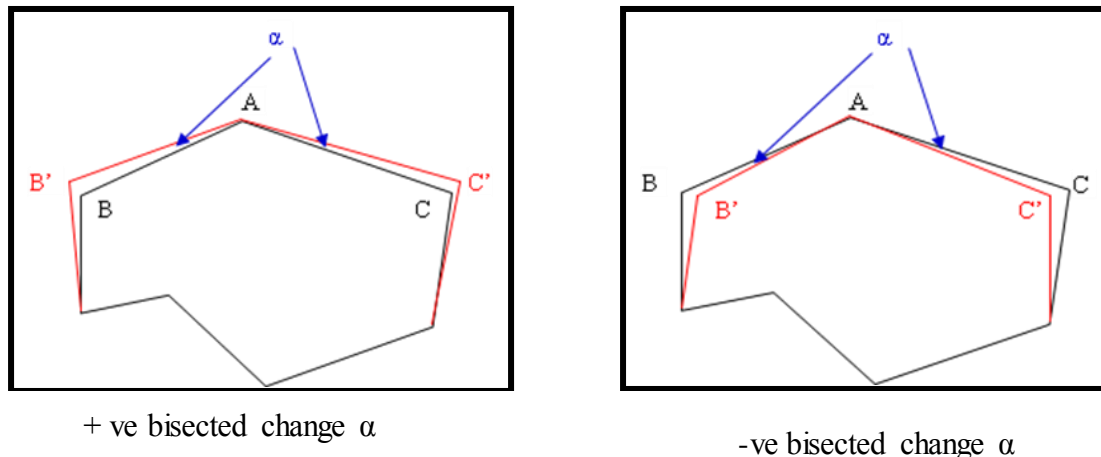


Figure (8): Embedding Angle Watermark

The modified coordinates can be calculated using formula (4). Figure 8 is example of positive and negative bisected change values [6] and the new coordinate B'(x,y) for point B(x,y) can be calculated as follows:

$$\begin{aligned} B'x &= Ax + (Bx - Ax) * \cos \alpha - (Bx - Ax) * \sin \alpha \\ B'y &= Ay + (By - Ay) * \sin \alpha + (By - Ay) * \cos \alpha \end{aligned} \quad (4)$$

Where Bx,By and Ax,Ay are the original coordinates, B'x, B'y and A'x,A'y are the watermarked coordinates, And α is the bisected change in angle (change/2). Similarly, C'(x,y) can be calculated.

The distance watermark is embedded in a side of the polygon. The two endpoints of the chosen side should be different from the two end points of the sides of the watermarked angle, e.g., for the polygon in figure 8, any side that does not contain A, B, or C as endpoints could be selected. As a result, we could embed a distance watermark if a polygon contains more than 5 points. The distance watermark is embedded by displacing the two endpoints of a side using the formula (5) [17].

The slope m for the side with end points E(x,y) & D(x,y) is given by:

$$m = (Ey - Dy) / (Ex - Dx)$$

The coordinates of the watermarked point, D'(x,y), of the point D(x,y), is given by:

$$\begin{aligned} D'x &= Dx \pm (d1 / \sqrt{1 + m^2}) \\ D'y &= Dy \pm m * (d1 / \sqrt{1 + m^2}) \end{aligned} \quad (5)$$

Where, d1 is the bisected change (change/2) in the distance between the two points E & D. and (- or +) depends on whether the watermarked point moves to the left or to the right of the original point .

The embedding of both angle and distance watermarks is shown in Figure 9. Angle watermark is embedded by changing the coordinates B, C to B', C'. The distance watermark is embedded by changing the coordinates D, E to D', E'.

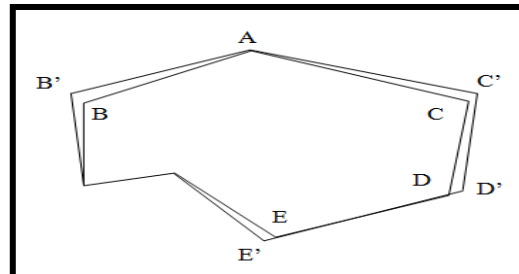


Figure 9: Embedding Angle & Distance Watermarks

The pseudo-code for embedding the watermarks is given in Figure 10:

```

Watermarking Embedding Algorithm
Input:
The original map: I
User input keys: k1, k2, k3
Least Significant Bit (LSB): ζ
Output:
The watermarked map: I'

Generate Random Table (R)
G(x) = Compute_user_ownvalue(k1,k2,k3)
For each polygon P in map I
a2=b2+c2-2bc*cosA //Compute interior angle at starting point using cosine rule, Where A is the angle to
//be calculated, a is the side of the triangle that facing the angle A, b and c are the other two sides of the triangle.
P(x) = compute_changed(Θ,R) //Change computed angle to watermarked angle using
//Random Table
Watermark = G(x) - P(x) %G(x)
Θ' = embed_watermark(Θ, Watermark, ζ) //replace the LSB (ζ) by the watermark
Change = Θ' - Θ //change in angle (watermarked angle - original angle)
α = change/2 // Bisected change
B'x = Ax + (Bx - Ax) * Cos α - (Bx - Ax) * Sin α
B'y = Ay + (By - Ay) * Sin α + (By - Ay) * Cos α //Where Bx, By and Ax, Ay are the original coordinates,
//B'x, B'y and A'x, A'y are the watermarked coordinates, And α is the bisected change in angle (change/2).
//Similarly, C'(x, y) can be calculated
If #Of points > 5
d = √((x2 - x1)² + (y2 - y1)² //Calculate the length between two points; these two points are
//not affected by the angle watermark
D(x) = compute_changed(d,R) //Change computed distance to watermarked
//distance using Random Table
Watermark1 = G(x) - D(x) %G(x)
d' = embed_watermark(d, Watermark1, ζ) //replace the LSB (ζ) by the
// watermark1
change1 = d' - d //change in distance (watermarked distance - original distance)
d1 = change1/2 // Bisected change
m = (E y - D y) / (E x - D x) // Where m is The slope for the side with end points E(x, y) & D(x, y).
If d' is to right of d //If new coordinates is to right side of original point
D'x = D x - (d1 / sqrt(1 + m²)) // Where D'(x, y) coordinates of the watermarked point, of the point
//D(x, y). d1 is the bisected change (change/2) in the distance between the two points E & D
D'y = D y - m * (d1 / sqrt(1 + m²)) //D'x, D'y are the watermarked coordinate
endif
Else If d' is to left of d //If new coordinates is to left side of original point
D'x = D x + (d1 / sqrt(1 + m²))
D'y = D y + m * (d1 / sqrt(1 + m²))
endif
endif
endfor
    
```

Figure 10: Embedding Watermarks pseudo-code

3.9 Extracting Watermark

The proposed watermark extraction process is similar to the embedding process. First the angles and distances are calculated in the watermarked data then new values are obtained using the random table. The watermarks are calculated using the user's own value, P(x), and G(x). Finally check if the calculated watermarks are reflected in the calculated angles and distances (check if the data are watermarked). CR [6] (Correspondence Ratio) and RMSE [5] (Root Mean Square Error between the map coordinates of the original and the watermarked map) are used to validate the embedding effectiveness as shown in Formula (6), (7).

$$RMSE = \sqrt{\sum \frac{(V_{xy} - V_{xy}^d)^2}{(V_{xy})^2}} \quad (6)$$

Where RMSE is the Root Mean Square Error, $(V_{x,y})$ is the original coordinates and $(V_{x,y}^d)$ is the Watermarked coordinates.

$$CR = \frac{\text{the number of the extracted watermark}}{\text{the number of the objects}} \quad (7)$$

CR1 is computed for the angles watermarks as the number of extracted angles watermarks divided by the total number of map features. CR2 is computed for distances' watermarks as the number of extracted distances' watermarks divided by the total number of embedded distances' watermarks. CR3 is calculated by applying OR operation between CR1 and CR2.

The pseudo-code for detecting the watermarks is given in Figure 11:

Watermarking Detection Algorithm	
Input:	The watermarked map: I' Least Significant Bit (LSB): ζ
Output:	Correspondence Ratios: CR1, CR2, CR3
<pre> Retrieve Random Table (R) G(x)=retrieve_user_ownvalue (k1,k2,k3) For each polygon P in map I' a2=b2+c2-2bc*cosA //Compute interior angle at starting point using cosine rule, Where A is the //angle to be calculated, a is the side of the triangle that facing the angle A, b and c are the other //two sides of the triangle. P(x)=compute_changed(θ,R) //Change computed angle to watermarked angle using //Random Table Watermark = G(x) - P(x) %G(x) Ext_WM_Angle = extract_watermark(θ, ζ) If Watermark = Ext_WM_Angle ++detected_angles Flag1 = true endif endif If #Of points > 5 $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ //Calculate length of polygon's side D(x)=compute_changed(d,R) //Change computed distance to watermarked //distance using Random Table Watermark1 = G(x) - D(x) %G(x) Ext_WM_Distance = extract_watermark(d, ζ) If Watermark = Ext_WM_Distance ++detected_distances Flag2 = true endif endif Flag3 = Flag1 Flag2 If Flag3 = true ++detected_total endif CR1 = detected_angles / total #Of objects CR2 = detected_distances / total #Of objects CR3 = detected_total / total #Of objects Endfor </pre>	

Figure 11: Extracting Watermark pseudo-code

4. Experiments & Results

The presented watermarking algorithm is tested on a polygons landuse vector map for Heliopolis district in Cairo, Egypt. The features of the map are as follows: Number of polygons = 923 polygons, Projection: Geographic Lat/Lon, Extents: min x, min y = (31.3112, 30.0807), max x, max y = (31.3584, 30.106).

PostgreSQL/ PostGIS are used to store and manage geographical data, and SharpMap is used to visualize data. The algorithm is implemented in C#. The original & watermarked data are shown in Figure 12. The initial watermark parameters are set as follows: user input keys k1:Z, k2:H, k3:R, LSB: 5th digit is selected to insert the watermark. The intersection test shows that the watermarking algorithm preserves topology. The algorithm shows that watermarking is more robust against attacks. The recorded RMSE value is 1.99E-09 which is too small, so that the fidelity does not change. The embedding effectiveness CR found to be 100%.

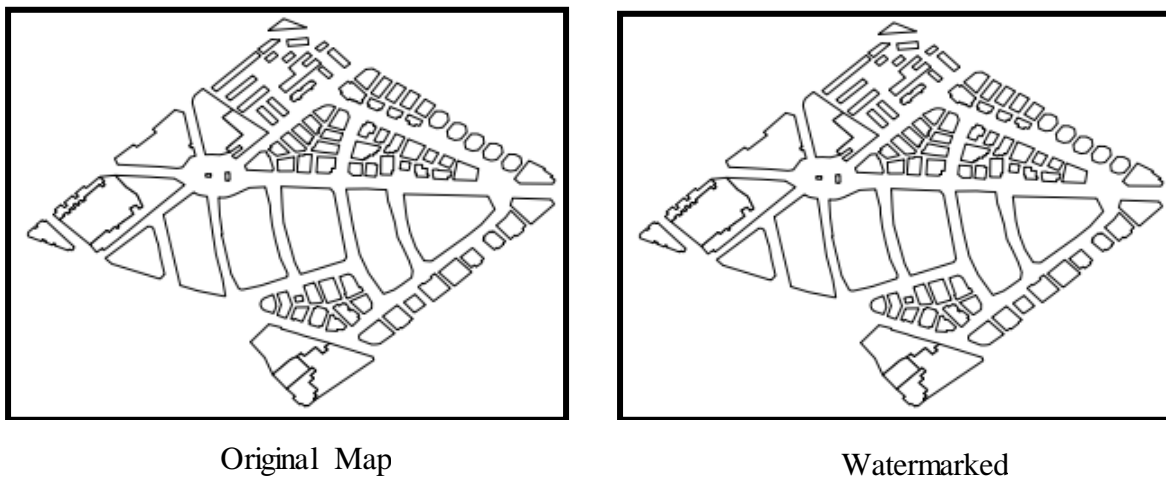


Figure 12: Original & Watermarked data

The GIS vector data attacks are classified into 3 types: geometrical attacks, vertex attacks, and noise attacks [3]. All attacks' types are tested to check the robustness. The algorithm found to be robust against geometrical attacks (translation, scaling, and rotation). The simplification test, based on Douglas-Peucker algorithm with different simplification distances and different LSB, as shown in Figure 13, recorded satisfactory results since the presented method cannot extract all embedded watermarks because some watermarks are removed with simplification. Meanwhile, the value of CR3 remains high. The presented algorithm is also robust against vertex interpolation with different LSBs as shown in Figure 14. Adding new features to the map that do not contain watermarks reduce CR3, as shown in Figure 15. The algorithm is also robust against moving vertex, noise attacks, and cropping see figure 16.

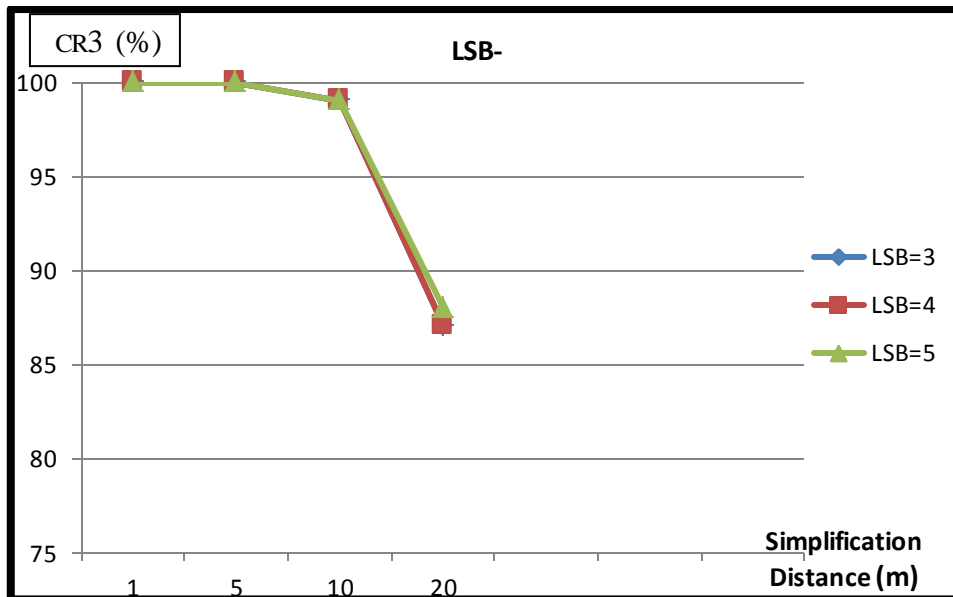


Figure 13: Simplification Robustness Diagram

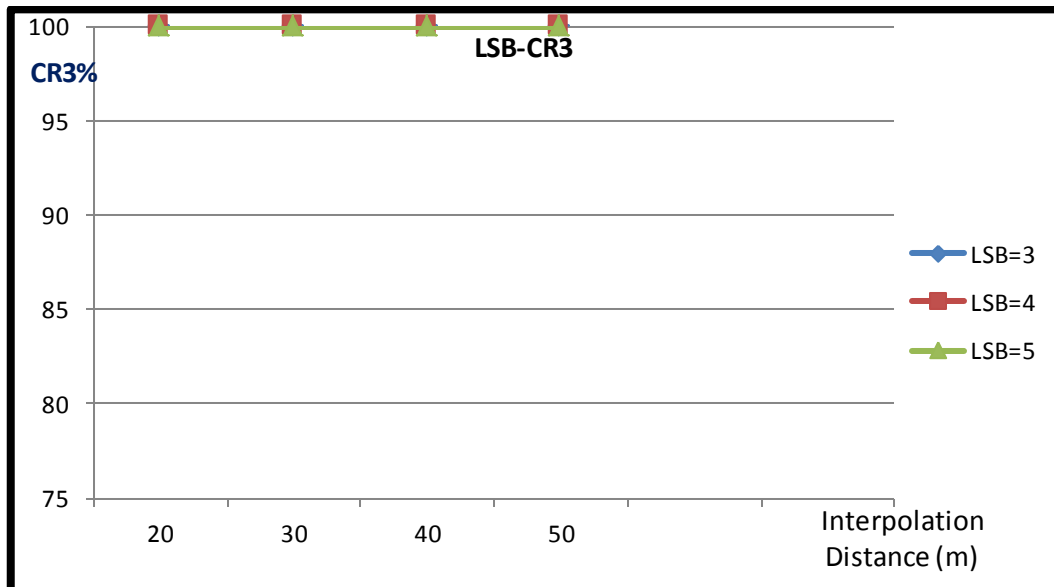


Figure 14: Interpolation Robustness Diagram

Where the X-ax represents the simplification and interpolation distances and its unit is meter (m) and Y-ax represents the CR3 and it is unit is percentage.



Figure 15: Adding New Features

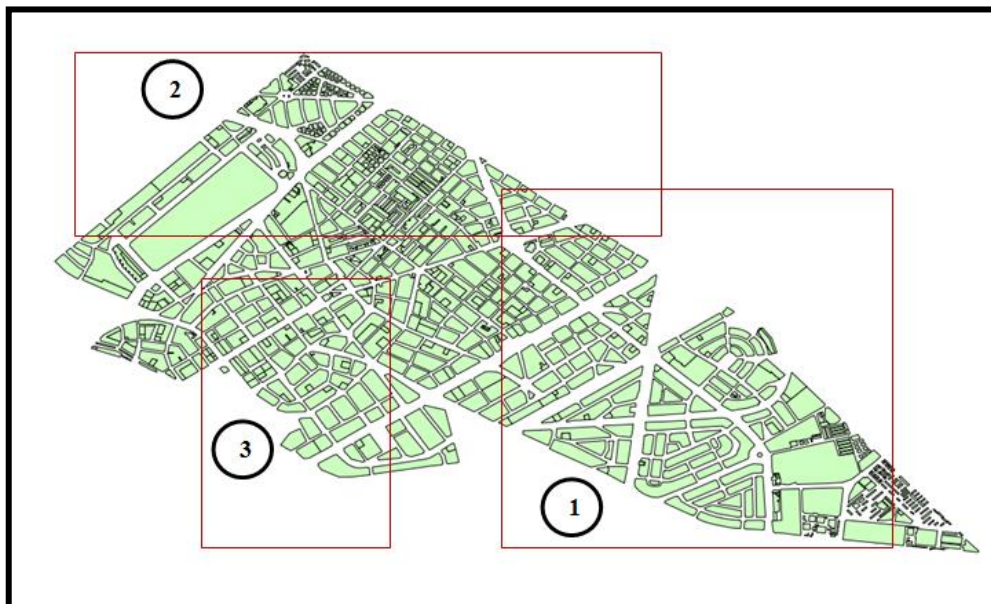


Figure 16: Cropping

Where the polygons in green color are the original map's polygon and the polygons in yellow color are the new added polygons to the map.

5. A Comparative Study

The proposed algorithm is tested a set of C# classes and functions are implemented by calling some functions from PostGIS to test the algorithm robustness to different types of attacks with different input parameters (e.g. Simplification, Interpolation, Rotation, Translation, Scaling and noise attacks). Also Quantum GIS and Map Window GIS (an open source GIS software) are used to view and test the watermarked maps. The recorded results showed that the proposed algorithm has a good fidelity and is robust against all types of attacks and is superior to J. Kim [6] watermarking algorithm. as shown in Table 2.

Table 2: Results of the Comparative study

Attack Type	Attack Value(range)	The proposed Method	J. Kim Method
Translation			
	1°	100%	100%
	10°	100%	100%
	100°	100%	100%
Scaling			
	Scaling Factor 0.5 (shrink 50%)	100%	100%
	0.8 (shrink 20%)	100%	100%
	1.2 (expand 20%)	100%	100%
	1.5 (expand 50%)	100%	100%
Simplification		100%	85%
	1	100%	97%
	5	99%	84%
	10		
Adding object	//		95%
	Adding 13 features to the Map	99%	//
	Adding 32 features to the Map	97%	//
	Adding 49 features to the Map	96%	//
By Rotation			
	0.1°	100%	96%
	0.01°	100%	99%
	0.001°	100%	100%
By translation			
	0.1	100%	100%
	0.01	100%	100%
	0.01	100%	100%
Adding noise per rea			
	First	100%	100%
	Second	100%	100%
Cropping			
	Cropping 35.6% of the Map	100%	//
	Cropping 62.9% of the Map	100%	//
	Cropping 83.7% of the Map	100%	//
Moving Vertexes			
	Moving vertexes of 17 features	100%	No
	Moving vertexes of 28 features	100%	No
	Moving vertexes of 53 features	100%	No
Interpolation			
	20m (9752 vertexes interpolated to 16127)	99%	//
	30m (9752 vertexes interpolated to 12955)	100%	//
	40m (9752 vertexes interpolated to 11612)	100%	//
	50m (9752 vertexes interpolated to 10928)	100%	//

The term '//' in the table 2 means that the corresponding ratio is not mentioned by authors.

The term 'No' in table 2 means that the algorithm was not tested against this type of attack .

The proposed algorithm improves the J. Kim [6] algorithm in robustness against moving vertex attack which may damage either the angle or the distance watermark. At the time, the recorded results, shown in table3, shows that the proposed algorithm improves the C. Wang [7] algorithm in robustness against cropping.

Table 3: A Comparative Study with Previous Work

Algorithm	Domain	Blind	Fidelity			Robustness							
			Shape preserving	Error	Topology	geometric attacks			Simplification	Vertex interpolation	Moving Vertex	Noise Addition	Cropping
						Rotation	Scaling	Translation					
J. Kim	Spatial	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
C.WANG	Spatial	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No
The Proposed Method	Spatial	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

6. Conclusion

A novel digital watermarking algorithm has been presented for polygons vector map using angles and sides to improve previously mentioned algorithms: algorithms that were not robust in either changing angles (Moving Vertex) or against cropping. The proposed algorithm embedded two watermarks (an angle & a side) per object which enhanced the ability to extract either watermark when the other was corrupted as a result of different types of watermarking attacks. Random table values and keys were used to enhance watermarking security. A feature of the proposed algorithm is that the same method was used for watermarking both the angles and polygon sides. The watermarks were embedded on a LSB (Least Significant Bit) in both the angle and the side values which improved the RMSE.

In this paper, watermarks were embedded into a spatial domain. Topology of the map data was preserved through the intersection test. A feature of the proposed method is blind watermarking, which extracted watermarks using a method similar to that was used to embed them. The experimental results showed that this method has fidelity that is vital to map data. Moreover, the watermarking was robust to attacks.

References

- [1] W. Baiyana, W. Wei, and M. Dandan , “2D vector map watermarking based on the spatial relations”, *Int. Conf. on Earth Observation Data Processing and Analysis (ICEODPA)*, Vol. 7285, 728532, 2008.
- [2] L. Zheng , Y. Jia, and Q. Wang, "Research on vector map digital watermarking technology", First Int. Workshop on Education Technology and Comput. Sci., 2009.
- [3] X. Niu, C. Yong Shao, and X. Wang, “A survey of digital vector map watermarking” *ICIC Int.*, Dec. 2006.
- [4] M. Sakamoto, Y. Matsuura, and Y. Takashima, “A scheme of digital Watermarking for geographical map data”, *Symposium on Cryptography and Inform. Security*, 2000.
- [5] K. Hwan ,K. Kab, C. Jong," A vector watermarking using the generalized square mask", Proc. of the International Proc. International Symposium on Information Technology: Coding and Computing, 234–236.2001
- [6] J. Kim, “Robust vector digital watermarking using angles and a random table,” *Advances in Inform. Sci. and Service Sci.* vol. 2, no 4, Dec. 2010
- [7] C. Wang ,Z. Peng ,Y. Peng ,L.Yu , J. Wang, Q. Zhao “Watermarking geographical data on spatial topological relations,” *Springer Sci+Business Media, LLC*, 2010.
- [8] C. Wang, Z. Peng, Y. Peng, and L. Yu, “Watermarking 2D vector maps on spatial topology domain," in *Proc. IEEE Int. Conf. Multimedia Inform. Networking and Security*, Wuhan, China, 2009, pp.71-74.
- [9] C. Wang, Q. Zhao, F. Zhong,” A Shape-preserving and robust watermarking algorithm for vector maps” *Int. Conf. Computational and Inform. Sci., IEEE*, 2010.
- [10] W. Baiyan, W. Wei, P. Zhiyong, and D. Daosheng, “A new algorithm for watermarking building polygons”, *Int. Conf. Digital Image Process. IEEE*, 2009.
- [11] F. Cheng, H. Yin, X. Zhang, and D. Zhang, “A Digital Watermarking Algorithm for Vector Map”, *Int. Conf. on Challenges in Environmental Sci. and Comput. Eng., IEEE*, 2010.
- [12] S. Tao, X. Dehe, L. Chengming, and S. Jianguo, “Watermarking GIS data for digital map copyright protection”, *In Proc. 24th Int. Cartographic Conf., Santiago, Chile*, Nov. 15-21, 2009, pp.1-9.
- [13] S. Chaudhari and P. Venkatachalam, “Robust watermarking for protection of geospatial data”, *IPCSIT*, vol. 29, 2012.
- [14] C. Zhu, C. Yang, and Q. Wang, “A watermarking algorithm for vector geo-spatial data based on integer wavelet transform”, *the Int. Archives of the Photogrammetry, Remote Sensing and Spatial Inform. Sci.*, vol.37. Part B4. Beijing, 2008.
- [15] C. Wang, L. Zhang , B. Liang, H. Zheng, W. Du, and Y. Peng, “Watermarking Vector Maps Based on Minimum Encasing Rectangle”, *Fourth Int. Conf. Intelligent Computation Technology and Automation, IEEE*, 2011.
- [16] B. Liang, J. Rong, and C. Wang, “A vector maps watermarking algorithm based on DCT domain,” Shihezi Univ., College Inform. Sci. Technology, 2009.
- [17] <http://www.freemathhelp.com/forum/threads/70424-Find-coordinates-given-original-point-and-slope>, accessed on 3rd Jan. 2013.