Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2014)
"Excellence in Engineering To Enhance a Country's Productivity" July 22 - 24, 2014 Guayaquil, Ecuador.

Application of Data Mining Techniques to eliminate duplication of points in Geometric Datasets used for the production of Additive Manufacturing components or 3D Printed models

Rafael Obregon
Western Illinois University, Macomb, IL, United States, mfro@wiu.edu

ABSTRACT

Additive Manufacturing and 3D Printing processes begin with the creation of computer generated models of the parts to be produced. Once shapes are developed they are saved or exported into a format suitable for reproduction, most commonly to a polygonal or tessellated pattern. Geometric datasets tend to be large and include a considerable number of duplicated parameters. Data Mining can be applied to determine which elements of the dataset are most relevant and sufficient to replicate the shape. Algorithms and references are used to regroup the data into new patterns, eliminating duplications, and reducing the size of the dataset.

Keywords: 3D Printing, models, datasets, duplications, Data Mining

RESUMEN

La Fabricación Aditiva o Impresión 3D inicia con la creación de modelos computarizados de las partes a producir. Una vez desarrollados los modelos se exportan a un formato adecuado para su reproducción, comúnmente a un patrón poligonal o teselado. El conjunto geométrico de puntos tiende a ser muy extenso e incluye un considerable número de datos repetidos. La Minería de Datos puede aplicarse para determinar cuáles de los elementos en este conjunto de datos es relevante y suficiente para reproducir el modelo. Algoritmos y referencias son usados para reagrupar los datos en nuevos patrones, eliminando duplicidades y reduciendo el tamaño de la base de datos.

Palabras claves: Impresión 3D, modelos, puntos geométricos, duplicación, Minería de Datos

1. INTRODUCTION

Additive Manufacturing (AM) and 3D printing denote processes that embrace a wide range of materials and derivative processes employed to build parts suitable for end-use service. The virtually unlimited design freedom enabled by the technology allows the creation of shapes and the integration of feature and function that previously often required complex, costly, and long lead-time tooling (Graybill, B., et al).

The seemingly new technology, its origins dating back to the late 1980s, entails a formation process for developing solid objects by the sequential delivery of energy and/or matter to specified points in space to produce a part, even using different colors or materials. Current practice is to control the manufacturing process by computer using a mathematical model created with the aid of software. AM done in parallel batch production can provide a large advantage in speed and cost compared to alternative manufacturing techniques such as plastic injection molding or die casting. Results may involve custom parts, replacement parts, short run production, or series production. When the part is used in the development process only, the appropriate term is Rapid Prototyping or 3D printing (Laser Institute of America).

As the technology becomes more available and sales of equipment keep growing due to substantial price drops, the proliferation of AM and 3D Printing instruments are generating another Industrial Revolution (The Economist).

The technology is currently used in an unbelievable and always growing number of fields, including: automotive, aerospace, architecture, clothing, construction, dental, education, engineering, geographic information systems, jewelry, food, medical industries, and many others. It improves the transformation of concepts into physical models (*art-to-part*) in a matter of hours or days.

1.1 COMPUTER GENERATED MODELS

AM and 3D Printing processes begin with the creation of computer generated models of the parts to be produced. Once shapes are developed they are saved or exported into a format suitable for reproduction, most commonly to a polygonal or tessellated pattern. Geometric datasets tend to be large, and include a considerable number of duplicated parameters.

Most commercial software programs for drafting and design (like AutoCAD, Inventor, SolidWorks, etc.) can export or generate the datasets needed for the replication processes; none will directly produce *small* files. The graphic information created by means of a modeling software, normally referred to as Computer Aided Drafting/Design (CAD), is converted into a set of finite triangles –distributed over the *external* faces of the model– by which the object is represented.

1.2 DATA MINING

In general, Data Mining techniques are utilized to discover and extract *disguised* information, properties, or patterns from large databases. Data Mining can be employed to determine which elements of a CAD-generated dataset are most relevant and sufficient to replicate an object. Algorithms and references are used to regroup the data into new patterns, eliminating duplications, and consequently reducing the size of the dataset.

2. POINT-DATA REDUCTION ALGORITHM

The motivation to determine how to *streamline* CAD-data originated from the need to communicate design information among members of a development team; members might include designers, engineers, administrators, and even customers. Not everyone needs access to the same data, but all require suitable information to understand end-results. Furthermore, some team members may not have the software to retrieve CAD files.

CAD files are proprietary and commonly encrypted. Also, the data structure of CAD files incorporates geometric definitions, element attributes, and software-specific parameters that allow users to adjust, manipulate, transform, and visualize design information. If visualization of shapes or objects is all that is needed, most of the CAD-generated data is not essential.

The search for a simple approach to *convey* only geometric design data, essentially for visualization purposes, lead to the development of an application to extract only point-data from the CAD models. Initial attempts consisted in utilizing some of the established methods for data-exchange (e.g., DXF, IGES, STEP). Most commercial CAD programs have the capability to produce these formats, which relay on vector-defined sets of information. Each data-exchange format has a different file structure, complicating the process of extracting individual and unique points. Additionally, point-data duplication is natural; the same coordinate parameters will be present several times in the database.

The *STL* file format, native to the *Stereolithography* process, provides a simpler and more congruent source for data extraction and simplification even though it presents point-data replication. STL data can be also produced by most commercial software programs. Moreover, with the advent and proliferation of prototyping equipment, relying only on the traditional methods for data-exchange will exclude using the geometric datasets for AM or 3D Printing tasks.

The algorithm for extracting point-data was developed along with a procedure to *re-structure* the information so that it could be visualized through a universally accessible platform (i.e., a web browser), while maintaining the integrity needed to produce prototypes. Ultimately, the resulting files can be displayed almost effortlessly in web browsers, requiring only plug-ins in some cases, and are fully compatible with the STL format required for AM or 3D Printing. The information can be disseminated without ambiguities or the need for specific programs.

The development of the algorithm to eliminate point duplication in CAD-generated datasets for visualization or prototyping of designs was first achieved using AutoLISP (an implementation of the List Processor programming language in Autodesk's AutoCAD).

2.1 IMPLEMENTATION

The simplification process of CAD data begins with the creation of a Standard Tessellation Language (*STL*) file, developed by 3D Systems Inc. The STL format has become the industry's *de facto* standard for data transmission. This format approximates the surfaces of a model with triangles. The number of triangles needed to define the geometry can be considerable. The more complex the shape, the more triangles produced (3D Systems, Inc.).

Inherently, the tessellation process will generate point sets for every triangle (or facet) required to represent the object using the Cartesian coordinate system. Facets are connected to other facets, and each facet-defining point (vertex) is registered as many times as needed. The same *XYZ* coordinate values will be present several times, thus the *duplication* or replication of the *same* data.

The STL format, in standard plain ASCII character-encoding, can easily be analyzed. Point data can be identified, mined, and minimized. The analysis incorporates concepts of Feature Extraction and Clustering. The objective is to isolate each unique piece of data, in the form of *XYZ* coordinates.

After point data has been minimized and adjusted, proper connectivity needs to be reestablished. This task is accomplished by comparing vertex connectivity (ordering) of the original STL against the reduced data set. Once connectivity is regenerated and *enhanced*, facet redefinitions can be finalized.

The last step is to reconstruct the STL file based on the reduced point data and the reorganized connectivity. As the file is structured in the correct format it can be uploaded into any software or equipment capable of importing STL format files.

A standalone application is still under development, and proprietary, but the concept can easily be applied, demonstrated, utilized and validated by means of generating an output file compatible with the Virtual Reality Modeling Language (*VRML*).

The VRML file format (*.wrl*) is based on the same tessellation foundation of the STL format, what differs is the resulting output file. Also, for demonstration, and primarily visualization, VRML data is simpler to display.

Although VRML has been superseded by the Extensible 3D (X3D) standard, files can be read by any X3D-savvy application, and the specification is still useful (Web3D Consortium). The National Institute of Standards and Technology indirectly supports the format (NIST).

2.2 ASSESSMENT

The basic steps in assessing the algorithm involve:

- Creating Models and STL data
  – generate different sample models with CAD software and save them to STL format
- Extracting Data
  – isolate vertex information from STL data files developed for each sample model
- Data Mining
  – determine minimum number of points (vertices) required to represent shapes
- Restructure Connectivity
  – relate points that define geometry and regenerate connectivity
- Create Output
  – structuring of output model files based on reduced point sets and enhanced connectivity

2.3 EXAMPLE

The operation of the algorithm can be demonstrated using the geometric model of a simple cube (Figure 1).

After generating the model of a cube, the next step consists in exporting the object to STL. The file incorporates the definition of 36 vertices (two triangles × six faces) of a 2" length cube. A portion of the resulting STL output file, in plain ASCII character-encoding, is shown in Table 1.

Table 1: Partial STL Output File

```
facet normal 0.0000000e+000 0.0000000e+000 1.0000000e+000
  outer loop
    vertex 2.0000010e+000 2.0000010e+000 2.0000010e+000
    vertex 0.0000000e-006 2.0000010e+000 2.0000010e+000
    vertex 2.0000010e+000 0.0000000e-006 2.0000010e+000
  endloop
endfacet
facet normal -0.0000000e+000 0.0000000e+000 1.0000000e+000
  outer loop
    vertex 0.0000000e-006 2.0000010e+000 2.0000010e+000
    vertex 0.0000000e-006 0.0000000e-006 2.0000010e+000
    vertex 2.0000010e+000 0.0000000e-006 2.0000010e+000
  endloop
endfacet
facet normal 0.0000000e+000 0.0000000e+000 -1.0000000e+000
  outer loop
    vertex 2.0000010e+000 0.0000000e-006 0.0000000e-006
    vertex 0.0000000e-006 0.0000000e-006 0.0000000e-006
    vertex 2.0000010e+000 2.0000010e+000 0.0000000e-006
  endloop
endfacet
facet normal 0.0000000e+000 0.0000000e+000 -1.0000000e+000
  outer loop
    vertex 0.0000000e-006 0.0000000e-006 0.0000000e-006
    vertex 0.0000000e-006 2.0000010e+000 0.0000000e-006
    vertex 2.0000010e+000 2.0000010e+000 0.0000000e-006
  endloop
endfacet
```

Once the STL file is generated the next step consists in extracting from it the minimum number of points that define the shape. In the case of a cube, the minimum number of points (*XYZ* coordinates) will be just eight, one at each corner (Table 2).

Table 2: Minimum Points

2.000000	0.000000	2.000000
2.000000	2.000000	2.000000
0.000000	2.000000	2.000000
0.000000	0.000000	2.000000
0.000000	2.000000	0.000000
2.000000	2.000000	0.000000
2.000000	0.000000	0.000000
0.000000	0.000000	0.000000

The first task performed by the algorithm is farming (reading and converting) data from the STL file. As the data is identified it is converted from ASCII characters into *real* numbers.

Once the end of the file is reached, the *XYZ* points are mined and minimized. Additionally, the algorithm *centers* the points about the absolute *XYZ* Cartesian origin, based on the extreme points of the shape (Table 3).

Table 3: Adjusted Points

1.000000	-1.000000	1.000000
1.000000	1.000000	1.000000
-1.000000	1.000000	1.000000
-1.000000	-1.000000	1.000000
-1.000000	1.000000	-1.000000
1.000000	1.000000	-1.000000
1.000000	-1.000000	-1.000000
-1.000000	-1.000000	-1.000000

After the points have been minimized and adjusted, an identification index is assigned to each point by the order in which they are registered in the original STL file. An index set is then assembled (Table 4). This index set represents the original tessellated pattern, or connectivity map.

Table 4: Index Set

0	1	2
0	2	3
4	5	6
7	4	6
0	3	7
6	0	7
3	2	4
7	3	4
2	1	5
4	2	5
1	0	6
5	1	6

The index set can also be minimized based on surface (facet) topology. As the tessellation process generates triangular facets, flat surfaces are defined by several triangles. Flat surfaces can be defined using *n-sided* polygons instead of triangles, which allow for further data set reductions. In the case of a cube, six *4-sided* facets are sufficient to define the shape, instead of twelve *3-sided* facets (two per face). Transforming the topology of flat surfaces to *n-sided* polygons, instead of just sets of triangles, results in an enhanced connectivity map (Table 5).

Table 5: Enhanced Index Set

0	1	2	3
3	0	6	7
7	3	2	4
4	2	1	5
0	1	5	6
4	5	6	7

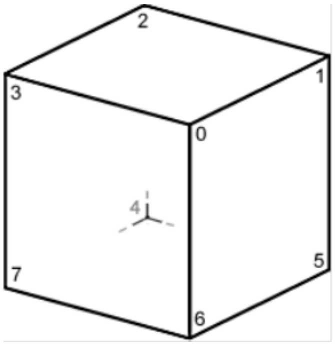

Figure 1: Cube Model

The last step of the process is to construct a suitable output file, based on the reduced dataset and the enhanced connectivity map. The output file for this example, in VRML format, is shown in Table 6.

Table 6: VRML Output File (Cube)

```
#VRML V2.0 utf8

# AutoCAD to VRML 2.0
# By: Rafael Obregón

Background { skyColor  [1.0 1.0 1.0] }
NavigationInfo { type "EXAMINE" }

DEF SamplePart1
        Transform {
                scale 1.0 1.0 1.0
                children [
                        Shape {
                                appearance Appearance {
                                        material Material {
                                                diffuseColor 0.5 0.5 0.5
                                        }
                                }
                                geometry IndexedFaceSet {
                                        ccw TRUE
                                        convex TRUE
                                        creaseAngle 0.5
                                        solid FALSE
                                        coord Coordinate {
                                                point[
                                                        1.000000 1.000000 -1.000000,
                                                        1.000000 1.000000 1.000000,
                                                        -1.000000 1.000000 1.000000,
                                                        -1.000000 1.000000 -1.000000,
                                                        -1.000000 -1.000000 1.000000,
                                                        1.000000 -1.000000 1.000000,
                                                        1.000000 -1.000000 -1.000000,
                                                        -1.000000 -1.000000 -1.000000,
                                                ]
                                        }
                                        coordIndex [
                                                        0,1,2,3,-1,
                                                        3,0,6,7,-1
                                                        7,3,2,4,-1
                                                        4,2,1,5,-1
                                                        0,1,5,6,-1,
                                                        4,5,6,7,-1,
                                        ]
                                }
                        }
                ]
        }
```

3. BENCHMARKS

In order to test the effectiveness of the algorithm, the generated output was compared to commercial products with the capability to export a VRML format. The chosen products were Autodesk's 3ds Max Design and VRMLout. VRMLout can be obtained from CAD Studio, Czech Republic (cadstudio).

In both cases, the files generated by the benchmark products for a cube of the exact same characteristics as the one used for the example include 24 points, compared to the eight generated by the algorithm.

3.1 ADDITIONAL MODELS

The process described in the example of section 2.2 was repeated for an additional three different models, along with benchmark tests. The output files for each of the different models processed with the algorithm were always smaller than those generated by the commercial products. The reductions of data sets varied from 20% to over 80%, depending on the complexity of the model.

The additional models, identified as Model 1, Model 2, and Model 3, are shown in Figure 2.

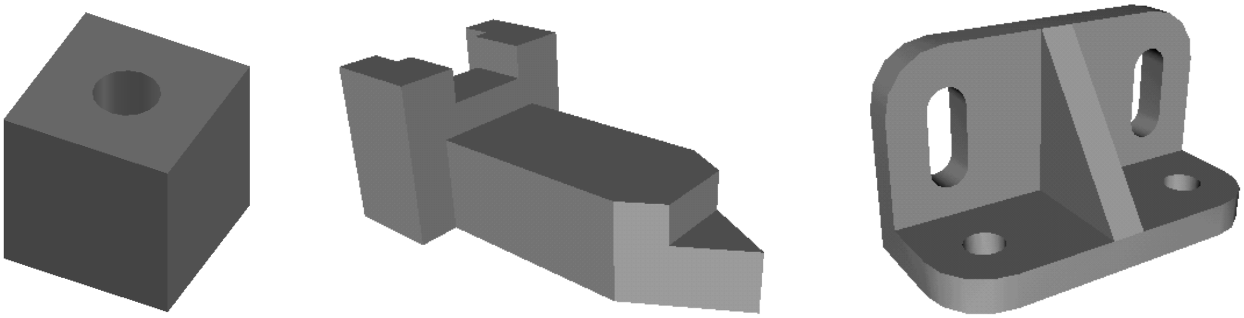

Figure 2: Additional Models

Point reductions and benchmark results are presented in Table 7.

Table 7: Additional Models Point Data

Model	Standard STL Output	Commercial Output	Algorithm Output
1	432	88	72
2	216	144	38
3	1344	391	218

Note that as model complexity increases, STL data sets grow dramatically, particularly when the models include round elements. The tessellation of curved surfaces requires the development of a considerably larger number of facets.

4. REDUCTION OF *STL* DATASETS

The strategy for reducing the size of STL data sets for AM or 3D Printing techniques is similar to the one previously described, but requires some additional considerations, and more steps.

First, since the objective is to create a physical reproduction of the computer generated model by AM or 3D Printing equipment, the polygonal mesh that represents the object has to be consistent. Consistency in this case is denoted by the concept of a *water-tight* surface. The facets generated by the tessellations of the STL format approximate all the elements related to the external faces of an object exactly in that way, producing a *water-tight* surface model. By using only triangles, the duplication of data is unavoidable.

As previously stated, flat polygonal faces can be represented and replicated using *n-sided* polygons, instead of just by the triangular facets generated by the standard STL format. In the case of a cube the reduction of vertices goes from 36 to 24 (a 33% reduction). When objects consist of exclusively flat *polygonal* faces it is possible to reduce the number of vertices. A schematic representation of facet reduction is depicted in Figure 3. In both cases, the resulting object is a *water-tight* surface model.

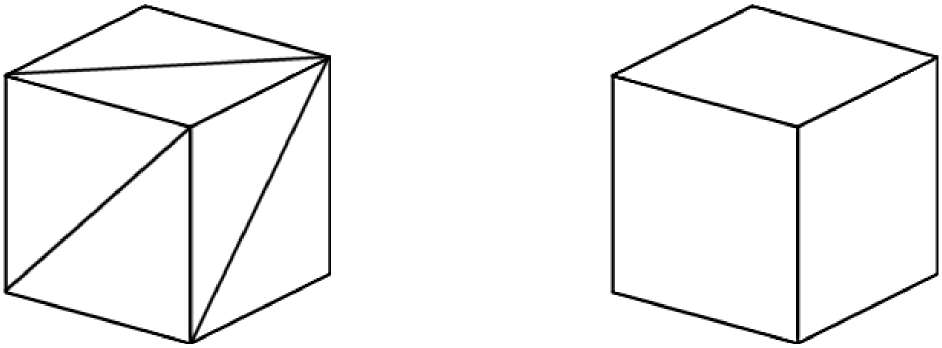

Figure 3: Facet Reductions

Round element facets can also be reduced, as long as the vertices follow the same topology. Consider Model 3. The filleted (rounded) corners of the object all have the same topology, thus it is possible to reduce the vertices of these elements too (Figure 4). Note that facets between linear and curved elements are not shown.

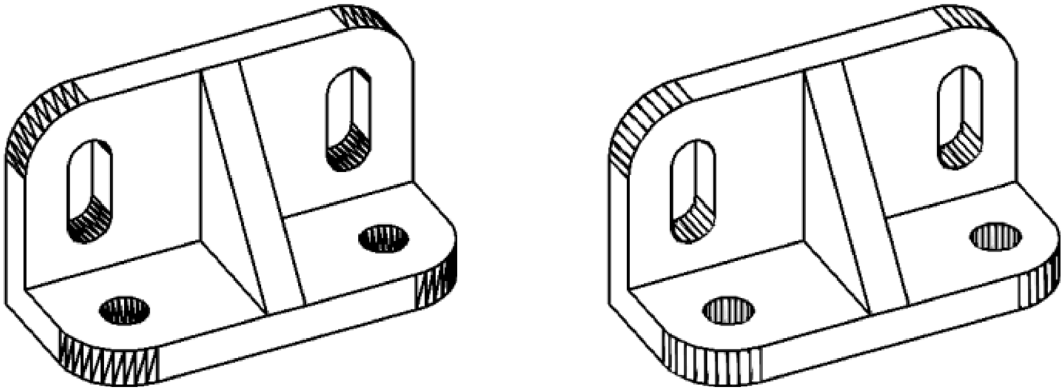

Figure 4: Facet Reductions of Curved Elements

The reduction process of tessellated faces on surfaces that combine linear and curved elements is more complex, although still achievable. The approach for such cases consists in isolating curved elements, reducing peripheral facets, not the facets around curves. An illustration of isolated facets is shown in Figure 5.

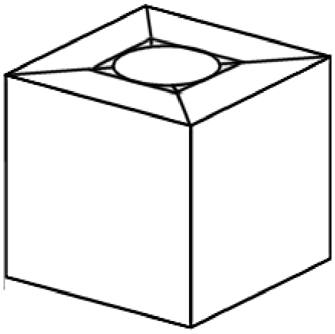

Figure 5: Facet Reductions of Linear-Curved Features

5. CONCLUSIONS

The algorithm for reducing STL data sets is more involved than the one used to reduce VRML data sets, but performs essentially under the same principles. The objective is to identify regular topologies, flat or curved, and to determine where to preserve or isolate facets, mostly when voids are present.

Generated output files represent a consistent *water-tight* surface, which is the main outcome, the objective.

The algorithm has proven to be accurate, and performs efficiently, it minimizes data sets and generates enhanced topology maps of the shapes that are to be produced by AM or 3D Printing techniques, even correcting minor modeling mistakes.

Furthermore, the algorithm is still evolving. Initially it was developed just with the aim to reduce the dataset size of the STL files generated by commercial software programs. The resulting mined STL files did not speed the AM or 3D Printing processes, which are determined by hardware firmware. Time savings occurred during the validation of the *water-tight* surface model, performed by the equipment's firmware.

Also, in cases where inconsistencies were present in the STL file (particularly gaps), due to modeling errors or loose tessellation tolerances, the algorithm eliminated the problems and rendered an improved, corrected model, based on the refinements and adjustments of point data and the enhanced connectivity maps.

Overall, results indicate that *improving* data sets and connectivity maps for AM and 3D printing processes is feasible.

REFERENCES

Graybill, B., Berlin, J., Hoyer, A., Corl, S., El-Gizawy, A. S., Kays, M. (2013). "Direct Digital Manufacturing for Fabrication of Functional Products". *Presented at: Composites Manufacturing 2012, SME*, Society of Manufacturing Engineers, Dearborn, MI 48128.

NIST – National Institute of Standards and Technology. (n/a). VRML Plugin and Browser Detector. http://cic.nist.gov/vrml/vbdetect.html, 01/24/2014.

Laser Institute of America. (2012). Additive Manufacturing – The Most Important Manufacturing Technology? http://www.lia.org/blog/2012/08/additive-manufacturing-set-to-become-the-most-important-manufacturing-technology, 02/11/2014.

cadstudio. (n/a). VRMLout for AutoCAD. http://www.cadstudio.cz/vrmlout, 02/11/2014.

The Economist. (2012). A third industrial revolution. http://www.economist.com/node/21552901, 12/12/2013.

3D Systems, Inc. (n/a). What Is An STL File? http://www.3dsystems.com/quickparts/learning-center/what-is-stl-file, 08/06/2013.

Web3D Consortium. (n/a). VRML97 and Related Specifications. http://www.web3d.org/x3d/specifications/vrml, 04/16/2012.

Authorization and Disclaimer