

Supplemental Material for “Fab Forms: Customizable Objects for Fabrication”

Maria Shugrina¹

Ariel Shamir²

Wojciech Matusik¹

¹MIT CSAIL

²The Interdisciplinary Center Herzliya

1 Implementation Details

The operation nodes in our system fall into the following categories:

- **Create Shape** (all leaves have this category), e. g. Box
- **Modify Shape**, e. g. Scale
- **Combine Shapes**, e. g. CSG Union
- Conditional
- Loop

where bolded categories are abstract C++ classes that can have any number of concrete implementations. Our data structures are flexible enough to work with arbitrary node implementations, simply linked into the binary. To demonstrate our method, we implement Combine nodes for CSG operations and several custom Create nodes, including a CreateOpenScad node that can evaluate arbitrary [OpenSCAD] scripts. Our implementations rely on Triangle [Shewchuk 1996], [Clipper], and, of course, [OpenSCAD].

Each node implementation is governed by a unique set of parameters from a small number of types (int, bool, double, vector).¹ To deal with such different sets of parameters in a consistent way, we use protocol buffers [Google] that enable one to get and set parameters by type and name.

In addition, protocol buffers provide a handy way to read and write trees of operations to and from file (i.e. using protobuf MessageSet), and to save and recombine information about the valid region after parallel precomputation. We also use formatted protocol buffers with fixed double precision as keys to the GeometryCache.

Likewise, we implement precomputation in a fully general way to accept designs of variable parameter types (bool, double, int) and dimensionality of the parameter space. In order to deal with continuous and discrete dimensions smoothly, we use a base C++ class Dimension. Subclasses of Dimension keep track of cuts along their dimension and report whether or not a new subdivision can be made. Each sample point is treated as a vector of doubles, but internally stores the double, integer and boolean values.

We use Gecode [Schulte et al. 2010] to solve for linear and non-linear constraints, and rely on OpenVDB [Museth 2013] for volumetric operations needed for the tests and ΔG computation.

2 ΔG as a Metric Space

Lemma: ΔG is a metric, where ΔG is defined as the symmetric difference (XOR) of the volumes of two shapes A and B , normalized by the volume of their union:

$$\Delta G(A, B) \triangleq \frac{|A \oplus B|}{|A \cup B|}$$

Proof:

Non-negativity, symmetry: hold trivially.

¹For some node implementations the set of parameters varies among instances of the same implementation (e.g. CreateOpenScad nodes have parameters that depend on the input script).

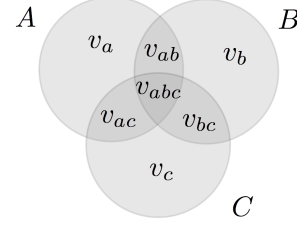


Figure 1: Space partitioned by three volumetric shapes, with regions of overlap (potentially empty) labeled by variables.

Coincidence: $\Delta G(A, B) = 0 \Leftrightarrow A = B$. ΔG is only zero if the symmetric difference of two shapes is zero, and that holds if and only if two volumes are identical.

Triangle inequality: $\Delta G(A, C) \leq \Delta G(A, B) + \Delta G(B, C)$ for all A, B, C . For any A, B, C , w. l. g. let the space be partitioned according to which of the three shapes overlap in that region of space, as shown in the Fig. 1, where each variable refers to the volume of the corresponding region. Then, we can trivially state the following identities:

$$|A \oplus C| = |A \oplus B| + |B \oplus C| - (2v_b + 2v_{ac}) \quad (1)$$

$$|A \cup B| = |A \cup C| + v_b - v_c \quad (2)$$

$$|B \cup C| = |A \cup C| + v_b - v_a \quad (3)$$

Consider the expanded triangle inequality:

$$\Delta G(A, C) \stackrel{?}{\leq} \Delta G(A, B) + \Delta G(B, C)$$

$$\frac{|A \oplus C|}{|A \cup C|} \stackrel{?}{\leq} \frac{|A \oplus B|}{|A \cup B|} + \frac{|B \oplus C|}{|B \cup C|}$$

Then, applying identities 1-3 we obtain:

$$\frac{|A \oplus B|}{|A \cup C|} + \frac{|B \oplus C|}{|A \cup C|} - \frac{2(v_b + v_{ac})}{|A \cup C|} \stackrel{?}{\leq} \frac{|A \oplus B|}{|A \cup C| + v_b - v_c} + \frac{|B \oplus C|}{|A \cup C| + v_b - v_a}$$

Then carrying over and combining terms:

$$\frac{-2(v_b + v_{ac})}{|A \cup C|} \stackrel{?}{\leq} \frac{(v_c - v_b) \cdot |A \oplus B|}{|A \cup C| \cdot (|A \cup C| + v_b - v_c)} + \frac{(v_a - v_b) \cdot |B \oplus C|}{|A \cup C| \cdot (|A \cup C| + v_b - v_a)}$$

Then simplifying and reapplying identities 2 and 3 in reverse:

$$-2(v_b + v_{ac}) \stackrel{?}{\leq} \frac{(v_c - v_b) \cdot |A \oplus B|}{|A \cup B|} + \frac{(v_a - v_b) \cdot |B \oplus C|}{|B \cup C|}$$

Then, reversing the inequality and collapsing the terms back:

$$2(v_b + v_{ac}) \stackrel{?}{\geq} (v_b - v_c) \cdot \Delta G(A, B) + (v_b - v_c) \cdot \Delta G(B, C)$$

It is sufficient to prove that the left expression is greater than or equal to the maximum possible value of the right expression. This happens when both terms on the right are positive, i. e. when $v_b > v_c$ and $v_b > v_a$, and when both ΔG terms attain the maximum possible value of 1. Thus it remains to show:

$$2v_b + 2v_{ac} \stackrel{?}{\geq} 2v_b - v_c - v_a$$
$$2v_{ac} \geq -(v_c + v_a)$$

This is trivially true, because all the volumes are nonnegative. \square

References

- Carve CSG. <http://carve-csg.com>.
- Clipper. <http://www.angusj.com/delphi/clipper.php>.
- GOOGLE. Protocol buffers. <http://code.google.com/apis/protocolbuffers/>.
- MUSETH, K. 2013. VDB: High-resolution sparse volumes with dynamic topology. *ACM Trans. Graph.* 32, 3, 27:1–27:22.
- OpenSCAD. <http://www.openscad.org>.
- SCHULTE, C., TACK, G., AND LAGERKVIST, M. Z., 2010. Modeling and programming with gecode. <http://www.gecode.org/>.
- SHEWCHUK, J. R. 1996. Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. In *Applied computational geometry towards geometric engineering*. Springer, 203–222.