Designing an Optimal Pit Removal Tool for Digital Elevation Models

GIS in Water Resources, Fall 2012
Stephen Jackson, University of Texas at Austin

Table of Contents

I. Introduction

The goal of the project is to develop a distributable ArcGIS implementation for optimal pit removal for a DEM.

The existing standard tool in ArcGIS for pit removal of a Digital Elevation Model (DEM) is the **Fill** tool within the Spatial Analyst extension. This method is effective, but raises the average elevation of the terrain and creates artificial flat areas. For many applications this may be acceptable, but some precise analyses may be negatively impacted. One situation for which an improved approach to pit filling may be desired is the hydrologic conditioning of high resolution DEMs which pick up roads but not culverts, effectively creating a dam. In such cases the F**ill** procedure would obliterate the upstream river, while a carving procedure could simulate a culvert through the dam.

Two alternate methods for pit removal have been proposed: carving (Soille et al. 2003) and a hybrid approach (Soille 2004). Carving lowers elevation along a path from the pit to an outlet cell, and thus suffers from the reverse problem associated with filling. The hybrid approach is termed optimal as it minimizes the cost, defined as the absolute total elevation change to the DEM, through utilizing both carving and filling. This is achieved by filling the pit area to some elevation, and then carving a path from that elevation to an outlet. As filling acts across a potentially wide two dimensional area and carving acts along a one dimensional path, it is expected that in general the point of minimal total cost would be closer to the pure carving solution. An algorithm to achieve this optimal result is described by Soille, but an implementation is not currently publicly available. This project aims to provide such a tool.

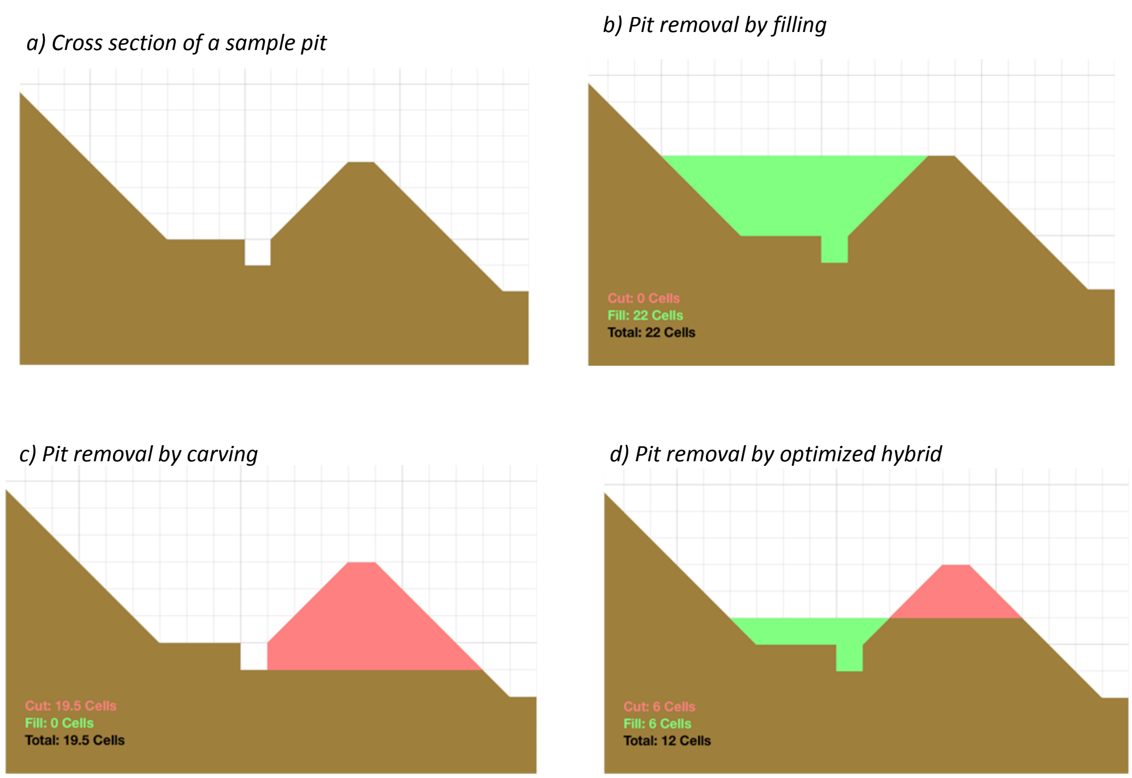


Figure 1: Conceptual depiction of pit removal techniques

II. Methodology

Project Scope

The project was separated into three phases, described below. In designing the final algorithm, the approach described by Soille was used as a guide, but modifications were freely made to address difficulties as they arose.

- ***Phase 1: Develop Synthetic Landscape Creation Program (VB.net)***

The first phase involved building a standalone program that creates an ASCII grid DEM of arbitrary size for which all points drain to the border, which is then seeded with pits in a customizable fashion. The purpose of this is twofold. First, it serves as an exercise to gain experience working with raster terrains programmatically. Second, these outputs would be used as a controllable testing ground for the pit removal tool during development.

- ***Phase 2: Develop Pit Removal Prototype (VB.net)***

This standalone Visual Basic program serves as a proof-of-concept. The purpose of this stage is to gain a clear understanding of the algorithm and its effects using a language which the researcher is most familiar with.

- ***Phase 3: Develop Distributable ArcGIS Pit Removal Tool (Python/C++)***

The final phase involves translating the prototype into more suitable languages and making any improvements required. This serves both to familiarize the researcher with the ArcGIS 10.1 programming environment and to create a useful deliverable.

Tool structure

ArcGIS provides many options for developing custom tools. In previous versions, there was considerable support for integrating Visual Basic and C++ code. With ArcGIS 10.0, the software moved to Python as the preferred language. In deciding how to structure the tool and which languages to use, the TauDEM package provided a good roadmap (Tarboton). These tools use standalone C++ console applications to perform all complex calculations and transformations. As console applications, they are initialized with a single line of text which contains all input and output file locations and any additional required variables. A short Python script serves to translate the ArcGIS user inputs into this required text format, activate a command line, and run the application. Using the functionality built in to ArcGIS, this Python script is loaded as a tool. Within the tool, metadata such as the variable names, variable types, and basic help documentation is added. The completed tool then functions just like any other native ArcGIS tool and can be incorporated into Model Builder workflows. Only three files are required to distribute the final product: the compiled console application executable, the Python script, and the ArcGIS tool containing the metadata.

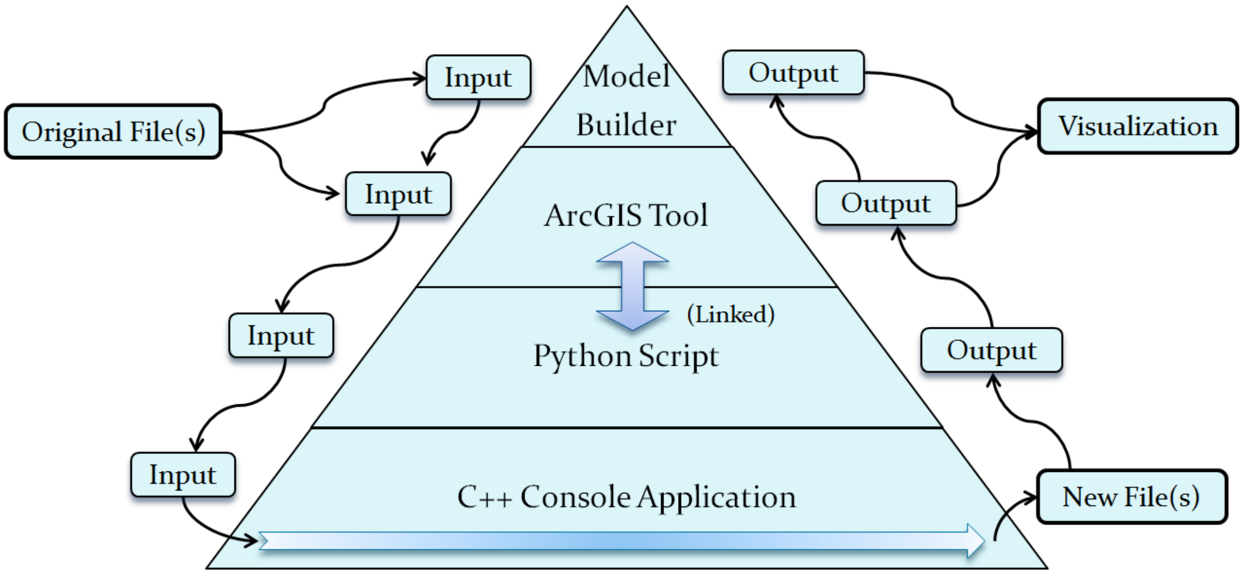


Figure 2: Selected organizational structure

There are two primary benefits to structuring the tool with a separation between the core functionality and the ArcGIS integration. The first is that it allows the main portion of code to be written in whatever language the developer prefers, rather than being tied to Python. For this present tool, C++ was chosen for run-time speed and consistency with the larger TauDEM project. A review of online recommendations suggested that while Python is a simpler language to code in, C++ has notably better performance. As the data sets and iterations required for this tool could get arbitrarily large, performance is a high priority. The second benefit to the separable structure is it greatly increases the durability of the tool. ArcGIS has changed preferred languages several times in the past, and may do so again in the future. It would be highly impractical to completely reprogram all custom tools with each version change. This structure means that only the short and standardized Python interface would need to be adapted. An additional benefit of separating the core function from the interface is it allows for integration with other GIS software, or even for use on computers with no GIS software at all.

III. Results & Discussion

A summary of the development process and current status of each phase is presented below. At present the evaluation of output terrains is primarily visual and qualatative, using the ArcGIS **Sink** tool to identify remaining pits, **Raster Calculator** to display changes in landscape, and spot checking elevations to investigate the programmatic logic.

- ## Phase 1: Synthetic Landscape Generator

The **Synthetic Landscape Generator** creates an ASCII grid with header information ready to be imported to ArcGIS. The algorithm used to generate the terrain involves starting with a random elevation at the corner of a square grid and spiraling inwards. Each cell is assigned a random elevation, subject to a maximum slope constraint and a requirement that at least one neighboring cell has a lower value. This creates a surface with randomized ridges and valleys, but a guaranteed flowpath to the border. Randomized multi-celled modifications are then made to this, either by cut (in a fashion which guarantees a pit) or by filling (which creates a pit only if it blocks flow).

The completed version of the **Synthetic Landscape Generator** produces sample surfaces of user-controlled size and pit density. These proved very valuable in testing the various stages of the pit removal tools. This landscape generator may also be useful during the design of future tools, although the present algorithm is restricted to creating a single mountain centered on the grid.

a) User Interface *a) Sample Landscape*

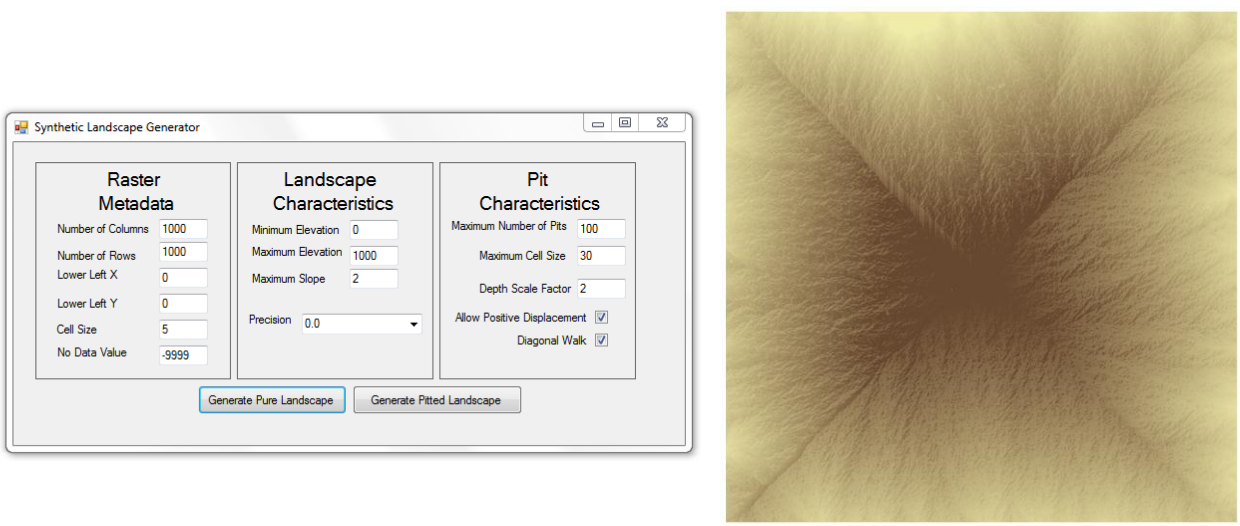


Figure 3: Synthetic Landscape Generator. The characteristic 4-ridged mountain shape is typical of the generated landscapes.

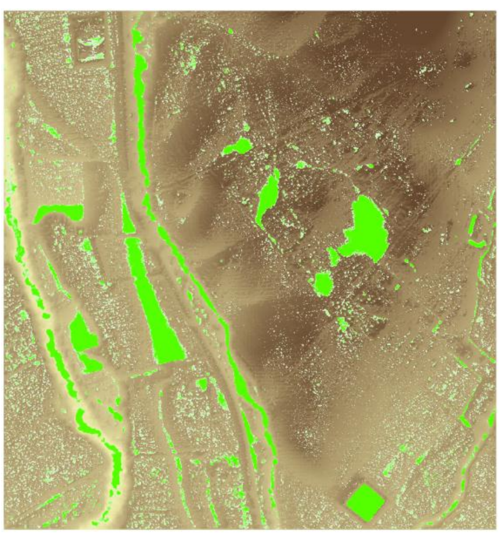
- ## Phase 2: Pit Removal Prototype

The design of the Visual Basic prototype progressed very rapidly. It provided an opportunity to become familiar with both the conceptual approach to pit removal as described by Soille, and some of the programming methods required. The most notable of these is the priority queue, which is a specialized form of lists where each item in the list is given a priority, and items are accessed and removed from the list in order of priority. This was used in the pit removal algorithm to simulate a rising flood, where cells with a lower elevation are affected first. Visual Basic does not contain a native implantation of priority queues, so one was added as a Class.

The completed version of the **Pit Removal Prototype** produced good results on the synthetic landscape which were clearly a hybrid of cut and fill. When applied to an actual LiDAR dataset, the results were dramatic and demonstrated very clearly the lesser overall impact the optimized approach had compared to the fill method. In both the synthetic and the LiDAR results, however, an analysis of the modified terrain revealed some remaining pits. It is uncertain whether these where due to bugs in the program or flaws in the approach. However, as a proof-of-concept, the prototype was very successful, and development was begun on the final phase.

a) Original LiDAR Terrain

*b) Cells modified by ArcGIS **Fill** tool*



*c) Cells modified by **Pit Removal Prototype***

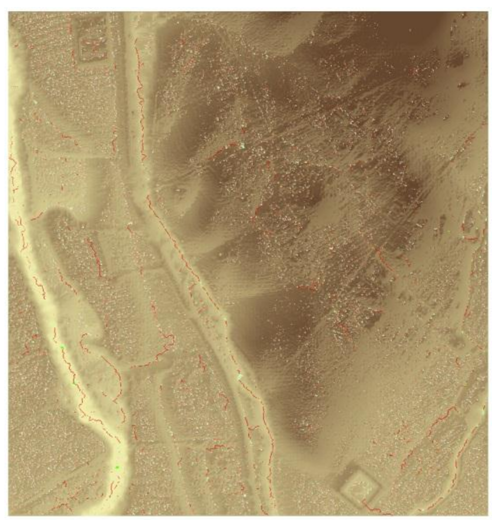


Figure 4: Pit Removal on LiDAR grid. 1079x1025 cells excerpt of Napa Watershed, CA, Data Tile 14 at 1m resolution. Green represents cells with increased elevation; red represents cells with decreased elevation.

- ### Phase 3: Hybrid Pit Removal Tool

The creation of the final tool was the most challenging portion of the project, as it involved coding in two unfamiliar languages. The Python script was based heavily on TauDEM source code, with a small tweak to the command line instantiation. This was due to a difficulty getting the application to run when additional arguments were specified. Aside from this, it was relatively straightforward to prepare the inputs and import the script to ArcGIS as a tool. Progress has begun on the internal help documentation that can be accessed by the user through the ArcGIS tool interface.

a) C++ Console Application Interface *b) ArcGIS Tool Interface*

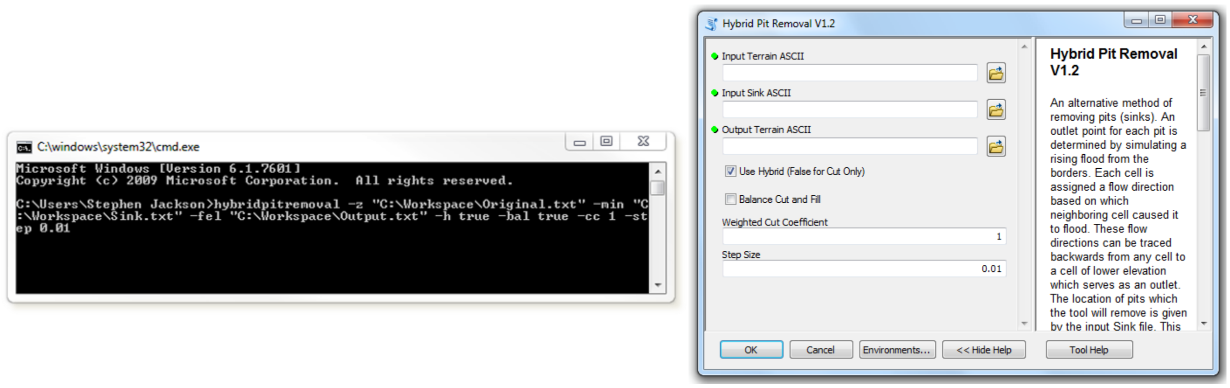


Figure 5: Hybrid Pit Removal Interface

The C++ console application required considerably more work. Two primary challenges were faced: speed and accuracy. The code was tested using a synthetic landscape of 1000x1000 cells with approximately 100 pits, and a LiDAR landscape of 1079x1025 cells with approximately 13,500 pits. Throughout the design, it was found to be more efficient to test the code via console commands rather than through the ArcGIS interface, as many trials did not run to completion and the console provides more immediate feedback.

The first step was translating the algorithm from the Visual Basic prototype into C++ language. When this code was run, it was far too slow to be practical, even for testing. This was in contrast to the high speed that was expected. The primary cause was found to be the way large array variables were passed between functions. Using global variables increased speed considerably, but not sufficiently. As the debugging tools within C++ are not as intuitive as those within Visual Basic, a method was discovered of printing punctuation marks and symbols to the console at the start of each function. This provided immediate feedback for which lines of code were lagging at runtime, and which were requiring excessive iterations. Using these cues, the code was substantially revised to seek higher efficiency. The most substantial change was to calculate and store the cut and fill cost functions at the start of each pit removal using a constant step size. This had the twin benefits of being leaner code and making the **Balance** mode (described below) trivial to implement.

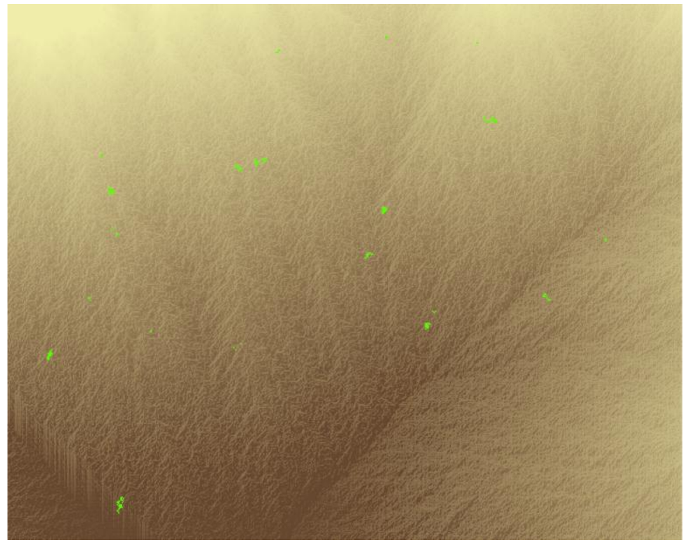
The approach used in the design of this tool allows it to be run in several different modes. In **Cut Only** mode, no optimization is performed and pits are removed simply by carving a one-dimensional path to the outlet. This is computationally efficient, and represents the opposite extreme from the **Fill** tool. In **Balance** mode, the cut and fill cost functions are calculated, and the point of intersection is found. As the functions are determined discretely rather than continuously, intersection is defined as the point of minimum difference. If multiple intersections exist, the present algorithm selects the one closest to the **Cut Only** solution. In **Hybrid** mode, the cut and fill cost functions are calculated and the point of minimum total cost is selected. Total cost is defined below. The Cut Coefficient is a user-defined parameter that is set to 1 by default. This can be adjusted if the user has reason to prefer one end of the spectrum over the other. The result of a suitably high coefficient would mimic the **Fill** tool, while a suitably low coefficient would mimic the **Cut Only** mode. Both the **Balance** and the **Hybrid** modes require Step Size as a user input, which determines the vertical resolution of the discrete cost functions.

$$Total\ Cost = Cut\ Coefficient * abs(Cost\ of\ Cut) + abs(Cost\ of\ Fill)$$

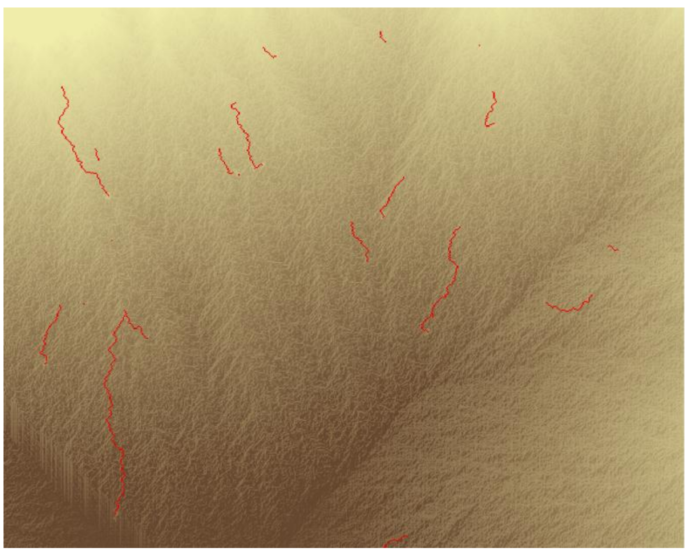
At the time of this writing, the **Hybrid Pit Removal** tool has been successfully tested on the synthetic landscape (see Figure 6), with results comparable to the prototype. High and low Cut Coefficients as well as the **Balance** and **Cut Only modes** have also been tested, with reasonable results. When applied to the LiDAR surface, however, the results leave much to be desired (see Figure 7). Many cells are changed by fractional amounts that ought to be untouched, and there is a much greater degree of filling than is expected or desired. In all cases tested, the finished landscape is not completely free of pits. There are, therefore, at least two bugs remaining in the code. The first affects the general optimization of the results when compared to the highly successful prototype. The second affects providing a path to an outlet in a few fringe cases of undetermined properties.

a) Original landscape expanded view

*b) Cells modified by ArcGIS **Fill** tool*



*c) Cells modified by **Hybrid Pit Removal** in **Cut Only** mode*



*d) Cells modified by **Hybrid Pit Removal** in **Hybrid** mode with Cut Coefficient of 1*

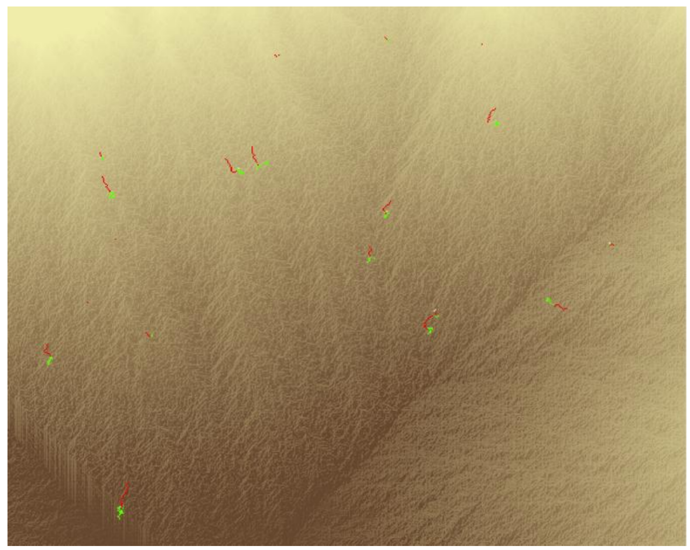


Figure 6: Pit Removal on Synthetic Landscape. Green represents cells with increased elevation; red represents cells with decreased elevation. The Hybrid approach partially fills each pit area resulting in a shorter cut path to an outlet, and a lower overall average elevation change than either extreme method.

a) Original LiDAR Terrain

*b) Cells modified by **Hybrid Pit Removal** tool in **Hybrid** mode with a Cut Coefficient of 1*

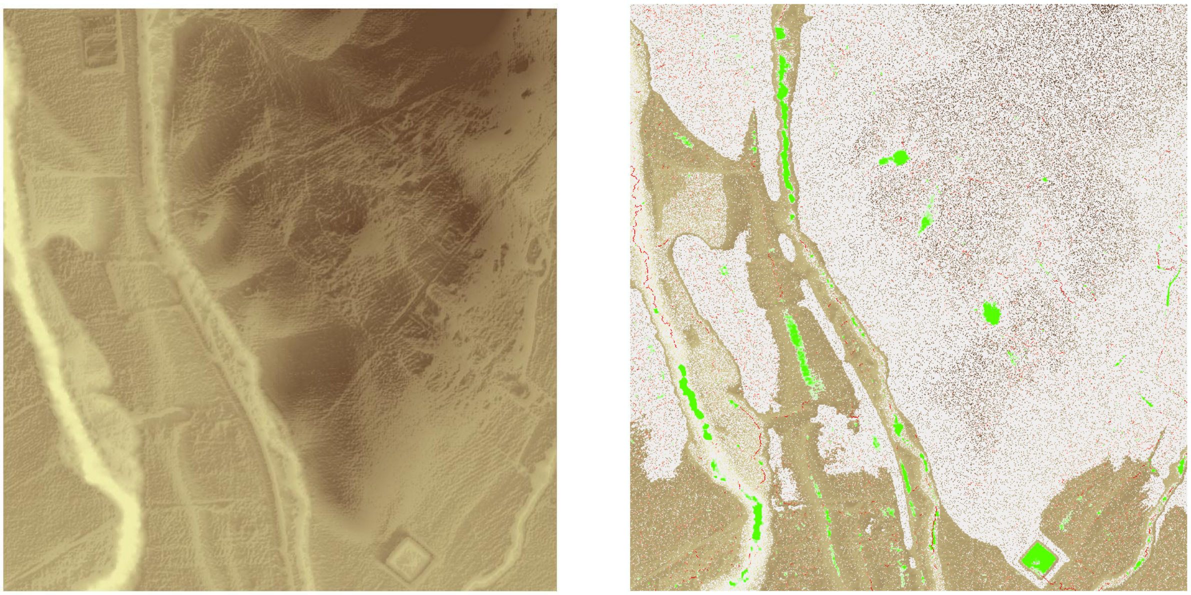


Figure 7: Pit Removal on LiDAR grid. 1079x1025 cells excerpt of Napa Watershed, CA, Data Tile 14 at 1m resolution. Green represents cells with increased elevation; red represents cells with decreased elevation. The washed out areas are cells modified by approximately 0.000001 m, likely due to a floating point error. The remaining modifications do not appear to be as optimal as the solution found by the Prototype as seen in Figure 4 c).

Further Work

The immediate tasks required to complete the current project scope are to debug the remaining issues described above, document the code and the final tool, and to package the result for easy distribution. Once this is completed, more detailed testing of the tool can commence. Two primary avenues of investigation present themselves. The first is to perform statistical comparisons of the results of the new tool in its various modes with results of the existing **Fill** tool. These comparisons should include the number of cells changed, the minimum, maximum and average change, and the total fill and total cut. The second is to compare the results of stream delineation procedures using various LiDAR terrains to determine if this tool provides a ready benefit.

Beyond that, there is room for future improved versions of the tool. These could include the addition of an option to minimize the number of cells modified, the use of various input file types, general enhancements of performance, and the ability to utilize multiple processors.

IV. Conclusions

Development Process

Having nearly completed the final tool, it is clear that the present separable tool structure is highly advantageous. As such, this project has served to provide a template for the development of future tools. In the future, it would be beneficial to learn more programming best practices for efficiency, both in general and for the specific language chosen. There are always numerous ways a task can be performed, and it is not always evident which variable types and loop structures will perform the quickest.

Pit Removal Applications

The practical value of this tool as an alternative to the **Fill** procedure has yet to be seen. Even with additional improvements, the iterations inherent in the method will require considerably more processing time than the standard method, so it does not win on speed. It is possible that this tool will allow for better stream detection in flat areas of highly detailed DEMs, though this remains to be confirmed. In many current workflows, hydraulically conditioned DEMs are not used for anything beyond stream delineation, and thus the tool's property of minimizing elevation changes has no practical value. However, the field of detailed LiDAR analysis is still in its infancy, and it is very possible that new approaches will be developed which will benefit from the range of options provided by this tool. The prospects are particularly bright for the **Balance** option, which would allow the conditioning of a landscape with negligible net change in the statistical elevation distributions. Thus, this tool is offered up not so much to fulfill an existing pressing demand, but rather to better equip researchers to meet an unknown future.

V. References

National Science Foundation. "NCALM Napa Watershed, CA, Data Tile 14." *The National Center for Airborne Laser Mapping*. Web 5 November 2012 <http://calm.geo.berkeley.edu/ncalm/ddc.html>

Planchon, Olivier, and Frédéric Darboux. "A fast, simple and versatile algorithm to fill the depressions of digital elevation models." *Catena* 46 (2001): 159–176.

Soille, Pierre, J. Vogt, and R. Colombo. "Carving and adaptive drainage enforcement of grid digital elevation models." *Water Resources Research* 39.12 (2003): 1366 1-13.

Soille, Pierre. "Optimal removal of spurious pits in grid digital elevation models." *Water Resources Research* 40, W12509 (2004): 1-9.

Tarboton, David. "TauDEM Toolbox." *Utah State University*. Web. 15 October 2012 <http://hydrology.usu.edu/taudem/taudem5.0/>