

Lab Report

Title: St. Paul Road Expansion Project (Lab 0)

Notice: Dr. Bryan Runc

Author: Mohsen Ahmadkhani

Date: 9/22/2021

Project Repository: <https://github.com/mohsen-gis/GIS5571.git>

Google Drive Link: -

Time Spent: 12 hours

Abstract

The department of transportation of the city of St. Paul has issued a request to identify the buildings that fall (partially or entirely) within the neighborhood of 10 meters of each street in the city. To address this problem the first step is to generate a buffer of 10 meters around each and every street in the city. And the second step is to download the building footprints and then run the spatial intersection analysis. The current project addresses the very first step that is performing a buffer analysis for the road network of St. Paul. I intended to use three different products of ESRI (i.e., ArcPy, ArcGIS Pro, and ArcOnline) to accomplish the project. I also compared the performance and results of the process using each platform. After running the project, the results showed that ArcPy was the fastest product of ESRI with 28 seconds of execution time followed by ArcGIS Pro with 35 seconds and ArcOnline with more than 2 minutes. However, it has the least user interactivity level and no graphic interface. And ArcGIS Pro was the easiest way of accomplishing this task thanks to the efficient GUI and design of its workspace.

Problem Statement

The St. Paul department of transportation wants to know which building blocks are within the 10 m vicinity of each street in the city. So, they can run a road expansion project accordingly. To perform this spatial analysis we want to experiment, compare and contrast three different products of ESRI namely ArcPy, ArcGIS Pro, and ArcGIS Online.

Table 1. The list of required data sets for the proposed study.

#	Requirement	Defined As	(Spatial) Data	Attribute Data	Dataset	Preparation
1	St. Paul Road Network	The road network dataset containing all streets and roads in the city	Road network (shapefile)	Name, oneway, bounds, etc	Open Street Map (osmnx)	

Input Data

There are multiple sources to download the road network of St. Paul. I used Open Street Maps python package named osmnx to download this data. I used this method because: 1. Its data is

reliable, 2. Accessible from everywhere, so, I don't need to carry the input data with the notebook as a supplementary file. 3. This package allows the users to change the area of interest free of charge. So, the user can expand or limit the area of study as needed.

Table 2. Input data description.

#	Title	Purpose in Analysis	Link to Source
1	St. Paul Road Network	Raw input dataset for performing buffer analysis.	osmnx

Methods

As illustrated in the data flow diagram of Figure 1, this project has four major steps. First, I used osmnx package to download the road network as a MultiDiGraph dataset. Then I used graph_to_gdfs() function to convert it to Geopandas data frame (GeoDataFrame) and saved it as a shapefile for further processing. Second, I used ArcPy module to perform a buffer analysis in place using the resulting GeoDataFrame. To accomplish this task, I used arcpy.Buffer_analysis() function. Third, I added the saved shapefile to the workspace of ArcGIS Pro to perform the buffer analysis using that platform. I used buffer analysis from Geoprocessing Toolset. Forth, I used ArcOnline to process the buffer analysis task. I used both GUI and notebook environments of ArcOnline to craft this analysis.

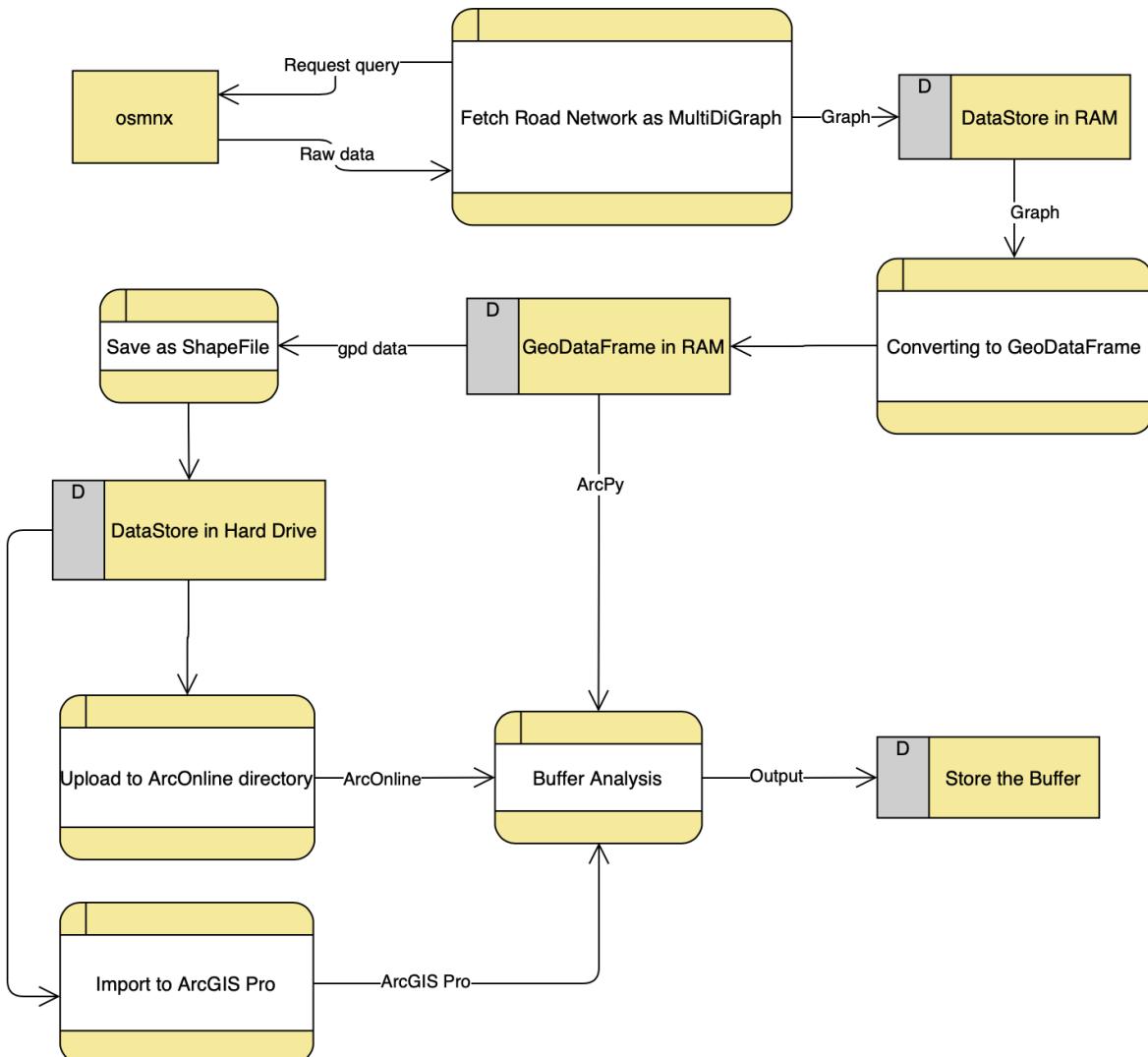


Figure 1. Data flow diagram.

Results

Figure 2 shows a visualization of the collected data. Also Figures 3 to 5 depict the result of the processes using different products of ESRI. In terms of execution time, the `ArcPy` package with about 28 seconds was the fastest among all. `ArcGIS Pro` stood second with about 35 seconds and `ArcOnline` with more than 2 minutes was the slowest platform. However, in terms of user interactivity and ease of use, `ArcGIS Pro` with its convenient GUI was relatively easy to use. One notable difference between the `ArcGIS Pro` and the other methods was that this method draws a buffer around each line independently. In the sense that each line of a polyline got buffered separately (see Figure 6). However, the result of `ArcPy`, for example, was different and polylines were buffered. Another notable difference was that the output of `ArcGIS Pro` was well detailed, and each polyline had its own record while `ArcPy` result generated a single polygon for the entire network. There are specific parameters to pass if we want to have exactly similar results.

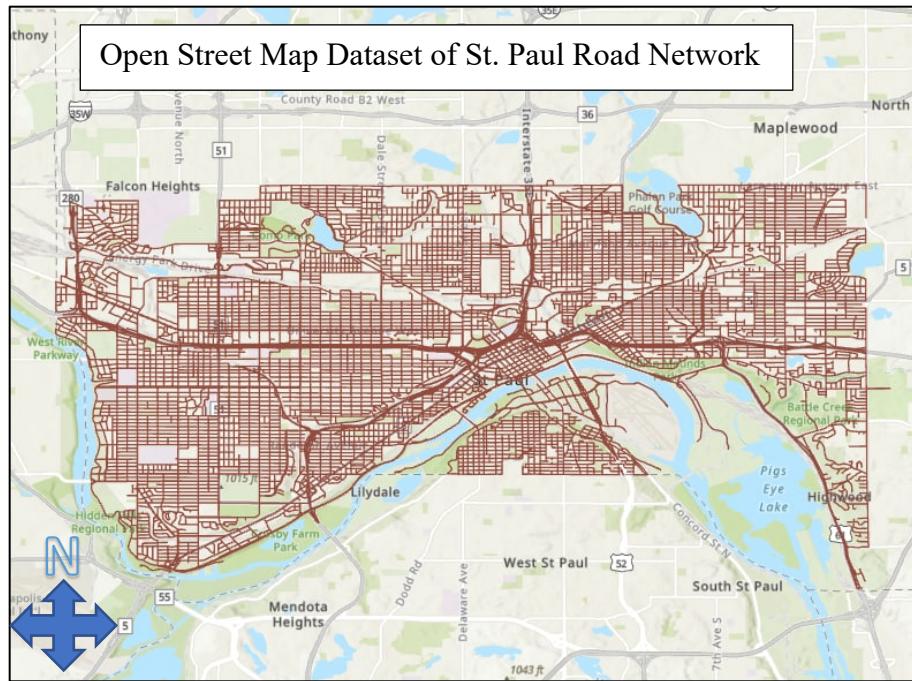


Figure 2: The road network of St. Paul downloaded from OSMnx dataset.

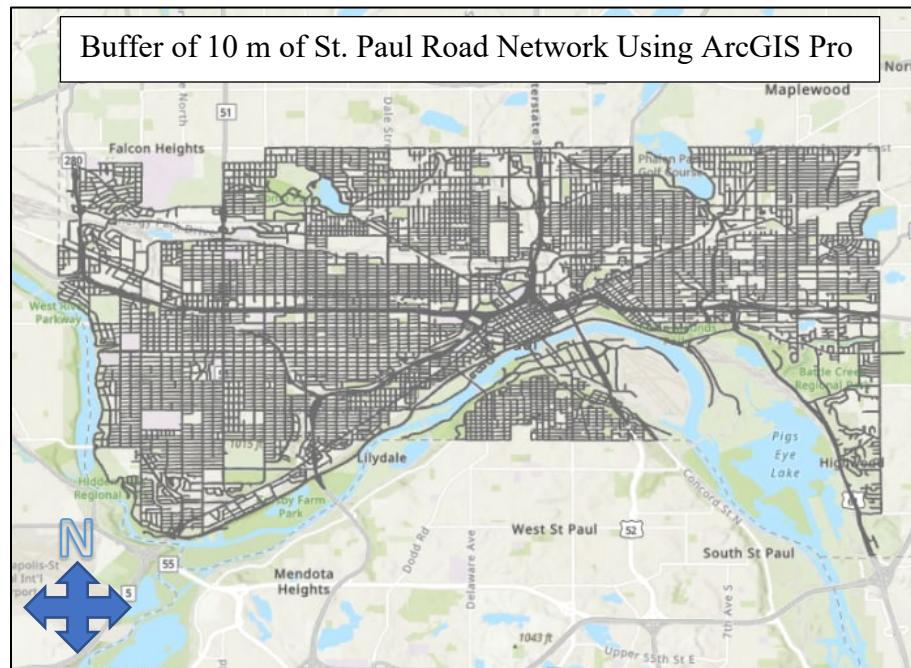
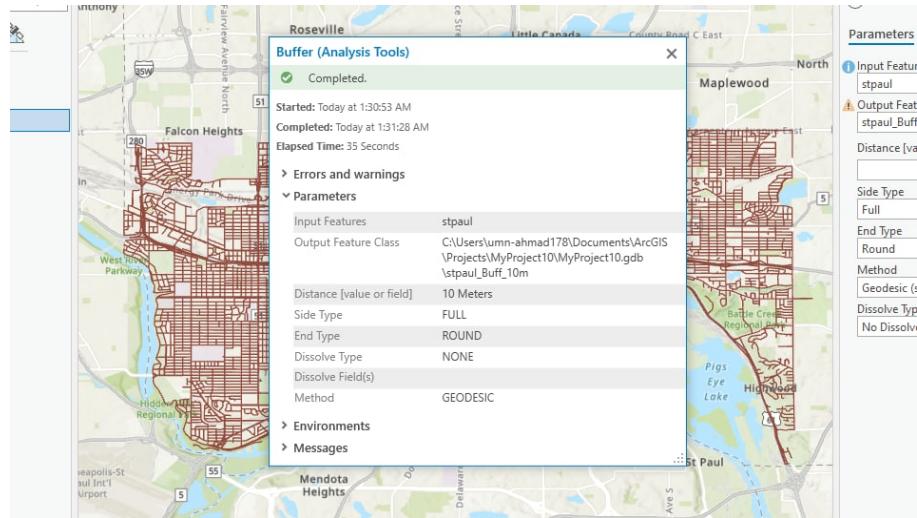


Figure 3: The process of buffer analysis using ArcGIS Pro

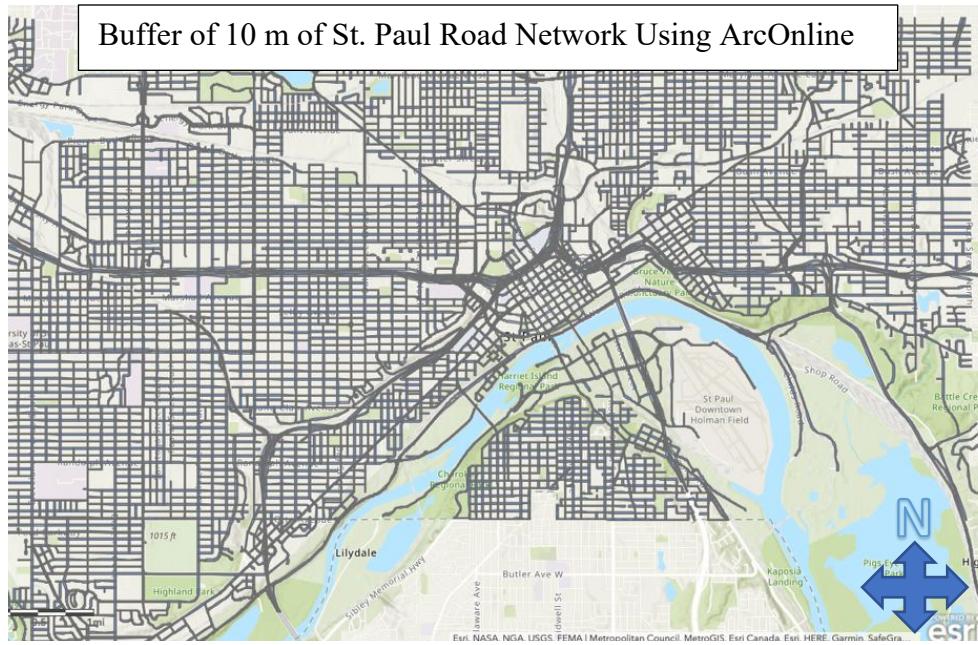


Figure 4: Buffer of 10 m of St. Paul Road Network Using ArcOnline

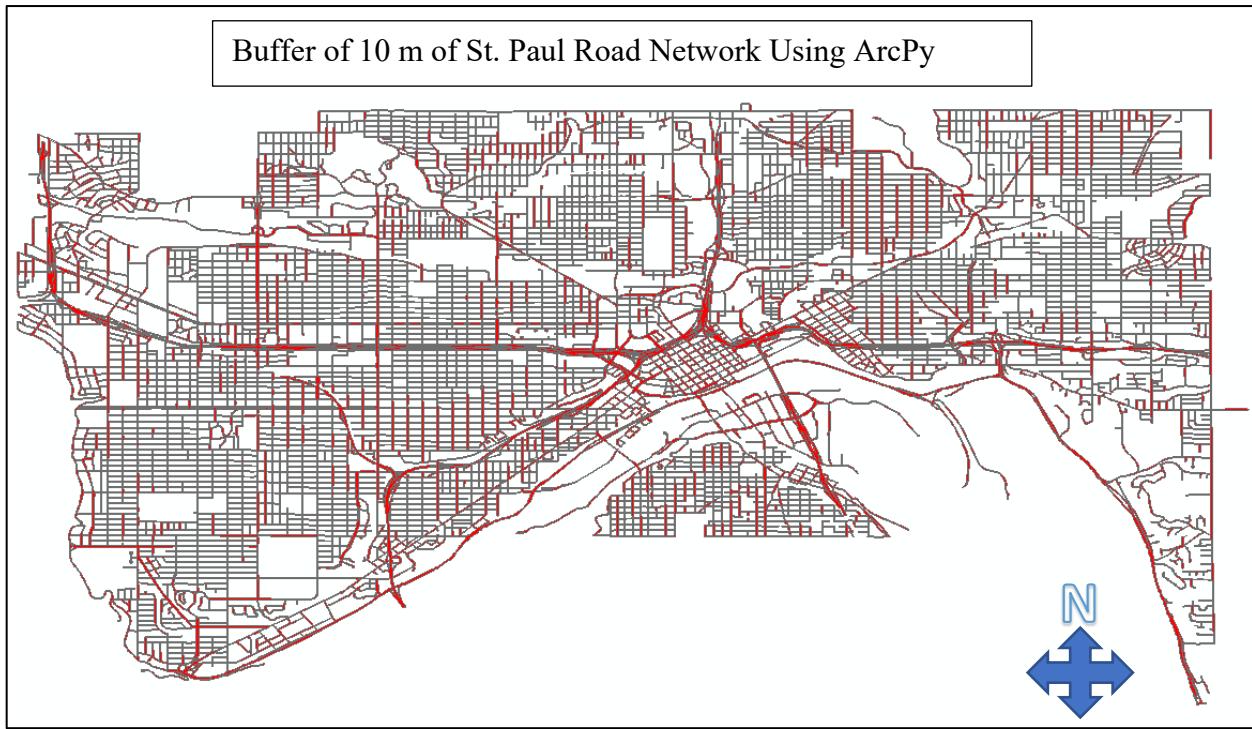


Figure 5: Buffer of 10 m of St. Paul Road Network Using ArcPy (intentionally removed the basemap for a better visualization of the polygons).

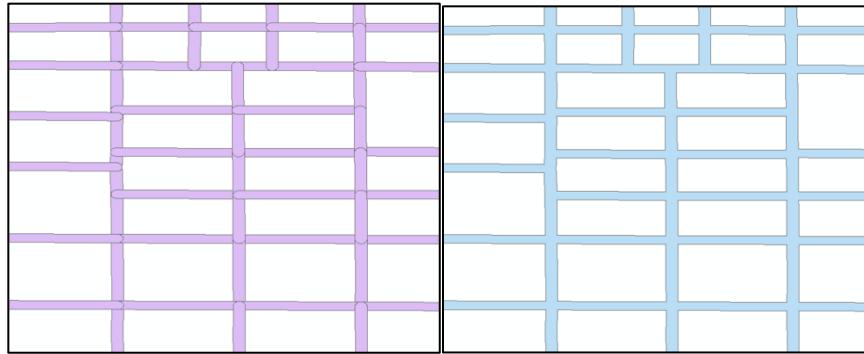


Figure 6: Left: an instance of ArcGIS Pro result. Right: an instance of ArcPy result that considers the whole network to draw a buffer.

Results Verification

To evaluate the results of the processes I compared all the results on a same map. Since all the maps overlapped each other, I could conclude that the results are consistent. Also, regarding the coordinate system of EPSG: 4326, the measuring unit was meters. The buffered distance also visually makes sense when looking at the results. Furthermore, using the Measure module of ArcMap, it is confirmed that the generated buffer has a width of 20 meters which is generated from a buffer radius of 10 meters. Figure 7 demonstrates the evaluation part.

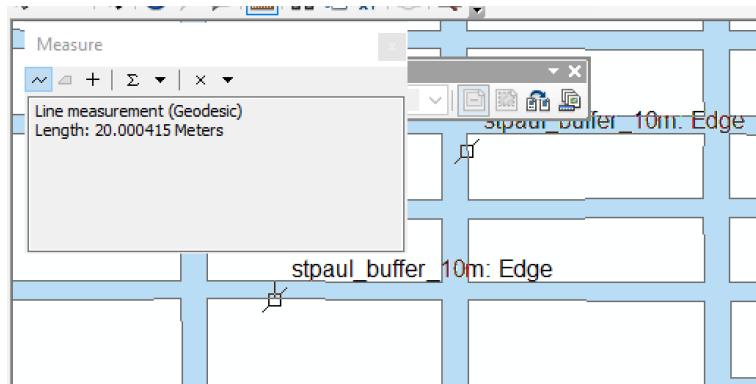


Figure 7: The measurement of the generated buffer around the line features.

Discussion and Conclusion

GitHub

I was already familiar with GitHub from a few previous projects. However, it'd been a while since I used the git commands to put my files in a repository. In the process, it worked well when I used git clone, git pull, and git add ., however, it failed when I tried to git push. It was facing an authentication failure as a fatal error. After reading the Git documentation [1], I learned that the authentication policies have been changed since August 13th, 2021. In the sense that I no longer was able to access my repositories using my regular password. Instead, I generated a token and used it for git operations (i.e., git push and git commit).

Spatial Analysis

This analysis showed that how ESRI products are accessible from everywhere and how convenient it is to use them to perform spatial analysis. Particularly, ArcOnline notebooks and

platform are so valuable in the sense that they provide the user a next level of interactivity with maps and spatial data. This is an ability that is consistent with the pace of technology and web evolution.

From ArcGIS Pro and ArcPy I learned that even though ArcGIS Pro provides a nice user-friendly GUI environment, it is slower and less efficient compared to ArcPy python package. Also, looking at the data flow diagram drawn in Figure 1, the process of buffer analysis for ArcOnline is so much slower because there is a middleware (i.e., ArcGIS Server) module inside the process box that is hidden. So, flowing data within that hidden box takes much longer compared to the rest.

References

1. Matthew Langlois, Token authentication requirements for Git operations [accessed on September 21, 2021] <https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/>
2. ESRI, A quick tour of ArcPy [accessed on September 21, 2021] <https://desktop.arcgis.com/en/arcmap/latest/analyze/arcpy/a-quick-tour-of-arcpy.htm>
3. ESRI, Publish hosted feature layers [accessed on September 21, 2021] <https://doc.arcgis.com/en/arcgis-online/manage-data/publish-features.htm>
4. ESRI, Portal for GIS [accessed on September 20, 2021] <https://gislab.depaul.edu/portal/portalhelp/en/portal/latest/use/geoanalytics-create-buffers.htm>
5. Shannon Kalisky, The ArcGIS Notebooks public beta for ArcGIS Online is here. [accessed on September 20, 2021] <https://www.esri.com/arcgis-blog/products/arcgis-online/analytics/arcgis-notebooks-public-beta/>
6. ESRI, Introduction to the Spatially Enabled DataFrame [accessed on September 20, 2021] <https://developers.arcgis.com/python/guide/introduction-to-the-spatially-enabled-dataframe/>
7. Abdi Shakur, Retrieving OpenStreetMap data in Python [accessed on September 19, 2021] <https://towardsdatascience.com/retrieving-openstreetmap-data-in-python-1777a4be45bb>

Self-score

Category	Description	Points Possible	Score
Structural Elements	All elements of a lab report are included (2 points each): Title, Notice: Dr. Bryan Runck, Author, Project Repository, Date, Abstract, Problem Statement, Input Data w/ tables, Methods w/ Data, Flow Diagrams, Results, Results Verification, Discussion and Conclusion, References in common format, Self-score	28	26
Clarity of Content	Each element above is executed at a professional level so that someone can understand the goal, data, methods, results, and their validity and implications in a 5 minute reading at a cursory-level, and in a 30 minute meeting at a deep level (12 points). There is a clear connection from data to results to discussion and conclusion (12 points).	24	20

Reproducibility	Results are completely reproducible by someone with basic GIS training. There is no ambiguity in data flow or rationale for data operations. Every step is documented and justified.	28	28
Verification	Results are correct in that they have been verified in comparison to some standard. The standard is clearly stated (10 points), the method of comparison is clearly stated (5 points), and the result of verification is clearly stated (5 points).	20	20
		100	94