Branching Support Structures for 3D Printing

Ryan Schmidt, Nobuyuki Umetani
Autodesk Research
Toronto, Canada
Overhangs

• Local Y-minima

• Shallow draft angle
Support == Waste

• Waste of **Time** and **Material**

• Longer print ➔ increased chance of failure

• More support ➔ more post-processing
Goals

1) Easy to Remove

2) Minimize Print Time
   – Support Volume
   – Travel Time

3) Reliable…
Advantages

• 75%+ reduction in support material usage
• 25-50% reduction in print time
• Improvements increase with print size — (for very small prints we see smaller differences)
• Posts are easier to remove
• Fewer surface artifacts
User Feedback

• First released in September 2013

• Immediately popular with 3D printing enthusiasts
  – “Despite all of its rough edges, we’re using meshmixer-generated support for 5-10 prints a day at our office.”
    nick@socialbicycles.com
Cone Constraint

• FDM printers can print at some *draft angle*
Cone Constraint

- FDM printers can print at some *draft angle*

- Support for Overhang can be anywhere inside *Support Cone*
Support Hierarchy Generation

• Assume we have a point that needs support

• Can connect point (via support post) to:
  1) Ground
  2) Object
  3) Another post

• Problems to solve:
  1) Support Points?
  2) Connection Strategy?
Support Points

- Compute point-sampling of overhang areas
- Support each Y-minima
- Densely cover overhangs from bottom to top ("watershed")
Top-Down Growing Strategy

1) Add support points to priority queue
   - Sorted top-down

2) While queue is not empty
   - Pop topmost point P
   - Try to connect downwards to ground/object.
   - Else try to pair with closest free point
   - Else grow downwards a fixed step.
   - Add new point to queue
Decision Problem

• At a given point, where should we “grow” or connect to?
Same Volume!
Shorter Travel!
Travel Time Optimization

• Exploit cone constraint

• Prefer connections closer to:
  – print surface
  – other posts

• “look ahead” in growing step, to try to end up near model
Search Problem

• Many potential connection points

• How to find them?
• How to decide?

• Lots of heuristics:
  – Distance
  – Angle
  – Collisions
  – …?
Termination Points

- Point-sampling of object and ground
- Filter out risky connections (e.g., onto steep areas)
- Allows efficient spatial queries
Support Graph

- Algorithm generates graph
- Generate support mesh from graph
- Try to optimize the graph as postprocess
Graph Optimization

- Move graph nodes towards surface or other nodes
- Satisfy cone constraints, avoid collisions
- Complex minimization problem
  - Many Degrees-of-Freedom
  - “Energy” is hard to formulate, bound to be expensive...
- Simple force-based approach
  - Surface & Graph Edges “pull” on each Node
  - Take small timesteps, fixed number of iterations
  - Converges, sometimes
• Lower travel speeds

• Adjacent surface limits post flexing

• Stronger “hair-bond” with adjacent surface
Graph Pruning
Post Geometry

- Tips: truncated sheared cones
- Posts: sheared cylinders
- Bases: complicated
  - Like a little raft for each post
  - Smooth transition distributes stress, better layer bonding

- Triangle count is a problem with large-format printers
Reality of “Modern” Slicers…
“Convert to Solid”

• Create implicit representation

• distance field around support graph (*not mesh*)

• Mesh with Marching Cubes
Future Work

• Tailoring to different print processes
  – eg. For dissolvable material, tips could be (clipped) spheres instead of cones

• Incremental stability/robustness analysis
  – Take layered manufacturing process into account
  – Make sure print is physically stable during the entire process
G-Code Optimizations

• Currently sending support as part of model mesh

• Optimizations at G-Code level:
  – Print at higher speed
  – Lower quality (eg 2x layer height)
  – Improve connections with model

• Need support from slicer
Bridging the Gap: Automated Steady Scaffolding for 3D Printing

Jérémie Dumas
Université de Lorraine, INRIA

Jean Hergel
INRIA, Université de Lorraine

Sylvain Lefebvre
INRIA, Université de Lorraine

Figure 1: The upper leg of the Poppy robot (www.poppy-project.org) cannot be 3D printed on low cost FDM printers without support. Our technique automatically generates scaffolding made of horizontal bridges supported by vertical pillars, shown in purple. The print is shown in the middle and on the right after cleanup. Bridges are strong and stable, increasing the print reliability while having a low material usage.

Abstract

Fused Filament Fabrication (FFF) is the process of 3D printing objects from melted plastic filament. The hot plastic exits a nozzle and fuses with the part just below, adding a layer of material to the object being formed. However, filament can only be deposited on top of an existing surface. Therefore, overhangs require a disposable support structure to be printed, temporarily supporting the threads of plastic that would otherwise hang in empty space.

Existing techniques for support generation fall into two categories: the first allow for very reliable prints by enclosing the bottom of the object in a dense structure, at the expense of increased material usage and build times. The second generate thin hierarchical structures connecting to the surface in a sparse number of points. This

1 Introduction

Fused filament fabrication (FFF) is a popular technique for turning 3D models into real, tangible objects. A hot filament is melted through a heated nozzle and fuses with the part just below, adding a layer of material to the object being formed. Advantages are the low cost of both printers and filaments, the relative ease of use requiring few manual steps before and after printing, and the wide availability of printers which can be bought from a variety of manufacturers, e.g. Ultimaker, Replicator, 3D Systems, Stratasys.

A major drawback of the process, however, is that filament can only be deposited on top of an existing surface. Therefore, overhangs require a disposable structure to be printed, temporarily supporting the threads of plastic that would otherwise hang in empty space.
Meshmixer 2.5 – released yesterday!

123dapp.com/meshmixer