DOI:10.1145/3344808

OpenFab: A Programmable Pipeline for Multimaterial Fabrication

By Kiril Vidimče, Szu-Po Wang, Jonathan Ragan-Kelley, and Wojciech Matusik

Abstract

3D printing hardware is rapidly scaling up to output continuous mixtures of multiple materials at increasing resolution over ever larger print volumes. This poses an enormous computational challenge: large high-resolution prints comprise trillions of voxels and petabytes of data, and modeling and describing the input with spatially varying material mixtures at this scale are simply challenging. Existing 3D printing software is insufficient; in particular, most software is designed to support only a few million primitives, with discrete material choices per object. We present OpenFab, a programmable pipeline for synthesis of multimaterial 3D printed objects that is inspired by RenderMan and modern GPU pipelines. The pipeline supports procedural evaluation of geometric detail and material composition, using shader-like *fablets*, allowing models to be specified easily and efficiently. The pipeline is implemented in a streaming fashion: only a small fraction of the final volume is stored in memory, and output is fed to the printer with a little startup delay. We demonstrate it on a variety of multimaterial objects.

1. INTRODUCTION

State-of-the-art 3D printing hardware is capable of mixing many materials at up to 100s of dots per inch resolution, using technologies such as photopolymer phase-change inkjet technology. Each layer of the model is ultimately fed to the printer as a full-resolution bitmap where each "pixel" specifies a single material and all layers together define on the order of 10^8 voxels per cubic inch. This poses an enormous computational challenge as the resulting data is far too large to directly precompute and store; a single cubic foot at this resolution requires at least 10^{11} voxels and terabytes of storage. Even for small objects, the computation, memory, and storage demands are large.

At the same time, it is challenging for users to specify continuous multimaterial mixtures at high resolution. Current printer software is designed to process polygon meshes with a single material per object. This makes it impossible to provide a continuous gradation between multiple materials, an important capability of the underlying printer hardware that is essential to many advanced multimaterial applications (e.g., Wang et al.[20]). Similarly, there is no support for decoupling material from geometry definition and thus no ability to specify material templates that can be reused (e.g., repeating a pattern that defines a composite material, or defining a procedural gradation for functionally graded materials).

We think the right way to drive multimaterial 3D printers is a programmable synthesis pipeline, akin to the rendering pipeline. Instead of a static mesh per piece of material, OpenFab describes a procedural method to synthesize the final voxels of material at full printer resolution on demand. This provides efficient storage and communication, as well as resolution independence for different hardware and output contexts. It also decouples material definition from geometry. A domain-specific language and pipeline features specific to 3D printing make it much easier for users to specify many types of procedurally printed output than they could by writing standalone programs for every different material or fabrication application.

The OpenFab pipeline offers an expressive programming model for procedurally specifying the geometry and material of printable objects. A scene graph describes geometry and attributes, although *fablets* procedurally modify the geometry and define the material composition much like shaders in the rendering pipeline. Fablets are written in a domain-specific language (OpenFL) and provide a flexible toolset that supports many common material specification tasks.

We also propose a scalable architecture for implementing the OpenFab pipeline. As the total computational cost is large and it is impossible to fit the entire output volume into memory, the pipeline is designed to progressively stream output to the printer with a minimal up-front precomputation and with only a small slab of the volume kept in memory at any one time. An OpenFL compiler analyzes and transforms the procedural computation described by the fablets as needed for efficient implementation in the fabrication pipeline.

We evaluate the system on a variety of multimaterial 3D objects that have been specified and computed using our pipeline. We discuss how our system can be used to easily describe metamaterials, graded materials, and objects that contain materials with varied appearance and deformation properties. We print a number of results using a commercial multimaterial 3D printer and evaluate the performance of our prototype implementation.

2. BACKGROUND AND RELATED WORK

"3D printing" is an umbrella term for a variety of additive manufacturing processes where parts are built up from constituent materials, typically one layer at a time, in an incremental fashion. Processes vary by what kind of materials

The original version of this paper appeared in *ACM Transactions on Graphics 32*, 4 (July 2013).