Design in the age of **3-D printing**

BY HOD LIPSON

Designers of all stripes can now create their own sneakers with online tools.

A few months ago, I asked some students to design a pencil holder to be fabricated on a 3-D printer. After explaining the virtually unlimited designs capable of being produced by the printer, I asked them to come back the next time with some wild looking pencil holders. "Think outside the box—literally," I said. I was hoping to see some avant-garde pencil holders. Next class, the students came back with pencil holders; some more innovative than others, but none nearly as wild as I had hoped for. What went wrong?

As additive manufacturing technologies such as 3-D printing and rapid prototyping become increasingly capable, traditional barriers of resources and skill for manufacturing are all but vanishing. The limit is now only our imagination, but our imagination is, unfortunately, limited.

Over recent years I've seen how designers faced with the blank page of their computer-aided design software and the unlimited capability of a 3-D printer design nothing more than a plain object with a few simple features, forgoing the freedom of creation afforded by 3-D printing. This myopia may have evolved out of years of observing mass-produced objects made subject to traditional manufacturing constraints. But it may also be due to the design thinking imposed by conventional CAD software and the lack of new design tools

Hod Lipson is an associate professor in the Sibley School of Mechanical and Aerospace Engineering at Cornell University and also in the university's Faculty of Computing and Information Science. His latest book, written with Melba Kurman, is Fabricated: The New World of 3D Printing, *to be published by John Wiley Press in November.*

NETFABB

Thinking Outside

the CAD Box

SCULPTEO

If you can draw it—as with this Sculpture Ranger—you can create it, with the help of online design tools and 3-D printers.

that take advantage of the vast new design space opened by 3-D printing capabilities.

For the last four decades, CAD tools have played an increasingly critical role in the product design process and in shaping our design thinking. To a large extent, however, CAD tools have remained relatively unchanged in their design philosophy. Interfaces have become more user friendly, geometric manipulations have become faster, and video tutorials replace thick manuals, but conceptually CAD software remains today a 3-D drawing board that records our intentions but offers little insight or ideas of its own and offers limited access to the vast new space of geometric complexity.

While it's clear that the classical CAD paradigm will remain dominant for the foreseeable future, new paradigms for design tools are beginning to emerge. Many of these ideas have existed but have been dormant for decades, awaiting a manufacturing process that can carry them from theory to practice.

Complexity Is Free

For the first time in human history, making more complex objects is not more difficult, expensive, or time-consuming than making simpler objects. Printing a block with holes, notches, and rounded edges takes no more resources or skill than printing a plain solid block. Manufacturing an elaborate object can take no more effort than printing a paperweight.

The diminishing cost of manufacturing complexity is a marked departure from most of human history, where making more complex objects required substantially more investment in time, equipment, energy, and labor. Barriers of resource and skill that traditionally prevented many ideas from being realized are gradually being eliminated. The marginal cost of adding an additional design feature is near zero, potentially triggering nothing short of a design revolution similar to the first industrial revolution, which was triggered when the cost of power was sharply reduced.

The elimination of resources and skills factors from manufacturing is leading to two profound changes in design: First, it is unleashing a new design space, and second, it is unleashing a new kind of designer. Both these trends will change the shape of CAD.

Three-dimensional printers are giving designers unprecedented control over the shape and composition of matter. High-end 3-D printers today can combine multiple materials into arbitrary patterns at a resolution nearing ten micrometers, leading to the ability to create geometry with fidelity and complexity that rivals that of the natural world.

As printers become more capable, consumers and designers will expect CAD systems to keep pace.

Design tools' philosophy has always been driven and constrained by the underlying way they represent shape—how geometry is encoded and stored in computer memory. A particular encoding will encourage or discourage certain types of manipulations. The inability of conventional CAD to keep pace with the complexity afforded by 3-D printers may be due in part to inappropriate internal representations.

We can think of geometric representations as languages that describe shape; some of the languages are lower level, some are higher level. Like spoken words, higher-level languages can often describe complex notions more compactly. Higher-level representations can be translated into lower level, though the other way round is not necessarily easy or even possible. For example, every C++ program can be translated into machine code, but not every machine code program can be translated into C++.

Most CAD systems today represent geometry explicitly, similar to the way traditional blueprints did for centuries. Boundary representations, or B-Reps, describe geometry by surfaces and computational solid geometry, or CSG, representations describe volume using Boolean combinations of primitive shapes.

Voxel-based representations, though relatively uncommon, describe geometry using an array of 3-D pixels that occupy its volume. Parametric representations revolutionized the CAD industry in the early 1990s when they introduced CSG models with adjustable dimensions.

While robust and straightforward, these traditional encodings do not scale well to describe complex structures. Anyone trying to model a lattice structure, for a chair perhaps, on a conventional CAD system would quickly run into limits of memory and computational power.

In trying to address the gap between today's CAD and 3-D printing capabilities, we can try to learn from nature. Biology employs fundamentally different methods for encoding shape and function. The genome encodes how the geometry developed, not the final geometry directly.

The periodic, repeating structure of the chair's lattice immediately suggests that a procedural way of describing the geometry might be much more appropriate.

Specifying geometry as a construction process or a geometric program makes sense: Repeated structures, semi-periodic structures that vary with location, and hierarchical structures composed of smaller substructures bear a close analogy with structured programming languages. Using such a representation, specifying a lattice composed of one million repeating rectangular cells arranged in a lattice would require little more memory than a block made of only ten units.

Generative representations are even more indirect. These encodings specify how a design develops from an initial seed, according to a provided set of rules. For example, imagine two rules: One rule specifies that any "A" is replaced by "BA," and a second rule specifies that any "B" is replaced by "A." Now, throw in an "A" and watch what happens when the rules are applied simultaneously: The "A" turns into "BA"; the "BA" turns into "ABA," which then turns into "BAABA," then "ABABAABA" and so forth. Before long, we have a pattern that looks complex, but really is just a result of the two simple rules and the seed "A."

Replace the symbols "A" and "B" with geometric shapes, such as a sphere and a cube, and you have what is known as a shape grammar. Simple rules allow the generation and exploration of complex, organic-looking geometries.

Both the direct and indirect representations are still rather ballistic: they specify a geometry in an "open loop," without regards to its context. Taking a further cue from biology, future CAD systems will employ feedback-based design specifications that unfold in reaction to their environment. Such "dynamical blueprints" will unfold into different shapes depending on their environments.

Consider the blueprint governing the shape of a

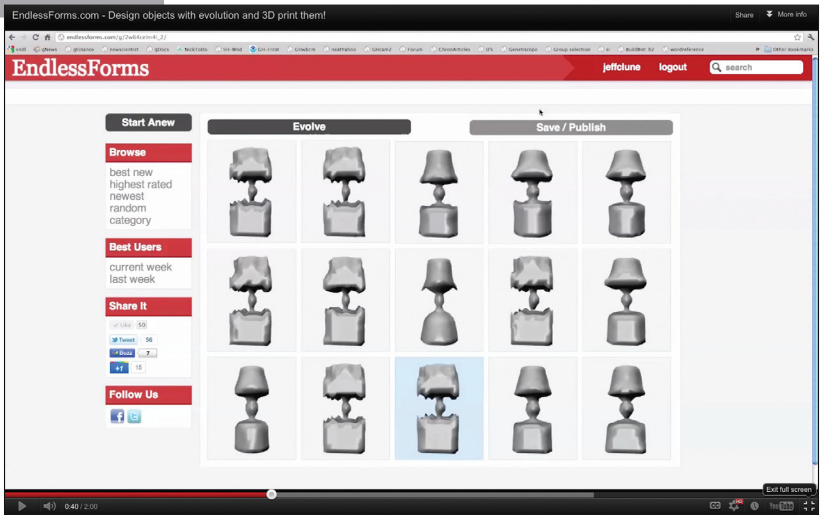
Now, design and print tools allow for organic design such as this Hyphae lamp from Nervous System, based on veins in leaves.

NERVOUS SYSTEM

This Venetian pot was created with help from My3DScanner, which produce 3-D models from digital photographs. The models can then be printed in 3-D.

MY3DSCANNER

plant—both its above-ground portion and its roots. The plant's upper portion geometry is determined by a set of generative rules that are governed by sensors, such as external sunlight sensors and internal structural stress sensors. The plant will grow into a different shape depending on lighting conditions and the amount of mechanical stress it can support, reaching an equilibrium that is specific to its context. Similarly, the roots will develop according to a different set of rules that are governed by moisture content and osmotic pressure.



At EndlessForms.com you choose an object from the website that you can further evolve into an object you like. Or, you can develop an object from scratch.

How can we apply this to product design? Imagine a dynamical blueprint for a lampshade, specified using a set of rules that develop the geometry to produce even illumination in a room. Place the lampshade in a corner of a simulated room with a window and the design will develop into one geometry; place it in a different room next to another light fixture and the same dynamic blueprint will manifest into a different final shape. As the complexity of these sensor-based rules evolves, we essentially specify the target form not by its target geometry, but by its target behavior.

New Designers, New Tools

The growing accessibility of personal manufacturing tools, such as 3-D printers, is democratizing design and enabling new types of designers. Many of these new designers are unlikely to have the formal engineering training for which traditional CAD tools were developed.

We are increasingly seeing artists, casual users, and even children interested in computer-assisted design. This new and growing community—much larger than the original CAD community—offers new opportunities to develop design tools, interfaces, and paradigms.

Some of these developments may pay off for engineers, as well, by making it possible to focus more on the design and less on the software and by empowering new generations of designers.

One way to create design tools for casual users is based on natural user interfaces. In contrast to graphical interfaces, natural interfaces rely on more fluent modes of communication such as gestures, sketching, and speech. For example, sketch-based interfaces allow users to progressively sketch 3-D objects into existence, without worrying about a particular construction sequence.

When combined with physics-based interaction using real-time simulation, natural interfaces can allow users to be even more productive. They can shape virtual clay with their hands or interact with fluid flow to create streamlined objects. Realistic physical simulation can provide instant performance feedback that allows untrained users to find optimal solutions faster.

Increasing numbers of users are opting to create designs from data collected, rather than generating objects from scratch. Users scan existing shapes they find around them—natural objects or synthetic objects—for which a CAD model doesn't exist.

A variety of new scanning devices, from the Kinect for Xbox to video capture, are capable of producing detailed color 3-D models. Other nonvisual data sources may also be used in the future, such as simulations or sensor networks.

Since scanners generate surface mesh models, the lowest level of geometric representations, today's CAD systems struggle to process them effectively. Yet users will expect to be able to perform increasingly complex manipulations on scanned objects—from simple scaling and stretching to merging objects, interpolating between objects, and performing smart parametric adjustments that are feature sensitive.

Perhaps one of the most interesting challenges is creating systems suitable for designers who have almost no interest in traditional modeling, but who have a sense of aesthetics and know what they want. Catering to such designers involves new artificial intelligence techniques to infer design rationally from a relatively brief user interaction. People often find it easier to say what they like and don't like, rather than designing from scratch. For example, it's easier to vet floor plans of potential houses than to explicitly list requirements. A good architect, however, can often design a suitable house based on this limited feedback.

One way to generate designs from brief user interaction is the so-called blind watchmaker process, which is based on interactive evolution. Here, the system presents the user with a series of randomly shaped objects. The user then selects which of these shapes she likes. Based on this sparse feedback, the system eliminates undesired objects and produces new objects that are slight variations of the selected shapes. Over several iterations, forms begin to emerge that match the intended goals.

Developing an interactive evolution system like this, however, requires sophisticated back-end algorithms that can represent arbitrary geometries and automatically parameterize features in order to create meaningful variations. For example, when a user is generating lampshade forms, the software automatically creates variations of lampshades—some circular, some rectangular, some smooth, and some shaggy—without knowing anything about lampshades, smoothness, or other relevant dimensions of lampshade design.

Perhaps the ultimate indirect form of design tool is what we call the matter compiler. A matter compiler will compile high-level design requirements and constraints into an optimized print file specific to the 3-D printer capabilities and materials at hand.

Consider a case where a user wants to design a bracket for holding up a shelf on a wall. While the user may have little engineering background, he or she could specify the volume the bracket needs to fit within and the load it needs to carry. Armed with these requirements and constraints, as well as information about the materials available in the user's 3-D printer, the software will design the optimal bracket.

If the user has access to a metal printer, the product might look very different than it would if the user had access only to a plastic printer. An object compiled for a multimaterial printer might look different from one compiled for a single-material printer.

Once the design is complete, the user might realize that the bracket also needs to accommodate space for a wall pipe. Adding this additional constraint and recompiling the design will lead to a modified design, and so forth, much like the iterative nature of software development.

FabApps

We are also seeing a shift from general-purpose CAD tools to specific design tools for niche applications. At the far end of the spectrum we see an emergence of the FabApp—an application targeted and making one and only one type of object. Such applications offer the user the freedom to design the object, yet contain all the know-how to guarantee that the object is successful.

Shapeways.com, Sculpteo.com, and other companies' websites offer creation tools for making customized jewelry and housewares using dedicated FabApps.

The trend towards niche design tools and playful interfaces is leading towards what can only be described as the "gamification" of CAD. If you had told any engineers just a decade ago that the CAD software they studied in college would some day become a popular game, they would have had a hard time taking you seriously. But the trend is now clear.

MineCraft, a voxel-based online 3-D design tool (www.minecraft.net) is accumulating users rapidly. It's developer, Markus Persson, calls it a game about building anything the player can imagine.

With over 30 million registered users and six million paying customers, MineCraft may possibly be one of the most widely used CAD software systems and certainly is the fastest growing CAD market. If this addictive design paradigm and collaborative design environment could be harnessed by traditional CAD software, we may be able to create a truly new generation of designers.

The combination of new geometric representations, new design paradigms, and new interfaces leads to new challenges and opportunities in the CAD field as never before. Good design tools are often the hidden enabler of technological innovation.

Balancing the existing performance engine with a new paradigm shift is a tough act, but if there was ever a time to innovate within the realm of CAD it is today. ■

MINECRAFT

Minecraft players build 3-D constructions out of textured cubes. With six million customers, the game is one of the fastest growing CAD markets.