Partition Clustering for GIS Map Data Protection

Ahmed M. Abubahia
School of Computing
University of Portsmouth
Portsmouth, United Kingdom
Email: ahmed.abubahia@port.ac.uk

Mihaela Cocea
School of Computing
University of Portsmouth
Portsmouth, United Kingdom
Email: mihaela.cocea@port.ac.uk

***Abstract*—One of the main research issues of digital data is defined by copyright protection, and digital watermarking is a potential solution to this issue. While there is an abundance of research on digital watermarking for image data, there is far less research on digital watermarking for vector map data, a data format used to store complex information in Geographical Information Systems (GIS). Recently, data mining methods have been used in the process of watermarking vector data. In this paper, we argue that the security of the watermarked vector maps can be increased by employing more suitable data mining methods. In particular, in this paper, we advocate the use of k-medoids partition clustering and compare its deployment with a previous watermarking scheme in which k-means partition clustering is used. The experimental results show that it outperforms the approach based on k-means according to a set of evaluation metrics.**

Keywords*—GIS; vector data; digital watermarking; copyright protection; k-medoids partition clustering; ESRI shapefile

I. Introduction

In last four years, the compelling need for protecting the copyright of digital vector maps has become an emergent topic within the GIS (Geographic Information System) research community that stemmed from the rapid growth of intelligent tools and devices [1], [2]. One of the main economic, social and legal aspects of using GIS data is defined by copyright protection [3]. This has been enforced and administrated internationally by UN-WIPO (United Nations - World Intellectual Property Organization), by considering the digital maps as software products [4].

Unlike other physical data, digital data has its own features of being intangible and dynamic, which make it easy to be copied, modified or distributed through different media such as CDs, DVDs, USBs or via internet servers [5].

In the digital context, the copyright offers an exclusive right to secure and protect the livelihood of original work producers. This helps prevent illegal digital copies being distributed on internet web sites and used instead of the original productions. In case of copyright dispute, digital watermarking can be used for claiming ownership. Digital watermarking has been proposed, in recent years, as an effective solution to combat this threat of piracy.

In watermarking research, digital multimedia data such as images, text, audio and videos received more attention by researchers and scholars than digital vector map data [6]. The spatial structure and topological relations within the vector map type of data are features that make it different from other multimedia data. The key difference between vector data and image data, as illustrated in Table I, is the small redundancy available to hide the watermark due to the precision intolerance of vertices' coordinates. In addition, digital vector data has great economic significance due to the value of its accurate content [7]. Digital maps are developed for complex data, which makes them suitable to be used in many applications where accuracy is important, such as navigation, strategic planning, military services and decision making [1].

TABLE I. VECTOR DATA VERSUS IMAGE/RASTER DATA

Aspect	**Vector Data**	**Image Data**
Feature Representation	Points/Lines/Polygons	Array of pixels
Resolution Determination	Precise coordinates	Pixel size
Efficiency	Sparse data	Dense data
Spatial Relations	Exist	Do not exist
Storage Requirement	Small space	Large space
Redundancy Size	Small	Large

In recent years, a considerable amount of research has been carried out to solve the issue of copyright protection in the context of digital vector data, e.g., [3], [6], [8]. A handful of research papers proposed watermarking methods that use data mining tools in the context of digital vector data copyright protection [9]–[12]. These methods can be categorized into two main categories: clustering-based methods [9]–[11] and classification-based methods [12]. In the literature, clustering-based methods are more prevalent than classification-based methods; consequently, we focus on clustering methods.

In particular, we advocate that the clustering method used has an influence on the security of the watermarked vector map, where security is measured through specific evaluation metrics, which are outlined in Section II. More specifically, we propose the use of a k-medoids partition clustering approach; there are several implementations of this approach, of which the PAM (Partitioning Around Medoids) method is the most popular [13], [14]. In this paper, we investigate whether the use of PAM leads to a more secure watermarked map in comparison with a k-means partition clustering method.

The rest of this paper is organized as described in the following. In section II, a detailed overview of relevant previous work is presented. Section III describes the geospatial data format and the platform that has been utilized for the experimental evaluation of the approach proposed in this paper. Section IV presents the full explanation of our approach including: selecting the embedding positions and implementing the embedding and extraction strategy. Section V describes our

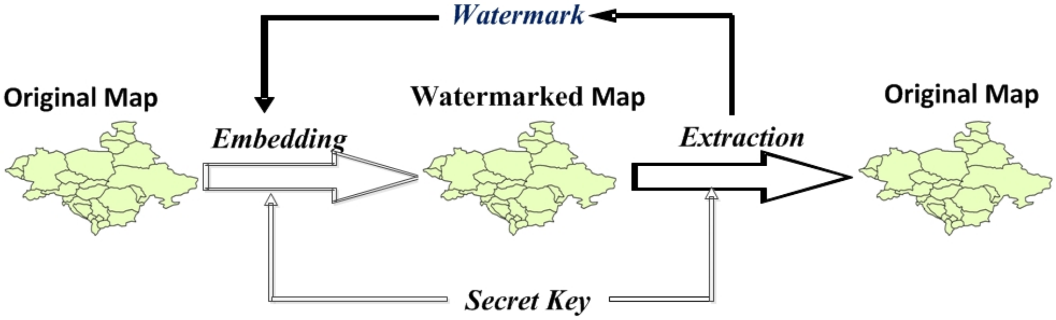


Fig. 1. The General System of Digital Vector Map Watermarking.

experiments and discusses the findings. Section VI concludes the paper.

II. Related Work

In GIS vector map data, a sequence of vertices' coordinates is used to represent geographical locations of the digital map object, which can take one of three types of geometry shapes: point, polyline and polygon [8].

A digital vector map watermarking system, as shown in Fig. 1, consists of two substantial stages: embedding and extraction. The embedding stage refers to the process of inserting copyright information, which is called a watermark, into the host data.

In the former stage, one or more secret keys are used for adding more security to the embedded locations in the digital map, as well as keeping these locations unknown to potential attackers. The stage of watermark extraction aims to obtain the watermark from the host data by using the aforementioned secret key(s). The purpose of extraction is to obtain the watermark so that the original map can be retrieved.

In the literature, digital map watermarking algorithms are classified into two main types: spatial domain and transform domain. Spatial domain algorithms are concerned with embedding the watermark directly into different spaces, such as Cartesian coordinates, polar coordinates, blocks and topology relations. Transform domain algorithms deal with inserting the watermark into a transformed form of data. The most frequently used data transformations in the watermarking context are wavelet transform, Fourier transform and cosine transform [8].

The security of a watermarked map is evaluated by looking at four aspects: capacity, fidelity, computational time and robustness [8].

Capacity refers to the number of bits that can be embedded in the host data [15], [16]. In addition to the number of embedded bits in the host data, these bits should be spread across the whole map in order to provide more robustness to cropping attacks, which refer to cutting parts of the host map [17]. The use of clustering methods in the process of watermarking ensure a good spread by identifying locations for embedding throughout the map.

Fidelity refers to the fact that the watermark embedding process should not affect the quality of the host data and that the watermark should not be noticeable to the human eye [18].

There is a trade-off between capacity and fidelity: inserting many watermark bits, i.e., increased capacity, leads to a loss of fidelity or quality of the host map [8]. Consequently, there is a need to balance the capacity of the map with its fidelity to achieve good security without loss of quality.

Computation time/complexity refers to the period of time that is required to perform the embedding process and obtaining the watermarked data [19].

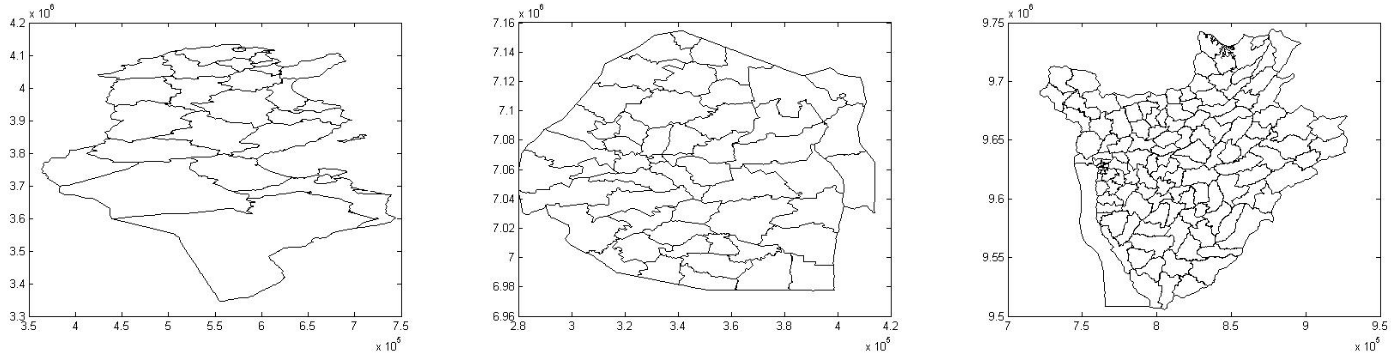
Robustness refers to the ability of the watermaked map to withstand any kind of modifications, called attacks, to the host data [15]. Examples of these attacks are geometric modifications processes such as rotation, translation and scaling [9].

To the best of our knowledge, there are only three published watermarking methods that used clustering data mining approaches to watermark GIS vector map data. In the following, we review these watermarking methods and outline their advantages and disadvantages in relation to the evaluation metrics mentioned above.

Jianguo et al. [10] proposed an algorithm that used fuzzy spatial clustering analysis for embedding a 1-dimensional binary code watermark into a digital vector map. Their evaluation indicated that the algorithm outperforms some shifting, cosine transform and Fourier transform based algorithms in terms of data fidelity. Although this algorithm maintains the fidelity of the map data features, it is vulnerable to geometric attacks such as rotation, translation and scaling, which can easily result in the loss of the embedded watermark.

Haowen [11] proposed a watermarking algorithm for embedding a 2-dimensional binary image watermark with a size of 32x96, into a vector point data set. In this approach, however, neither the capacity nor the trade-off between capacity and fidelity metrics were taken into account, which have crucial implications on the security of the digital map.

The approach of Huo et al. [9] used k-means partition clustering for watermarking a digital map based on the polygon geometry type of the ESRI shapefile; properties of this particular format are described in the next section. Polygons' centers were clustered into 80 clusters, equal to the number of watermark bits. 80 random centers were used as initial centers for the k-means clustering method. The watermark bits were embedded into the mean-distance, also called average-distance [20] length of polygons. This algorithm used 3-keys for accomplishing a good security. Unlike the previous two approaches, this watermarking method took into consideration



(a) Map 1 (27 polygons)

(b) Map 2 (53 polygons)

(c) Map 3 (132 polygons)

Fig. 2. The maps used in the experiments.

both the robustness to attacks and the trade-off between capacity and fidelity.

In this paper we argue that the partition clustering method used in the process of identifying the location for embedding the watermark has an influence on the security of the watermarked map measured in terms of capacity, fidelity, computational time and robustness. To investigate this, we propose a k-medoids approach and compare it with the approach of Huo et al. [9] as a best representative of partition clustering-based watermark embedding approaches, because it takes into consideration both the trade-off between capacity and fidelity, and the robustness to geometric attacks.

III. Materials

This section describes the data format used and the platform that has been utilized for implementing the proposed approach in this paper.

A particular data format is used, which is called shapefile (.shp) and was developed by ESRI[1], a major company supplying Geographic Information System (GIS) software and geodatabase management applications which are widely used in over 200 countries. The shapefile is a popular format used in GIS applications due to its prominent characteristics. These characteristics can be summarized as [21]:

1) It requires less storage space than image data;
2) It has considerable speed in drawing and editing shapes;
3) It stores spatial features, in the form of coordinates, and their attribute information;
4) It supports all types of geometry, i.e points, lines and polygons;
5) It is easy to read and write.

Three maps covering three countries in Africa were used for the research presented in this paper, which are illustrated in Fig. 2; these maps are freely available from the Map Library website[2]. As shown in Fig. 2, we used the administrative areas map of three countries: Tunisia (27 polygons), Swaziland (53 polygons) and Burundi (132 polygons).

For the watermark embedding and extraction processes, and for measuring the computational time, MATLAB was used with the following details: version R2013b (8.2.0.701), 32-bits and license No. 484067. For more information regarding MATLAB, see the Mathworks website[3].

IV. The Proposed Approach

Our proposed approach aims to assess the influence of k-medoids in comparison with k-means partition clustering on the trade-off between the capacity and fidelity metrics, as well as on the computational complexity and robustness metrics. For this purpose, we use the approach of Huo et al. [9] and vary the two aspects highlighted in Fig. 3.

The GIS map watermarking approach, as illustrated in Fig. 3, consists of determining the embedding positions (first four steps) and embedding the watermark into the host map by applying odd-even indexing method (last three steps). These steps are explained in sub-sections IV-A and IV-B, respectively. Regarding the watermark extraction, the last part of the watermarking process illustrated in Fig. 1, it is explained in sub-section IV-C.

A. The Watermark Embedding Positions

Embedding positions refer to a set of map locations to be modified by inserting the watermark bits. In Huo et al. [9], the process of selecting the embedding positions includes the following steps: calculation of polygons' centers, selecting random centers to be used as initial cluster centers for k-means clustering and selecting the centers of polygons to be used for embedding. The last step is accomplished for each cluster, by choosing the closest point to the center of the cluster. Finally, the mean distance length is calculated, to be used in the watermark embedding method.

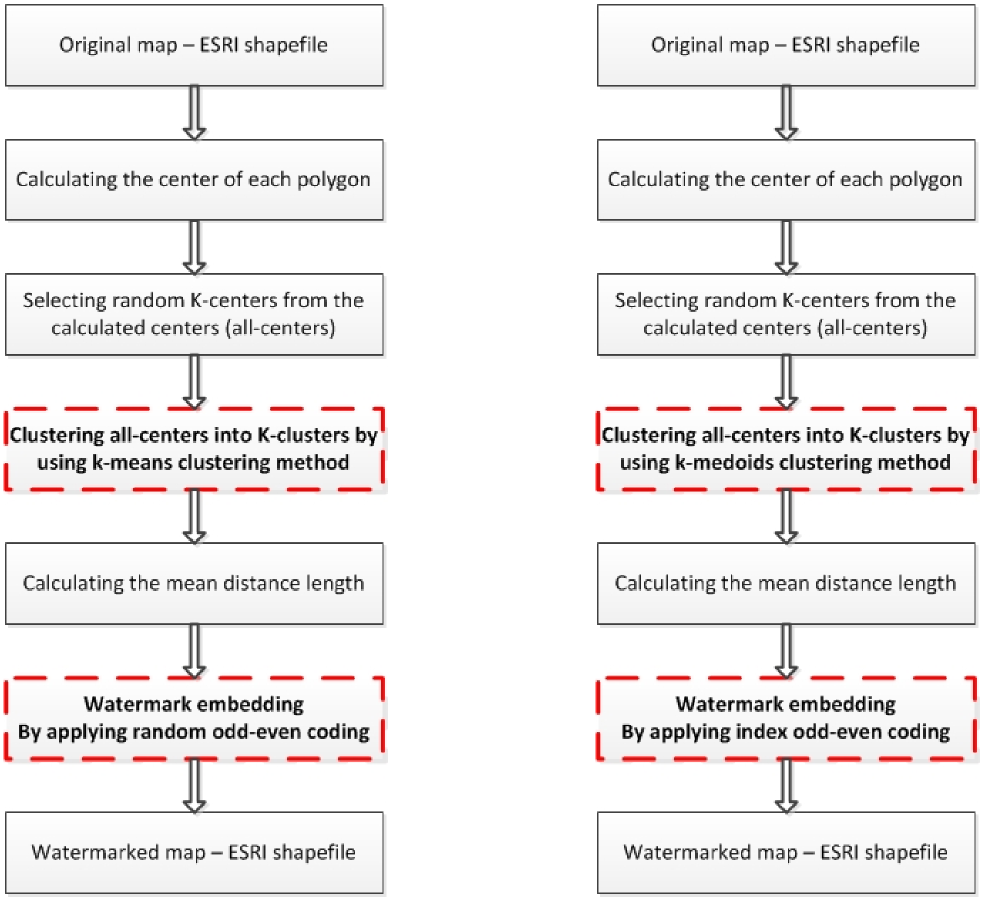
Our approach uses k-medoids instead of k-means and is presented in detail below.

- The calculation of polygons' centers: Polygons' centres are calculated in both axes [22], as shown in Equations (1) and (2), by summing all vertices coordinates for each polygon and then dividing by the

[1] http://www.esri.com/

[2] http://www.mapmakerdata.co.uk.s3-website-eu-west-1.amazonaws.com/library/stacks/Africa/index.htm

[3] http://www.mathworks.co.uk/


Original map – ESRI shapefile
Calculating the center of each polygon
Selecting random K-centers from the calculated centers (all-centers)
Clustering all-centers into K-clusters by using k-means clustering method
Calculating the mean distance length
Watermark embedding By applying random odd-even coding
Watermarked map – ESRI shapefile
Original map – ESRI shapefile
Calculating the center of each polygon
Selecting random K-centers from the calculated centers (all-centers)
Clustering all-centers into K-clusters by using k-medoids clustering method
Calculating the mean distance length
Watermark embedding By applying index odd-even coding
Watermarked map – ESRI shapefile

(a) Huo et.al Scheme [9]

(b) Our Proposed Approach

Fig. 3. The Compared Embedding Framework.

number of vertices minus one; the subtraction of one is due to the last vertex coordinates being the same as for the first vertex, according to the polygon shapefile format [21].

$$x_c = \sum_{i=1}^{n-1} \frac{x_i}{n-1} \qquad (1)$$

$$y_c = \sum_{i=1}^{n-1} \frac{y_i}{n-1} \qquad (2)$$

where: x_c and y_c are the coordinates of polygon's center in both x and y axes respectively; n is the number of all vertices within the same polygon; i is the order of the vertex in the polygon.

- Clustering of polygons' centers. In contrast to the scheme of Huo et al. [9], our approach uses a k-medoids based partition clustering method called PAM (Partitioning Around Medoids). PAM method works firstly by arbitrarily assigning initial representative objects, called seeds. Subsequently, it replaces the seeds by other representative objects iteratively. This process continues until the resulting medoids, i.e. clusters' representative objects, can not be improved or changed [13], [14]. Polygons' centers are clustered into k-clusters and the resulting medoids are kept as a secret key ($key1$). The k-medoids mechanism [14] is summarized in **Algorithm 1**. Unlike k-means, the centers of clusters are actual polygon centers, not artificial points which did not exist in the initial data set of polygon centers [14].
  The k-medoids method outperforms the k-means method by its robustness to outliers, i.e. objects that are far from the majority of the data within the same cluster. Both k-means and k-medoids need the number of clusters to be specified by the user [14], [23], which has an advantage of controlling the number of watermark embedding locations, which have a good influence on increasing the capacity.
  Another specific advantage of applying k-medoids method in the context of watermarking GIS vector data, is that clusters' centers are actual data points from the map data sets. In contrast, the clusters' centers in the k-means method are artificial points, which introduces an element of approximation that is not present in the k-medoids algorithm.

- Calculating the mean distance length. The mean-distance length is the average of distances from the polygon's center to each of its surrounding vertices within the same polygon [9], [20]. The values of

Algorithm 1 k-medoids (PAM) method for partitioning based on medoid.

```
$Input$:
$k$: the number of clusters,
$D$: a data set containing n objects.
$Output$: A set of $k$ clusters.
$Method$:
    ○ arbitrarily choose $k$ objects in $D$ as the initial representative objects or seeds;
    ○ repeat
    ○ assign each remaining object to the cluster with the nearest representative object in terms of Euclidean distance;
    ○ randomly select a non-representative object, $O_{random}$;
    ○ compute the total cost, $S$, of swapping representative object, $O_j$, with $O_{random}$;
    ○ if $S < 0$ then swap $O_j$ with $O_{random}$ to form the new set of $k$ representative objects;
    ○ until no change;
```

mean-distance lengths are kept as another secret key ($key2$) and used as targeted positions for watermark embedding. Equation (3) demonstrates the way of calculating the mean-distance length of polygons that are selected by using the k-medoids partition clustering method.

$$L_c = \frac{1}{n-1} \sum_{v=1}^{n-1} \sqrt{(x_c - x_v)^2 + (y_c - y_v)^2} \quad (3)$$

where: L_c is the mean distance length; n is the number of vertices in a polygon; v is the vertex order; x_c and y_c are the center coordinates in x and y axes, respectively; x_v and y_v are the vertex coordinates in x and y axes, respectively.

B. The Watermark Embedding Method

The watermark is structured on the basis of the zero watermark concept [9]. Zero watermarking aims to utilize some key characteristics of the host map data in order to generate a more robust watermark. In this case, the characteristic of the host map data that is used is the mean-distance length of polygons. The watermark is constructed by adding or subtracting a bit value of 1 from the mean-distance length of polygons.

The watermark is embedded by applying an odd-even indexing condition [9], [24], as outlined in Equation (4). The index of each mean-distance value is used in this approach, instead of using an additional random sequence proposed by [9], to simplify the implementation and also to have more consistent positions for embedding the watermark.

This indexing plays a vital role in combination with the clustering process by:

1) Maintaining the security of the watermark position by storing the index values as a key instead of utilizing a random sequence that is not relevant to the used data;
2) Ensuring that all selected polygons are used as watermark carriers to attain a maximum value of capacity;
3) The ability to increase the watermark capacity while preserving the map fidelity, whereas the use of random sequence and indexing condition in [9] will limit that choice of control.

$$W_i = \begin{cases} T+1, & \text{if } OES(I) = odd \\ T-1, & \text{if } OES(I) = even \end{cases} \quad (4)$$

where: W_i is the ith bit value of the watermark; OES stands for Odd-Even Status; I is the order index of the mean-distance length value in the matrix; T is the value of the 4th digit of the mean-distance length value, after the decimal point [9].

As shown in Equation (4), the watermark is embedded by comparing the OES (Odd-Even Status) of both I and T variables. The conditions are set based on two scenarios as following:

- If the OES of I is odd, 1 will be subtracted from the value of T.
- In contrast, if the OES of I is even, 1 will be added to the value of T.

After applying the OES to change the values of the mean-distance length L_c, the new values will be represented by L_c^*. This new mean-distance length values are stored as an additional secret key ($key3$), to secure the positions in which the watermark is embedded. Following, the change rate α_c is calculated as depicted in Equation (5):

$$\alpha_c = \frac{L_c^*}{L_c} \quad (5)$$

The change rate α_c is used to change all vertices of polygons that belong to each cluster's center on the basis of embedding condition, as given in equations 6 and 7:

$$v_x^* = \alpha_c v_x + x_c(1 - \alpha_c) \quad (6)$$

$$v_y^* = \alpha_c v_y + y_c(1 - \alpha_c) \quad (7)$$

where: v_x^* and v_y^* are the new vertices' coordinates after embedding the watermark according to the aforementioned condition, in Equation (4).

C. The Watermark Extraction Method

The watermark extracting process is flexible and quite similar to the embedding process. It is performed by using the keys stored during the embedding process. Firstly, we calculate the center of each polygon, then dividing all centers into k number of clusters by using the k-medoids partition clustering algorithm. In the next stage, the mean-distance length is computed for the watermarked map in the same way as given in the watermark embedding process.

V. Experimental Results and Discussion

To assess the difference introduced by the k-medoids partition clustering method, we carried out a set of experiments regarding fidelity, robustness and capacity, which are described in the following sub-sections, i.e. V-A, V-B, V-C and V-D respectively. These experiments are carried on the three maps that are shown in Fig. 2.

To enable this comparison, we simulated the scheme of Hou et al. [9], as given in their paper and implemented our approach as described previously. This enabled us to compare the two schemes and assess the improvement that could be achieved regarding map data protection.

Table II shows the experimental results of the our implementation in terms of capacity and fidelity, which will be discussed in the following subsections in more detail; the compared results according to computation time, are given in Table III, while robustness is discussed separately. We used different proportions of map size, i.e. 25%, 33% and 50%, to verify the consistency of results.

A. The Watermark Capacity Evaluation

Capacity refers to the number of watermark bits that is embedded in the host map. In this paper, the watermark capacity is expressed by the number of vertices that carry the watermark bits. Table II compares the effectiveness of our approach against the approach in [9], in relation to the map size proportions of 25%, 33% and 50%, respectively.

These percentages represent the amount of watermarked polygons within the original map, and has a vital implication on adding more resilience to cropping attacks. Cropping refers to the process of cutting some parts in the host map [17]. It is required that each cluster should contain more than one polygon's center, therefore it does not make sense to work with more than 50% of the map data. To illustrate the relation between the map size proportions and the number of clusters, Map 1, Map 2 and Map 3 are used. Thus, for Map 1, 25%, 33% and 50% corresponds to 7, 9 and 13 clusters, respectively, and for Map 2, 25%, 33% and 50% corresponds to 14, 18 and 27, respectively, while for Map 3, 25%, 33% and 50% corresponds to 33, 44 and 66 clusters, respectively.

Table II shows that our approach results in a higher capacity compared with the approach of Hou et al. [9]. Moreover, this is done without negatively affecting the fidelity. When using a quarter of the polygons, the capacity achievement of our approach was more than 58% higher than the compared approach, whereas using a third and half of the polygons, the capacity was raised by more than 50%.

As shown in Table II, our approach outperforms the compared approach due to the indexing mechanism of k-medoids partition clustering. This mechanism selects the centers of polygons in relation to surrounding vertices which result in embedding the watermark in more vertices.

B. The Map Fidelity Evaluation

The fidelity metric aims to measure the perceptual similarity between the watermarked map data and the original map data. It reflects the degree of invisibility the embedded watermark could have. Hou et al. [9] measured this invisibility by using PSNR (Peak Signal to Noise Ratio), in decibels. There is no specific range for PSNR values but a higher PSNR would normally indicate that the data is of higher quality [25]. The typical values are considered to be between 30 and 50 dB, in the context of digital images [26].

We used the same metric and Table II shows that although both our approach and the compared approach give the same fidelity results, our approach (k-medoids-based) outperforms the k-means-based approach in balancing the trade-off between the watermarked map fidelity and the watermark capacity. This is achieved by increasing the capacity without decreasing the fidelity of GIS map data. According to this, the fidelity value of infinity, as shown in Table II, is definitely considered as ideal outcome for the required invisibility.

Fig. 4 compares the original map in Fig. 4(a) with the watermarked maps using the clustering methods of k-means in Fig. 4(b) and k-medoids in Fig. 4(c), respectively. The figure illustrates that in both approaches, the watermarked maps are not different from the original one to the human eye.

C. The Computation Time Evaluation

Computational time refers to the time period, in seconds, for embedding the watermark bits into the host map. Table III compares our approach versus the scheme of Hou et al. [9], in terms of the time required to create the watermarked map. This table shows that our approach uses half the time in comparison to [9], making it more computationally efficient.

D. The Watermark Robustness Evaluation

Robustness reflects the watermark's resistance to a set of attacks or modifications. This paper focuses on geometric attacks such as rotation, translation and scaling because they are more relevant to the geometric nature of polygons in the digital maps context.

Using the mean-distance length values of the selected polygons as watermark carriers has a good implication on the effectiveness of the proposed watermarking approach due to the robustness of mean-distance values to both rotation and translation attacks, and having a way of estimating the scaling factor in the case of scaling attack. These characteristics of using the mean-distance values make the described watermarked approach robust to the geometric attacks [9], [20].

More specifically, attacks like rotation and translation have no effect on the embedded watermark because they affect equally all vertices' coordinate values, which, in turn, means

TABLE II. The compared results of our approach and the scheme of Hou et.al, [9].

Proportions of map size	Our Approach						Hou et al. [9]					
	Map 1		Map 2		Map 3		Map 1		Map 2		Map 3	
	Capacity	Fidelity	Capacity	Fidelity	Capacity	Fidelity	Capacity	Fidelity	Capacity	Fidelity	Capacity	Fidelity
25%	1318	INF	1881	INF	5451	INF	743	INF	1186	INF	2855	INF
33%	1870	INF	2478	INF	9604	INF	613	INF	1377	INF	6353	INF
50%	3315	INF	3806	INF	16417	INF	2288	INF	1912	INF	9123	INF

TABLE III. The compared computational time results of our approach and the scheme of Hou et.al, [9].

Proportions of map size	Our Approach (seconds)			Hou et al. [9] (seconds)		
	Map 1	Map 2	Map 3	Map 1	Map 2	Map 3
25%	0.055854	0.077921	0.134387	0.123270	0.183788	0.300455
33%	0.064616	0.082182	0.143285	0.129038	0.188778	0.303841
50%	0.065580	0.086568	0.143847	0.148653	0.202478	0.319511

that the distances between these vertices are not affected. Consequently, since the mean-distance length is used to construct the watermark, such attacks do not affect it.

In the case of a scaling attack, the scaling factor could be computed by dividing the mean-distance values of the modified/attacked map by the mean-distance values of the original map. Consequently, it is easy to retrieve the modified map to its original form before scaling was applied.

E. The Watermark Position Security Evaluation

In the described watermarking approach, securing the positions of the embedded watermark is achieved by the use of a set of secret keys. The first key is the values of clusters' centers, the second is the values of mean-distance lengths of the selected polygons by using the technique of OES, and the third key is the indexes of the of mean-distance values. These keys are stored for two main purposes: to be used in the extraction process, and for security purposes because they are kept secret from the attackers.

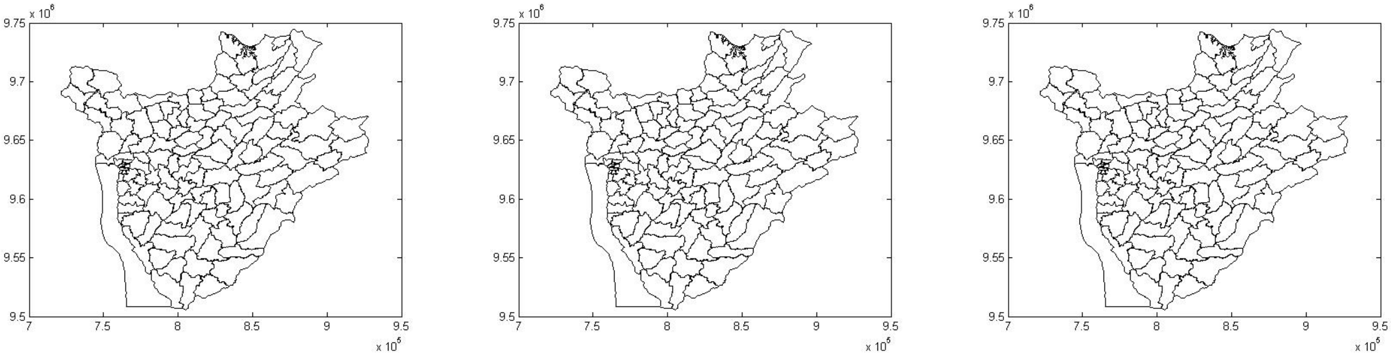
VI. Conclusions

In this paper we investigated the influence of the partition clustering method used in the watermarking process on the security of the watermarked map. We worked with the scheme proposed by Huo et al. [9] by replacing (a) their k-means clustering step with a k-medoids clustering approach, and (b) changing the indexing condition. While in k-medoids partition clustering the centers of clusters are data points from the data sets, in k-means partition clustering, the centers of clusters are artificial points. Consequently, k-means comes with an element of approximation that is not present in the k-medoids approach.

To evaluate the influence the partition clustering method had on the security of the watermarked map, we looked at four aspects: capacity, fidelity, computational time and robustness. The experimental results show that both k-medoids and k-means approaches result in high fidelity, while the k-medoids-based approach achieves a more balanced trade-off between capacity and fidelity, as well as better computational efficiency due to the k-medoids characteristics. In terms of robustness, the results are similar, although an argument could be put forward that this is improved indirectly in the k-medoids approach because of the higher capacity.

For measuring fidelity, PSNR was used to be consistent with Huo et al. [9]. This metric is used widely in image watermarking and has also been utilized for vector map data [15], [16], [27]. The map is converted to an image format to meet the applicability of PSNR. In future work we will investigate different metrics that would be more suitable for this type of data.

For further research, we will experiment with a fixed set of initial representatives for our k-medoids-based watermarking approach to achieve more efficiency and predictability, thus



(a) Original Map

(b) Watermarked Map using Kmeans, 50%

(c) Watermarked Map using Kmedoids, 50%

Fig. 4. Comparison between the original map (a) with the watermarked maps using k-means (b) and k-medoids (c).

eliminating the randomness involved in the initial selection of the centers involved in the typical PAM-based k-medoids partition clustering method. Also, we will experiment with density-based spatial clustering approaches proposed in the data mining literature to thoroughly explore the influence of different clustering methods on the security of the watermarked maps.

References

[1] K.-T. Chang, *Introduction to geographic information systems*. McGraw-Hill, 2012.

[2] P. A. Longley, M. Goodchild, D. J. Maguire, and D. W. Rhind, *Geographic Information Systems and Science*, 3rd ed. John Wiley and Sons, 2011.

[3] J. Wu, F. Yang, and C. Wu, "Review of digital watermarking for 2d-vector map," in *IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 2013, pp. 2098–2101.

[4] T. Fenwick and I. Locks, Eds., *Copyright in The Digital Age: Industry IssIss and Impacts*, 1st ed. Wildy, Simmonds and Hill Publishing, 2010.

[5] D. I. Bainbridge, *Information Technology and Intellectual Property Law*, 6th ed. Bloomsbury Professional, 2014.

[6] P. Bhanuchandar, M. S. G. Prasad, and K. P. Srinivas, "A survey on various watermarking methods for gis vector data," *International Journal of Computer and Electronics Research*, vol. 2, no. 3, 2013.

[7] J.-G. Sun, G.-Y. Zhang, A.-H. Yao, and J.-P. Wu, "A reversible digital watermarking algorithm for vector maps," *International Journal of Network Security*, vol. 16, no. 1, pp. 40–45, January 2014.

[8] T. A. Abbas and M. J. Jawad, "Digital vector map watermarking: Applications, techniques and attacks," *Oriental Journal of Computer Science & Technology*, vol. 6, no. 3, pp. 333–339, September 2013.

[9] X.-J. Huo, K.-S. Moon, S.-H. Lee, T.-Y. Seung, and S.-G. Kwon, "Protecting gis vector map using the k-means clustering algorithm and odd-even coding," in *17th Korea-Japan Joint Workshop on Frontiers of Computer Vision*. IEEE, February 2011, pp. 1–5.

[10] S. Jianguo, K. Liang, and X. Songzhu, "Research of lossless digital watermarking technology," *Applied Mechanics and Materials*, vol. 333, pp. 1219–1223, July 2013.

[11] Y. Haowen, "Watermarking algorithm for vector point clusters," in *7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, Sept 2011, pp. 1–4.

[12] M. M. Raafat, H. M. Zawbaa, E. Al-Shammari, A. E. Hassanien, and V. Snasel, "Blind watermark approach for map authentication using support vector machine," in *Advances in Security of Information and Communication Networks*. Springer Berlin Heidelberg, 2013, pp. 84–97.

[13] J. Han, J.-G. Lee, and M. Kamber, *An Overview of Clustering Methods in Geographic Data Analysis*, 2nd ed. Taylor & Francis Group, LLC, 2009, ch. 7, pp. 150–187.

[14] J. Han, M. Kamber, and J. Pei, *Data Mining: Concept and Techniques*, 3rd ed. Waltham: Morgan Kaufmann, 2012.

[15] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, ser. The Morgan Kaufmann Series in Multimedia Information and Systems. Elsevier Science, 2007.

[16] X. Niu, C. Shao, and X. Wang, "A survey of digital vector map watermarking," *International Journal of Innovative Computing, Information & Control*, vol. 2, no. 6, pp. 1301–1316, December 2006.

[17] Q. Zhao, L. Sui, C. Wang, and X. Yin, "Publicly verify the integrity of the geographical data using public watermarking scheme," in *Geo-Informatics in Resource Management and Sustainable Ecosystem*. Springer Berlin Heidelberg, 2013, vol. 398, pp. 646–652.

[18] J. Nin and S. Ricciardi, "Digital watermarking techniques and security issues in the information and communication society," in *27th International Conference on Advanced Information Networking and Applications Workshops*, March 2013, pp. 1553–1558.

[19] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*, ser. Signal Processing and Communications. Taylor & Francis, 2004.

[20] W. Xun, H. Ding-jun, and Z. Zhi-yong, *A Robust Zero-Watermarking Algorithm for 2D Vector Digital Maps*, ser. Lecture Notes in Electrical Engineering. Springer Netherlands, 2012, vol. 107, ch. 56, pp. 533–541.

[21] ESRI, "Esri shapefile technical description," Environmental Systems Research Institute, Inc., 380 New York Street, Redlands, CA 92373-8100 USA, Tech. Rep., July 1998. [Online]. Available: www.esri.com/library/whitepapers/pdfs/shapefile.pdf

[22] S. Elhami, A. Saalfeld, and H. Kang, "Using shape analyses for placement of polygon labels," in *Esri International User Conference*, San Diego, CA, 2001. [Online]. Available: http://proceedings.esri.com/library/userconf/proc01/professional/papers/pap388/p388.htm

[23] E. Kolatch, "Clustering algorithms for spatial databases: A survey," 2001. [Online]. Available: http://citeseer.ist.psu.edu/436843.html

[24] W. Baiyan, W. Wei, and M. Dandan, "2d vector map watermarking based on spatial relations," *Proc. SPIE*, vol. 7285, pp. 728 532–728 537, 2008.

[25] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics Letters*, vol. 44, no. 13, pp. 800–801, 2008.

[26] R. Hamzaoui and D. Saupe, "Fractal image compression," in *Document and Image Compression*, M. Barni, Ed. CRC, 2006, ch. 6, pp. 145–177.

[27] L. Huang, W. Zhou, R. Jiang, and A. Li, "Data quality inspection of watermarked gis vector map," in *Geoinformatics, 2010 18th International Conference on*, June 2010, pp. 1–5.