Topology-Preserving Watermarking of Vector Data

Stefan Huber[1] Martin Held[1] Roland Kwitt[2] Peter Meerwald[1]

FB Computerwissenschaften
Universität Salzburg
Austria

Kitware Inc.
NC, USA

EuroCG2012 in Assisi, Italy
March 19–21

Introduction: digital watermarking

Digital watermarking of "raster data" is a thoroughly investigated problem:

- We possess a valuable digital asset (video, music, picture, ...)
  $\rightarrow$ "host signal".
- We want
  - to be able to prove our ownership and
  - to be the only one who is able to do so.
- Basic idea: embed **imperceptible** yet **detectable** distinguished statistical features in the host signal that are based on a **secret key**.
- Only if one possesses the secret key one can detect the presence of the statistical features belonging to this key.

Watermarking vector data

Only limited attention to vector data so far. But vector data carries valuable assets as well:

- Geographic maps (open street map, Google maps, etc.)
- CAD designs
- Circuit board designs

Watermarking vector data:

- Consider a PSLG G as input.
- Watermarking means: embedding statistical features by dislocating vertices.

Geometric constraints

Novel geometric requirements

Watermark embedding must not introduce intersections among edges in G:

- Rivers and streets must not overlap.
- Electrical shortcuts among wires most not be introduced.

More precisely, we want to ensure that after the watermark embedding,

(T1) the numbers of vertices and edges,

(T2) all containment relations

(T3) all incidence orders at vertices

remain unchanged, and that

(T4) **no intersections are introduced**.

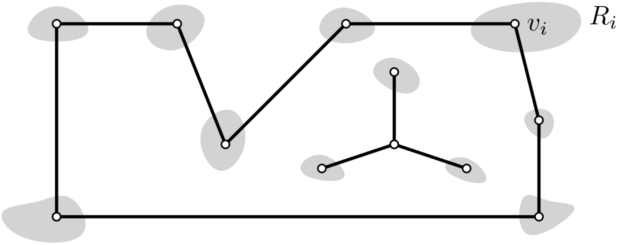
Maximum perturbation regions

Consider a PSLG $G = (V, E)$, which is to be watermarked.

- $V = \{v_1, \ldots, v_n\}$ is the vertex set of G.
- v_i' is the watermarked counterpart of v_i,
  - $V' = \{v_1', \ldots, v_n'\}$ the watermarked vertex set,
  - $G' = (V', E')$ the watermarked graph.

Maximum perturbation region

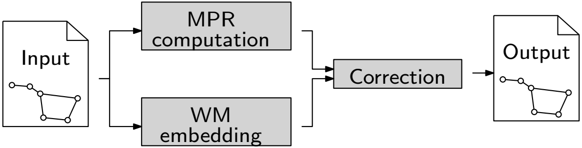
We seek *maximum perturbation regions* (MPRs) $R_1, \ldots, R_n$, with $v_i \in R_i \subset \mathbb{R}^2$, such that: If $v_i' \in R_i$ holds for all $1 \leq i \leq n$ then T1–T4 hold for G'.



Watermarking framework

We designed and implemented a general watermarking framework, consisting of three steps:

1. Computing MPRs,
2. Embedding the watermark with a conventional WM-algorithm,
3. Correcting the watermarked output in order to respect the MPRs.



The correction step potentially weakens the watermark.

- Hence, MPRs should be as large as possible.

Computing MPRs

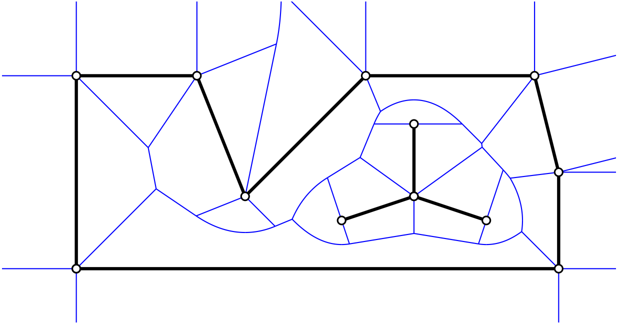
We present two algorithms that compute MPRs:

1. **Using Voronoi diagrams:**
   - $O(n \log n)$ time.
   - Supports other edge types too, e.g., circular arcs.
2. **Using triangulations:**
   - $O(n \log n)$ time.
   - Admits generalization to polyhedra in $\mathbb{R}^3$.

MPRs based on Voronoi diagrams

Key observation

If for all edges $e \in E$ the perturbed counterpart $e' \in E'$ does not intersect Voronoi cells of edges and vertices non-adjacent to e then G' remains planar.



MPRs based on Voronoi diagrams: Phase 1

- For each vertex v_i compute the larges value t_i, such that the union, denoted by $T(v_i)$, of
  - a disk at v_i with radius t_i and
  - rectangles with width $2t_i$ centered at the incident half-edges of v_i

  fit into the union of Voronoi cells containing v_i.
- **Lemma:** Interiors of $T(v_i)$ do not overlap.

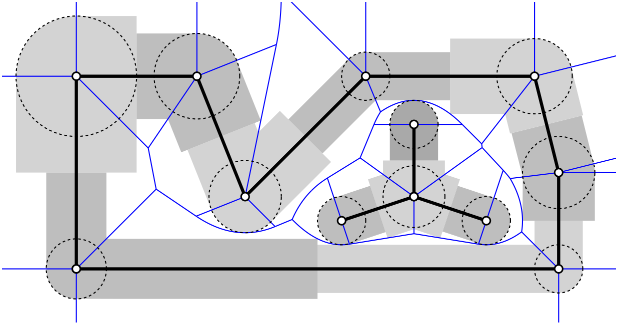
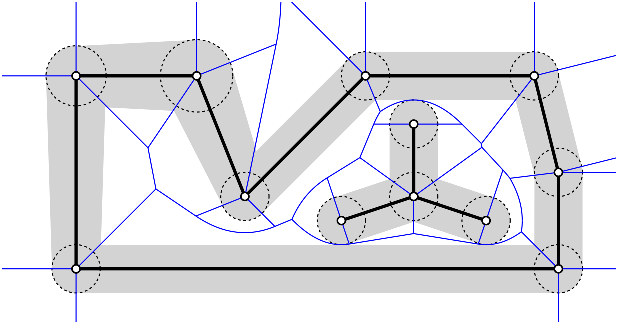


Figure: Shaded areas illustrate sets $T(v_i)$.

MPRs based on Voronoi diagrams: Phase 2

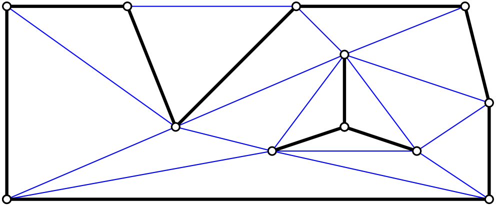
- For each v_i determine $r_i := \min\{t_i\} \cup \{t_j : v_j \text{ adjacent to } v_i\}$.
- Define MPR R_i as the disk centered at v_i with radius r_i.
- **Theorem**: T1–T4 hold.
- MPRs are assigned in a “fair” manner.
- MPRs are not necessarily the largest possible.
- Approach supports other edge types, too.
  - We use VRONI [Held and Huber, 2009] to compute Voronoi diagrams, which also processes circular arcs.



MPRs based on triangulations

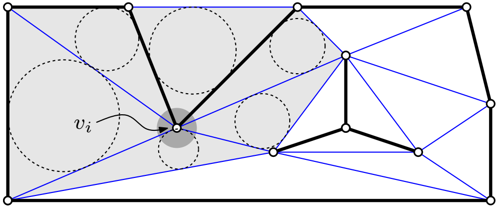
Key observation

Consider a constrained triangulation T of G. If dislocating vertices of V violates T2–T4 then at least one triangle changed its orientation.



MPRs based on triangulations

- Let r_i denote the minimum of the incircle radii of all triangles incident to v_i.
- Define MPR R_i as the disk centered at v_i with radius r_i.
- **Lemma:** triangles preserve their orientations.
- **Theorem:** T1–T4 hold.
- Voroni-based MPRs are in general a bit larger, but triangulations are simple to compute.
- Approach admits a straight-forward generalization to polyhedra in $\mathbb{R}^3$.



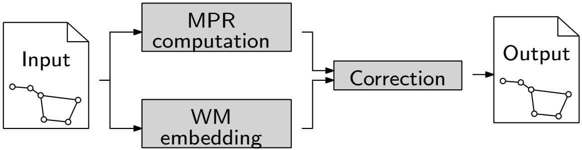
Increasing incircle radii

We prefer triangulations with large incircle radii.

How to increase incircle radii?

- Similar to Shewchuk's "guaranteed quality triangulations":
  - Adding Steiner points such that triangles become more and more equilateral.
- **New problem:** find Steiner triangulation where incircles are maximized.
  - Skinny triangles are fine, if they are large.
- We apply a simple heuristic which increases the average incircle radius by a few percent, as demonstrated by our tests:
  - If we have a triangle with large incircle and its three neighboring triangles have small incircles then we add a Steiner vertex in the center of the large incircle.

Correction step

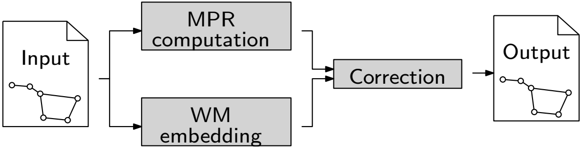


Variant 1: Consider a vertex v_i and the watermarked counterpart v_i'.

- If $v_i' \in R_i$ then nothing needs to be done.
- If $v_i' \notin R_i$ then we project v_i' on the boundary of R_i.

A simple $O(n)$ algorithm.

Correction step



Variant 1: Consider a vertex v_i and the watermarked counterpart v_i'.

- If $v_i' \in R_i$ then nothing needs to be done.
- If $v_i' \notin R_i$ then we project v_i' on the boundary of R_i.

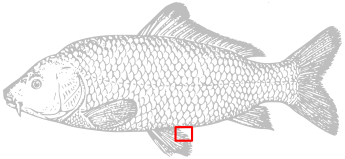
A simple $O(n)$ algorithm.

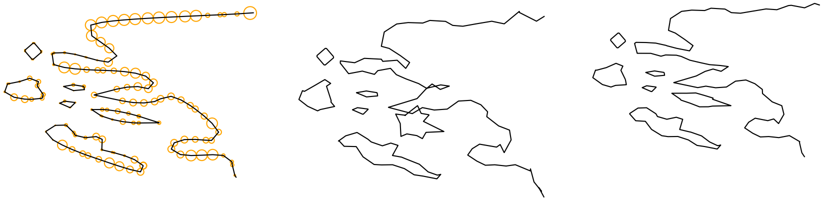
Variant 2: a conditional approach:

- Only correct vertices if incident edges actually violate T2–T4.
  - Note: Correcting an edge can introduce new intersection!
- Better preservation of the embedded watermark.
- Higher computational complexity:
  - Algorithm 1 takes $O(nk)$ time, where $k \in O(n)$ is the number of edges having at least one vertex not in its MPR.
  - Algorithm 2 takes $O(n \log n + m)$ time, where $m \in O(n^2)$ denotes the number of intersections among $E \cup E'$.

Experiments

- The following carp picture contains 24 000 vertices. 1 600 vertices were corrected by the conditional correction strategy.
- Probability of miss:
  - Using unconditional correction strategy: $\approx 10^{-20}$.
  - Using conditional correction strategy: $\approx 10^{-60}$.





(a) original input (b) watermarked input (c) corrected output

Summary

Our contribution:

- We investigated the preservation of the topology of a PSLG after vertices were dislocated due to watermarking.
- Introduced a watermarking framework based on the concept of maximum perturbation regions.
  - Voronoi-based MPRs: $O(n \log n)$ time, can be generalized to more general edge shapes.
  - Triangulation-based approach: $O(n \log n)$ time, can be generalized to $\mathbb{R}^3$.
- We investigated conditional correction strategies. How to efficiently correct only those vertices whose incident edges lead to intersections?
  - Correcting an edge can introduce new intersections!

Future research:

- Watermarking vector data leads to interesting geometrical questions on preserving certain properties.
  - How to preserve right angles in CAD drawings or PCB circuits?
  - How to preserve parallelism?
- How to compute constrained triangulations for which the smallest incircle of all triangles is maximal?

Bibliography

Held, M. and Huber, S. (2009).
Topology-Oriented Incremental Computation of Voronoi Diagrams of Circular Arcs and Straight-Line Segments.
Comput. Aided Design, 41(5):327–338.

Solachidis, V. and Pitas, I. (2004).
Watermarking polygonal lines using Fourier descriptors.
IEEE Computer Graphics and Applications, 24(3):44–51.

Additive spread-spectrum watermarking

Embedding:

- Host signal is a finite sequence $s = (s_0, \ldots, s_n), s_i \in \mathbb{C}$.
- Its Fourier transform is denoted by $\hat{s} = (\hat{s}_0, \ldots, \hat{s}_n), \hat{s}_i \in \mathbb{C}$.
- Generate a random sequence $w = (w_1, \ldots, w_n), w_i \in \{-1, 1\}$, the secret key.
- Compute $\hat{s}' = \hat{s} + \alpha \cdot w$, where $\alpha > 0$ denotes the embedding strength.
- Transform $\hat{s}'$ back to s', which is the watermarked host signal.

Detection:

- Fourier transform the (watermarked?) host signal s' to $\hat{s}'$.
- Compute correlation coefficient c between $\hat{s}'$ and w, e.g., using the linear correlation $c = {}^1/_n \, \hat{s}' \cdot w$. Hence,
  - $c = {}^1/_n \, \hat{s} \cdot w$ if s' carries no watermark, i.e. $s' = s$.
  - $c = {}^1/_n \, \hat{s} \cdot w + {}^1/_n \, \alpha w \cdot w_2 \approx {}^1/_n \, \hat{s} \cdot w$ if s' carries a different watermark w_2
  - $c = {}^1/_n \, \hat{s} \cdot w + \alpha$ if s' carries watermark w

  If c exceeds a threshold, we say to have the watermark detected.