Repairing Hydroconditioning in Pin2Flood

Prepared by Tim Whiteaker

Center for Water and the Environment
The University of Texas at Austin

12 October 2023

Pin2Flood is a Texas Division of Emergency Management application that enables personnel to delineate local flood extent in the field by standing at the edge of flood waters and dropping a pin in an app. For TxDOT, Pin2Flood's curated set of precomputed flood inundation polygons could be used to create a similar precomputed library of roadway flooding under various flood scenarios. However, the Pin2Flood polygons are based on elevation data with many roads and bridges carved out of the landscape at stream crossings to better trace the movement of water, and so the resulting polygons always indicate flooding at stream crossings even when no flood is present. This document presents an ArcGIS workflow for repairing this hydroconditioning of elevation data to produce flood polygons that properly include roads and bridges.

Background

In Pin2Flood, a hydroconditioned digital elevation model (DEM) and a representation of the stream network are used to create a gridded dataset where each grid cell stores the height of that cell above the nearest stream. This is called Height Above Nearest Drainage (HAND). To create a flood polygon for a flood corresponding to a water depth of five feet in a stream (HValue = 5), one simply selects all grid cells associated with that stream which have a HAND value less than five, and then one converts the group of cells into a polygon. This process is repeated for all desired flood stages, producing a stack of flood inundation polygons. For TDEM, these polygons were produced at 1-foot stream depth increments.

If one has access to elevation data representing road and bridge surfaces, as well as the data described above for producing the Pin2Flood polygons, then one can mask out road and bridges areas that should not be flooded in a given scenario. Texas has nearly statewide coverage of LiDAR data, which provides point clouds (LAS files) of elevation data at high resolution and vertical accuracy, where each point is classified according to ground cover type such as car, tree, or ground. By selecting points classified as bridges, culverts, and ground, and by confining the selection to areas known to be roadways, one can create a subsetted LAS that just represents road elevations. Companies such as Ecopia have demonstrated that the areal extent of roadways can be determined via machine learning, providing the confining polygons required for creating the road LAS.

Required Data

The data required to carry out this tutorial are:

- DEM - The hydroconditioned DEM used in Pin2Flood.
- HAND - The HAND grid derived from the DEM.
- Flood polygons - The Pin2Flood polygons. These are named FPZoneRiver and are available in a geodatabase named FP.gdb. Do not use polygons from the FPWeb.gdb geodatabase, since FPWeb is optimized for the Web and not meant for editing.
- Road LAS - The point cloud representing road elevations. This is typically provided in the compressed LAZ format.
- Road polygons - (Optional) Polygons representing the road surface.

Procedure

The following procedure is written for ArcGIS Pro version 3.1. The names of ArcGIS Geoprocessing tools are highlighted **in bold text**. One can use similar tools in other GIS software to achieve the same result.

1. Run **Convert LAS** to convert the Road LAS from LAZ format to LASD (LAS dataset) format, road.lasd.
2. Run **Extract LAS** to project the LAS to match the DEM which is in a USA Albers projection, EPSG:5070. The result is road_albers.lasd. Note: You can try to run LAS Dataset To Raster to skip this step and go straight to the desired projection, but it doesn't seem to honor the snap raster setting in this case.
3. Run **LAS Dataset To Raster** to convert road_albers.lasd to a 3-meter DEM, road_dem_3m, using the original DEM as the snap grid and cell size. With the default settings, this tool interpolates between data points when the points are spaced further apart than the cell size, which could happen if a tree or a large truck was obstructing the LiDAR sensor's view of the ground. However, though the Road LAS only covers the road surface, the gaps outside of the road surface (i.e., everywhere that is not a road) is filled as well.
4. Run **Clip Raster** to clip the raster to areas within the road polygons. This optional step can save on disk space since only the road network is retained, and it also enables one to later analyze all parts of the road surface. If only an overlay of a road line along the surface is desired (e.g., for computing length of inundated road), then clipping is not necessary.
5. Run **Minus** to find the difference between road_dem_3m and the original DEM. Save it as tmpRoadHeight.
6. Run **Set Null** on tmpRoadHeight to set values less than or equal to zero, to Null. Save it as RoadHeight.
7. Run **Plus** to add RoadHeight to HAND. Save it as HAND_healed.
8. Run **Create Feature Class** to create FPZoneRiver_healed, using FPZoneRiver as a template.
9. For each feature in FPZoneRiver:
    a. Read the HValue.
    b. Run **Greater Than** on HAND_healed to find where it exceeds HValue.
    c. Run **Raster To Polygon** on the result.

d. Select where the result polygons have a value of 1.
e. Run **Pairwise Erase** to remove the result from the FPZoneRiver feature.
f. Run **Append** to add the result to FPZoneRiver_healed.
g. Save the mask polygon to a temporary feature class, and compute an HValue field.

10. Optional: Merge the mask polygons to an output feature class named Mask. Perform this step if one wants a record of the masks used.