

Terrain Modeling Using Solid Freeform Fabrication

Josh Mueller
Department of Software Engineering
Milwaukee School of Engineering
1025 N. Broadway Street
Milwaukee, WI 53202

Rapid Prototyping Center
Milwaukee School of Engineering
1025 N. Broadway Street
Milwaukee, WI 53202

Faculty Advisor: Professor Michael McGeen

Abstract

Terrain data (elevation) and data that require location to be understood (roads, rivers, contamination levels, vegetation, etc) are currently stored in many formats. The United States Geological Survey provides data such as elevation, road location, and satellite imagery in Spatial Data Transfer Standard (SDTS) format. The United States Census Bureau also provides census data in the SDTS format. The Milwaukee Metropolitan Sewerage District provides water quality, water contamination, and water level data which can be accessed by running a query on a database. This project investigates methods that would convert these data from various sources and in various formats into a format that solid freeform fabrication machines can produce. This research has allowed for the rapid creation of highly-detailed, three-dimensional physical models, which could open the doorway for future exploration of relationships between sets of data. This research involved the investigation of data transfer standards, meshing algorithms, data storage formats, and data visualization.

Keywords: Terrain Modeling, Solid Freeform Fabrication, Data Visualization, United States Geological Survey, Meshing, Triangulation

1. Introduction

When data are examined in a new way or when different data are combined new relationships can emerge. This project is a first step at a new way of looking at terrain data and data that requires location to be understood. Wolf-Dieter Rase showed that data that requires location to be understood can be created using solid freeform fabrication (SFF) [1], and Laura Jacobs showed that terrain data can be created using SFF [2]. This project attempts to bring these components together to solve some of the problems encountered by each.

2. Background

2.1. solid freeform fabrication

Solid freeform fabrication is the name for a group of additive manufacturing processes. These processes build parts layer by layer from specialized materials. Traditional fabrication processes remove material to

form the required part. Solid freeform fabrication allows for the creation of complex geometries, such as a Klein bottle and a human skull.

SFF is particularly useful for this application as terrain can be very complex. Terrain may even have undercut areas that could not be produced by traditional manufacturing techniques.

2.2. spatial data transfer standard (SDTS) [3]

The Spatial Data Transfer Standard is intended as a way to transfer spatial data between dissimilar computer systems. This standard is used by the United States Geological Survey (USGS) to provide geologic information. The SDTS was approved by the National Institute of Standards and Technology (NIST) in Federal Information Processing Standard (FIPS) Publication 173 in August 1992.

A SDTS transfer has a specific method of organizing data. Each transfer is a grouping of files in the ISO 8211 (FIPS 123) physical file encoding. Usually each file contains one SDTS module. A SDTS module is a grouping of records, each record is a grouping of fields, and each field is a grouping of subfields. The subfield contains the actual data that is stored. The field specifies how the data is to be interpreted.

The SDTS can be extended by the use of profiles. The two profiles used in this project are the Topological Vector Profile and the Raster Profile. The data from the USGS that is in the Topological Vector Profile are the Digital Line Graph (DLG). This data is made up of the following layers: Public Land Survey System, boundaries, transportation, hydrography, hypsography, surface cover, non-vegetative surface features, survey controls and markers, and manmade or cultural features. The data in the Raster Profile are the Digital Elevation Map (DEM). This data has one useful layer, the raster layer containing the elevation data.

2.3. mesh / meshing [4]

Let V be a set of points. A mesh M of V is a set of triangles such that:

- Each triangle in M has three vertices of V for its corners, and intersects no other vertex of V .
- Triangle interiors do not intersect.
- Every point in the convex hull of V is covered by a triangle.

Although there are other names, in this paper the set of triangles will be referred to as a mesh, and the process of forming this mesh will be referred to as meshing.

Meshing is important to this project because SFF machines cannot build a model from a set of points; they require a surface model of the object. A mesh created from a set of points will provide this surface model.

2.4. greedy insertion

The algorithm used to perform the meshing is 'greedy insertion'. This type of algorithm is called greedy because it makes decisions that cannot be changed later on. This implementation chooses points that are not in the mesh and adds them to the mesh one at a time. It continues to do this until the mesh satisfies a set of criteria. The point that is chosen for insertion is the point that is farthest from the current version of the mesh.

There are several benefits to this algorithm: the mesh contains the smallest number of triangles possible, the algorithm can be stopped when a specific error value is reached, and the algorithm can be stopped when a specific point limit is exhausted.

When this algorithm adds a point to a triangle it then evaluates the edges surrounding the triangle for a certain property. It forms the polygon of the new triangle and the neighbor triangle; this polygon can have two possible sets of triangles formed by the diagonal. Figure 1 illustrates the point P being added to triangle T . Next the polygons $APCD$, $APBE$, and $CPBF$ are evaluated for the mesh property. If this evaluation fails the diagonal of the polygon is swapped. In polygon $APBE$ this would be swapping edge PE for edge AB .

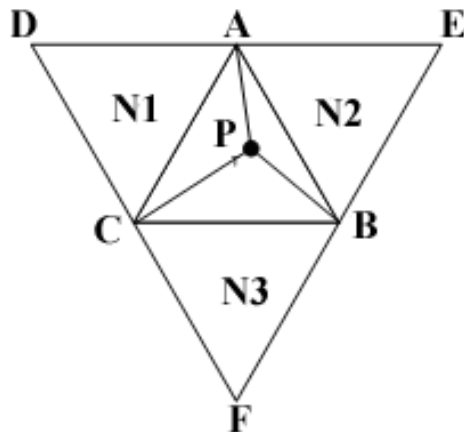


Figure 1: simple mesh example

The mesh properties investigated in this project were data dependent meshing and Delaunay meshing. Data dependent meshing chooses the set of triangles that best match the data being meshed. Delaunay meshing will swap the edge if the circumcircle of the neighbor triangle is not empty; meaning it contains point P. The circumcircle of a triangle is the circle formed by the three vertices of the triangle. The grey area illustrated in Figure 2 is an example circumcircle.

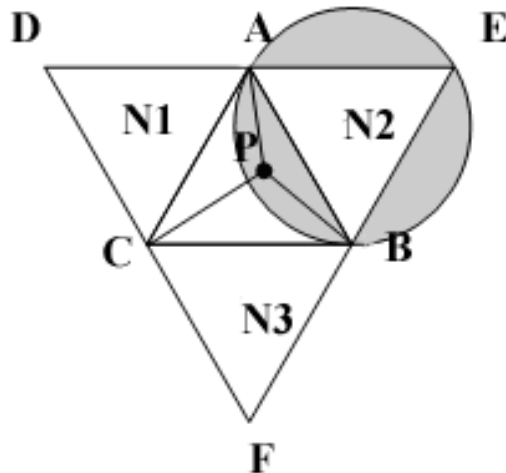


Figure 2: circumcircle example

3. Objective

The objective of this project is to convert terrain data and data that requires location to be understood from various sources to a format that SFF machines can produce. The data for the terrain model was found in the form of a SDTS DEM. The other data for the model can be found in SDTS DLG-3 files, United States Census Bureau SDTS/TIGER files, and the Milwaukee Metropolitan Sewerage District Corridor Study Database which contains water quality data for the Milwaukee River Basin.

In order to accomplish this integration it was necessary to investigate how to interpret these various forms of data from their various sources. It was also necessary to construct a framework to convert these data into a format that the SFF machines can construct.

4. Approach

To accomplish these goals several methods were tried. The first was to investigate ways of working with the SDTS transfers. The second was to investigate methods of meshing. Before writing new software to accomplish these goals current software was investigated.

4.1. SDTS transfer

A first attempt to read a SDTS transfer was using the SDTS++ library which was produced by the USGS. This library required another library to build the source code, and this second library required a third. This method was abandoned due to its complication.

The second attempt to read a SDTS transfer was using the SDTS abstraction library (SDTS_AL) which was designed and implemented by Frank Wamerdam [5]. It was constructed specifically because the SDTS++ was too complicated for use. After consulting with the author and reviewing previous work, this library was used to successfully read several SDTS transfers.

4.2. Meshing

A first attempt to create a mesh was with the Scape program constructed by Michael Garland of the University of Illinois at Urbana-Champaign and Paul Heckbert formerly of Carnegie Mellon University [6]. Although this program did create meshes, it was very difficult to understand the code.

The second attempt to create a mesh was with the Terra library which was the successor of the Scape library. This library was constructed by Michael Garland to fix some of his previous design mistakes. This library created meshes as well, but was constructed in such a way that extracting the meshing code for use in the new framework would have been difficult. The Terra library is also incomplete and unsupported.

The third attempt to create a mesh was with a new meshing library constructed specifically for this application. This library implements both data dependent greedy insertion and Delaunay greedy insertion.

4.3. obstacles

The latest version of the meshing algorithm was too memory intensive for use with large scale meshes. The time was therefore taken to extract and finish the Terra library meshing components and use them in this project.

5. Software model

5.1. software model overview

The framework constructed has three stages: the data input stage, the meshing stage, and the data output stage.

Each of these stages was constructed in such a way that whichever version is chosen the interface to the rest of the software is the same.

5.2. data input

The data input portion of this framework has the ability to read SDTS DEM transfers, Portable Grey Map (pgm) images, and bitmap images (bmp).

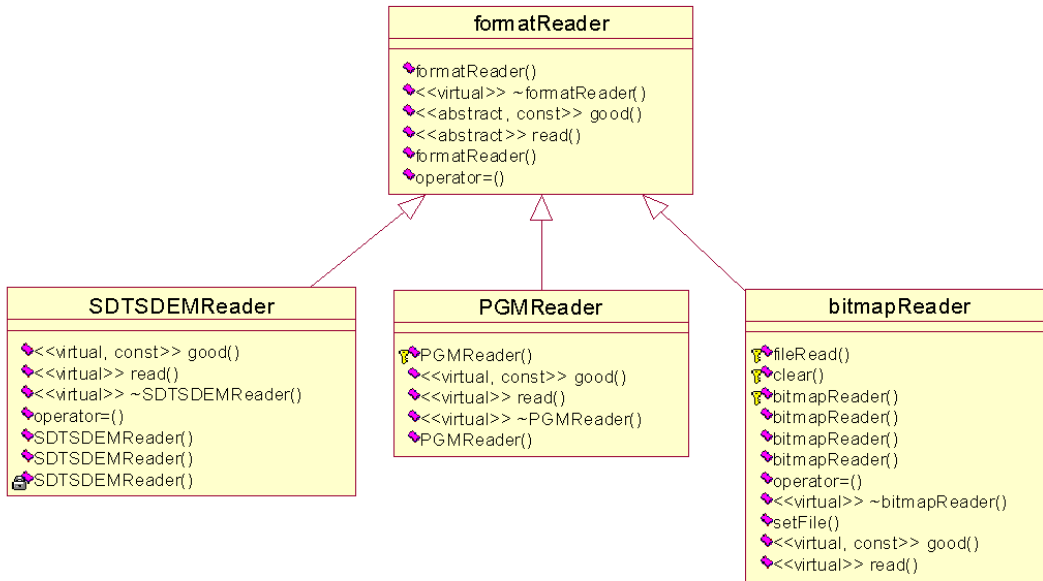


Figure 3: data input section class relationships

5.3. meshing

The meshing portion of this framework has the ability to mesh using data dependent greedy insertion, Delaunay greedy insertion, or using the Terra implementation.

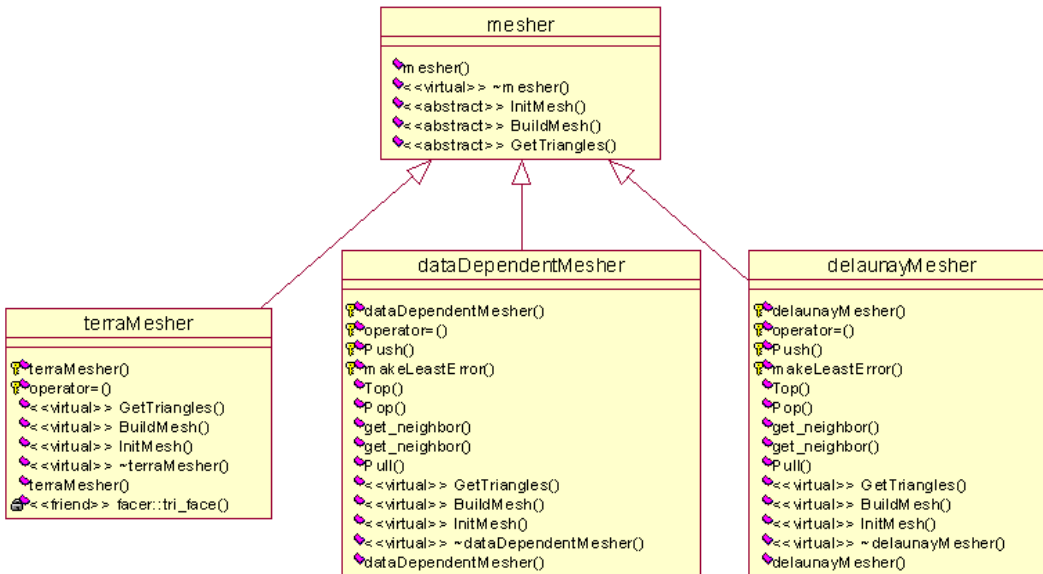


Figure 4: meshing section class relationships

5.4. data output

The current framework can write in the ASCII stereolithography file (STL) format [7] as well as the ASCII polygon file (PLY) format [8]. The PLY format can contain color, while the STL format cannot.

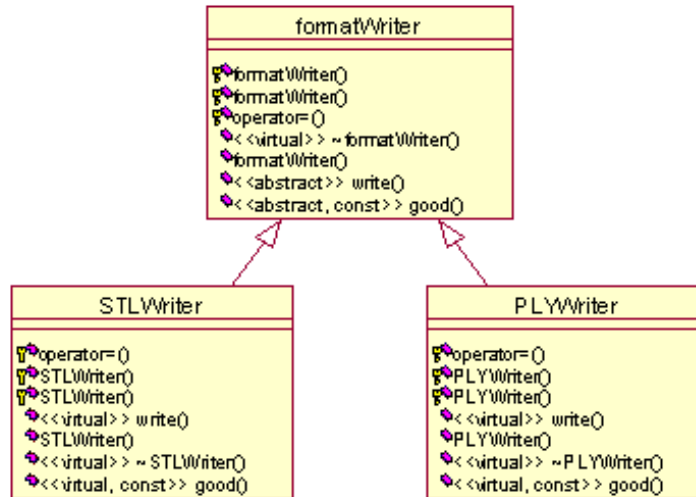


Figure 5: data output section class relationships

6. Implementation

The implementation of the framework to accomplish the project goals was only difficult when working within the meshing stage.

6.1. data input

The implementation of this part of the framework was easily executed once the intricacies of the underlying theory for the SDTS transfer were understood, and the implementation of the library used was determined.

6.2. meshing

The point selection behavior of the greedy insertion algorithm was fairly easy to implement. The insertion and reevaluation, however, caused much trouble. When the algorithm had been completed and testing began a problem was discovered; this meshing algorithm consumes so much computer memory that it cannot work with more than 250 points. The time was therefore taken to extract, finish, and convert the Terra meshing implementation for use in this framework. The Terra meshing algorithm has no trouble working with the approximately 200,000 points that make up the current input data set.

6.3. data output

This part of the framework was the easiest to complete, once the standards for the file formats to be used were understood. The two file formats used were the ASCII STL file format and the ASCII PLY file format.

7. Future Directions

The use of this framework for the creation of large scale terrain will require additional work to be done. An interpreter will be necessary to merge the multiple input files into one grid of points that require meshing. This is due to the fact that the data in the SDTS transfer is rotated several degrees relative to the grid that it is transported in.

The questions of which types of data to display as well as how to display them must also be addressed.

The current meshing algorithm works only for surfaces, however, the SFF machines can only produce solid objects. Several additions were made to transition from a surface to a solid object.

Implementation of a meshing algorithm that can operate on a convex hull of points would be a next logical step.

8. Conclusions

This project has led to the investigation of meshing algorithms, data transfer formats, and SFF machine input file formats.

Figure 6 illustrates a height shaded computer model created from USGS SDTS DEM data. This computer model was also successfully created on a SFF machine. Figure 7 illustrates the overlay of a bitmap onto a model created from USGS SDTS DEM data. This overlay bitmap was created to represent height coloring as well as USGS SDTS DLG-3 data, and is intended as an overview of what may be possible in the future.

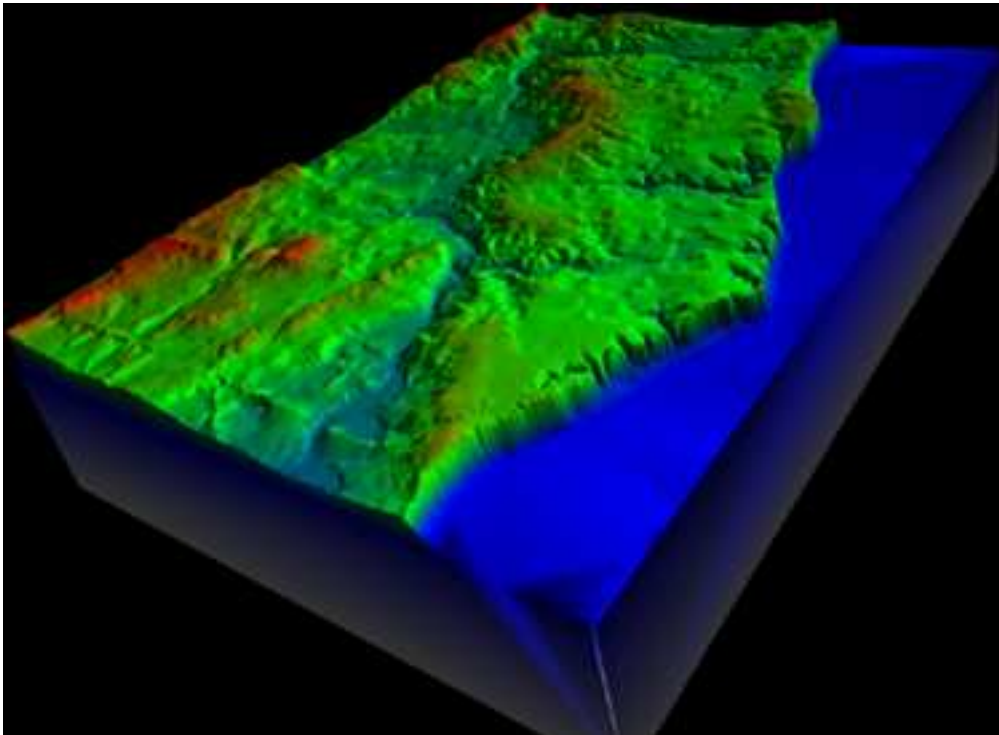


Figure 6: An example terrain that has been shaded based on elevation

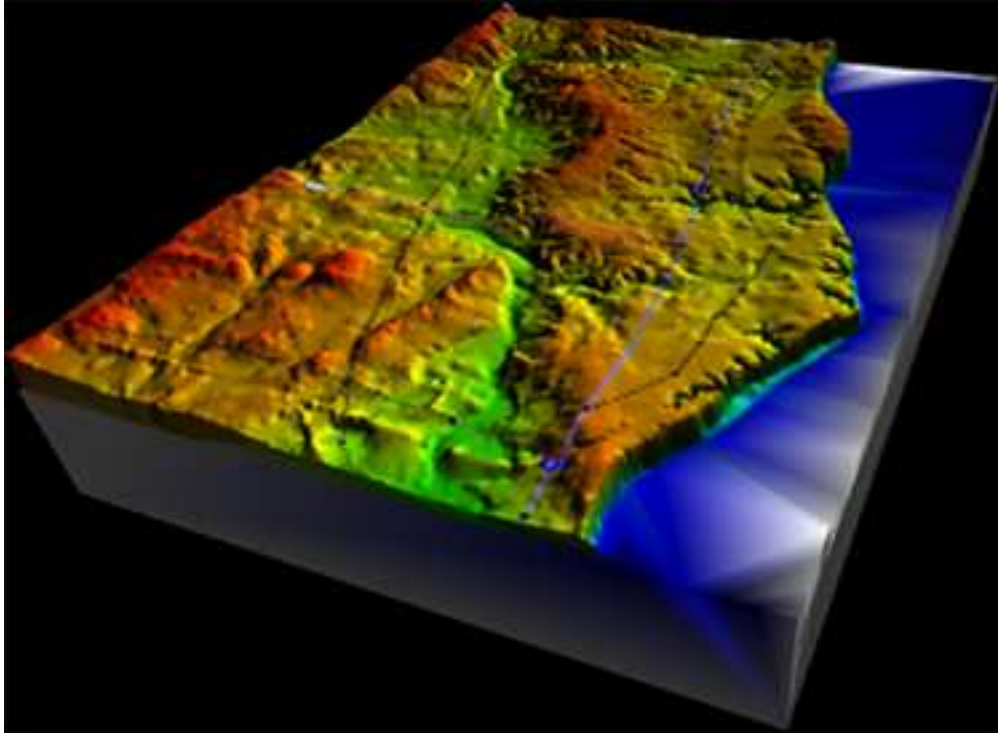


Figure 7: An example terrain with road overlay

9. Acknowledgements

The author would like to thank Professor Michael McGeen as well as all others who offered their help, expertise, and support.

This material is based upon work supported by the National Science Foundation under Grant No. EEC-0139142. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

10. References

- [1] Rase, Wolf Dieter. Physical Models of GIS Objects by Rapid Prototyping. Bonn, Germany: Federal Office for Building and Spatial Planning, 2001.
- [2] Jacobs, Laura. Terrain Modeling Using Rapid Prototyping. West Lafayette, Indiana: School of Civil Engineering Purdue University work performed at Milwaukee School of Engineering, 2003
- [3] U.S. Geological Survey, I. (2003 September 11). How is USGS implementing SDTS?. (Online), 2004 July 18. <http://mcmweb.er.usgs.gov/sdts/implement.html>
- [4] Shewchuk, Jonathan, I. (1999). CS 294-5: Meshing and Triangulation (Autumn 1999) Lecture 1. (Online), 2004 June 21. <http://www.cs.berkeley.edu/~jrs/mesh/1/lecture.html>
- [5] Frank Wamerdam., I. (1999). SDTS Abstraction Library. (Online), 2004 June 14. <http://gdal.velocet.ca/projects/sdts/index.html>
- [6] Michael Garland and Paul S Heckbert , I. (1995 September 19). Fast Polynomial Approximation of Terrains and Height Fields. (Online), 2004 July 18. <http://graphics.cs.uiuc.edu/~garland/papers/scape.pdf>
- [7] Tele-Manufacturing Facility, I. Tele-Manufacturing Facility Project. (Online), 2004 August 13. <http://www.sdsc.edu/tmf/Stl-specs/stl.html>
- [8] PLY – Polygon File Format. (Online), 2004 August 13. <http://astronomy.swin.edu.au/~pbourke/geomformats/ply>