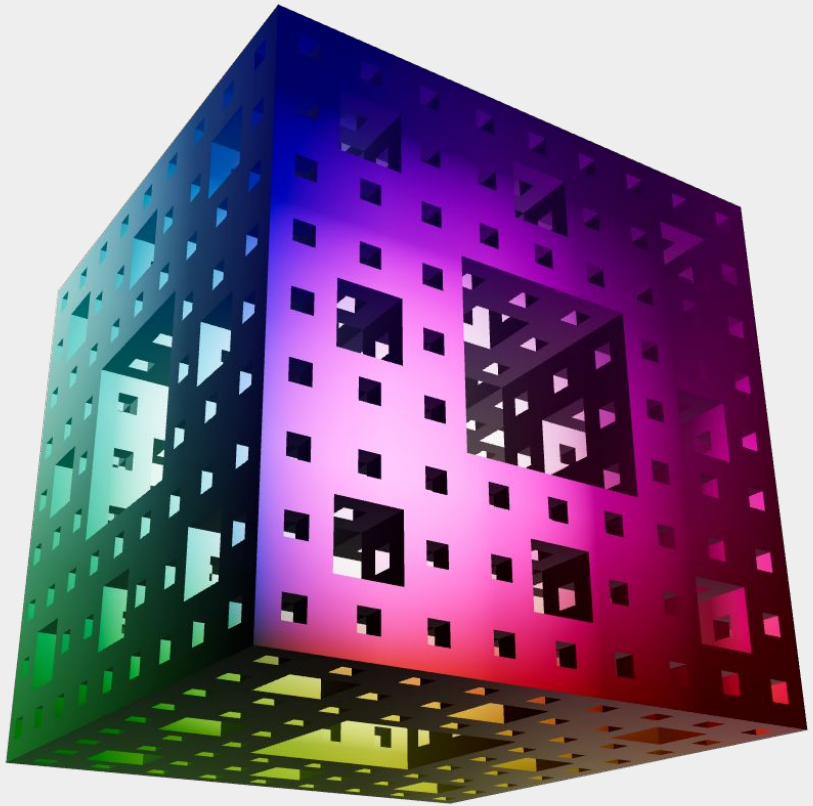
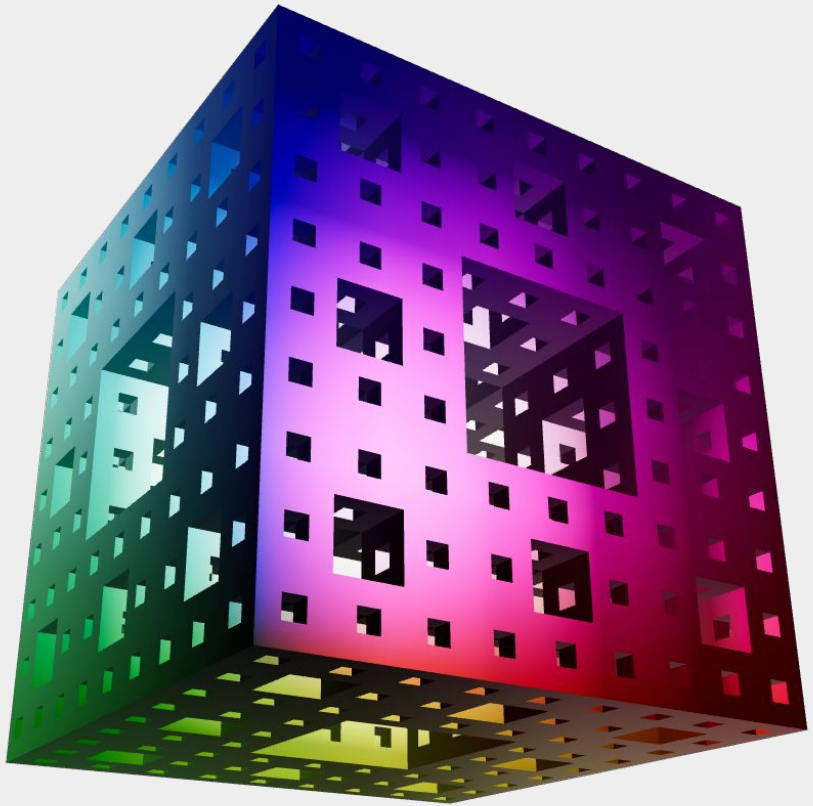
Optimizing Mesh Booleans by Being Lazy (Experience Report)

Chun Kit Lam (HKUST)



~~Optimizing Mesh Booleans by Being Lazy (Experience Report)~~

Optimizing a CG Library without learning CG



elalish / manifold Public

Notifications | Fork 217 | Star 2.1k

Code | Issues 28 | Pull requests 13 | Discussions | Actions | Wiki | Security and quality | Insights

master | 14 Branches | 28 Tags | Go to file | Code

elalish fix camera radius reset (#1733) ✓ 2c98e8b · 2 days ago | 1,559 Commits

.github	Add MANIFOLD_NO_IOSTREAM build-time option (#1690)	3 weeks ago
.vscode	Some partial fixes for normals (#1716)	last week
bindings	fix camera radius reset (#1733)	2 days ago
cmake	Add CrossSection backend selector (#1710)	2 weeks ago
docs	Whoops - looks like Doxyfile needs to be top level (#965)	2 years ago
extras	Fix MSVC strict conversion warnings (#1726)	3 days ago
include/manifold	Thread ExecutionContext into MeshGL ingest via Executio...	3 days ago
samples	Deprecate compose (#1489)	4 months ago
scripts	v3.5 (#1730)	3 days ago
src	Thread ExecutionContext into MeshGL ingest via Executio...	3 days ago
test	Thread ExecutionContext into MeshGL ingest via Executio...	3 days ago
.clang-format	added MANIFOLD_ASSERT (#1225)	last year
.codecov.yml	Changed MeshRelation API (#184)	4 years ago
.gersemirc	format cmake files (#975)	2 years ago
.gitattributes	Add MANIFOLD_NO_IOSTREAM build-time option (#1690)	3 weeks ago
.gitignore	Updated Java bindings (#1617)	2 months ago
AUTHORS	Add Geoff deRosenroll to author list (#351)	3 years ago
CMakeLists.txt	v3.5 (#1730)	3 days ago
CONTRIBUTING.md	Remove CLA requirement from CONTRIBUTING.md (#144...	6 months ago
Doxyfile	fix docs and binding defaults (#1729)	3 days ago
LICENSE	Initial commit	7 years ago

About

Geometry library for topological robustness

Readme
Apache-2.0 license
Contributing
Activity
2.1k stars
35 watching
217 forks
Report repository

Releases 23

v3.5.0 Latest
3 days ago

+ 22 releases

Packages

No packages published

Contributors 84

+ 70 contributors

Languages

C++ 71.6% | TypeScript 14.2% | JavaScript 5.1% | Python 4.9% | CMake 2.2% | HTML 0.6%

blender

OpenSCAD

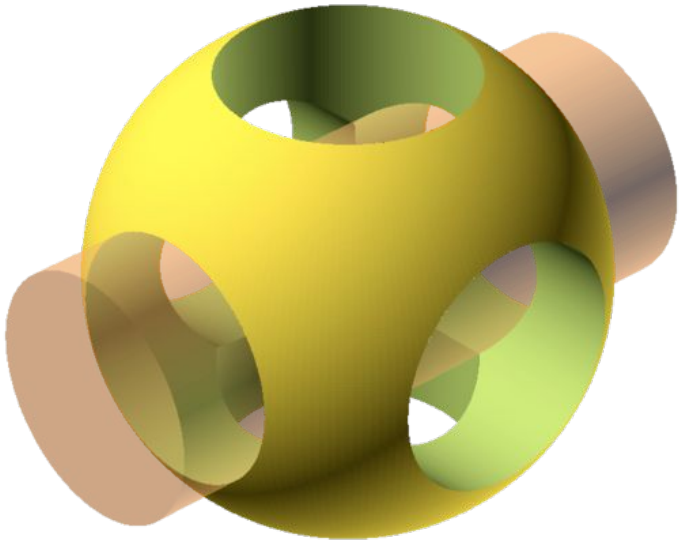
GODOT Game engine

babylon.js

Background: OpenSCAD

- Functional Language *(not a great one…)*
- Constructive Solid Geometry (CSG)

```
difference() {
  sphere(d=50);
  cylinder(d=25, h=65, …);
  rotate([90, 0, 0])
    cylinder(...);
  rotate([0, 90, 0])
    cylinder(...);
}
```

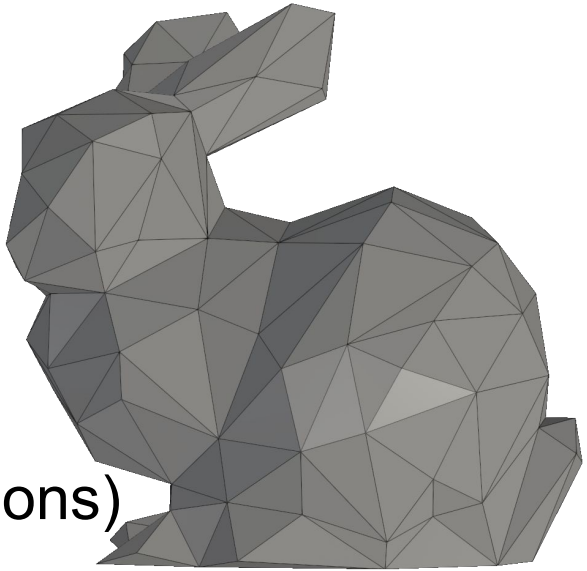


Problem: OpenSCAD is *SLOW…*

- Compile time: minutes/hours
- Unacceptable for me: Need fast iterations
- More than 100x faster with manifold

Why OpenSCAD is slow?

- Solid mesh modeling:
  More than just triangle soup
- Topo: Closed 2-manifold
- Geometry: No self-intersections
- Float: Robustness issues
  (Violates Euclidean Geometry Assumptions)
- Exact: Slow, Snap-Rounding

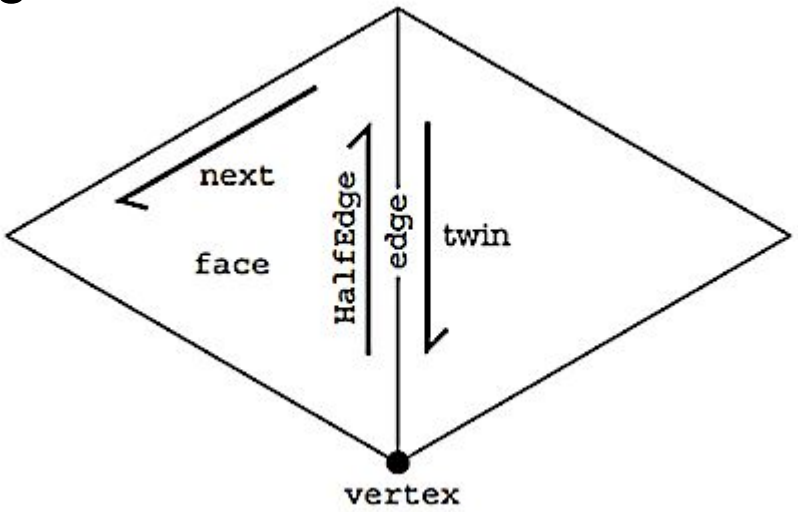


Model credit: https://www.thingiverse.com/thing:151081

How manifold solves this?

Implements Smith's robust inexact algorithm

- Careful floating point predicates
- Guaranteed topology:
  Even for random coords input



Topology over geometry correctness

- Broken topology is harder to fix
- Some algo/data structure relies on this
- Geometrically incorrect models are everywhere

Manifold's API

Lazy:

- Copy (just shared)
- Booleans
- Affine Transforms

Triggers Eval:

- Queries

Simple, intuitive
No other libs have this (AFAIK)

Boolean Operations		
$\cup, -, \cap$	$M \to M \to M$	Mesh Booleans
Transformations		
translate, scale, rotate	$M \to \mathbb{R}^3 \to M$	Common Affine Transforms
transform	$M \to \mathbb{R}^{3\times4} \to M$	Affine Transform
warp	$M \to (\mathbb{R}^3 \to \mathbb{R}^3) \to M$	Vertex Transform
Constructors		
from_mesh	Mesh $\to M$	Construct a Manifold mesh object from a raw mesh
sphere	$\mathbb{R} \to M$	Sphere constructor
cube	$\mathbb{R}^3 \to M$	Rectangular cuboid constructor
Miscellaneous		
to_mesh	$M \to$ Mesh	Get mesh output
decompose	$M \to$ List M	Decompose the mesh into disjoint components.
surface_area	$M \to \mathbb{R}$	Surface area of the mesh
volume	$M \to \mathbb{R}$	Volume of the mesh
genus	$M \to \mathbb{N}$	Genus of the mesh

Notes about Correctness

- Mesh Boolean not associative/commutative
- Rewrite changes results
  Like floating point rewrites
- Users should not depend on exact results
- Only guarantee determinism

A + (B + C)

≠

(A + B) + C

≠

(B + A) + C

AST Rewriting

- Goal: Maximize arity
- Use optimized variadic Boolean

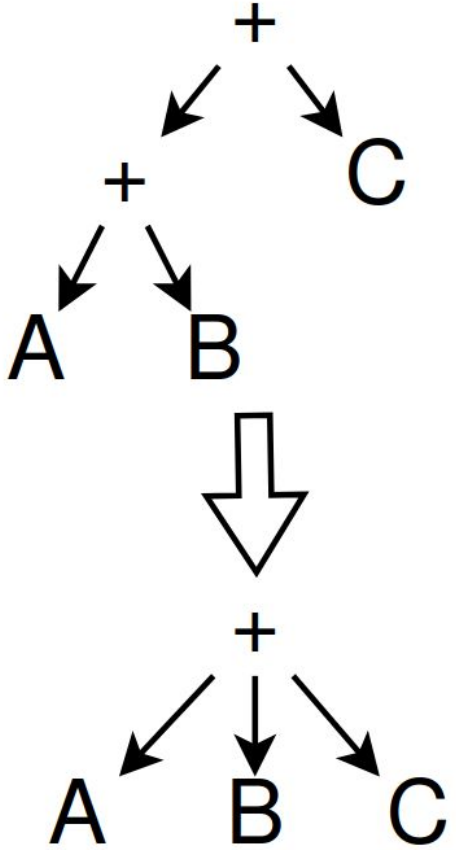
1. Propagate transforms to leaves

$$t(a \cup b) = (t\,a) \cup (t\,b)$$

2. Maintain a normal form

$$\bigcirc \overline{n_1} - \overline{n_2} = \bigcirc_{n \in \overline{n_1}} n - \bigcup_{m \in \overline{n_2}} m$$

$$\bigcirc \in \{\bigcup, \bigcap\}$$



$$\boxed{[\![m]\!]_t = n}$$

ATOM

$$\frac{}{[\![p]\!]_t = \cup\,(t\ p)}$$

TRANSFORM

$$\frac{}{[\![t_0\ m]\!]_t = [\![m]\!]_{t \times t_0}}$$

SUBTRACTION-1

$$\frac{[\![m_0]\!]_t = \bigcirc \overline{n_0} - \overline{n_0}' \qquad [\![m_1]\!]_t = \cup\,\overline{n_1}}{[\![m_0 - m_1]\!]_t = \bigcirc \overline{n_0} - (\overline{n_0}', \overline{n_1})}$$

SUBTRACTION-2

$$\frac{[\![m_0]\!]_t = \bigcirc \overline{n_0} - \overline{n_0}'}{[\![m_0 - m_1]\!]_t = \bigcirc \overline{n_0} - (\overline{n_0}', [\![m_1]\!]_t)}$$

VARIADIC-1

$$\frac{[\![m_0]\!]_t = \bigcirc \overline{n_0} \qquad [\![m_1]\!]_t = \bigcirc \overline{n_1}}{[\![m_0 \bigcirc m_1]\!]_t = \bigcirc (\overline{n_0}, \overline{n_1})}$$

VARIADIC-2

$$\frac{i \in \{0, 1\} \qquad [\![m_i]\!]_t = \bigcirc \overline{n_i}}{[\![m_0 \bigcirc m_1]\!]_t = \bigcirc (\overline{n_i}, [\![m_{1-i}]\!])}$$

VARIADIC-3

$$\frac{}{[\![m_0 \bigcirc m_1]\!]_t = \bigcirc ([\![m_0]\!]_t, [\![m_1]\!]_t)}$$

Rewriting Examples

Flatten:

$$(a \cup b) \cup (c \cup d) = a \cup b \cup c \cup d$$

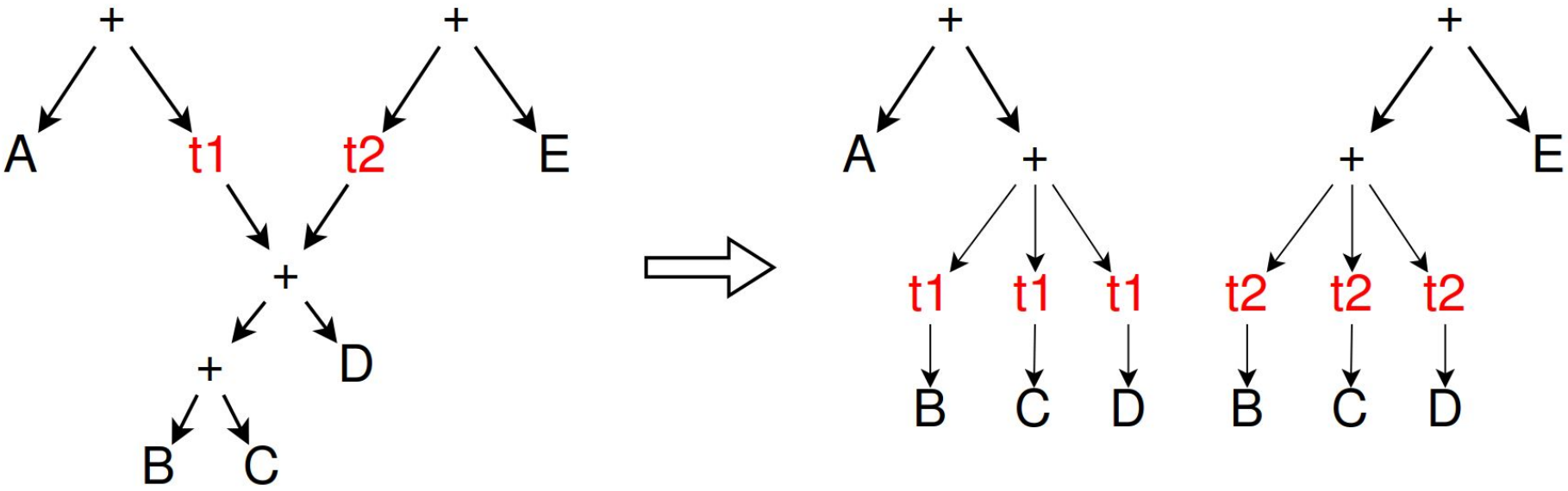
Negative operands:

$$(a - b) - (c \cup d) = a - (b \cup c \cup d)$$

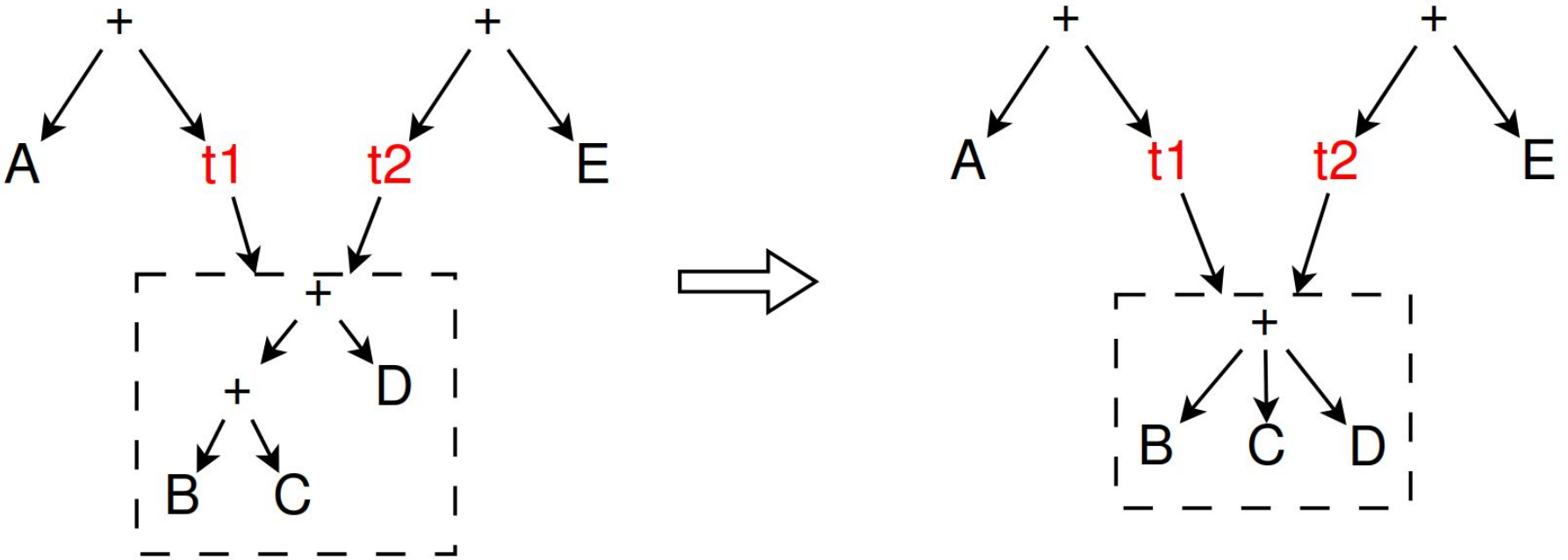
Positive operands:

$$(a - b) \cap (c \cap d) = (a \cap c \cap d) - b$$

Shared Nodes and Work Duplication



Shared Nodes and Work Duplication

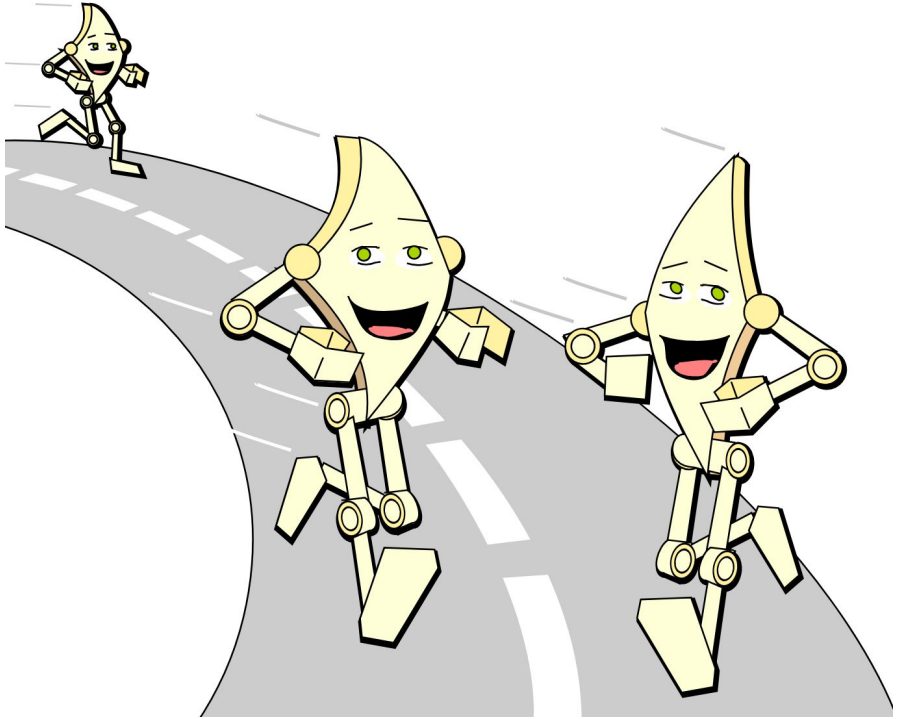


Optimizing Variadic Booleans

1. Compose disjoint meshes
2. Smaller meshes first
3. Parallelize Booleans

Naive implementation may cause data race!

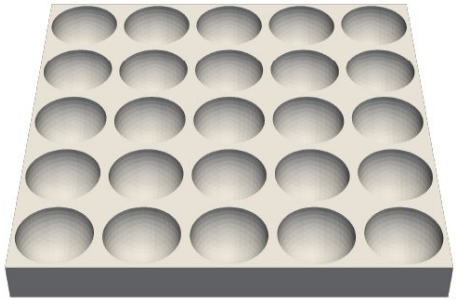
Needs fixed-size batching



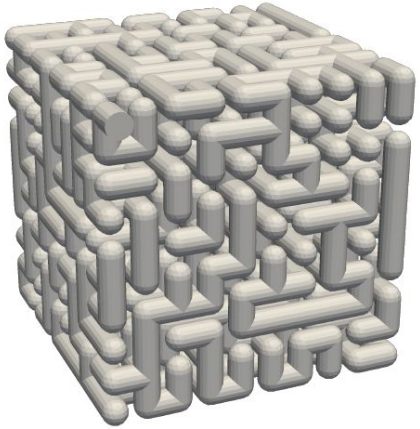
Cartoon adapted from https://gitlab.com/linux-kernel/perfbook

Example

Two models from OpenSCAD PR #4533

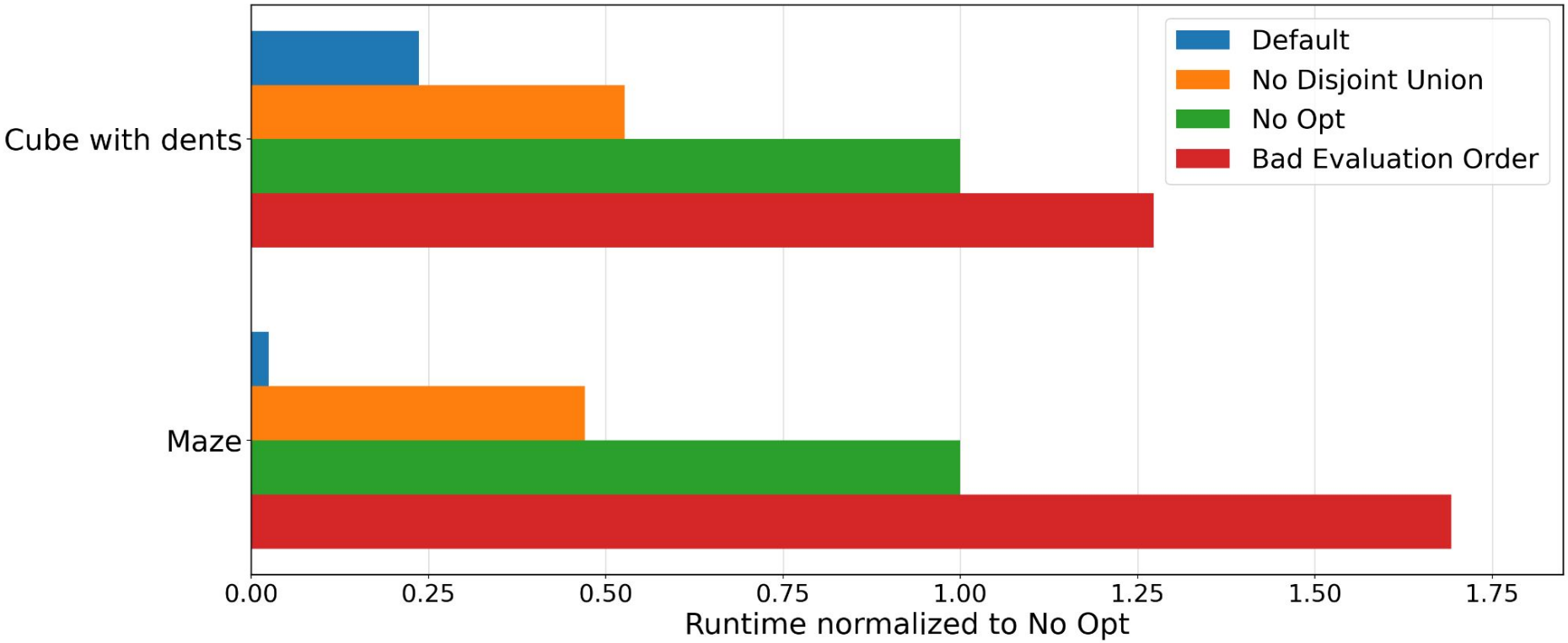


Cube with dents



Maze (negative part)

Example


Default
No Disjoint Union
No Opt
Bad Evaluation Order
Cube with dents
Maze
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
Runtime normalized to No Opt

Future Plan

- Dynamic rewriting: Subtraction vs Union
- More laziness internally, for better performance
- Support for constraint-based modeling
- More work on the geometry side…

https://manifoldcad.org/