Executive Summary

Hierarchical Volumetric Object Representations for Digital Fabrication Workflows

by Matt Keeter

Problem

The modern CAD/CAM workflow includes many disparate formats, often based on object boundaries (e.g. NURBS or triangles). These representations complicates digital fabrication, which requires closed volumes. Furthermore, boundary representations requires sophisticated calculations to perform even basic computation solid geometry (e.g. finding the intersection of two parts).

Approach

I propose to develop a full digital fabrication workflow operating on a volumetric representation of solid models. This workflow will begin with hierarchical design tools based on functional representations, which will populate a sparse voxel octree representing a solid object's volume. Next, specialized path planning tools will calculate toolpath trajectories for a variety of machines and processes. Finally, these toolpaths will be executed on existing machines to produce solid objects. This workflow could also be extended to importing volumetric scan data and path-planning for five-axis machining.

Novelty

Many of the individual aspects of this workflow are subjects of academic research; functional representations and octree rendering in particular are well-documented in the literature. However, to my knowledge, the proposed volumetric workflow is the first of its kind. Existing workflows and CAD tools rely on NURBS and triangulated meshes, requiring STLs for 3D printers and SVG/DXF files on 2D cutting tools. The fab modules, developed by my research group, use PNG images as a general-purpose lattice representation; this project will extend that workflow to hierarchical three-dimensional objects.

Evaluation

There are three immediate applications: direct CAD/CAM from volumetric scanning, design and fabrication of digital composites, and 3D printing without triangles. The workflow can also be evaluated on its efficiency, examining how well it handles increasingly sophisticated objects (both with empirical measurements and asymptotic analysis of the core algorithms).

Hierarchical Volumetric Object Representations for Digital Fabrication Workflows

Thesis Proposal for the degree of
Master of Science

by Matthew Keeter

Fall 2012

Thesis Advisor: ______________________________

Dr. Neil Gershenfeld
Professor of Media Arts & Sciences
MIT Center for Bits & Atoms

Thesis Reader: ______________________________

Dr. Neri Oxman
Assistant Professor of Media Arts & Sciences
MIT Media Lab

Thesis Reader: ______________________________

Dr. Erik Demaine
Professor of Computer Science
MIT CSAIL

Contents

Abstract

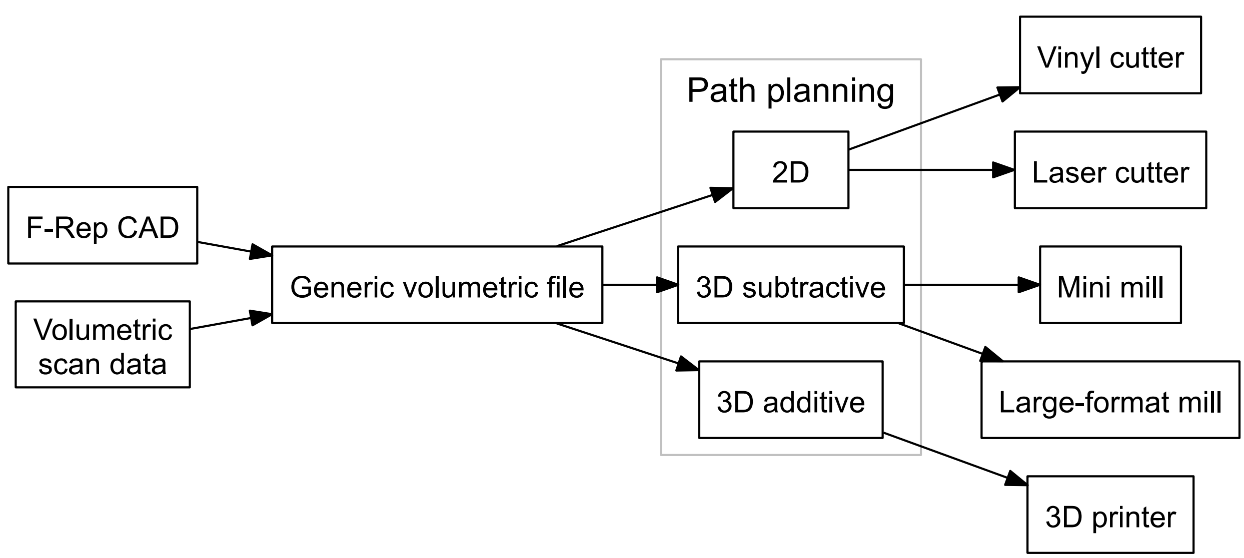
I propose to develop a set of tools for end-to-end volumetric digital fabrication workflows, emphasizing hierarchical representations of the target object at each stage. The workflow will include design tools that generate volumetric representations and toolpath generation for fabrication on a variety of machines. The proposed workflow can be contrasted with modern CAM systems, which operate on boundaries representations of objects (e.g. triangles or NURBS). A volumetric workflow will eliminate typical pain points, guaranteeing closed three-dimensional forms without post-processing. The workflow will maintain a sense of hierarchy at all stages, allowing both users and software to manage design complexity. The functionality of the workflow will be tested on a set of design and fabrication challenges; its performance on increasingly complex objects will also be examined.

Motivation

The current workflow for personal digital fabrication, from design tools to path planning to computer-controlled machines, is dominated by boundary representations, whether triangles or NURBS. These representations define the boundary of an object, rather than the object itself. This is an artifact of historically limited computing power; modern computers are powerful enough to support novel object representations.

Moving from surfaces to volumetric representations has the potential to eliminate pain points in digital fabrication. Meshes have to be made water-tight, without duplicate points or incorrect normals [10]; volumetric data is guaranteed to define a three-dimensional closed form without time-consuming pre-processing.

In my thesis, I propose to extend an existing digital fabrication workflow with volumetric data representation. The proposed workflow will start with design tools that populate sparse octrees, continue through path planning for additive and subtractive processes, and conclude with machine output on a variety of machines.



Volumetric Object Format

In the software used in fab labs around the world, PNG bitmap images are used as a "glue format": they are easy to export from a variety of design tools, and the software includes path-planning algorithms that use PNGs as a general-purpose 2D lattice.

This thesis will extend that workflow with a volumetric file format, suitable for 2D and 3D design and fabrication. This file format will describe a sparse voxel octree in a bounded three-dimensional region, allowing for more efficient storage than full voxel representations [9]. Such a volumetric format also lends itself to manipulating data from volumetric scanners, leading to CAD/CAM workflows from CT scans without triangulation.

An octree-based format is inherently hierarchical, which offers advantages in rendering and manipulation. Octrees offer level-of-detail rendering for free (rendering less detail as objects occupy fewer pixels): recursion down the tree can halt when we've reached a single pixel (regardless of how much detail exists further down the tree). This theme of hierarchy continues through the design workflow.

From preliminary research, no standardized octree format exists; octrees are often used to manipulate point clouds [6], but the raw format for such clouds is a simple list of points [2]. I considered the Additive Manufacturing Format [1], but it only allows for lattice (e.g. full voxel) data, without any hierarchy.

One more general alternative may be the OpenVDB format [3], which allows for sparse volumetric data storage and manipulation. I have not yet decided whether to rely on this library. Though it appears to be a high-quality toolkit, it was released very recently and is not compatible with my primary development environment; it may also introduce unnecessary overhead compared to a simple homebrewed solution.

If I choose not to rely on OpenVDB, one thesis deliverable will be a light-weight binary file format optimized for storing sparse volumetric octrees.

Design Tools

To create volumetric objects, I plan to continue my work on hierarchical design tools based on functional representations. Functional representations, or F-Reps, are a flexible way of representing objects as mathematical expressions [4]; e.g. a sphere of radius one would be represented as $X^2 + Y^2 + Z^2 \leq 1$.

In my work on the fab modules [11], I have developed a solver that uses interval arithmetic and subdivision (as described in [5]) to rapidly evaluate these expressions, creating a greyscale height-map image. Using interval arithmetic dramatically reduces the problem size; instead of evaluating on every voxel in the volume, the solver evaluates an entire region at once, returning either "Full", "Empty", or "Ambiguous". In the ambiguous case, the solver divides the region and recurses.

This type of evaluation, based on recursive subdivision of a volume, naturally lends itself to octree-based representations; nodes in an octree can be set as empty or full based on the value of the F-Rep in that spatial region. Extending these design tools to support octree export will provide a useful entry point into the digital fabrication workflow.

These design tools express the theme of hierarchy in two different ways. First, the tools themselves allow for hierarchical design: one is based on the Python programming language, allowing functions, recursion, loops, etc.; another (written by Prof. Gershenfeld) creates a graph of information flow to represent a real-space object. Second, the F-Reps are parsed into hierarchical evaluation trees; evaluation is performed by recursive tree descent.

Path Planning & Fabrication

To complete a digital fabrication workflow, a design file must be converted into a toolpath for a particular machine. As part of this thesis, I plan to deliver a set of utilities for path planning from a volumetric object.

The precise form of a toolpath varies from machine to machine: for laser and vinyl cutters, the desired toolpath is a 2D cutout around the boundaries of the design, with an offset added to compensate for beam or knife width.

For 2.5D milling (e.g. on a Modela or Shop-Bot mill), there are two different classes of toolpaths: rough and finish cuts. A rough cut toolpath removes material in a series of steps down the z-axis, while a final cut smoothes out along the x and y axes.

Finally, for 3D printing, the toolpath describes an additive process to build up a set of layers.

The fab modules include small software utilities to convert from PNG images to 2D and 2.5D toolpaths. This thesis will deliver similar utilities specialized for operating on octree-based volumetric data.

In general, the following steps are required for toolpath generation: From the original octree, an octree-specialized distance transform must be generated (e.g. as in [7]). Values in this transformed octree correspond to distance from the surface of the original object. Next, the distance-transformed values are thresholded based on the desired tool diameter. Finally, the outline(s) of the resulting shape(s) are traced and saved as toolpaths.

This process is equally applicable for 2D and 2.5D toolpath generation. In 2D, the process as described above suffices; in 2.5D, the process can be repeated to generate each slice of a rough or finish cut. It may also be possible to improve efficiency in 2.5D toolpath generation by taking advantage of octree regularity.

If time permits, path planning will be extended to 3D printing, which has a different set of requirements: Objects must be filled with either solid or semi-solid lattices, and overhangs must be supported by support material.

Finally, the octree could potentially be used for full five-axis path planning. This requires more sophisticated path planning, based on extracting surface normals from the shape.

Evaluation

The goal of this thesis is to create, document, and demonstrate a volumetric workflow for digital fabrication. There are three immediately applicable test cases to demonstrate the system's functionality.

First, my research group is collaborating with an artist who uses CT scanners extensively, scanning artifacts for later reproduction. I plan to scan a delicate figurine provided by the artist and use that data directly in a CAD/CAM workflow. Skipping the triangulation step has the potential to save this studio hundreds of man-hours spent cleaning up meshes.

Second, one of the members of my research group is interested in using my novel CAD tools to design digital composites, sparse space-filling lattices made of individual composite parts.

Third, I am interested in demonstrating 3D printing directly from this volumetric workflow, without triangulation.

The workflow can also be evaluated by how it scales: How does performance degrade with increasingly complex designs? With increasing resolution? To what extent can it be parallelized? Scaling behavior can be evaluated empirically, by examining the speedups with and without to various optimizations; it could also be approached from a theoretical perspective (e.g. by examining the asymptotic behavior of the core algorithms). The system's performance can also be examined as a function of part rotation: octrees introduce a preferred orientation, so solids will be more or less efficiently stored based on angle.

Time Frame

November 2012 - December 2012	Define volumetric octree format
	F-Rep to volumetric format
December 2012 - January 2013	Visualization tools and editing workflow
February - March 2013	Path planning and machine output
March - April 2013	Workflow examples, demos, and documentation

Related Work

Many of the components in this thesis are subjects of considerable academic and industrial research. Though components of this thesis are well-studied, my literature review has not found any complete workflow similar to the one proposed here.

The Hyperfun project [4] is a F-Rep based design system, but is not designed for real-time CAD work (and appears to be abandoned). More generally, there is a substantial amount of academic work on F-Reps and implicit surfaces, e.g. [5].

Similarly, there is extensive research on fast octree rendering, e.g. [8]. Toolpath generation using octrees is somewhat addressed in [13]; this is a special case of a more general path planning problem, e.g. [14].

Finally, CT scan data was used to inform 3D printing in [12], but that project used full voxel data rather than a hierarchical representation.

Resources

The proposed thesis relies on equipment already owned by the MIT Center for Bits and Atoms. The CBA compute cluster can be used as a number-crunching computational geometry engine, and CBA already owns 2, 3, and 5 axis CNC fabrication machines.

If their production schedule permits, I also plan to use a Form1 printer from FormLabs to demonstrate 3D printing directly from volumetric data.

References

[1] Additive Manufacturing File Format. [Online]. Available: http://amf.wikispaces.com/

[2] The PCD (Point Cloud Data) file format. [Online]. Available: http://pointclouds.org/documentation/tutorials/pcd_file_format.php

[3] D. Animation. (2012) OpenVDB. [Online]. Available: http://www.openvdb.org/

[4] Digital Materialization Group. Hyperfun. [Online]. Available: http://hyperfun.org/

[5] T. Duff, "Interval arithmetic recursive subdivision for implicit functions and constructive solid geometry," in *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '92. New York, NY, USA: ACM, 1992, pp. 131–138.

[6] J. Elseberg, D. Borrmann, and A. Nuchter, "Efficient processing of large 3d point clouds," in *Information, Communication and Automation Technologies (ICAT), 2011 XXIII International Symposium on*, oct. 2011, pp. 1 –7.

[7] D. Jung and K. Gupta, "Octree-based hierarchical distance maps for collision detection," in *Proc. of the IEEE Int. Conf. Robotics and Automation*, 1996, pp. 454–459.

[8] A. Knoll, "A short survey of octree volume rendering techniques," in *Proceedings of 1st IRTG Workshop*, ser. GI Lecture Notes in Informatics, Jun. 2006. [Online]. Available: http://www.cs.utah.edu/~knolla/publications.html

[9] S. Laine and T. Karras, "Efficient sparse voxel octrees," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 8, pp. 1048 –1059, aug. 2011.

[10] J. LaMarche. Prepping blender files for 3d printing. [Online]. Available: http://www.shapeways.com/tutorials/prepping_blender_files_for_3d_printing

[11] MIT Center for Bits and Atoms. (2012) Fab Modules. [Online]. Available: http://kokompe.cba.mit.edu

[12] N. Oxman. Pneuma 2. [Online]. Available: http://web.media.mit.edu/~neri/site/projects/pneuma2/pneuma2.html

[13] Y.-J. Tseng, "Machining of free-form solids using an octree volume decomposition approach," *International Journal of Production Research*, vol. 37, no. 1, pp. 49–72, 1999.

[14] L. Wu and Y. Hori, "Real-time collision-free path planning for robot manipulator based on octree model," in *9th IEEE International Workshop on Advanced Motion Control*, 2006.