2D VECTOR MAP REVERSIBLE DATA HIDING WITH TOPOLOGICAL RELATION PRESERVATION

Nana Wang, *Xiangjun Zhao*

School of Computer Science and Technology, Jiangsu Normal University, Xuzhou 221116, P. R. China

ABSTRACT

Topological relation preservation is an important issue for 2D vector map reversible data hiding methods. Using the idea of controllable perturbation region (CPR), we propose a method that preserves topological relations between map objects. In particular, we propose calculating the CPR for each vertex and each line segment, selecting *eligible* vertices according to the CPRs of the vertex-line segment pairs, and embedding data into the *eligible* vertices in a reversible manner. Since the embedding operation on an *eligible* vertex does not change its relation with any line segment, topological relations between map objects can be preserved. Moreover, the calculation of CPRs not only ensures correct data recovery, but also provides good invisibility. We provide experimental results to demonstrate the effectiveness of our method.

Index Terms— Reversible data hiding, topological relation preservation, 2D vector map

1. INTRODUCTION

A 2D vector map is normally composed of independent map objects (e.g., points, polylines and polygons), which accurately represent geographical objects in the real world. Due to high accuracy requirement for 2D vector maps in some applications, such as military and geological exploration, reversible data hiding has been introduced to this kind of data for transmitting secrets about the host vector map itself (e.g., authentication messages and metadata) [1].

Generally, hiding data in a vector map will distort the coordinates of the map. Although the distortion can be removed upon data extraction in reversible algorithms, it should be unnoticeable on transfer security grounds. This not only requires the embedding distortion to be below the precision tolerance, but also means the spatial relations (i.e., distance, directional and topological relations) between map objects should be preserved [2-5]. This paper focuses on preserving the topological relations.

Topological relations describe the geometric relations between map objects that are invariant under affine transformations such as rotation, scaling, and translation. For instance, if *a* contains *b*, *a* remains to contain *b* if the vector map is rotated. Data hiding may modify the topological relations between map objects. For example, *a* and *b* are disjoint before data hiding (Fig. 1(a)), while after data hiding, the embedded *a* and the embedded *b* are intersected (Fig. 1(b)). Changes of topological relations may attract malicious attacks on the vector map or even lead to the acquisition of the camouflaged data.

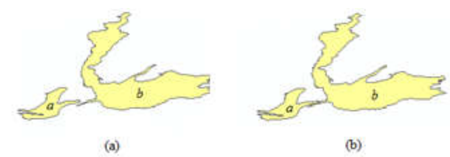

Fig. 1. Topological relation between two polygons before and after data hiding.

There have been some 2D vector map reversible data hiding methods proposed (Table 1). The first article on this topic was published in 2004 by Voigt *et al*. [6]. They embedded data by modifying the integer discrete cosine transform (DCT) coefficients of the map coordinates. As this algorithm is realized in the transform domain, it is difficult to control the embedding distortion. Based on the idea of difference expansion (DE) [7], Wang *et al*. [8] presented two methods: one takes the differences between coordinate pairs and the other adopts the Manhattan distance between neighboring vertices as the cover data. The distortion can be controlled, but the capacity is not very high. After that, some improvements [9-11] have been made to acquire higher capacity. Also aiming at providing high capacity, Cao *et al*. [12] have introduced a method based on iterative embedding. Furthermore, several QIM based methods which provide embedding distortion control ability and high capacity have been presented in [3, 13-15]. However, these algorithms cannot preserve the topological relations between map objects. This drawback is also shared by the afore-mentioned algorithms [6, 8-12]. A CAD engineering graphics, for which several reversible data hiding methods have been presented [16-20], may be considered to be content similar to a CAD drawing. But topological relations between objects cannot be preserved in these methods, either.

In this paper, we propose a topological relation preservation method based on controllable perturbation region (CPR). This method can be integrated with several existing quantization based 2D vector graphics reversible data hiding algorithms [3, 13-18]. In particular, for a vector map to be embedded, the CPR of each vertex and the CPR of each line segment are first calculated. Then, according to the CPRs of the vertex-line segment pairs, the vertices are divided into two types: *eligible* and *non-eligible*. After that, *eligible* vertices are selected to hide data. While the selection of *eligible* vertices ensures topological relation preservation, the calculation of CPRs guarantees correct data recovery and good invisibility.

Table 1 Distortion control and topological relation preservation ability of some 2D graphics reversible data hiding methods.

The work	Embedding method	Distortion control	Topological relation preservation
[6]	DCT coefficient modification	no	no
[8-11]	DE	yes	no
[12]	Iterative embedding	no	no
[3, 13-15]	QIM	yes	no
[16]	Improved QIM and improved DE	no	no
[17]	Iterative embedding and QIM	yes	no
[18]	Improved QIM	no	no
[19]	Improved histogram shifting	no	no
[20]	Improved DE	no	no

2. REVERSIBLE DATA HIDING WITH TOPOLOGICAL RELATION PRESERVATION

2.1 Topological relation preservation

Let us define a polyline as a sequence of adjacent line segments, each of which contains two connected vertices (Fig. 2(a)). If the first vertex coincides with the last one, this polyline will be a polygon (Fig. 2(b)). Our method guarantees that no relation between any vertex and any line segment will change due to data hiding. Hence, the relation between any two line segments will remain the same, and the topological relations between map objects are preserved.

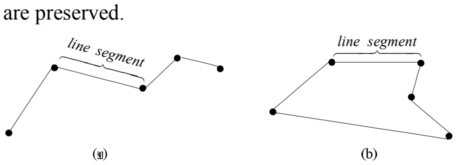

Fig. 2. (a) a polyline with four line segments, and (b) a polygon with five line segments.

Let v_i (x_i , y_i) be a vertex, and $S_{j,k}(v_j, v_k)$ be a line segment with v_j and v_k being its two endpoints. There are four types of relations between v_i and $S_{j,k}$ (Fig. 3) :

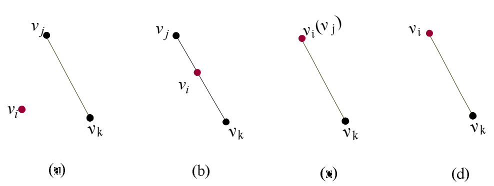

Fig. 3. (a) an example of *type*1 relation, (b) an example of *type*2 relation, (c) an example of *type*3 relation (i.e., v_i overlaps with v_j), and (d) an example of *type*4 relation (i.e., v_i is v_j).

*type*1: v_i is on one side of $S_{j,k}$
*type*2: v_i is on $S_{j,k}$
*type*3: v_i overlaps with one endpoint of $S_{j,k}$, i.e., v_i is a common vertex
*type*4: v_i is one endpoint of $S_{j,k}$

For *type*4, v_i is still one endpoint of $S_{j,k}$ after data hiding. Since preserving the disjoint relation between the two endpoints of $S_{j,k}$ will help to maintain the shape of the polyline or polygon $S_{j,k}$ is within, we also consider this type.

To avoid changing the relation between v_i and $S_{j,k}$ from one type to another, we calculate the CPR for each vertex and each line segment, and select *eligible* vertices for data hiding based on the CPRs of the vertex-line segment pairs.

2.1.1 CPR of a vertex and CPR of a line segment

1) CPR of a vertex v_i: We define the CPR of v_i as a region that not only contains both v_i and its embedded version v_i', but also can be correctly calculated during data extraction. We can calculate a CPR for each vertex for several existing quantization-based 2D vector graphics reversible data hiding algorithms [3, 13-18]. These methods hide data by moving the vertex within a quantization interval. After data hiding, both the vertex and its corresponding version are in the same quantization interval.

For example, in [15], data are embedded into a vertex v_i based on the quantization of the distances between the vertex and two orthogonal lines. Assuming the quantization interval is Q_w, v_i and its corresponding watermarked version v_i' are in the same square region R_i with each side measuring Q_w (Fig. 4(a)). v_i may move to any position within R_i due to data hiding, and R_i can be correctly calculated during data extraction according to v_i' and Q_w. R_i can be regarded as the CPR of v_i for [15].

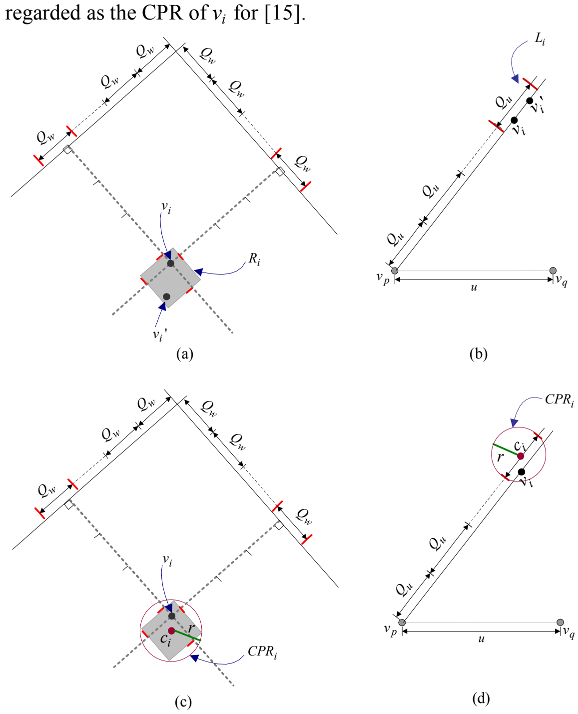

Fig. 4. (a) R_i of the vertex v_i, (b) L_i of the vertex v_i, (c) the corresponding CPR of (a), and (d) the corresponding CPR of (b).

For one more example, in [16], data are embedded into a vertex v_i based on the quantization of the distance between the vertex and a fixed point v_p (Fig. 4(b)). Assuming Q_u ($Q_u = 2^b \Delta u$, where b, Δ and u are three embedding parameters in [16]) is the quantization interval, both v_i and v_i' are in the same interval L_i. v_i may move to any position within L_i due to data hiding, and L_i can be correctly calculated during data extraction according to v_i'. L_i can be regarded as the CPR of v_i for [16].

For convenience of description, we regard the region CPR_i, which is enclosed by the circumcircle of R_i (or L_i), as the CPR of v_i (Fig. 4(c), Fig. 4(d)). Denote the center and radius of CPR_i by c_i and r, respectively.

2) CPR of a line segment $S_{j,k}$: According to the CPRs of v_j and v_k (i.e., CPR_j and CPR_k), we can get the tubular region in which $S_{j,k}$ may lie after data embedding (Fig. 5). We regard this region as the CPR of $S_{j,k}$, denoted by $CPR_{j,k}$.

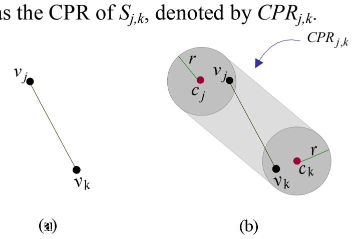

Fig. 5 (a) a line segment $S_{j,k}$, and (b) the CPR of $S_{j,k}$ (i.e., $CPR_{j,k}$).

2.1.2 eligible vertex selection

Let M be a 2D vector map to be embedded. We scan the vertices of M from the first vertex to the last, and mark each vertex as an *eligible* vertex or a *non-eligible* vertex, according to the CPRs of the vertices and line segments. If the embedding operation on a vertex does not change its relation with any line segment, i.e., it can be used to hide data, we call it an *eligible* vertex; otherwise, we call it a *non-eligible* vertex.

For different vertex-line segment types, we have different vertex marking methods.

vertex marking for *type*1: If CPR_i and $CPR_{j,k}$ are disjoint, v_i will remain on the same side of $S_{j,k}$ after embedding, then v_i is marked as a possible *eligible* vertex (Fig. 6 (a)); otherwise, v_i, v_j and v_k are marked as *non-eligible* vertices to preserve the *type*1 relation. That is,

$$\begin{cases} v_i \text{ is a possible } eligible \text{ vertex}, & if \;\; DS(c_i, S^c_{j,k}) > 2r \\ v_i, v_j, v_k \text{ are } non-eligible \text{ vertices}, & if \;\; DS(c_i, S^c_{j,k}) \le 2r \end{cases}, \quad (1)$$

where $S^c_{j,k}$ is the line segment between the center c_j of CPR_j and the center c_k of CPR_k, and $DS(v_i, S^c_{j,k})$ represents a method that calculates the Euclidean distance between the vertex v_i and the line segment $S^c_{j,k}$.

1) vertex marking for *type*2: v_i, v_j and v_k are all marked as *non-eligible* vertices for preserving the *type*2 relation.

2) vertex marking for *type*3: Assume v_i overlaps with v_j (Fig. 6(b)). v_i may be a common vertex for several line segments. During this step, we only consider how to preserve the disjoint relation between v_i and v_k, and the approach about how to preserve the overlapping relation of the common vertices will be given after this step. Here, if CPR_i and CPR_k are disjoint, v_i and v_k will remain disjoint after data hiding, and v_i is seen as a possible *eligible* vertex; otherwise, v_i, v_j and v_k are marked as *non-eligible* vertices for preserving the disjoint relation between v_i and v_k. That is,

$$\begin{cases} v_i \text{ is a possible } eligible \text{ vertex}, & if \;\; |c_i - c_k| > 2r \\ v_i, v_j, v_k \text{ are } non-eligible \text{ vertices}, & if \;\; |c_i - c_k| \le 2r \end{cases}. \quad (2)$$

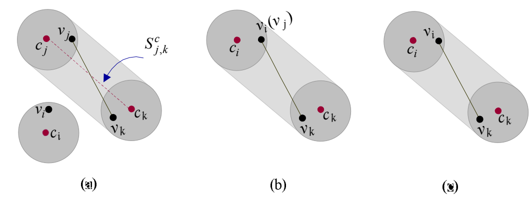

Fig. 6 (a) disjoint CPR_i and $CPR_{j,k}$ in *type*1 relation, (b) disjoint CPR_i and CPR_k in *type*3 relation, and (c) disjoint CPR_i and CPR_k in *type*4 relation.

3) vertex marking for *type*4: Assume v_i and v_j are the same vertex (Fig. 6(c)). For preserving the disjoint relation between v_i and v_k, we mark the vertices by

$$\begin{cases} v_i \text{ is a possible } eligible \text{ vertex}, & if \;\; |c_i - c_k| > 2r \\ v_i, v_k \text{ are } non-eligible \text{ vertices}, & if \;\; |c_i - c_k| \le 2r \end{cases}. \quad (3)$$

If we have that v_i is a possible *eligible* vertex after checking all the relations between v_i and any line segment in the vector map, v_i is marked as an *eligible* vertex. If the *eligible* vertex v_i is also a common vertex (i.e., the vertex-line segment relation type between v_i and at least one line segment is *type*3), we call it an *eligible* common vertex. Then, v_i and the vertices that overlap with it should hide the same data in order to preserve the *type*3 relation.

2.2 Data embedding and data recovery

During data embedding, we embed data into each *eligible* vertex using one of the methods in [3, 13-18]. If a vertex v_i is an *eligible* common vertex, v_i and the vertices that overlap with it hide the same data. After data embedding, an embedded vector map M' is derived. Since the embedding operation on the *eligible* vertices does not change any vertex-line segment relation, the topological relations between map objects can be preserved.

In the data recovery phase, the embedded *eligible* vertices are first selected using our topological relation preservation technique proposed in Section 2.1. After that, the hidden data are extracted from the embedded *eligible* vertices and the original content of M' can be recovered. Since the original CPRs of each vertex-line segment pair can be correctly calculated according to the received content, the embedded *eligible* vertices can be identified during data extraction. Therefore, correct data extraction and recovery can be ensured.

3. EXPERIMENTAL RESULTS AND ANALYSIS

We ran some experiments on a set of vector maps to test the performance of the proposed reversible data hiding method. Four of the vector maps are a coastline map of Taylor Rookery [21] (M1), a coastline map of Windmill islands [22] (M2), a flying bird colony map of Rauer islands [23] (M3), and a forest map (M4). Table 1 gives the detailed information of the four vector maps, including the map object type, the number of map objects /vertices, the scale and the precision tolerance τ (i.e., the maximum distortion that a vector map can bear). During data hiding, [15] was used to hide data, and the parameters were selected as $b = s = 1$.

To evaluate the topological relation preservation ability, we define a metric *TCR* (*Topological Change Rate*) as

$$TCR = T_c \Big/ \binom{N}{2}, \tag{4}$$

where T_c denotes the number of map object pairs whose directional relations have changed after data hiding, N represents the number of map objects of the vector map *M*, and $\binom{N}{2}$ is the total number of combinations of 2 map objects selected from N map objects.

Table 1 Properties of original vector maps

Maps	Map object type	Map objects/ vertices	Scale	τ (m)
M1	polyline	18/4279	1:5000	0.5
M2	polyline	496/38082	1:50000	5
M3	polygon	42/3172	1:50000	5
M4	polygon	26/15831	1:1000000	100

Table 2 shows the *TCR* values of the proposed method and the methods in [12-13, 15-16]. We observe that due to the selection of the *eligible* vertices for data hiding, the proposed method preserves the topological relations between map objects. Since the relations between vertices and line segments are not considered in [12-13, 15-16], the topological relation preservation ability of the proposed method is better.

Table 2 *TCR* values of different methods

Maps	[12]	[13]	[15]	[16]	Proposed
M1	0.0131	0.0065	0.0654	0.0523	0.0000
M2	0.0006	0.0001	0.0017	0.0017	0.0000
M3	0.0244	0.0197	0.0197	0.0197	0.0000
M4	0.0031	0.0031	0.0123	0.0031	0.0000
Average of 50 maps	0.0034	0.0022	0.0102	0.0118	0.0000

For measuring the distortion introduced by embedding, the average distortion *d*(*M*, M') and the maximum distortion *Maxd*(*M*, M') were used,

$$d(M, M') = \frac{1}{n}\sum_{i=1}^{n}\left|v_i - v_i'\right|, \tag{5}$$

$$Maxd(M, M') = \max(\left|v_i - v_i'\right|), \quad (i = 1, 2, \ldots, n)$$

where v_i and v_i' are the corresponding vertices in the original vector map *M* and the embedded vector map M', and *n* denotes the total number of vertices in the vector map *M*. Table 3 shows the *d* values and the *Maxd* values of the proposed method and the methods in [12-13, 15-16]. From Table 3, we can see that the *d* values and the *Maxd* values of the proposed method are smaller than [12]. The method in [12] hides data by adding/subtracting a real number to/from each vertex set, and some higher digits may be modified when the added/subtracted real number is too great. We can also observe that the invisibility of the proposed method is comparable to [13, 15]. That's because we use [15] to hide data, and the embedding parameter selection technique of [15] can guarantee that the distortion does not exceed the precision tolerance τ of each vector map. When applying [16] to a 2D vector map, the distortion introduced by embedding may not be well controlled, e.g., M2, M3 and M4. That's because the quantization interval is closely related to the distance between two reference vertices in [16], and the random selection of the reference vertices may make the embedded vector map quality undesirable. The proposed method provides better invisibility than [16].

Table 3 *Maxd* and *d* values of different methods (m)

Metric	Method	M1	M2	M3	M4
Maxd	[12]	4.958	27.039	33.590	704.559
	[13]	0.245	2.493	2.481	49.880
	[15]	0.389	4.147	3.967	80.447
	[16]	0.041	28.679	5.498	389.301
	Proposed	0.415	3.937	3.554	78.512
d	[12]	0.336	4.577	6.072	149.859
	[13]	0.134	1.322	1.323	26.922
	[15]	0.161	1.610	1.615	32.508
	[16]	0.017	12.407	2.394	168.195
	Proposed	0.128	0.445	0.377	12.574

4. CONCLUSIONS

In this paper, we describe a 2D vector map reversible data hiding scheme with topological relation preservation. By calculating the CPR for each vertex and each line segment, and selecting *eligible* vertices according to the CPRs of the vertex-line segment pairs for data hiding, topological relations between map objects can be preserved.

In this paper, the topological relation preservation method was described as integrated with [3, 13-18]. The method can, however, be integrated with other data hiding algorithms (e.g., [24]) in order to preserve the topological relations between map objects, as long as the data hiding algorithm can calculate a CPR for each vertex.

5. ACKNOWLEDGEMENTS

We thank Prof. Mohan S Kankanhalli, National University of Singapore, for his thoughtful advice and comments. This work is supported by the National Natural Science Foundation of China (Grant No. 61602218), University Science Research Project of Jiangsu Province (Grant No. 16KJB520011) and the State Scholarship Fund of China.

6. REFERENCES

[1] A. Abubahia and M. Cocea, "Advancements in GIS map copyright protection schemes - a critical review," *Multimedia Tools and Applications*, vol. 76, no. 10, pp. 12205-12231, 2017.

[2] S.-H. Lee and K.-R. Kwon, "Vector watermarking scheme for GIS vector map management," *Multimedia Tools and Applications*, vol. 63, no. 3, pp. 757-790, 2013.

[3] N. Wang and X. Zhao, "2D vector map data hiding with directional relations preservation between points," *AEU-International Journal of Electronics and Communications*, vol. 71, pp. 118-124, 2017.

[4] S. Huber, M. Held, P. Meerwald, and R. Kwitt, "Topology-preserving watermarking of vector graphics," *International Journal of Computational Geometry & Applications*, vol. 24, no. 01, pp. 61-86, 2014.

[5] J. Lafaye, J. Béguec, D. Gross-Amblard, and A. Ruas, "Blind and squaring-resistant watermarking of vectorial building layers," *Geoinformatia*, vol. 16, no. 2, pp. 245-279, 2013.

[6] M. Voigt, B. Yang, and C. Busch, "Reversible watermarking of 2D-vector data," *Proceedings of the Multimedia and Security Workshop*, pp. 160-165, 2004.

[7] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896, 2003.

[8] X. Wang, C. Shao, X. Xu, and X. 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.

[9] L. Zhou, Y.-j. Hu, H.-f. Zeng, "Reversible data hiding algorithm for vector digital maps," *Journal of Computer Applications*, vol. 29, no. 4, pp. 990-993, 2009.

[10] H. Sun, G. Li, J. Zhu, and Y. Shi, "Improved reversible watermarking algorithm for vector map based on difference expansion and shifting," *Geomatics and Information Science of Wuhan University*, vol. 37, no. 8, pp. 1004-1007, 2012.

[11] J. Hu and M. Geng, "A reversible watermarking algorithm for 2d vector maps," *2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*, pp. 1101-1104, 2013.

[12] L. Cao, C. Men, and R. Ji, "High-capacity reversible watermarking scheme of 2D-vector data," *Signal, Image and Video Processing*, vol. 9, no. 6, pp. 1387-1394, 2015.

[13] N. Wang, H. Zhang, and C. Men, "A high capacity reversible data hiding method for 2D vector maps based on *virtual* coordinates," *Computer-Aided Design*, vol. 47, pp. 108-117, 2014.

[14] N. Wang, X. Zhao, and C. Xie, "RST invariant reversible watermarking for 2D vector map," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 11, no. 2, pp. 265-276, 2016.

[15] N. Wang, "Reversible watermarking for 2D vector maps based on *normalized* vertices," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 20935-20953, 2017.

[16] D. Xiao, S. Hu, and H. Zheng, "A high capacity combined reversible watermarking scheme for 2-D CAD engineering graphics," *Multimedia Tools and Applications*, vol. 74 , no. 6, pp. 2109-2126, 2015.

[17] F. Peng, Q. Long, Z.X. Lin, and M. Long. "A reversible watermarking for authenticating 2D CAD engineering graphics based on iterative embedding and *virtual* coordinates," *Multimedia Tools and Applications*, pp. 1-21, 2017.

[18] F. Peng, and Y.Z. Lei, "An effective reversible watermarking for 2D CAD engineering graphics based on improved QIM," *International Journal of Digital Crime and Forensics*, vol. 3, no. 1, pp. 53-69, 2011.

[19] F. Peng, Y. Liu, M. Long, "Reversible watermarking for 2D CAD engineering graphics based on improved histogram shifting," *Computer-Aided Design*, vol. 49, pp. 42-50, 2014.

[20] F. Peng, Y.Z. Lei, M. Long, and X.M. Sun, "A reversible watermarking scheme for two-dimensional CAD engineering graphics based on improved difference expansion," *Computer-Aided Design*, vol. 43, no. 8, pp.1018–1124, 2011.

[21] U. Harris, "Taylor Rookery 1:5000 Topographic GIS Dataset Australian Antarctic Data Centre - CAASM Metadata," *https://data.aad.gov.au/metadata/records/Tayl5k*, 1999, updated 2014.

[22] U. Harris, "Windmill Islands 1:50000 Topographic GIS Dataset Australian Antarctic Data Centre - CAASM Metadata, " *https://data.aad.gov.au/metadata/records/Wind50k*, 1999, updated 2014.

[23] U. Harris, "Rauer Group 1:50000 Topographic GIS Dataset Australian Antarctic Data Centre - CAASM Metadata," *https://data.aad.gov.au/metadata/records/Raur50k*, 1999, updated 2014.

[24] H. Yan, L. Zhang, and W. Yang, "A normalization-based watermarking scheme for 2D vector map data," *Earth Science Informatics*, pp. 1-11, 2017.