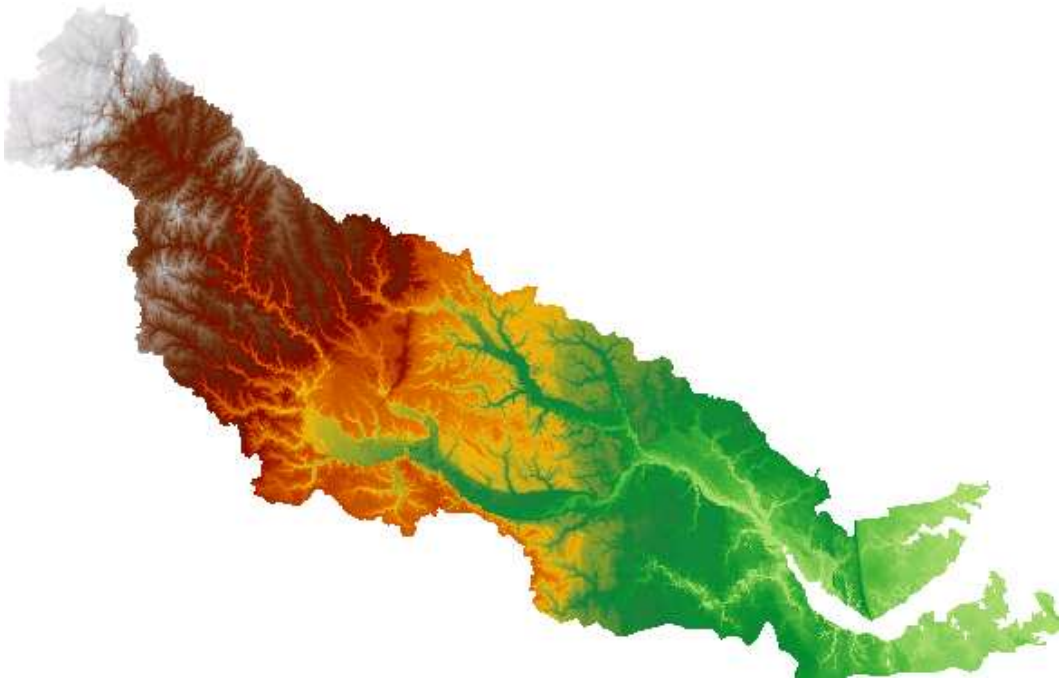


Figure 2.3: Neuse river basin data set

Points from neighboring quad-tree segment are not found in advance as in our algorithm, but are found when interpolating a given quad-tree segment q; the algorithm creates a window w by expanding q in all directions by a width δ and querying the quad tree to find all points within w. The width δ is adjusted by binary search until the number of points within w is between a user specified range $[n_{\min}, n_{\max}]$. Once an appropriate number of points is found for a quad-tree segment q, the grid cells in q are interpolated and written directly to the proper location in the output grid by randomly seeking to the appropriate file offset and writing the interpolated results. When each segment has a small number of cells, writing the values of the T output grid cells uses $O(T) \gg \text{sort}(T)$ I/Os. Our approach constructs the output grid using the significantly better $\text{sort}(T)$ I/Os.

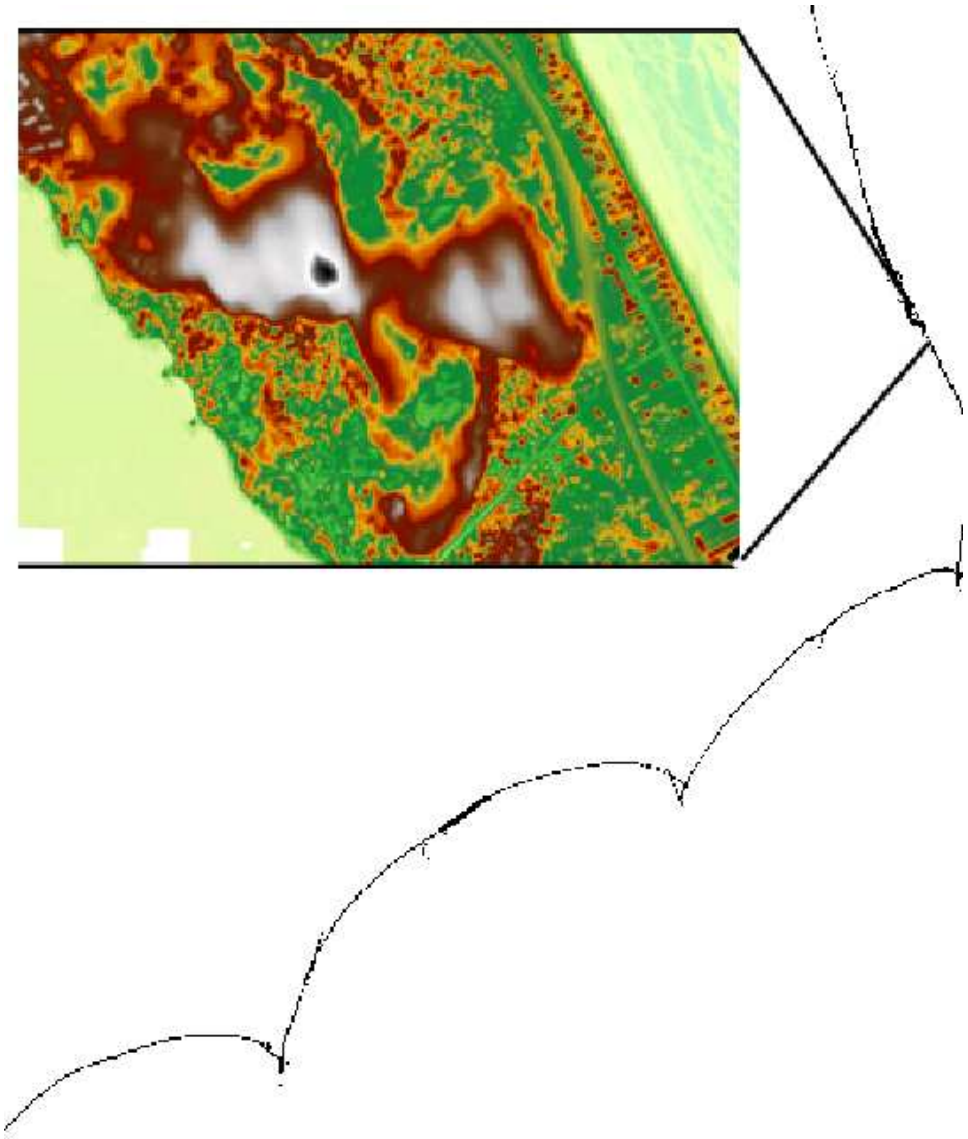


Figure 2.4: Outer Banks data set, with zoom to very small region.

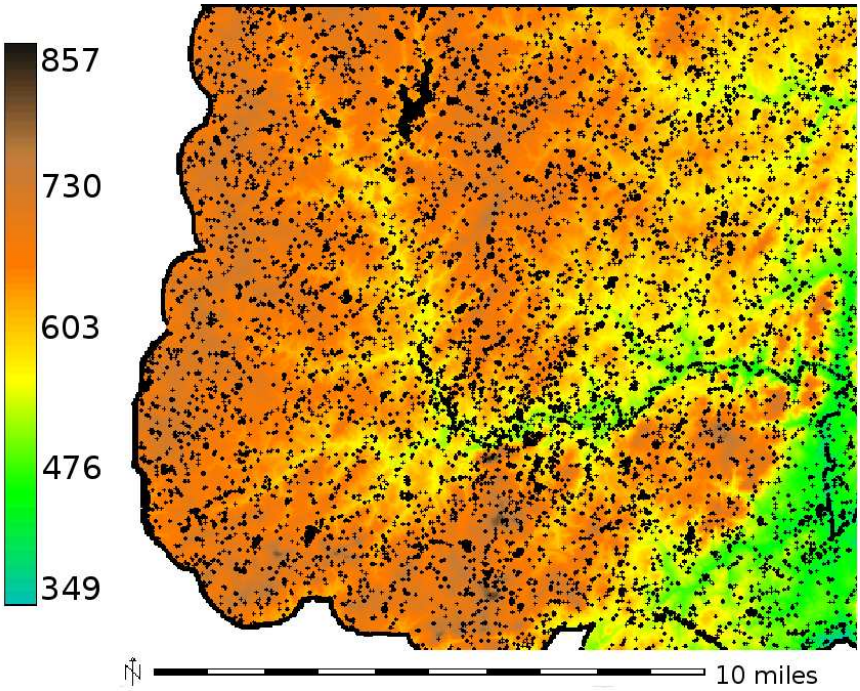
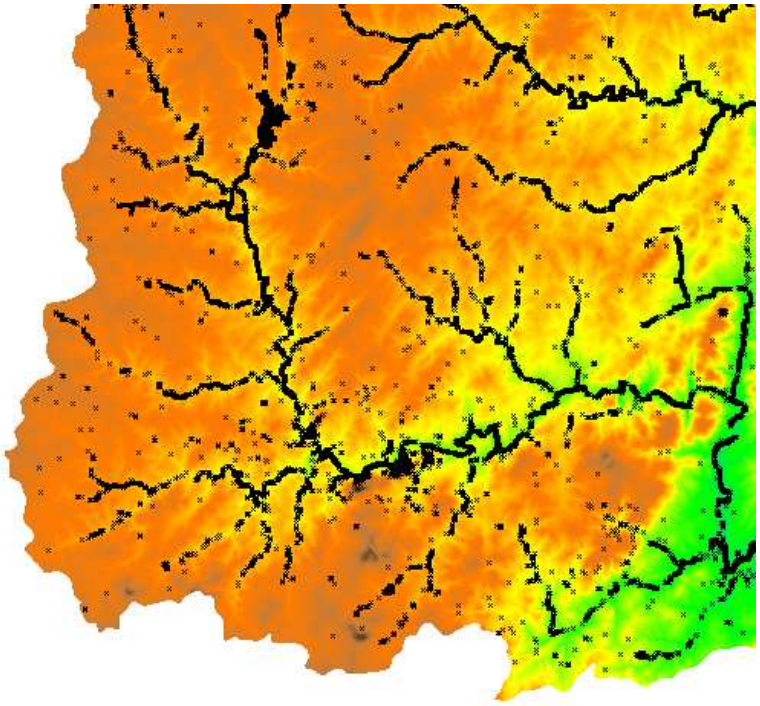


Figure 2.6: Interpolated surface generated by our method. Black dots indicate cells where the deviation between our method and `v.surf.rst` is greater than three inches.

ment between the interpolated surfaces and the base grid is not as strong as the agreement between the algorithms when compared to each other. An overlay of regions with deviation greater than two feet on base map shown in Figure 2.7(a) reveals the source of the disagreement. A river network is clearly visible in the figure indicating that something is very different between the two data sets along the rivers. NC Floodmaps uses supplemental break-line data that is not part of the lidar point set to enforce drainage and provide better boundaries of lakes in areas where lidar has trouble collecting data. Aside from the rivers, the interpolated surface generated by either our method or the prior GRASS implementation agree reasonably well with the professionally produced and publicly available base map. Furthermore, it was recently observed by Hodgson et al. [55], that the mean absolute error and the RMSE of the lidar signals themselves are 8.7 inches and 13.0 inches respectively in smooth open terrain and these errors can be over two feet in forested or mixed cover terrain.



(a)

Figure 2.7: Interpolated surface generated by our method. Black dots indicate cells where the deviation between our method and ncfloodmap data is greater than two feet.

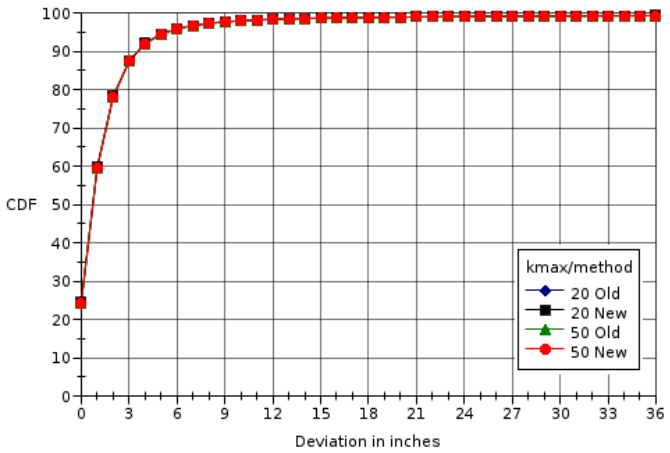
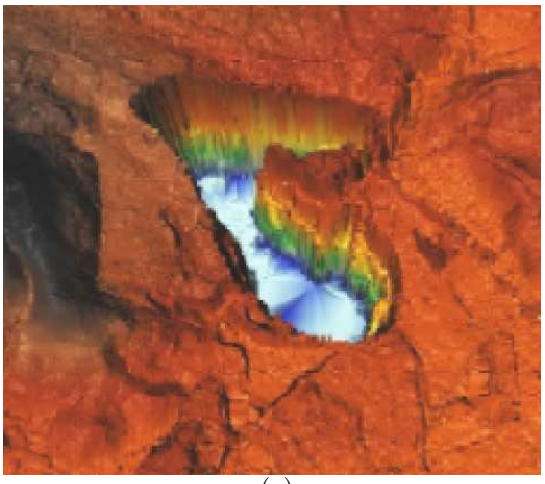
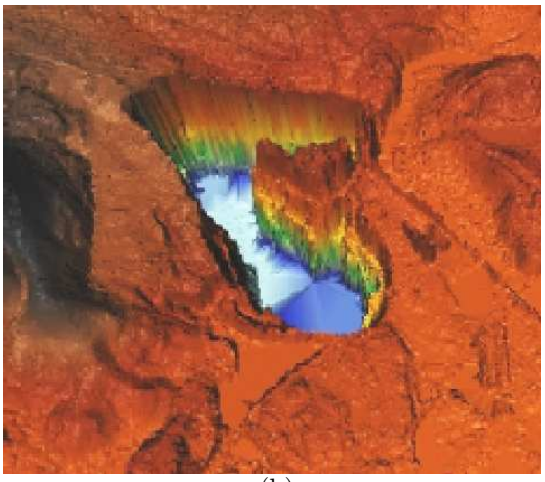


Figure 2.8: Cumulative distribution of deviation between interpolated surface and data downloaded from ncfloodmaps.com. Deviation is similar for both our method and `v.surf.rst` for all values of $k_{\max}$.



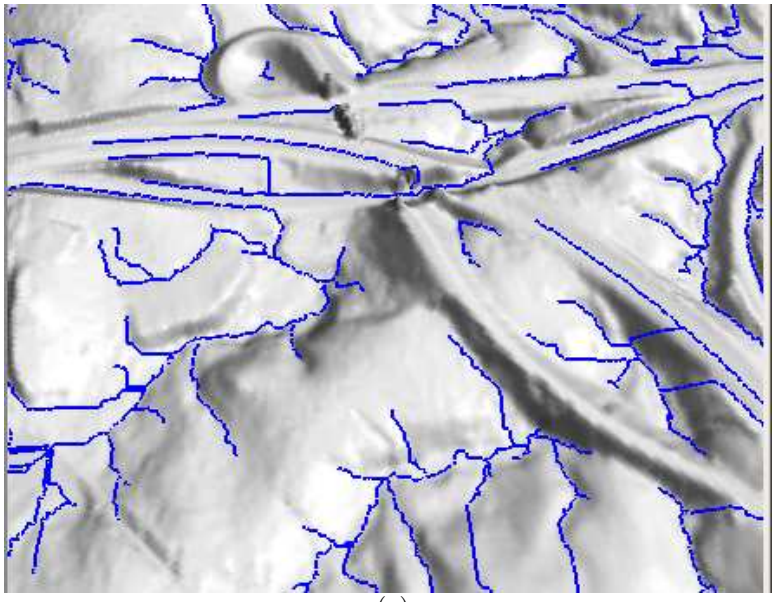
(a)



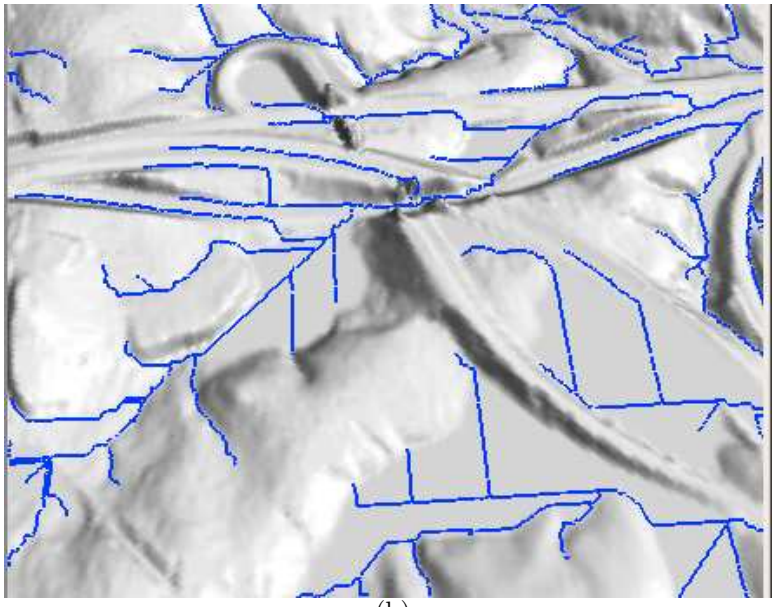
(b)

(c)

Figure 3.4: (a) Original terrain. (b) Terrain flooded with persistence threshold $\tau = 30$. (c) Terrain flooded with $\tau = \infty$.

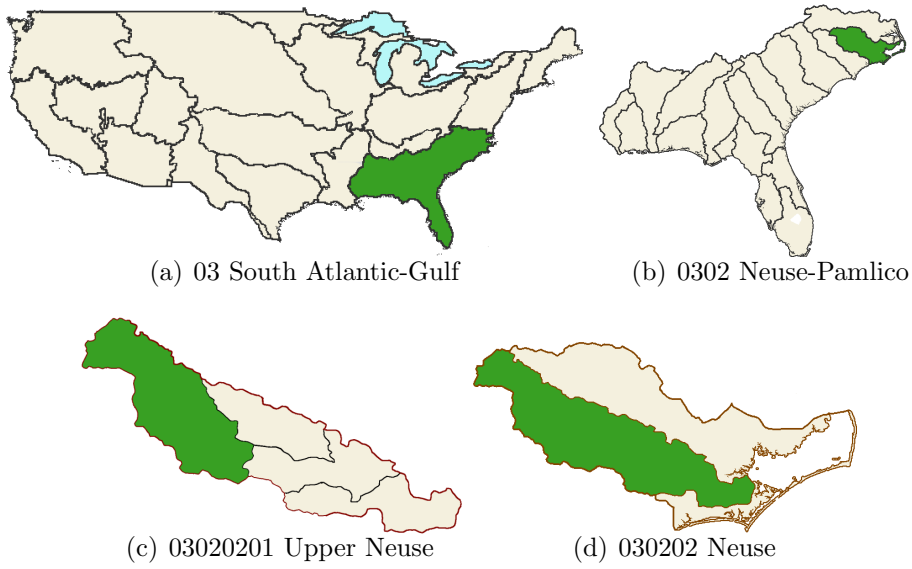


(a)



(b)

Figure 3.9: (a) Terrain and flow graph edges shown in blue with flooding of only low persistence sinks (b) Terrain and flow graph edges with flooding of all sinks.



(a) 03 South Atlantic-Gulf (b) 0302 Neuse-Pamlico

(c) 03020201 Upper Neuse (d) 030202 Neuse

Figure 4.1: A region, sub-region, basin and sub-basin in the USGS Hydrologic Unit System.

of the terrain in the United States, it has some disadvantages. First, while the HUC boundaries are available for download, there is no automatic way to compute the USGS hydrological units given a digital elevation model. As the quality and resolution of digital elevation models improve, the published HUC boundaries may not exactly match the boundaries suggested by the data. Second, HUCs at the sub-basin level may be too large for a particular application. Further sub-levels are in development but are not complete at this time. Third, HUCs are only available for the United States. Other countries and organizations have other coding methods [90]. Finally, the digits chosen for a particular HUC are, for the most part, arbitrary. Given two HUCs, it is often difficult or impossible to determine if water from one HUC flows into the other based on their numbering alone. Because finding the hydrological units upstream and downstream from a given location is a common task, a numbering scheme that allows a computer or user to relate hydrological units, without the need for visual inspection, would be helpful.

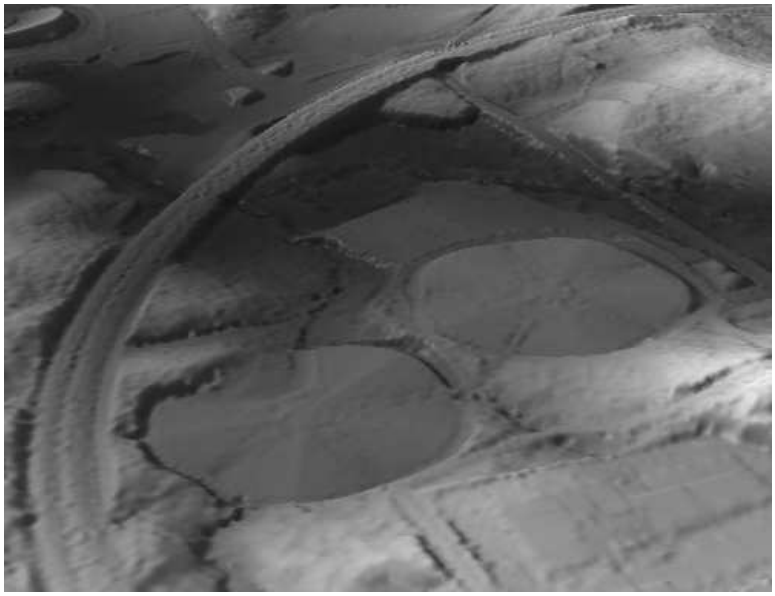


Figure 6.2: Sample bare Earth lidar data still shows some man-made features including the bridge in the center left and eight baseball fields grouped into the two circles shown in the foreground.

area of roughly 6200 square miles with an average point spacing of approximately 20 feet. However, the point spacing is rather heterogeneous, ranging from nine feet in open areas to more than 50 feet in densely vegetated regions. Because lidar pulses are absorbed over water, there are few data points over large bodies of open water. The data have been pre-processed by the data providers to remove large amount of vegetation and many buildings from the terrain. However, many man-made features still exist, including bridges, as shown in Figure 6.2. Because this lidar data was collected for the purposes of flood mapping, some bridges like the one shown in Figure 6.3 have been cut during pre-processing to allow the flow of water under the bridge. While many bridges across major waterways have been cut, Figure 6.2 shows that this is not always the case.

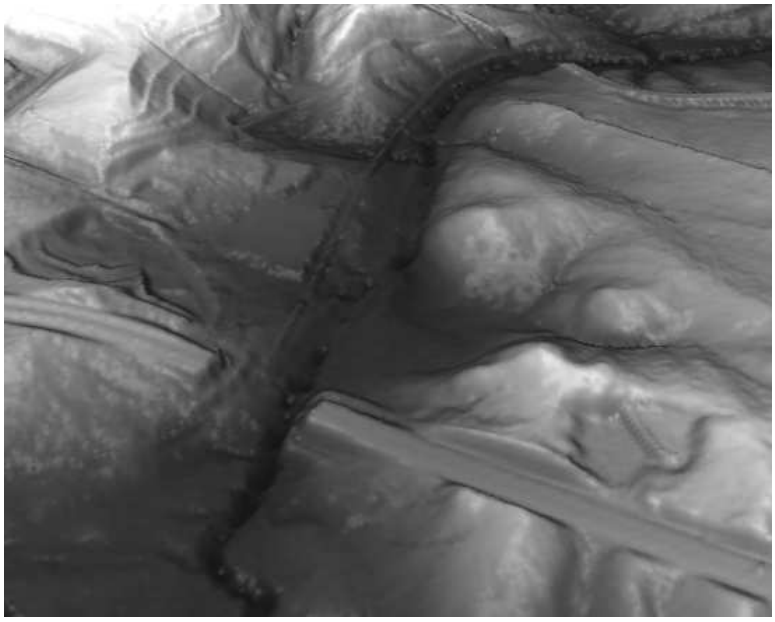


Figure 6.3: In this lidar example, the bridges have been cut by the data providers to allow water to flow through. Many bridges across major waterways have been cut, but many bridges across smaller streams have not.

6.2.1 Software and Hardware

We implemented our algorithms using the C++ programming language and the Linux operating system. We built our code on a number of external software libraries, primarily TPIE, GRASS, and GDAL. As mentioned in the introduction, TPIE [11], is a templated, portable, I/O environment written in C++ that provides support for implementing I/O-efficient algorithms and data structures. Our implementation work was greatly simplified by the fact that all main primitives of our algorithms—scanning, sorting, stacks and priority queues—are already implemented I/O-efficiently in TPIE. For data visualization and basic data manipulation, we used the open-source GIS GRASS [52], written primarily in C. In particular, for our grid DEM construction algorithm, we used the regularized spline with tension interpolation code that exists in the GRASS module `s.surf.rst`. Because of scalability problems with the new vector engine in GRASS 6.2, we used GRASS 5.4 for our

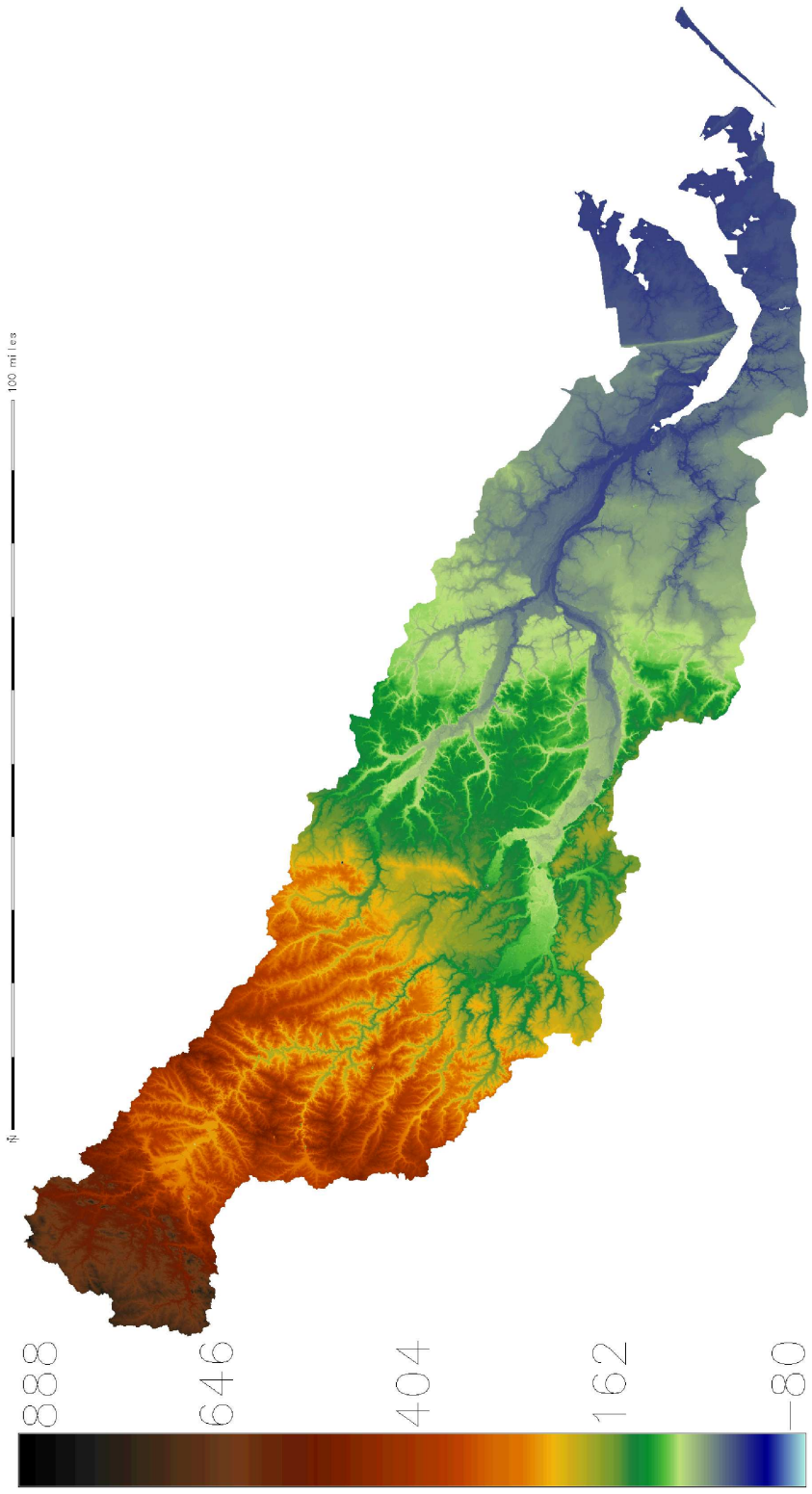
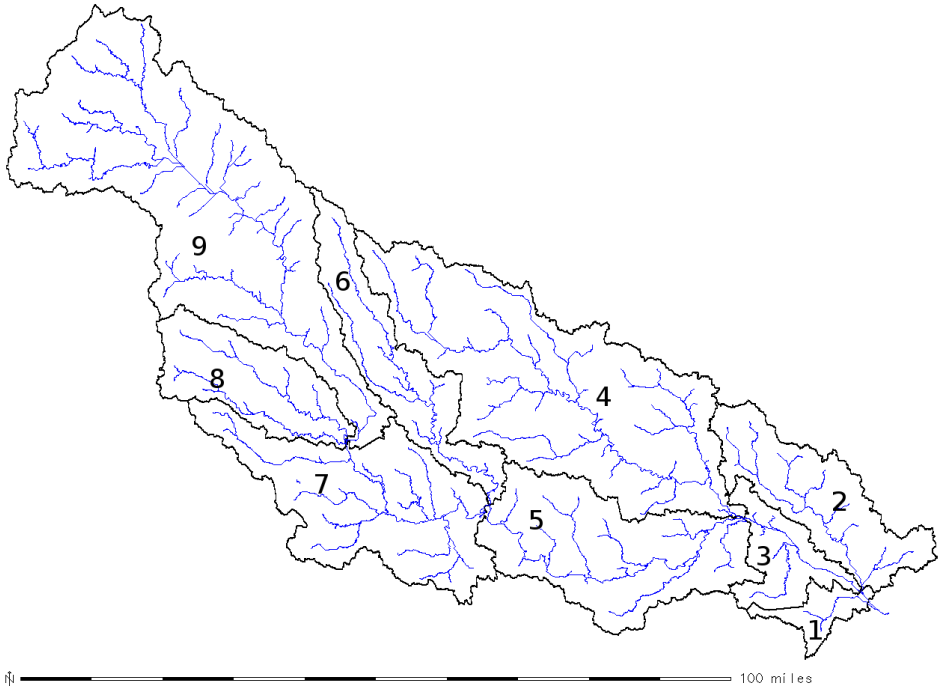
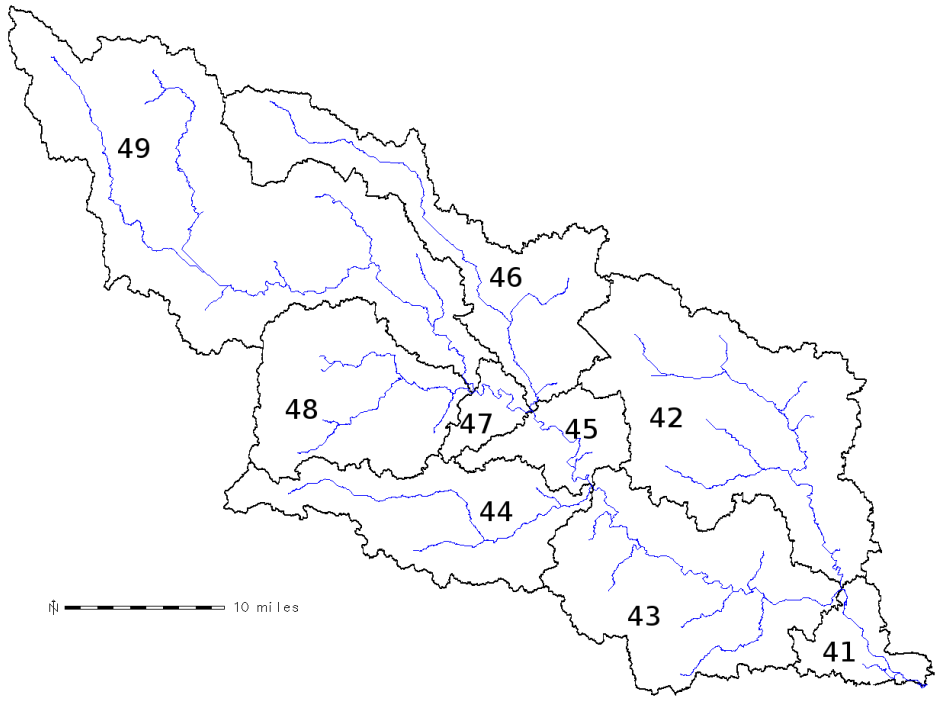


Figure 6.4: DEM of Neuse river basin derived from lidar points



(a)



(b)

Figure 6.5: (a) First level of Pfafstetter watershed labels for largest basin in Neuse. (b) Recursive decomposition of basin four.

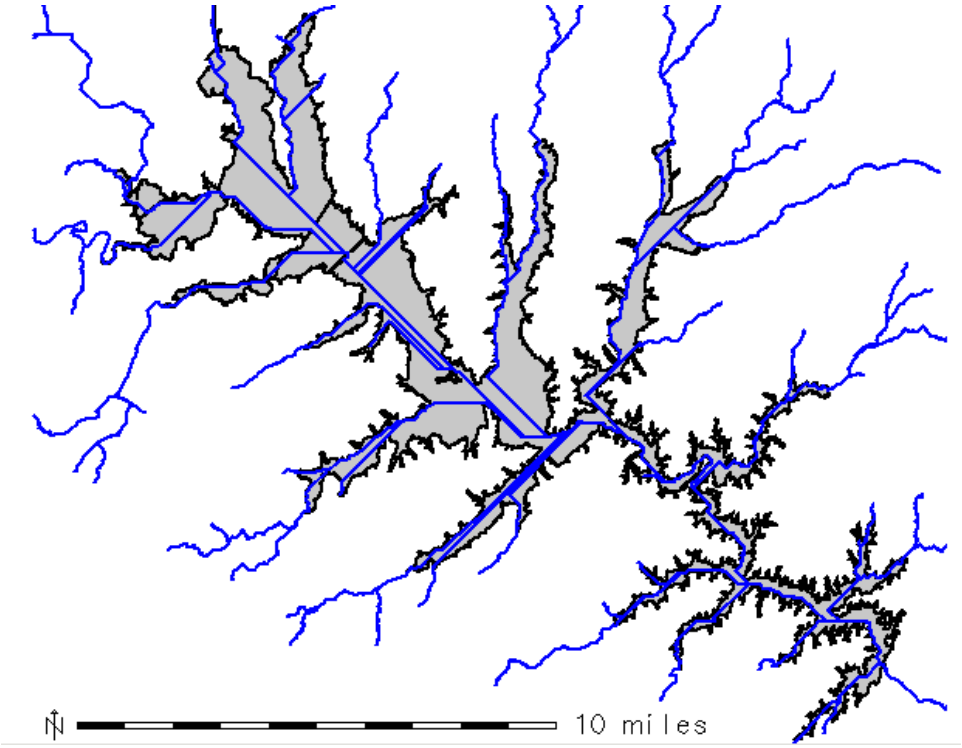


Figure 6.6: Falls lake, with a dam located near the Southeast corner of the figure. The boundary of the Falls lake flat is outlined in black while blue lines show rivers entering the reservoir and routed across the flat area.

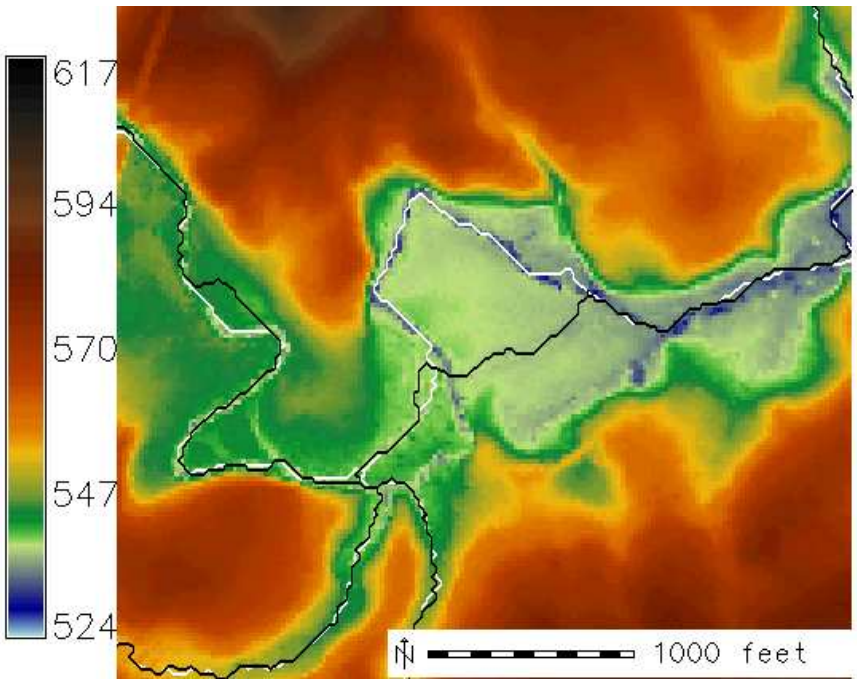
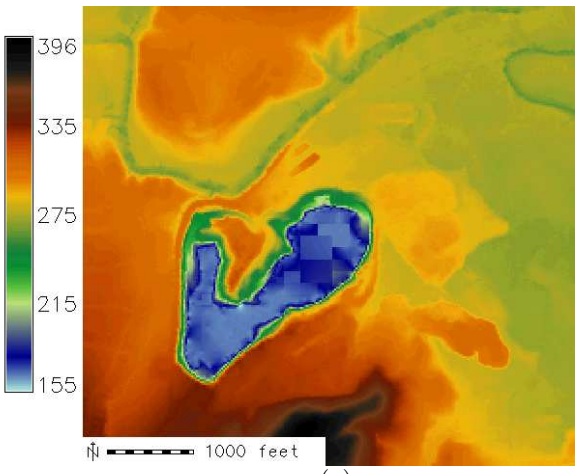


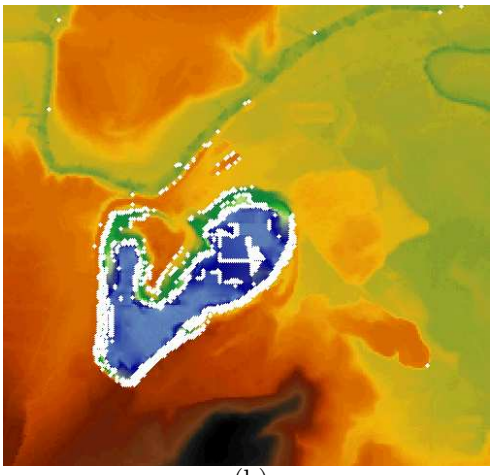
Figure 6.7: Rivers extracted using ε =10 ft (white) and 20 ft (black).

20ft case however, the extracted river (black) follows a different course. Because the default ε=10ft results in a more accurate grid DEM than the 20ft case, we chose the default ε of one-half the grid cell size for our other tests. We found that decreasing ε further only increased the run time of the construction without significantly improving the quality of the DEM.

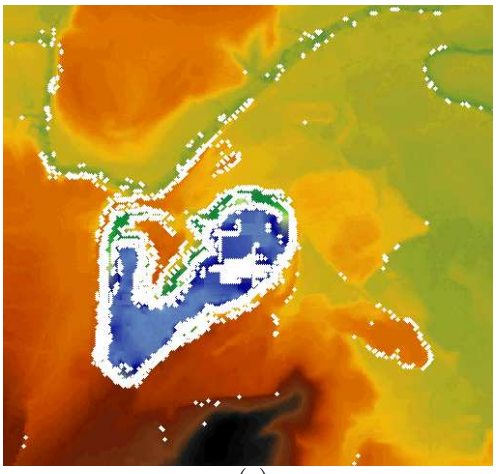
For our final set of experiments on grid construction parameters, we varied the smoothing parameter while using $k_{\max} = 8$ and ε =10ft. A smoothing parameter of 0 results in a interpolated surface that passes exactly through the input points. For a non-zero smoothing parameter, the algorithm constructs an approximation surface in which the input points can deviate from the constructed surface. The default smoothing parameter is 0.1. Smoothing only effects the interpolation routine, and not the quad-tree construction. By increasing the smoothing parameter, we can decrease the number of sinks in the constructed terrain. Our results are summarized in Table 6.5. We compute the RMS deviation by comparing the grids to the base



(a)

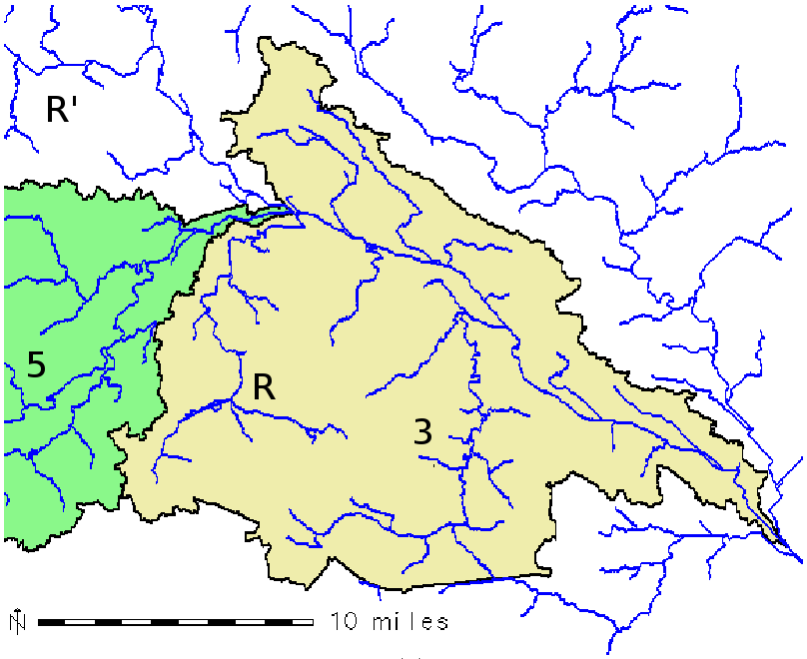


(b)

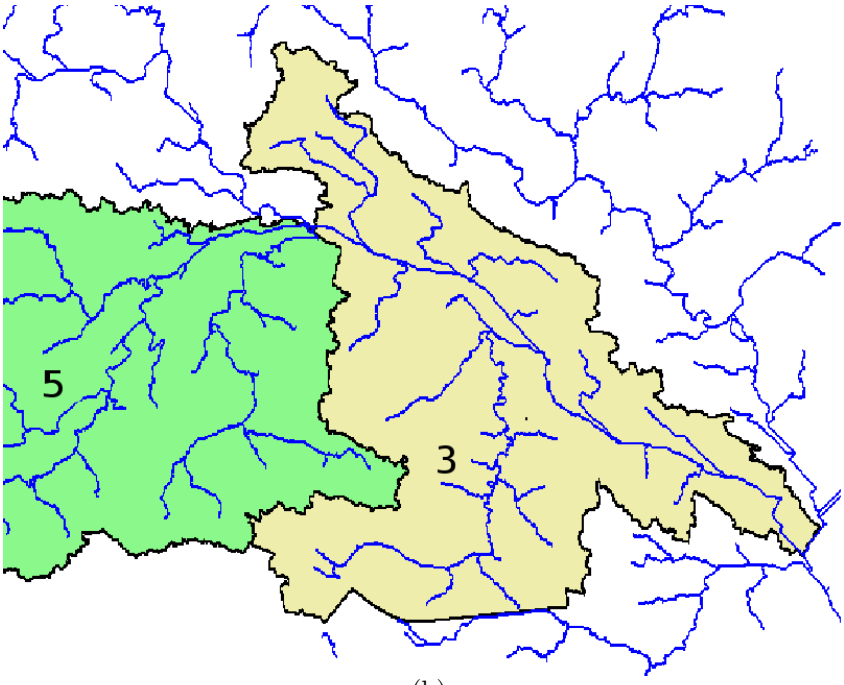


(c)

Figure 6.9: Spatial distribution of deviations from (a) 20ft grid elevations for (b) 10ft grid (c) 40ft grid. White points indicate spots where vertical elevation deviation exceeds 5ft.



(a)



(b)

Figure 6.10: Two watershed regions labeled by Pfafstetter algorithm have quite different boundaries in the (a) 10ft grid and (b) 40ft grid.

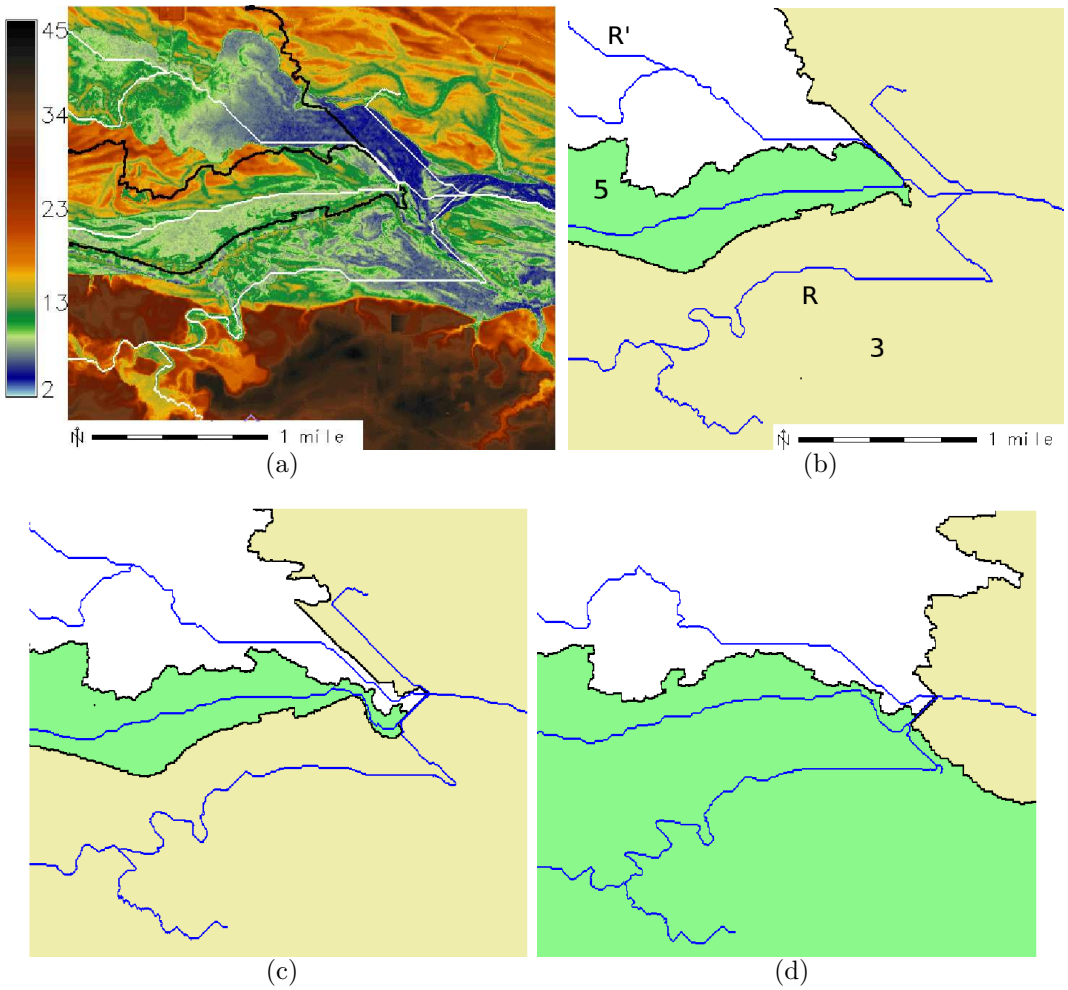


Figure 6.11: A detailed view of Figure 6.10. (a) Base terrain shown at 10ft resolution. Watershed boundaries at (b) 10ft, (c) 20ft, and (d) 40ft grid resolutions

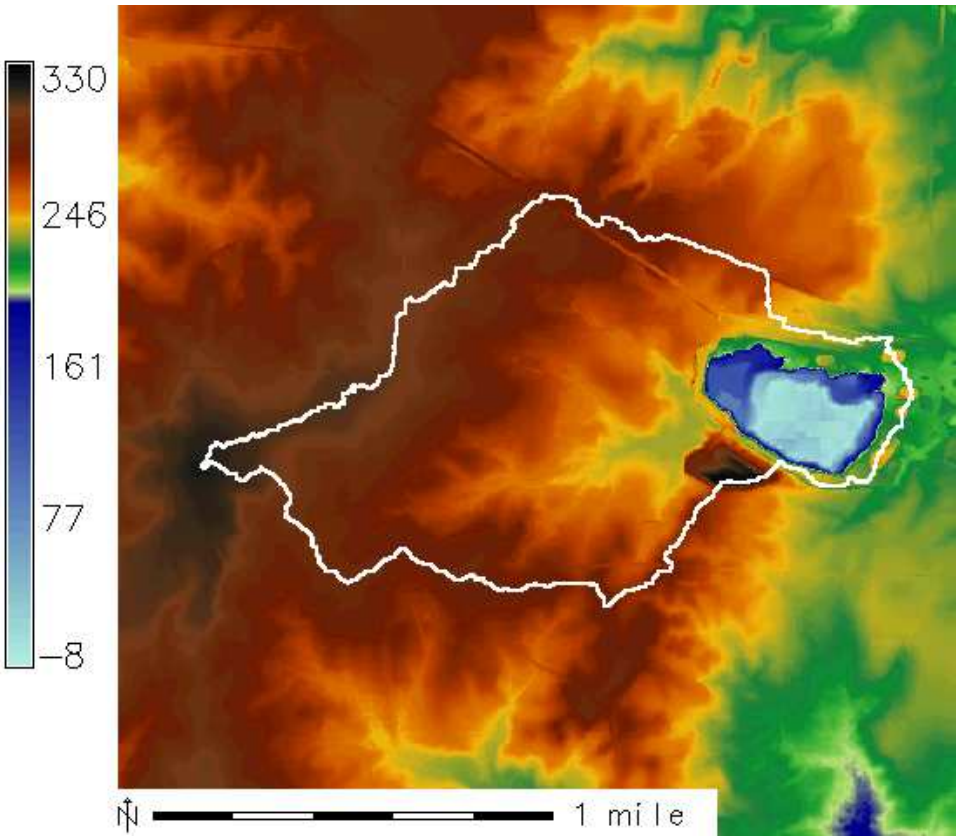


Figure 6.12: A quarry and its 600 acre watershed is preserved with a persistence threshold of 220 feet or less.

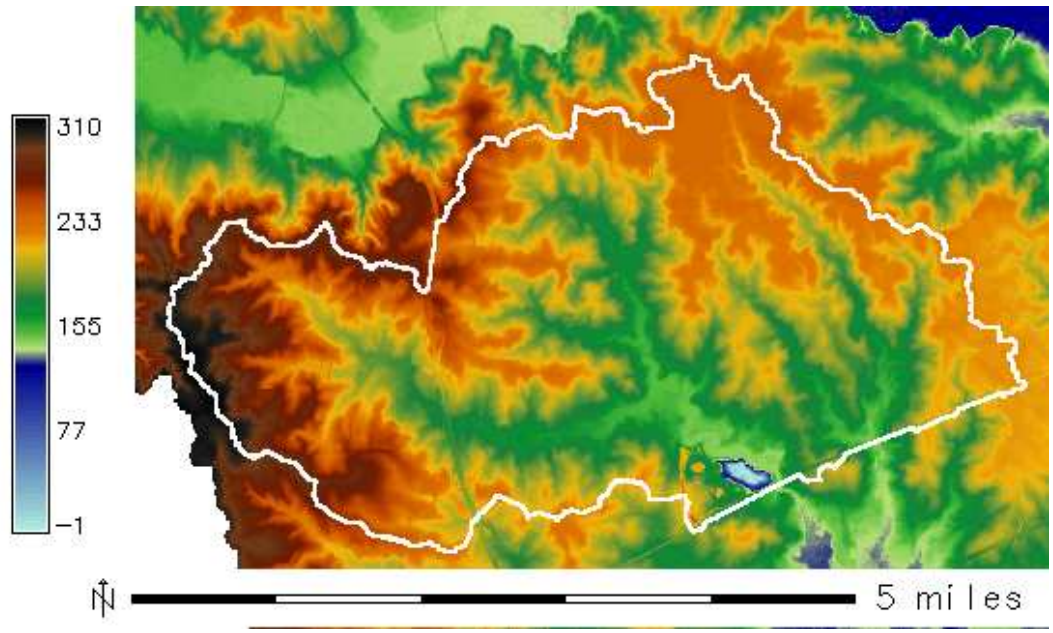
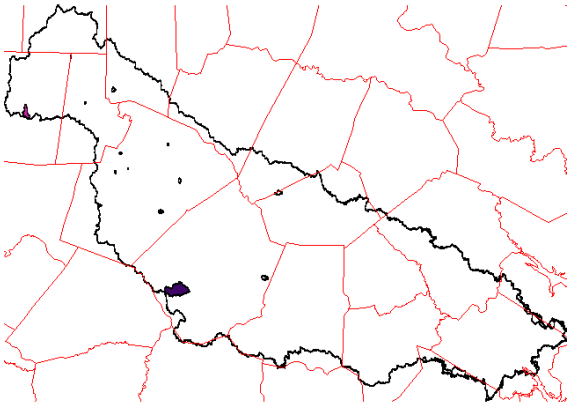
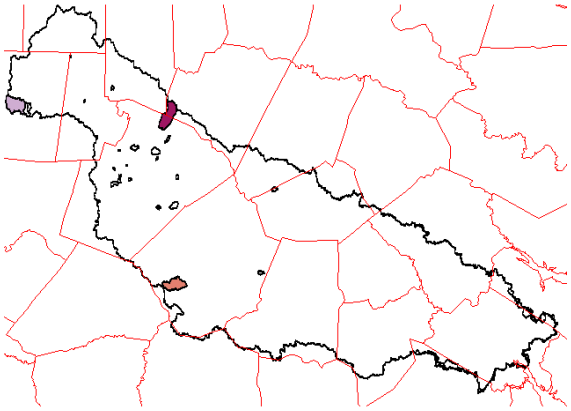


Figure 6.13: The largest (7300 acres) incorrectly computed closed basin, shown in white, for a persistence threshold of 50ft. A bridge in the southeast blocks flow.

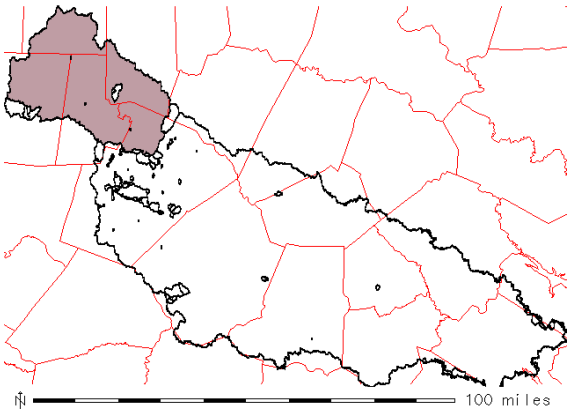
As we lower the persistence threshold, more sinks are kept and fewer are removed. At a persistence threshold of 40ft, 13 additional sinks appear. The drainage area boundaries are shown in Figure 6.14(b). These additional sinks are all examples of small streams being blocked by bridges, but the drainage area of the additional sinks is small and does not dramatically effect the watershed boundaries or the river network. In particular, most of the Neuse river basin drains to a single outlet along the coast in the southeast corner of the figure. However, if we lower the persistence threshold to 30ft, we see dramatic changes in the number and drainage area of the preserved sinks as illustrated in Figure 6.14(c). The most obvious observation is that water upstream of the Falls lake dam, shown in pink shading, is disconnected from the rest of the basin. Also, many more minima appear, especially in urban areas such as Wake county, just South of the disconnected Falls lake basin. A brief inspection of a number of these sinks in Wake county revealed 62 total sinks, 47 of which were caused by bridges blocking rivers, 10 of which were around quarries and five whose



(a)



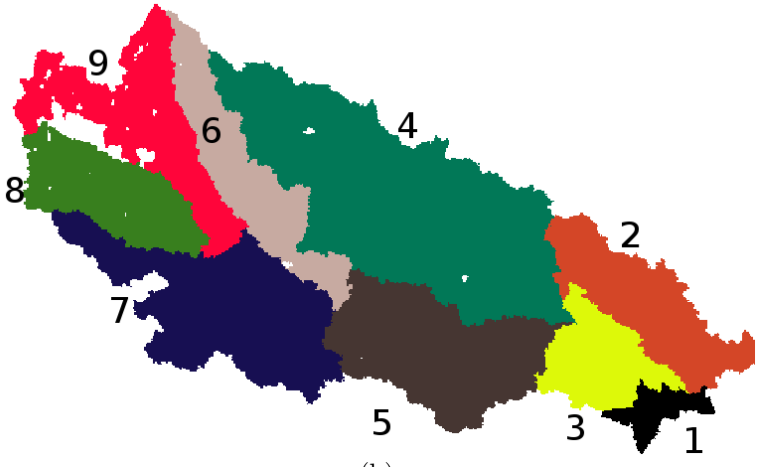
(b)



(c)

Figure 6.14: Drainage area boundaries of sinks shown in black with overlay of North Carolina county boundaries. A persistence threshold of (a) 50ft removes almost all sinks caused by bridges and creates one large primary basin. A threshold of (b) 40ft results in 28 remaining sinks, but the primary basin is intact. For a threshold of (c) 30 ft, the Neuse river basin becomes disconnected at the Falls lake dam (Northwest/shaded), and 96 sinks remain, most of which are due to bridges.

(a)



(b)

Figure 6.15: Pfafstetter basins for (a) persistence threshold of 50ft and (b) 30ft. Even though the headwaters are disconnected in the 30ft case, the ordering of the basin remains unchanged.

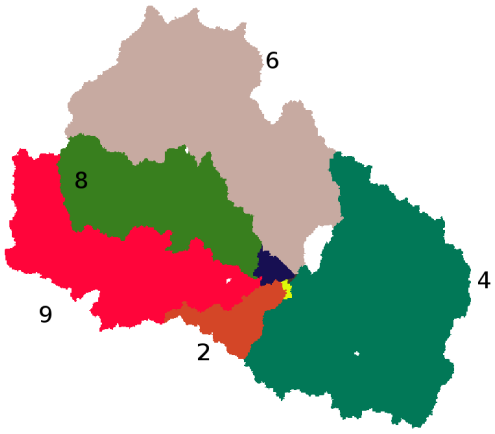


Figure 6.16: Watershed of Falls Lake area when persistence threshold is 20ft. Rivers computed in the southeast region eventually drain to a sink in the center of the image, instead of flowing under the dam which is to the the southeast

threshold of 55ft in this case study preserves all the quarries while routing flow across bridges. This new method of scoring and removing sinks below a threshold score could prove to be a valuable tool for many hydrological studies.

6.8 Conclusions

In this Chapter, we demonstrated that the algorithms presented in this thesis form a scalable and flexible pipeline that efficiently process massive amounts of data derived from modern remote sensing methods. Our primary emphasis in this thesis was on scalable algorithms, but we have seen that our tunable design allows us to explore interesting modeling issues as well. While lidar provides many potential benefits to the GIS community, our experiments highlighted the need for additional work in some areas. Bridges are particularly problematic for hydrological flow routing. We have seen in this Chapter that the topological persistence of most sinks blocked by bridges have a high persistence value, but this value is much lower than features such as quarries. We believe further improvements in th GIS modeling using topological persistence can help identify bridges effectively and lead to improved hydrological