

© IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

WATERMARKING OF 2D VECTOR GRAPHICS WITH DISTORTION CONSTRAINT

Stefan Huber, Roland Kwitt, Peter Meerwald, Martin Held, Andreas Uhl

Dept. of Computer Sciences, University of Salzburg,
Jakob-Haringer-Str. 2, A-5020 Salzburg, Austria
Email: {shuber, rkwitt, pmeerw, held, uhl}@cosy.sbg.ac.at

ABSTRACT

We study the watermarking of 2D vector data and introduce a framework which preserves topological properties of the input. Our framework is based on so-called maximum perturbation regions (MPR) of the input vertices, which is a concept similar to the just-noticeable-difference constraint. The MPRs are computed by means of the Voronoi diagram of the input and allow us to avoid (self-)intersections of input objects that might result from the embedding of the watermark. We demonstrate and analyze the applicability of this new framework by coupling it with a well-known approach to watermarking that is based on Fourier descriptors. However, our framework is general enough such that any robust scheme for the watermarking of vector data can be applied.

1. MOTIVATION

Watermarking is a technology to enable copyright protection by embedding an imperceptible, yet detectable signal in digital content [1]. Watermarking research has primarily focused on raster data (audio and video content). However, increasingly more complex models of computer-aided design (CAD) or huge maps and infrastructure data stored in geographical information systems (GIS) also constitute valuable digital assets and make the protection of vector data more important.

When embedding watermark information in a collection of geometric primitives not only perceptual constraints have to be met but also geometrical properties must be preserved: For example, the banks of a river in a geographic map should not cross due to the embedding. Similarly, the pads of a printed circuit board should not overlap afterwards. This is particularly important for industrial 2D vector data, where copyright protection has not received much attention.

Watermarking of vector data has been proposed for 2D polygons and 3D meshes. In this work, we focus on 2D polygonal data. Zheng et al. [2] provide an overview of the state-of-the-art in vector watermarking and Li et al. [3] review technical and legal copyright issues with watermarking of geo-spatial datasets.

Although fidelity of the watermarked data is generally considered (if only by visual inspection), distortion constraints and preservation of geometrical properties of the watermarked data received only very limited attention so far: Ohbuchi et al. [4] report an acceptable error of 75 cm in the real world on a 1:2500-scale geographical map. Doncel et al. [5] consider polygonal chains sharing a number of points such as the border of neighboring countries; the chains must be kept coincident at the corresponding locations after watermark insertion.

2. OVERVIEW

We introduce a general distortion constraint framework for geometric data watermarking which preserves essential geometric properties after the embedding: it guarantees that no line segments cross due to vertex perturbation. Hence, the input topology is preserved. For each vertex we compute a radius which bounds the allowed perturbation. By deliberately choosing a smaller radius, the error bound mentioned by Ohbuchi et al. [4] can be implemented.

Let us define a (simple) polygonal chain as a (possibly closed) sequence of adjacent straight line segments where non-consecutive segments are not allowed to intersect. We represent such a polygonal chain by the series of its vertices; if the chain is closed, the first and last vertex coincide. The input of our framework is a set of polygonal chains where two chains may only intersect at their endpoints.

The proposed framework consists of three parts as depicted in Fig. 1: (i) a geometric pre-processing step computing the so-called maximum perturbation region (MPR) of each input vertex; (ii) the watermark embedding process; and (iii) the correction step which outputs the watermarked polygonal chains subject to the distortion constraint.

The remainder of this work is organized as follows. In Section 3, we describe the computation of the maximum perturbation regions. We exemplarily show how to apply the MPR distortion constraint on a well-known vector graphics watermarking approach based on Fourier descriptors [6, 5] in Section 4. In Section 5, we investigate the impact on detection performance when adopting the MPR framework and finally summarize our results in Section 6.

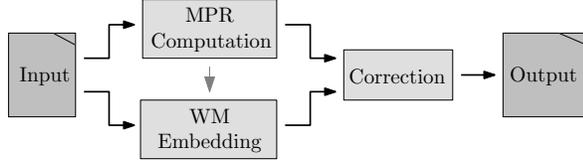


Fig. 1: The framework consisting of three processing steps.

3. MAXIMUM PERTURBATION REGIONS

Let us consider the set $V = \{v_1, \dots, v_n\}$ of vertices of the input chains and the set E of straight-line segments from the input chains. As we already mentioned, segments of E do not cross but eventually meet at their endpoints. Strictly speaking, (V, E) forms a so-called planar straight-line graph. Using this terminology, we call two segments of E sharing a common vertex of V adjacent. Two vertices of V connected by a segment of E are called neighbors. The task is to determine for any given vertex $v \in V$ a region $R(v) \subset \mathbb{R}^2$ with $v \in R(v)$ such that as long as vertices v are displaced within their $R(v)$ (and the incident segments of E accordingly), the resulting set E of segments remains crossing-free. In this section we present a procedure that computes the regions $R(v_k)$ based on the so-called (generalized) Voronoi diagram of G .

The basic idea regarding MPRs is the following: if all input sites s are perturbed only within the interior of their Voronoi cell $\mathcal{VC}(s)$ then they will not intersect, since Voronoi cells of different sites do not overlap. In our application, the vertices are perturbed due to watermark embedding and the incident line segments must not intersect.

Voronoi diagrams of points and line segments are a fundamental concept in the field of computational geometry. They have been well-studied in the last three decades and lead to convenient solutions for a large number of geometric problems, including tool path planing, computing contour parallel offset curves, shape representation, and the like, see [7]. The input to the Voronoi algorithm is a set of input sites given by points and line segments that only intersect at their boundary points. Roughly speaking, the Voronoi diagram decomposes the plane into so-called Voronoi cells $\mathcal{VC}(s)$ around each input site s such that any point in a given Voronoi cell $\mathcal{VC}(s)$ is at least as close to s as to any other input site, cf. Fig. 2. In other words, the closest neighbor of a point in $\mathcal{VC}(s)$ within all input sites is s . For technical reasons it is generally assumed that for any input line segment the endpoints are input sites as well. For a precise definition and an overview of the Voronoi diagram of points and line segments, see [7].

Let us rephrase the desired property which leads to the MPR distortion constraint for watermark embedding. We want to define regions of allowed perturbation ('hoses') around the line segments which, on one hand, should be as large as possible to achieve watermark embedding capacity but, on the other hand, must satisfy the requirement that

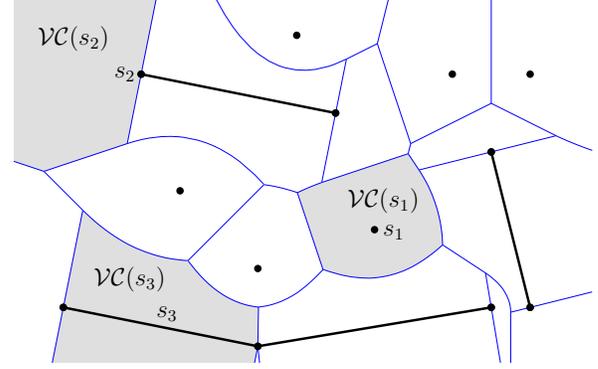


Fig. 2: The Voronoi diagram (thin lines) of the input line segments and points (bold). Three Voronoi cells $\mathcal{VC}(s_1)$, $\mathcal{VC}(s_2)$ and $\mathcal{VC}(s_3)$ have been shaded in light-grey.

hoses of non-adjacent segments do not overlap, cf. Fig. 3b. This constraint, however, is strongly related to the computation of offset curves [8] which are easily computed using the Voronoi diagram. We construct MPRs in two phases:

Phase 1 We consider for each vertex v the incident line segments l_1, \dots, l_m and denote by \hat{l}_j the half of l_j which has v as endpoint. We denote by $|l|$ the length of the line segment l . For $r > 0$ and a segment l we define the set $B_l(r) \subset \mathbb{R}^2$ as the rectangle with width $2r$ and length $|l|$ such that l is its center-line. Further, we denote by $D_v(r)$ the (open) disk with center v and radius r . Then we determine for each vertex v the maximum $t_v \leq \min_{1 \leq j \leq m} |\hat{l}_j|$ such that

$$D_v(t_v) \cup \bigcup_{j=1}^m B_{\hat{l}_j}(t_v) \subseteq \mathcal{VC}(v) \cup \bigcup_{j=1}^m \mathcal{VC}(l_j). \quad (1)$$

We interpret t_v as the radius of the disk $D_v(t_v)$, cf. Fig. 3a. The actual computation of t_v can be done by simply traversing the boundary of the corresponding Voronoi cells and by computing the minimum distances to their input sites. This is a standard task also applied when computing offset curves, see [8].

Phase 2 We denote by $\hat{v}_1, \dots, \hat{v}_m$ the vertices adjacent to v and compute the value

$$r_v := \min\{t_v, t_{\hat{v}_1}, \dots, t_{\hat{v}_m}\} \quad (2)$$

for each vertex v . The resulting perturbation region $R(v)$ for a vertex v is then given by the disk

$$R(v) := D_v(r_v). \quad (3)$$

In Fig. 3b we illustrate for all vertices v their regions $R(v)$ as dark-grey disks. Furthermore, we shaded the areas which

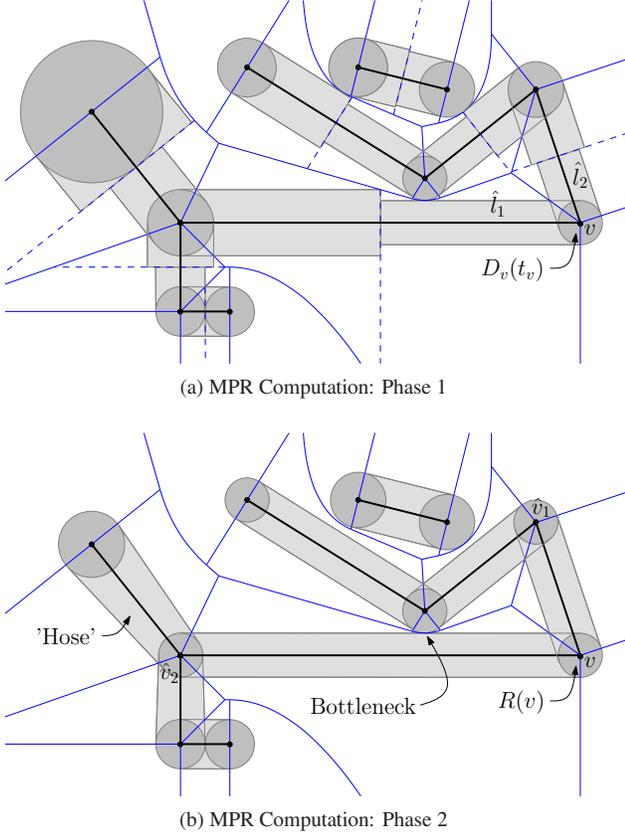


Fig. 3: The two phases of determining the MPRs. The bold lines depict the input, thin lines illustrate the Voronoi diagram of the input. (a) For each vertex v we shaded $D_v(t_v)$ dark-grey. In light-grey we shaded the corresponding rectangles $B_{\hat{l}_k}(t_v)$ for the adjacent half segments \hat{l}_k of v . (b) The dark-grey disks depict the MPRs and the light-grey regions the area ('hose') in which the segments after perturbation may lie.

are enclosed by the disks of two adjacent vertices and their tangents. These areas ('hoses') depict the region in which a segment may lie after perturbation. Since the light-grey areas of Fig. 3b are contained in those of Fig. 3a it follows that line segments can not intersect if all vertices remain within their corresponding region $R(v)$ after perturbation. In order to avoid touching hoses at bottlenecks such as the one shown in Fig. 3b, one may multiply all r_v by a factor $1 - \epsilon$ with $0 < \epsilon \ll 1$.

The question concerning the maximality of the perturbation region arises. Obviously, the presented scheme does not lead to the maximal possible radii of the corresponding disks $R(v)$, cf. the leftmost vertex in Fig. 3b. One could add an additional phase in which disks are increased within their corresponding Voronoi cells if possible. For the matter of simplicity we skip this step in our actual implementation.

Our method is fair in the following sense: A bottleneck is a locus in the Voronoi diagram where a local minimum of

the distance to the input sites occurs. Roughly speaking, our method allocates the same radii for the affected vertices at those bottlenecks and thus achieves fairness, ignoring the obvious case that the radii are actually limited by the length of the adjacent line segments, cf. Fig. 3b.

Voronoi diagrams can be computed in $O(n \log n)$ time in theory. In our implementation, we compute Voronoi diagrams using the software package VRONI [7] which provides an expected runtime complexity of $O(n \log n)$ and has been proven to be fast and stable in practice. The Voronoi diagram of the *Carp* dataset (24134 vertices) presented in Fig. 5 can be computed in about half a second on a middle-class PC. Once the Voronoi diagram is available, we traverse the corresponding Voronoi cells in phase 1 and visit the neighbors of each vertex in phase 2. Note that due to Euler's formula for planar graphs, the Voronoi diagram is of linear size and hence both phases can be done in $O(n)$ time. To sum up, the time complexity of the entire MPR computation is in $O(n \log n)$. In practice, this allows to process the *Carp* dataset in a few seconds. The MPR computation has to be done once for each input graph. The complexity of the correction step, which is critical for fingerprinting applications, is discussed in the next section.

4. WATERMARKING WITH DISTORTION CONSTRAINT

For image raster data, Podilchuk et al. [9] pioneered the concept of perceptual shaping of the watermark according to the just-noticeable-difference (JND) value for each coefficient. The JND gives the approximate amount of modification a coefficient can tolerate before the change becomes visible according to a perceptual model. Perceptual models are often formulated in a transform domain such as the DCT [10] or DWT [11] domain. For vector data, not only perceptually noticeable changes must be avoided but also geometric properties must be preserved. The MPR constraint presented in Section 3 is given in the coordinate domain.

Watermark embedding is either performed directly in the coordinate domain or in some transform domain: Ohbuchi et al. [4] utilize the mesh-spectral domain, Solachidis et al. [6] propose the complex DFT domain due to its invariance against a number of attacks (translation, rotation, scaling). An advantage of transform-domain approaches is that selection of mid-frequency coefficients for watermarking provides a convenient way to embed information in a significant portion of the host data without causing severe, noticeable perturbation. For transform-domain watermarking methods, the MPR distortion constraint has to be imposed after embedding, see Fig. 1. Vector data watermarking techniques operating in the coordinate domain can incorporate the MPR constraint directly in the ways indicated by Podilchuk et al. [9].

Let \mathbf{v} denote a polygonal chain as a series of n vertices $v_k = (v_k(x), v_k(y))$ which can be considered as a complex signal $\mathbf{x} : x_k = v_k(x) + i \cdot v_k(y)$ with $1 \leq k \leq n$ where

the real and imaginary components are the x and y coordinates of the 2D vertices; we adhere to the notation introduced by Solachidis et al. [6]. Let \mathbf{x}' denote the watermarked vertex data. In case the watermarked vertex x'_k lies outside the disk defined by the corresponding MPR with radius r_{v_k} , i.e. $|x'_k - x_k| > r_{v_k}$, we project x'_k on the MPR boundary thereby producing a new watermarked vertex x''_k subject to the geometric distortion constraint:

$$x''_k = x_k + \frac{r_{v_k} \cdot (x'_k - x_k)}{|x'_k - x_k|} \quad (4)$$

This process is illustrated in Fig. 4. Note that polygonal chains intersecting other chains are avoided this way.

Depending on the watermarking application, the MPR constraint can be used in two ways: (i) either all vertices outside their MPR are projected on their MPR boundary, or, (ii) only those vertices which actually cause line segments to cross are corrected. In the first case, the correction process can be easily done in $O(n)$ time, n being the number of vertices. The second case requires to solve the line segment intersection problem which consumes in the worst-case up to $O(n^2)$ time. The latter case, which we refer to as *conditional MPR* (cMPR), clearly imposes a less stringent distortion constraint, but is probably not suitable for real-time watermark or fingerprinting applications due to runtime complexity.

Using the well-known vector graphics watermarking approach based on Fourier descriptors [6, 5] as an example, a multiplicative spread-spectrum watermark \mathbf{w} can be applied on selected complex DFT coefficient magnitudes $|\tilde{\mathbf{x}}|$ of signal \mathbf{x} ,

$$|\tilde{x}'_k| = |\tilde{x}_k|(1 + \alpha w_k), \quad (5)$$

where $\alpha > 0$ is the embedding strength and $w_k \in \{-1, 1\}$ with equal probability generated by a pseudo-random number generator seeded with a secret key. Watermarking the DFT magnitudes provides invariance to geometric operations such as scaling, translation, rotation, change in traversal starting index, and mirroring.

Given a received polygonal chain \mathbf{z} , a blind watermark detector has to decide without reference to the original data between the two hypothesis

$$\begin{aligned} \mathcal{H}_0 &: \text{no/other watermark} \\ \mathcal{H}_1 &: \text{watermarked with } \mathbf{w}. \end{aligned} \quad (6)$$

In [6], detection using linear correlation (LC) with test statistic $\rho_{LC} = \frac{1}{n} \sum_{k=1}^n |\tilde{z}_k| w_k$ was proposed. Doncel et al. [5] improved the detector by noting that the DFT coefficient magnitudes can be accurately modeled by a Rayleigh distribution. The derived (estimate-and-plug [12]) likelihood-ratio test (LRT) conditioned on the host signal model decides \mathcal{H}_1 in case

$$\rho_{LRT} = \sum_{k=1}^n |\tilde{z}_k|^2 \frac{(1 + \alpha w_k)^2 - 1}{2\hat{\beta}_k^2 (1 + \alpha w_k)^2} > T_\rho \quad (7)$$

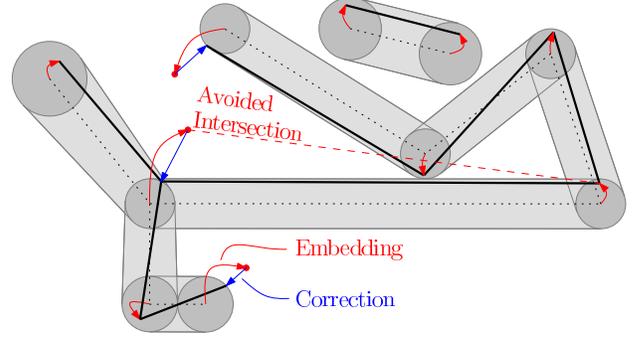


Fig. 4: Correcting the watermarked vertex data subject to the MPR distortion constraint.

where $\hat{\beta}_k = \sqrt{\frac{1}{2(2p+1)} \sum_{i=k-p}^{k+p} |\tilde{z}_i|^2}$ is the maximum likelihood estimate of the Rayleigh distribution parameter within a sliding window of size $2p + 1$ and T_ρ denotes the detection threshold (see Section 5). It can be argued that according to the Central-Limit theorem both detection statistics follow a Normal distribution for reasonably large data size, say $n > 1000$. Due to MPR correction after watermark embedding, the watermark power is partially damped. MPR correction can be interpreted as a noise source on the watermark signal. In Section 5 we assess this degradation experimentally using the detectors presented above.

5. EXPERIMENTAL RESULTS

Vector images used for this work are freely available and can be downloaded from <http://openclipart.org> and <http://openstreetmap.org> in SVG format. Python source code for the watermarking schemes is available at <http://www.wavelab.at/sources>. Experiments have been performed on a number of datasets of different size and type. Due to limited space, we can present only two representative examples here, see Figs. 5 and 7.

Figure 5 shows the watermarked *Carp* vector graphics consisting of 24134 vertices in 836 polygonal chains. The watermarking algorithm based on [6, 5] selects polygonal chains with 200 or more vertices and hence modifies 4890 vertices by embedding with strength $\alpha = 0.5$. Eventually, 1589 vertices were subjected to MPR correction.

The ρ_{LC} detection statistic histograms under \mathcal{H}_0 and \mathcal{H}_1 (with and without MPR correction) from 1000 experiments with different watermarking keys can be observed in Fig. 8 and confirm the assumption of a Normal distribution. In Fig. 6 we zoom in on the tail end of the *Carp*'s ventral fin. On the left, Fig. 6a, we show the original vector data with the MPRs superimposed. In the middle, Fig. 6b, is the watermarked data where we can observe that polygonal chains are intersecting (with themselves and other chains). On the right, Fig. 6c,

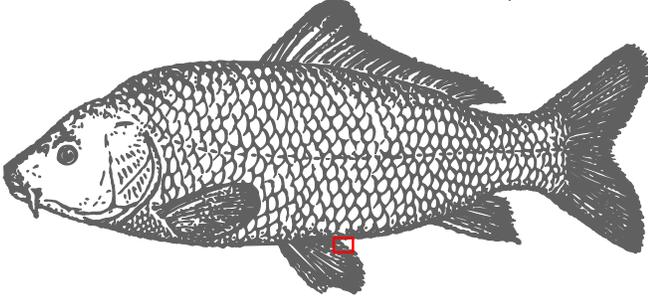


Fig. 5: Watermarked *Carp* graphics (24134 vertices, embedding in 4890 vertices, and 1589 vertices MPR corrected).

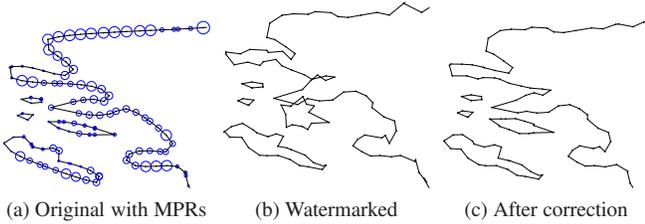


Fig. 6: Part of the *Carp* vector data: (a) original with MPRs superimposed, (b) watermarked without constraint, (c) watermarked after MPR correction.

is the watermarked data after correction based on the MPRs computed from the original geometry. Imposing the distortion constraint preserves the geometrical properties and polygonal chains do not cross.

When MPR correction is applied after the watermarking stage, the correction dampens part of the embedded watermark information. In order to evaluate the detection performance with and without MPR distortion constraint as a function of the embedding strength, we estimate the parameters of the detection statistics ρ under \mathcal{H}_0 and \mathcal{H}_1 using Monte-Carlo simulation with 1000 test runs. The detection threshold based on $\hat{\mu}_{\rho|\mathcal{H}_0}$ and $\hat{\sigma}_{\rho|\mathcal{H}_0}$ is given by $T_\rho = \sqrt{2}\hat{\sigma}_{\rho|\mathcal{H}_0} \operatorname{erfc}^{-1}(2P_{fa}) + \hat{\mu}_{\rho|\mathcal{H}_0}$ for the desired probability of false-alarm (P_{fa}); we set $P_{fa} = 10^{-6}$ for our experiments. Using $\hat{\mu}_{\rho|\mathcal{H}_1}$ and $\hat{\sigma}_{\rho|\mathcal{H}_1}$, the probability for missing the watermark (P_m) can be determined by

$$P_m = \frac{1}{2} \operatorname{erfc} \left(\frac{\hat{\mu}_{\rho|\mathcal{H}_1} - T_\rho}{\sqrt{2}\hat{\sigma}_{\rho|\mathcal{H}_1}} \right), \quad (8)$$

see Barni et al. [13] for details.

We plot the probability of missing the watermark with the LC detector based on the *Carp* graphics for a range of embedding strength factors ($\alpha \in \{0.05, 0.075, \dots, 0.7\}$) without and with distortion constraint based on MPR and cMPR in Fig. 9a. When increasing the embedding strength, more and more watermarked vertices lie outside their MPR and have to be corrected. As expected, this reduces the perfor-

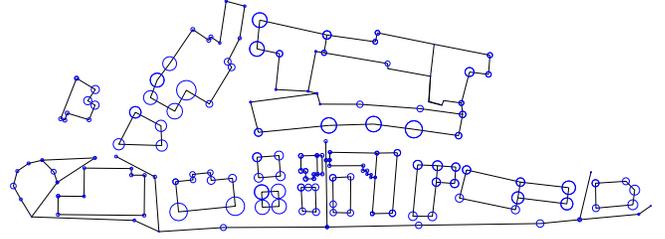


Fig. 7: MPRs on a GIS dataset of the city of Salzburg.

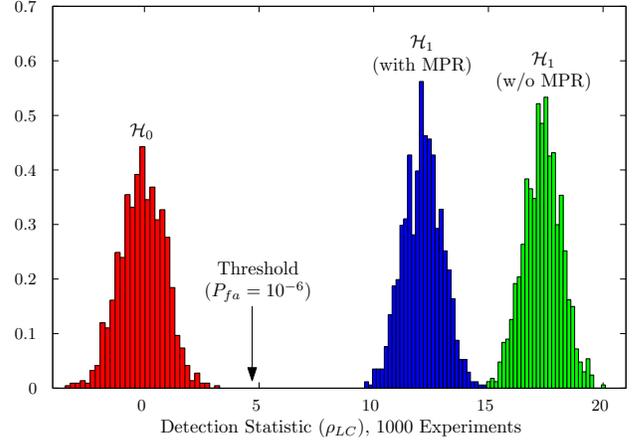


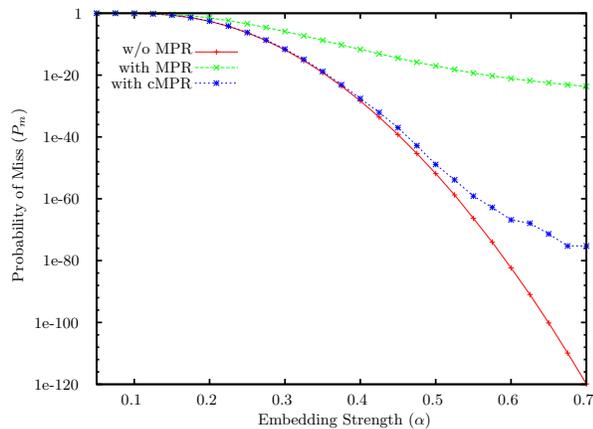
Fig. 8: Detection statistic of ρ_{LC} under \mathcal{H}_0 and \mathcal{H}_1 (with and without MPR correction) on *Carp* with $\alpha = 0.5$.

mance of the watermarking scheme with MPR relative to the unconstrained scheme but guarantees that the geometry of the graphics is preserved. When using the cMPR constraint, performance decreases noticeable only for high embedding strength ($\alpha > 0.5$). Recall that cMPR correction only adjusts watermarked vertices actually causing line segment intersections and may have significantly higher runtime requirements.

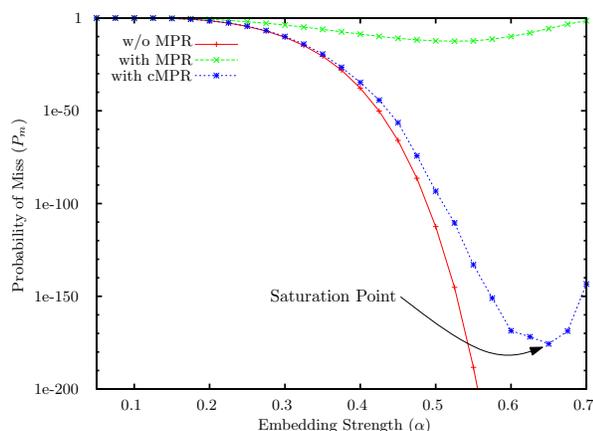
The same experiment is conducted with the LRT-Rayleigh detector and results are shown in Fig. 9b. Compared to the LC detector, the probability of miss decreases faster for the unconstrained scheme. However, when imposing the MPR constraint we observe that P_m reaches a saturation point around $\alpha = 0.5$. For cMPR, saturation occurs around $\alpha = 0.65$. Remember that the test statistic of the LRT detector depends on knowledge of the embedding strength. In the experiment we match the strength for embedding and detection, an assumption which does not hold when more and more vertices are corrected. Oostveen et al. [14] propose a maximum-likelihood estimate of the embedding power for multiplicative watermarking, albeit in a slightly different detector setting.

6. CONCLUSION

We introduced a framework for the watermarking of 2D vector data which allows the computation of a reasonable geo-



(a) Linear Correlation Detector (ρ_{LC})



(b) LRT-Rayleigh Detector (ρ_{LRT})

Fig. 9: Comparing Probability of Miss (P_m) without and with constraint (MPR and cMPR) for varying embedding strength; (a) for the LC and (b) for the LRT-Rayleigh detector.

metric distortion constraint by means of a maximum perturbation region of each vertex. The framework is applicable to robust watermarking schemes in the coordinate and transform domain. Recent advances in the computation of Voronoi diagrams [15] allow our framework to handle circular arcs in the vector data. We are working on extending the approach to 3D vector data by means of conforming Delaunay triangulations.

REFERENCES

- [1] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, Morgan Kaufmann, 2007.
- [2] L. Zheng, Y. Jia, and Q. Wang, "Research on vector map digital watermarking technology," in *Proceedings of the First International Workshop on Education Technology and Computer Science, ETCS '09*, Mar. 2009, pp. 304–307.
- [3] A. Li, Y. Chen, B. Lin, W. Zhou, and G. Lu, "Review on copyright marking techniques of GIS vector data," in *Proceedings of the 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IHHMSP '08*, Aug. 2008, pp. 989–993.
- [4] R. Ohbuchi, H. Ueda, and S. Endoh, "Watermarking 2D vector maps in the mesh-spectral domain," in *Proceedings of Shape Modeling International 2003*, May 2003, pp. 216–225.
- [5] V. Doncel, N. Nikolaidis, and I. Pitas, "An optimal detector structure for the Fourier descriptor domain watermarking of 2D vector graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 5, pp. 851–863, Sept. 2007.
- [6] V. Solachidis and I. Pitas, "Watermarking polygonal lines using Fourier descriptors," *IEEE Computer Graphics and Applications*, vol. 24, no. 3, pp. 44–51, May 2004.
- [7] M. Held, "VRONI: An Engineering Approach to the Reliable and Efficient Computation of Voronoi Diagrams of Points and Line Segments," *Computational Geometry. Theory and Applications.*, vol. 18, no. 2, pp. 95–123, Mar. 2001.
- [8] M. Held, "Voronoi Diagrams and Offset Curves of Curvilinear Polygons," *Computer Aided Design*, vol. 30, no. 4, pp. 287–300, Apr. 1998.
- [9] C. I. Podilchuk and W. Zeng, "Image-adaptive watermarking using visual models," *IEEE Journal on Selected Areas in Communications, special issue on Copyright and Privacy Protection*, vol. 16, no. 4, pp. 525–539, May 1998.
- [10] A. B. Watson, H. A. Solomon, A. Ahumada, and A. Gale, "DCT basis function visibility: Effects of viewing distance and contrast masking," in *Human Vision, Visual Processing, and Digital Display IV*, B.E. Rogowitz, Ed., 1994, pp. 99–108.
- [11] M. Barni, F. Bartolini, and A. Piva, "Improved wavelet-based watermarking through pixel-wise masking," *IEEE Transactions on Image Processing*, vol. 10, no. 5, pp. 783–791, May 2001.
- [12] S.M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*, vol. 2, Prentice-Hall, 1998.
- [13] M. Barni and F. Bartolini, *Watermarking Systems Engineering*, Marcel Dekker, 2004.
- [14] J. Oostveen, T. Kalker, and J.-P. Linnartz, "Optimal detection of multiplicative watermarks," in *Proceedings of the 10th European Signal Processing Conference, EU-SIPCO '00*, Tampere, Finland, Sept. 2000.
- [15] M. Held and S. Huber, "Topology-Oriented Incremental Computation of Voronoi Diagrams of Circular Arcs and Straight-Line Segments," *Computer Aided Design*, vol. 41, no. 5, pp. 327–338, May 2009.