

Auto-refreshing the QGIS map pane for visual debugging algorithms

Martijn Meijers

Wednesday, January 31, 2018

IHE Delft Institute for Water Education

Delft, 16:25 – 16:45

```
412 ... Function that will be executed when file is cache_redraw():
413 ... clear the cache and redraw
414 ... print 'Function clear_cache_redraw called'
415 ... iface.mapCanvas().setCachingEnabled(False)
416 ... iface.mapCanvas().resetCachedContent()
417 ... iface.mapCanvas().refresh()
420 ... Watch the file system
421 ... from PyQt4.QtCore import QFileSystemWatcher
422 ... watcher = QFileSystemWatcher(['/home/martijn/
423 ... watcher.fileChanged.connect(clear_cache_redra
424
```

Introduction

- Delft University of Technology, GIS technology

Introduction

- Delft University of Technology, GIS technology
- Look under the hood of GIS

Introduction

- Delft University of Technology, GIS technology
- Look under the hood of GIS
- Map generalization: Make vector maps simpler

Introduction

- Delft University of Technology, GIS technology
- Look under the hood of GIS
- Map generalization: Make vector maps simpler
- Develop tools/algorithms/workflows (with Python)

Introduction

- Delft University of Technology, GIS technology
- Look under the hood of GIS
- Map generalization: Make vector maps simpler
- Develop tools/algorithms/workflows (with Python)
- “Share something useful”: Auto-refreshing the QGIS map pane*

*No complete plugin, nor polished solution

Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)

- Given a sufficiently dense sampled polygonal shape

Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)

- Given a sufficiently dense sampled polygonal shape
- Produce points near the approximate center of the shape

Case

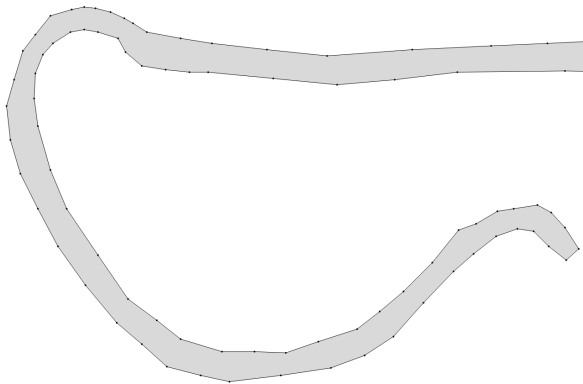
Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)

- Given a sufficiently dense sampled polygonal shape
- Produce points near the approximate center of the shape
- By fitting circles inside the shape

Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)

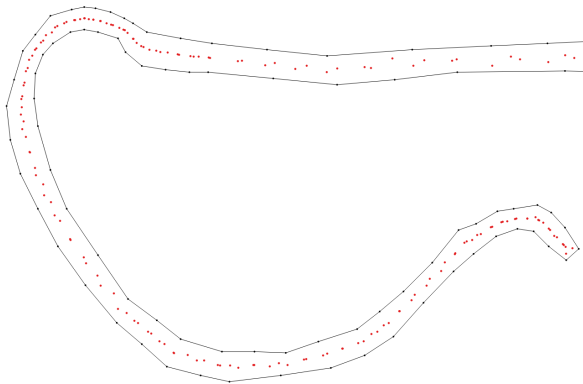
- Given a sufficiently dense sampled polygonal shape
- Produce points near the approximate center of the shape
- By fitting circles inside the shape



Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)

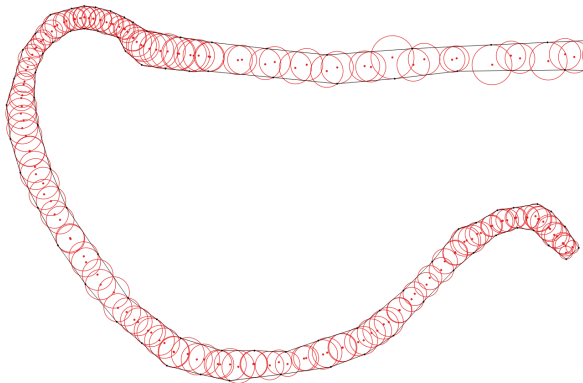
- Given a sufficiently dense sampled polygonal shape
- Produce points near the approximate center of the shape
- By fitting circles inside the shape



Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)

- Given a sufficiently dense sampled polygonal shape
- Produce points near the approximate center of the shape
- By fitting circles inside the shape



Visual Debugging Algorithms

- Input ▷ Algorithm ▷ Output

Visual Debugging Algorithms

- Input ▷ Algorithm ▷ Output
- Algorithm produces intermediate results: Look at these for debugging

Visual Debugging Algorithms

- Input ▷ Algorithm ▷ Output
- Algorithm produces intermediate results: Look at these for debugging
- Graphical debuggers, e.g. Eclipse with PyDev, give tabular view on variables

Visual Debugging Algorithms

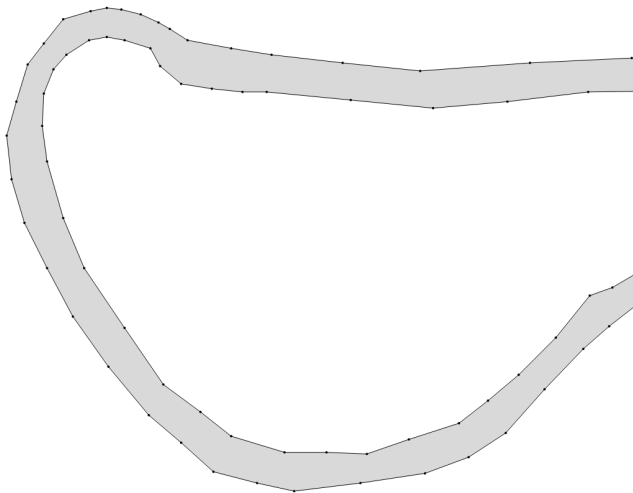
- Input ▷ Algorithm ▷ Output
- Algorithm produces intermediate results: Look at these for debugging
- Graphical debuggers, e.g. Eclipse with PyDev, give tabular view on variables
- No display of geometry of vector data

Visual Debugging Algorithms

- Input ▷ Algorithm ▷ Output
- Algorithm produces intermediate results: Look at these for debugging
- Graphical debuggers, e.g. Eclipse with PyDev, give tabular view on variables
- No display of geometry of vector data
- QGIS to the rescue!
 - Write text file: Tab Separated Values
 - Use Well Known Text (WKT) for geometry (POINT, LINESTRING, POLYGON)
 - Layer ▷ Add Layer ▷ Add Delimited Text Layer...
 - Style layer as usual
(e.g. convey direction with marker lines)!

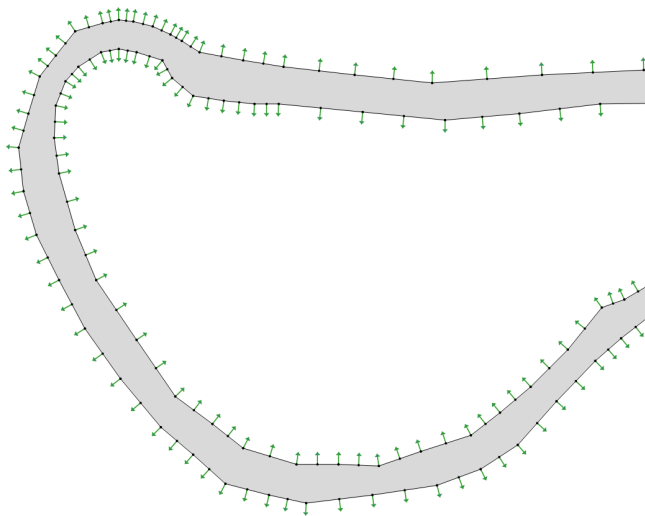
Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)
intermediate results we are interested in:



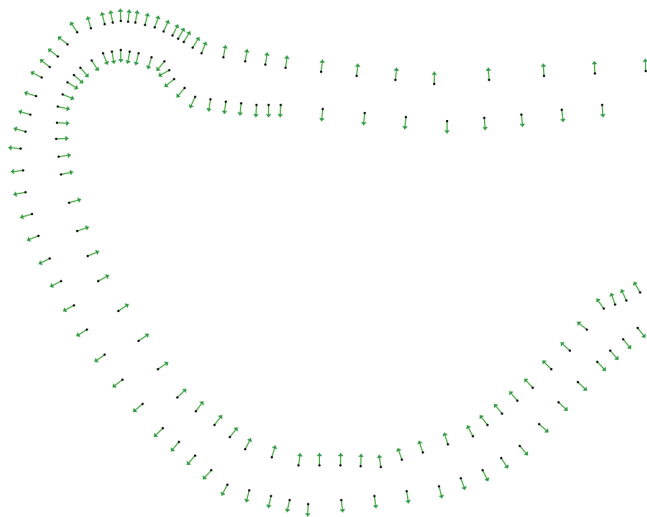
Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)
intermediate results we are interested in:



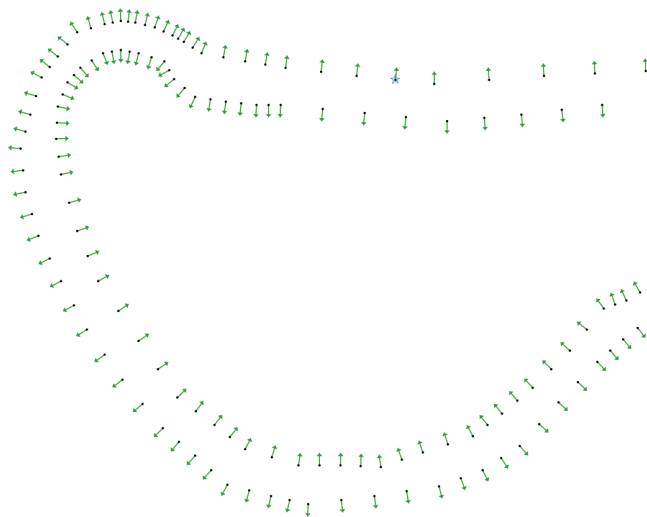
Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)
intermediate results we are interested in:



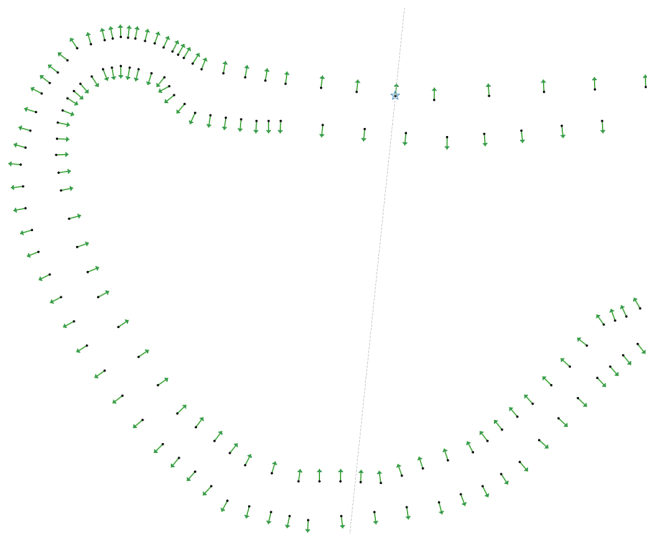
Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)
intermediate results we are interested in:



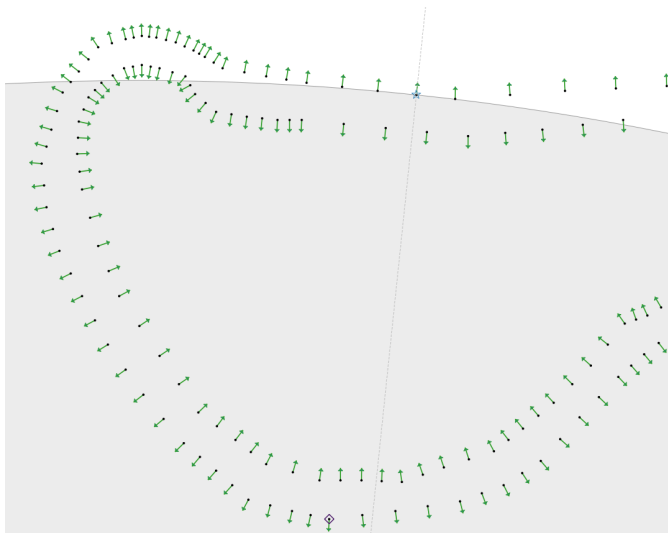
Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)
intermediate results we are interested in:



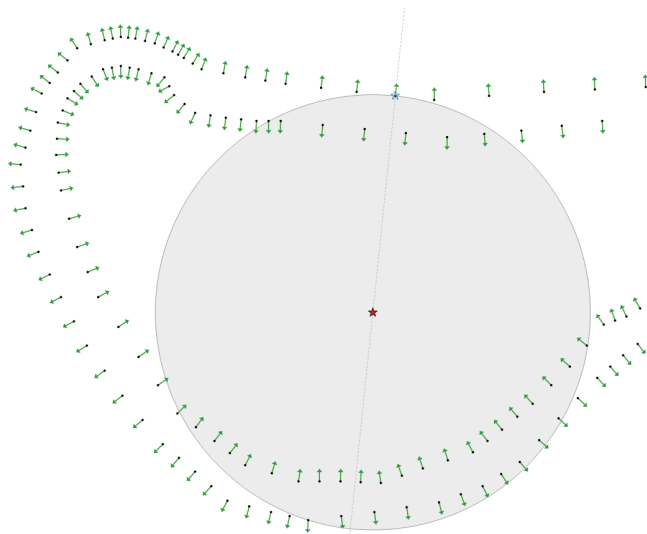
Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)
intermediate results we are interested in:



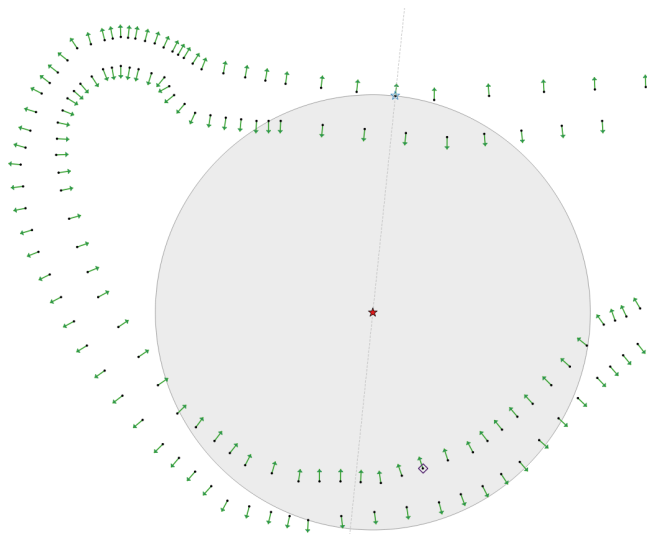
Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)
intermediate results we are interested in:



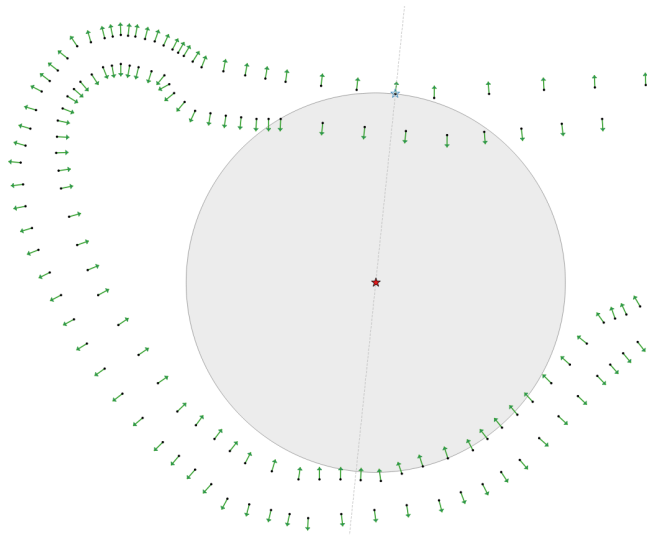
Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)
intermediate results we are interested in:



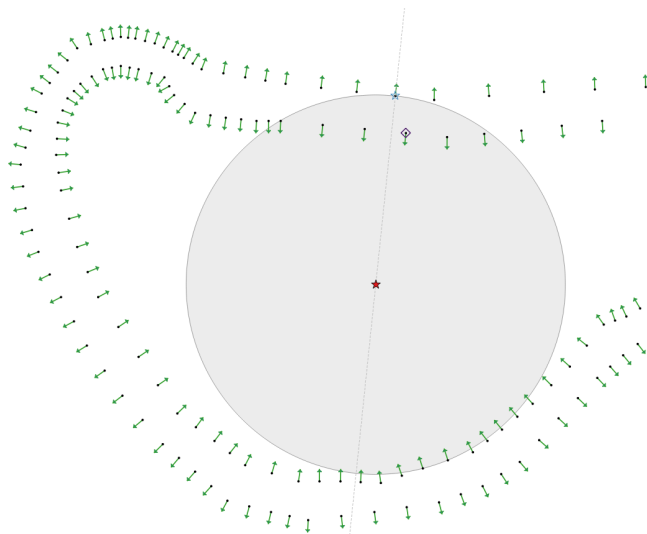
Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)
intermediate results we are interested in:



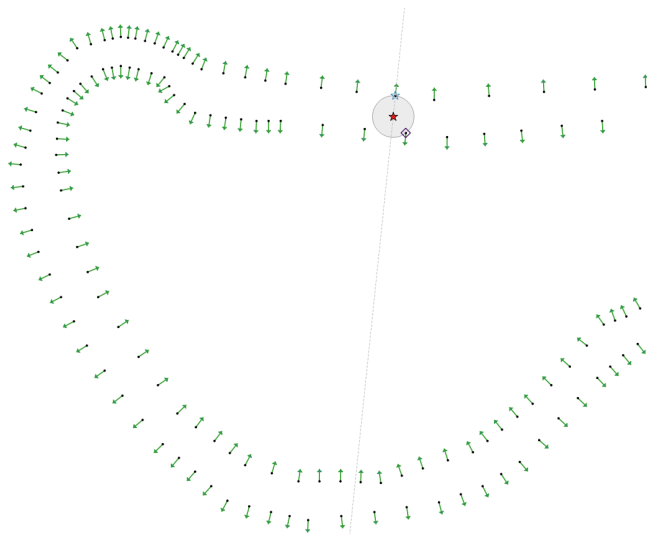
Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)
intermediate results we are interested in:



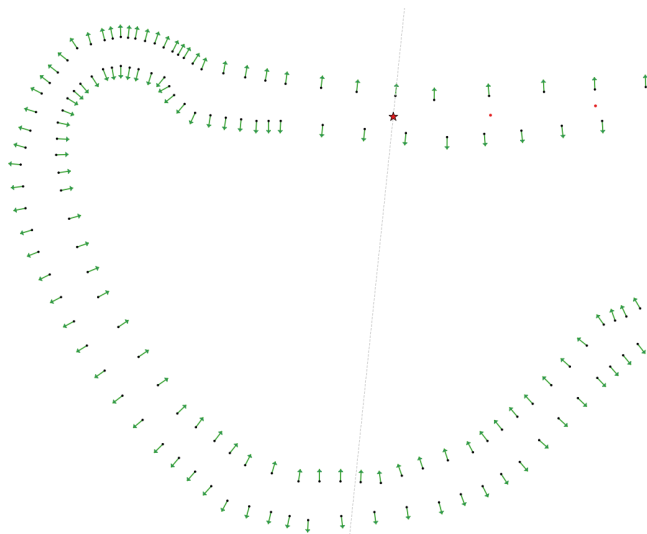
Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)
intermediate results we are interested in:



Case

Algorithm (2D version of algo by Ma et al., 2012; Peters and Ledoux, 2016)
intermediate results we are interested in:



Auto-refreshing the QGIS map pane

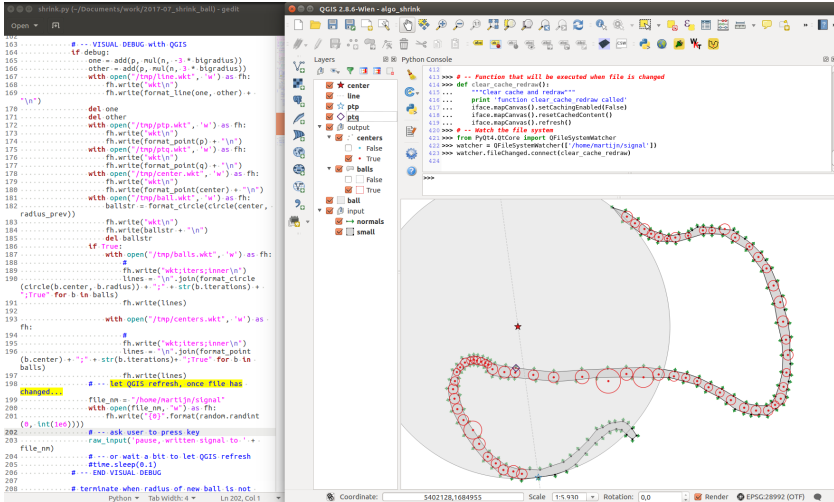
1. Instruct QGIS to watch for changes on a file (using Python console and QFileSystemWatcher)
 2. Run program in terminal / Interactive Development Environment (IDE)
 3. Program writes intermediate results (as TSV)
 4. Program writes random value to 'watch file'
 5. Pause program (using `raw_input()` in Python or break points in IDE)
 6. QGIS auto-refreshes the map pane!
- No need to leave terminal / IDE!

Watching for changes on a file using PyQt

Inside QGIS: Plugins ► Python console

```
1 # -- Function that will be executed when file is changed
2 def clear_cache_redraw():
3     """Clear cache and redraw"""
4     print 'function clear_cache_redraw called'
5     iface.mapCanvas().setCachingEnabled(False)
6     iface.mapCanvas().resetCachedContent()
7     iface.mapCanvas().refresh()
8
9 # -- Watch the file system
10 from PyQt4.QtCore import QFileSystemWatcher
11 watcher = QFileSystemWatcher(['/home/martijn/signal'])
12 watcher.fileChanged.connect(clear_cache_redraw)
```

Demo



The screenshot displays the QGIS 2.8.6-Wien interface. The left pane shows a Python script in the editor, and the right pane shows the map view and the Python console.

Python Code (Left Pane):

```
103 # --- VISUAL DEBUG with QGIS
104 if debug:
105     one = add(p, mul(n, -3 * bigradius))
106     other = add(p, mul(n, 3 * bigradius))
107     with open("/tmp/line.wkt", "w") as fh:
108         fh.write("wkt\n")
109         fh.write(format_line(one, other) + "\n")
110
111 del one
112 del other
113 with open("/tmp/ptp.wkt", "w") as fh:
114     fh.write("wkt\n")
115     fh.write(format_point(p) + "\n")
116     fh.write("ptp\n")
117     fh.write(format_point(q) + "\n")
118     fh.write("center\n")
119     fh.write(format_point(center) + "\n")
120     with open("/tmp/ball.wkt", "w") as fh:
121         ballstr = format_circle(circle(center,
122 radius_prev))
123         fh.write("wkt\n")
124         fh.write(ballstr + "\n")
125         del ballstr
126         if True:
127             with open("/tmp/balls.wkt", "w") as fh:
128                 #
129                 fh.write("wkt\n"); inner(n)
130                 lines = "\n".join(format_circle
131 (circle(b.center, b.radius)) + "\n" + str(b.iterations) +
132 ; if True for b in balls)
133                 fh.write(lines)
134
135 with open("/tmp/centers.wkt", "w") as
136 fh:
137     #
138     fh.write("wkt\n"); inner(n)
139     lines = "\n".join(format_point
140 (b.center) + "\n" + str(b.iterations) + "; True" for b in
141 balls)
142     fh.write(lines)
143
144 # --- let QGIS refresh, once file has
145 changed...
146 file_nm = "/home/martijn/signal"
147 with open(file_nm, "w") as fh:
148     fh.write("0")
149     format(random.randint
150 (0, int(1e5)))
151
152 # --- ask user to press key
153 raw_input('pause, written signal to: ' +
154 file_nm)
155
156 # --- or wait a bit to let QGIS refresh
157 time.sleep(0.1)
158 # --- END VISUAL DEBUG
159
160 # terminate when radius of new ball is not
```

Map View (Right Pane):

The map view shows a large grey circle (ball) with a red star at its center. A path of smaller red circles (balls) is visible, moving towards the center. The map is displayed in a coordinate system with a scale of 1:5,930 and a rotation of 0.0. The map is rendered using EPSG:28992 (OTF).

Python Console (Right Pane):

```
412 # --- Function that will be executed when file is changed
413 def clear_cache_redraw():
414     """clear cache and redraw"""
415     print "function clear cache redraw called"
416     if iface.mapCanvas().setCachingEnabled(False):
417         if iface.mapCanvas().resetCachedContent():
418             if iface.mapCanvas().refresh():
419                 #
420
421 # --- Watch the file system
422 from PyQt4.QtCore import QFileSystemWatcher
423 watcher = QFileSystemWatcher("/home/martijn/signal")
424 watcher.fileChanged.connect(clear_cache_redraw)
```

Python code and QGIS project available on Bitbucket:

<https://bitbucket.org/bmmeijers/shrink/src>

Auto-refreshing the QGIS map pane

- Also useful for producing a small video:
 1. ...
 2. QGIS auto-refreshes the map pane
 3. QGIS saves the map pane contents to disk as image
- Use another tool (ffmpeg) to process all images to a movie (e.g. for sharing on vimeo, youtube, ...)

Watching and saving the map pane as image

Plugins ► Python console

```
1 # -- Function that will be executed when file is changed
2 def clear_cache_redraw():
3     """Clear cache and redraw"""
4     print 'function clear_cache_redraw called'
5     iface.mapCanvas().setCachingEnabled(False)
6     iface.mapCanvas().resetCachedContent()
7     iface.mapCanvas().refresh()
8     global ct # we update the counter outside the function
9     ct += 1
10    iface.mapCanvas().saveAsImage(
11        '/tmp/shrink{0:05d}.png'.format(ct)
12    )
13
14 # -- Watch the file system
15 from PyQt4.QtCore import QFileSystemWatcher
16 watcher = QFileSystemWatcher(['/home/martijn/signal'])
17 watcher.fileChanged.connect(clear_cache_redraw)
18
19 # -- Initialize counter
20 ct = 0
```

Producing a small movie

With ffmpeg you can convert the .png files to a .mp4:

```
1 ffmpeg -framerate 8 -i shrink%05d.png -c:v libx264 -r 32 out_8_32  
   .mp4
```

I hope I 'shared something useful'

Questions?

- dr.ir. Martijn Meijers
@bmmeijers
b.m.meijers@tudelft.nl
<http://www.gdmc.nl/martijn/>
tel. (+31) 15 27 856 42
- Delft University of Technology
Faculty of Architecture and the Built Environment
OTB – Research for the built environment
GIS Technology

Acknowledgements

- Thanks to Ravi Peters, for explaining me the 3D algorithm in detail.
- A 3D version of the algorithm is available from him:
<https://github.com/tudelft3d/masbcpp>

References

- Ma, J., Bae, S. W., and Choi, S. (2012). 3D medial axis point approximation using nearest neighbors and the normal field. *The Visual Computer*, 28(1):7–19.
- Peters, R. and Ledoux, H. (2016). Robust approximation of the Medial Axis Transform of LiDAR point clouds as a tool for visualisation. *Computers & Geosciences*, 90(A):123–133.