MSc Geomatics thesis presentation

Validation and automatic repair of planar partitions using a constrained triangulation

Ken Arroyo Ohori

Friday, 27 August 2010 at 10:00

Grote Vergaderzaal
OTB Research Institute

Jaffalaan 9, Delft
Representing thematic information

- Pierre Charles Dupin’s 1819 map
  - Subdividing the feature space of that theme
  - Distinct visual representation for each class
Digital thematic maps

- CGIS, SYMAP (Late '60s)
- Polygons to represent boundaries
- Well suited for a computer
Planar partitions | CORINE 2000
Why planar partitions?

- Simple constraints: no gaps, no overlaps, (no disjoint regions)

- Easy to answer questions:
  - (Aggregation) What is the total area of the features of type A and B?
  - (Topology) Are features A and B adjacent?
The problem

- Validate a planar partition

- If it’s invalid, \textit{automatically} repair it.
The problem

- Validate a planar partition

- If it’s invalid, **automatically** repair it.
Why is it hard?

Discrete and approximate representations
Errors in computer operations

Orientation test
Valid polygons

ESRI

Polygons with holes

ISO

Valid polygons
Different sources

More ambiguity

Spain - Portugal border
Validation process

- **Rings:**
  - Closed, not self-intersecting, not zero area, correct winding, ...

- **Polygons:**
  - Nested rings, connected interior, not zero area, ...

- **Planar Partitions:**
  - No gaps, no overlaps, no disjoint regions...
Polygon validation constraints

- ArcGIS: too short line segments, unclosed rings, self-intersections, incorrect ring ordering.

- JTS/GEOS: self-touching rings, zero area rings, zero area polygons, improperly nested rings, duplicate vertices, spikes and gores, touching parts, crossing rings.

- Oracle Spatial: polygon with fewer than 4 vertices, unclosed rings, self-intersections, touching rings, overlapping rings, points too close together, wrong orientation.
Problems are unavoidable
Validation of planar partitions

- Nearly impossible without topology: finding gaps
- Creation of a planar graph based representation
- Plümer and Groger: no dangling edges, no zero-length edges, planarity, no holes, no self-intersections, no overlaps, connectivity.
Repair of planar partitions using snapping
Repair using snapping

- Extensively available: ArcGIS, FME, GRASS, Radius Topology
- Possibilities: point to point, point to line, line to line
Repair using snapping

Not a complete solution
High resolution data

Low resolution data

Repair using snapping

Clean-up required
After snapping:
constraints for planar partition repair

- Radius Topology: share node, node-split-edge, edge-split-edge

- GRASS: break at intersections, remove duplicate line segments, remove dangling edges, remove bridges, remove vertices within a threshold of a line segment, remove too small areas, remove too small angles

- But still, no guarantees
Topological planar partition repair

- Use topological constraints instead

- Snapping can still be performed

- Available in ArcGIS: must not overlap, must not have gaps, must not overlap with, must not have dangles, must not have pseudonodes, must not self intersect
Topological repair in ArcGIS

Manual editing of topology
Topological repair in ArcGIS

Manual editing of topology
The solution

• Repair individual polygons.

• Create a triangulation containing every edge of every polygon.

• Tag every triangle with the polygons it belongs to.

• Re-tag areas with multiple or no tags, according to predefined criteria.

• Reconstruct the polygons in the triangulation.
Triangulations  |  Point sets
Triangulations | Polygons
The constrained Delaunay triangulation

- Delaunay triangulation: empty circle property, uniqueness
- Constrained edges
Repair an individual polygon

- Use the same techniques devised to repair planar partitions.
- Create a triangulation from the polygon.
- Iteratively define exterior and interior when passing a constrained edge.
- Reconstruct polygon.
Create triangulation

• Add every edge of every (now valid) polygon to the triangulation as a constrained edge.

• Track whenever constrained edges are split.
Tag triangulation

• Mark each triangle with the polygons that it belongs to.

• No tags = gap

• Multiple tags = overlap
Repair operations

• At the end, ensure that each triangle has exactly one tag.

• Some possible options:
  
  • Assign triangle or region to the neighbour present on most sides.

  • Assign triangle or region to the neighbour with the longest boundary.

  • Assign region to the class with the highest priority.
Repair operations  Spain - Portugal border
3 repair operations

Triangle with longest boundary

Random region

Region with longest boundary
Polygon reconstruction

- Recursively create a chain of edges representing all boundaries (and some connecting segments).

- Cut where more than two edges join.

- Join small chains in the correct order to form rings.
The prototype

- C++ with CGAL and OGR
- Open source and freely available
Comparisons with other software

• Test polygons with specific problems (ArcGIS)

• Significant differences in interpretation

• Standards specify how to define a certain polygon, not how to interpret an existing one
Comparisons with other software

- Large “normal” data sets (ArcGIS, FME, GRASS)

<table>
<thead>
<tr>
<th>Test</th>
<th>CT memory</th>
<th>time</th>
<th>ArcGIS memory</th>
<th>time</th>
<th>FME memory</th>
<th>time</th>
<th>GRASS memory</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>E41N27</td>
<td>124 MB</td>
<td>46s</td>
<td>145 MB</td>
<td>1m3s</td>
<td>158 MB</td>
<td>31s</td>
<td>59 MB</td>
<td>3m09s</td>
</tr>
<tr>
<td>4tiles</td>
<td>100 MB</td>
<td>3m25s</td>
<td>113 MB</td>
<td>37s</td>
<td>105 MB</td>
<td>31s</td>
<td>49 MB</td>
<td>53s</td>
</tr>
<tr>
<td>16tiles</td>
<td>1.51 GB</td>
<td>1h20m</td>
<td>crashes</td>
<td>–</td>
<td>636 MB</td>
<td>15m48s</td>
<td>crashes</td>
<td>–</td>
</tr>
<tr>
<td>Mexico</td>
<td>983 MB</td>
<td>18m53s</td>
<td>216 MB</td>
<td>&gt;1d</td>
<td>264 MB</td>
<td>2m45s</td>
<td>408 MB</td>
<td>11m38s</td>
</tr>
</tbody>
</table>

- Good performance, considering that it does much more
Conclusions

• Planar partition validation and automatic repair with a constrained triangulation is theoretically simple, yet powerful.

• It keeps a valid topology throughout, without a complex set of rules to check every step of the way.

• Changes that are made to the triangulation have only a local effect.

• New repair operations, based on different criteria, can be easily implemented without breaking the validity of the planar partition.

• Snapping is possible, but not required.
Future work

- Optimisations for simpler polygons
- Improved algorithms for extracting polygons from a triangulation
- Eliminating memory limitations
- Improving the order of point insertions
- Extension to 3D
- Implementation in a database
MSc thesis in Geomatics

Validation and automatic repair of planar partitions using a constrained triangulation

Ken Arroyo Ohori

August 2010