Flow Routing on Flat Terrains

Taylor Hamilton
Department of Computer Science
Swarthmore College
thamilt1@swarthmore.edu

Giovanna Thron
Department of Computer Science
Swarthmore College
gthron1@swarthmore.edu

Abstract

Most current flow routing algorithms use digital elevation models (DEMs) to construct flow models. In order to successfully use current techniques for flow routing, they flood local minima and then find a way of routing the flow across the flat surfaces. In this paper, we examine an alternative method for computing flow routing on these flooded surfaces that takes into account the original elevation data. Our approach is based upon Dijkstra's single source shortest path algorithm. We set the distance between two adjacent cells to be the elevation of one of the cells. While our results are not yet ideal, altering our distance formula shows promise for improvement.

1 Introduction

A current problem in geographic information systems is the automatic extraction of river networks from a set of elevation data using the method of flow accumulation. Calculating river networks is useful in determining flood insurance zones.

The basic idea for solving this problem is relatively simple: for each elevation point, route flow to the neighbor with the steepest downhill slope. This method is effective as long as there are no local minima. Local minima will be pits or valleys in which the water will get trapped. Ideally, all water should flow to some outlet point at the edge of the grid.

Current algorithms (Jenson and Domingue, 1988; Garbrecht and Martz, 1997; Soille and Colombo, 2003) tend to deal with this problem by flooding the minima until they are all removed. This approach is justified by the assumption that minima are the accidental result of poor sampling in the original data. However, this is not always the case. Many minima are caused by large-scale terrain features, such as a bridge over a river. When these minima get flooded, useful information about the underlying river is lost, as can be seen in Figure 1.

These flooding algorithms create large flat surfaces, which cause a new problem in flow routing. Without a steepest downslope neighbor it is not immediately obvious in which direction the water should flow across the surface.

Several algorithms have been developed that attempt to solve the problem of flow routing over flat terrain. A side effect common to all these algorithms is that they fail to use information about the original terrain with their flat terrain flow routing algorithms. We have developed an algorithm to route flow across flat surfaces that takes into account the original terrain. This provides river networks that more accurately match reality.

2 Related Work

Current algorithms for solving this problem do not produce ideal results. Jenson and Domingue (1988) focussed mainly on flooding and their method for flow routing on flat surfaces was not very involved. For each point on a flat surface, they assigned the flow to be in the direction directly towards the outlet. This resulted in artificial looking river networks

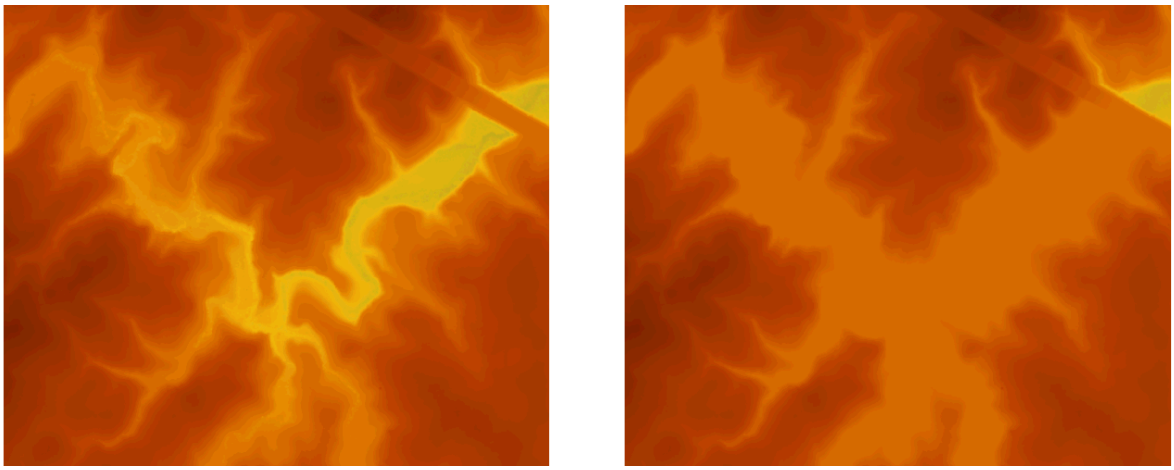

Figure 1: Original and flooded elevation data

because of long stretches of parallel lines.

Garbrecht and Martz (1997) improved upon this algorithm by not only routing flow towards the outlet, but also away from the bordering high terrain. This provided more natural looking river networks. However, as these river networks do not take into account the underlying elevations, they do not always accurately model the true flow of water in the region.

Soille et al. (2003) proposed the method of carving as opposed to flooding for removing minima, which reduces the number of flat areas. They also proposed a flat terrain flow routing algorithm that is an improvement on Garbrecht and Martz's method. Although Soille's algorithm is an improvement, it has the same fundamental issues as that of Garbrecht and Martz.

3 Methods

Our algorithm focuses on improving flow routing over flat terrains. To accomplish this, we use both the flooded terrain information and the original elevation data. The flooded terrain indicates the areas on which to concentrate, and the original terrain provides the elevation data needed for our method. We use Dijkstra's algorithm to calculate the shortest path to the spill points. We vary the metric for computing the distance between two adjacent cells.

We begin with digital elevation models in the GRASS ASCII format: one of the original terrain and one with the local minima flooded. We find the spill points, cells adjacent to the flooded terrain with lower elevations.

We used a single source shortest path algorithm to calculate flow directions for flat areas. This algorithm treats our grid as a connected graph where each cell is connected to its eight neighbors. The weights of the edges can be chosen independently of the algorithm, and we experiment with several options.

To compute the single source shortest path, we used Dijkstra's algorithm. The algorithm begins by initializing all the path lengths of any cell to the spill point to infinity. We create a priority queue that contains the spill points, and set their path lengths to zero. We continue extracting the point from the priority queue with the minimum path length until the queue is empty. Each time we remove a point, we look at each of its neighbors and update their paths if the path through the current point is shorter than the stored path. We then add each updated neighbor to the priority queue.

When the algorithm is finished, the result is a forest that spans the area of interest, where each tree is rooted at a spill point. The leaves of the trees are the points farthest away from the spill points. The path from a node to the root of a tree is the shortest path to a spill point.

We can use these trees to calculate flow accumulation. We imagine that a unit of water falls onto each cell in the grid. Using the flow directions of each cell, we can determine the amount of water that accumulates in each cell. Let p be any cell and $F(p)$

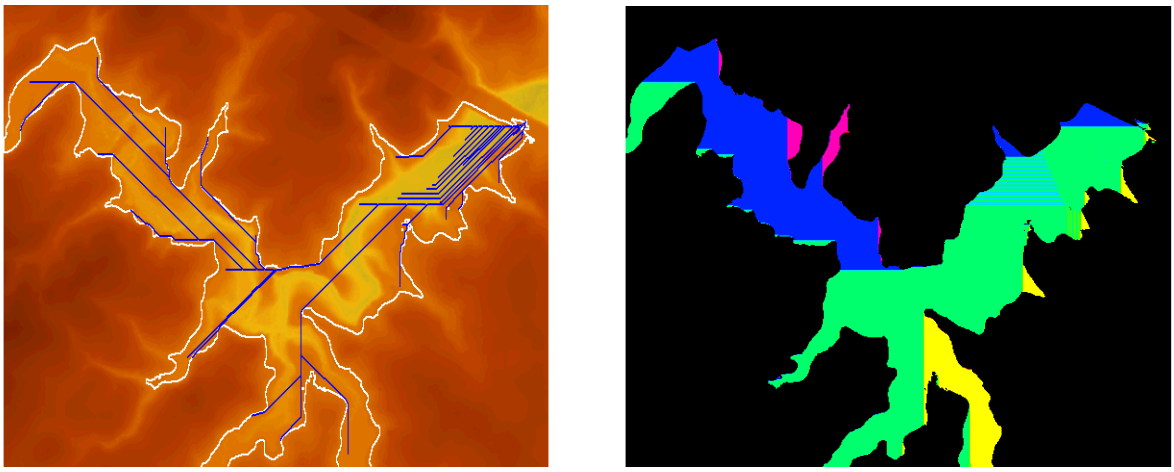

Figure 2: River networks and flow directions using the Euclidean distance metric

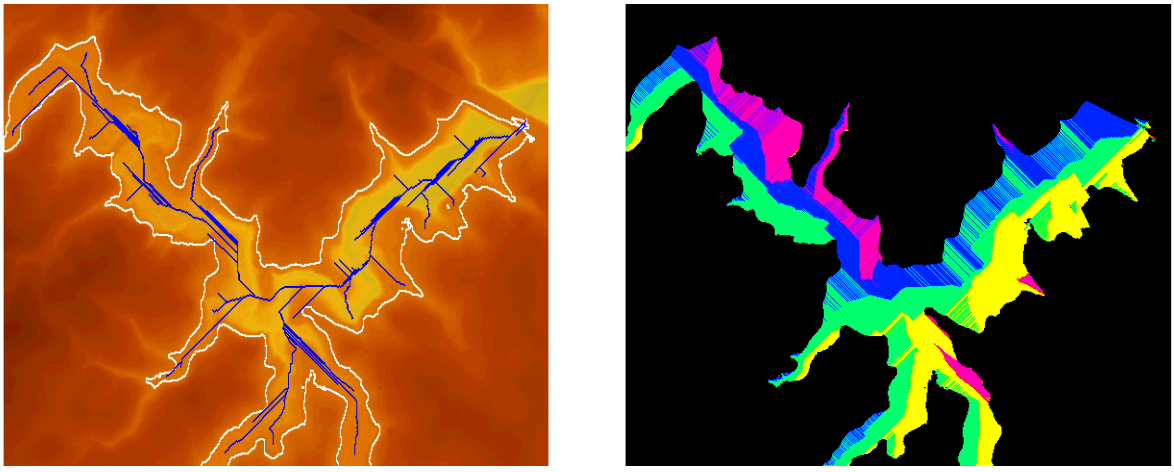

Figure 3: River networks and flow directions using Soille's algorithm

be the set of cells flowing into p.

$$\text{acc}(p) = 1 + \sum_{q \in F(p)} \text{acc}(q)$$

We calculate the flow accumulations of a node in the forest as the sum of the flows of each of its children plus one. A grid showing cells whose flow accumulations are greater than some threshold should show the locations of the rivers of the terrain.

Our algorithm outputs GRASS ASCII files with the flow directions and the flow accumulations at each cell of the grid. We represent flow direction with numbers 1 through 8, corresponding to the eight possible directions of flow. We create river networks based on the set of points with flow accumulations over a given threshold.

3.1 Metrics

Below we present the weights used to determine the distances between cells for the Dijkstra's algorithm. In all cases, we added an extra weight of $\sqrt{2}$ to diagonally adjacent cells to account for the difference in Euclidean distance.

3.1.1 Euclidean Distance

The simplest method we used was setting the distances between adjacent cells to 1. This resulted in the Single Source Shortest Path (SSSP) method, as used by Jenson and Domingue (1988). In effect this method just computes the shortest Euclidean distance from any point to the spill points, and routes flow over that path.

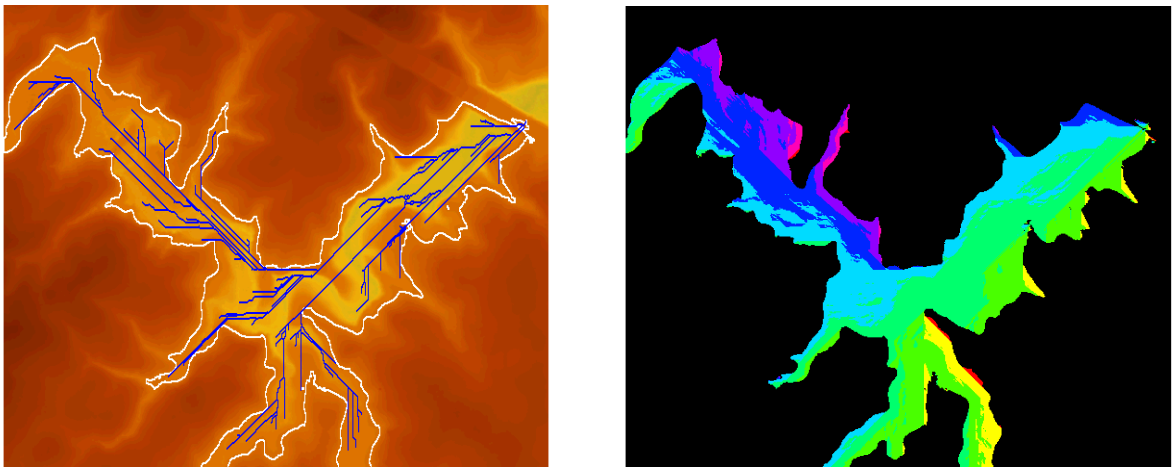

Figure 4: River networks and flow directions using the elevation distance metric

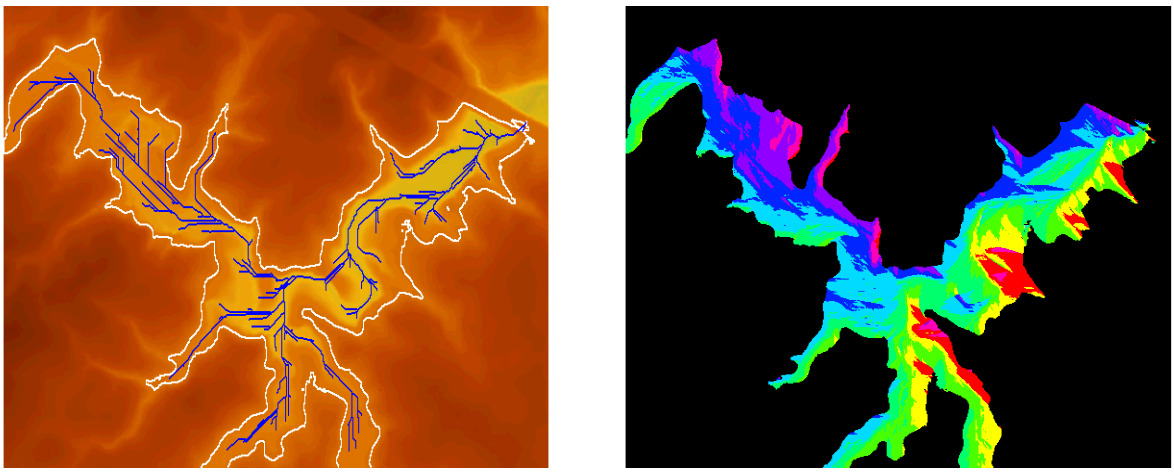

Figure 5: River networks and flow directions using the translated elevation distance metric

3.1.2 Soille's Algorithm

The flat terrain flow routing algorithm introduced by Soille (2003) is designed to route flow through the center of the terrain and avoid having straight parallel lines. We calculate the distance, $d(c)$ from each cell to the border of the flat terrain using a breadth first search away from the border. Let c be a cell and C the set of all cells.

$$w(c) = \max\{d(f)|f \in C\} + 1 - d(c)$$

3.1.3 Elevation

Our first metric that uses the original elevation data sets the distance between any two adjacent cells to the elevation of the cell flowed to. Thus, the total distance from a cell to the spill point is the sum of the elevations of the cells traversed. This encourages the flow to travel down to lower elevations, as well as traversing a small number of cells between the source and the spill points.

3.1.4 Translated Elevation

To weight more heavily the importance of flowing across low elevations, as opposed to traversing short distances, we translate the elevations of all cells down by the minimum elevation over the relevant area. Thus, the weight of a cell is the difference of the elevation of the cell and the minimum elevation.

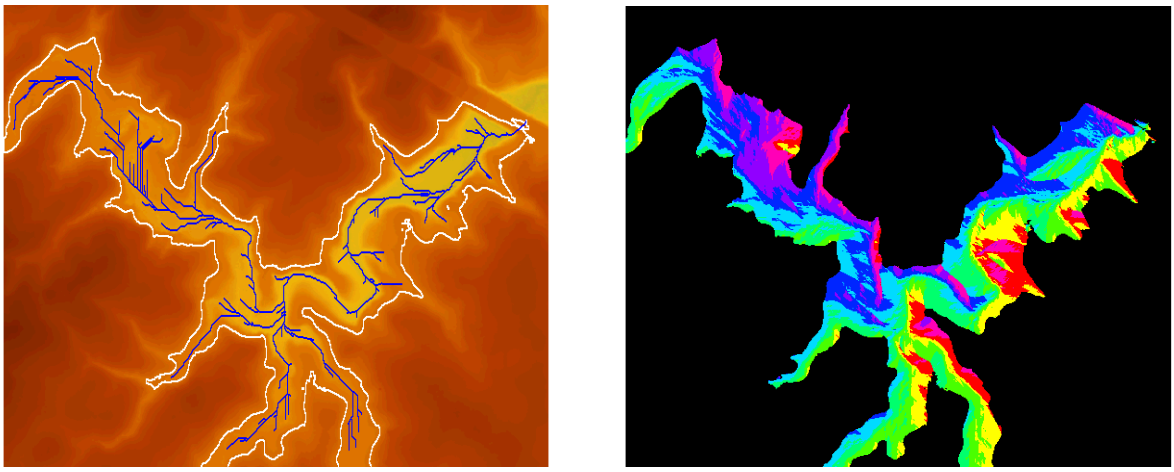

Figure 6: River networks and flow directions using the squared translated elevation distance metric

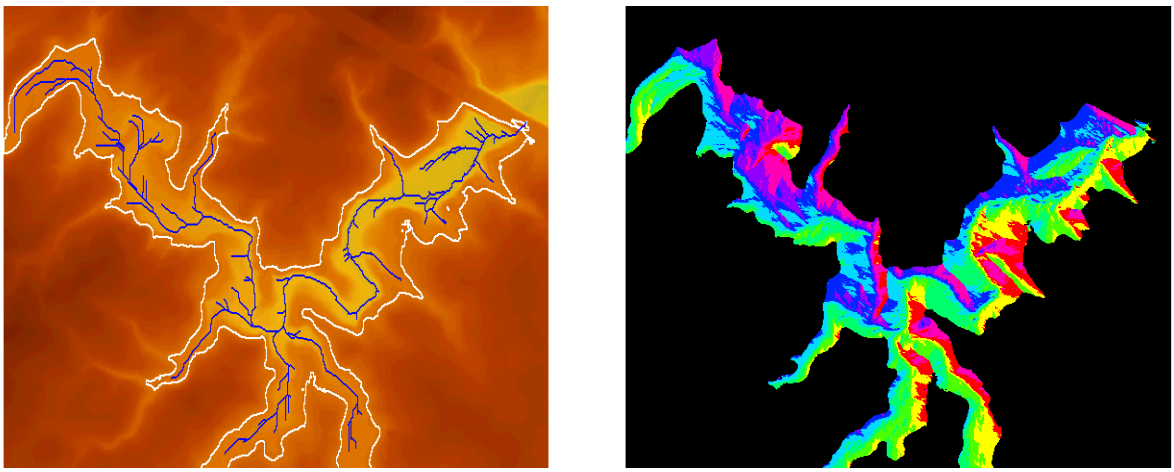

Figure 7: River networks and flow directions using the fourth power of the translated elevation distance metric

3.1.5 Power of Translated Elevation

Raising the translated elevation to a positive power puts a greater penalty on higher elevations. This further encourages flow to follow low elevations. A greater power will put more emphasis on traveling on low elevations.

4 Data

We used raster data of the North Carolina river basin at 10 foot resolution. We used both the original elevation data and elevation data of the terrain with the sinks flooded.

5 Results

For each metric, we show the river networks and flow directions. In computing the river networks, we used an accumulation threshold of 150 cells (150,000 ft^2). In the flow directions figures, each color indicates a different flow direction.

The results of using the Euclidean distance metric are shown in Figure 2, Soille's algorithm in Figure 3, the elevation metric in Figure 4, the translated elevation metric in Figure 5, the squared translated elevation metric in Figure 6, and the fourth power of the translated elevation metric in Figure 7.

6 Discussion

As can be seen in Figures 2 through 7, the results improve with each alteration of our algorithm, eventually producing natural looking river networks that follow the elevations of the original terrain.

By comparison, the Euclidean metric (Figure 2 fails to produce natural looking or accurate rivers. The flow is routed in straight parallel lines and hugs the boundaries of the region, as the algorithm searches for the shortest Euclidean distance.

Soille's algorithm (Figure 3), on the other hand, produces more natural looking river networks. However, as this algorithm fails to take into account the original elevation data, the rivers do not follow the terrain features. As an example of this behavior, observe the oxbow near the center of the region. Rather than following the bend in the river, the river stays in the center of the region.

Each of Figures 5 through 7 shows an improvement on the previous river network. As you can see in Figure 7, our river network both looks natural and accurately models the terrain. Our river network tends to stay in the lighter yellow areas, which corresponds to the lowest elevations in the region.

From the river networks in Figure 7 it can be seen that our algorithm tends to perform better on lower elevations than on higher ones. This is a result of translating the elevations by the minimum elevation of the region. While this translation succeeds in appropriately weighting elevation against Euclidean distance at lower elevations, this balance too heavily in favor of Euclidean distance at higher elevations.

7 Future Work

We would like to develop a distance metric that does not have the drawbacks at higher elevations that our current algorithm displays. To accomplish this, we have experimented with other distance metrics without success. We began by taking the exponential of the elevation, but discovered that this this resulted in numbers that overflowed Python's float type. We would like to experiment with the taking the differences of elevations of neighboring cells.

References

J. Garbrecht and L. Martz. 1997. The assignment of drainage direction over flat surfaces in raster digital elevation models. *Journal of Hydrology*, 193:204–213.

S. Jenson and J. Domingue. 1988. Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric Engineering and Remote Sensing*, 54:1593–1600.

J. Vogt Soille, P. and R. Colombo. 2003. Carving and adaptive drainage enforcement of grid digital elevation models. *Water Resources Research*, 39:10–1–10–13.