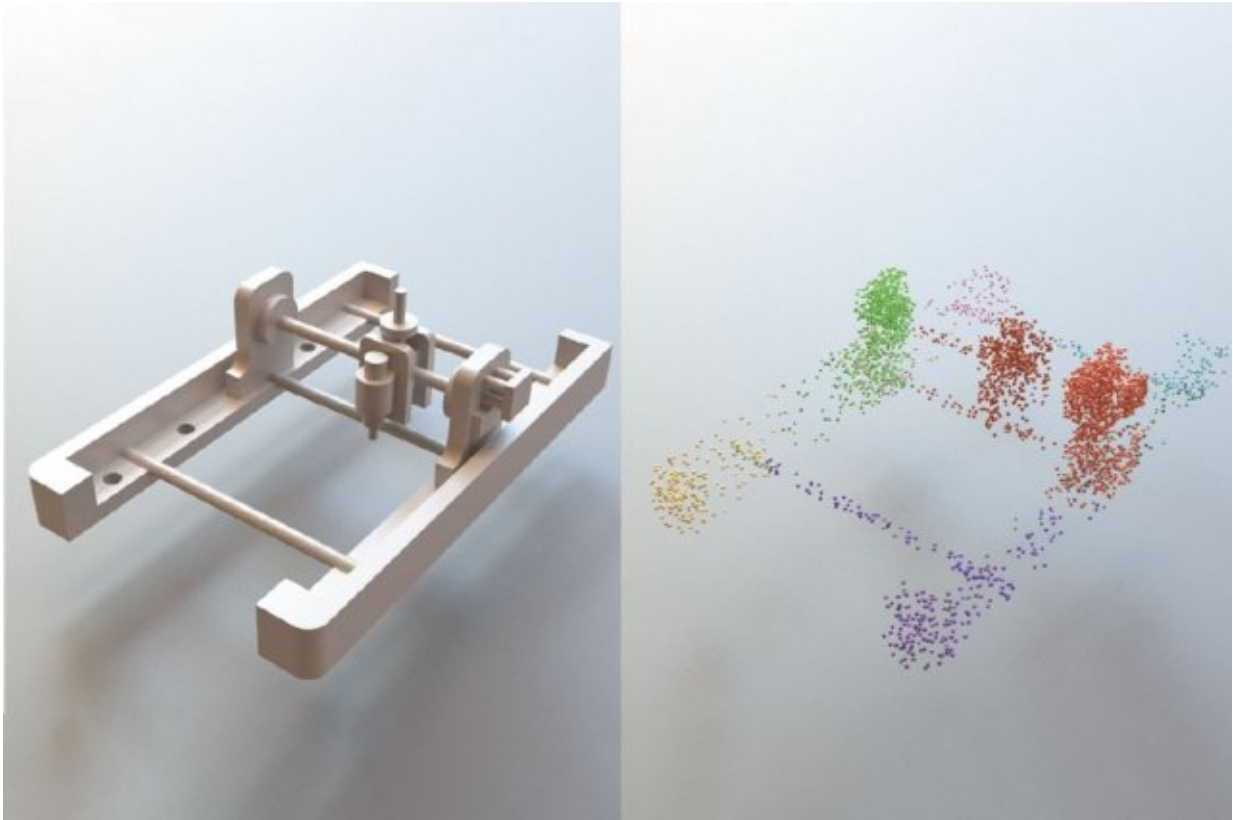


# Customizing computer-aided design

January 3 2019, by Rob Matheson

---



MIT researchers have devised a technique that “reverse engineers” complex 3-D computer-aided design (CAD) models — breaking them down into the many individual shapes they’re made of — to make them far easier for users to customize for manufacturing and 3-D printing applications. Credit: Massachusetts Institute of Technology

MIT researchers have devised a technique that "reverse engineers"

complex 3-D computer-aided design (CAD) models, making them far easier for users to customize for manufacturing and 3-D printing applications.

Nearly all commercial products start as a CAD file, a 2-D or 3-D [model](#) with the product's design specifications. One method that's widely used to represent today's 3-D models is constructive solid geometry (CSG), a technique where numerous basic shapes, or "primitives," with a few adjustable parameters can be assembled in various ways to form a single object. When finalized, the compiled digital object is converted to a mesh of 3-D triangles that defines the object's shape. These meshes are used as input for many applications, including 3-D printing and virtual simulation.

Customizing that mesh, however, is no easy task. For example, adjusting the radius in one portion of the object requires individually tweaking the vertices and edges of each affected triangle. With complex models comprising thousands of triangles, customization becomes daunting and time consuming. Traditional techniques to convert triangle meshes back into shapes don't scale well to complex models or work accurately on low-resolution, noisy files.

In a paper presented at the recent AMC SIGGRAPH Asia conference, MIT researchers describe a system that applies a technique called "program synthesis" to break down CAD models into their primitive shapes, such as spheres and cuboids. Program synthesis automatically constructs computer programs based on a set of instructions.

Essentially, to build CAD models, designers assemble individual shapes into a final object; the researchers' method does the reverse, disassembling the CAD models into individual shapes that can be edited. As input, the system takes a 3-D triangle mesh and first determines the individual shapes that make it up. Program synthesis crawls through the

shapes, trying to figure out how the shapes were put together and assembled into the final model. In doing so, it breaks down the mesh into a tree of nodes that represent the primitive shapes and other nodes detailing the steps for how those shapes fit together. The final shapes contain editable parameters for users to tweak that can be reuploaded to the mesh.

## Foundational shapes

The researchers built a dataset of 50 3-D CAD models of varying complexity. In experiments, the researchers demonstrated their system could reverse engineer CAD files composed of up to 100 primitive shapes. Simpler models can be broken down in around a minute. While run times can be quick, the key advantage of the system is its ability to distill very complex models into simple, foundational shapes, the researchers say.

"At a high level, the problem is reverse engineering a triangle mesh into a simple tree," says Tao Du, a Ph.D. student in the Computational Fabrication group of MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL). "Ideally, if you want to customize an object, it would be best to have access to the original shapes—what their dimensions are and how they're combined. But once you combine everything into a triangle mesh, you have nothing but a list of triangles to work with, and that information is lost. Once we recover the metadata, it's easier for other people to modify designs."

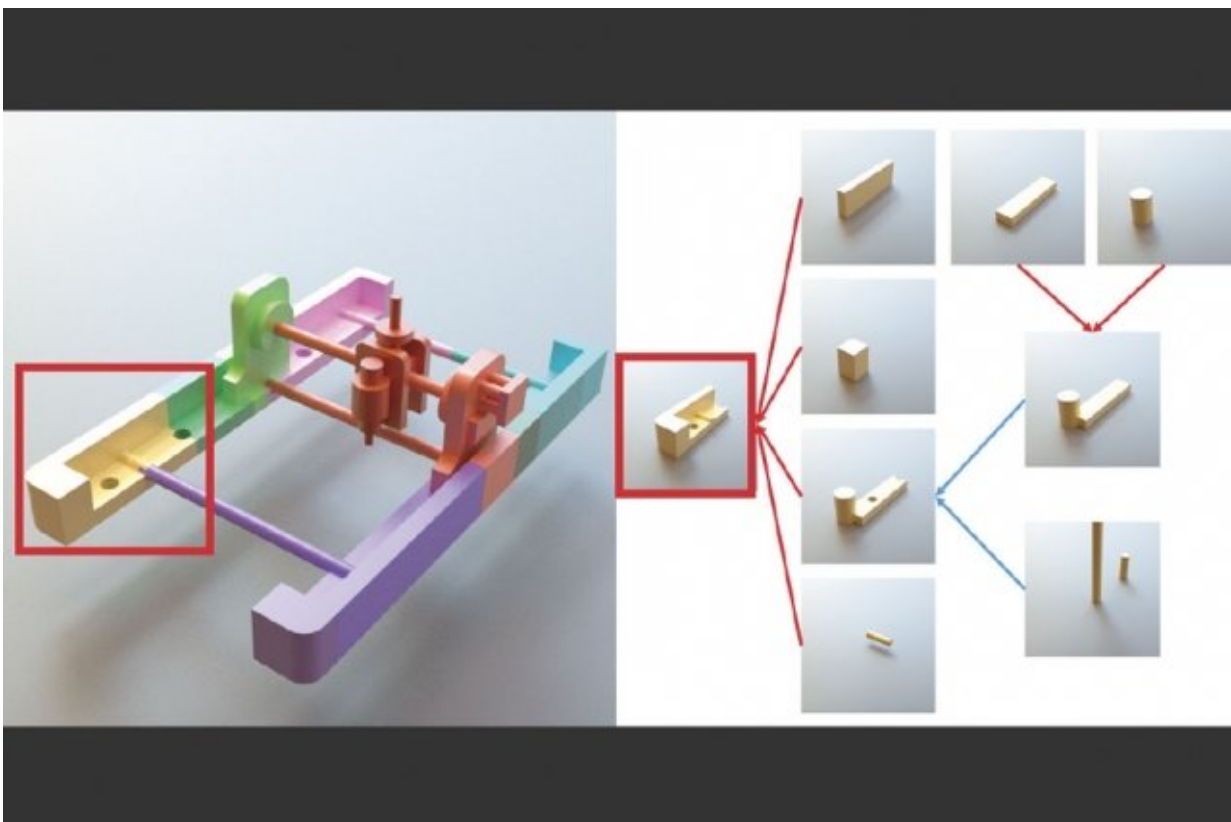
The process could be useful in manufacturing or when combined with 3-D printing software, Du says. This is especially important in the age of design sharing, where amateur 3-D-printer users upload 3-D-print models to websites for online communities to download and modify. Uploads are mostly triangle meshes, because meshes are far more universally accepted across platforms than the original CSG-based CAD

files.

"We have tons of mesh models, but comparatively few CAD files behind them," Du says. "If users want to reproduce the design at home and customize it a little, then this technique could be useful."

## Trees and triangles

Program synthesis automatically finds candidate computer programs given a specific "grammar," meaning the structure it must work within, such as trees, and mathematical specifications. Using those constraints, program synthesis works its way back and fills in the blanks to construct an algorithm that satisfies those specifications, given new input. The technique is used, for example, for simple components of software engineering.



MIT researchers have devised a technique that “reverse engineers” complex 3-D computer-aided design (CAD) models — breaking them down into the many individual shapes they’re made of — to make them far easier for users to customize for manufacturing and 3-D printing applications. Credit: Massachusetts Institute of Technology

In the researchers' work, the grammar is CSG, represented as trees. Each final node (with no branching nodes) represents a primitive shape with clearly defined parameters, and intermediate nodes represent basic ways the shapes converge and relate.

The researchers developed a method that lets program synthesis scan an entire 3-D mesh and, essentially, think of each possible CSG tree it could create as a new candidate program.

After the system receives an input mesh, a preprocessing step detects the possible locations, orientations, and parameters of all primitive shapes. This process creates a massive point cloud across the surface of the triangle mesh. A special "primitive-detection" algorithm infers from these points the dimensions for each primitive shape that makes up the mesh.

The researchers then sample tons of points in the entire 3-D space and flag them as either inside or outside the mesh. This helps determine how the shapes converge or relate to one another. A simple example is a mesh consisting of two spheres, A and B, merged together. If one sampled point falls inside sphere A, one inside sphere B, and one at the intersection of the two (inside both A and B), it's most likely a union of the two shapes.

Given this information, along with the primitive dimensions, program synthesis could potentially create a CGS tree. But, 3-D meshes of even low complexity would require program synthesis to sample tens of thousands of points. This would create a massive search space that's computationally impractical to handle. "Directly feeding all the samples will choke the program synthesizer," Du says.

To ensure the system worked efficiently, the researchers designed a sampling method that creates several small subsets of point samples across the 3-D space, which is much easier for program synthesis to compute. By sampling these subsets, it creates a new candidate "[program](#)," or CGS tree, that could be considered correct. After numerous iterations—and using techniques to eliminate certain points and trees—the system lands on the correct CGS tree for each [shape](#), with correct intermediate steps and final parameters. Any edited shapes are fed back into the [mesh](#) as the system computationally follows the intermediate steps back to the final object.

Currently, the system only handles four primitive shapes—spheres, cylinders, cuboids, and tori (donut shapes). Next, the researchers aim to increase the complexity of CSG grammar to handle far more shapes and more modifiers outside just Boolean operators.

**More information:** Tao Du et al. InverseCSG, *SIGGRAPH Asia 2018 Technical Papers on - SIGGRAPH Asia '18* (2018). [DOI: 10.1145/3272127.3275006](https://doi.org/10.1145/3272127.3275006)

*This story is republished courtesy of MIT News ([web.mit.edu/newsoffice/](http://web.mit.edu/newsoffice/)), a popular site that covers news about MIT research, innovation and teaching.*

Provided by Massachusetts Institute of Technology

Citation: Customizing computer-aided design (2019, January 3) retrieved 19 April 2024 from <https://techxplore.com/news/2019-01-customizing-computer-aided.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.