Drowning in Triangle Soup: The quest for a better 3-D printing file format

File formats for additive manufacturing are lagging behind the capabilities of 3-D printing technology itself, and no one is doing anything about it.

By Jesse Louis-Rosenberg

DOI: 10.1145/2893503

At Nervous System—a design studio that combines art, science, and technology—we create designs that push the limits of 3-D printing hardware and software.[1] By combining generative simulations, design, and 3-D printing we create complex, customized products (see Figure 1). Through our work developing software systems and designs for digital fabrication, we've repeatedly run up against the limitations of how geometry is defined for printing, which has caused us to start to think about and develop volumetric alternatives.

3-D printing has experienced an explosion in popularity in the last five years. Not merely in the public eye, with the advent of home printers, but also in industry and scientific research. Researchers, who had been toiling in obscurity for 20 years, suddenly have grants and corporate backing. However, commercially available technology has advanced surprisingly little in the 30 years since the technology was invented. Perhaps the area that has developed the least is software and how we represent geometry for 3-D printing.

THE STUPID TRIANGLE LIST

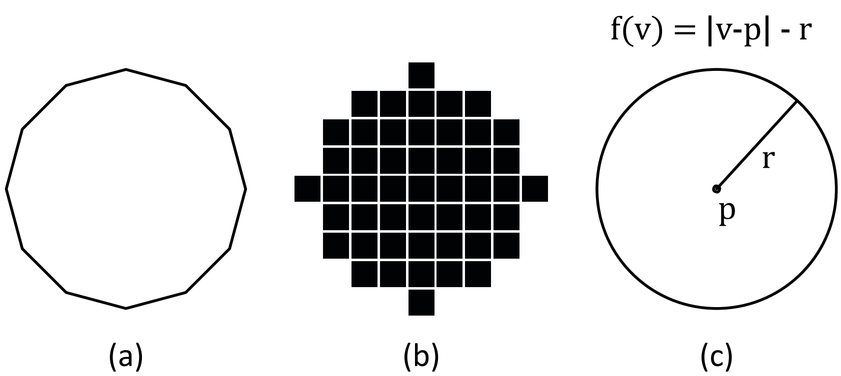
Chuck Hull invented stereolithography in 1986, and at the same time developed the first file format for 3-D printing, STL, which remains the primary format used today [1]. An STL file is quite simple; it is a list of triangles. The triangles are defined by three points represented by 32-bit floats. There are a couple of idiosyncrasies: Each triangle also has a normal direction (which is almost never used because three points already define a normal plane) and two extra miscellaneous bytes. The triangles define the surface of a three-dimensional shape, a representation referred to as a "mesh." There is no information to gauge which triangles are next to which, so people commonly create files that do not define a coherent shape at all. Other common mesh formats define shapes with vertex indices, supplying topological connectivity information and reducing file size. Triangles can be inside out, form non-manifold surfaces, or simply not

1 http://nervo.us

Image by Steve Marsel

Figure 1. The Kinematics Dress by Nervous System is composed of thousands of unique interlocking components. Each dress is 3-D printed as a single folded piece and requires no assembly.

Figure 2. 2-D equivalents of different representation methods for volumes: (a) boundary surface, (b) voxels, and (c) functional representation.



be closed at all. The format is inextensible, providing no way to include additional semantic information.

The main problem is the core idea of a boundary surface representation, which defines a volume by its surrounding surface. This type of representation causes all sorts of headaches: the aforementioned open surfaces, overlapping shapes, gnarly edge cases, etc. There is a more critical concern, 3-D printers create volumes but our file format only defines surfaces. What happens inside the surface is just as important as the exterior skin. An emerging area of 3-D printing is multi-material printing. By specifying materials with different properties in complex configurations, we can create meta-materials with properties that could not exist before. Let us take a relatively simple example from nature; a squid's mouth has a gradation of stiffness that allows its hard and sharp beak to attach to its gelatinous body without tearing it apart [2]. Currently, printing gradients of material requires either laborious workarounds or direct access to the firmware or special APIs. To advance 3-D printing technology it will be necessary to have a universal format for specifying material properties that can change throughout a volume.

ALTERNATIVES: FUNCTIONAL REPRESENTATION AND VOXELS

There are two primary candidates for volumetric representations that could replace meshes: functional representation (f-reps) and voxels (see Figure 2). F-reps are essentially implicit surface equations: Define a function that takes a point and returns whether or not it is inside a volume. In some ways, this is the most basic and fundamental way to define a volume mathematically. Extending this idea slightly, we can have a function that returns color, density, Young's modulus, or other material properties. F-reps are incredibly expressive and can create complex shapes with more compression than any other method. However, that expressivity is also their greatest flaw. Because you can describe anything that you can define mathematically, it is difficult to create a cross-platform standard. How would existing CAD packages, which use boundary surface representations, read or write such a format? You can create a f-rep of a boundary surface, but that would essentially be a wrapper for the boundary representation itself and no better. Truly utilizing an f-rep format requires an exclusively f-rep workflow, which creates a barrier to getting CAD software developers and 3-D printer manufacturers on board.

Voxels are the extension of pixels to 3-D. We unitize space to a certain resolution and then create a rectilinear grid of values that contain material

Working with voxels gives us the potential to take advantage of new 3-D printing technology and frees us from many of the headaches of traditional boundary representations.

Figure 1 Photos by Steve Marsel

properties. The primary advantage of this method is its simplicity. It is easy to understand, read, and write and there is no ambiguity as to what material occupies a position in space.

The problem with voxels is size. Let's assume you want to represent your voxel space at the resolution of your printer, around 50 microns is reasonably fine, and each voxel takes up 1 byte, which is probably an underestimate for a complex material. That would mean even a relatively small print area of a 128mm cube would take up 15.625GB. Even if you compress the file on disk, it quickly becomes a problem to work within the memory of most desktop computers, and there are many printers with higher resolutions and larger build volumes than this example. For this reason, people have created hierarchical voxel formats; the most common being an oct-tree, a cascade of smaller and smaller cubes where each level gets divided into eight equally-sized smaller cubes. In practice, you do not need to specify the material at the printer's resolution everywhere. Most of the space will either be empty or uniform with changes only occurring in a fraction of the volume. A hierarchical voxel format only uses the finest resolution where necessary. This compression comes at the expense of some of the simplicity that we value in voxels, hindering the creation of a standard that is easy to implement and performant enough for diverse applications. Additionally, some 3-D printer manufacturers are concerned with the possibility of someone wanting to specify every single point of a printer's volume with a different material. With a hierarchical format, that would be even larger than simple vanilla voxels.

WHAT IS BEING DONE

People in the industry have known for a long time that STL is an insufficient file format. However, no one has taken the lead in developing new standards. 3-D printer manufacturers do not want to make their machines run on files that no CAD software can create, and software developers do not want to produce files that no one can use. Two consortiums, AMF and 3MF, have recently formed to bring together partners from different parts of the industry to draft new standards [3]. However, both of these have resulted in what is basically an XML wrapper around an STL with some extra features for specifying materials. My personal loathing of XML-based formats aside, this does little to advance the technology. In theory, it is extensible and can eventually incorporate other data types, like voxels, but this shifts the same issues from file formats now to standards compliance in the future.

There are companies incorporating voxel formats currently. Autodesk's Ember printer is the first commercially available printer that can directly print voxel data. The format is a bit of a hack though. It is a zip file with a list of PNG images for each layer and a JSON file with print settings. Shapeways, a 3-D printing service bureau, has begun accepting similar files—zipped PNG stacks—which still have to be converted to STLs to interface with their printers. While it is encouraging to see companies adopt voxel formats, it is doubtful a solution like a zip file of images could become standard. First, it is not extensible. What happens when there is a five-channel printer, like RGBA plus hardness? Second, it does not encourage adoption in software development. Reading and writing such a format feels like a strange workaround that most CAD software is unlikely to implement. As stated previously, working with the raw, uncompressed voxels becomes imprac-

Figure 3. (a) A diamond engagement ring 3-D printed in wax and cast in white gold. (b). A 3-D lamp with an LED fixture.

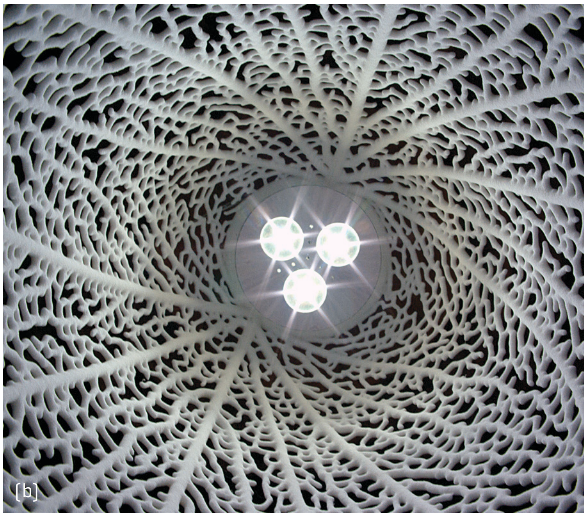


Figure 4. A 3-D printed sculpture defined by 1.67M lines of varying thickness.

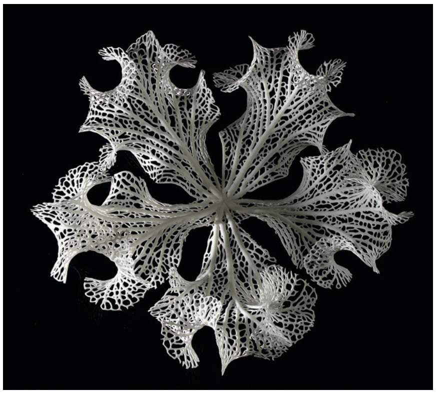


Figure 5. New Balance's 3D-printed midsoles with customized cushioning.

Figure 5 Photo Courtesy of New Balance

tical without significant advances in system memory.

LOOKING OUTWARD FOR INSPIRATION

New ideas in volumetric representations are emerging out of other industries like game development and visual effects. The sandbox game "Dreams" by Media Molecule allows users to create their own 3-D environments and characters based on volumetric tools [4]. Their system stores models as a list of constructive solid geometry (CSG) operations assembling primitive shapes with some f-rep-like effects. It then turns those operations into a signed distance function in real time.

OpenVDB is a library for voxel data developed by Dreamworks for the film industry [5]. VDB is a hierarchical voxel format based on B+ trees, which are often used in databases. It overcomes many of the shortcomings of traditional hierarchical voxels formats, allowing for arbitrary bounds and fast random and sequential access. Though the format is complex, the OpenVDB library provides a cross-platform, open-source implementation with functionality that makes it easy to incorporate into existing workflows, including conversion to and from meshes. It is extensible and allows for embedding metadata. While it is missing some functionality that might be desired for 3-D printing, it provides a solid foundation to develop from, or at least a strong example of, what other industries are doing.

UTILITY OF VOXELS

At Nervous System, we have been developing in-house tools to work with voxel data based on the OpenVDB library. Voxels not only provide a truly volumetric representation for 3-D printing, but afford many other advantages compared to boundary representations. This tool has quickly become integral to the projects we work on. Voxels allow for simple and fast Boolean operations to combine multiple shapes. These operations are often the Achilles' heel of complex mesh shapes, and almost no CAD software can do them robustly or quickly. In voxel space, a union is simply an OR operation on each element of an array. We use this to add hardware attachments to the organic shapes we design, like inserting gemstone settings into an engagement ring or fixtures into a lamp (see Figure 3).

By specifying materials with different properties in complex configurations, we can create meta-materials with properties that could not exist before.

Similarly, voxels provide an efficient format for performing CSG operations. This is a way of defining complex shapes by combining many simple, easily described shapes. Often times, a structure we design will be described as a dense network of lines with varying thicknesses. Turning this into a 3-D shape involves combining hundreds of thousands of cylinders. This process used to take us several minutes, but now using VDB only requires a few seconds, allowing for rapid design iteration (see Figure 4).

The OpenVDB library also provides functionality for level-set operations. Level sets are a generalization of voxels where each data point contains the signed distance to the surface, so negative values are inside and positive are outside. This allows for defining a shape with much higher accuracy than the resolution of the voxel space. We can use linear interpolation to find the zero crossings of this field to define our surface. It also makes doing surface offsets easy, another operation that is impractical for complex shapes using boundary representations. One simply adds the offset distance to each value of the level set. We have been using this to create thin molds of some of our designs.

Hierarchical voxel spaces can provide utilities outside of direct geometric modeling as well. They form the basis of adaptive numerical simulations for fluids, crystal growth, finite elements, etc. It also serves as a structure for performing intersection tests, finding if any point in space is inside our surface. With a boundary surface, we have to make a separate data structure to accelerate these queries—like an AABB tree, which is slow and error prone. With VDB, the data itself is the acceleration structure. All of these techniques have helped us develop 3-D printed midsoles for New Balance sneakers, which have a cellular structure adapted to a user's running data (see Figure 5).

Working with voxels gives us the potential to take advantage of new 3-D printing technology and frees us from many of the headaches of traditional boundary representations. We use it to perform demanding geometric operations that would be impossible with existing CAD software. I do not know if OpenVDB is the correct solution for a 3-D printing file format moving forward, but for me it is shows a format should not just solve technical issues or appease many parties. It should be a tool that inspires people to use 3-D printing to its full potential.

References

[1] Hull, C.W. Apparatus for Production of Three-Dimensional Objects by Stereolithography. Patent US4575330. 11 Mar. 1986.

[2] Miserez, A., Schneberk, T., Sun, C., Zok, F. W. and Waite, J. H. The transition from stiff to compliant materials in squid beaks. *Science* 319, 5871 (2008), 1816-19.

[3] ASTM ISO / ASTM52915-13, Standard Specification for Additive Manufacturing File Format (AMF) Version 1.1, ASTM International, West Conshohocken, PA, 2013. *http://www.astm.org/*

[4] Evans, A. Learning from failure: A survey of promising, unconventional and mostly abandoned renderers for 'Dreams PS4', a geometrically dense, painterly UGC game. SIGGRAPH 2015. Advances in Real-Time Rendering, Part II. 2015.

[5] Museth, K. VDB: High-resolution sparse volumes with dynamic topology. *ACM Transactions of Graphics (TOG)* 32, 3 (2013), 27.

Biography

Jesse Louis-Rosenberg is the co-founder and chief science officer at Nervous System. He studied math at MIT and worked at Gehry Technologies in building design automation. Founded in 2007, Nervous System has pioneered the application of new technologies in design. Their work has been featured in numerous publications including *WIRED* and the *New York Times*, and is part of the permanent collection of Museum of Modern Art and the Cooper-Hewitt Smithsonian Design Museum.

1528-4972/16/03 $15.00