Divide-and-Conquer for Voronoi Diagrams Revisited

Wolfgang Aigner
Institute for Software Technology, TU Graz

joint work with
O. Aichholzer, F. Aurenhammer, T. Hackl (TU Graz),
B. Jüttler, E. Pilgerstorfer, M. Rabl (JKU Linz)

EuroCG, March 2009
Overview

1. Voronoi Diagram $\iff$ Medial Axis
2. State-of-the-art and our Merits in Comparison
3. A Closer Look at the Approach
4. Implementation and Application
Voronoï diagram \(\Rightarrow\) Medial axis

- Voronoï diagram (VD) and medial axis (MA) closely related structures
- *Common approach*: VD of point sample on boundary used to approximate MA

We do: Medial axis \(\Rightarrow\) Voronoï diagram
Medial axis $\Rightarrow$ Voronoi diagram

Edge-graph of \textbf{VD} w. selfedges for set of sites $S$
Medial axis $\Rightarrow$ Voronoi diagram

Edge-graph of $\text{VD}$ w. selfedges for set of sites $S$

MA of circular domain with set of holes $S$
Problems of algorithms for VD...

• **Points:** Robust and efficient algorithms/implementations exist

• **Line segments:** Trouble already starts here
  - *Insertion algorithms:* Two-dimensional bisectors during construction
  - → Additional introduction of endpoint-sites
  - *Divide-and-Conquer:* Computation of many temporary curves
  - → Costly merge operations and error propagation

• **General curves/domains:** It is getting worse...
Problems of algorithms for VD...

- **Points**: Robust and efficient algorithms/implementations exist

- **Line segments**: Trouble already starts here
  - *Insertion algorithms*: Two-dimensional bisectors during construction
  - $\rightarrow$ Additional introduction of endpoint-sites
  - *Divide-and-Conquer*: Computation of many temporary curves
  - $\rightarrow$ Costly merge operations and error propagation

- **General curves/domains**: It is getting worse...
Problems of algorithms for VD...

- **Points:** Robust and efficient algorithms/implementations exist

- **Line segments:** Trouble already starts here
  - *Insertion algorithms:* Two-dimensional bisectors during construction
  - → Additional introduction of endpoint-sites
  - *Divide-and-Conquer:* Computation of many temporary curves
  - → Costly merge operations and error propagation

- **General curves/domains:** It is getting worse...
...and how we evade them

- VD/MA is computed in a **combinatorial way**
- → No bisector computation during construction
- → No additional sites
- → No intermediate structures (perhaps selfedges)
- → No error accumulation

- Possible sites:
  - *Points*
  - *Arbitrary non-selfintersecting curves*
  - *Arbitrary planar simple domains*
...and how we evade them

- VD/MA is computed in a **combinatorial way**
- → No bisector computation during construction
- → No additional sites
- → No intermediate structures (perhaps selfedges)
- → No error accumulation

**Possible sites:**
- *Points*
- *Arbitrary non-selfintersecting curves*
- *Arbitrary planar simple domains*
Augmenting disks

Main idea:
Perforated domain $\xrightarrow{\text{augment}}$ Combinatorially simple domain
Cyclic medial axis $\xrightarrow{\text{augment}}$ Tree-like medial axis

- Break cycles $\rightarrow$ keep simply connected!
- How: Augment the domain at maximal disks
- Crossing of both arcs $\rightarrow$ $\infty$ distance
Breaking the cycles

Main idea:

Perforated domain $\xrightarrow{\text{augment}}$ Combinatorially simple domain
Cyclic medial axis $\xrightarrow{\text{augment}}$ Tree-like medial axis

- Augmenting disks may partially overlap
- $\rightarrow$ Recursive definition of distance function
- Augmented domain has tree-like MA
Augmenting disk sweep

- $s_i$ swept over → add at $p(s_i)$
- $s_l$ nearer than $s_k$ → $D_L$ updated
- $D_L$ swept over → augmenting disk
Augmenting disk sweep

- $s_i$ swept over
  $\to$ add at $p(s_i)$

- $s_l$ nearer than $s_k$
  $\to$ $D_L$ updated

- $D_L$ swept over
  $\to$ augmenting disk
Augmenting disk sweep

- $s_i$ swept over
  $\rightarrow$ add at $p(s_i)$
- $s_l$ nearer than $s_k$
  $\rightarrow D_L$ updated
- $D_L$ swept over
  $\rightarrow$ augmenting disk
Divide…

Medial axis via Divide-and-Conquer (*Aichholzer et al. 2008*)

- Compute random maximal disk → decompose shape
- Apply divide step recursively down to base cases
- Decomposition lemma by *Choi et al. (1997)*
  → independent computation of partial medial axes
Medial axis via Divide-and-Conquer (Aichholzer et al. 2008)

- Merge step $\rightarrow$ simple concatenation of partial axes
- No numeric problems from potentially intricate merging
- Combinatorial representation of MA through base cases $\rightarrow$ not a single bisector computation required
VD of simple sites

- Arbitrary sites as input allowed for VD
- Acceptable runtimes for simple (e.g. point) sites

<table>
<thead>
<tr>
<th># sites</th>
<th>secs</th>
<th>atomic steps</th>
<th>$n \log^2(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>0.11</td>
<td>7591</td>
<td>0.25</td>
</tr>
<tr>
<td>2000</td>
<td>0.76</td>
<td>54662</td>
<td>0.23</td>
</tr>
<tr>
<td>4000</td>
<td>1.71</td>
<td>143391</td>
<td>0.25</td>
</tr>
<tr>
<td>20000</td>
<td>12.45</td>
<td>1015149</td>
<td>0.25</td>
</tr>
<tr>
<td>40000</td>
<td>28.22</td>
<td>2659149</td>
<td>0.28</td>
</tr>
<tr>
<td>200000</td>
<td>280.89</td>
<td>19820012</td>
<td>0.32</td>
</tr>
</tbody>
</table>

Runtimes on Pentium 4 with 2.80GHz

- Simple sites: robust algorithms are known for some time
- Our strength: handling of large/complex sites
VD of complex sites

- Computation of Voronoi diagram for 40 complex sites
- Sites boundaries $\rightarrow$ biarc approximation of varying quality

<table>
<thead>
<tr>
<th># arcs</th>
<th>secs</th>
<th>at. steps</th>
<th>$n \log(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1714</td>
<td>0.32</td>
<td>45660</td>
<td>2.48</td>
</tr>
<tr>
<td>2824</td>
<td>0.53</td>
<td>75717</td>
<td>2.34</td>
</tr>
<tr>
<td>5622</td>
<td>1.05</td>
<td>152511</td>
<td>2.18</td>
</tr>
<tr>
<td>11822</td>
<td>2.20</td>
<td>324121</td>
<td>2.03</td>
</tr>
<tr>
<td>25210</td>
<td>4.72</td>
<td>729264</td>
<td>1.98</td>
</tr>
<tr>
<td>54214</td>
<td>10.56</td>
<td>1636895</td>
<td>1.92</td>
</tr>
<tr>
<td>116460</td>
<td>23.50</td>
<td>3718057</td>
<td>1.90</td>
</tr>
</tbody>
</table>
Offset computation

- Areas of applications for VD and MA are manifold
- Combinatorial representation of VD/MA via base cases → recommends itself for e.g. **trimmed offset computation**
- With given MA: any offset computable in linear time

<table>
<thead>
<tr>
<th>error</th>
<th># arcs</th>
<th>MA</th>
<th>off</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k \cdot 10^{-1})</td>
<td>468</td>
<td>0.07</td>
<td>0.02</td>
</tr>
<tr>
<td>(k \cdot 10^{-2})</td>
<td>916</td>
<td>0.16</td>
<td>0.04</td>
</tr>
<tr>
<td>(k \cdot 10^{-3})</td>
<td>1860</td>
<td>0.30</td>
<td>0.07</td>
</tr>
<tr>
<td>(k \cdot 10^{-4})</td>
<td>3872</td>
<td>0.64</td>
<td>0.15</td>
</tr>
<tr>
<td>(k \cdot 10^{-5})</td>
<td>8156</td>
<td>1.39</td>
<td>0.31</td>
</tr>
</tbody>
</table>
Thank You for Your Attention!!