Watermark Road Maps against Crop and Merge Attacks

Jacky Jiang

June 5, 2013

Outline

1 Introduction

Digital road map

A digital road map, $\mathcal{M}$, is a view of a graph G with a set of vertices V and a set of edges E. $\mathcal{M}$ consists of a set of k roads R_i , where each road is a polyline represented by a sequence of vertices in the form of (x, y) coordinates in a geographical coordinate.

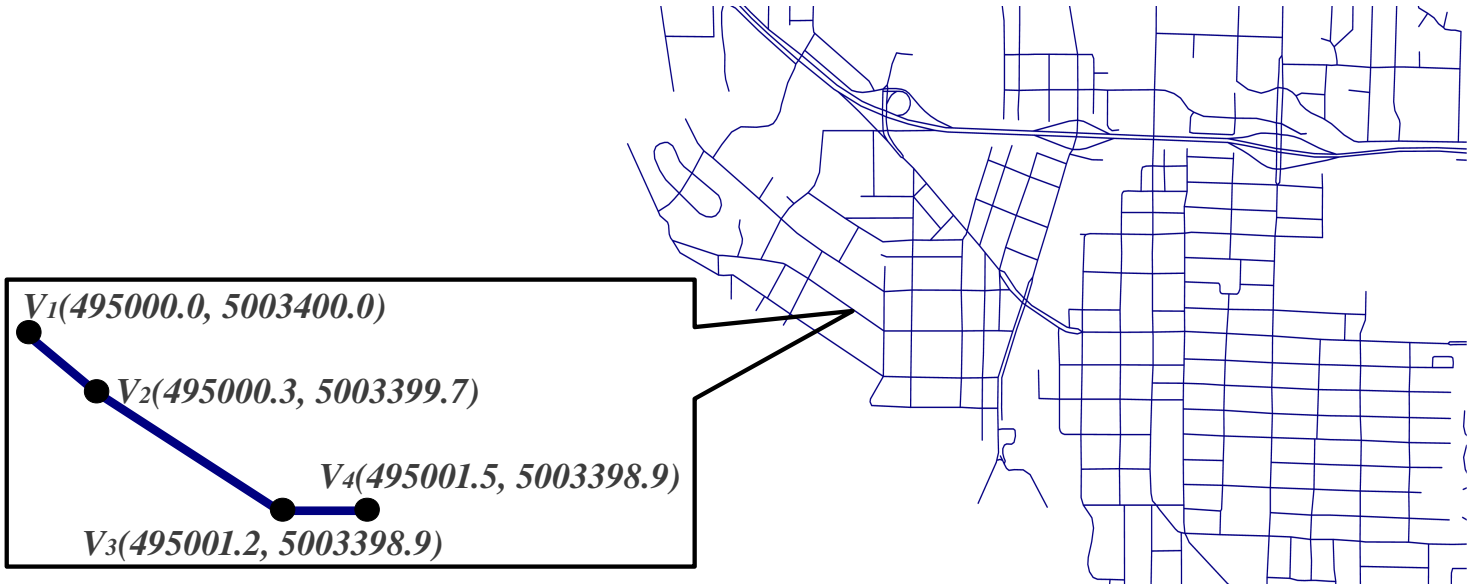


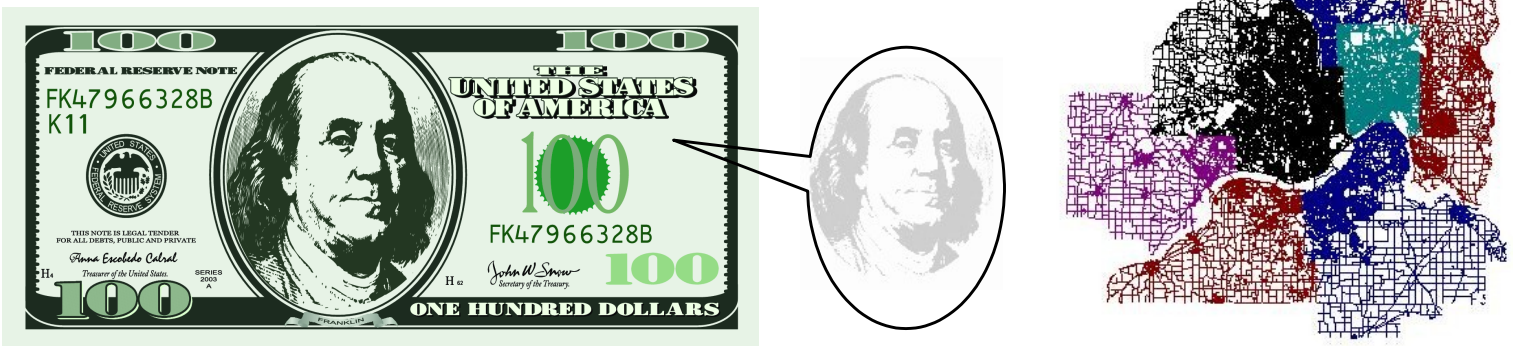
Figure 1: Digital Road Map

Motivation

Generally GIS vector maps are extremely expensive to produce. However, on the other hand, the digital nature of any vector map leaves it vulnerable to being copied and resold by a 3rd party without permission.

Digital watermarking is an important technique to protect the copyrights of digital products.

A watermark is small amount of digital noise embedded into the digital representation of the products.



(a) Note

(b) Digital map

Figure 2: Watermark

The standard watermarking framework for digital road maps adopts a two-step approach – the watermark **insertion** and **detection** algorithms.

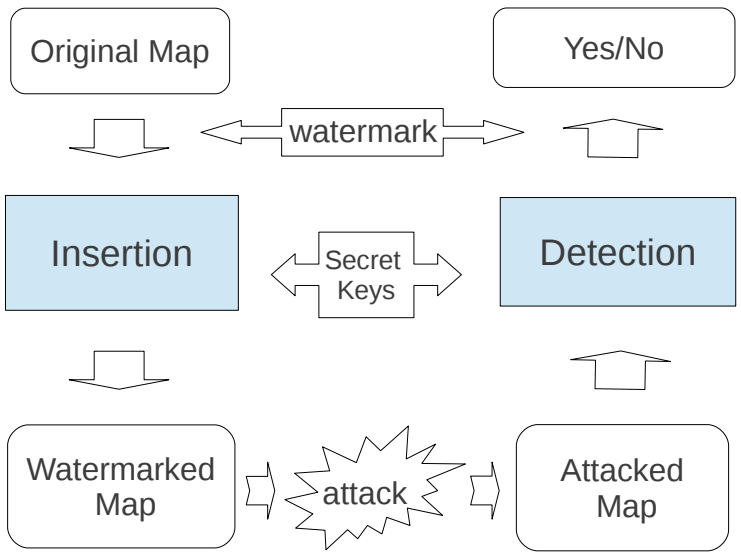


Figure 3: General Watermarking Framework For Road Maps

2 Problem Definition

Two specifical attacks

1. Noise attack
2. Reorder attack
3. Simplify/Addition attack
4. Crop attack
5. Merge attack

Crop Attack

An attacker crops a geographical region from the watermarked map and use it as if its a new map. We define cropping attack as:

$$Crop(M) = \{subseg(R) \mid R \in M'\}$$

where $M' \subseteq M$ and $subseq(R)$ returns a subsequence of road R.

Merge Attack

An attacker crops parts from different maps and piece them together to make a new map.

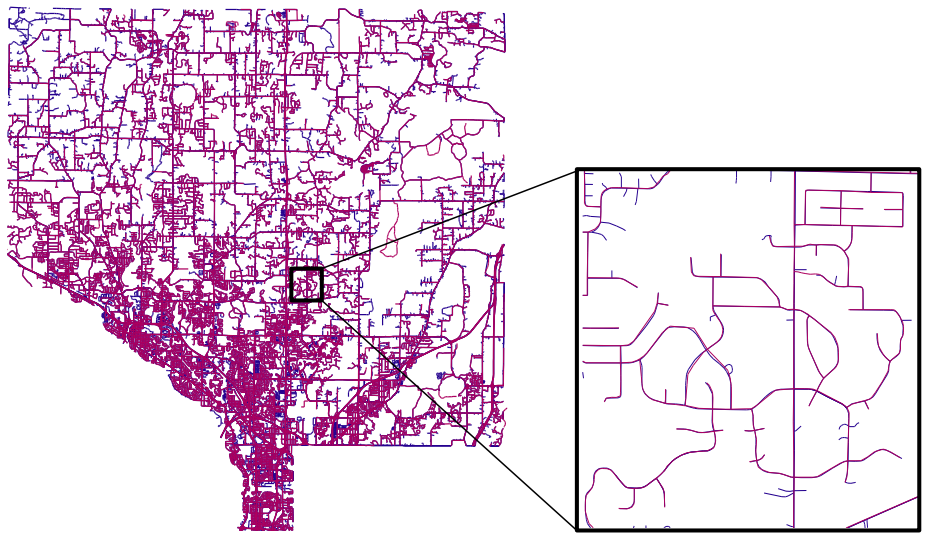


Figure 4: Overlap of Two Maps

Watermark Road Maps against Crop and Merge Attacks

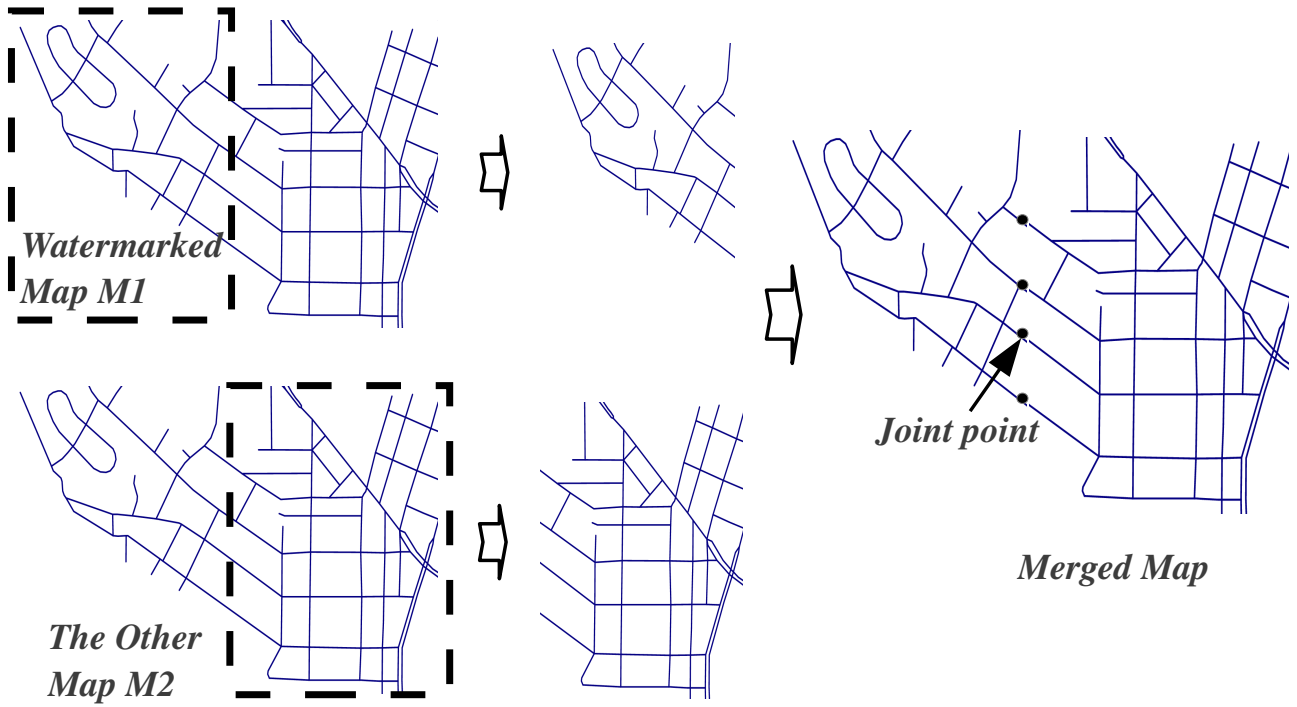


Figure 5: Merge Attack

3 Our approach

Secret Keys

They should only be known to the map owner.

Secret Grid

The **secret grid** G is a coordinate gird.

- An origin, certain position with precise latitude and longitude
- A step size, defines the granularity of the grid

$$G = \{Origin(x_0, y_0), Step\}$$

Secret MBR

Given an original map to watermark, it can be laid out on the master grid, according to the coordinates of the vertices in the map. The **secret MBR** is then the smallest rectangle which coincides with the grid lines and completely encloses the whole map.

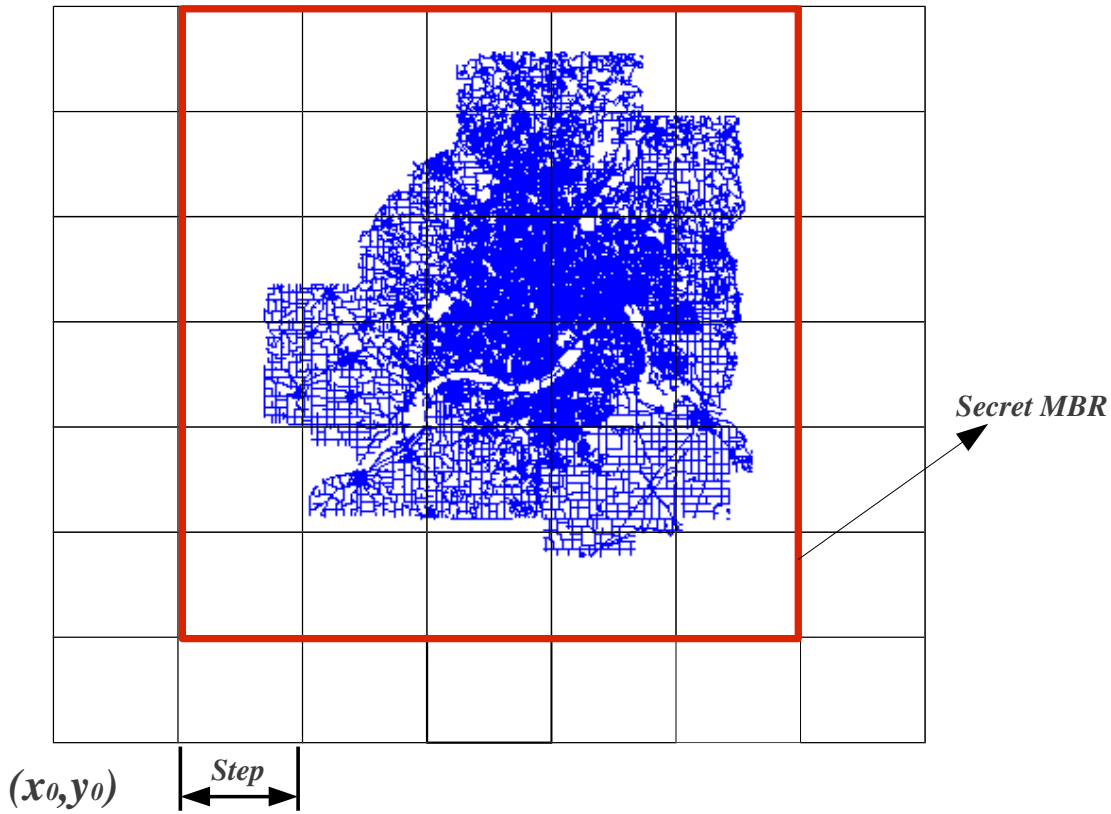


Figure 6: Master Grid and MBR

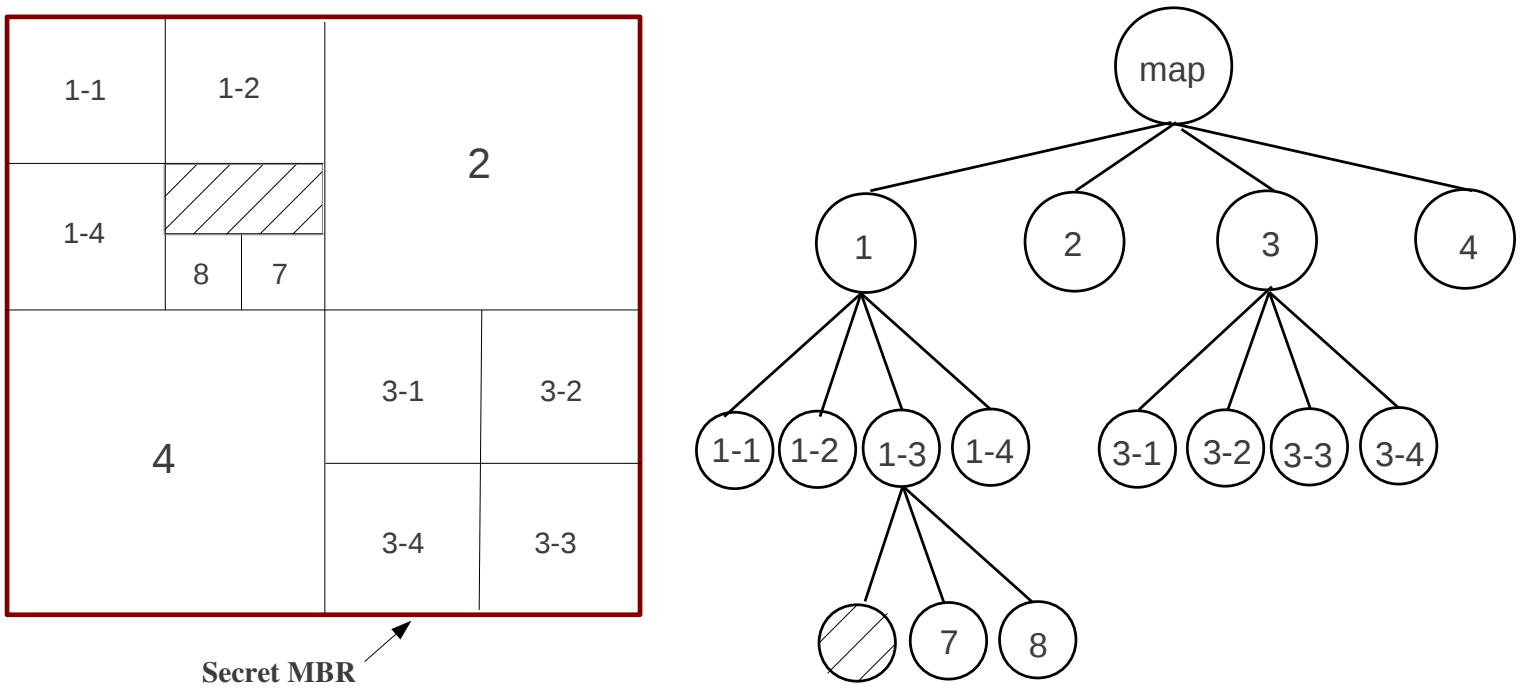
Secret Square Size

The **secret square size** is an integer number l that determines the size of a square box that we used to select a small neighborhood of road segments from which to compute the watermarks.

Partition

It recursively partitions the space bounded by the secret MBR for a map into small regions using a modified quad-tree structure (known as MQtree) according to the density of the roads. Each node in MQtree represents a sub-region of the map.

- if roads length within one region $> 4\theta$, partition it into 4 sub-regions.
- if roads length within one region $< \theta$, merge it with neighbor.



(a) Map (b) MQtree

Figure 7: MQtree

The time complexity of partition is

$$\mathcal{T} = O(|E| \log \mathcal{L}),$$

where $\mathcal{L}$ is the total length of all roads in map M.

Insertion

- The watermark is inserted into a road vertex $P(x, y)$ which is closest to the center of the sub-region.
- The algorithm draws a square box of size l centered at $P(x, y)$.
- Let sl be the length of road segments that intersects the square box
- Let j be a value hashed from sl.
- Set the j^{th} least significant bit in x coordinate of P to "1".

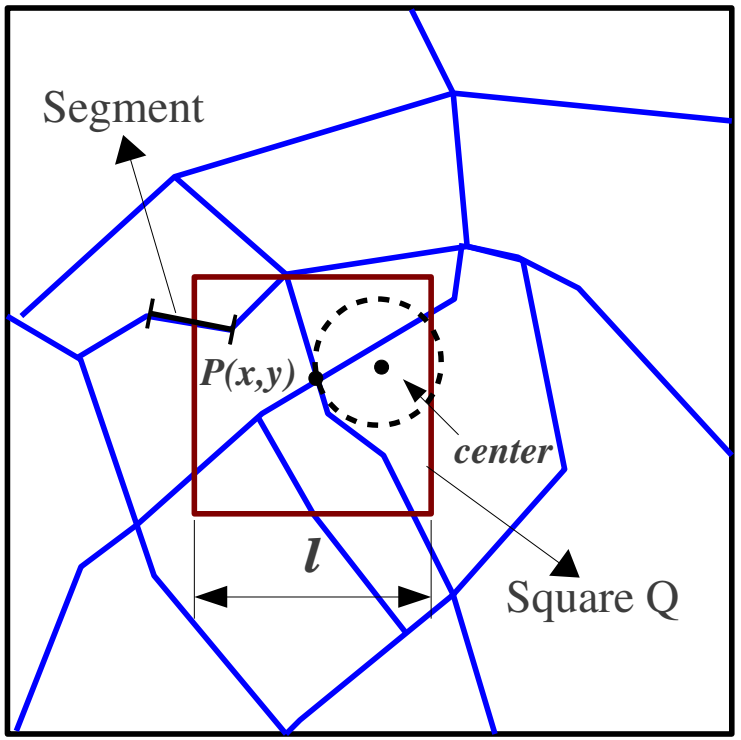


Figure 8: Insertion Strategy

The hash function in this algorithm must guarantee to hash to the same value before and after watermarking. Even if the watermarked map is attacked, the hash function must still hash to the same j.

J_{max}: the biggest LSB that can be changed within the possible distortion of a single end point.

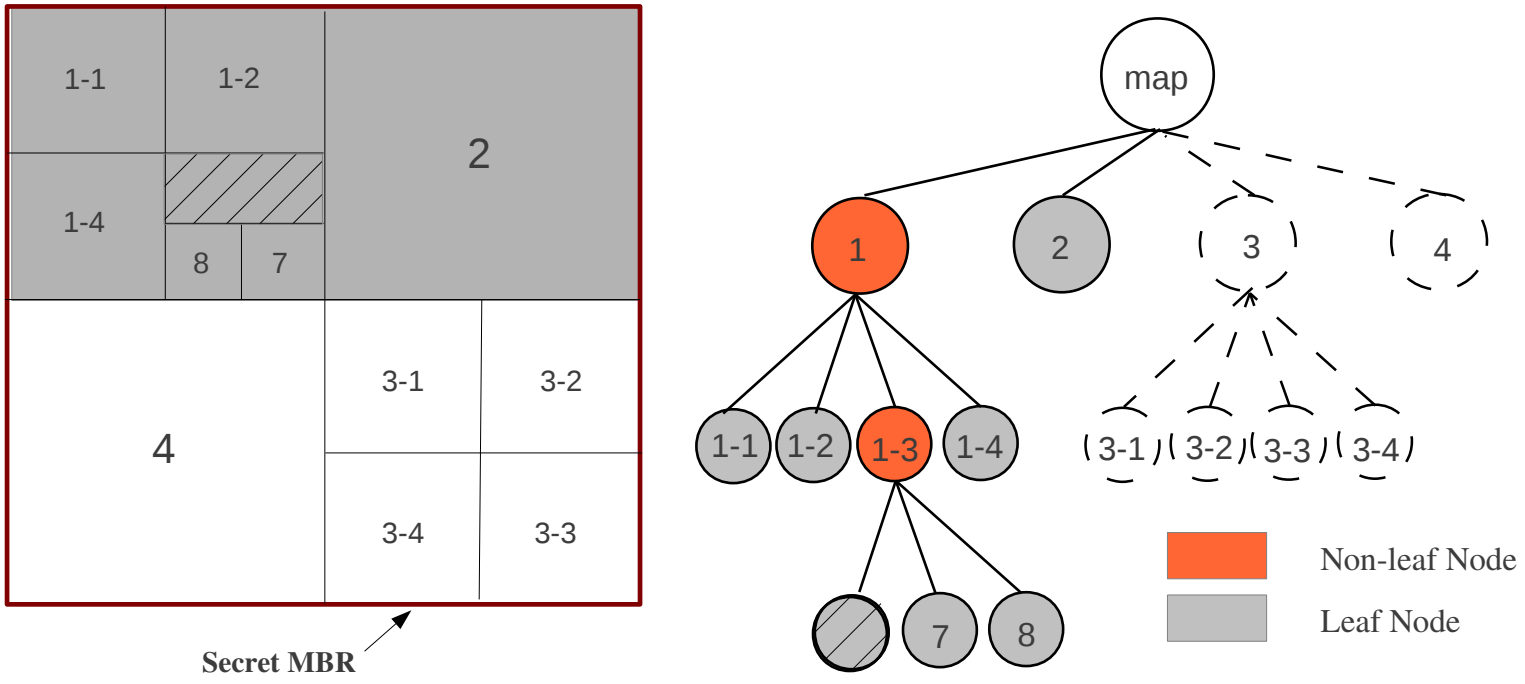
$$j = Hash(Trans(sl), k)$$

$$\text{where} \quad Trans(x) = x \wedge \underbrace{11\ldots1\overbrace{00\ldots0}^{J_{max}}}_{bit\_length(x)}$$

Detection

We partition the map using the same strategy as insertion. Then we select the data points closest to the center of the regions to detect whether the watermark exists there.

Each sub-region which is detected to contain a watermark casts a vote which collectively contributes to the final decision of whether a larger area is watermarked as a whole.



(a) Map (b) MQtree

Figure 9: Detection Strategy

Confidence

We define the detection confidence of detection as

$$conf = 1 - \sum_{i=n}^{N} \binom{N}{i} \left(\frac{1}{2}\right)^{N-i} \left(\frac{1}{2}\right)^{i} \tag{1}$$

where N is total number of leaf nodes and n is number of leaf nodes that match.

Theorem (Detection Confidence)

Given a map M with total length $\mathcal{L}$ and an algorithm threshold θ, the minimum detection confidence for M:

$$conf_{min}(M) = 1 - \sum_{i=\lceil \rho\mathcal{L}/4\theta \rceil}^{\lceil \mathcal{L}/4\theta \rceil} \binom{\lceil \mathcal{L}/4\theta \rceil}{i} \left(\frac{1}{2}\right)^{\lceil \mathcal{L}/4\theta \rceil}$$

where ρ is the ratio between the number of leaf nodes that match and the total number of leaf nodes in M.

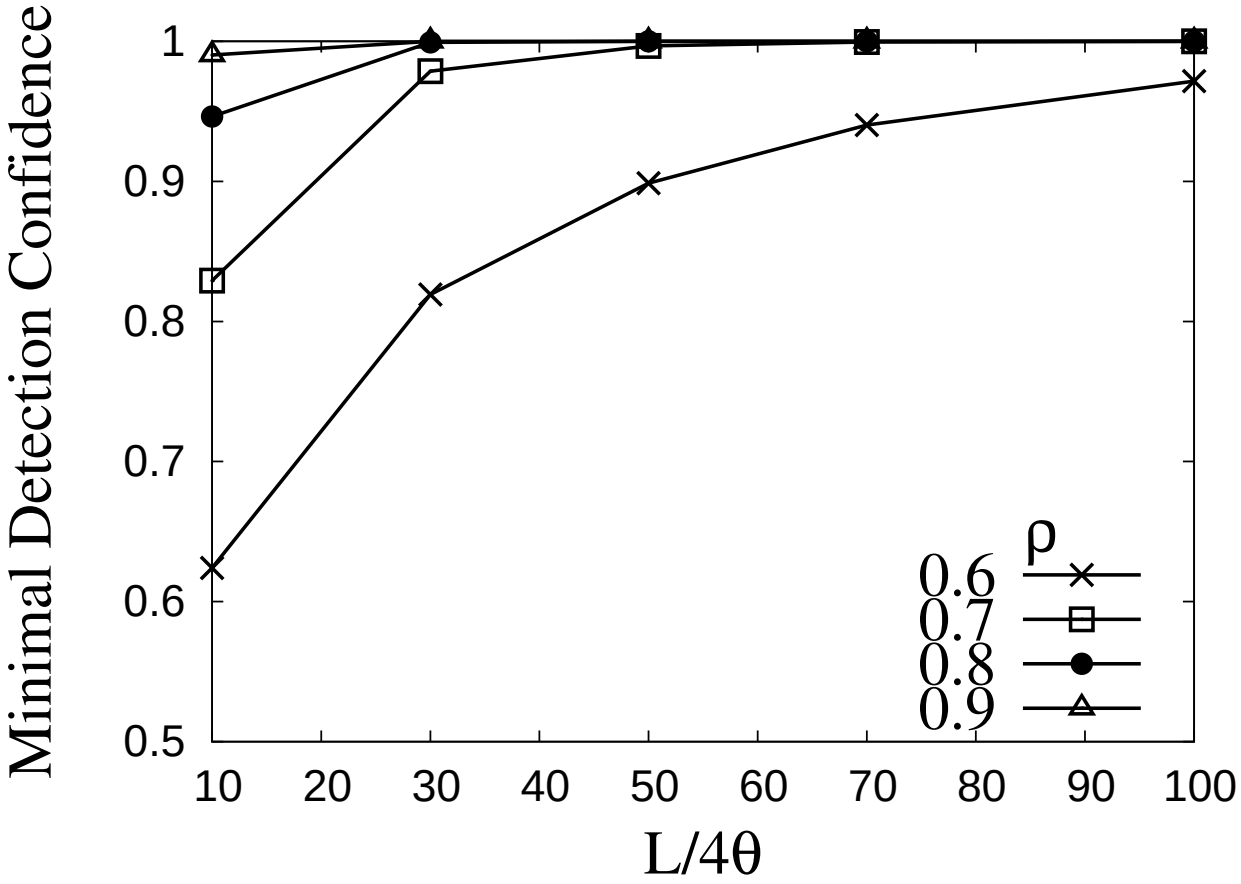


Figure 10: Accuracy of Detection

4 Experiment Results

We implement and test the performance of the algorithms under different potential attacks. The map data is from two different sources, MN/DOT[1] and United States Census Website[2].

We compare the proposed approach with two other watermarking algorithms proposed by Pu[3] and Voigt[4]. Both methods are blind watermarking algorithms and provide some resistance to crop attack.

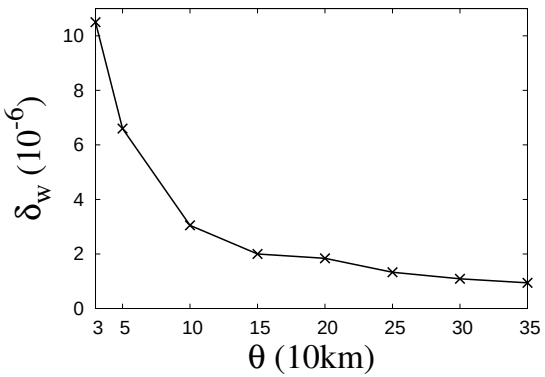
Performance under Cropping

The partition criterion θ has a large influence on the result of our watermark algorithm. We also design a series of experiments to figure out the influence of the θ.

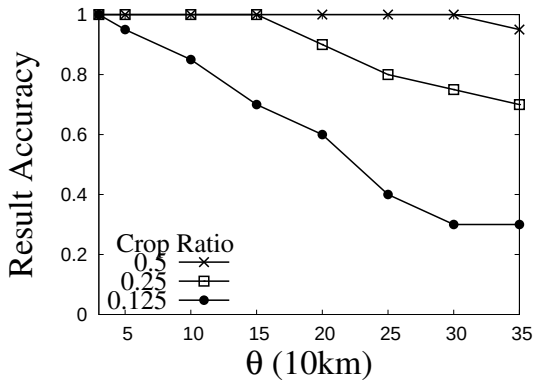
County	Total	Match	Confidence	Result
Anoka	55	51	1.000	positive
Carver	28	23	0.999	positive
Dakota	68	61	1.000	positive
Hennipin	141	137	1.000	positive
Ramsey	58	56	1.000	positive
Scott	29	25	0.999	positive
Washington	46	43	1.000	positive

Table 1: Crop Attack Detection

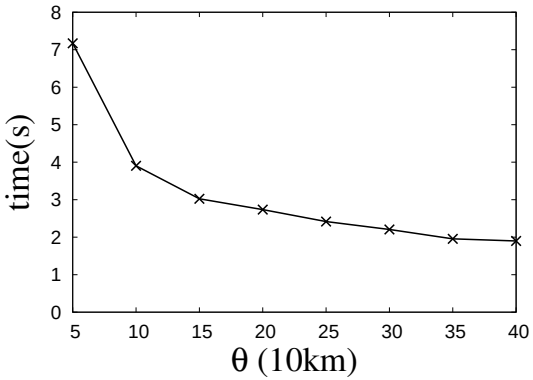
Watermark Road Maps against Crop and Merge Attacks



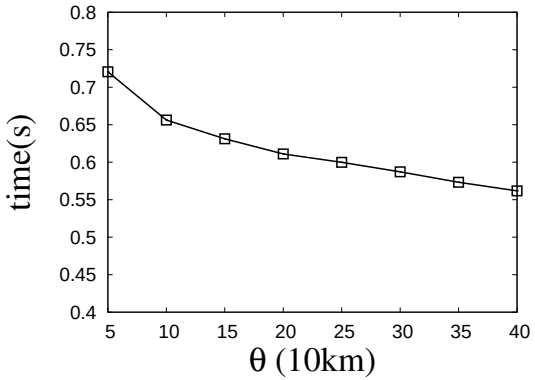
(a) Distortion



(b) Different Crop Ratio

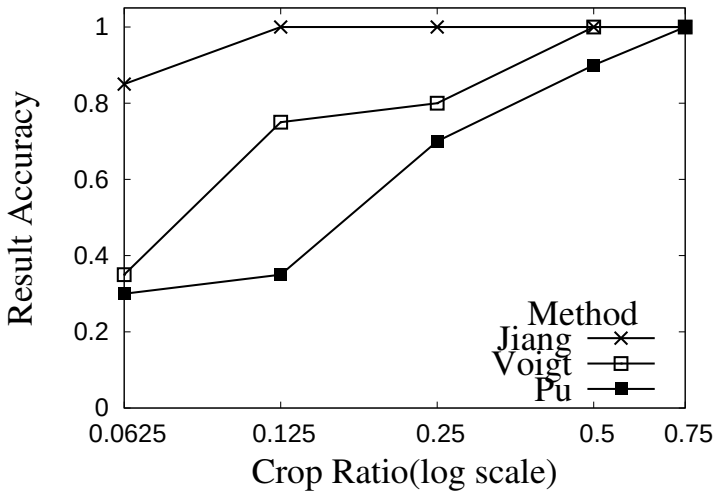


(c) Time of Insertion

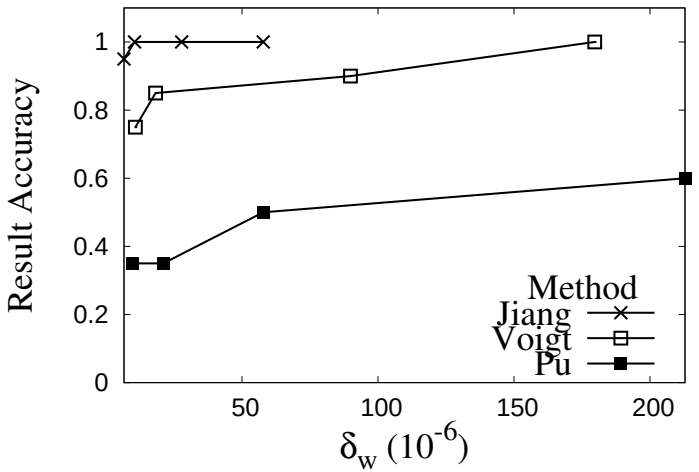


(d) Time of Detection

Watermark Road Maps against Crop and Merge Attacks



(e) Performance under Cropping



(f) Accuracy in $1/8$ crop ratio

Figure 11: Performance under Crop Attack

Performance under merging

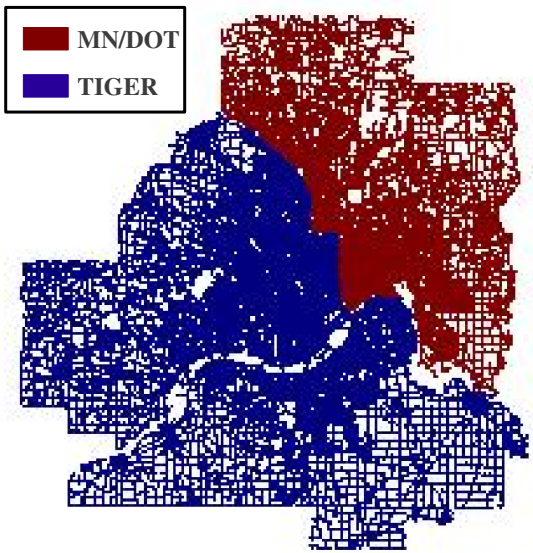


Figure 12: Merge Attack

Detection Result

Region	Total	Match	Confidence	Result
1	147	122	1.000	positive
2	12	11	0.997	positive
3	23	18	0.994	positive
4	15	12	0.983	positive

Table 2: Merge Attack Detection

Watermark Road Maps against Crop and Merge Attacks

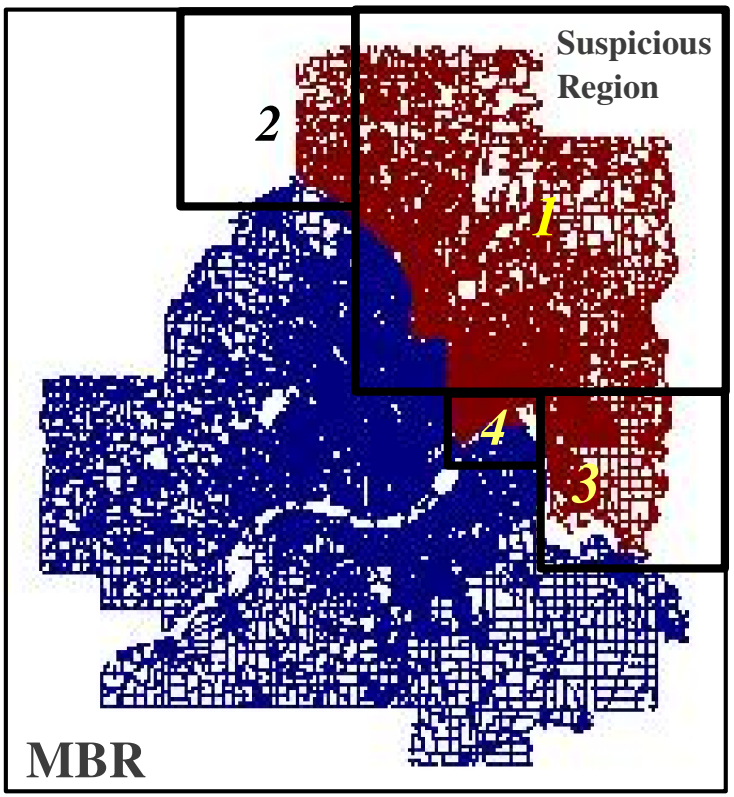
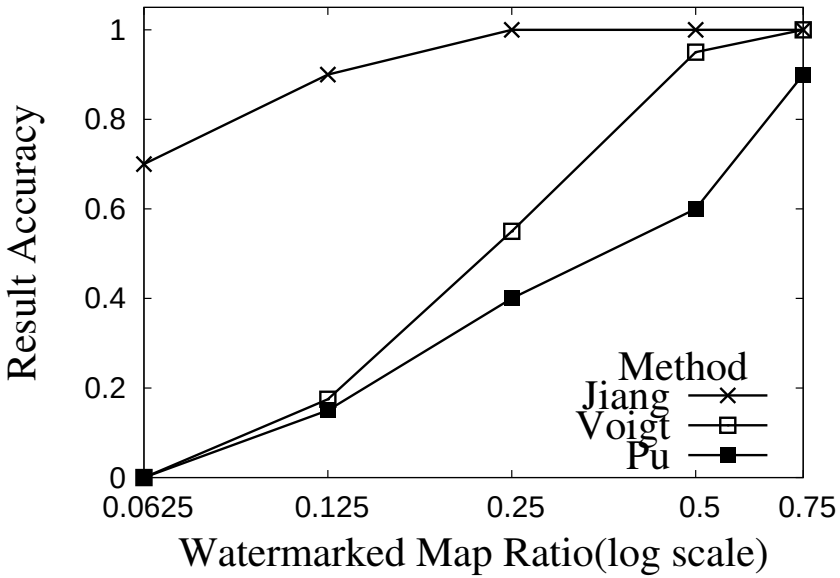
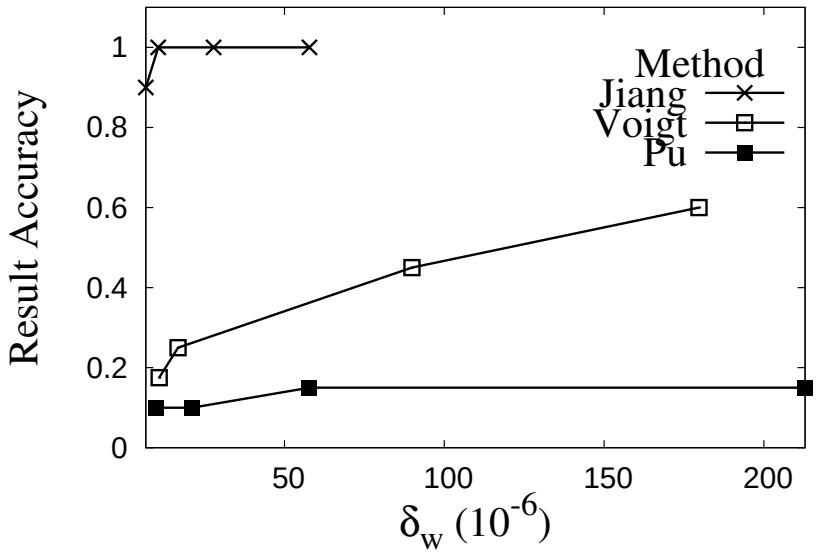


Figure 13: Merged Map

Watermark Road Maps against Crop and Merge Attacks



(a) Performance under merging



(b) Accuracy in $1/8$ merge ratio

Figure 14: Performance under Merge Attack

5 Conclusion

1. We proposed a new blind watermarking scheme for digital vector road maps.

2. The algorithm dynamically partitions a given map according to road density and inserts one-bit watermarks to one of the least significant bits of points.

3. Our preliminary evaluation shows that this algorithm is resilient to massive crop and merge attacks.

References

[1] MN/DOT. `http://www.dot.state.mn.us/`.

[2] TIGER. `http://www.census.gov/`.

[3] Yu-Chi Pu, Wei-Chang Du, and I-Chang Jou. Toward blind robust watermarking of vector maps. In *ICPR (3)*, pages 930–933, 2006.

[4] Michael Voigt and Christoph Busch. Feature-based watermarking of 2d-vector data. In *SPIE,Security and Watermarking of Multimedia Content.Santa Clara,*, pages 359–366, 2003.

Questions?