1 Dec, 2016

Robert Trzeciak

Program Manager

Northrop Grumman IS

468 Viking Drive

Virginia Beach, VA 23452

757-498-5544 work

757-635-2628 mobile

**Progress Report – 1 Nov 2016 – 30 Nov 2016**

Contract Number: HSHQDC-06-D-00022

Contract Number 7500097279

Order Number: HSCG23-07-J-TED150

Task Order – Performance Work Statement (PWS) 1.12

Attachments: (1) SAROPS subcontractor financial reports.

1. **Frustrating month on geometry. Finally, at the end of the month, I graphically displayed answers to the 3 fundamental questions of geometry, that must be solved in order to do land processing efficiently. These 3 problems are 1. Find the closest shoreline edge and point for a given point, 2. Find the first closest shoreline edge, point, and polygon, when going straight north, and 3. Find the first crossing for a given edge. These were all solved, but the solutions took too much time, too much memory, and the code was far too complex. I kept thinking of better ways to solve these, and I finished them well enough to show at the TWG on Nov 29. There was a glitch in the first crossing, but I believe that I fixed that on Nov 30. The hardest of these 3 is #1. #3 would be difficult for long edges, but we rarely apply #3 to long edges; that is most often applied to the distress travel of a particle within a timestep. #1 was simply too slow, because I had no way to discard vast chunks of data when I didn’t know how close the closest point would be. I must start the search for the closest point by considering anything within 180 degrees (degree here is a measurement of distance), and this doesn’t allow me to throw out many sub-packets of edges. Finally I realized that any sub-packet will have an edge that is no farther away than the farthest point of that sub-packet; I don’t have to find an edge to cut down on the distance I need to search. This made a huge difference. I kept track of the number of packets and the number that I searched. Typically, it went from 300 of 13,000 to 3 of 13,000 when I applied this new post-IPR piece of logic.  
    The basic structure of my solution for all 3 problems is the sorting that I presented at the IPR, but I changed that slightly. Now I sort by increasing left endpoints and then *de*creasing right endpoints, and this little twist made the rest of the code simpler. Here’s a simple example of this idea; suppose (and I do) want to nest the polygons. Sort the polygons by their bounding latLng boxes, south to north, by increasing lowest latitude. Then, with another trip through the data, one can find the potential surrounding polygons for each polygon by considering only the lat-ranges of the latLng boxes. Then, for each polygon *P*, sort the potential surrounding polygons by increasing area, and pick the first one *Q* that contains the first point of *P.* *Q* is the “tightest” surrounding polygon of *P* (i.e, there is no polygon between *P* and *Q*). This code is short and fast.**
2. **Fixed the John Squires Problem; this was a glitch that popped up because of inaccurate code-copying when I went from 2.03 (where it was fixed) to 2.1 (where the bug occurred. Comparing the 2 code-bases, the fix was easy (I hope). It’s hard to reproduce so we have to hope that fix is what was wrong. I haven’t heard from John since the fix went in.**
3. **There was a Planner Hang that I fixed. The problem occurred for some cases that had frozen Srus; for an Sru, certain rounding errors could cause confusion when trying to come up with a “containment polygon.”**
4. **Had a bug with “Latest Seed.” This is an important feature since we really don’t want to start Planner over every time we “re-optimize,” and that’s what will happen without “Latest Seed.”**
5. **Am looking “Martian” in input now, but still have a bug. Judy is now writing the “On Mars” flag for me.**
6. **Discussions on linking Patterns (in Planner, we assign each “Sru” to a Pattern, and we will have to change the term for Sru because what are usually considered Sru’s can now have multiple Patterns).**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Activity Worked** | **Hours Worked** | **Hourly Cost** | **Total Cost** |
| Kratzke | Coding/Doc/Travel | 184.5 | -- | -- |
|  |  |  |  |  |
| **Totals** |  | 184.5 |  |  |
|  |  |  |  |  |