QGIS Application - Bug report #9480 Labels for polygon centroids should always originate inside a polygon

2014-02-03 11:37 AM - Tim Sutton

Status:	Closed			
Priority:	Normal			
Assignee:	Larry Shaffer			
Category:	Labelling			
Affected QGIS version:2.0.1		Regression?:	No	
Operating Syste	e m: all	Easy fix?:	No	
Pull Request or Patch supplied:		Resolution:		
Crashes QGIS or corru ptis data:		Copied to github as	; #: 18072	
Description				

When labelling polygons the label placement is often suboptimal. I am attaching a contrived example which illustrates the point. Often the centroid of the polygon is calculated to be outside the boundaries of the polygon and thus a label appears to reference a neighbouring polygon. Using the 'real centroid' (for example as implemented in the realcentroid plugin http://plugins.qgis.org/plugins/realcentroid/) would be one way to address this.

History

#1 - 2014-02-03 11:38 AM - Tim Sutton

- File qgis-issue-9480.zip added

#2 - 2014-02-04 12:16 AM - Martin Dobias

There is something called Distance Transform that computes distance from each point to the border. We could probably use it to get nice labeling of complex polygons: the idea is to do this transform for each (rasterized) polygon - that would trigger a kind of heatmap (high score for central areas, low score for areas close to the border) which would be used as a source for costs of label candidates.

http://en.wikipedia.org/wiki/Distance_transform

#3 - 2014-02-04 12:17 AM - Jukka Rahkonen

Doesn't QGIS come with GEOS? It could be worth having a try if label placement would look better by using GEOS PointOnSurface method which is the same as JTS getInteriorPoint

 $\underline{http://tsusiatsoftware.net/jts/javadoc/com/vividsolutions/jts/geom/Geometry.html#getInteriorPoint%28\%29$

The speed is not necessarily much slower.

PostGIS implements this as <u>http://www.postgis.org/docs/ST_PointOnSurface.html</u>.

#4 - 2014-02-05 08:15 AM - Martin Dobias

Yeah, PointOnSurface could be an alternative to centroid - it would be good to try it out and have a comparison!

#5 - 2014-03-02 10:00 PM - Alvaro Huarte

Using the 'Horizontal' or 'Free' placements in label configuration the text is drawed inside of polygon. The PAL label library, that QGIS uses, only forces the label inside of polygon with these placements.

See:

https://github.com/qgis/QGIS/blob/master/src/core/ggspallabeling.cpp#L3243 https://github.com/qgis/QGIS/blob/master/src/core/pal/feature.cpp#L1342 https://github.com/qgis/QGIS/blob/master/src/core/pal/feature.cpp#L1034

Best Regards

Alvaro

#6 - 2014-03-07 05:42 PM - Larry Shaffer

Hi Alvaro,

Alvaro Huarte wrote:

Using the 'Horizontal' or 'Free' placements in label configuration the text is drawed inside of polygon. The PAL label library, that QGIS uses, only forces the label inside of polygon with these placements.

Neither of those placements are based upon the polygon's centroid, but instead upon available area. Tim's request is to base it off the centroid *and* ensure it is inside the polygon.

#7 - 2014-03-12 07:41 PM - Alvaro Huarte

Hi Larry, I propose a new pull request (https://github.com/qgis/QGIS/pull/1238) using the 'GEOSPointOnSurface' method.

The code has three commits:

- + New 'pointOnSurface' method for QgsGeometry using 'GEOSPointOnSurface'.
- + New option for CentroidFillSymbolLayerV2 to force the location of centroid inside of polygons.
- + Force label position inside of polygons in PAL labeling library.

Best Regards

Alvaro

#8 - 2014-05-24 10:25 AM - Alvaro Huarte

- File other_example_test_issue_9480.zip added

#9 - 2014-05-29 02:02 PM - Larry Shaffer

- File 9480_edge-point.jpeg added
- Status changed from Open to Closed

Should be fixed, excepting that sometimes the point is moved to an edge, which leaves the resultant label in a vague spot when between two polygons.

See label 683/3:



Further work to ensure such new edge points should be further moved to a 'centered' midpoint between an opposite side, if it exists, or to some other more ideal point.

Commit commit:fe42b004

#10 - 2014-05-30 11:49 AM - Alvaro Huarte

- File testInsidePolygon.JPG added

Larry Shaffer wrote:

Should be fixed, excepting that sometimes the point is moved to an edge, which leaves the resultant label in a vague spot when between two polygons.

See label 683/3:



Further work to ensure such new edge points should be further moved to a 'centered' midpoint between an opposite side, if it exists, or to some other more ideal point.

Commit commit:fe42b004

Thanks Larry, this algorithm cat get good results ?

1) Find closest segment to the outside centroid.

2) Calculate all intersections of a perpendicular ray from centroid to the closest segment

3) Using each pair of intersection points, from nearest point calculate if the middle point of each segment is inside.



#11 - 2014-05-30 12:01 PM - Larry Shaffer

Yes, that seems like it might be faster, especially if the intersections can be done incrementally (if that matters). For example in this case, after the first two intersections are located, that would be all that is needed.

Though your algorithm skips doing a call to **pointOnSurface()** first, which should help with performance, don't we still need to do that call, then check if the resultant point is on, or really close, to an edge *before* doing such an algorithm? In my testing I found that ~75% of the time **pointOnSurface()** is already good enough. Just need to do something like what you are suggesting for the other ~25%, whenever it happens to fall on an edge. Or, is that what you are already suggesting?

PS: I think the discussion should stay here now, instead of github.com and here. :-)

#12 - 2014-05-30 12:16 PM - Alvaro Huarte

Larry Shaffer wrote:

Yes, that seems like it might be faster, especially if the intersections can be done incrementally (if that matters). For example in this case, after the first two intersections are located, that would be all that is needed.

Though your algorithm skips doing a call to **pointOnSurface()** first, which should help with performance, don't we still need to do that call, then check if the resultant point is on, or really close, to an edge before doing such an algorithm? In my testing I found that ~75% of the time **pointOnSurface()** is already good enough. Just need to do something like what you are suggesting for the other ~25%, whenever it happens to fall on an edge. Or, is that what you are already suggesting?

PS: I think the discussion should stay here now, instead of github.com and here. :-)

When the centroid is not inside of polygon, I think that first it must get all intersection points and after sort these points using the distance to the centroid. Then the algorithm test each middle of pair of points if inside of polygon.

Files			
qgis-issue-9480.zip	5.98 KB	2014-02-03	Tim Sutton
other_example_test_issue_9480.zip	1.36 KB	2014-05-24	Alvaro Huarte
9480_edge-point.jpeg	40.7 KB	2014-05-29	Larry Shaffer
testInsidePolygon.JPG	33.4 KB	2014-05-30	Alvaro Huarte