Lab Report

Title: Walking Path Optimization using Suitabilty Analysis
Notice: Dr. Bryan Runck
Author: Luke Zaruba
Date: 2 November 2022

Project Repository: Link Here
Time Spent: 12.0 hours

Abstract
In this project, the goal was to develop a pipeline to create least cost paths that could be generated iteratively using a range of model weights. Not only did the project rely on suitability modeling, but it also relied extensively on suing some of the ETL operations that were created in part 1 of the lab, specifically related to LiDAR processing. The pipeline that was developed can be generally applied to a variety of different scenarios other than the example that it was tested on, with potential implications for studying animal migration patterns, human mobility, and more.

Problem Statement
The problem can be broken down into three main components. The first is that we need to create some sort of dataset that will show how expensive it is to travel in a certain area, which is known as creating a cost surface. The second component is that we need to determine which route is best, based upon an analysis of the cost surface that was developed. Lastly, since the weights that we give different parameters in the second component are actually ambiguous, we need to develop a way that we can test a wide range of weights to see how they impact the mode outputs. Together, this creates a comprehensive pipeline for modeling routing suitability and determining the best path to take in a given scenario.

Table 1. Analysis Requirements for Part 2

#	Requirement	Defined As	(Spatial) Data	Attribute Data	Dataset	Preparation
1	LiDAR Data	Raw .LAZ/.LAS files	Point Cloud	N/A	MN DNR	Convert to DEM/Slope + Reclassify
2	Landcover Data	15 meter landcover dataset for Minnesota from 2013	Raster	LC Class	Minnesota Landcover	Reclassify

Input Data
The first dataset used was LiDAR data that was identical to the data used in the first part of this lab. It comes from the Minnesota DNR and the same exact ETL steps were used on the dataset. The second dataset used was landcover data, which was retrieved from the Minnesota Geospatial Commons. ETL for this dataset was fairly limited and not too complicated. The processes of clipping the datasets to an AOI and reclassification will be discussed in greater detail in the methods section.

Table 2. Input Datasets for Part 2

#	Title	Purpose in Analysis	Link to Source
1	MN DNR LiDAR	Raw input LiDAR point clouds	MN DNR
2	Landcover Data	15 meter landcover dataset of MN from 2013	Minnesota Landcover

Methods

First, I started by performing the LiDAR ETL steps that I also performed in the first part of the lab, so I will not discuss those in great detail since it is identical to that of the first part. I did convert the output DEM into a slope dataset, though, which was used as one of the two inputs into the model. The other input into the model was landcover, which together with the slope, would address all of the parameter requirements for the model. The two datasets were first clipped to an AOI dataset to cut down on run times and eliminate unnecessary data. After this, the data was then reclassified between 1 and 5, with 1 representing unfavorable conditions and 5 representing ideal conditions. Slope was simple to model with higher slopes being more unfavorable and lower slopes being more favorable. With landcover, there were to main concerns that needed to be met, which were avoiding fields (given a score of 1) and tending to avoid crossing water (given a score of 2). The complete breakdown of landcover relassifications can be seen below in table 3.

Table 3. Landcover Relassification.

#	Landcover Class	Score (1 = Worst, 5 = Best)
1-100	% Impervious	5
101	Emergent Wetlands	2
102	Forested Wetlands	2
103	Open Water	2
104	Extraction	3
105	Conifer Forest	2
106	Deciduous Forest	3
107	Mixed Forest	3
108	Managed Grassland	4
109	Hay/Pasture	1
110	Row Crops	1

After reclassification, a model was then created to iteratively use a range of weights to create a cost surface, which in turn could be used to find the least cost path for each cost surface. The algorithm used for iteration can be seen in greater detail below, in code 1. Essentially, it works by looping through a range of decimals, calculating the two weights based upon the decimal, and then using those weights to create the surface, which is then subsequently used to calculate the least cost path.

Code 1. Iterating over Model Weights to Create a Least Cost Path.

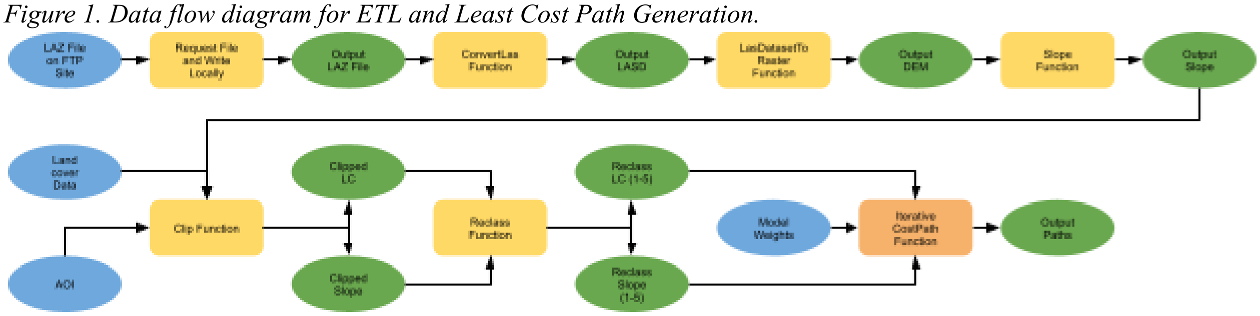
$$for\ i\ in\ range(0.1,\ 1.0,\ 0.1):$$
$$\quad \#\ Setting\ Model\ Weights$$
$$\quad W_{slope} = i$$
$$\quad W_{landcover} = 1\ -\ i$$

$$\quad \#\ Calculating\ Cost$$
$$\quad cost\ = (((slope\ \cdot\ W_{slope})\ +\ (landcover\ \cdot\ W_{landcover}))\ \cdot\ (-\ 1))\ +\ 6$$

$$\quad \#\ Calculating\ Least\ Cost\ Path$$
$$\quad cost\ path\ =\ optimal\ path(origin,\ destination,\ cost)$$

$$\quad return\ cost\ path$$

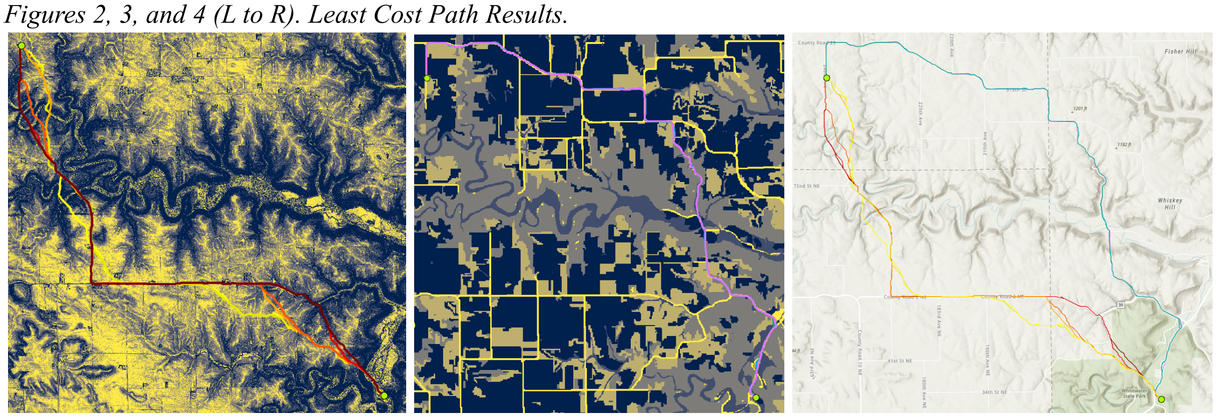
Figure 1. Data flow diagram for ETL and Least Cost Path Generation.



Results

In the end, I was able to produce ten different least cost paths, based on the ten different sets of weights I passed into the model (each parameter had a weight between 0.1 and 0.9, stepping by 0.1, while the other parameter would have a weight equal to 1.0 - i). The paths followed two rough patterns. Figure 2 shows the paths that had a high slope weight, which included 7 paths that followed this pattern. Figure 3 shows the paths that has a high landcover weight, which was made up of the other three weight combinations. Figure 4 shows all of the routes together.

Figures 2, 3, and 4 (L to R). Least Cost Path Results.



Results Verification
In order to verify the accuracy of my operations, I performed manual inspections of the output datasets. They do follow a logical pattern, so it seems like they are good results. Additionally, I was inspecting the data throughout the analysis to ensure that it looked correct after every operation that was performed.

Discussion and Conclusion
This analysis was a very interesting one to work on because it felt like the first one where I was building a more complete analytical pipeline that integrates everything from the ETL, to the model building, and creating results. I think that it was very beneficial to continue to expand upon all of the work that we have been doing and add in the analytical component.

References
N/A

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	**28**
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	**23**
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	**19**
		100	**98**