Subdividing Large 3D Models for Use in 3D Printing

Jason Kraft and Uyen Uong

Summary

3D printing has proven to be an extremely effective method for manufacturing complex models that were previously impossible to do with traditional methods such as milling or molding. Prominent applications for 3D printing range greatly from rapid prototyping parts for use in product design to character creation for use in stop motion films. While the cost of producing high quality 3D printers has decreased significantly over the years, restrictions due to size constraints severely limit the maximum volume a model can occupy. In addition, as is the case with common Fused Deposition Modeling printers, printing large models as one part runs with a higher risk of print failure.

Our goal with this project is to develop an algorithm to automate the process of subdividing large scale models into printable subcomponents. This algorithm will optimize these subdivisions based on printability (parts must fit inside the working volume of the printer), assemblability (it must be possible to assemble the parts into a finished model), efficiency (minimize the number subcomponents), and connector feasibility (each surface connecting to other parts must be large enough for male and female connector protrusions).

Background

We have found various research papers discussing implementations of automatic partitioning systems for 3D models. While they all describe methods of dividing up the model into printable-sized components that can later be reassembled, the papers' methods of partitioning vary. Because of that, our ultimate algorithm that we wish to implement will be a variant comprised of all of the papers, with emphasis on the first paper.

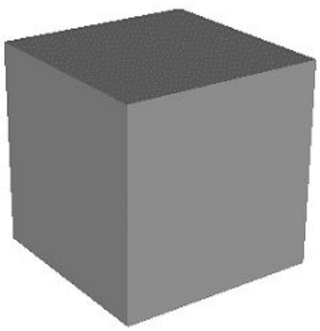
The first paper is called "Chopper: Partitioning Models into 3D-Printable Parts" (2012) by Luo et al. This paper describes the framework, Chopper, that the authors created, which divides the given 3D model into parts that are feasible to create within the printing volume. In order to accomplish this, the authors used a Binary Space Partitioning tree to represent the planar cuts used to divide the model. The cuts were determined based on a set of criteria that included printability, assemblability, efficiency, connector feasibility, structural soundness (no thin parts and seams are in areas of low stress), and aesthetics (seams are unobtrusive). Simulated annealing was used to determine the placement of the connectors, which are hexagonal prisms.

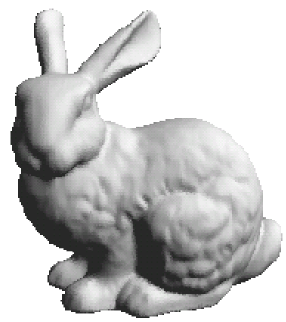
The second paper is called "Automatic Subdivision and Refinement of Large Components for Rapid Prototyping Production" (2006) by Medellin et al. This paper proposes a new technique for decomposing 3D models into various sizes and geometries, rather than thin sheets, which have previously been done. The system uses an iterative method of defining 3D units to divide the model. An initial 3D unit set is created using a regular 3D lattice and then refined through design operators that check for manufacturing difficulties. These operators include checking for thin sections, cusps and undersized volumes. A random pattern of spheres are used for the connectors to ensure that each connection has a unique orientation to aid in the assemblage.

The third paper is called [“Level-Set-Based Partitioning and Packing Optimization of a Printable Model”] (2015) by Yao et al. This paper describes a system for partitioning models not only to sizes that are printable, but also to reduce printing time and packing space. Their method for partitioning uses level sets to represent the model and focuses on minimal stress load, interface area and packed size, surface detail alignment, printability, and assembling. The authors used cylinders as connectors, using the same method as in the first paper.

Examples

We plan to run this algorithm on several models ranging from simple to very complex.





Cube

The cube model would consist of a single large cube which we will use as our simplest model.

Stanford Bunny

The Stanford bunny will act as a more complex genus 0 model. Though it is homeomorphic to the cube model, it will be interesting to see how the algorithm handles the relatively thin isthmuses of the bunny’s ears.

Happy Buddha

The happy buddha model will be our most complex model with a topological genus of 5. This will likely prove a big challenge for our algorithm since it has a number of thin features and none of them are perfectly symmetrical.

Timeline

4/6 - 4/12

- Develop BSP-tree class for representing partitions.
  - Jason
- Command line parser.
  - Uyen
- Basic renderer that shows partitions as different colors.
  - Uyen
- Load .obj files into program
  - Uyen

4/13 - 4/19

- Establish printability objective function.
  - Uyen
- Establish assemblability objective function.
  - Jason and Uyen
- Establish efficiency objective function.
  - Jason
- Writing partitions to .obj files.
  - Uyen

4/20 - 4/26

- Establish connective feasibility objective function.
  - Jason
- Adding connectors between partitions.
  - Jason
- Run test models through algorithm and debug.
  - Jason and Uyen
- (OPTIONAL) Print resulting test models with a 3D printer.
  - Jason and Uyen

4/27 - 5/1

- Write paper discussing our work and prepare presentation
  - Jason and Uyen
- (OPTIONAL) Slider for changing weights for objective functions
  - Jason and Uyen
- (OPTIONAL) Viewing separate partitions in same viewport
  - Jason and Uyen