3D PRINTING TOPOGRAPHIC DATA: AN INTRODUCTION AND TECHNIQUE

Tomasz Mrozewski
Laurentian University

INTRODUCTION

This paper documents ongoing research into the application of 3D printing processes to GIS data. As 3D printing becomes more accessible to a broad audience, its application to GIS data is being investigated by industry, academia and amateur Makers. This paper will outline a technique for converting contour lines into a digital, printable 3D models using ArcGIS and Rhino software; other methods will also be briefly discussed. The end goal of this research is to develop a set of easy-to-follow guidelines for replicating this process in the context of an academic library, drawing on the author's own experiences.

BACKGROUND

The genesis of this project was a reference question received in August, 2014. A faculty member in Laurentian University's Outdoor Adventure Leadership sought printed 3D maps as an experiment in novel learning objects for use in the program's field trips. Without a clear sense of what they might be, the faculty member was excited by the prospect of knowledge discovery that might emerge from visualizing terrain in a new way. The faculty member also provided proof that other individuals and organizations were already able to create 3D prints of topographic data, including a service of the Geospatial Information Authority of Japan[1] and a post on the MatterHackers website[2].

ELEMENTS OF A 3D MODEL

Before giving specific details about the technique, I shall define terms and discuss the elements of model-making in general terms to contextualize the steps in the process.

Throughout the paper I have tried to distinguish between *digital models* and *printed models*. The bulk of the paper concerns the creation of printable, digital models. Once a digital model is fully and successfully created, printed models will easily follow.

I propose that there are five key elements in converting GIS data into printed model.

1. Elevation data
2. Terrain mesh
3. Sidewalls and base
4. STL conversion
5. Printing

Elevation data for an area may come in vector contour line or raster DEM format. Possible sources for elevation data for Canada covering my AoI include the Ontario Base Map (OBM) series, the National Topographic Series (NTS) and Shuttle Radar Tomography Mission (SRTM).

A *mesh* (also called a polygon mesh) is "a collection of vertices, edges, and faces that describe the shape of a 3D object."[3] *By terrain mesh*, I mean the terrain modeled from the elevation data, represented as a contiguous set of planes in 3D space. ArcGIS users will have encountered terrain meshes by creating triangulated irregular networks (TINs) from contour lines, or by visualizing elevation from the surfaces of a DEM in ArcScene.

In order to print, a digital model must form a watertight solid – that is to say, that it fully encloses a volume of space with no holes. Although extended in 3D space, a mesh has no thickness and is not a

[1]Geospatial Information Authority of Japan. Retrieved from http://cyberjapandata.gsi.go.jp/3d/index.html on July 16, 2015.

[2]Smith, Roy. "How to 3D Print a Map of Anywhere in the World". Retrieved from http://www.matterhackers.com/news/how-to-3d-print-a-map-of-anywhere-in-the-world on July 16, 2015

[3]"What is a Mesh?" From *Blender 3D: Noob to Pro*. Wikibooks, 2014. Retrieved from https://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/What_is_a_Mesh%3F on July 24, 2015.

printable solid. To create a printable object, the terrain mesh must be given sidewalls and a base to stand on. The *sidewalls* and *base* are also mesh surfaces which join with the terrain mesh to form a shell. If the shell is not watertight, 3D printing software cannot process and will not print the model.

Up to this stage, the digital model may be created in a variety of file formats. For printing, however, it requires *conversion* to STL format. STL (for "stereolithography") appears to be the most widely accepted format for 3D models by printers, although it appears that some printers are able to take models in other formats such as OBJ. It is beyond the scope of this paper to consider the viability or desirability of non-STL formats for printing.

It is also prudent to double-check an STL model for watertightness before printing to ensure that it is properly handled by the printer and printer software. You may wish to do this with an application separate from the one used to create the STL to ensure that there is no glitch or idiosyncrasy in the STL conversion.

Finally, I consider *printing* to begin with importing an STL model into the printer software, and to end with the completion of the print. Scaling a model to the printer bed and orientation of the print are handled here.

CONVERTING CONTOUR LINES TO A PRINTABLE MODEL

On the basis of my research to date, I propose that there are three major alternative techniques for converting GIS data into a printable model:

1. Contour line to STL
2. DEM to STL
3. terrain2stl

I have found the first of these techniques – converting contour lines to a printable STL file – to be the most successful in creating a detailed representation of a landscape. I believe that the TIN best represents the sloping aspects of AoI at the scale requested by the faculty member who requested the prints. I also believe that this technique is the most novel contribution of this paper, although this novelty lies in linking pre-existing techniques in GIS and 3D modeling rather than in any net new innovation.

Contour line to STL

I chose OBM contour lines as my elevation data, sourced from the OGDE FTP server hosted by University of Toronto. The OBM provides contours at 10m intervals for the AoI. The AoI overlaps several OBM tiles, which were stitched together in ArcMap[4] using the Merge function. Then, a polygon shapefile was created for the bounds of the AoI and used to Clip the contour lines.

The terrain mesh was created in ArcScene. The clipped contour lines were imported and converted to TIN with following parameters:

- Height Field = Z_VALUES
- SF Type = Hard Line
- Constrained Delauney = checked

Z_VALUES is the name of the variable containing the contour lines' altitude in the OBM dataset; when using non-OBM datasets, choose the appropriate variable. The Hard Line designation is required to render the TIN properly. Checking the Constrained Delauney variable means that neighbouring triangles in the TIN share a single edge between them rather than two, parallel edges, which ensures that the resulting model will be watertight.

Creating the TIN may become more complicated where the AoI is bounded on one or more sides by water, as it is in the case of the AoI requested by the researcher. It may be desirable to extend a polyline outwards from land if one wishes to represent the water feature in the model. If this is not done, the edge of the TIN will more or less follow the shoreline (see figure 1). To retain a rectangular footprint, a polyline feature must be created from the edges of the shoreline into the water. It should follow the boundaries of the AoI polygon, completing the rectangular shape on the edges(s) bordered by water (see figure 2). Once the polyline is drawn, edit the Attribute Table for the feature and Add Field to create an altitude field where the value is equal to the altitude of the shoreline in the contour line dataset. This feature must then be merged with the contour lines and a new variable created which is equal to

[4]All ArcGIS operations performed with ArcGIS for Desktop 10.1.

Z_VALUES plus the height of the new water feature[5]. The Height Field of the TIN, then, should be this new variable instead of Z_VALUES. Before exporting the model, it is also possible to "cheat" part of the sidewall when creating the TIN. The benefit of this option is that it ensures that the output TIN has a fully rectangular shape, which is easier to match to a base in 3D modeling software. As was described above with respect to modeling AoIs bounded by water, edges of the TIN will not be perfectly squared if the contour lines do not extend right to the edge of AoI.

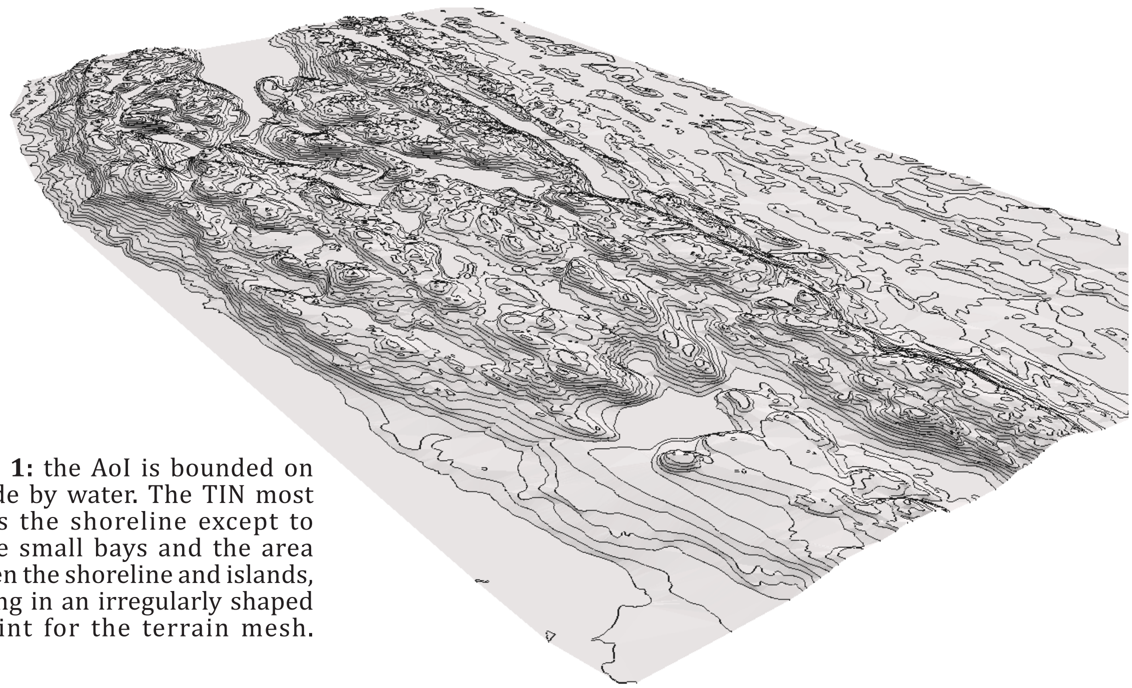

Figure 1: the AoI is bounded on one side by water. The TIN most follows the shoreline except to enclose small bays and the area between the shoreline and islands, resulting in an irregularly shaped footprint for the terrain mesh.

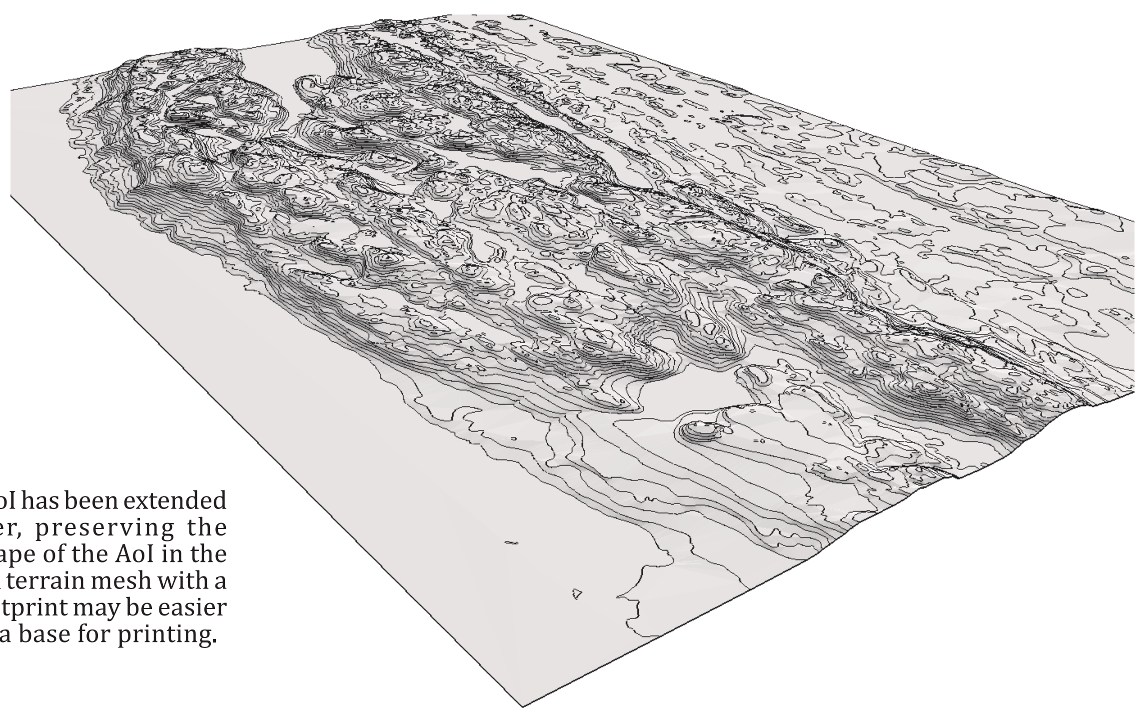

Figure 2: the AoI has been extended into the water, preserving the rectangular shape of the AoI in the terrain mesh. A terrain mesh with a rectangular footprint may be easier to mount onto a base for printing.

[5]For this new value to calculate properly, I found it necessary to export the merged feature class as a shapefile and then to calculate the final height field in the shapefile. The reason for this is that the output of the Merge function is a feature class in the geodatabase which place <null> values in each variable of the output feature class that is not shared by both input datasets. When attempting to calculate a new field by adding a number to a <null> value, the resulting value is also <null>. However, when the output feature class is exported as a shapefile, the <null> values are converted to 0. This allows the height fields of the contour lines and the water feature to be added in the new field without causing an error.

This will especially be a problem at the corners of the AoI, which will likely clip to the nearest contour line. To resolve this issue, create a polyline slightly larger than the AoI, which will become set the bounds of the output TIN. Assign a height value as low as or lower than the lowest contour line that meets the edge of AoI. The polyline feature will then be merged with the contour line dataset, as per the previous paragraph. The boundary polyline must be slightly larger than the AoI footprint to properly create the TIN because ArcScene's TIN creation function does not appear to be able to interpolate the vertical slope from the boundary polyline to the edges of the contour lines. The only downside I have found with this process is that the edges of the resulting model may show somewhat odd-looking sidewall artefacts.

Once a TIN is created, it can be exported as a VRML file using the Export Scene feature. At this point, the terrain mesh can be edited in 3D modeling software and given sidewalls and base.

The next step is to complete the shape of the model by adding sidewalls (if applicable), a base and converting to STL. These steps may be completed in 3D modeling software; for this project, I have chosen Rhinoceros software (also known as "Rhino"[6]. Note that 3D modeling software varies: everything from the mathematical modeling of the objects to the naturalism of the concepts to the names of the operations may vary from one to the next. I strongly recommend reading the documentation for your application of choice and completing tutorials on at least the most basic interactions.

In Rhino, the VRML TIN is treated as a mesh object. This is important because, by default, Rhino models 3D objects as NURBS (non-linear uniform rational b-splines)[7], so all manipulations to the model must be performed using the Mesh Tools toolset rather than the default toolset. Mesh objects cannot be converted to NURBs.

To create a base, use the Mesh Plane tool to draw a polygon that matches the footprint of the AoI, offset below it. Then, to create the sidewalls between the terrain mesh at the base, use the Single Mesh Plane to connect the corner vertices of the base with those of the terrain, or to connect the edges of the base with the edges of the terrain.

Although simple in principle, I found working in Rhino to be quite challenging in practice due to the specialized interface. Users with more experience in modeling or users of different software may find the process much simpler. For novices, I strongly recommending reading the manual and completing tutorials on basic interactions, if available, before attempting to manipulate models.

Once the sidewalls and base have been added, the model can be converted to STL by using Save As or Export Selected and selecting the file format; the two operations appear to be identical in Rhino.

Before printing the STL file, it is strongly recommended that the model be checked for watertightness. Holes in STL meshes may be undetectable to the naked eye, especially in large and complex models such as any landscape is bound to be. They may also be caused by reversed mesh faces, which may occur with software such as SketchUp. Alternatives include netFabb Basic (the free version of netFabb software that runs on desktop) and the Microsoft 3D Model Repair Service (a free, web-based service that runs netFabb software)[8]. Both netFabb Basic and the Model Repair Service will identify and automatically correct most holes in STL models.

Finally, the STL model can be imported into the 3D printer's native software. At this stage, the model may be scaled and oriented to the printer bed, if required. When modeling and printing terrain with relatively small changes in altitude, where low grade slopes challenge the minimum vertical resolution of the printer, it may be preferable to rotate the model 90° to print it standing on

[6]All Rhino operations performed with Rhinoceros 5 (64-bit) for PC.

[7]*Rhinoceros 5: User's Guide For Windows*. Robert McNeel & Associates, 2014. Retrieved from http://www.rhino3d.com/download/rhino/5.0/UsersGuide/ on July 24, 2015.

[8]3D Model Repair Service: https://netfabb.azurewebsites.net/ (retrieved on July 24, 2015).

the sidewall edge, rather than flat on the base[9].

Other methods: Working from DEMs and Terrain2stl
DEMs may also be used as source elevation data. A 2D DEM may be converted into a 3D shape by using ArcScene, exported and then treated according to the technique described above. Some 3D modeling software can directly convert a 2D DEM; Rhino, for example, has a tool called Heightfield from Image which will convert a DEM into a mesh[10]. Similarly, Global Mapper[11] GIS software can convert a DEM into an STL file. It has been beyond the scope of this project thus far to investigate any of these methods in detail.

Another interesting and convenient method for creating STL models is the Terrain2stl web app[12] and script[13] developed by Thatcher Chamberlain. Terrain2stl uses a Google Maps interface and a movable, sizable focal box to select an AoI. Once the AoI is selected, the user clicks Create STL File and then Download, which provides a fully watertight and printable model for printing. Terrain2stl works by querying the SRTM DEM dataset corresponding to the AoI, creating a vertex for every square in the DEM, joining each vertex with its neighbour on four sides and filling the space with a mesh surface, and then creating the base and sidewalls.

Despite its ease of use, Terrain2stl is not ideal for more precise work. Since is uses 3 arcsecond SRTM data, the resolution is significantly lower than with many openly accessible data sets, such as OBM contour lines and DEMs. The predefined shape of the AoI means that the output STL file may need to be trimmed in modeling software. There is also a built-in 300% vertical exaggeration in the script that may need to be scaled down if required. However, for lower-fidelity projects the application may be ideal.

Printing and results

I successfully created prints of my AoI by using the contour to STL conversion procedure described above and with Terrain2stl. I created test prints from both techniques in base-down and in sidewall-down orientations for comparison's sake. I found that using the contour to STL technique and printing on edge produced the most highly detailed. Unfortunately for purposes of this publication, it is difficult to photograph the monochromatic plastic prints in sufficient detail to warrant inclusion of illustrations.

I printed models on a commercial-grade Stratasys Dimension 1200 and a consumer-grade MakerBot Replicator Mini. Printers may vary in print size, consistency, resolution, and filling algorithm. Comparing prints of the same size from the same model from the Dimension 1200 and the Replicator Mini, it appears that the two printers handle long, gradual slopes differently. However, due the small size and small number of comparable prints, it is difficult to be more specific or to generalize.

As noted above, printing the model on its end rather than on its base resulted in a better rendering of the terrain. However, it also resulted in small blips in places where it appears that an extra blob of extruded filament had accumulated.

At time of writing, the printed models have yet to be presented to the faculty member who commissioned them. Should he decide to use them as learning objects, follow-up will be required to determine how useful the prints are and whether or not any novel forms of knowledge discovery have emerged.

Software choices

GIS
I used ArcGIS software for this project because of its accessibility through the university's Esri site license. Although ArcGIS is widely used easily accessible in many universities, I am interested in exploring the use of free GIS software – namely, QGIS – to develop a procedure accessible outside of the university context.

3D modeling
I chose Rhino for several reasons. It is a full-featured modeling platform, sufficiently powerful to handle the complexity of the terrain mesh, well-documented

[9]Credit is due to Greg Lakanen, Engineering technologist at Laurentian University, for suggesting this solution.

[10]I have found this to be only intermittently successful; however, I have not systematically investigated the reason for this or the parameters required for successful conversion.

[11]Global Mapper 16 for PC.

[12]Terrain2stl: http://jthatch.com/terrain2stl/ (retrieved on July 23, 2015).

[13]Code available on Github: https://github.com/ThatcherC/Terrain2STL (retrieved on July 23, 2015).

with good customer service, and is available through an affordable and flexible single-seat academic license with a 90-day trial period. The quality of documentation is especially important for users like myself with virtually no experience of 3D modeling. Unfortunately, the learning curve is fairly steep and, outside of the trial period, the software is not free.

I chose Rhino after failures to successfully create the digital models with SketchUp and Blender, both of which are free. SketchUp is a popular modeling program that is very easy to use due its more naturalistic – and, therefore, easier to learn – toolsets. It was designed for a broad audience and was owned by Google from 2006 to 2012[14], which may have contributed to its popularity. However, it suffered from major flaws when handling complex terrain models: it was slow and underpowered, sometimes requiring hours to complete certain processes or crashing entirely; user interface issues made it impossible to determine whether the application was processing or had crashed; STL output was often seriously flawed beyond the repair of netFabb or the Model Repair Service. I was unable to overcome these issues with the help of the fantastic SketchUp user community[15]. I encountered these issues using both the trial Pro editions and the free Make editions of the software[16].

The free, open-source Blender software proved to have difficulty handling the scale of TINs output from ArcScene and has an extremely high learning curve, so I abandoned it early in the research process. However, I was able to successfully convert TINs from VRML into SketchUp-compatible formats using Blender, even while being unable to properly display them. If the display and scaling issue could be addressed, I believe that Blender would be the best candidate for modeling in a completely free environment in conjunction with QGIS.

Future directions

My first priority is to develop techniques and step-by-step procedures for creating 3D models using entirely free datasets and software such as QGIS and Blender. I believe that opening the process up to a general public will be the surest way of discovering applications.

While the idea of 3D printing GIS data is novel and interesting, I do have doubts about its widespread utility. It is currently unclear to me that there are many situations in which the printed model is clearly more useful than a digital 3D model or even a topographic map. Due to the relatively low resolution of openly accessible elevation datasets, it is difficult to precisely model small areas. This limits its usefulness for purposes such as site planning, although this could be addressed by acquiring higher resolution terrain data.

A faculty member from the Laurentian School of Architecture has expressed interest using the technique I have developed to have students create models of the Sudbury Basin. For a sufficiently large printed model we will need to experiment with tiling the digital model – most printers have bed around 10” by 10” by 10” – and fitting the printed tiles together. It will be interesting to see how consistent printer output is and how well the tiles fit together. I suspect that architectural site modeling will also be one of the major applications of the conversion and printing technique described in this paper.

Several mining researchers and professionals with whom I have spoken expressed interest in creating models of mine shafts from 3D point clouds to aid high-level decision making, their belief being that the physical object may inspire knowledge discovery and discussion. In this case, it will be necessary to develop a new techniques for converting point cloud or structure from motion (SfM) data into a 3D model; in principle, this should not be difficult.

Conclusion

This research project, though not yet complete, has demonstrated that it is possible to convert contour line data into a 3D printable model of terrain. It also seeks to create a series of step-by-step instructions for doing so, using entirely free software if possible; this has yet to be accomplished. Finally, this project continues to seek and evaluate uses for the printing techniques in academic and other environments.

[14]“A Little SketchUp History.” Retrieved from https://www.sketchupschool.com/sketchup on July 23, 2015.

[15]Most notably on Reddit (https://www.reddit.com/r/Sketchup/) and SketchUcation (http://sketchucation.com/).

[16]Versions used throughout the project include SketchUp Pro 2014 and 2015, and SketchUp Make 2014 and 2015; no significant differences were encountered between any of these.