Longest Flow Path Tool

Stefen Gray

Application Development GISC4500K

Huidae Cho

University of North Georgia

Abstract: Finding the longest flow path from a known outlet, allows you to see the longest path uphill from which a material or fluid might flow from. Though I could not find any specific journal articles directly related to this, there are many practical uses for this information. For example, one could find information on where a possible pollutant may have come from. There are multiple ways to find the longest flow path. One possibility is a largely raster based method which takes a longer time, and requires much more work to receive a vector product. The possibility outlined in this paper is a method which requires only a D8 direction raster and desired outlet points. The method outputs a polyline of the longest flow path for each outlet, but does not return a raster of any kind. Ultimately, I plan to create a comparison of just how the two methods differ performance wise, however, this document merely depicts development of the D8 direction raster method.

Keywords: “Hydrological model”, ”Watershed”, ”Longest Flow Path”

1. Introduction

Formetta, et al. (2015) discuss broadly about the importance of sustainable use of water resources, as well as the importance and methods for studying hydrological processes in watersheds. In their paper they define watersheds as being "comprised of land areas and channels and may have lakes, ponds or other water bodies." (Formetta, et al. 2015) They find that GIS is a "suitable tool for the efficient management of large and complex database and to provide a digital representation of watershed characteristics used in hydrologic modeling." (Formetta, et al. 2015)

Another study, by Singh, et al. (2014) discusses the importance of DEMs and satellite images in the "assessment of drainage and extraction of their relative parameters for the Orr watershed Ashok Nagar district, M.P., India." (Singh, et al. 2014) Watershed management of the area utilized the hydrological model of ArcGIS. Specifically, they used the Shuttle Radar Topography Mission's DEM to delineate the watershed and do "morphometric analysis of the watershed" (Singh, et al. 2014)

With GIS Bruins, et al. (2019) were able to calculate the amount of water that ancient terraced fields we're able to collect. With GIS and incorporated D-Infinity flow algorithms, allowing them to estimate flow direction in degrees instead of cardinal directions, they created a hydrological model to "evaluate annual runoff yields in 13 ancient terraced fields at the rural archaeological site of Horvat Haluqim." Further analysis allowed them to calculate the runoff coefficient, which essentially summarized the efficiency of each terrace as well, the highest being 25.4% and the lowest being 2.8%

Bhatt, et al. (2014) Discuss the development of their hydrologic modeling framework: PIHMgis. As some of the advantages of the framework they list "ease in prototyping of a physics based, fully coupled, distributed hydrologic model simulation using national geospatial data products" (Bhatt, et al. 2014) As for testing their model, they undergo a case study of the Mahantango Creek Watershed in Pennsylvania, USA.

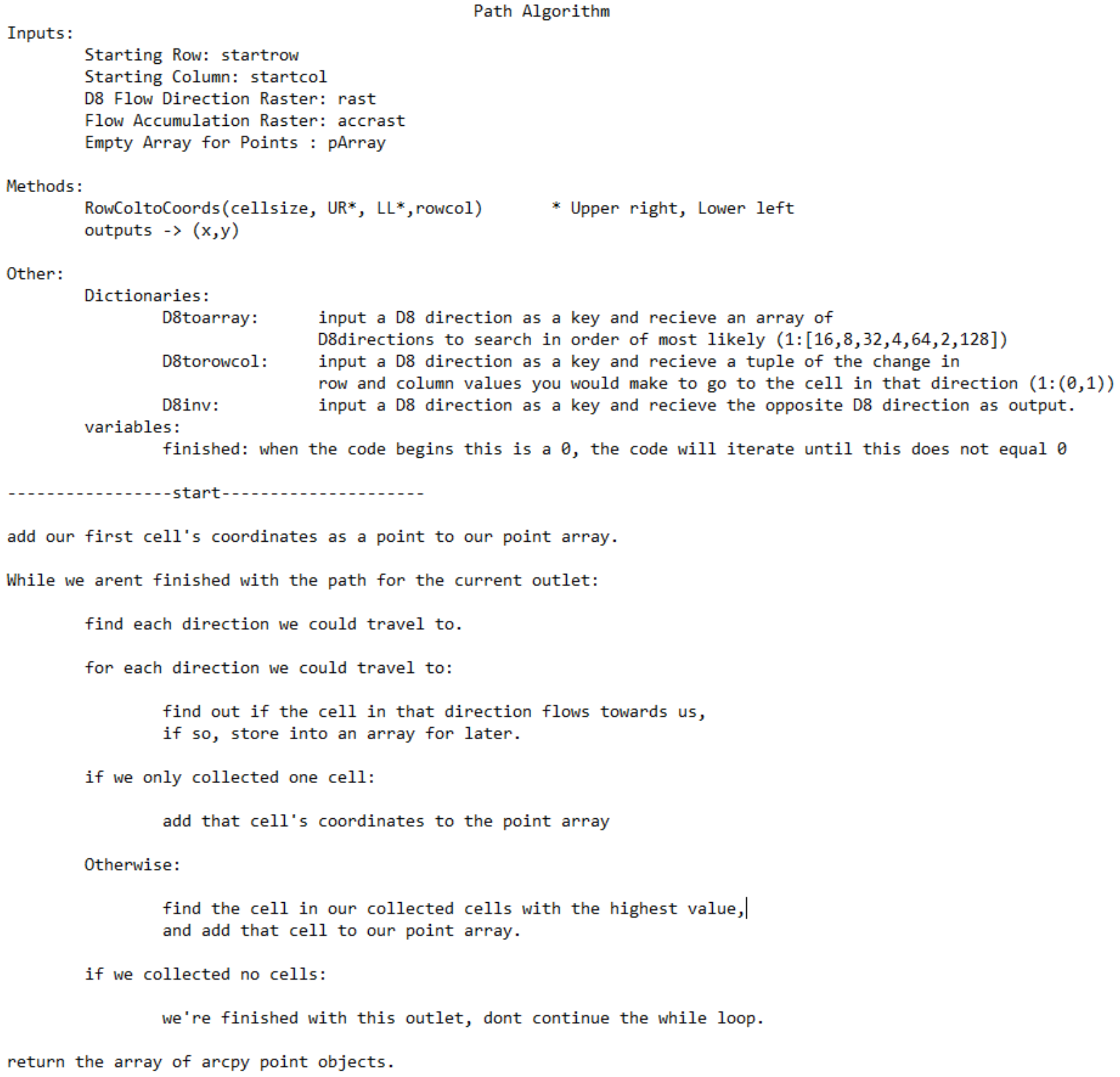
Formetta, et al. (2014) describe "the structure of JGrass-NewAge: a system for hydrological forecasting and modelling of water resources at the basin scale." (Formetta, et al. 2014) as an open-source deployment of a "reproducible-research system" based on three tools/frameworks: uDIG GIS, OMS3, and uDIG Spatial Toolbox, which is what links the previous two together. Ultimately this tool is not useful for this project, however, I will be delving into this tool to make sure I recognize when it may be usable in furture projects.

Hydrological modelling encompasses and vast amount of complex mathematical models. GIS allows for this processed to be modelled using geospatial data along with theoretical mathematical models. This makes GIS ideal for the creation of specific, complicated, or broad hydrological modelling with time, accuracy, and efficiency in mind.

My research objective during this project was to create a python toolbox with a tool that could take D8 direction raster, with a feature class of points representing outlets, and to be able to calculate the longest flow path from that outlet upstream. This is a far simpler task downstream, as you simply have to follow the D8 Flow Raster.

2. Materials and Methods

Shown in figure 2, From a list of outlets, it is possible to find a starting D8 raster cell for each. Following our Pseudocode in figure 1, from the outlet cell, we find the values of the 7 surrounding cells not including the one we came from, and look for any that flow into the cell. (The cell to the west, we would look for a value of 1) We can continue this up until a certain point, if two cells are flowing into the center cell this creates a problem, and we have to split into two lines. We know that the flow accumulation will be highest in the path with most accumulated cells, and thus the longest path, so we continue down the array surrounding 7 cells and find the one with the highest value.



```
                                        Path Algorithm
Inputs:
        Starting Row: startrow
        Starting Column: startcol
        D8 Flow Direction Raster: rast
        Flow Accumulation Raster: accrast
        Empty Array for Points : pArray

Methods:
        RowColtoCoords(cellsize, UR*, LL*,rowcol)       * Upper right, Lower left
        outputs -> (x,y)

Other:
        Dictionaries:
                D8toarray:      input a D8 direction as a key and recieve an array of
                                D8directions to search in order of most likely (1:[16,8,32,4,64,2,128])
                D8torowcol:     input a D8 direction as a key and recieve a tuple of the change in
                                row and column values you would make to go to the cell in that direction (1:(0,1))
                D8inv:          input a D8 direction as a key and recieve the opposite D8 direction as output.
        variables:
                finished: when the code begins this is a 0, the code will iterate until this does not equal 0

-----------------start---------------------

add our first cell's coordinates as a point to our point array.

While we arent finished with the path for the current outlet:

        find each direction we could travel to.

        for each direction we could travel to:

                find out if the cell in that direction flows towards us,
                if so, store into an array for later.

        if we only collected one cell:

                add that cell's coordinates to the point array

        Otherwise:

                find the cell in our collected cells with the highest value,
                and add that cell to our point array.

        if we collected no cells:

                we're finished with this outlet, dont continue the while loop.

return the array of arcpy point objects.
```

Figure 1: Pseudocode of the Path Algorithm used for finding the longest flow path in the python toolbox

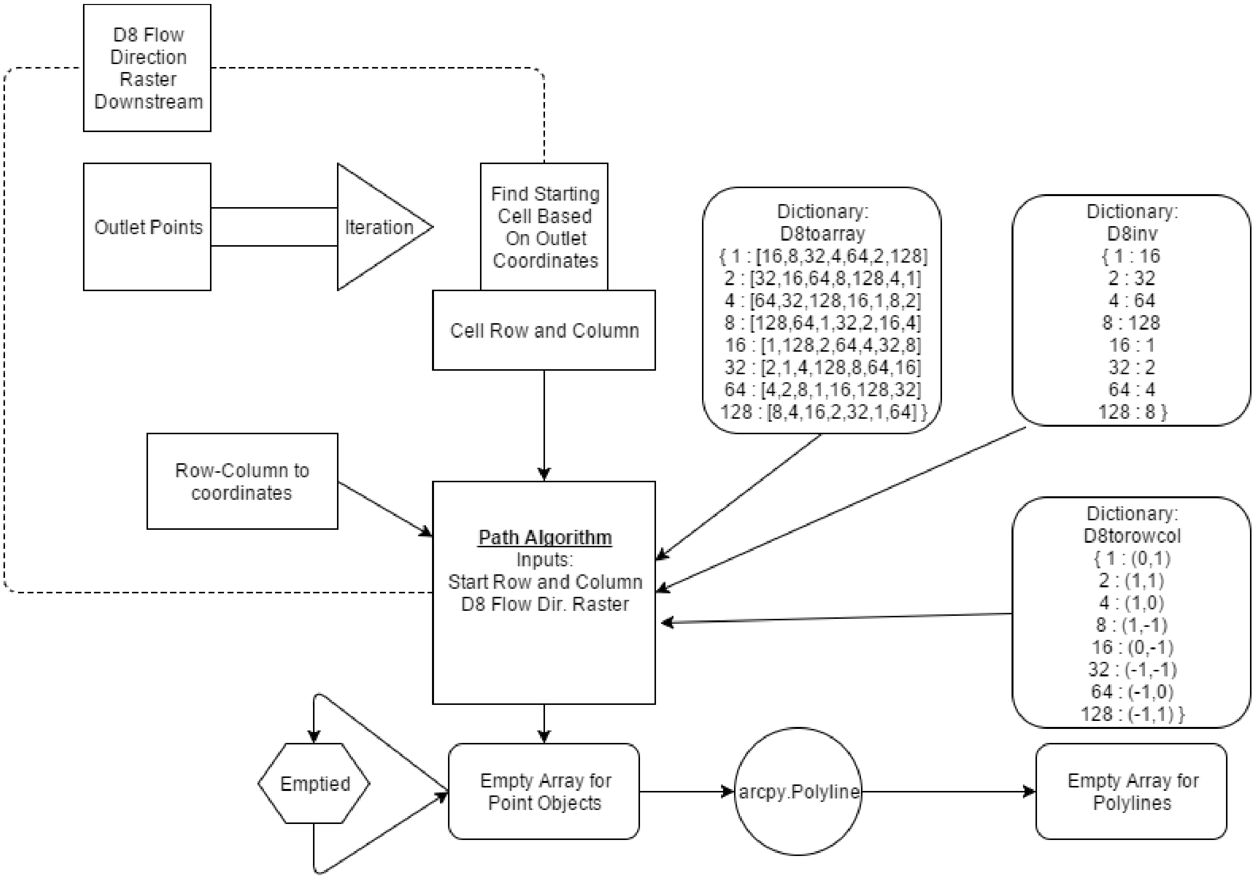


Figure 2: a graphical representation of the process of the code. most time is spend running through the Path Algorithm for each iteration of the Outlet Points.

3. Results and Discussion

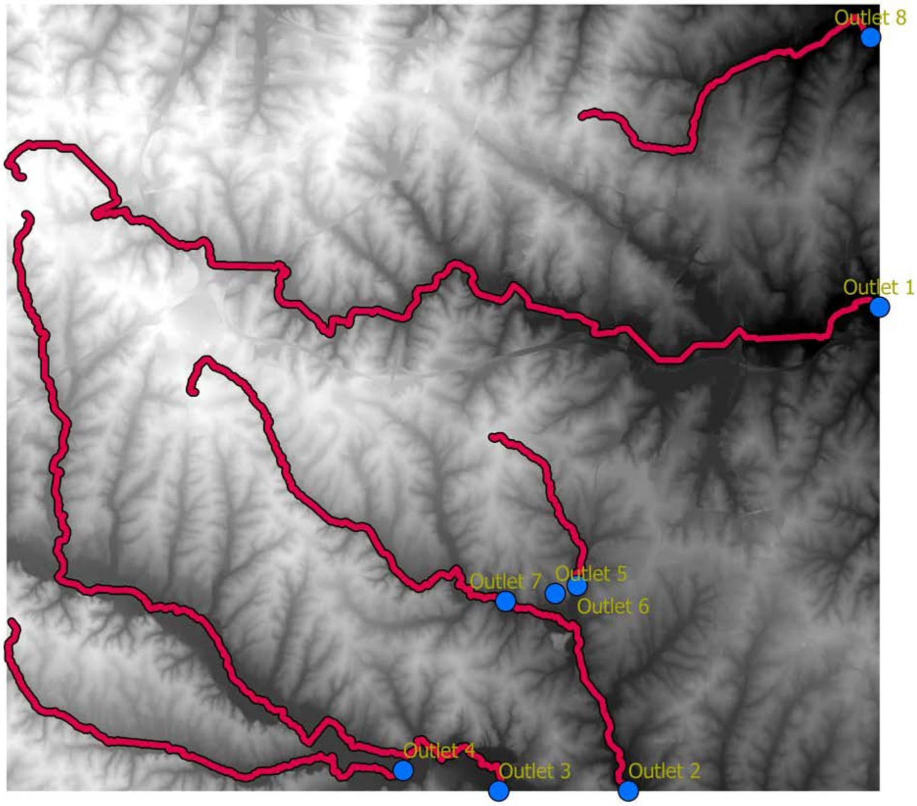


Figure 3: Output from out toolbox with 8 outlets entered, shown atop the DEM of the area.

As seen above, the code for the most part works. There are some “issues” with outlets not placed on major streams providing short 10-60 meter long flowlengths, this could be a desired outcome, however I may want to look further into connecting this paths with longer paths they are near but not quiet connecting. Future analysis would preferably by on a larger DEM with randomly placed outlets rather than manually placed outlets, for the sake of testing the robustness of the code.

For possible future expansions, I would like to provide comparisons to other LFP algorithms, similar to the research Dr. Cho plans on publishing.

4. Conclusion

Ultimately the only conclusion I can make on the project as it is now, is that it worked as intended. Future analysis is planned on calculating average time of production for LFP paths per point stored/cell processed. Future work will include 1 or more LFP algorithms to compare, as well as calculated processing times for each outlet, and reduced to seconds per point along the longest flow path.

References

1. G. Formetta, A. Antonello, S. Franceschi, O. David, R. Rigon, “Hydrological modelling with components: A GIS-based open-source framework”, Environmental Modelling & Software, Volume 55, 2014, Pages 190-200, ISSN 1364-8152, https://doi.org/10.1016/j.envsoft.2014.01.019. (http://www.sciencedirect.com/science/article/pii/S1364815214000292)

2. Prafull Singh, Ankit Gupta, Madhulika Singh, “Hydrological inferences from watershed analysis for water resource management using remote sensing and GIS techniques”, The Egyptian Journal of Remote Sensing and Space Science, Volume 17, Issue 2, 2014, Pages 111-121, ISSN 1110-9823, https://doi.org/10.1016/j.ejrs.2014.09.003. (http://www.sciencedirect.com/science/article/pii/S1110982314000271)

3. Hendrik J. Bruins, Hodaya Bithan-Guedj, Tal Svoray,“GIS-based hydrological modelling to assess runoff yields in ancient-agricultural terraced wadi fields (central Negev desert)”, Journal of Arid Environments, 2019, ISSN 0140-1963, https://doi.org/10.1016/j.jaridenv.2019.02.010. (http://www.sciencedirect.com/science/article/pii/S0140196319300266)

4. Gopal Bhatt, Mukesh Kumar, Christopher J. Duffy, “A tightly coupled GIS and distributed hydrologic modeling framework”, Environmental Modelling & Software, Volume 62, 2014, Pages 70-84, ISSN 1364-8152, https://doi.org/10.1016/j.envsoft.2014.08.003. (http://www.sciencedirect.com/science/article/pii/S1364815214002266)

5. G. Formetta, A. Antonello, S. Franceschi, O. David, R. Rigon, “Hydrological modelling with components: A GIS-based open-source framework”, Environmental Modelling & Software, Volume 55, 2014, Pages 190-200, ISSN 1364-8152, https://doi.org/10.1016/j.envsoft.2014.01.019. (http://www.sciencedirect.com/science/article/pii/S1364815214000292)