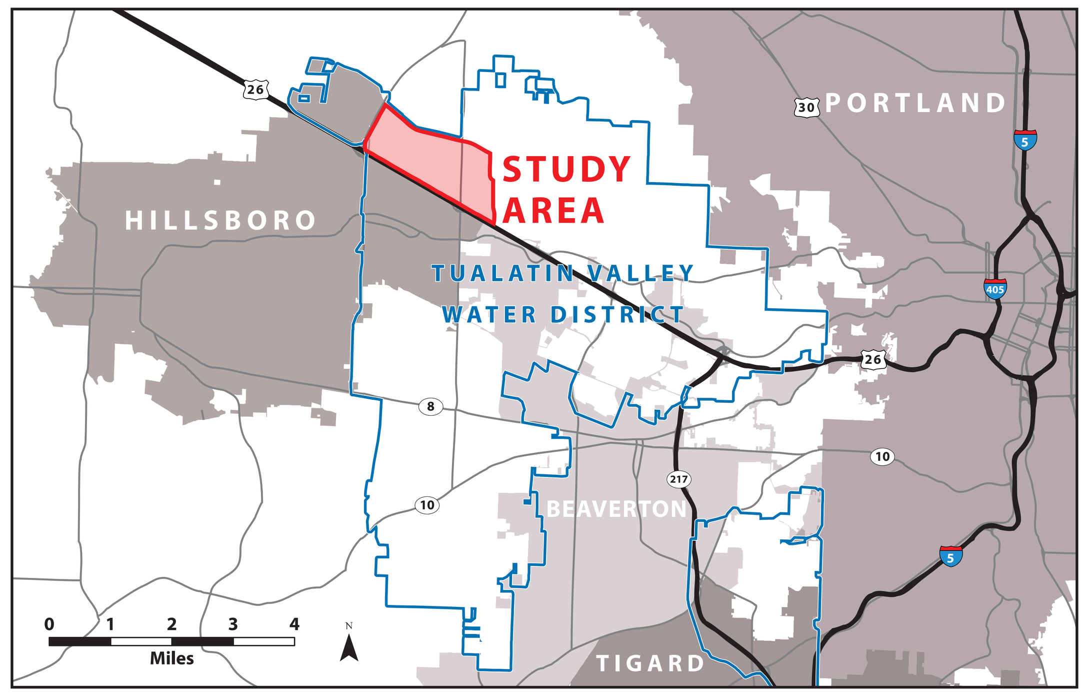
HydrantTools
An ArcGIS Python Toolbox

Introduction

Water providers are tasked with the important responsibility of ensuring the adequacy of fire hydrant coverage throughout their waters systems. However, the task of identifying areas lacking sufficient hydrant coverage is not trivial. A solution needs to be able to model access along real paths, such as a street network, and must also consider barriers to access. More so, a solution needs to be repeatable, and easily implemented. Thus, this study asks:

Can GIS be used to effectively model hydrant coverage? Can python automation make a solution easy and repeatable?



Study Area

The area chosen for this study is the Rock Creek Neighborhood in unincorporated Washington County, Oregon lying just outside the city limits of both Hillsboro and Beaverton, suburbs of Portland. The neighborhood is delineated by Highway 26 to the south, NW 174th Avenue to the east, NW Cornelius Pass Road to the west, and NW West Union Road to the north. The entire extent of the neighborhood falls within the Tualatin Valley Water District.

Data Used

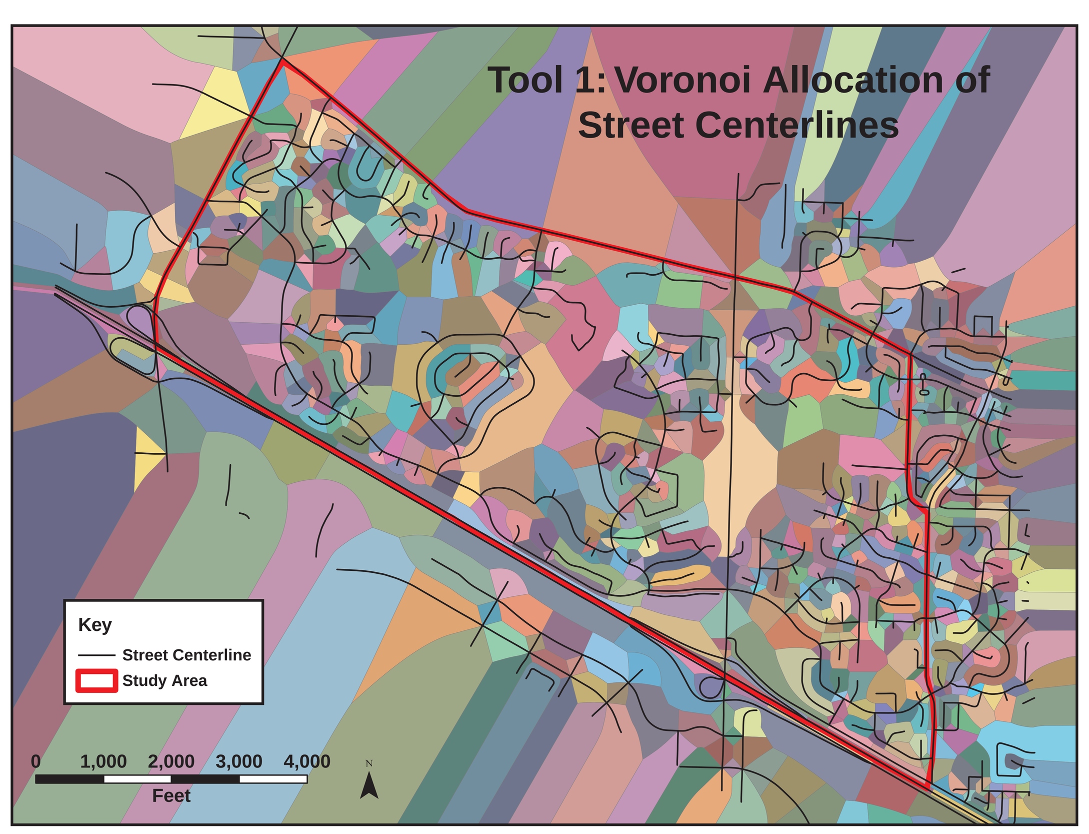
Hydrant data for this project were compiled from a number of sources. Most of the points were collected using RTK GPS surveying, and were provided by Tualatin Valley Water District (TVWD). Hydrants TVWD had not GPSed either were gathered from a dataset of GPS points from Washington County Fire District #9, or were digitized using Goolge Street View. The remaining datasets—cities, tax lots, building footprints, and street centerlines—came out of the Oregon Metro Regional Land Information (RLIS) database.

Data Reference:

Metro Data Resource Center. Regional Land Information System (RLIS). *cty_fill, streets, buildings, taxlots* [Shapefile geospatial data files]. Portland, OR: Metro (May 2014).

The HydrantTools Toolbox

The analysis is easily broken down into five steps, each of which was programmed as a tool inside a python toolbox for ArcGIS, called HydrantTools. The toolbox is available open-source under an MIT license, and is easy to use; all that is required are datasets for hydrants, tax lots, street centerlines, and buildings. Each of the five tools is detailed below.



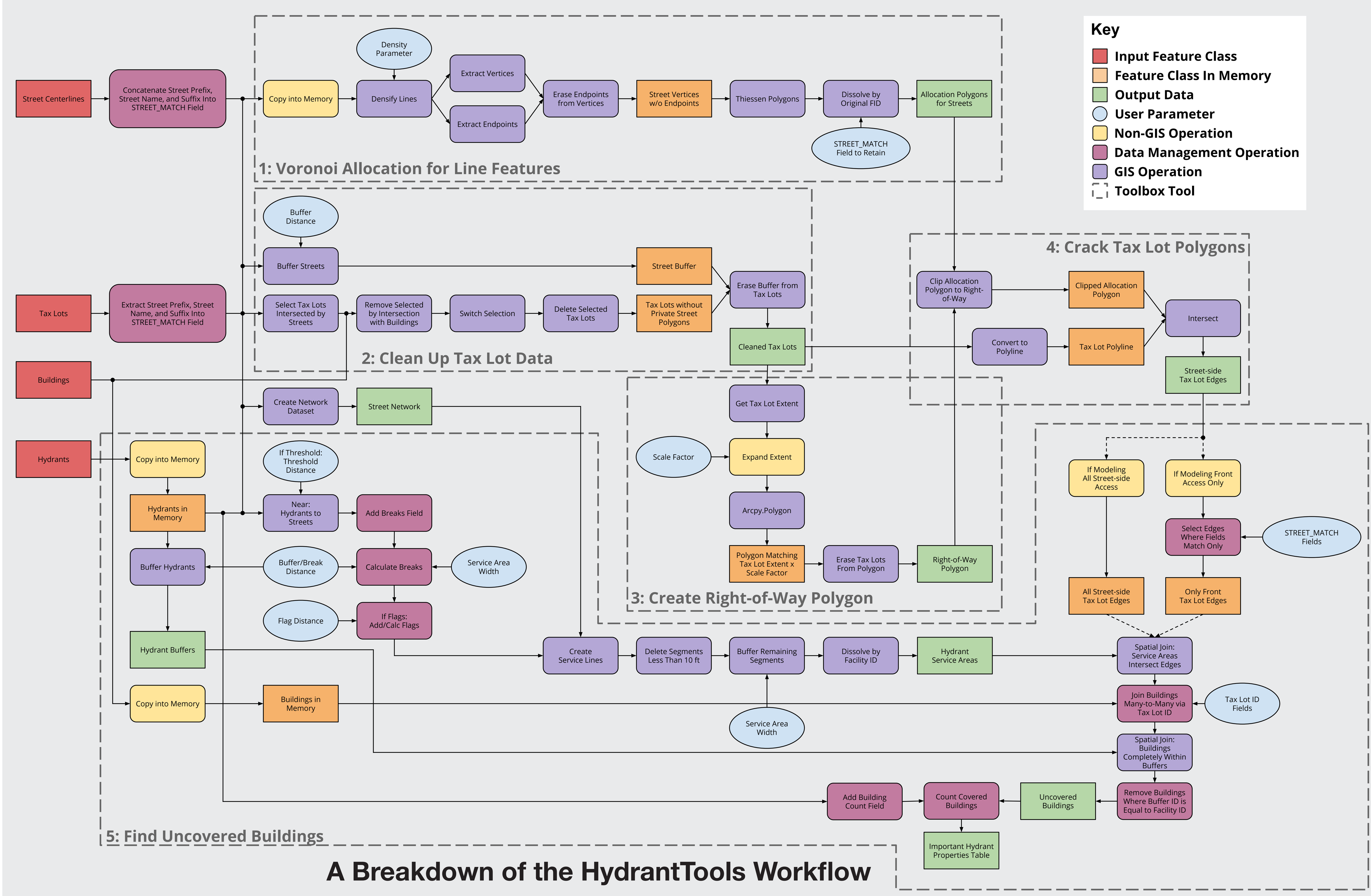
Tool 1: Voronoi Allocation of Street Centerlines

The first tool in the toolbox creates allocation polygons for the street centerlines. This allocation dataset is used with Tool 4 as a way to find the "corners" of tax lots at intersections, where the edges of tax lots can be split. As the Thiessen polygon tool included with ArcGIS does not work with line features, a new solution was devised which extracts vertices from the streets, finds the Thiessen allocation for each of those points, then dissolves those polygons by street.

Acknowledgements

I'd like to thank James Kivley and Tualatin Valley Water District for providing the opportunity for me to work on this project, for supporting me with the development of these tools, and for providing the hydrant data. I'd also like to thank Geoffrey Duh for providing feedback and ideas in the refining of the analysis methods, and Jon Erickson, a good friend and firefighter who contributed his knowledge of fire fighting.

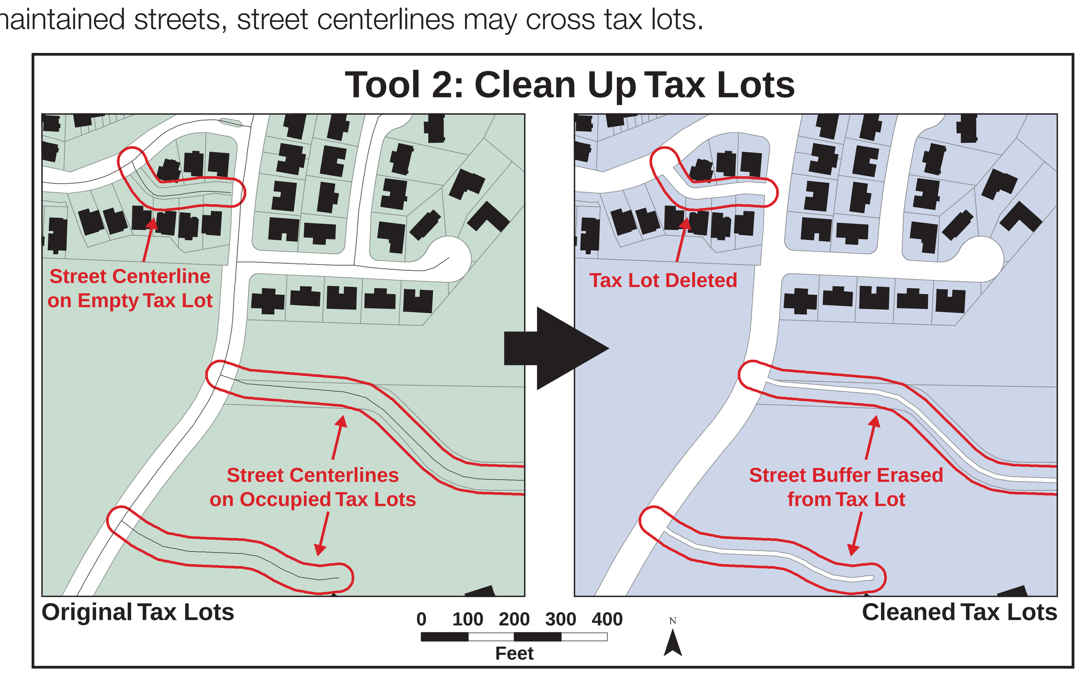
By Jarrett Keifer: jarrettkeifer@gmail.com



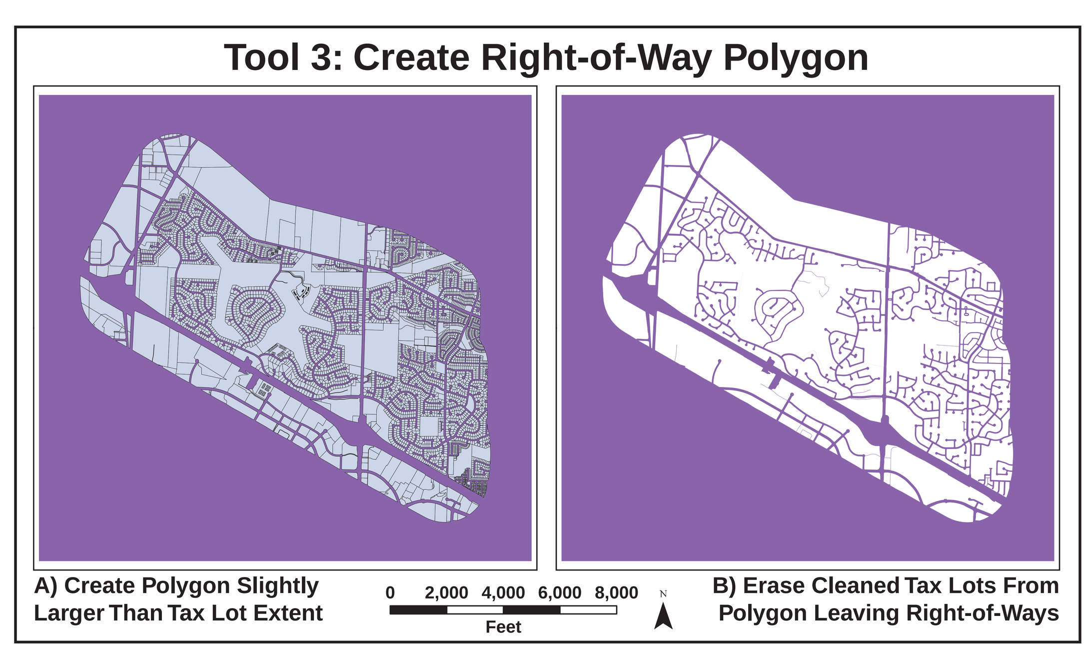
A Breakdown of the HydrantTools Workflow

Tool 2: Clean Up Tax Lots

The second tool in HydrantTools automates the cleanup of tax lot data. Later operations assume that all street centerlines are in the right-of-way. However, in some cases such as privately-maintained streets, street centerlines may cross tax lots.



To fix this problems, the tool deletes any tax lots that are intersected by street centerlines, but do not contain any buildings. For tax lots which contain both streets and buildings, the intersecting street segments are buffered by a user-defined distance, and this buffer is erased from any intersected tax lots.



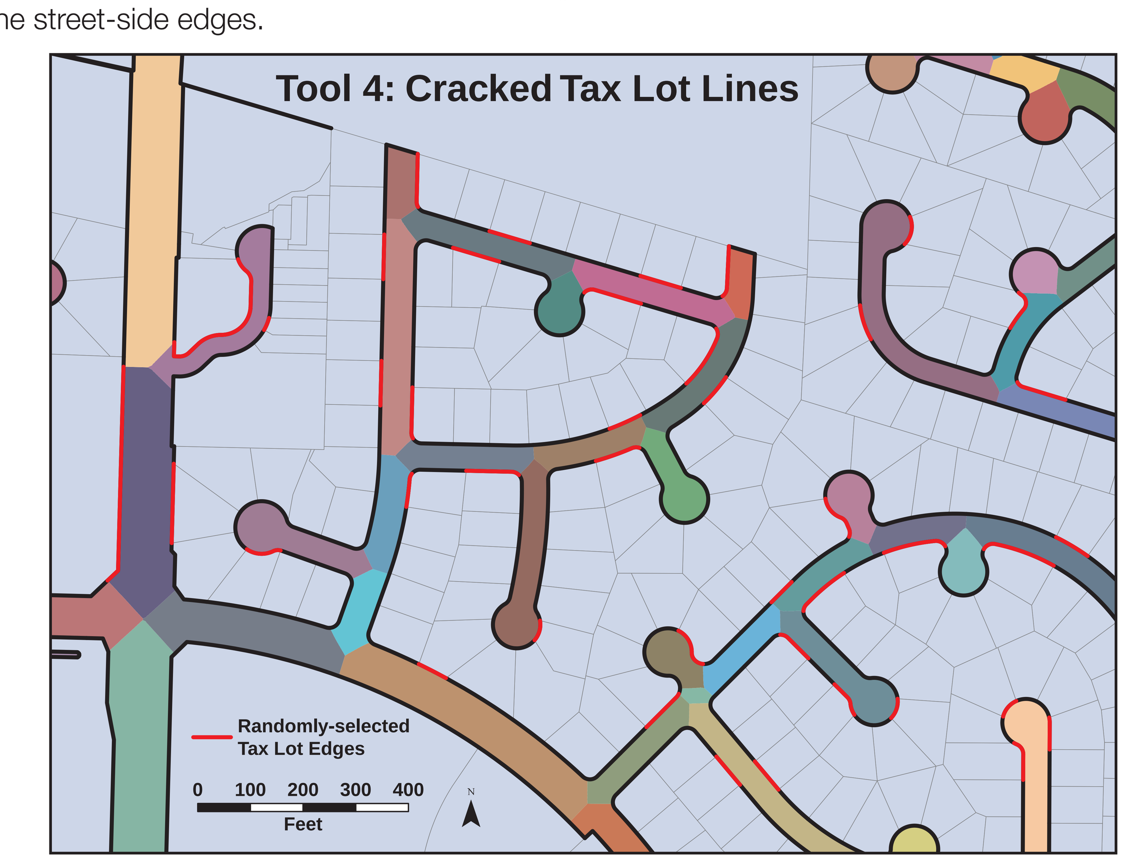
Tool 3: Create Right-of-Way Polygon

A right-of-way polygon is needed for the next tool, as it is used to clip the street allocation layer. To create this polygon, the tool finds the extent of the tax lot data, expands the extent by a user-provided scaling factor, then draws a polygon with these coordinates. The tax lots are then erased from this polygon, leaving only right-of-ways.

Tool 4: Crack Tax Lot Polygons

One of the main goals of this project is to model hydrant access along real paths, and to consider barriers, such as fences, walls, or slopes. To do this, we need to find the street-side edges of tax lots, and from those edges we must be able to select the front sides only.

To begin, tax lot polygons must be converted to polylines. This changes the polygons from solid objects into hollow line features, which we can consider to be the tax lot edges. At this point each tax lot is still represented by a single feature, so we are not yet able to pull out just the street-side edges.



The next step of this tool is to clip the street allocation polygons from Tool 1 to the right-of-ways using the polygon produced from the execution of Tool 3; the result you can see in the map above. The allocation polygons divide each intersection evenly, and the division lines extend outward from the intersection centers such that we can use them to divide the tax lot polylines. This provides a way to crack the polylines at street corners.

It turns out this is easily accomplished by intersecting the clipped allocation polygons with the tax lot polylines. Doing so also removes all the non-street-side tax lot edges. The result is shown in the map above as the bold black line features. A random selection of the edges is also shown in red to highlight how the edges are divided at corners. Additionally, the intersect joined the attributes of the tax lots with the streets, so the next tool can eliminate non-front edges by matching the street names from each.

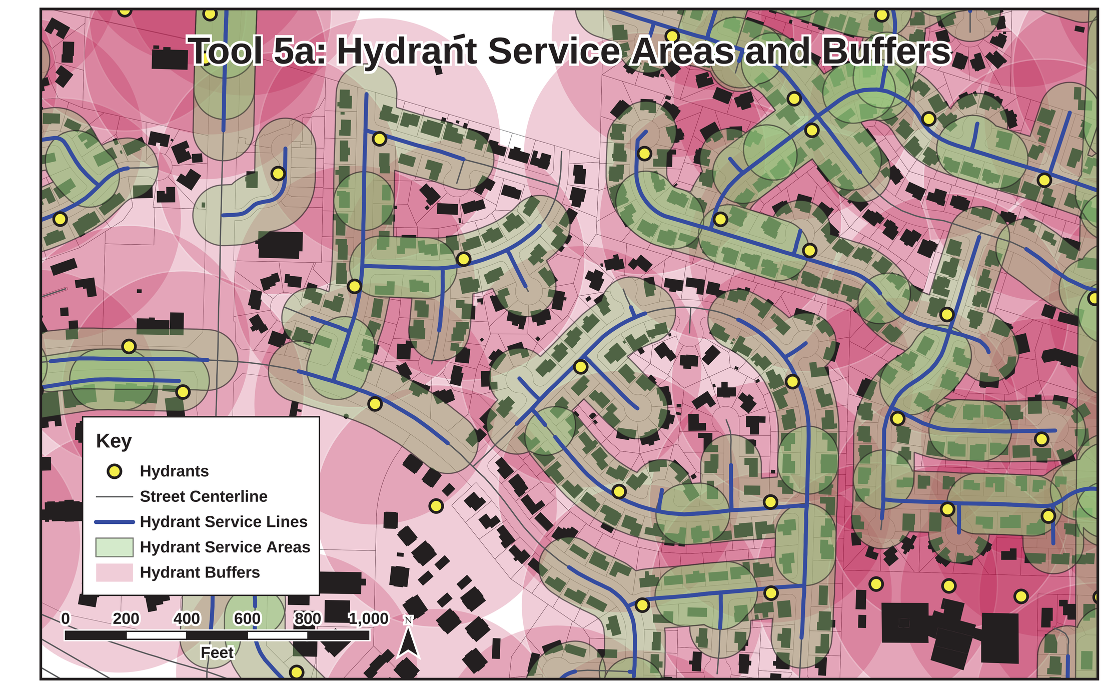
Tool 5: Find Uncovered Buildings

The final and most significant tool in the HydrantTools toolbox finds all buildings that fall outside the provided distance from a hydrant. For this analysis, a distance of 400 feet from a hydrant was used, as this is the distance specified by TVWD. This tool uses is a hybrid buffer/network approach: the tool generates service areas following the street centerlines, selects buildings which intersect those service areas, then ensures each of the buildings are entirely within the specified distance using buffers generated for each hydrant.

Generating the Hydrant Service Areas

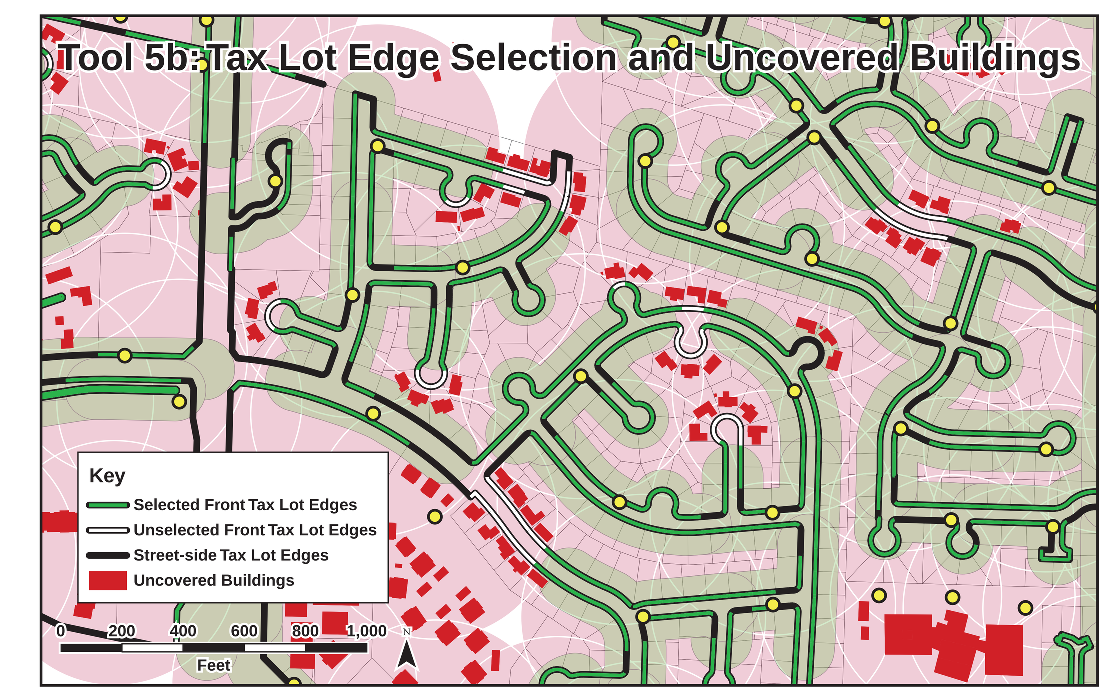
HydrantTools uses the ArcGIS Network Analyst service area tool to create service lines for each of the hydrants. The breaks for the service lines are calculated from the user's buffer distance by subtracting the width of the service areas, also specified by the user, and subtracting the distance between the hydrant and the street centerline. Therefore, in the case of this analysis with 400-foot buffers and 100-foot service area width, a hydrant 25' from the centerline will have service lines which traverse the street network 275 feet.

Once the service lines are created, any segments less than 10 feet are deleted to clean up the final shapes of the service areas. The remaining segments are dissolved together by hydrant ID. With a single line feature for each hydrant, the lines are then buffered by the user's service area width. This produces the final hydrant service areas; these, along with the hydrant buffers, are shown in the map below.



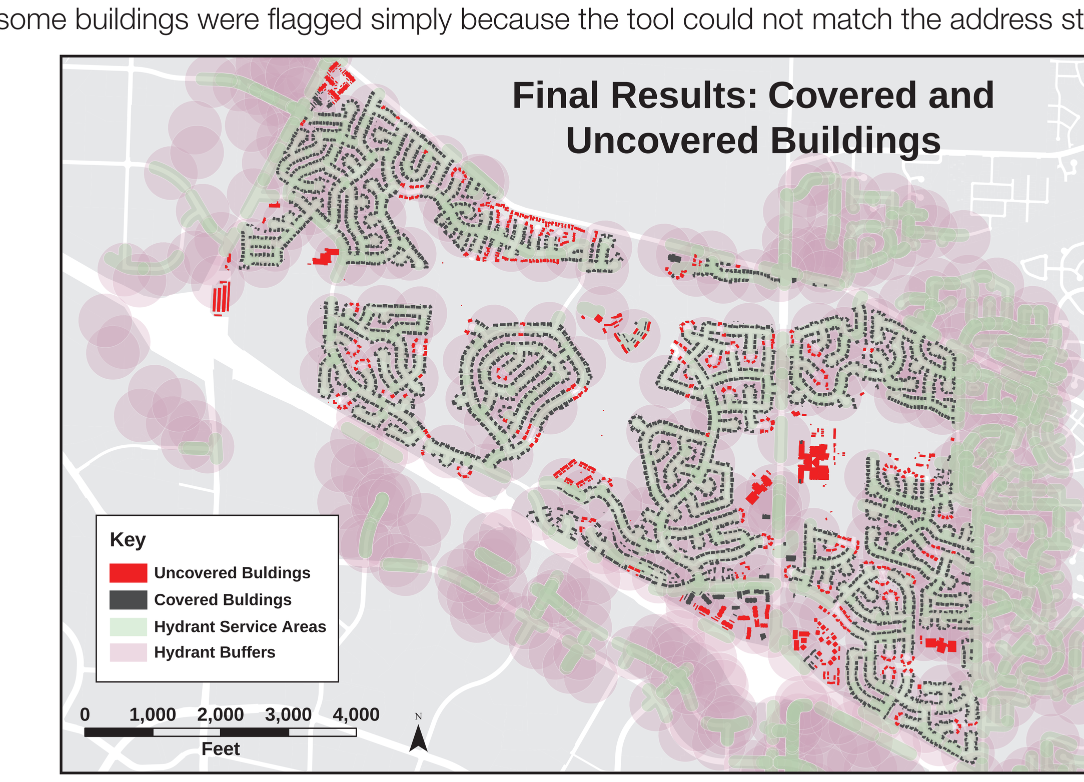
Finding Uncovered Buildings

To model access to the front sides of tax lots only, the tax lot edges layer is queried to find only the edges where the tax lot street prefix, name, and suffix match the intersected street allocation polygon's street prefix, name, and suffix. These front edges are then spatially-joined to the hydrant service areas where they intersect. Then, a many-to-many join is made between the edges and the building footprints using the tax lot ID as the join field. Finally, another spatial join is performed between the hydrant buffers and the joined buildings, where buildings are completely contained within buffers. With all these joins, the covered buildings can be found: where the buffer's original FID (from the hydrant) is equal to the facility ID of the service area (also from the hydrant), a building is covered. All covered buildings are then deleted from the buildings feature class, leaving only the uncovered buildings.



Final Results

Though further study is needed to fully understand the limitations of these tools, this project is unquestionably a success. Not only does HydrantTools provide a robust and repeatable method for finding gaps in hydrant coverage, but it is quick and easy: performing this analysis took only ten minutes! The final count came to 564 buildings potentially uncovered and in need of further review, out of 3,384 total. A brief assessment revealed many apartment buildings and large commercial/public buildings were flagged, though they likely have adequate fire protection. Other errors show that the tax lot data does not have complete address information, and some buildings were flagged simply because the tool could not match the address streets.



How to get HydrantTools:

HydrantTools is **free** and open-source! To download it, go to:

jkeifer.github.io/HydrantTools