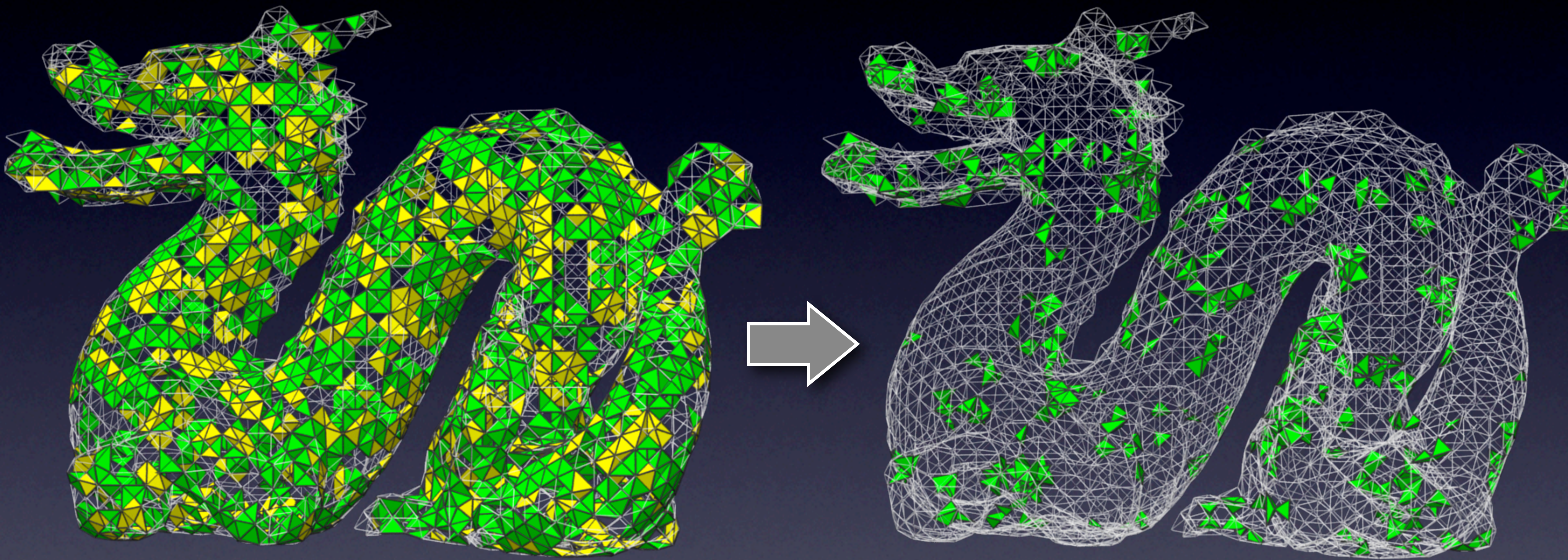
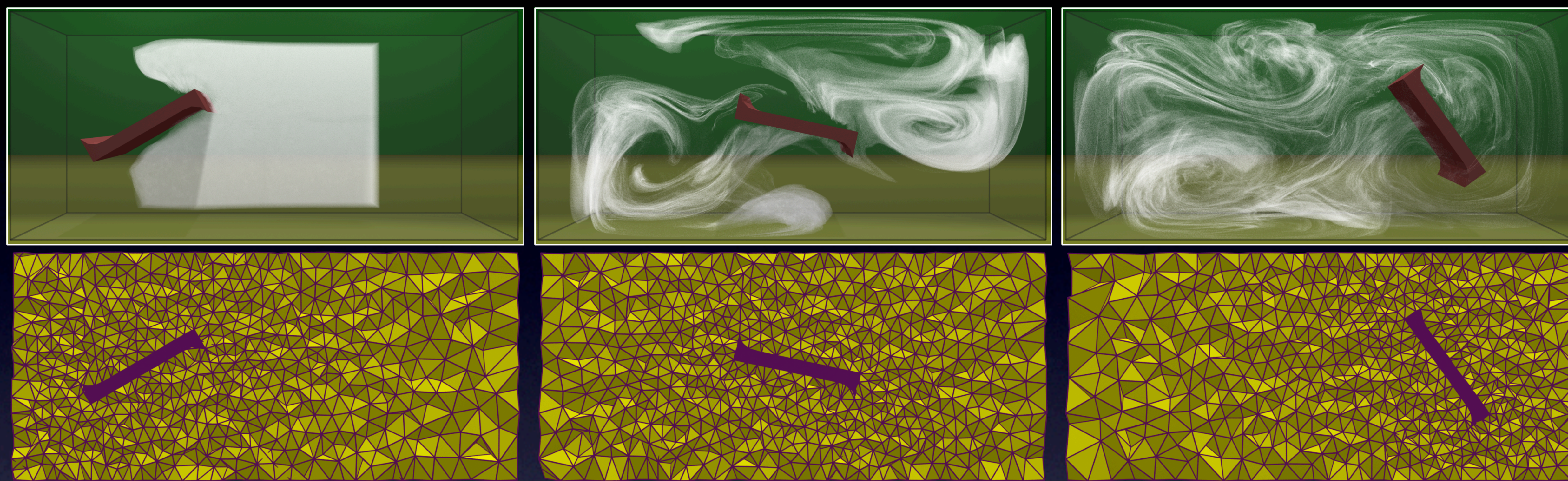
Tetrahedral Mesh Improvement



Bryan Klingner and Jonathan Shewchuk
12 March 2008

The usefulness of a mesh hinges on quality



bad elements

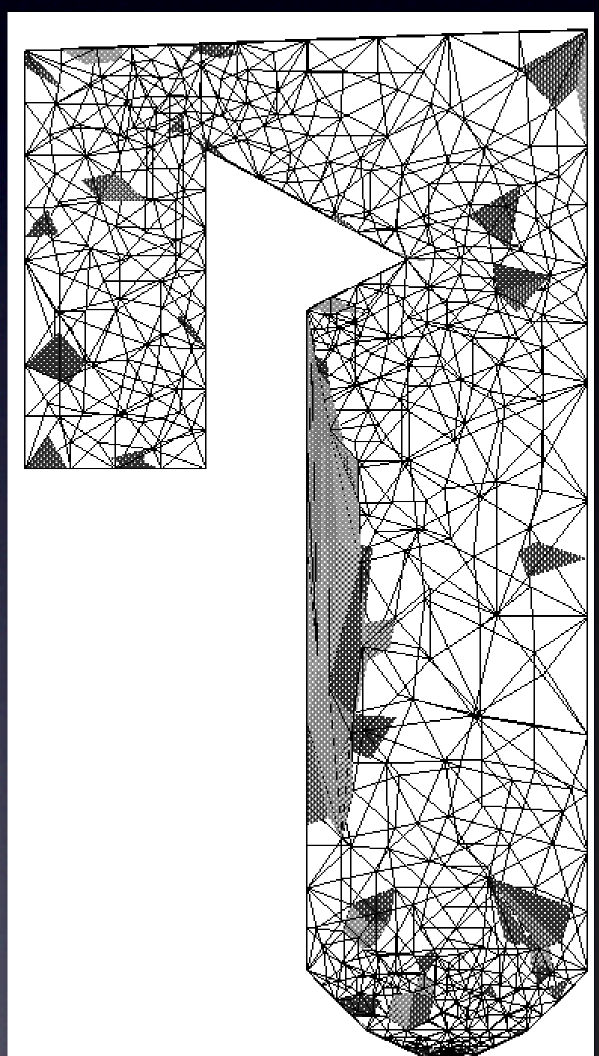
=

long running times

wrong answers

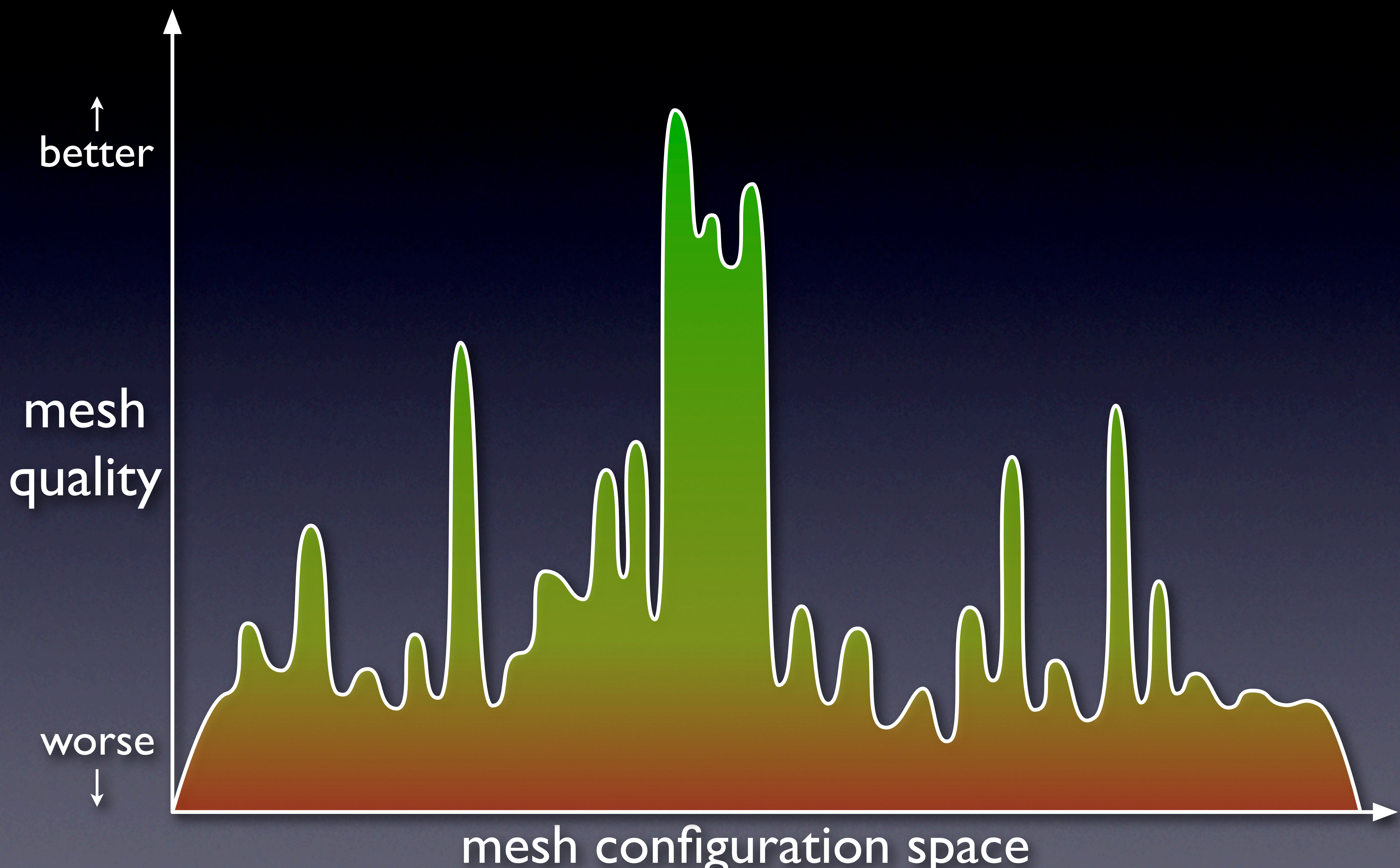
The best work so far:

"Tetrahedral Mesh Improvement Using Swapping and Smoothing"
Freitag and Ollivier-Gooch, 1997

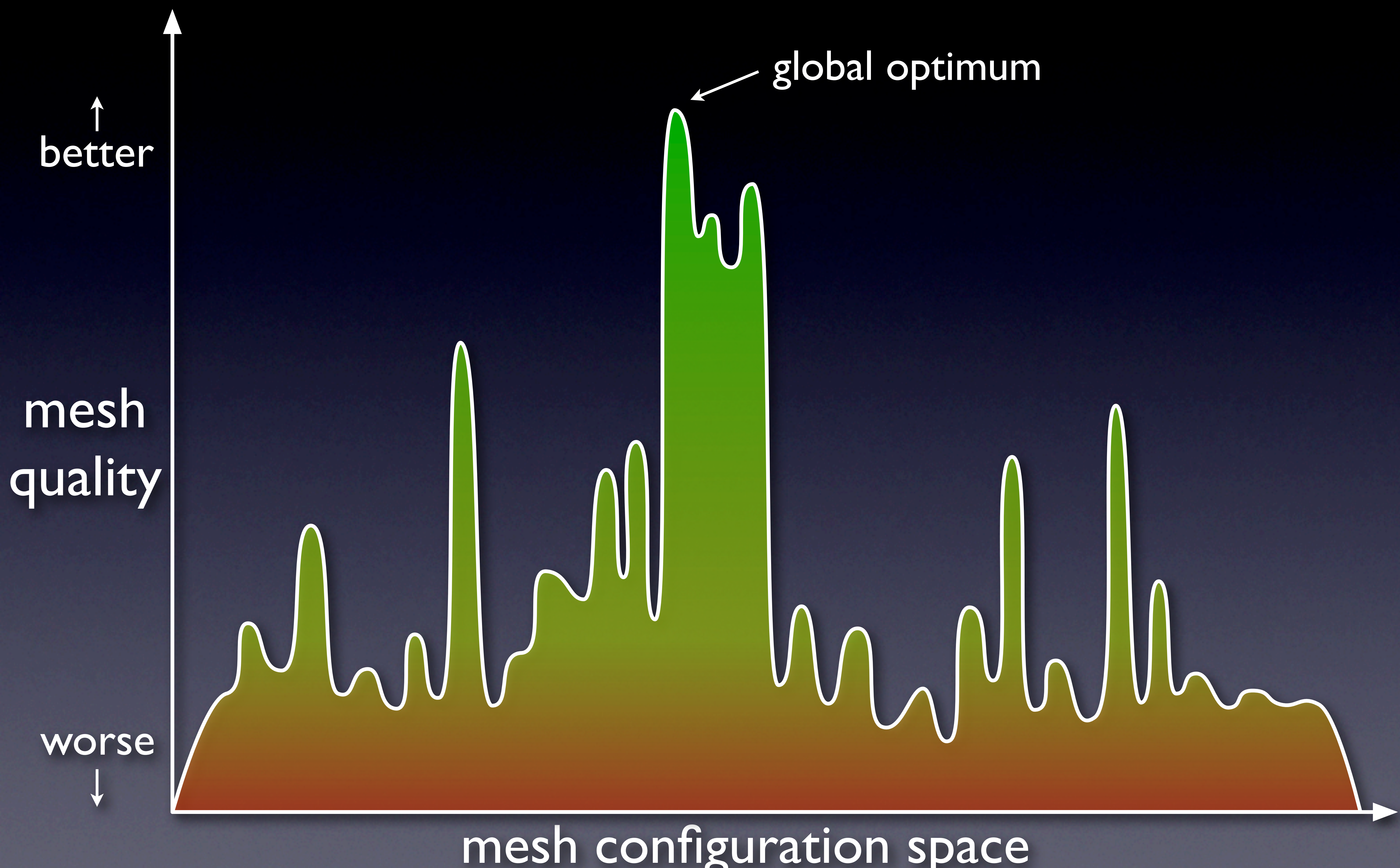


Collection of improvement operations
+
best experimental schedule
=
most bad elements removed

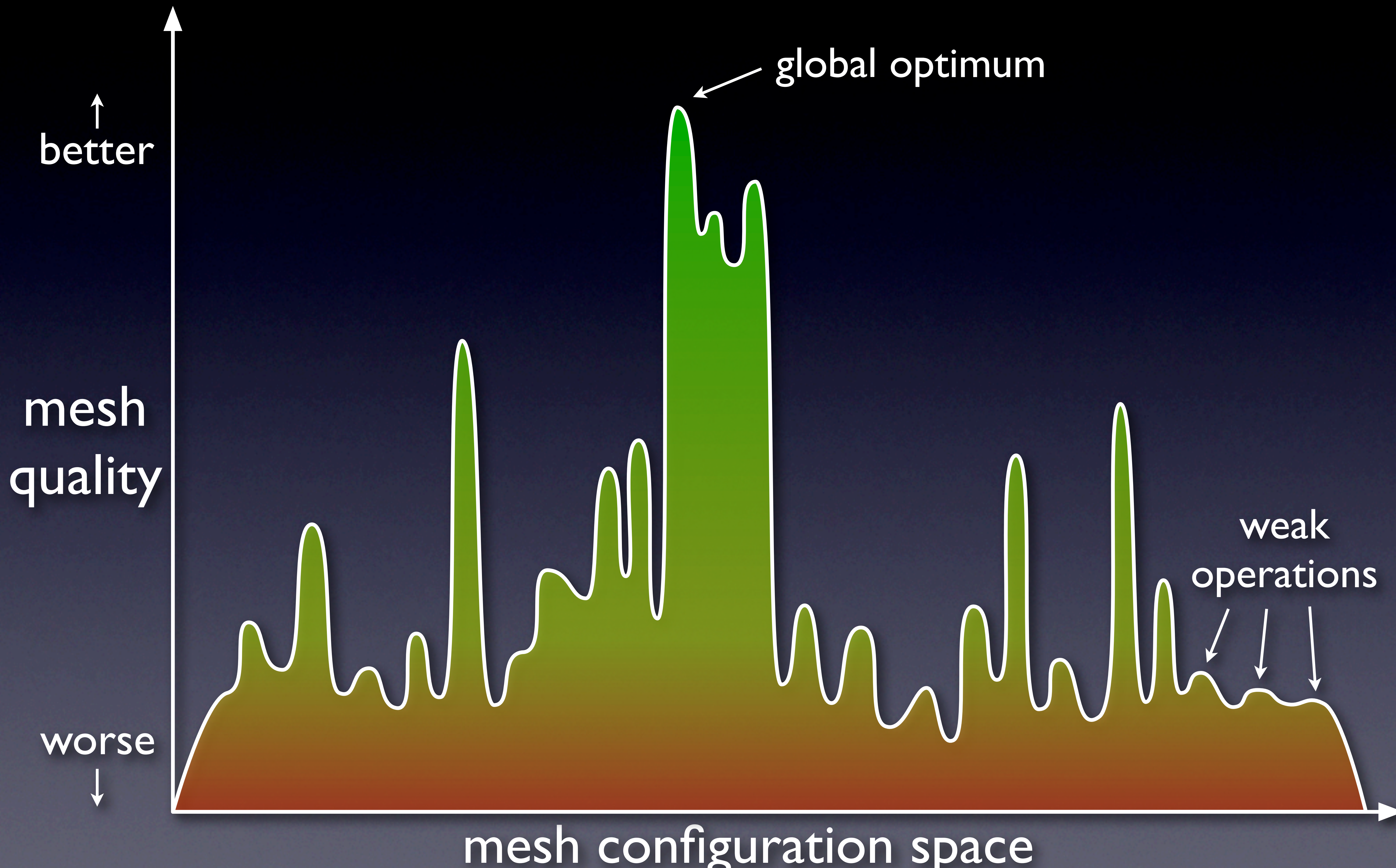
The mesh quality landscape



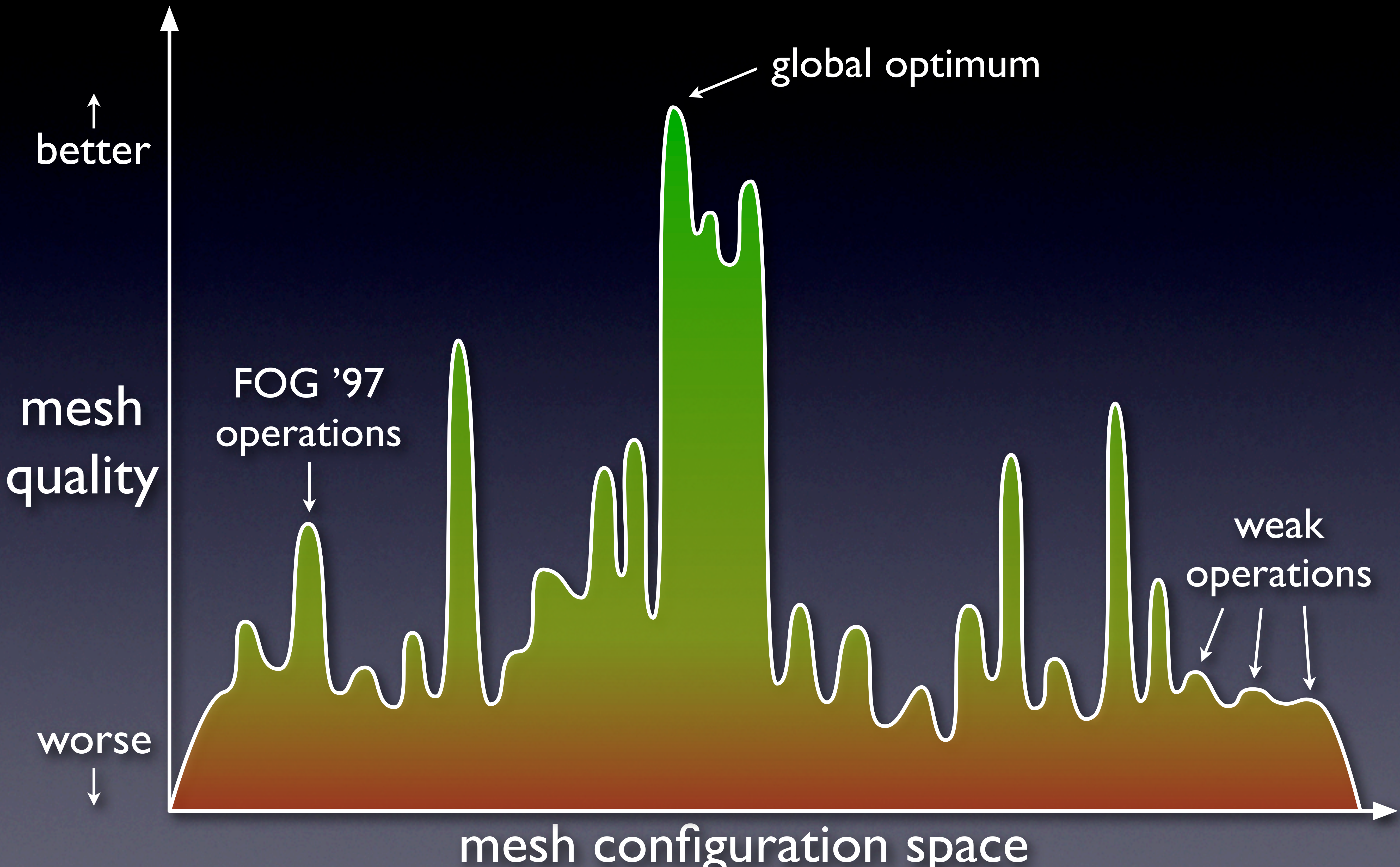
The mesh quality landscape



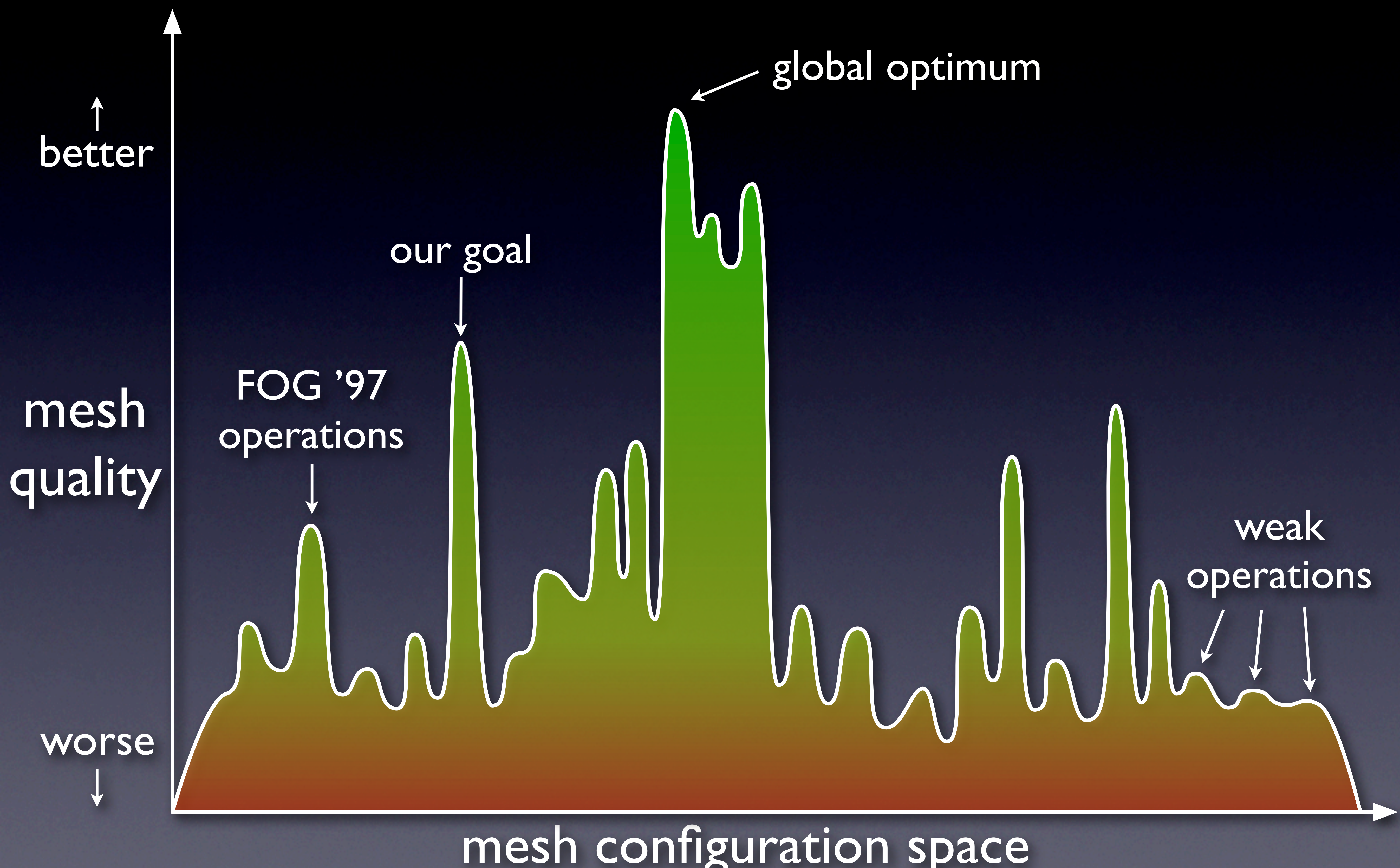
The mesh quality landscape



The mesh quality landscape

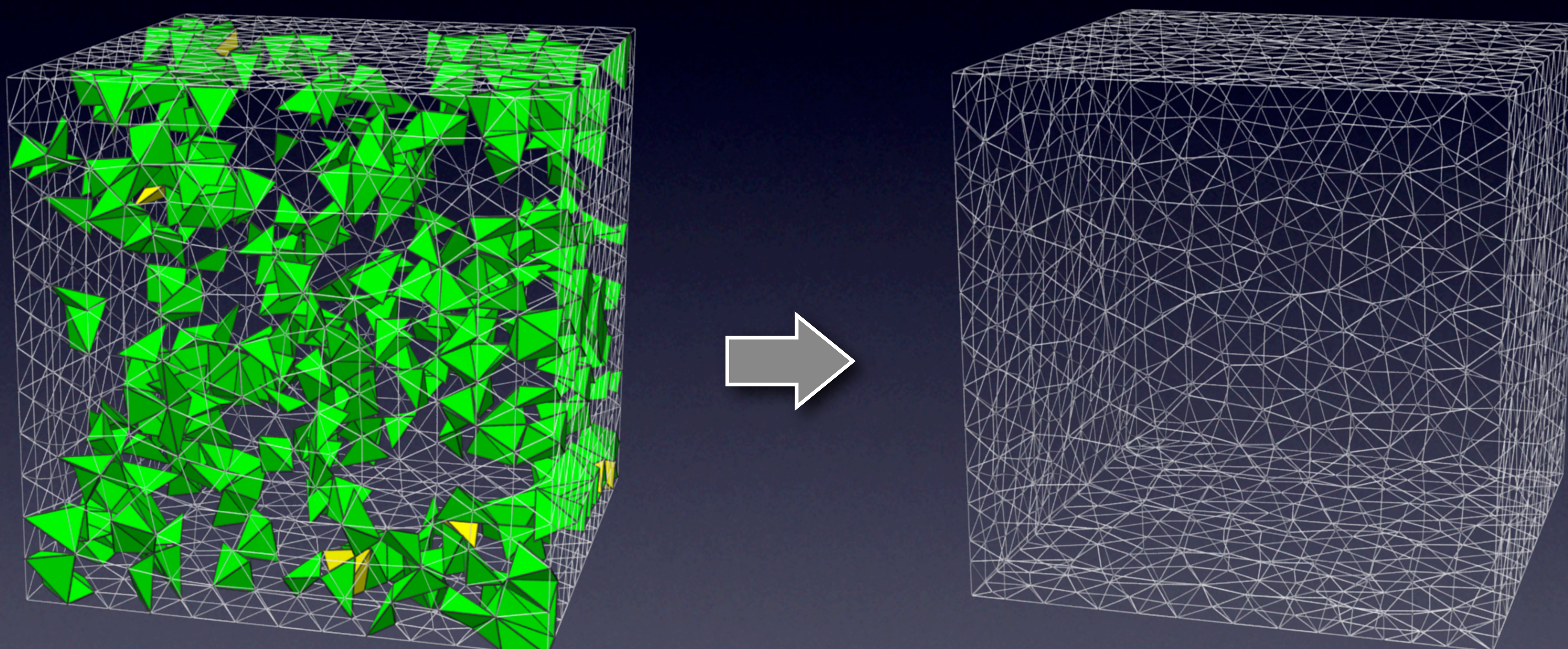


The mesh quality landscape



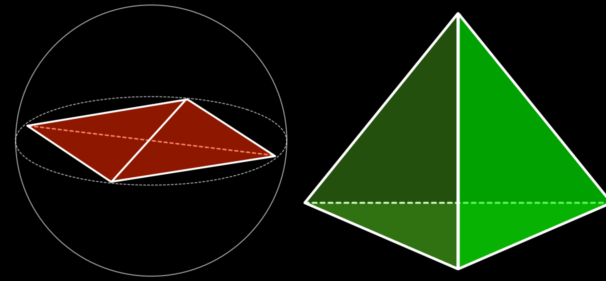
Our strategy:

Use every available tool (and a new one−vertex insertion)
and as much time as needed to produce the best mesh we can.

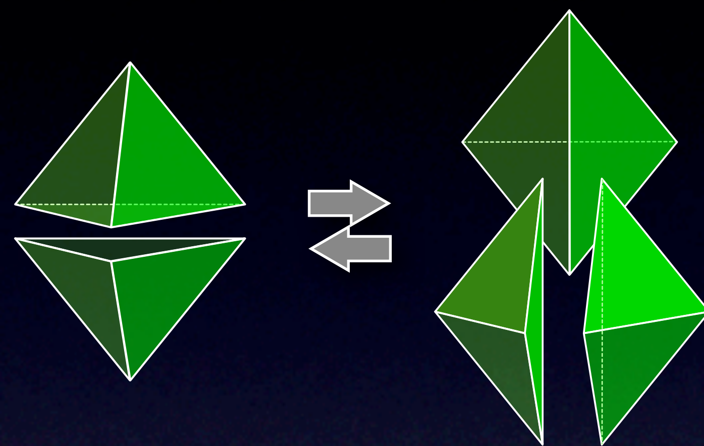


FOG ’97: 12 min / 160 max

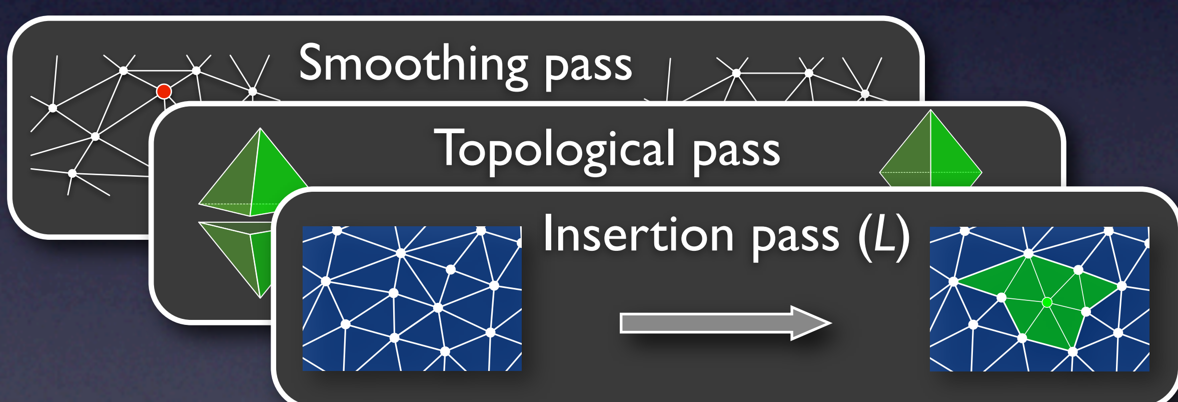
Our strategy: 30 min / $136^{\angle}$ max



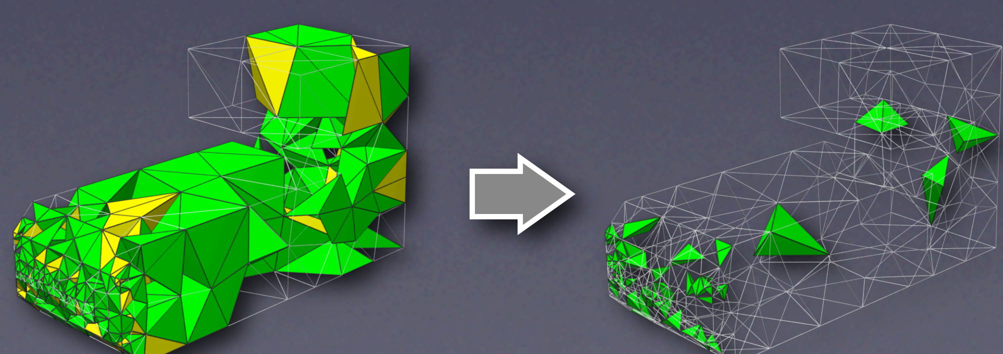
1 Mesh quality



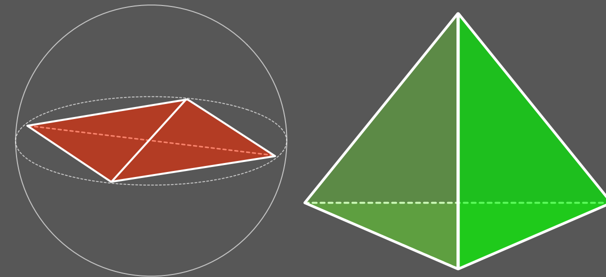
2 Improvement operations



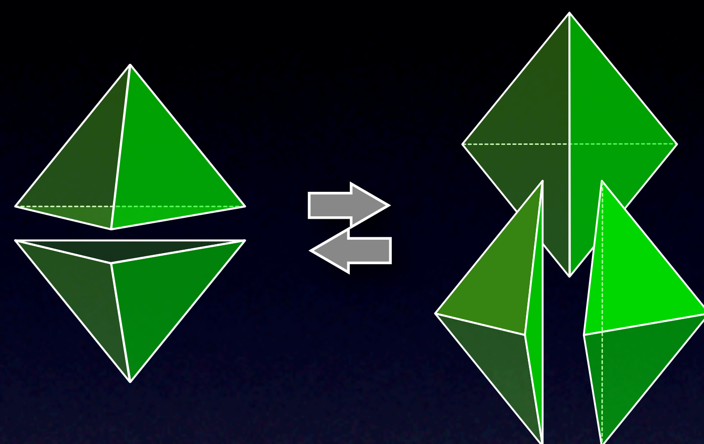
3 Improvement schedule



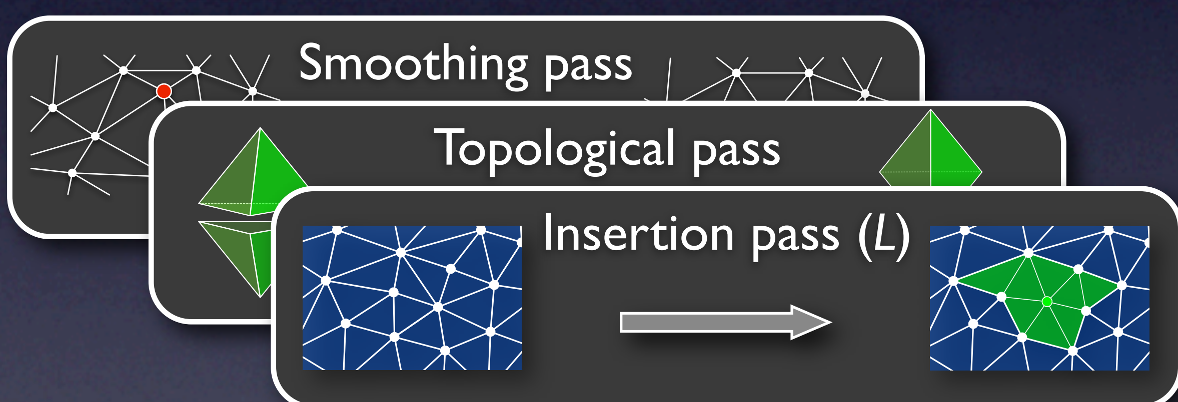
4 Results



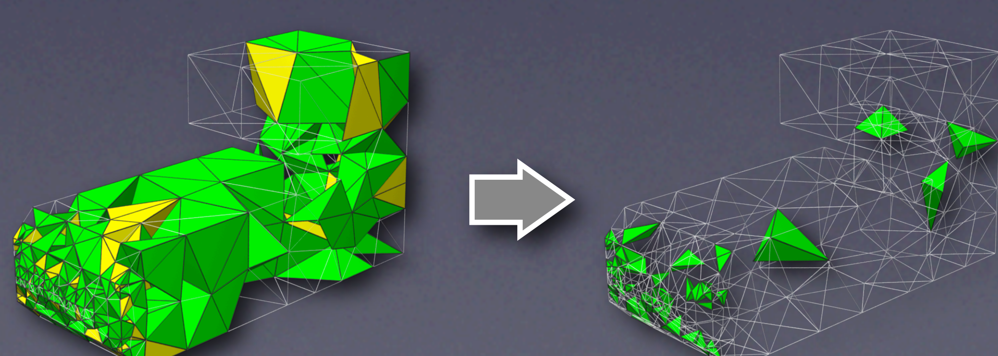
1 Mesh quality



2 Improvement operations

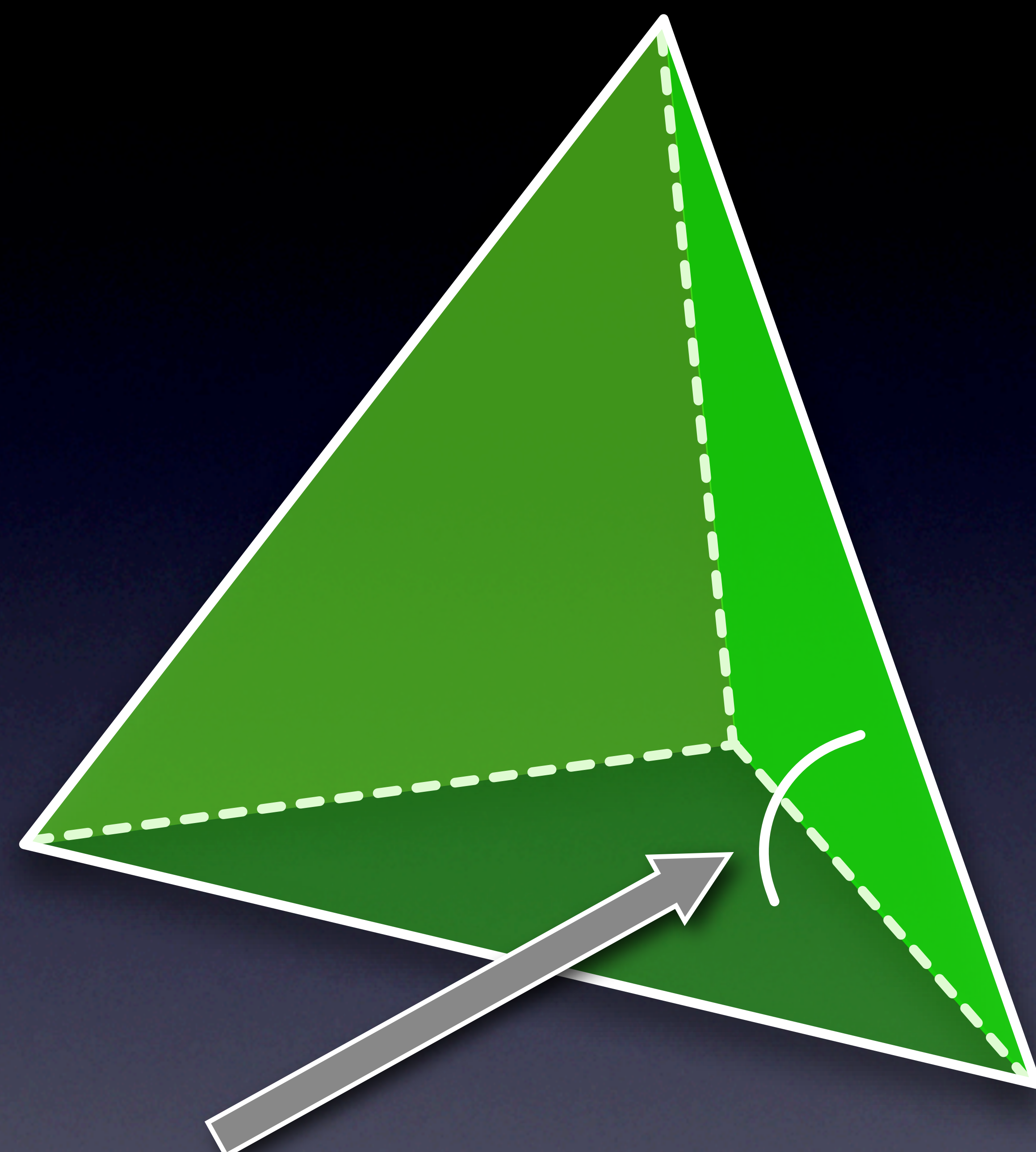


3 Improvement schedule



4 Results

What is a ‘bad’ element?



dihedral angle, the angle between two faces

Shewchuk, J. R. *What Is a Good Linear Finite Element? Interpolation, Conditioning, Anisotropy, and Quality Measures*, unpublished preprint, 2002.

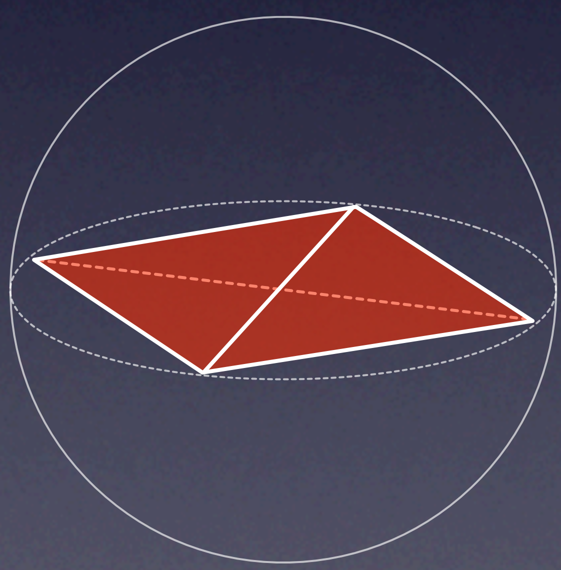
What is a 'bad' element?

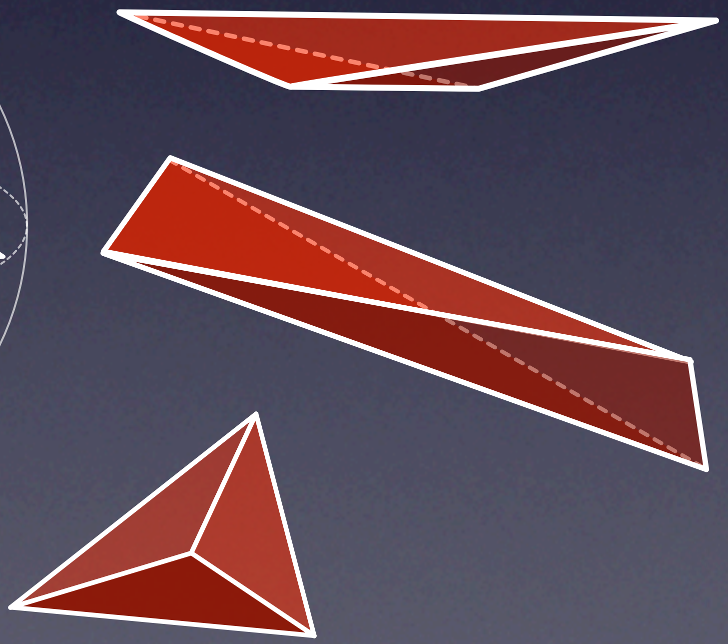
Extreme dihedral angles cause problems.

small dihedral angles may lead to poor conditioning, making the problem stiffer and slower to solve

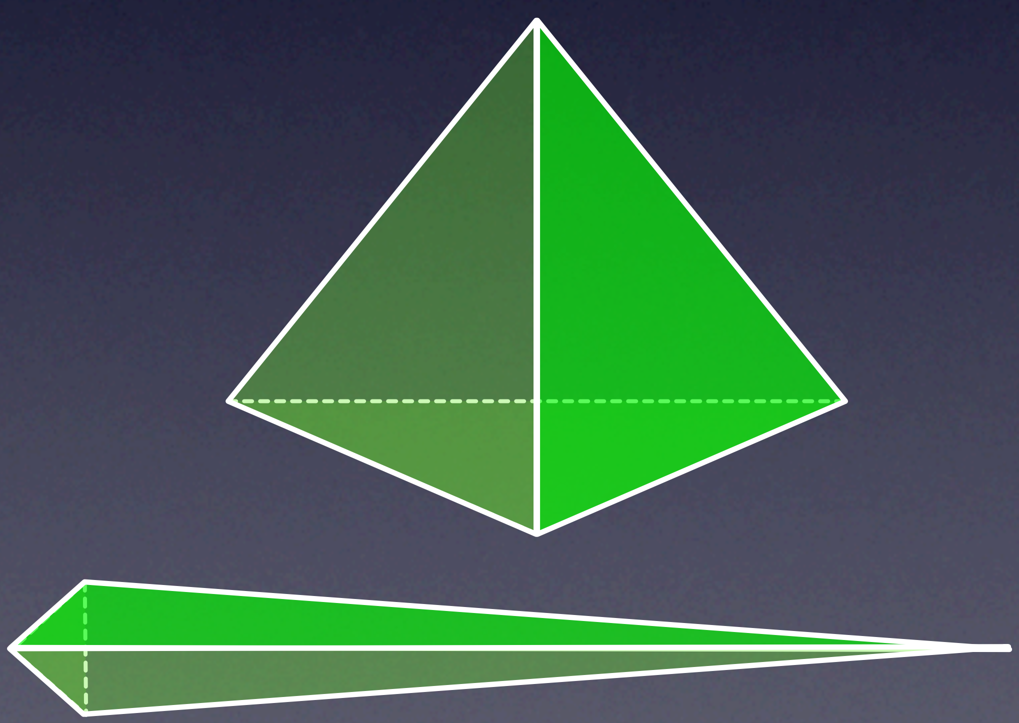
large dihedral angles lead to errors in discretization and interpolation

Bad shapes





Good shapes

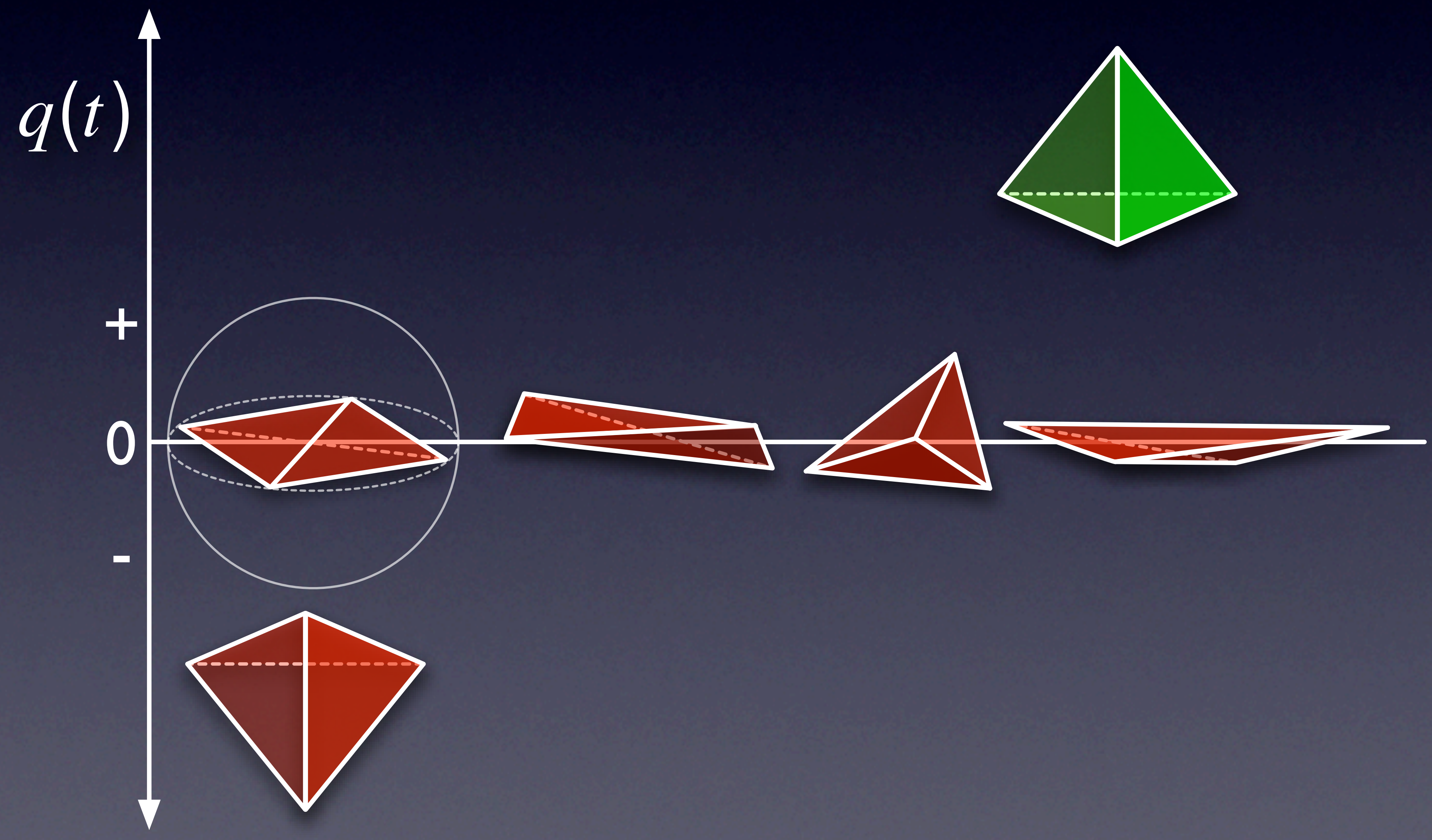


Shewchuk, J. R. *What Is a Good Linear Finite Element? Interpolation, Conditioning, Anisotropy, and Quality Measures*, unpublished preprint, 2002.

Given a tetrahedron t, let $q(t)$ be its quality.

$q(t)$ gets bigger for better elements

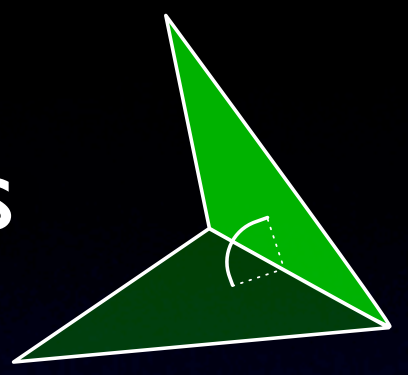
$q(t) \leq 0$ for degenerate or inverted elements



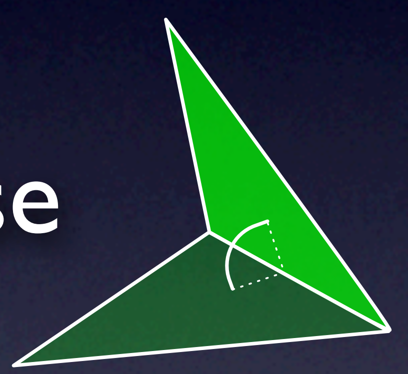
Shewchuk, J. R. *What Is a Good Linear Finite Element? Interpolation, Conditioning, Anisotropy, and Quality Measures*, unpublished preprint, 2002.

Quality measures: turning shape into a number

① Minimum sine of the six dihedral angles

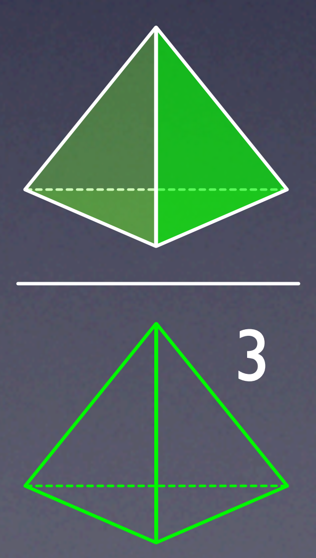


② Biased minimum sine - exaggerate obtuse



③

$$\frac{\text{Volume}}{\text{Root-mean-squared edge length}^3}$$



What is the quality of the whole mesh M?

A *quality vector* Q of each tetrahedron quality, sorted from worst to best:

$$Q(M) = \{1, 3, 10, 10, 15, 20, 23 \ldots\}$$

Compare quality vectors *lexicographically*: first by the first element, then by the second, and so on.

$$\{1, 100, 100, 100\} < \{2, 2, 2, 2\}$$

Improving the quality vector

{2, **5,** 5, 7, 9, 12, 15, **22,** 22, 34, 67, **104**}

{5, 22, 104}

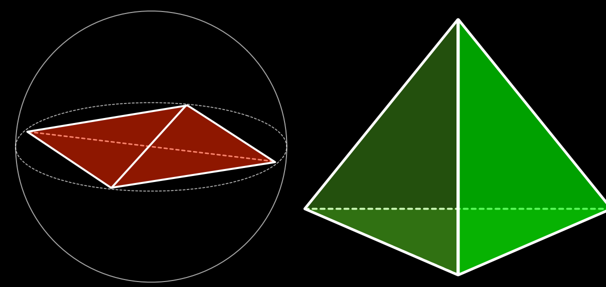
↓

Mesh Improvement Operation(s)

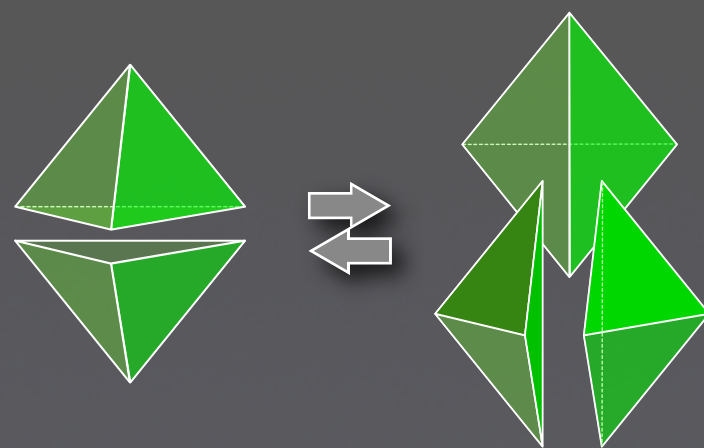
↓

{8, 10, 18}

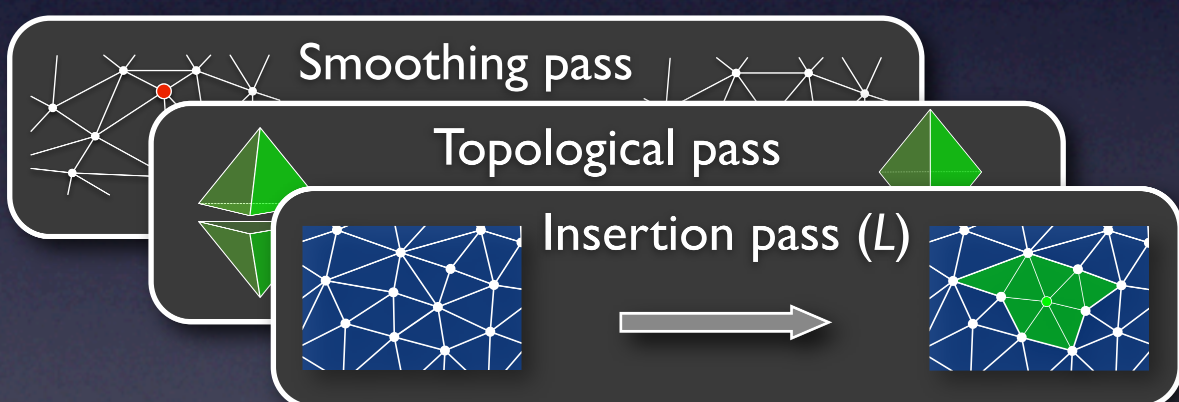
{2, 5, 7, **8,** 9, **10,** 12, 15, **18,** 22, 34, 67}



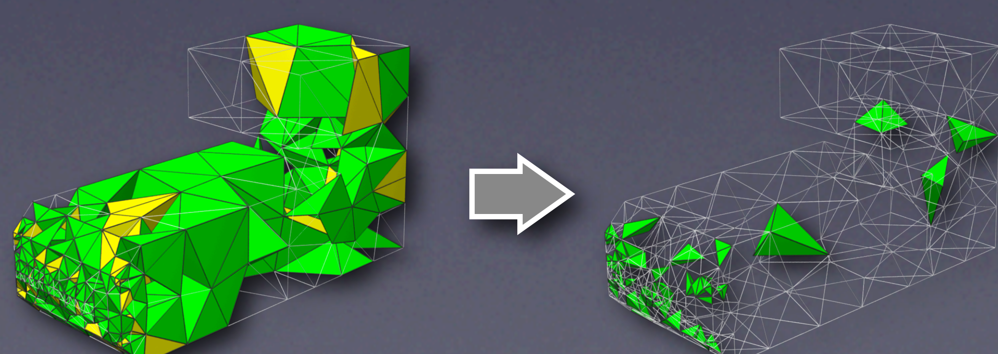
① Mesh quality



② Improvement operations

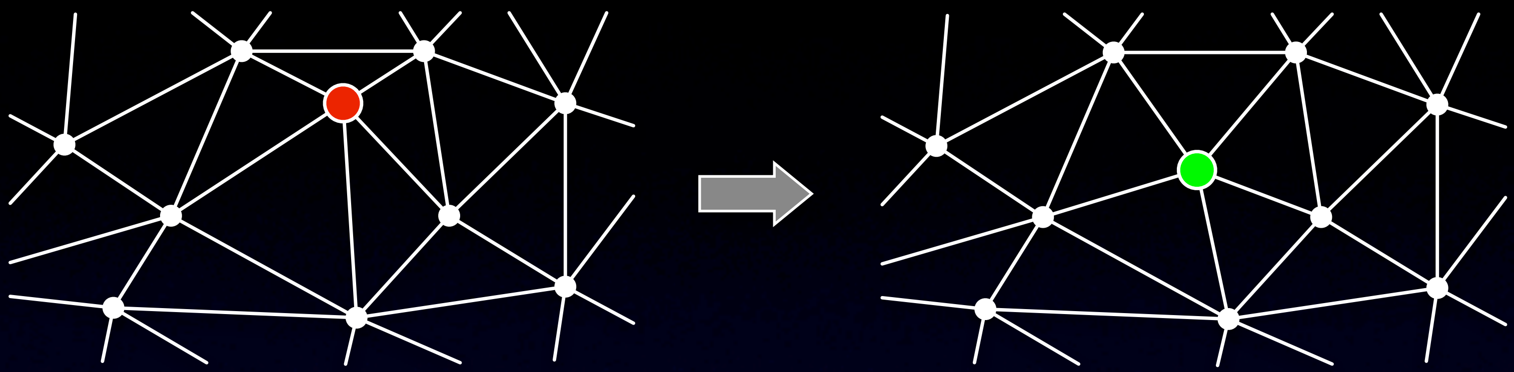


③ Improvement schedule

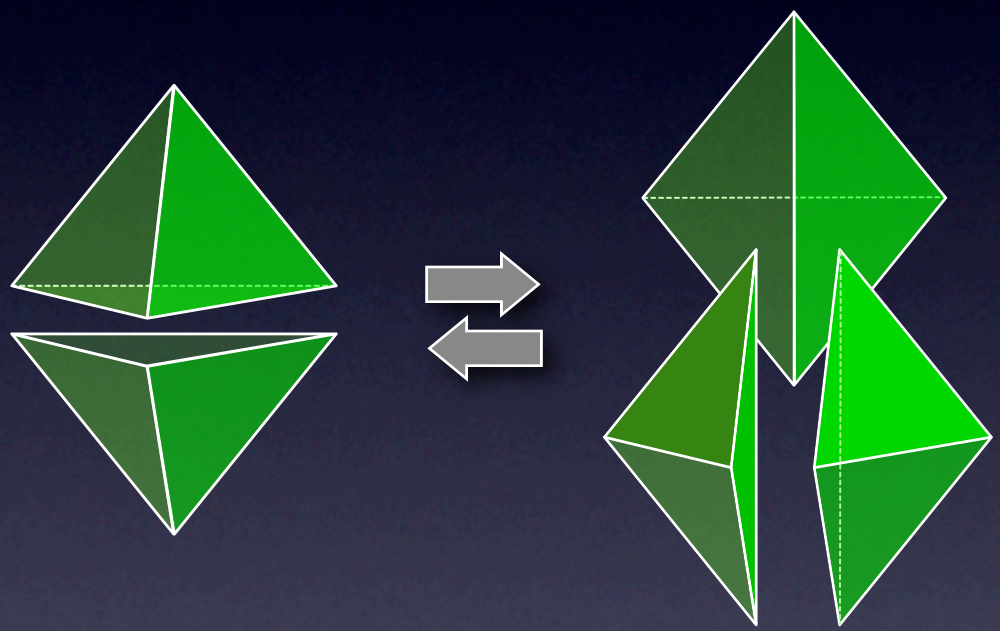


④ Results

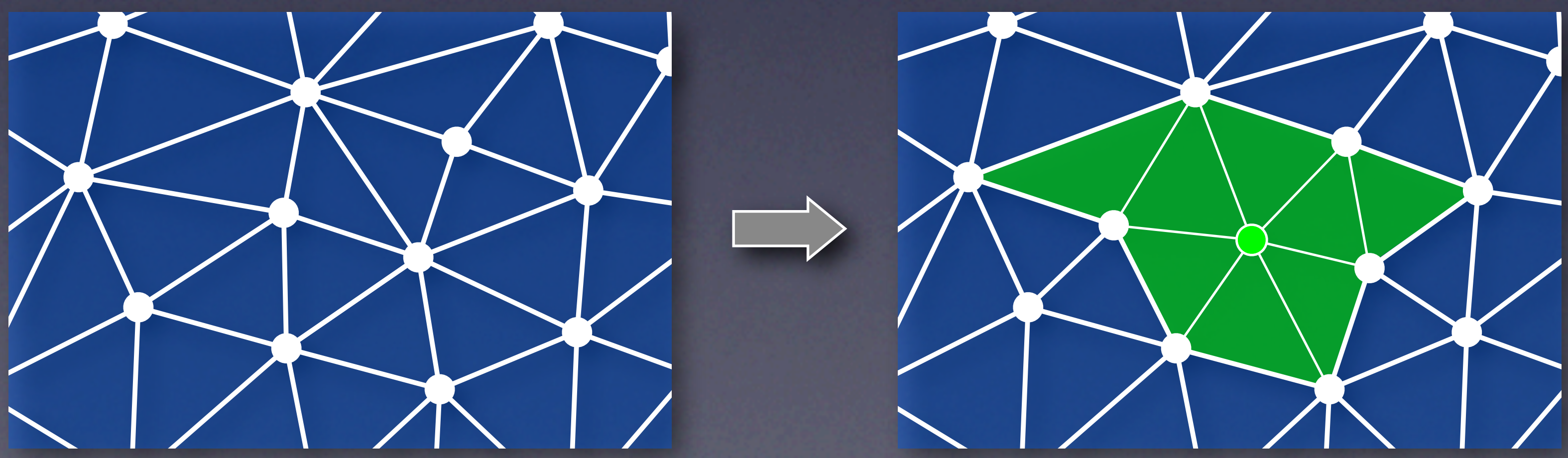
Mesh improvement operations



Vertex smoothing



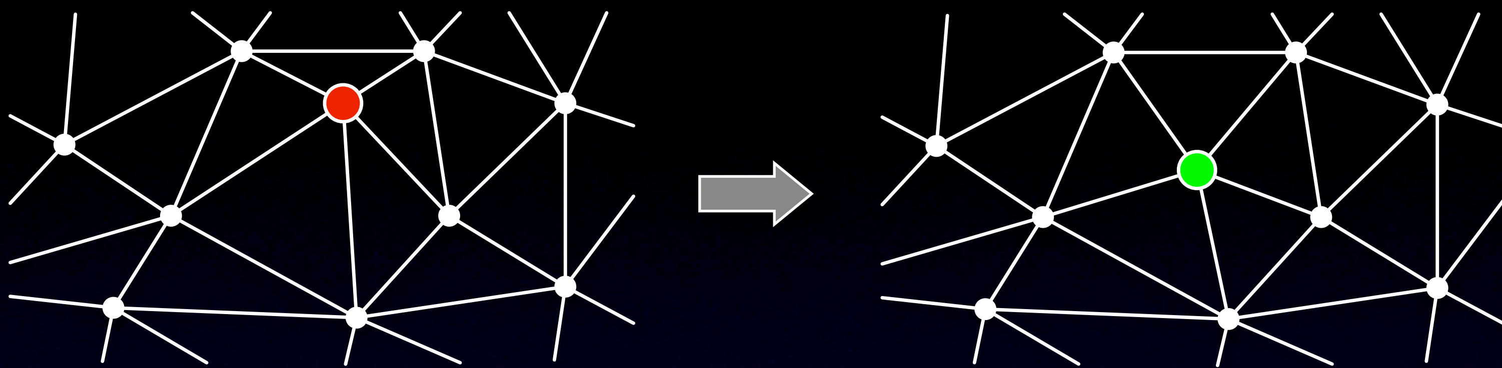
Topological improvement



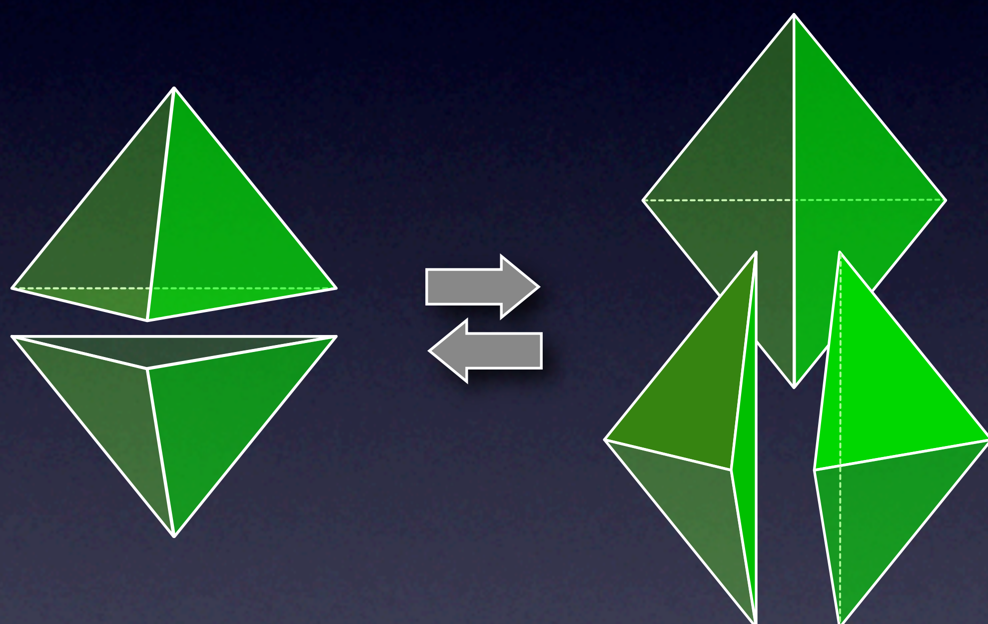
Vertex insertion

new

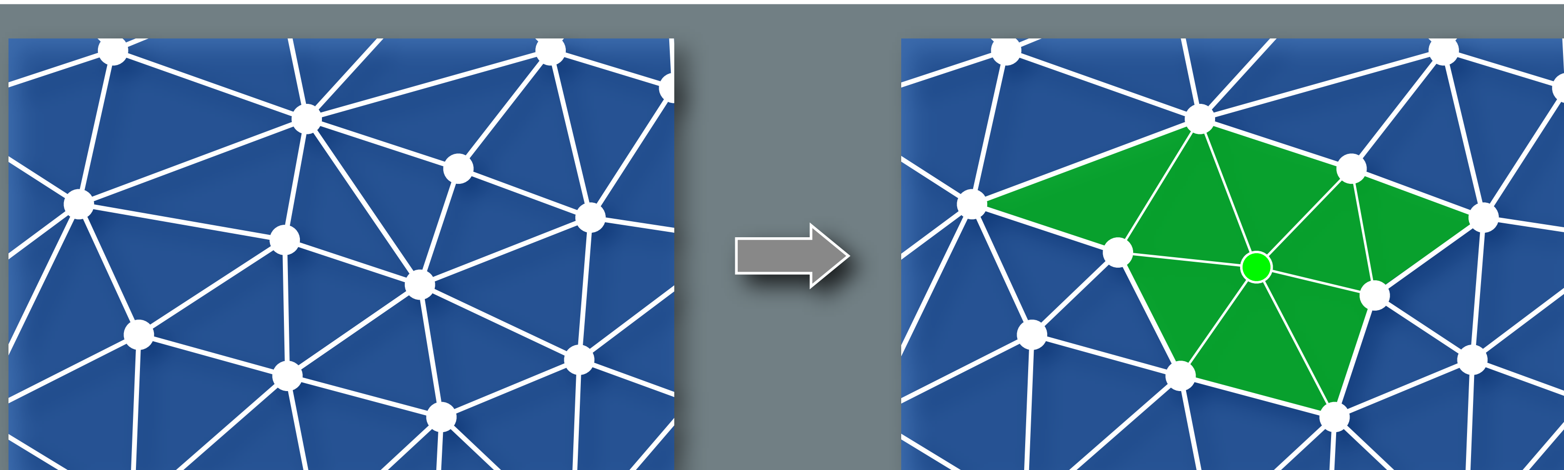
Mesh improvement operations



Vertex smoothing



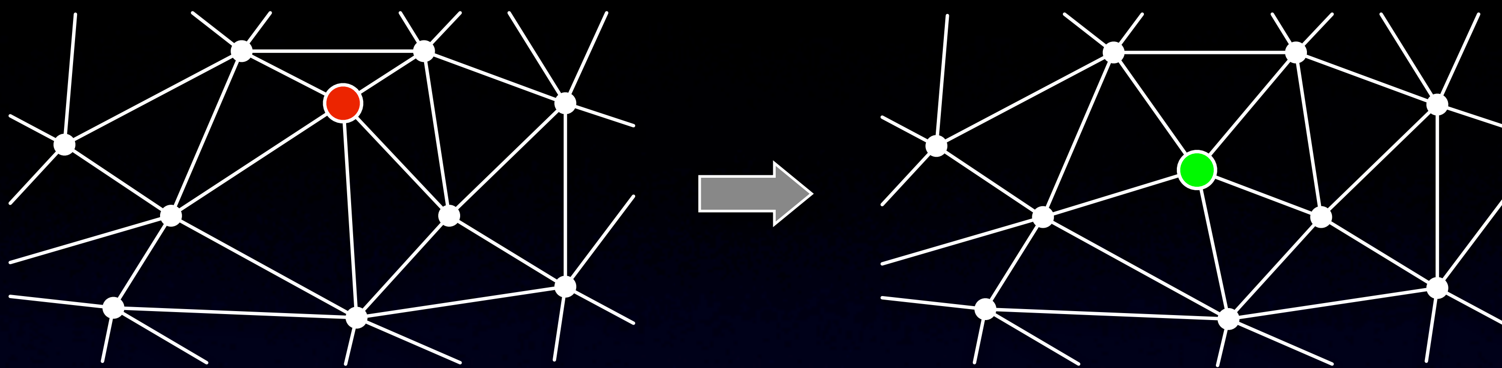
Topological improvement



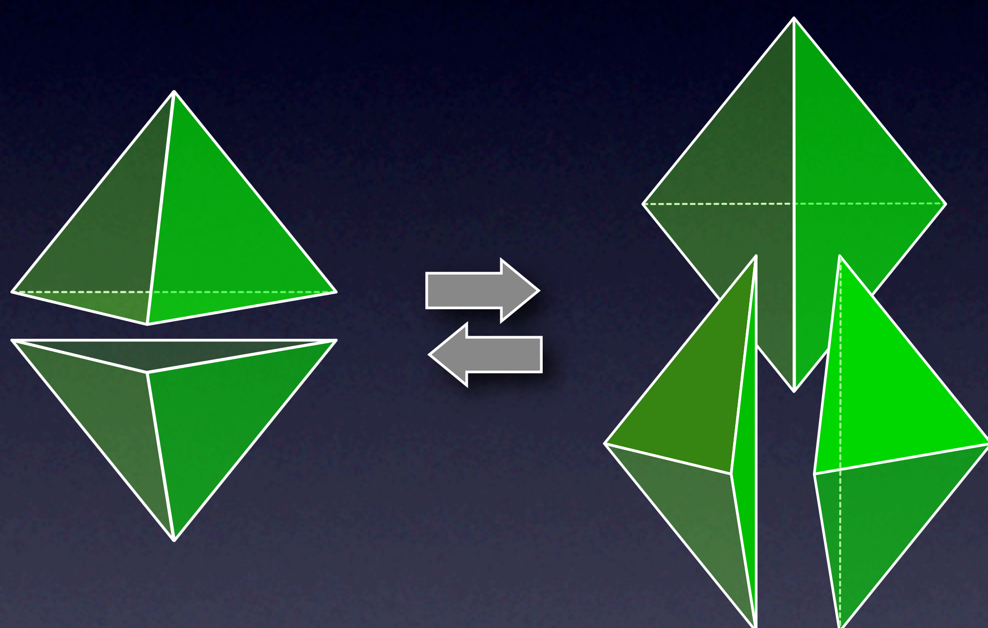
Vertex insertion

new

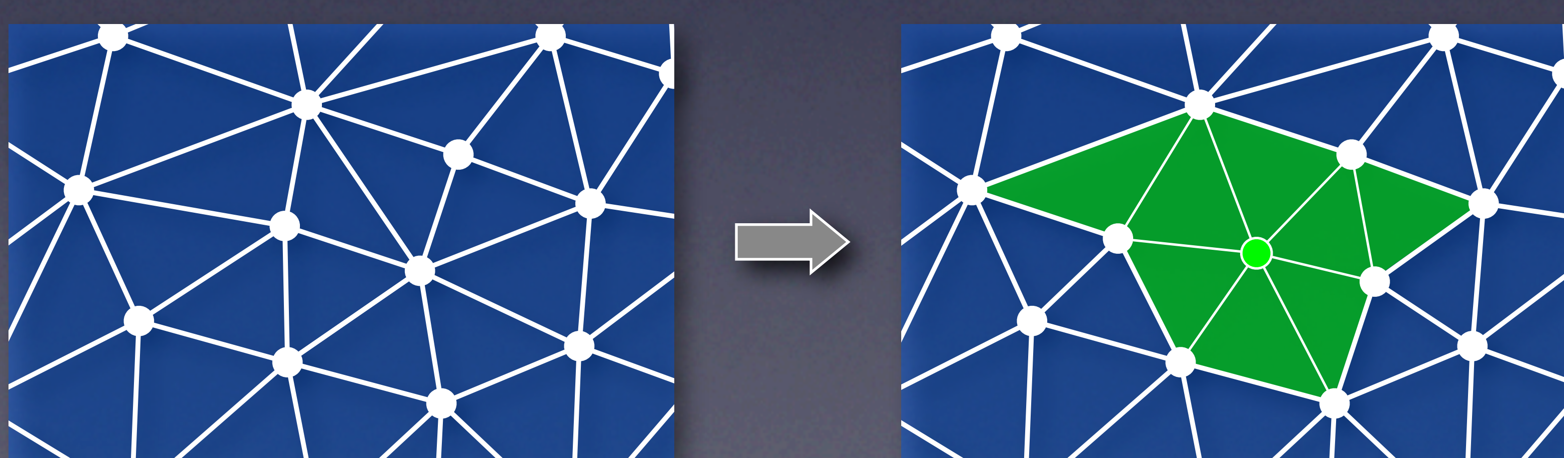
Mesh improvement operations



Vertex smoothing



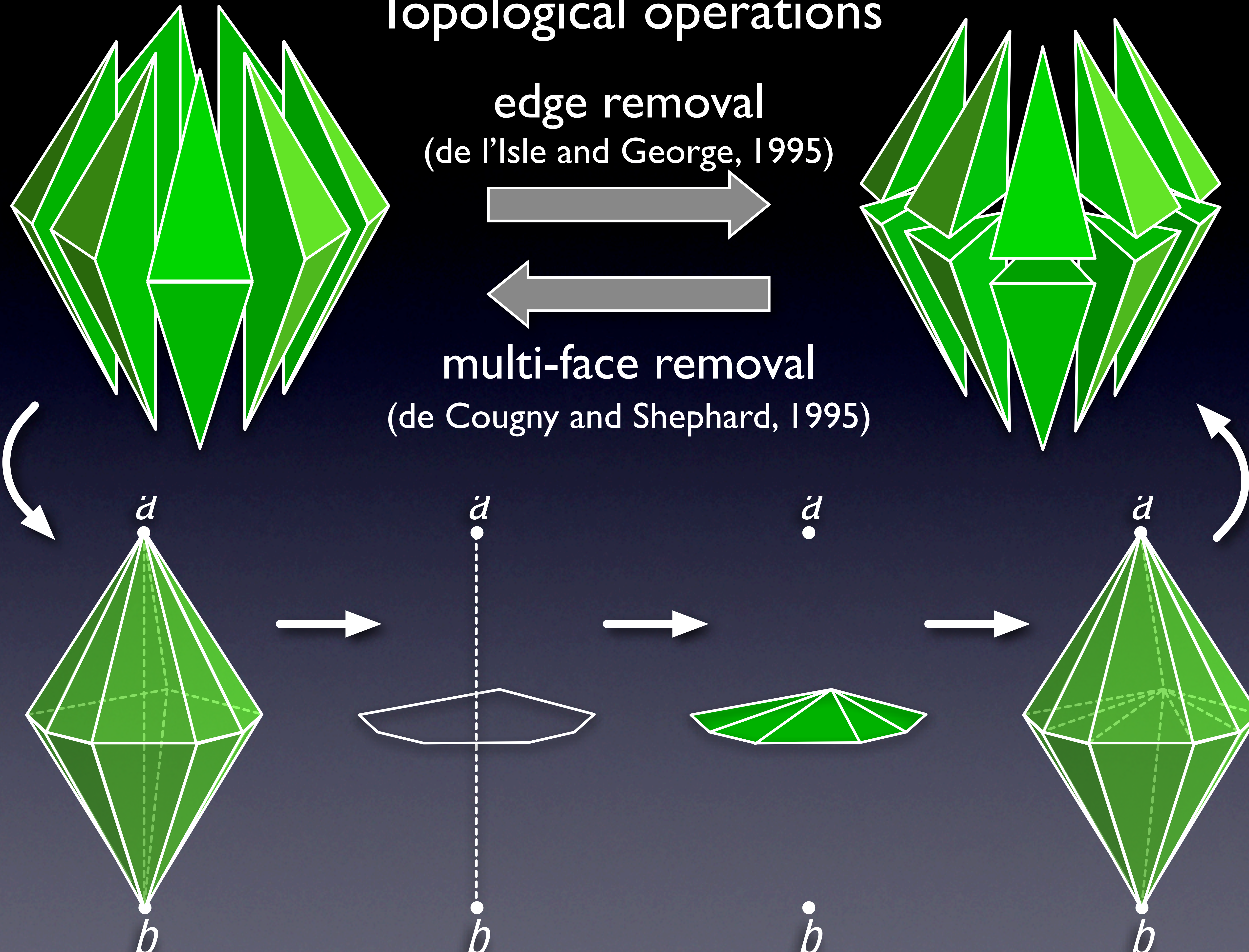
Topological improvement



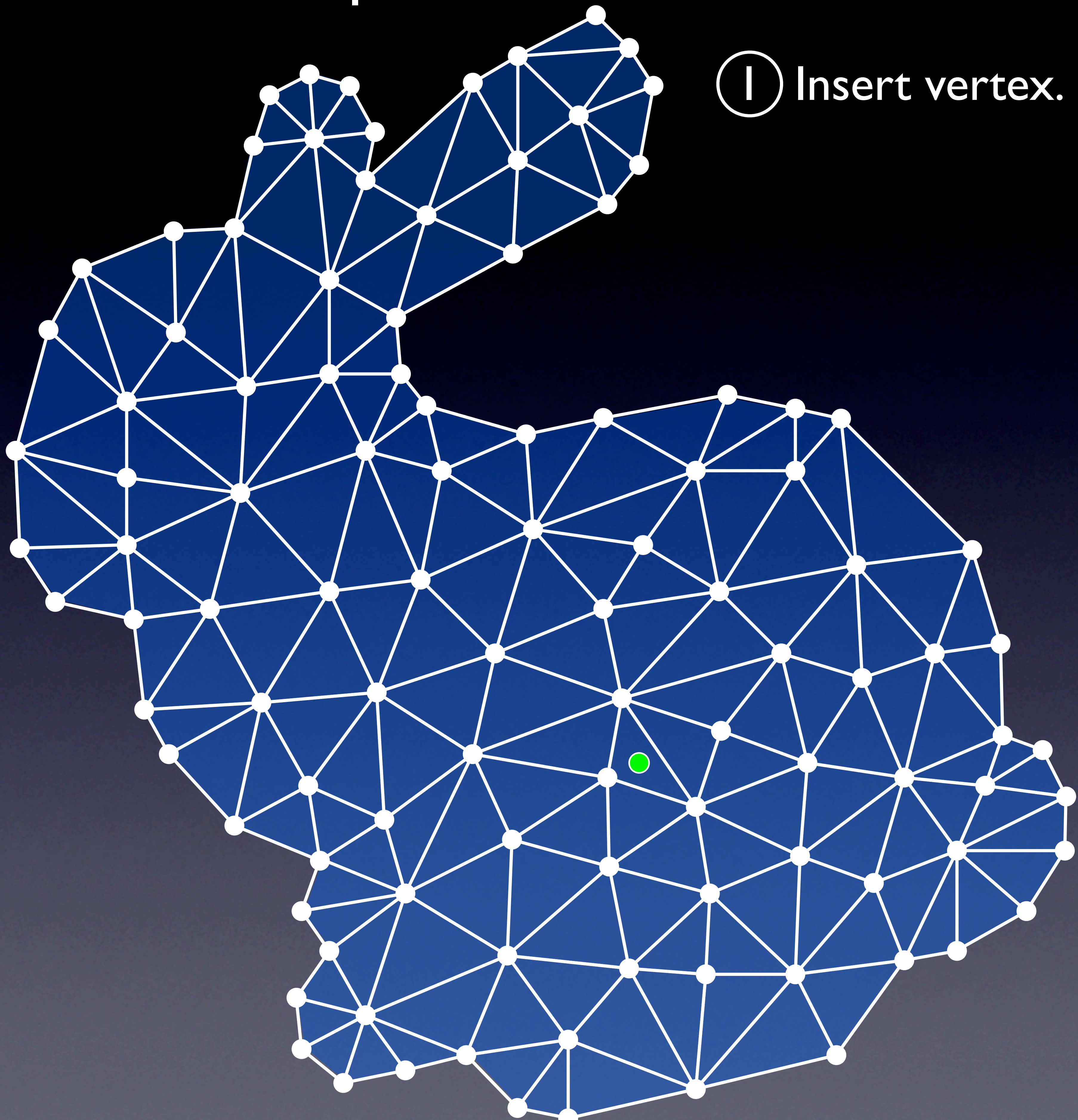
Vertex insertion

new

Topological operations

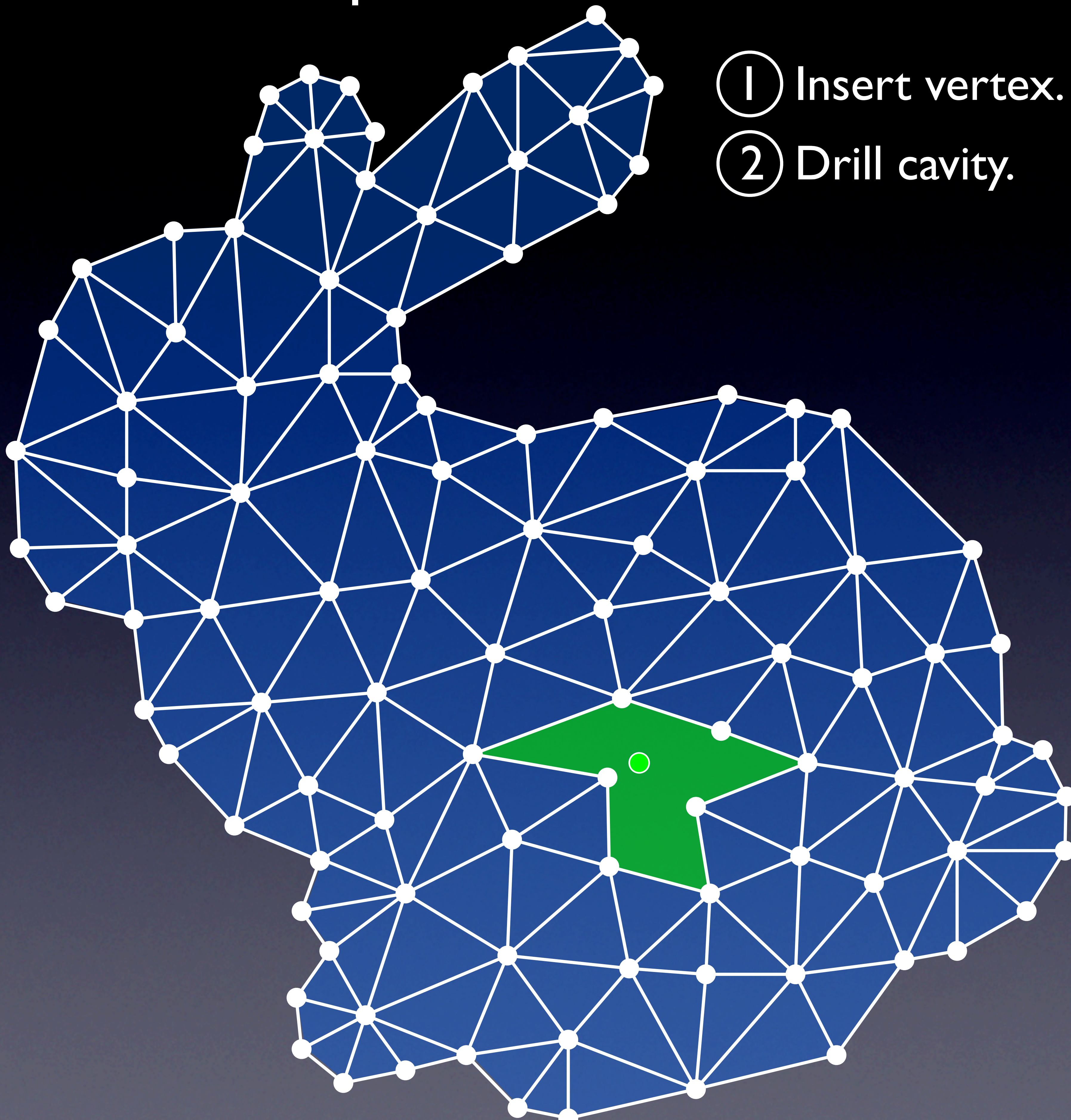


Our new operation: vertex insertion



① Insert vertex.

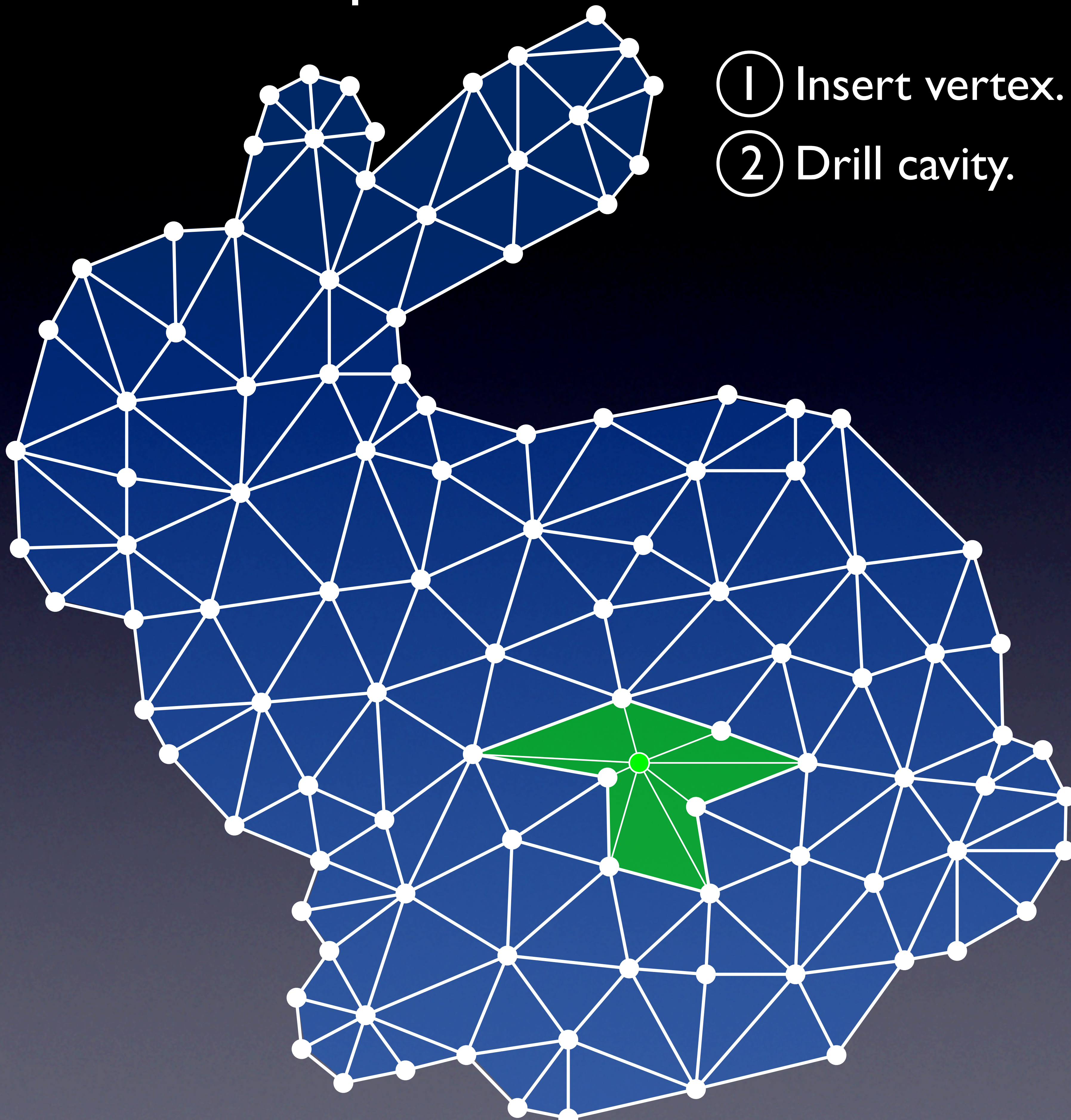
Our new operation: vertex insertion



① Insert vertex.

② Drill cavity.

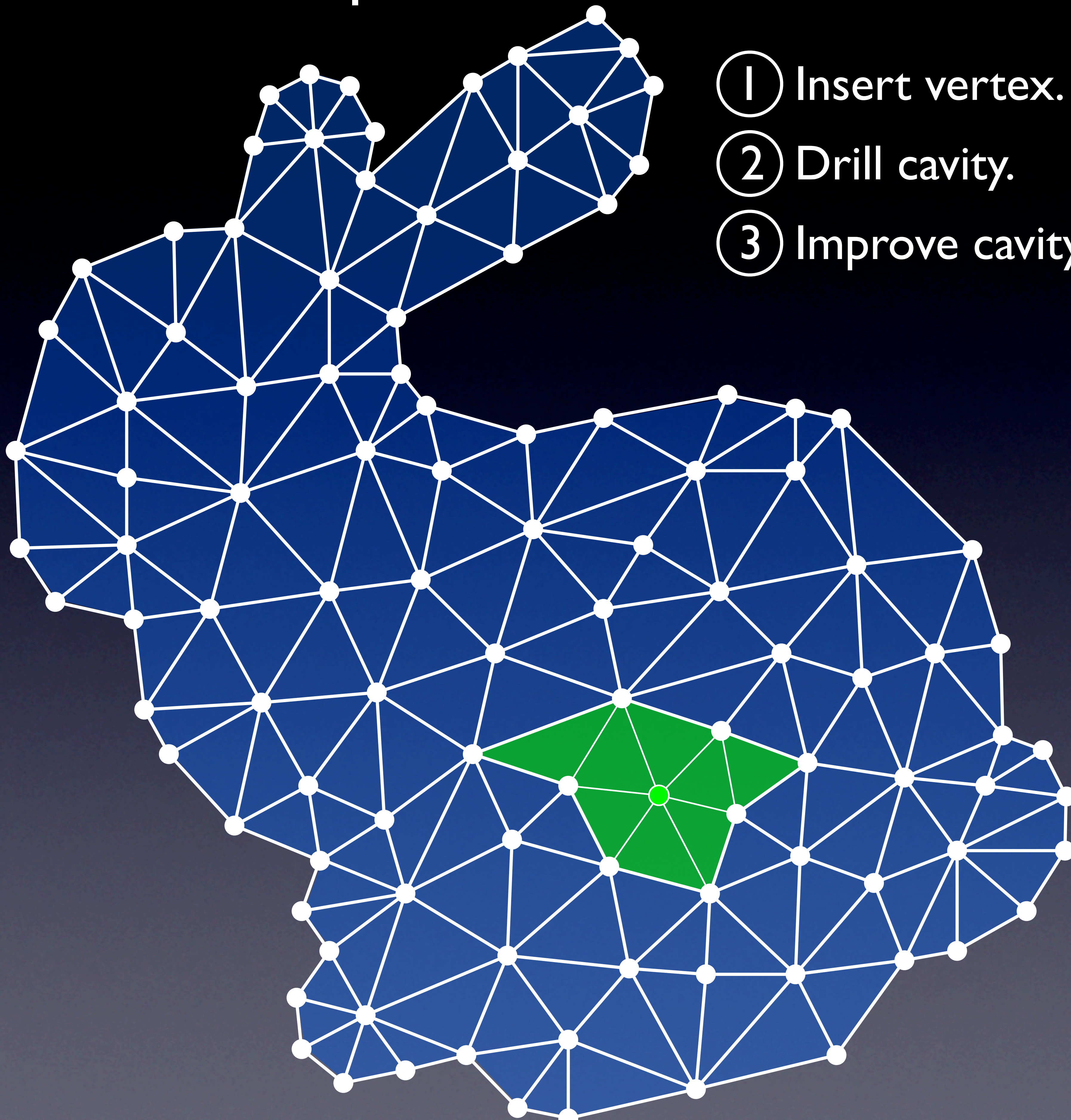
Our new operation: vertex insertion



① Insert vertex.

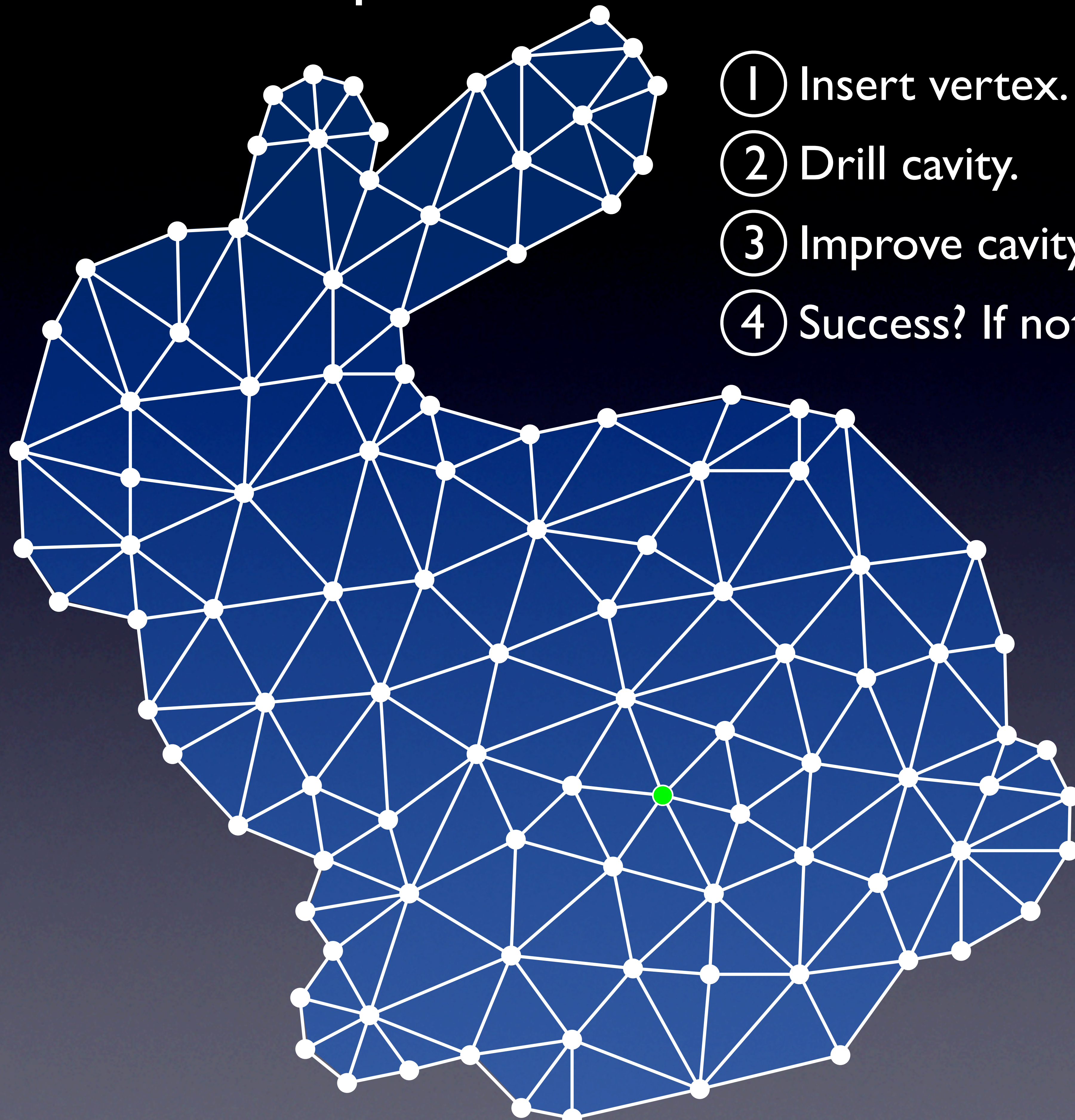
② Drill cavity.

Our new operation: vertex insertion



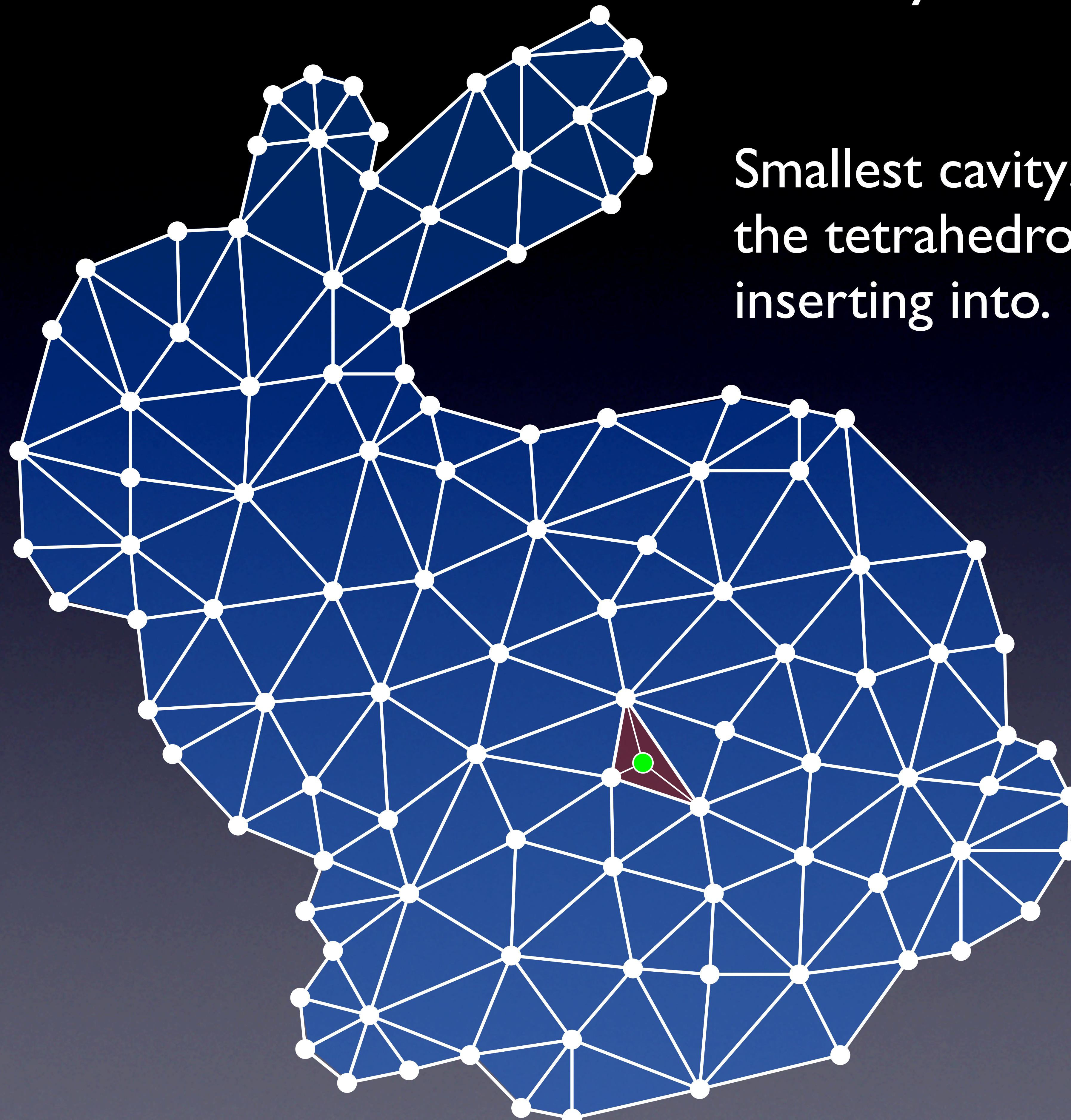
1. Insert vertex.
2. Drill cavity.
3. Improve cavity.

Our new operation: vertex insertion



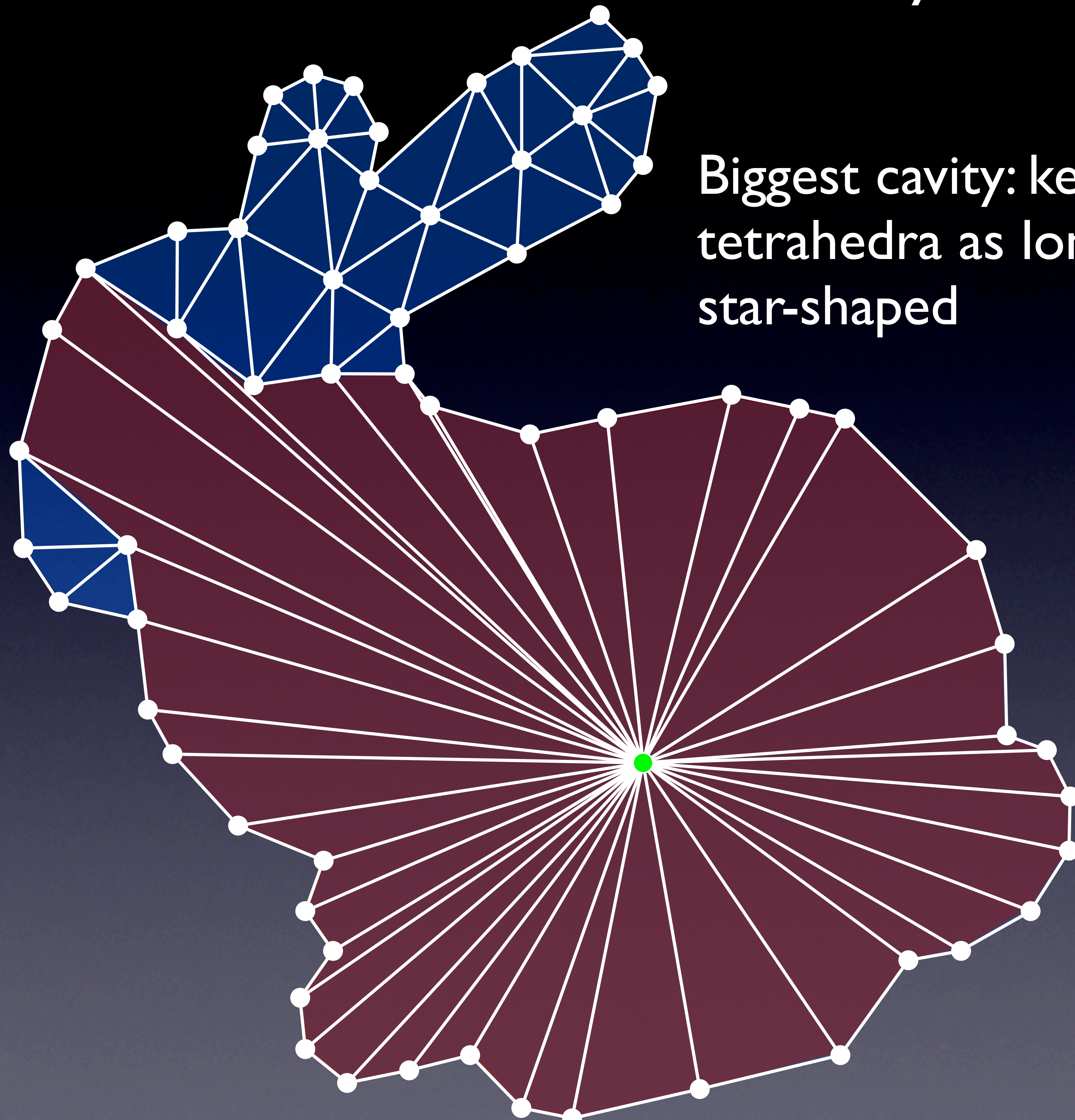
1. Insert vertex.
2. Drill cavity.
3. Improve cavity.
4. Success? If not, roll back.

How do we choose the cavity?



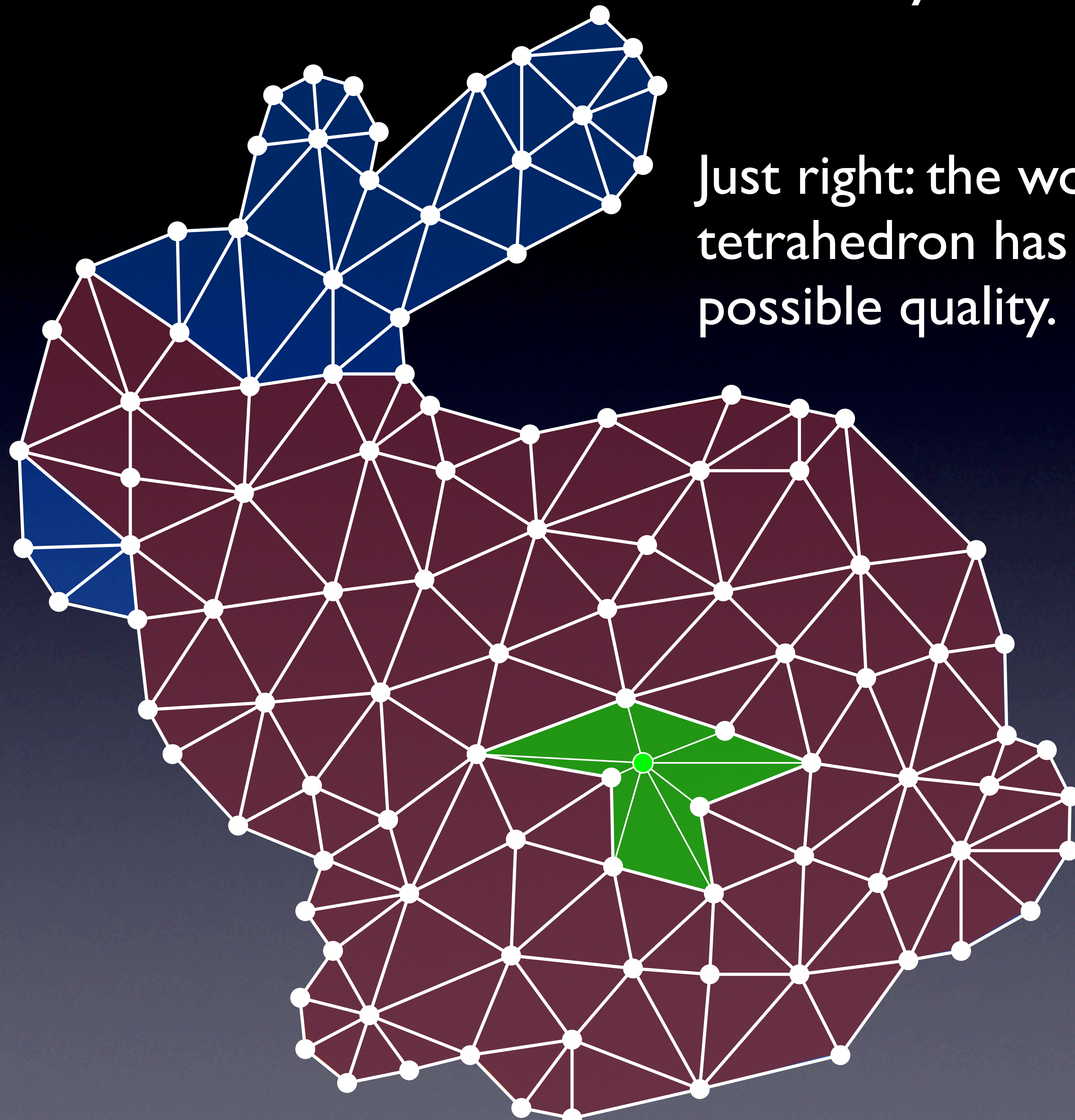
Smallest cavity: just split the tetrahedron we're inserting into.

How do we choose the cavity?



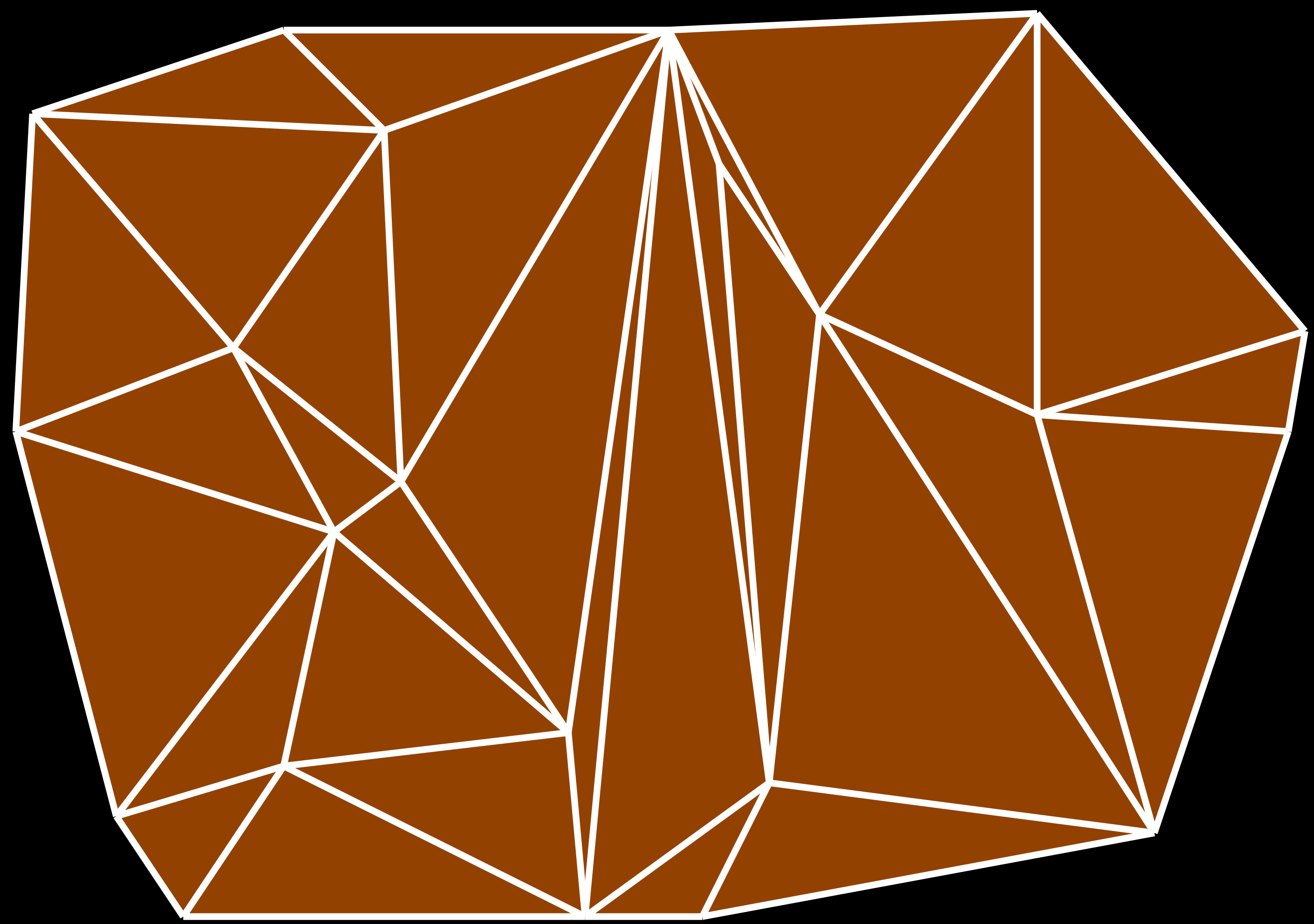
Biggest cavity: keep adding tetrahedra as long as it's star-shaped

How do we choose the cavity?

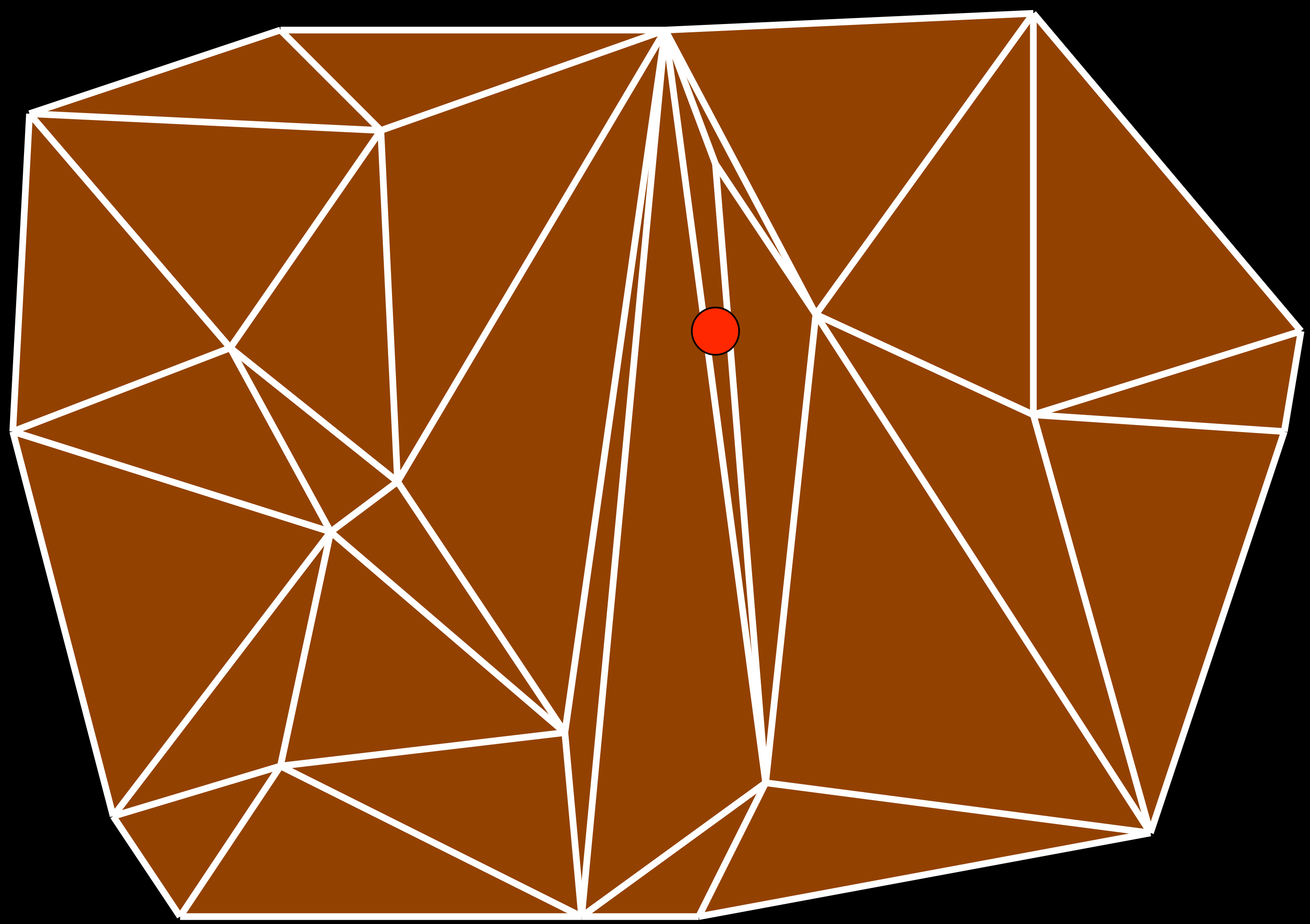


Just right: the worst new tetrahedron has the best possible quality.

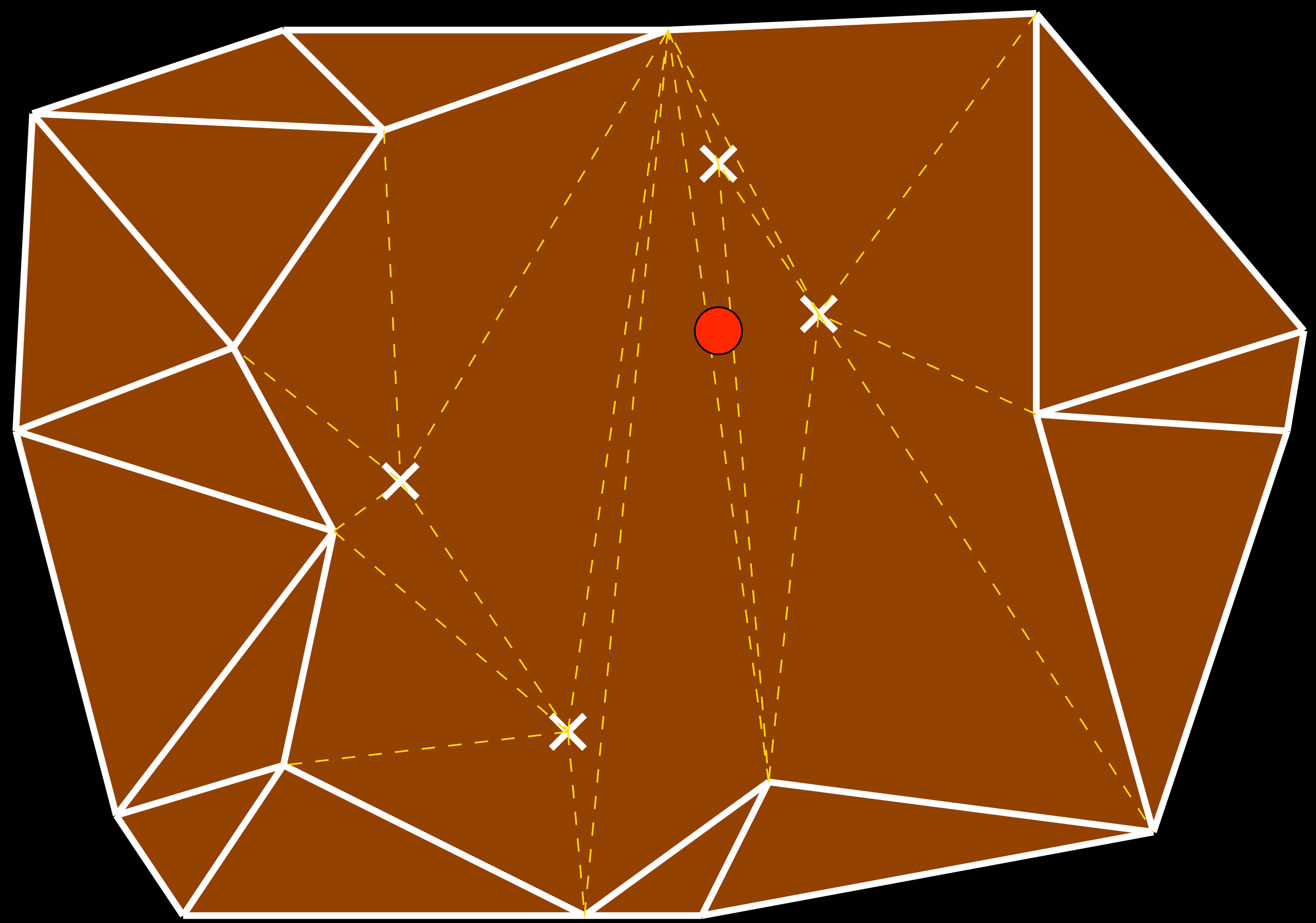
Finding the optimal cavity



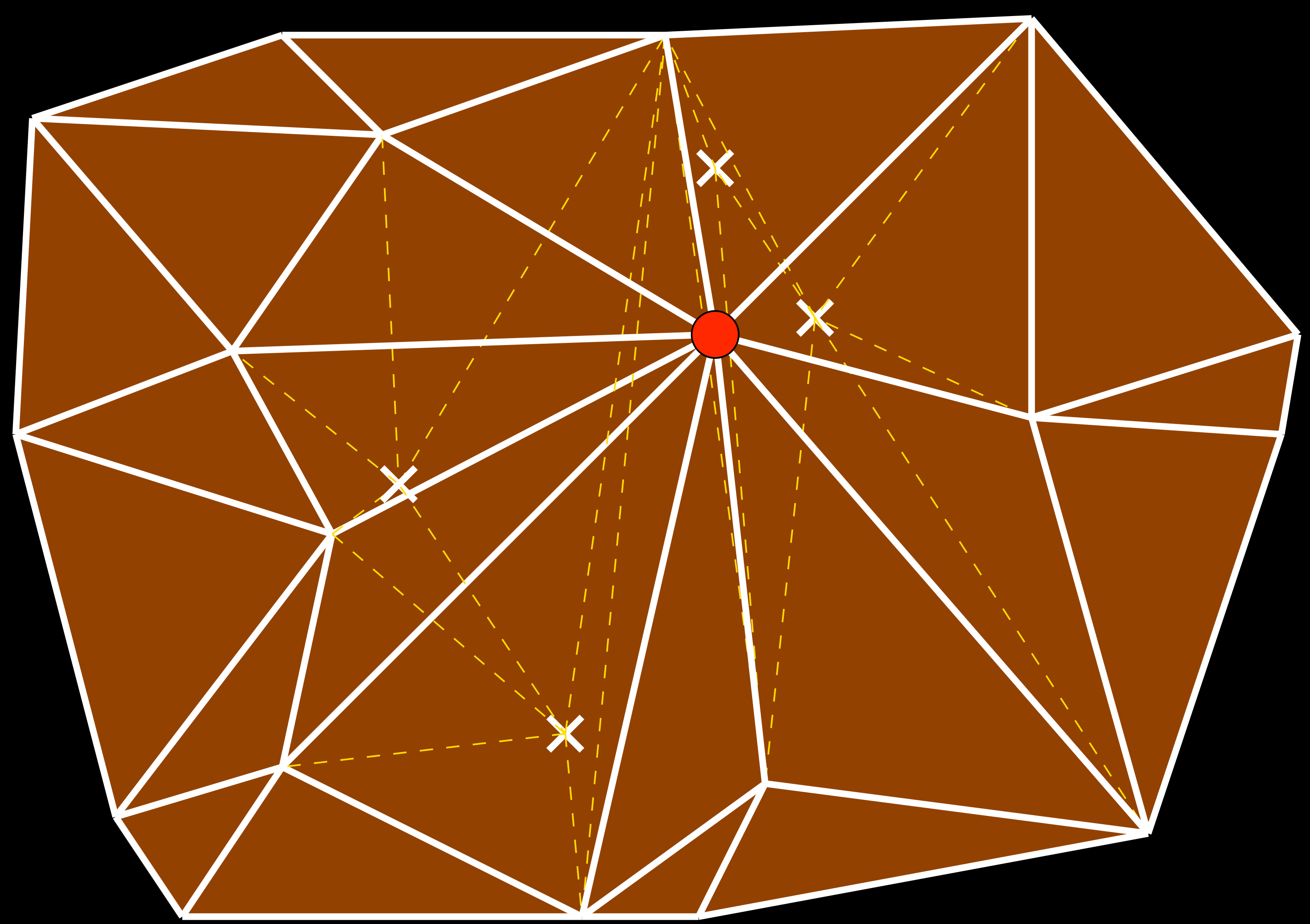
Finding the optimal cavity



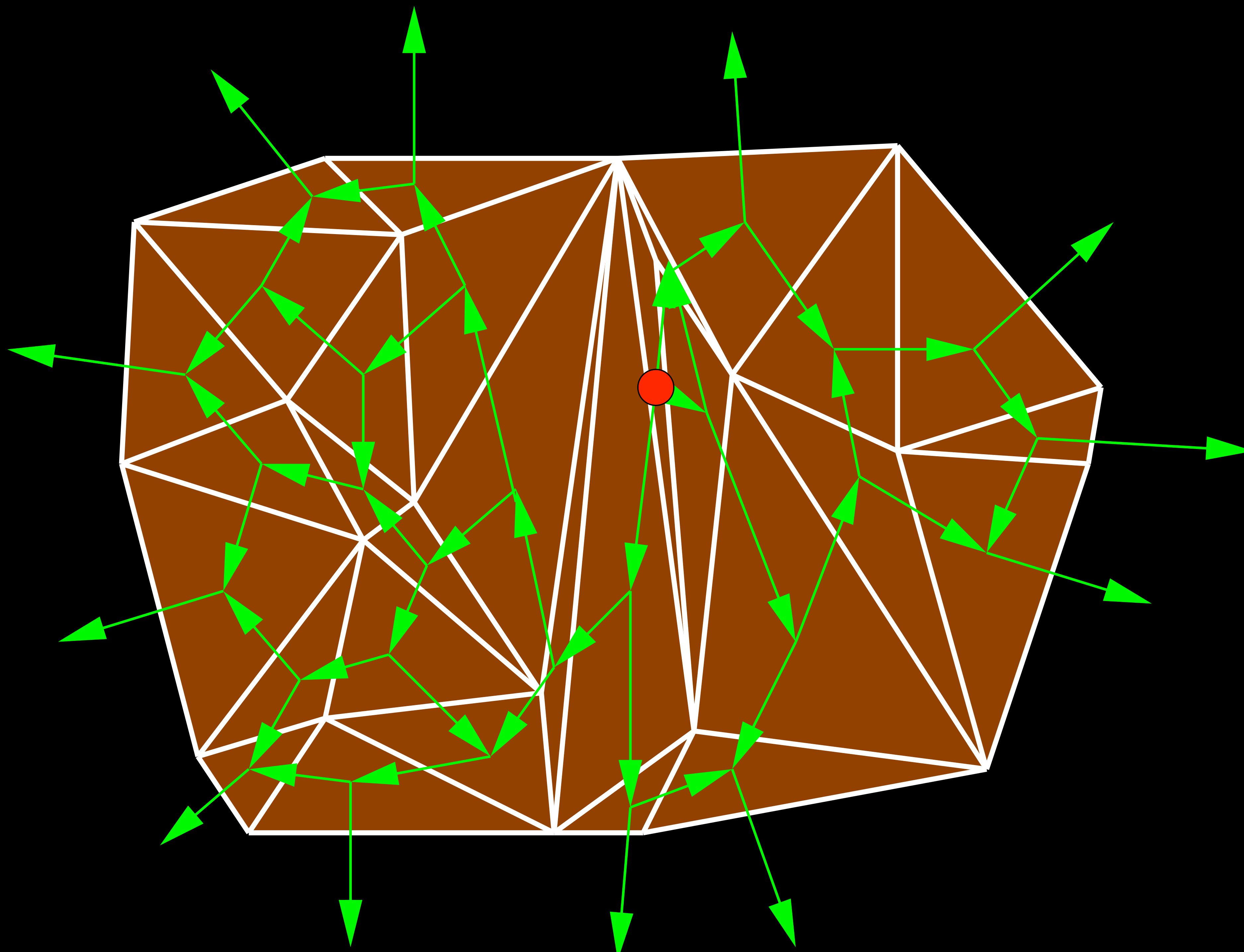
Finding the optimal cavity



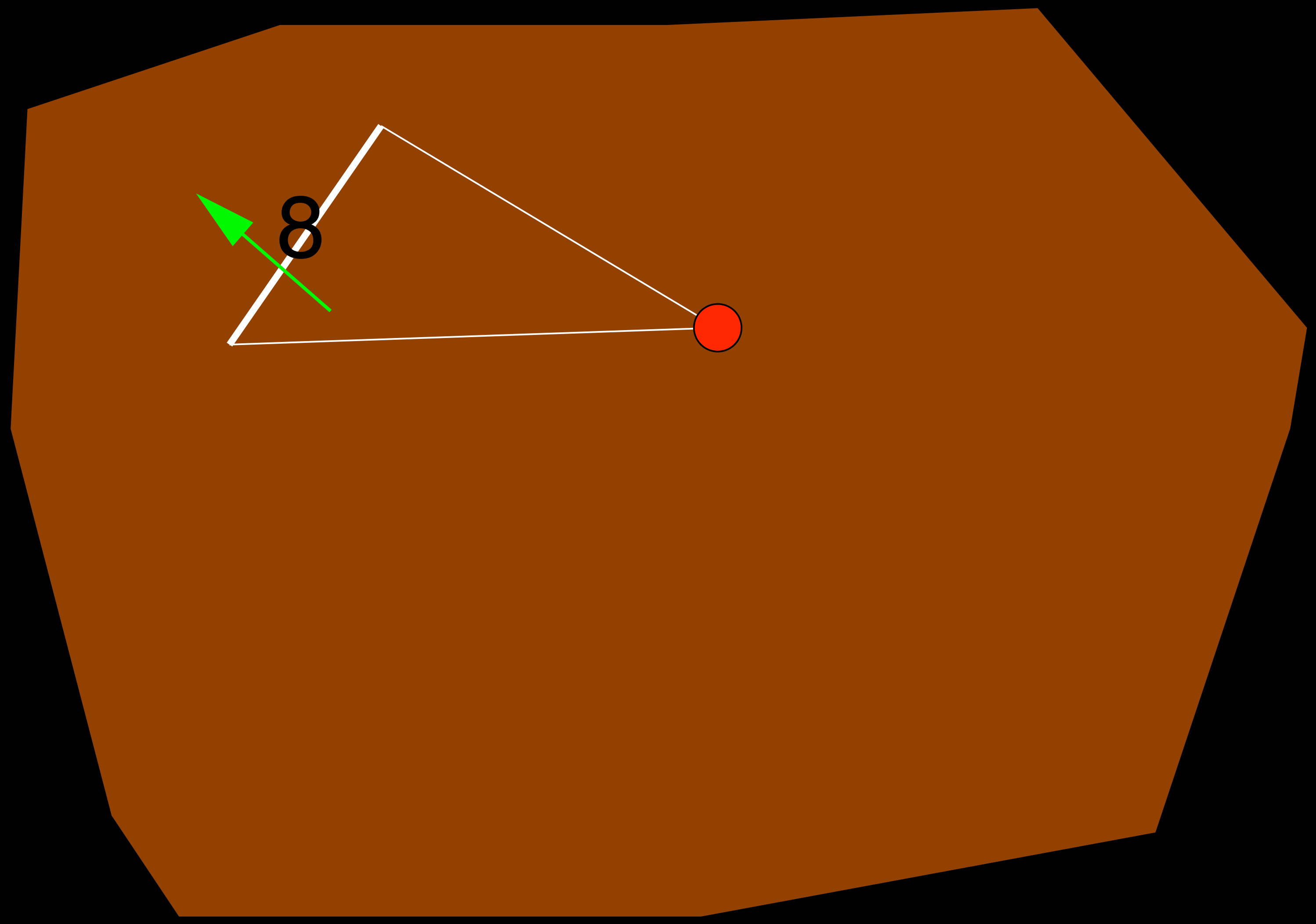
Finding the optimal cavity



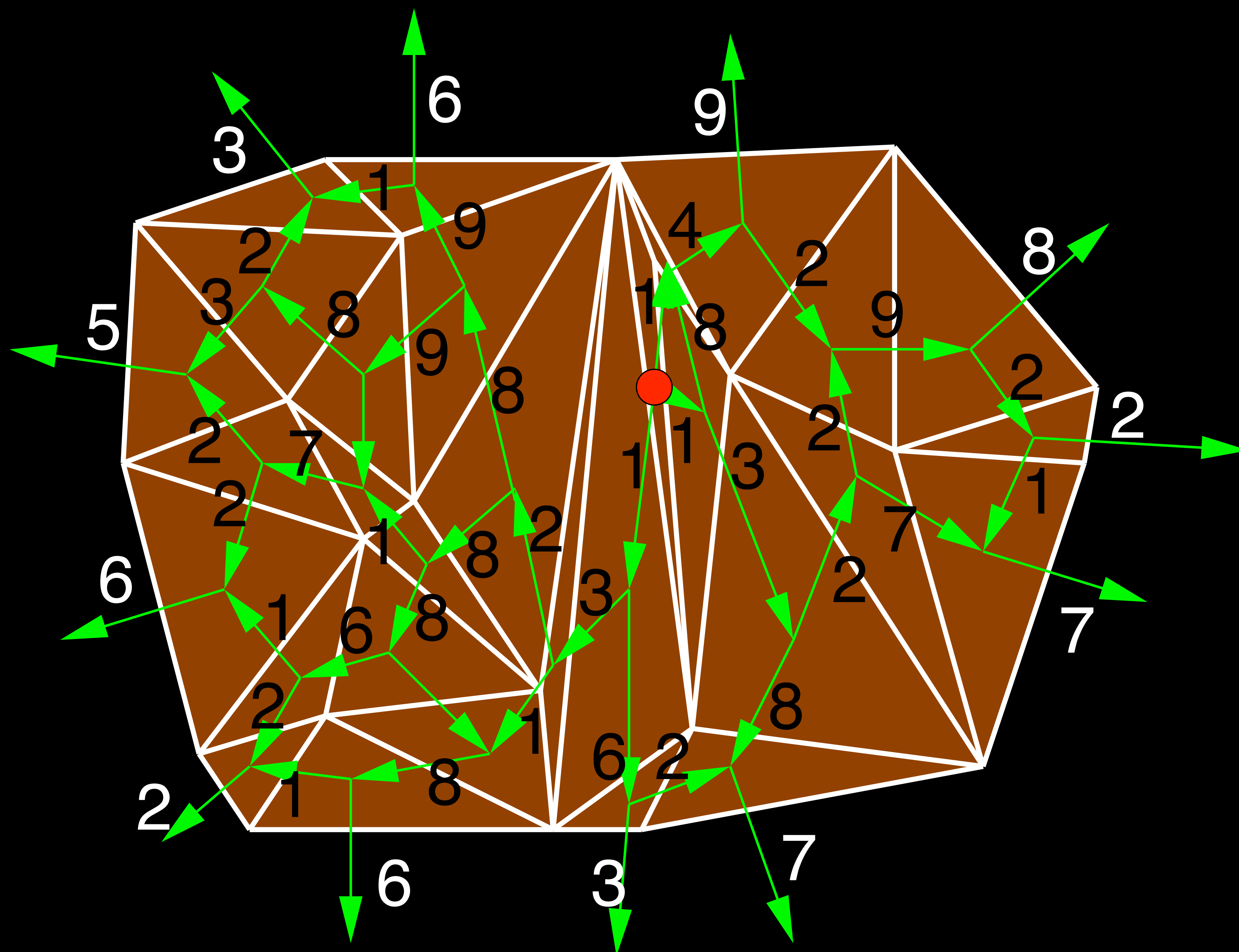
Finding the optimal cavity



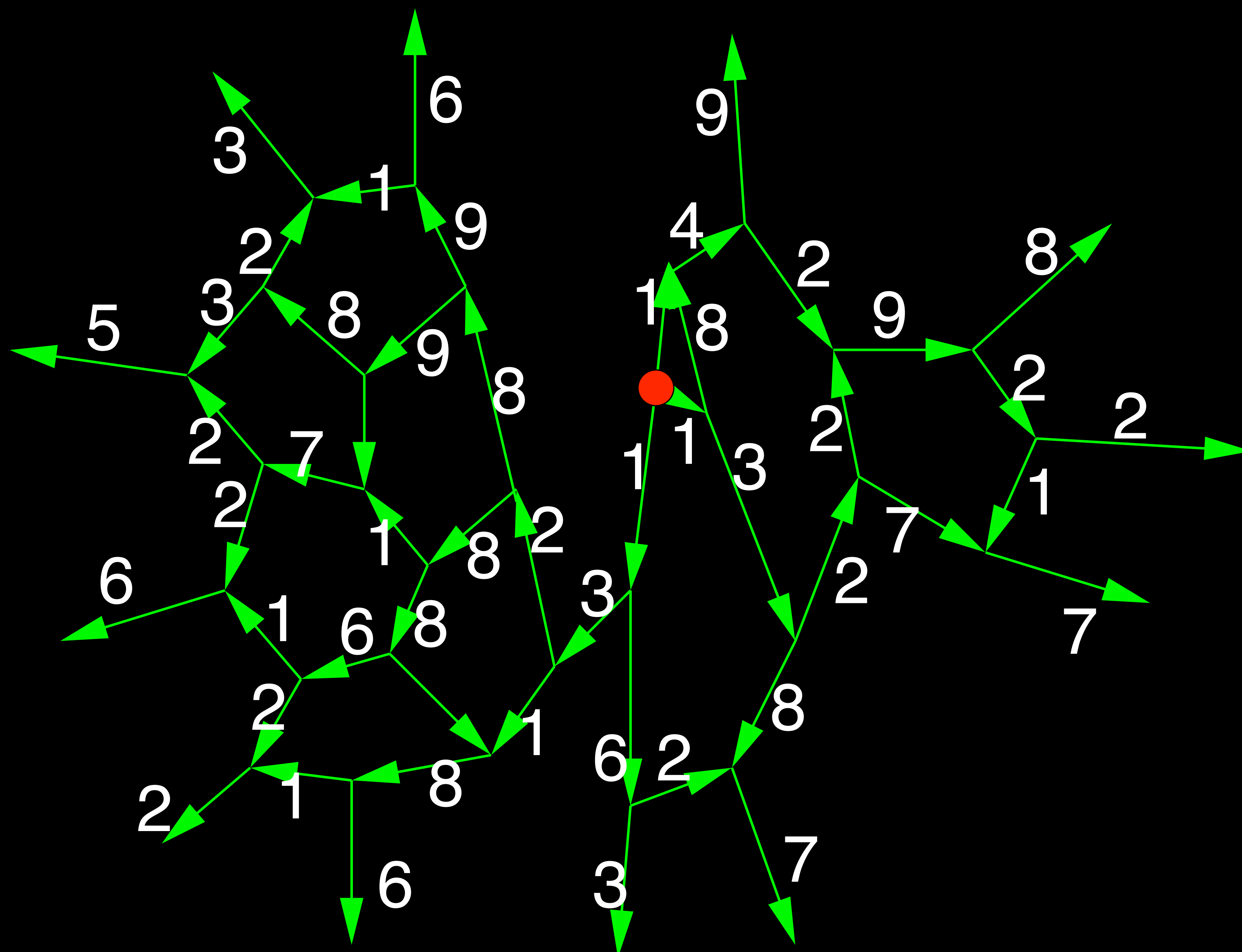
Finding the optimal cavity



Finding the optimal cavity

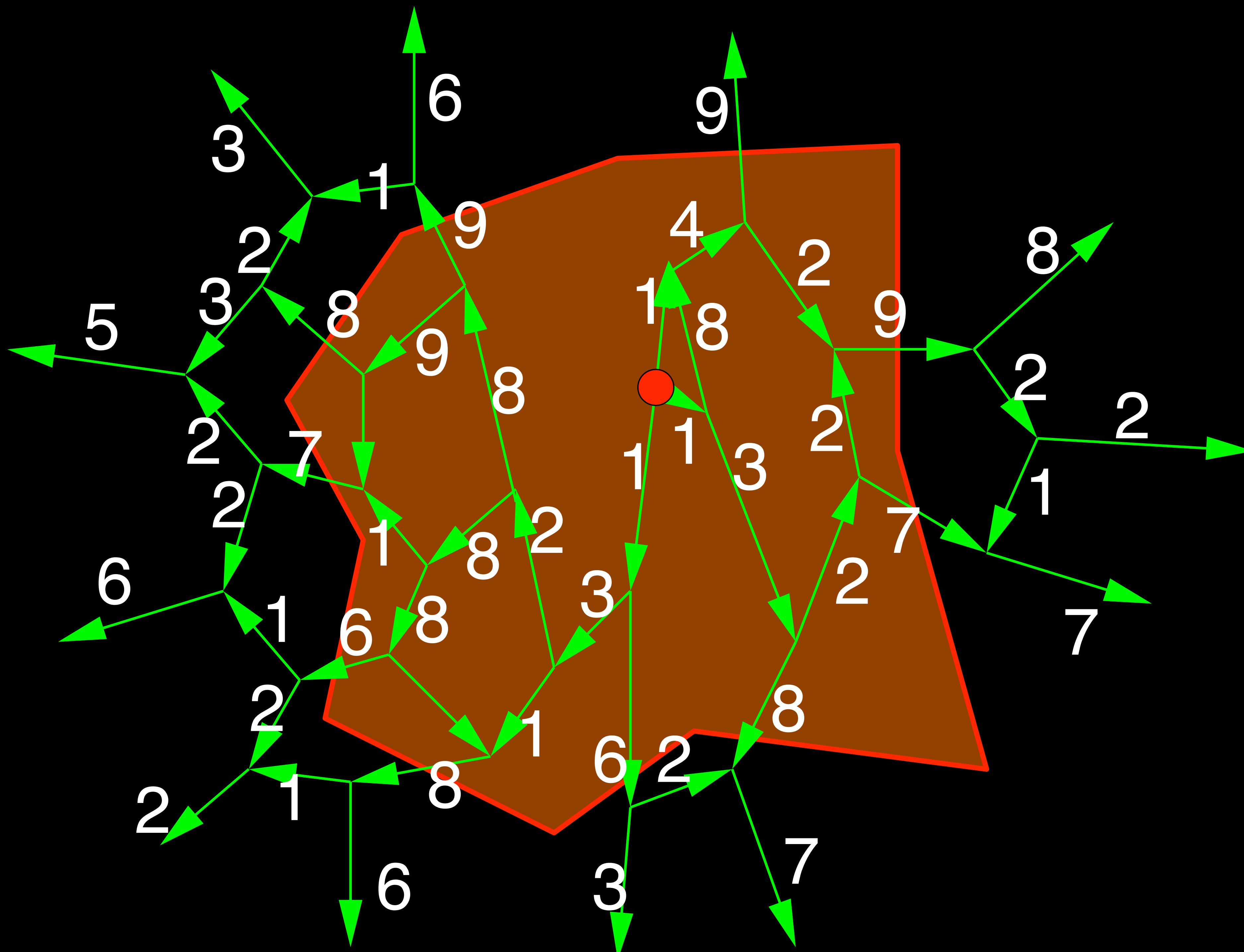


Finding the optimal cavity

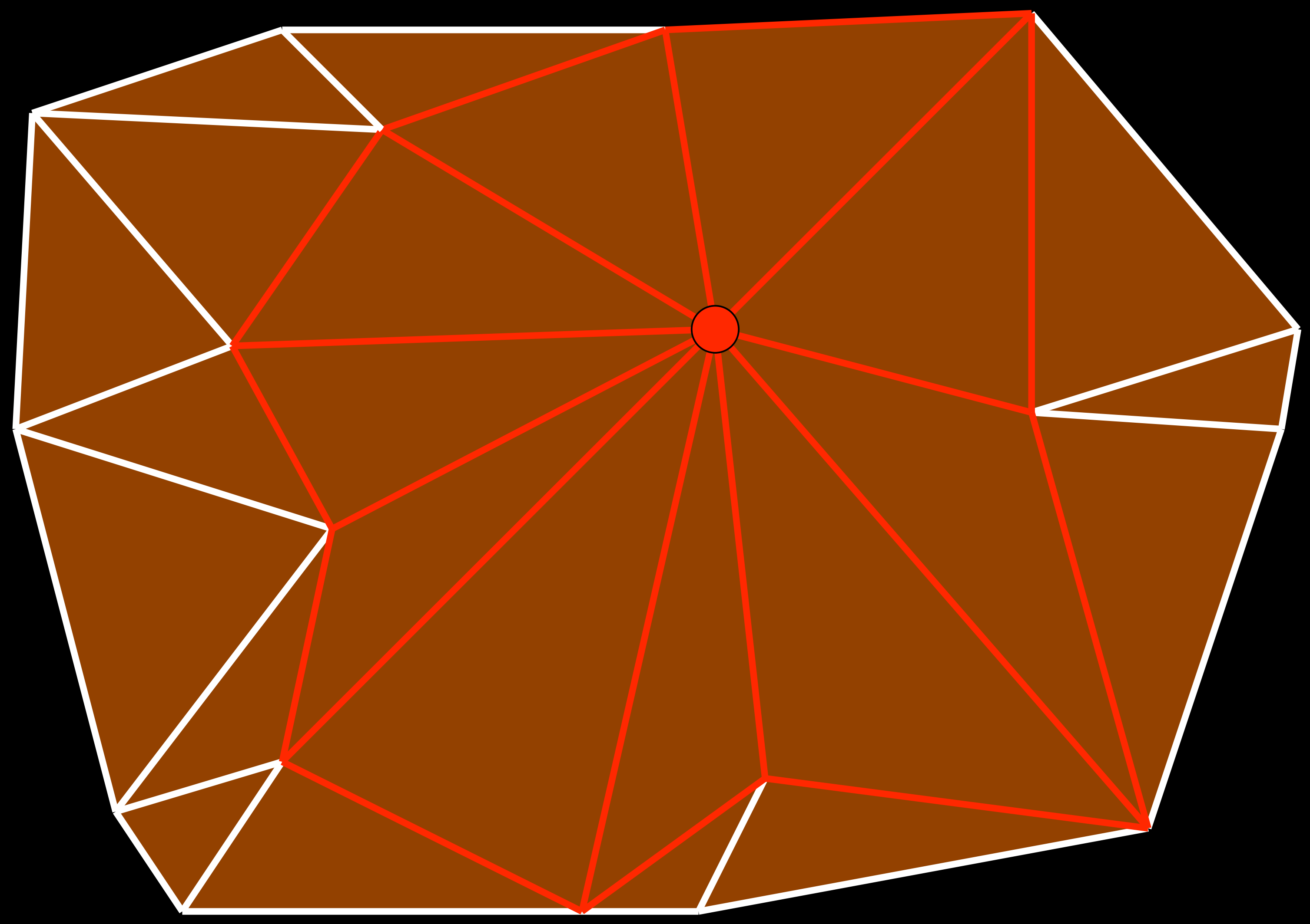


Find the cut between root and leaves
that maximizes the smallest edge.

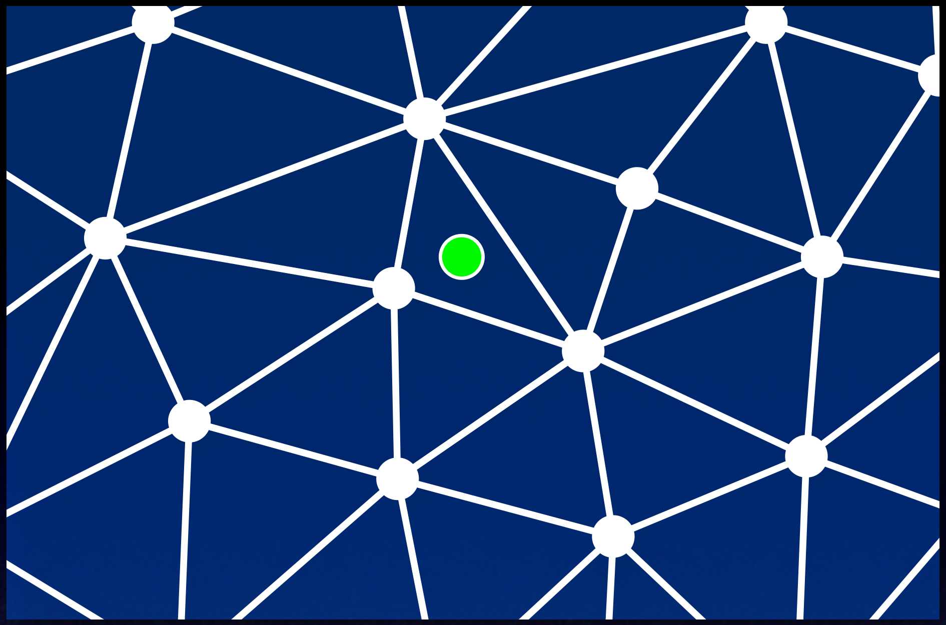
Finding the optimal cavity



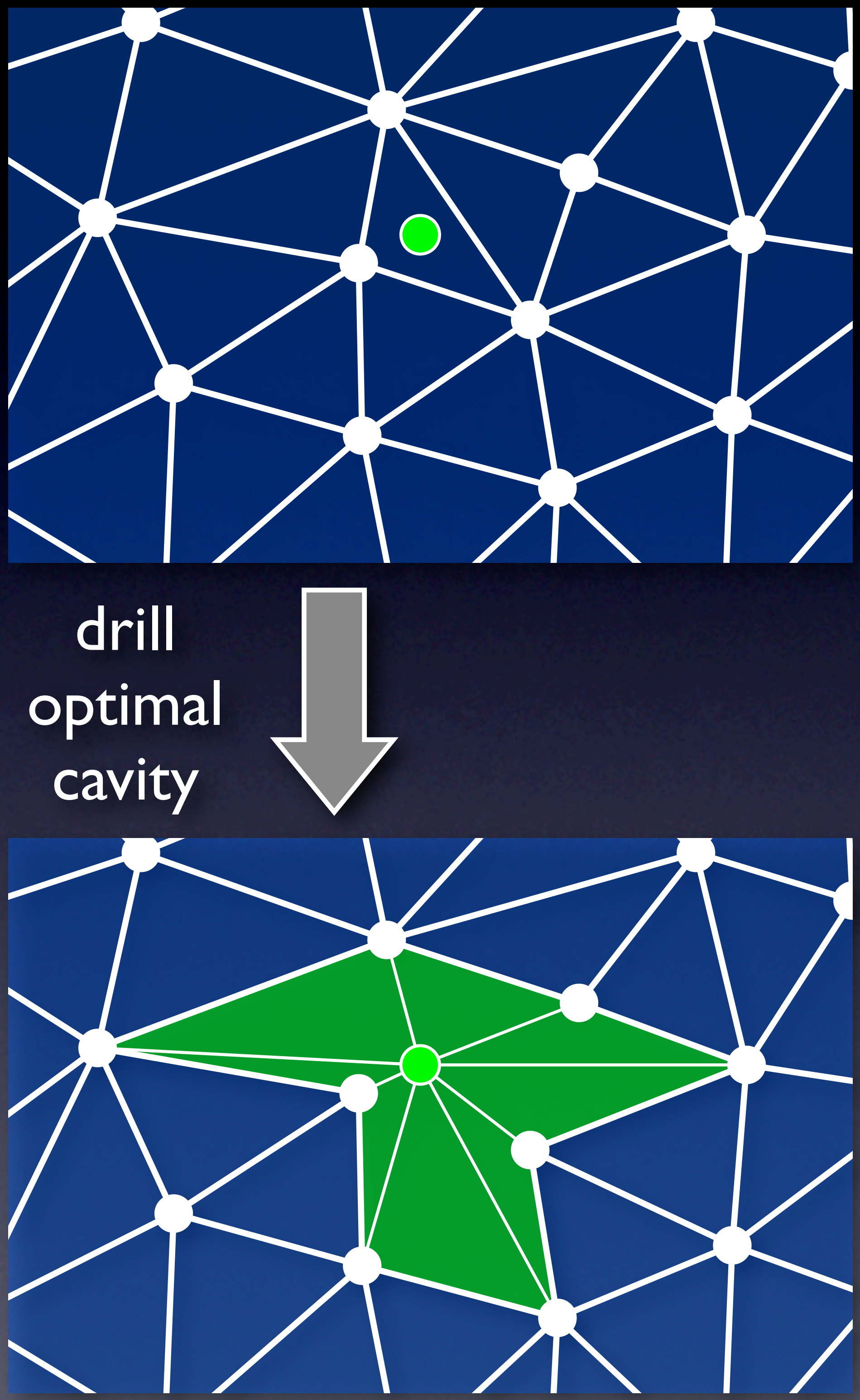
Finding the optimal cavity



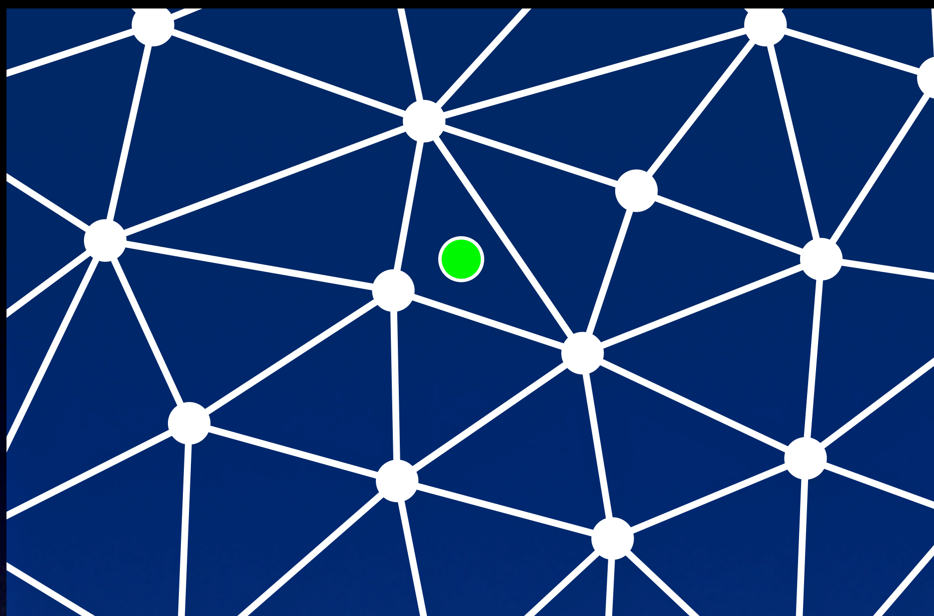
Cavity improvement



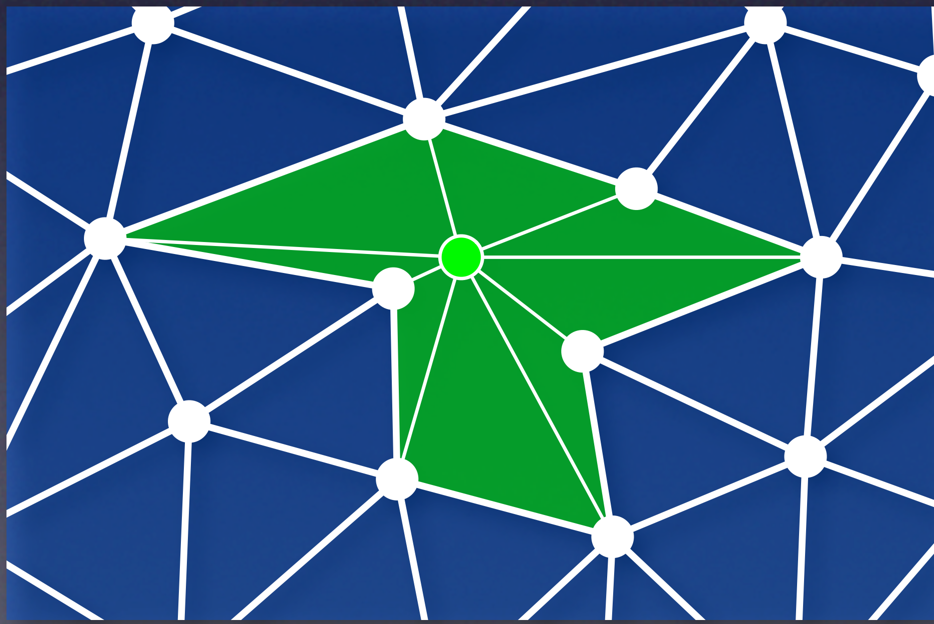
Cavity improvement



Cavity improvement

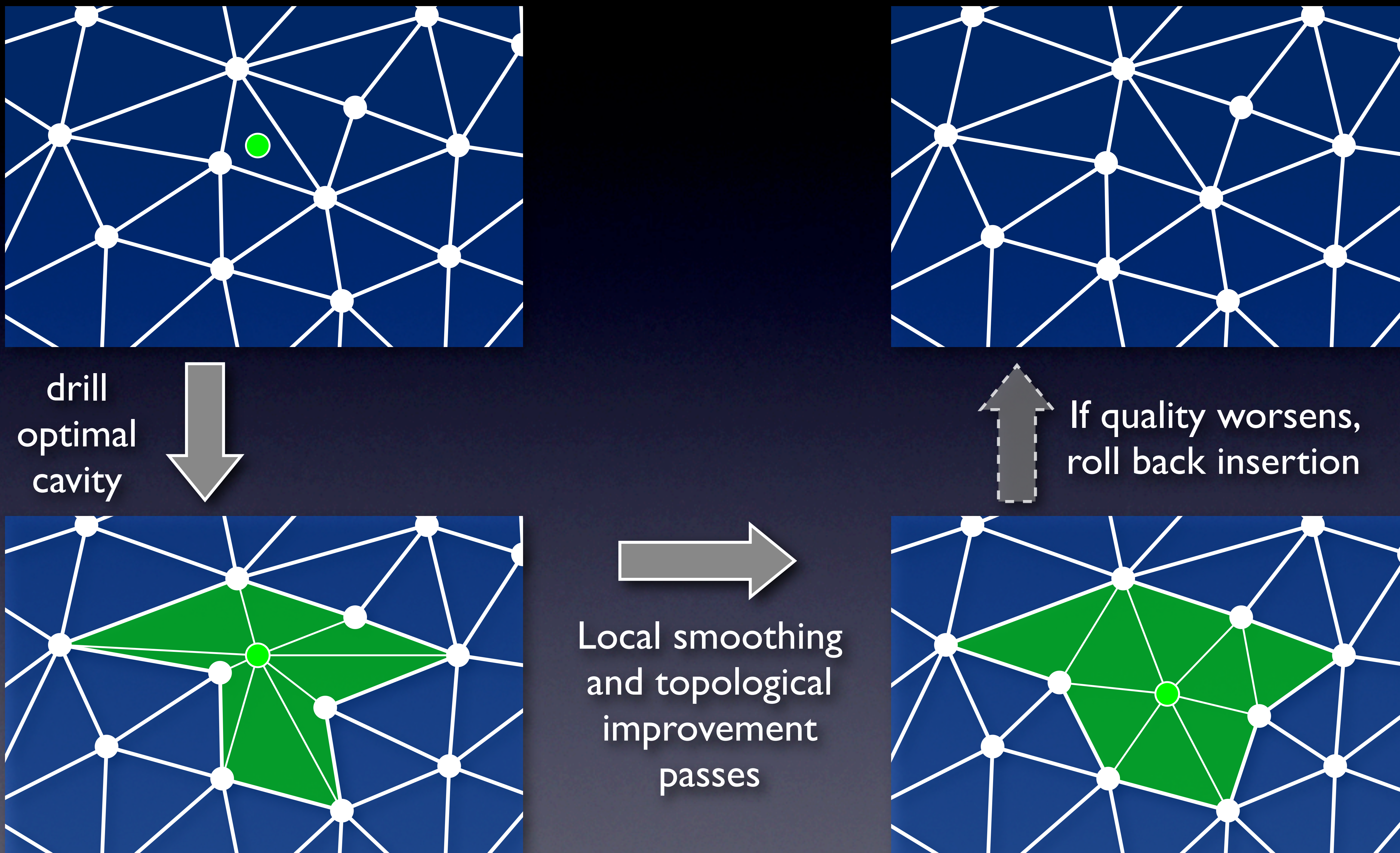


drill
optimal
cavity

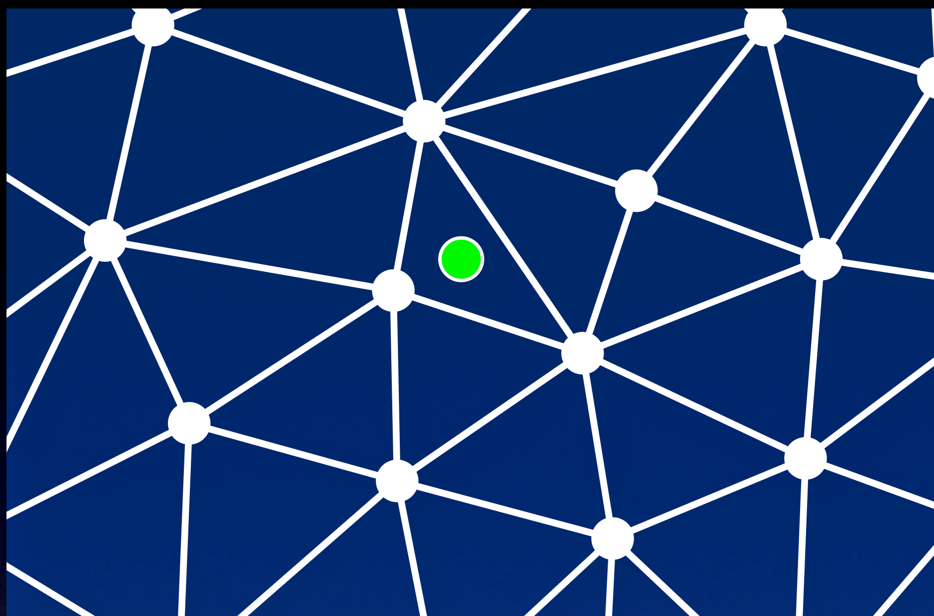


Local smoothing
and topological
improvement
passes

Cavity improvement



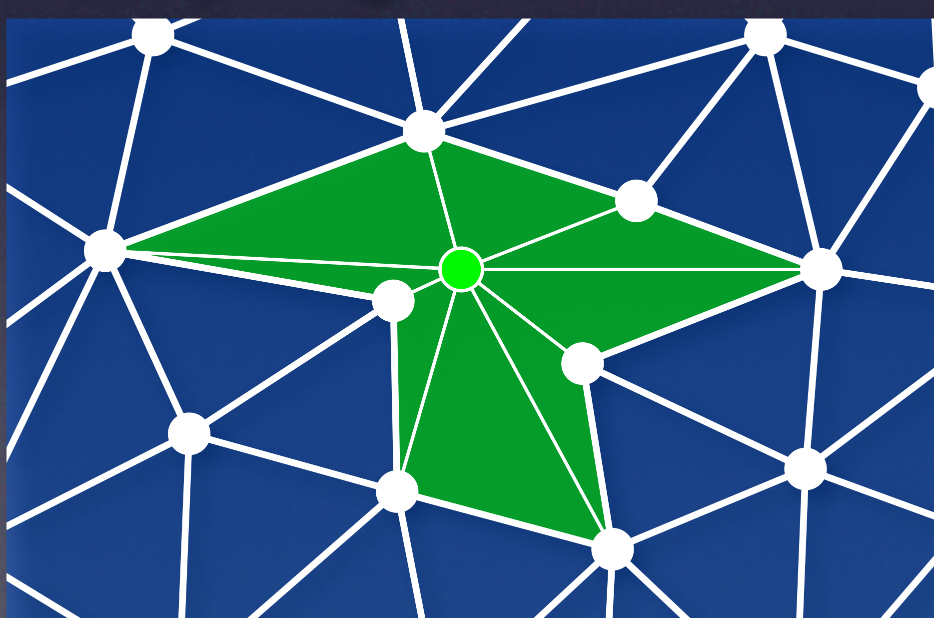
Insertion timing



drill optimal cavity

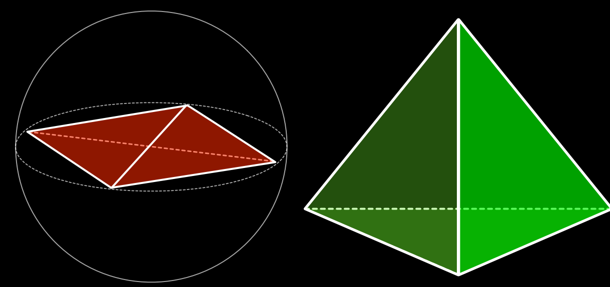
22% time finding biggest,
<1% time finding optimal

If quality worsens,
roll back insertion

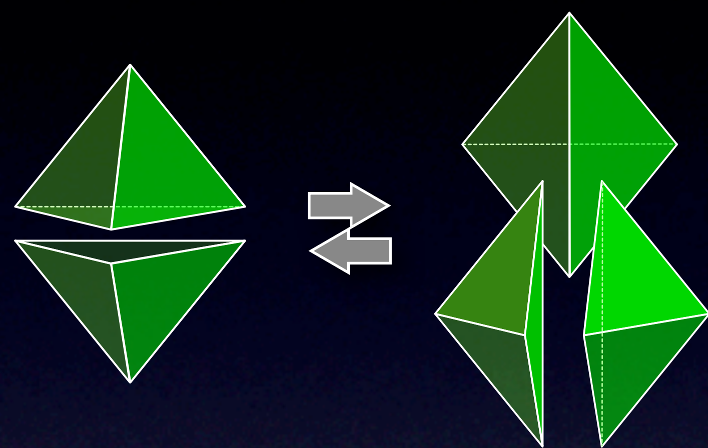


Local smoothing and topological improvement passes

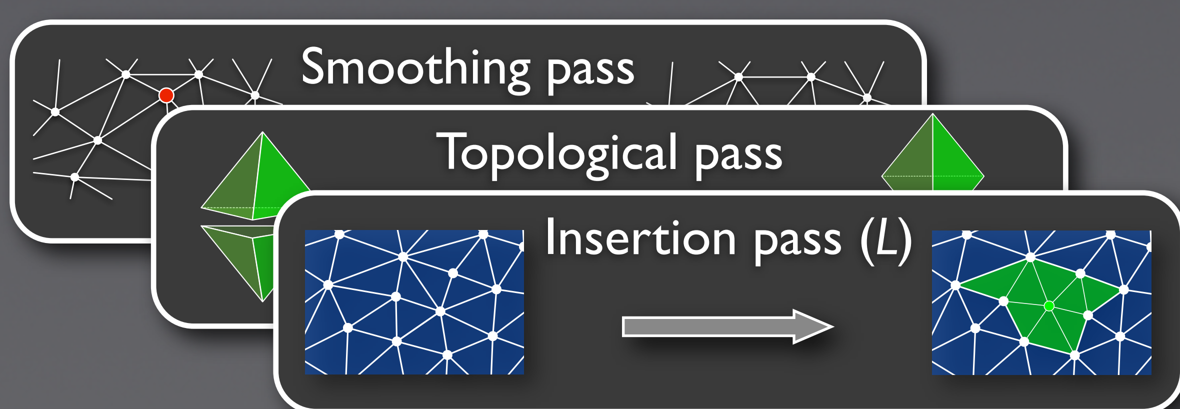
77% time



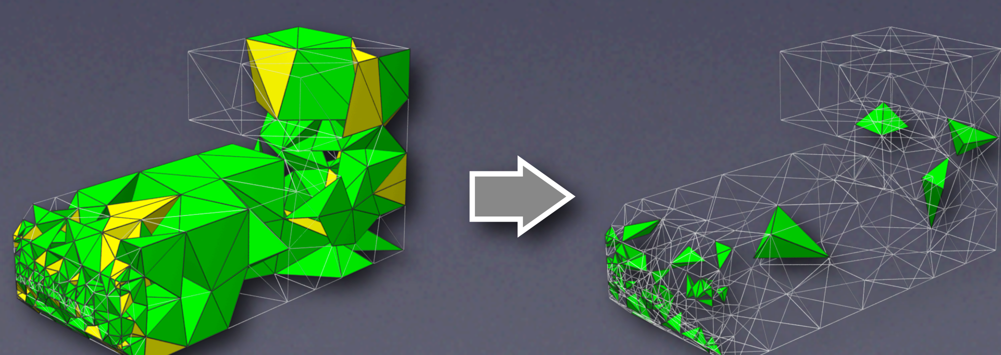
① Mesh quality



② Improvement operations



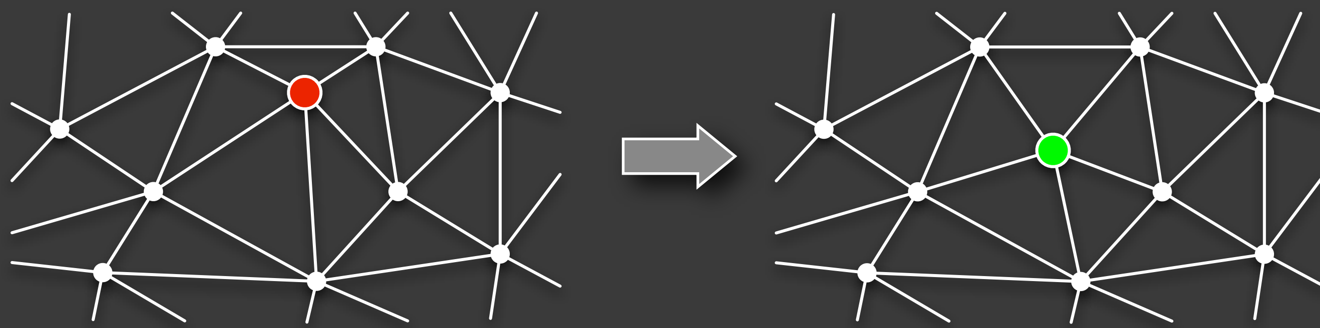
③ Improvement schedule



④ Results

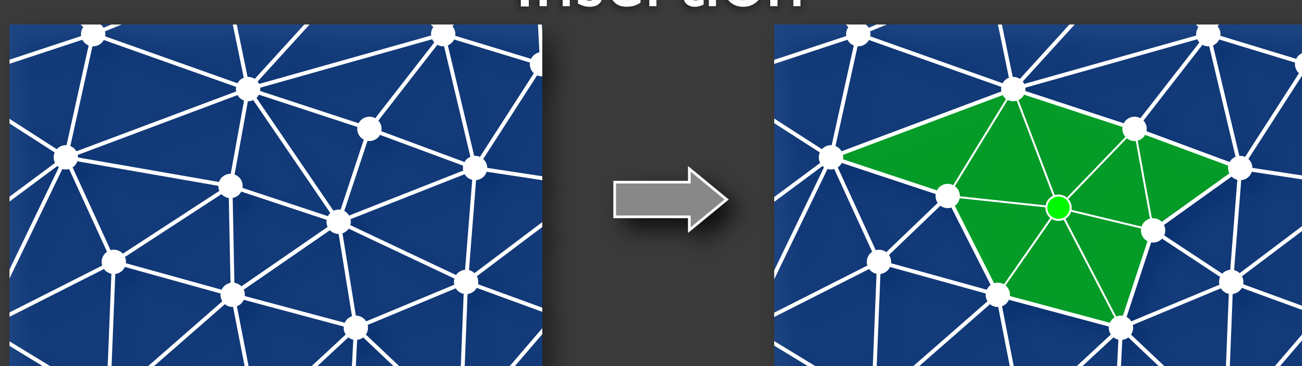
Building a schedule

Smoothing



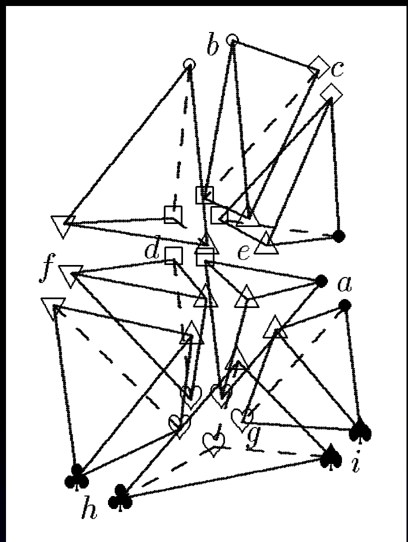
Topological ops

Insertion

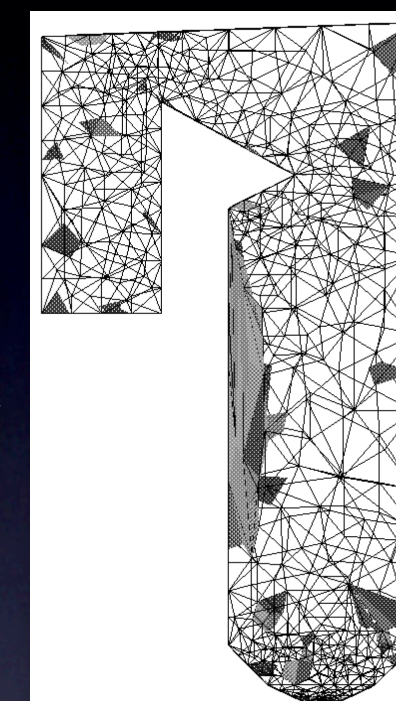


How do we turn these tools into a working improvement procedure?

Previous schedules



Joe, 1995. Repeatedly check every face to see if local topological improvements will help. Hard to gauge success.

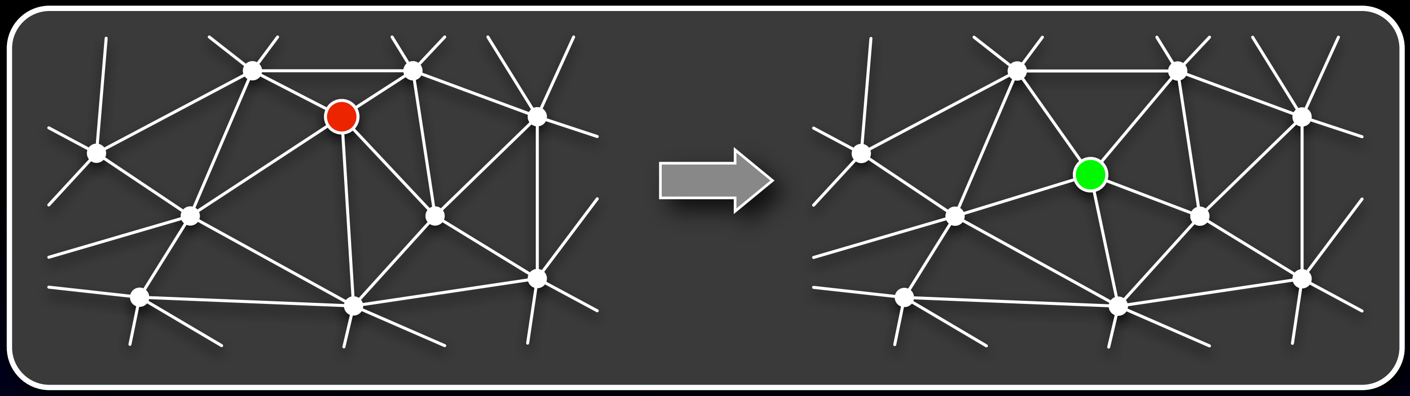


Freitag and Ollivier-Gooch, 1997. A fixed schedule of 2-3 flips, edge removal, and then optimization based smoothing. Most dihedral angles between 12 and 160 degrees.

Edelsbrunner and Guoy, 2001. Sequences of 2-3 and 3-2 flips. Most dihedral angles greater than 5 degrees.

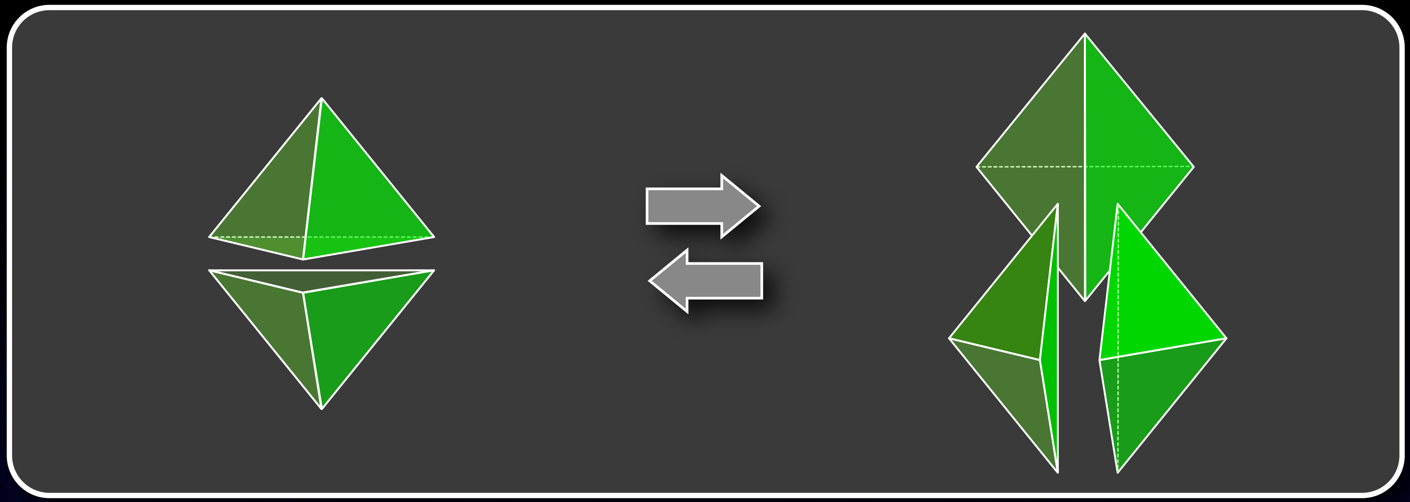
Alliez, Cohen-Steiner, Yvinec, and Desbrun, 2005. Alternates between global passes of smooth optimization-based smoothing and Delaunay retriangulation. No bounds given. In our experience, bad tetrahedra remain.

Smoothing Pass



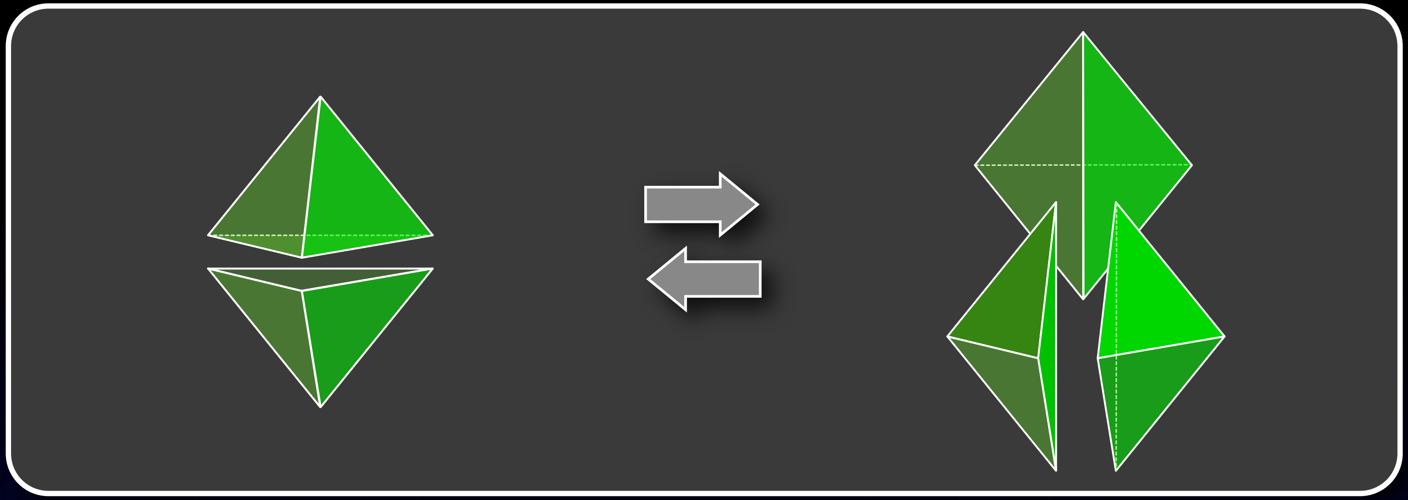
Perform optimization-based smoothing on each vertex in the mesh.

Topological Pass



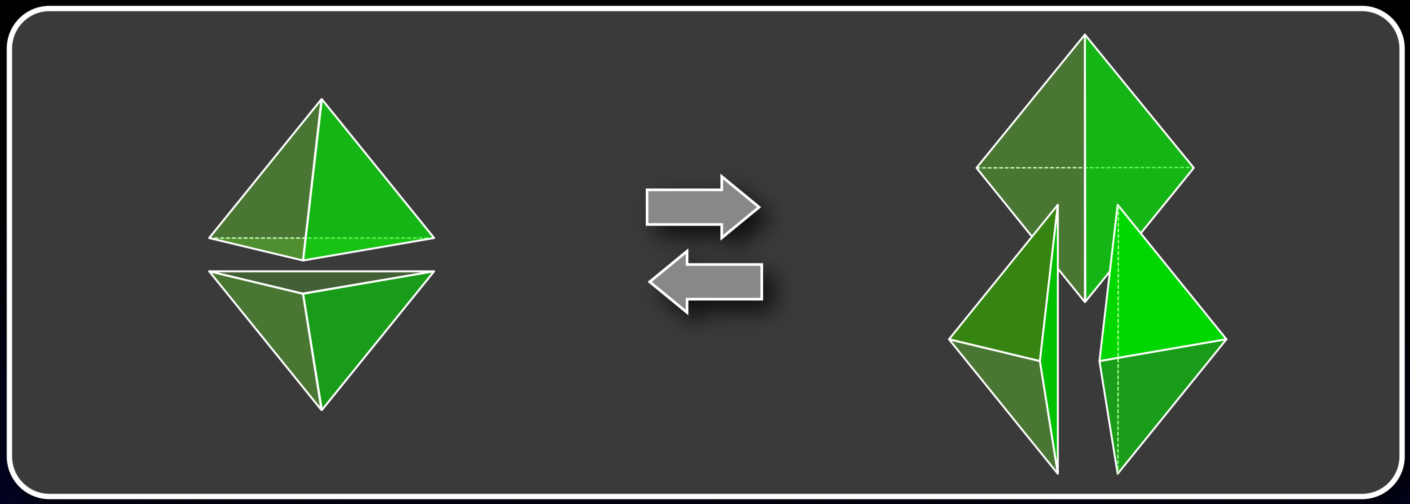
for each tetrahedron *t* in the mesh

Topological Pass



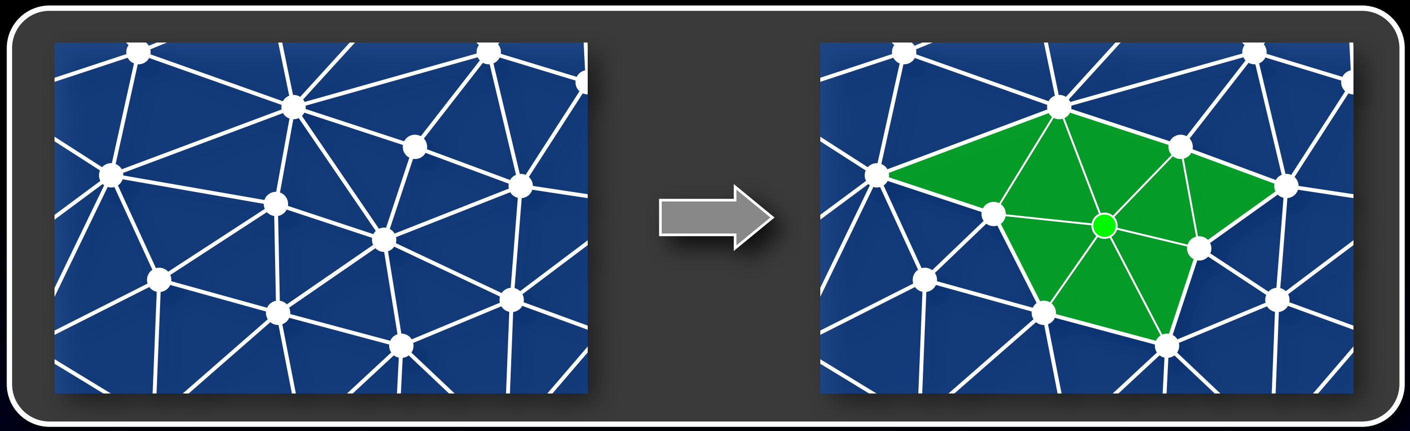
for each tetrahedron *t* in the mesh
for each edge e of *t* (if *t* still exists)
Attempt to remove edge e.

Topological Pass



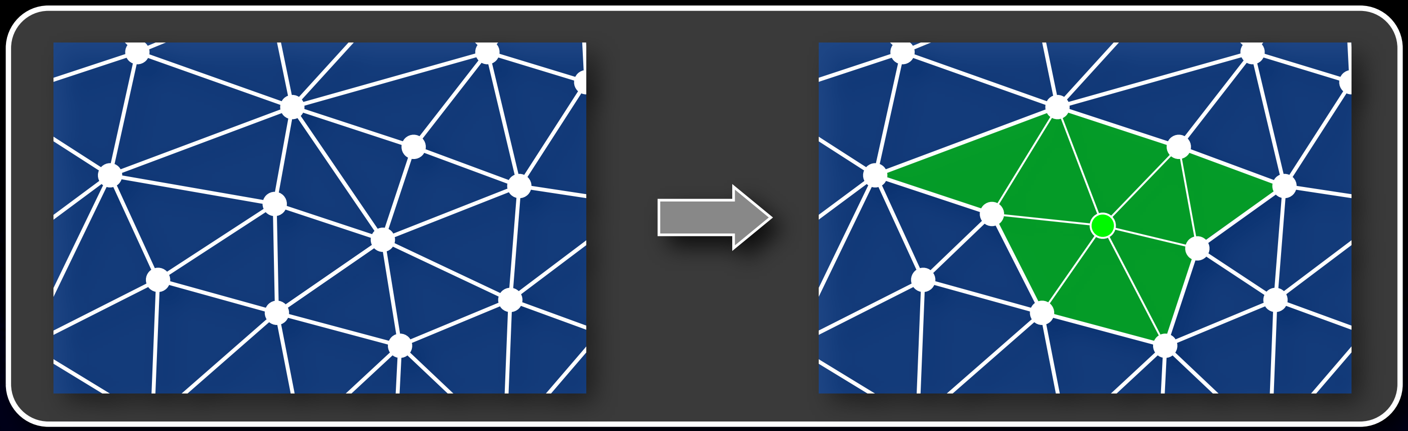
for each tetrahedron *t* in the mesh
  for each edge e of *t* (if *t* still exists)
    Attempt to remove edge e.
  for each face *f* of *t* (if *t* still exists)
    Attempt to remove face *f*.

Insertion Pass

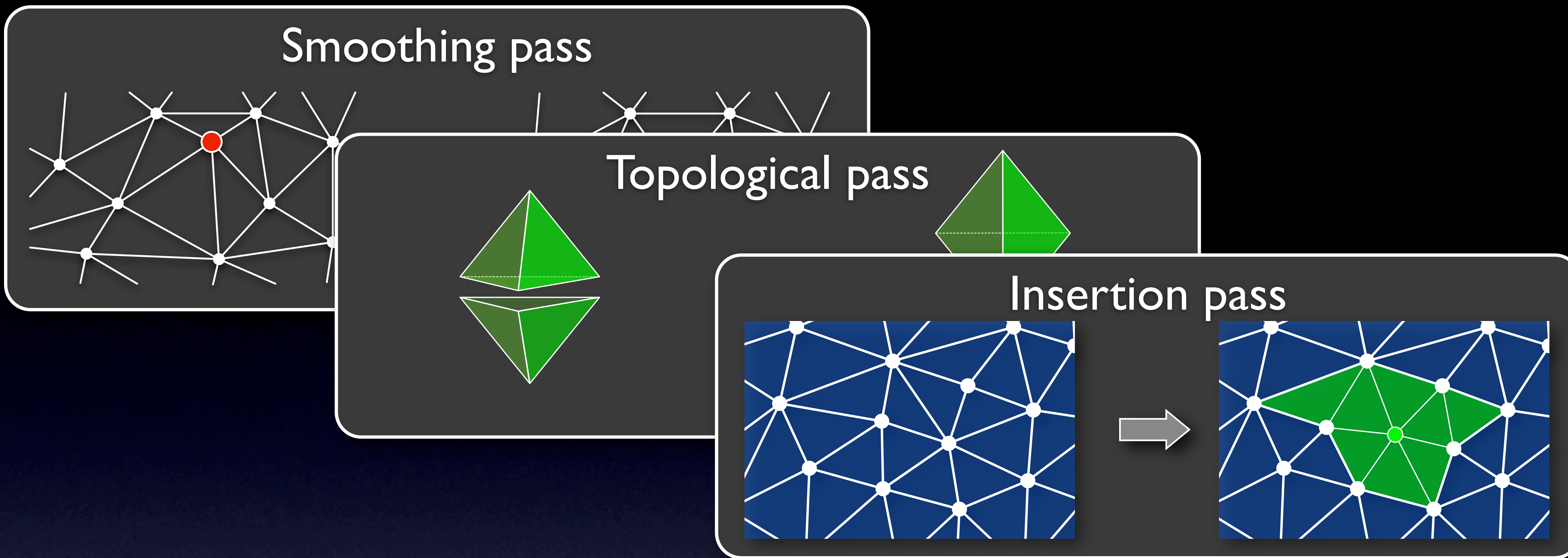


for each tetrahedron *t* in *L* that still exists

Insertion Pass



for each tetrahedron *t* in *L* that still exists
    Attempt insertion to split *t*.

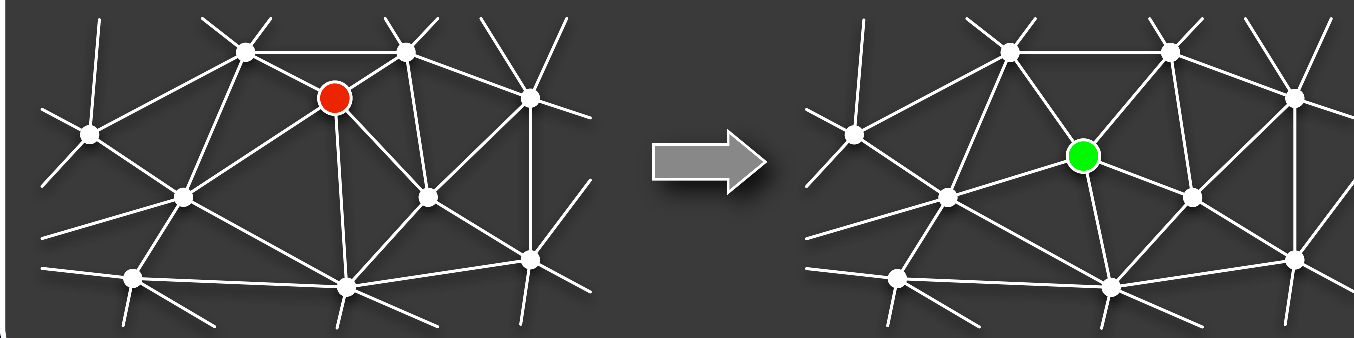


A pass succeeds if the overall mesh quality vector improves "enough."

$$Q(M) = \{1, 3, 10, 10, 15, 20, 23 \ldots\}$$

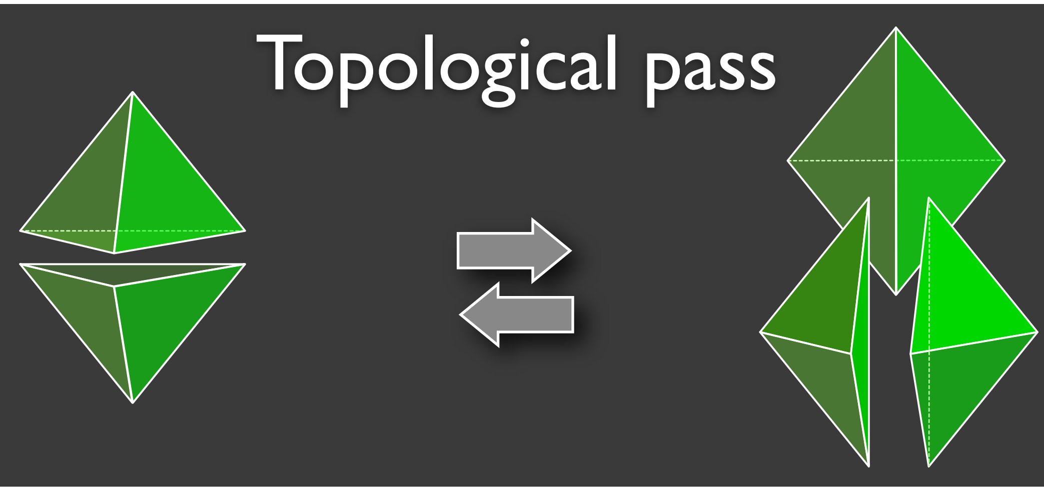
To get started,

Smoothing pass



then

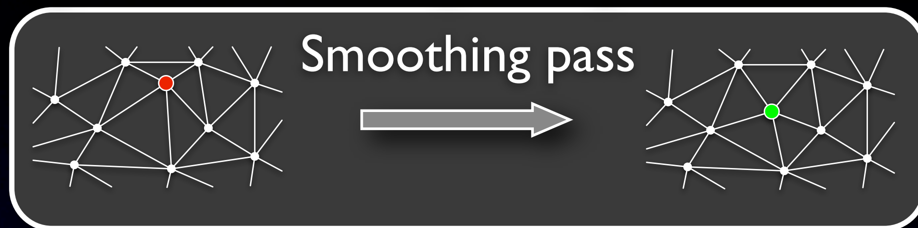
Topological pass



then...

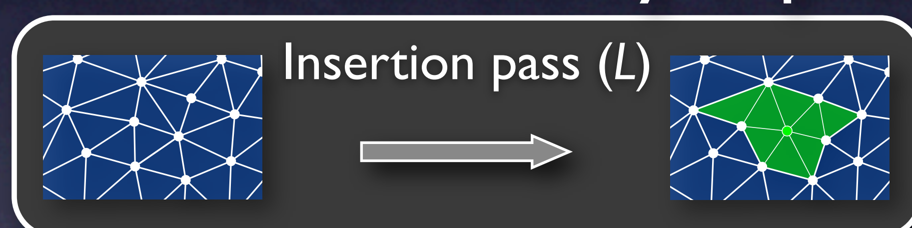
failed   0
while *failed* < 3

Q   list of quality indicators for the mesh



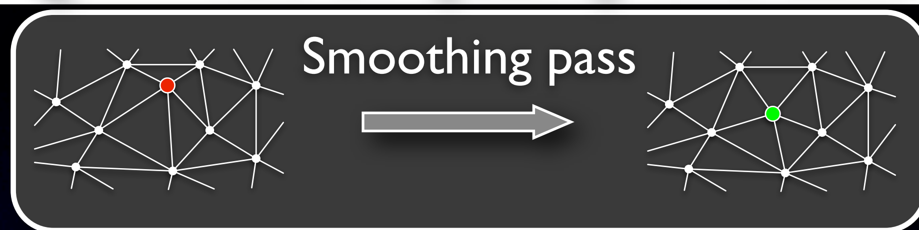
if mesh not sufficiently improved over *Q*

if mesh not sufficiently improved over *Q*



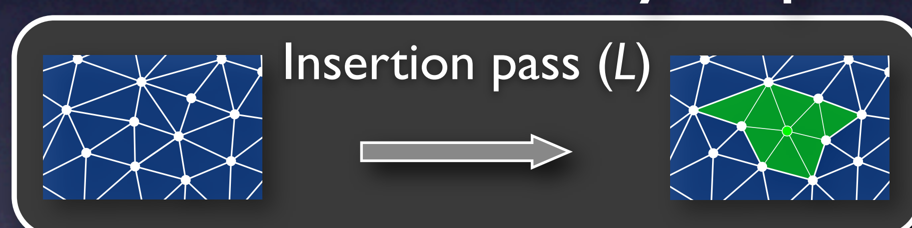
if mesh not sufficiently improved over *Q*
failed   *failed* + *1*
else *failed*   0 {insertion pass succeeded}
else *failed*   0   {topological pass succeeded}
else *failed*   0   {smoothing pass succeeded}

failed 0
while *failed* < 3
Q list of quality indicators for the mesh



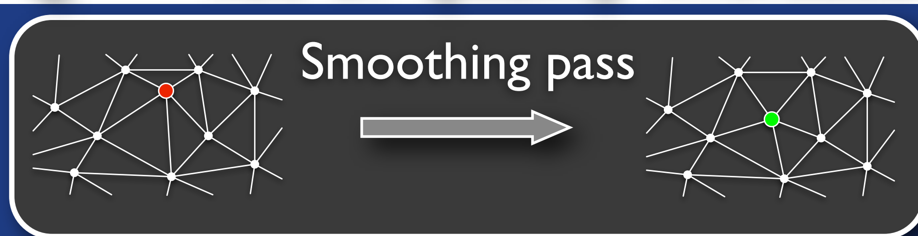
if mesh not sufficiently improved over *Q*

if mesh not sufficiently improved over *Q*



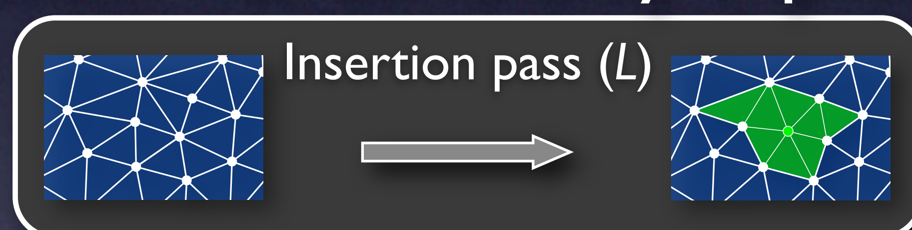
if mesh not sufficiently improved over *Q*
failed *failed* + *1*
else *failed* 0 {insertion pass succeeded}
else *failed* 0 {topological pass succeeded}
else *failed* 0 {smoothing pass succeeded}

failed 0
while *failed* < 3
Q list of quality indicators for the mesh



if mesh not sufficiently improved over *Q*

if mesh not sufficiently improved over *Q*



if mesh not sufficiently improved over *Q*
failed *failed* + *1*
else *failed* 0 {insertion pass succeeded}
else *failed* 0 {topological pass succeeded}
else *failed* 0 {smoothing pass succeeded}

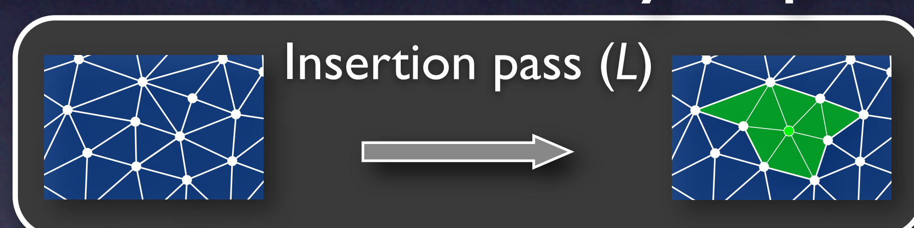
failed 0

while *failed* < 3

Q list of quality indicators for the mesh

if mesh not sufficiently improved over *Q*

if mesh not sufficiently improved over *Q*



if mesh not sufficiently improved over *Q*

failed *failed* + 1

else *failed* 0 {insertion pass succeeded}

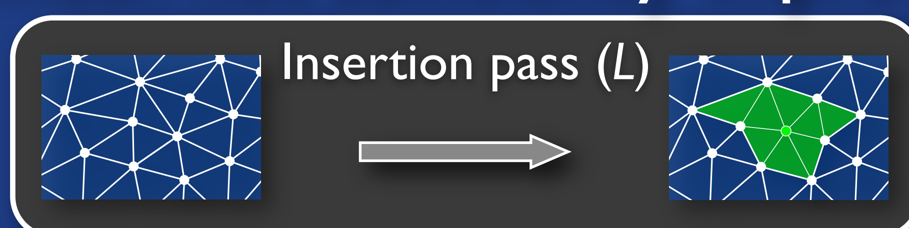
else *failed* 0 {topological pass succeeded}

else *failed* 0 {smoothing pass succeeded}

failed 0
while *failed* < 3
  Q list of quality indicators for the mesh

  if mesh not sufficiently improved over *Q*

    if mesh not sufficiently improved over *Q*



      if mesh not sufficiently improved over *Q*
        failed *failed* + *1*
      else *failed* 0 {insertion pass succeeded}
    else *failed* 0 {topological pass succeeded}
  else *failed* 0 {smoothing pass succeeded}

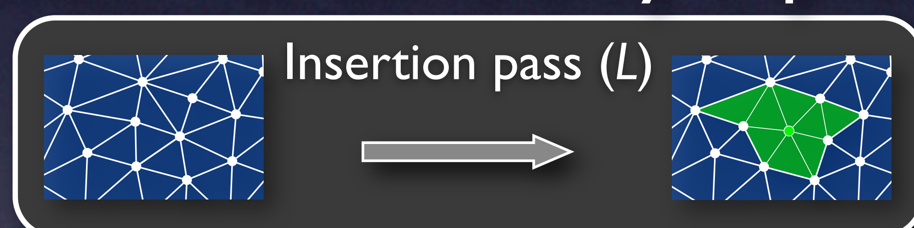
failed 0

while *failed* < 3

Q list of quality indicators for the mesh

if mesh not sufficiently improved over *Q*

if mesh not sufficiently improved over *Q*



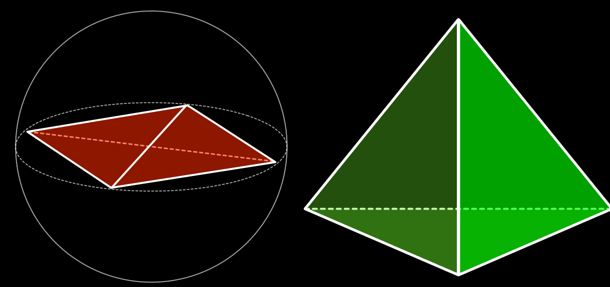
if mesh not sufficiently improved over *Q*

failed *failed* + *1*

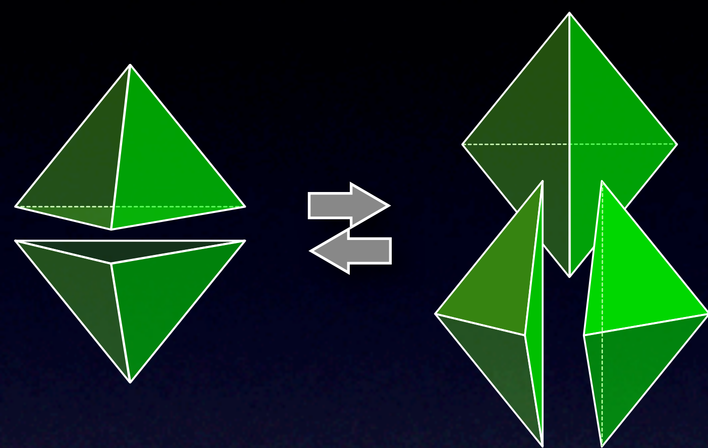
else *failed* 0 {insertion pass succeeded}

else *failed* 0 {topological pass succeeded}

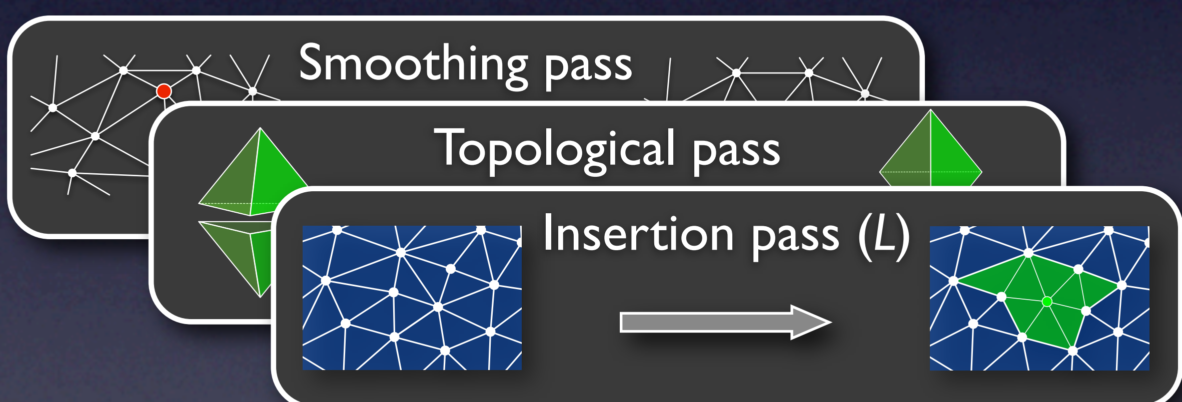
else *failed* 0 {smoothing pass succeeded}



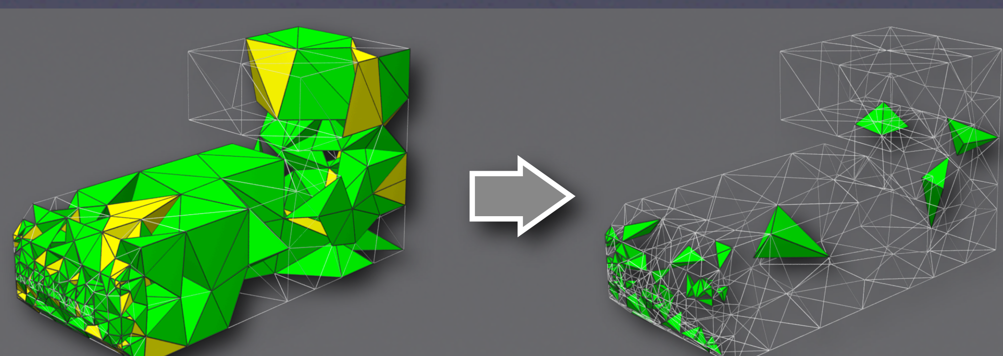
① Mesh quality



② Improvement operations

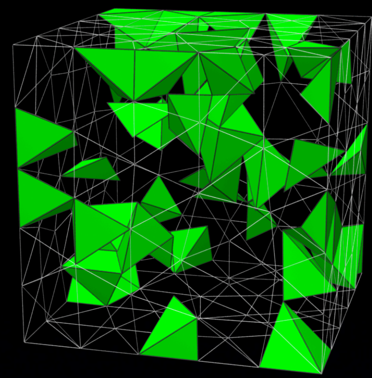


③ Improvement schedule

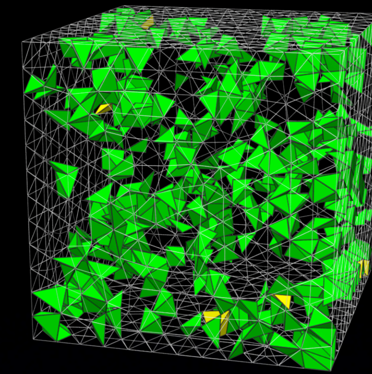


④ Results

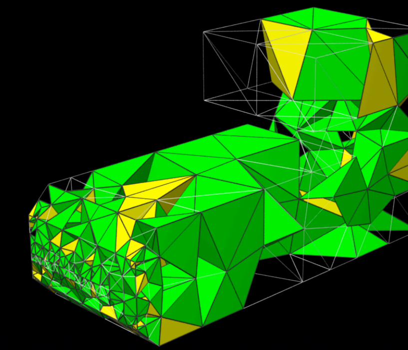
CUBE1K



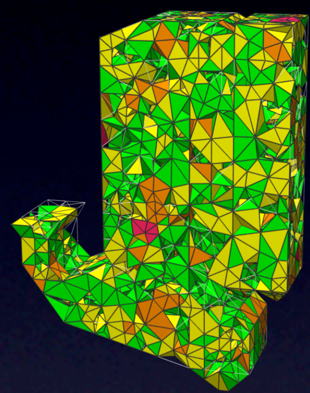
CUBE10K



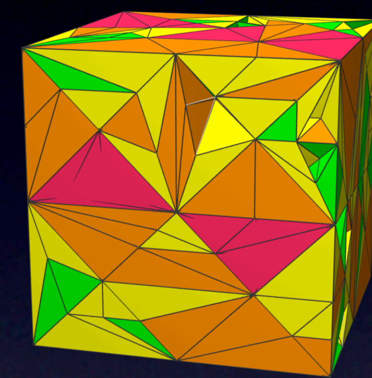
TFIRE



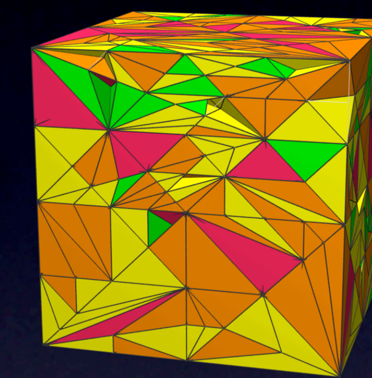
TIRE



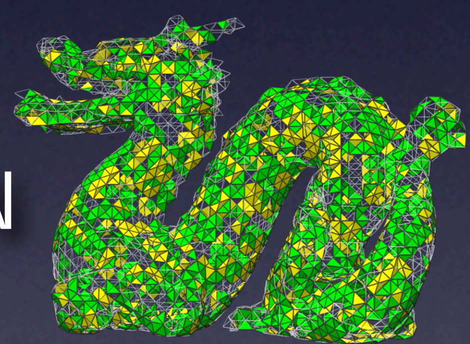
RAND1



RAND2

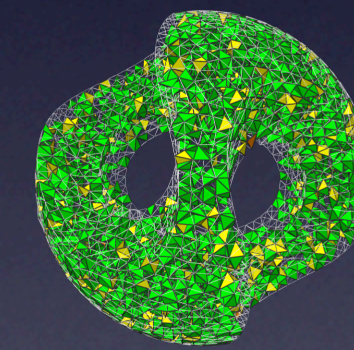


DRAGON

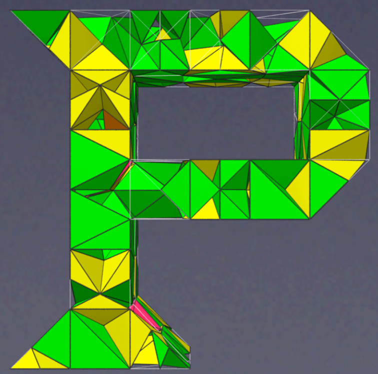


COW

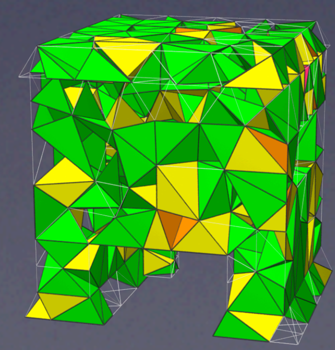
SCULPT



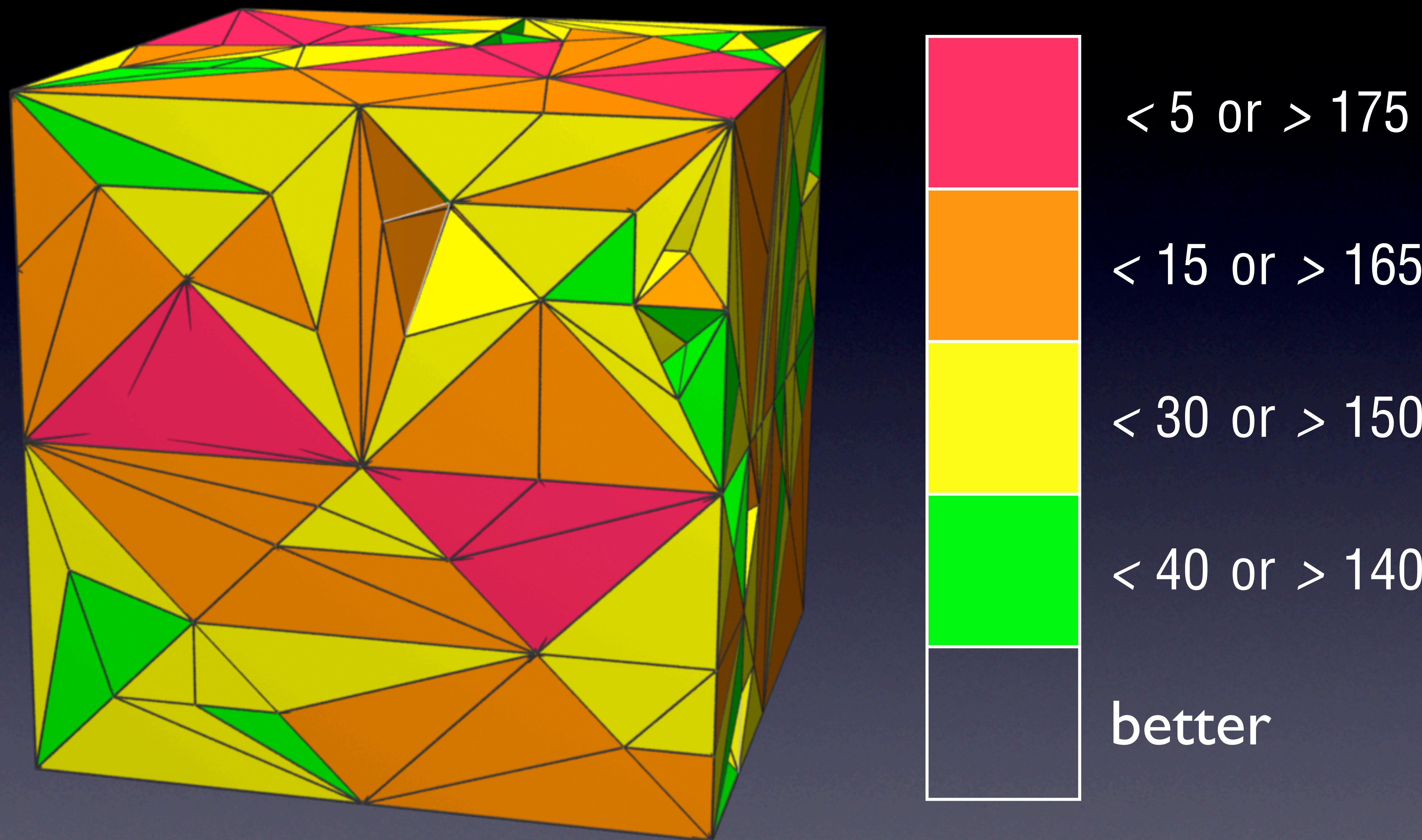
P

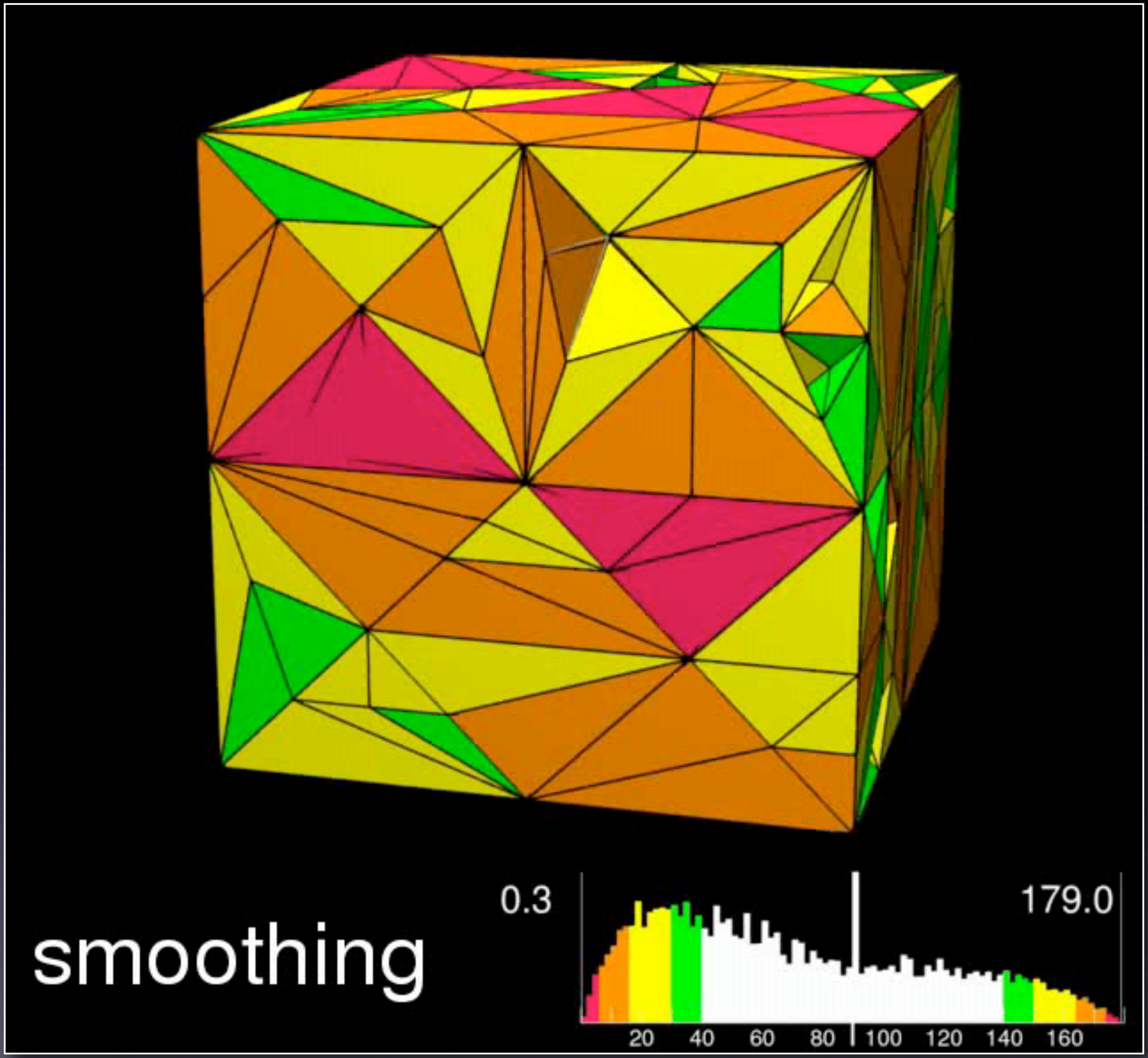


HOUSE

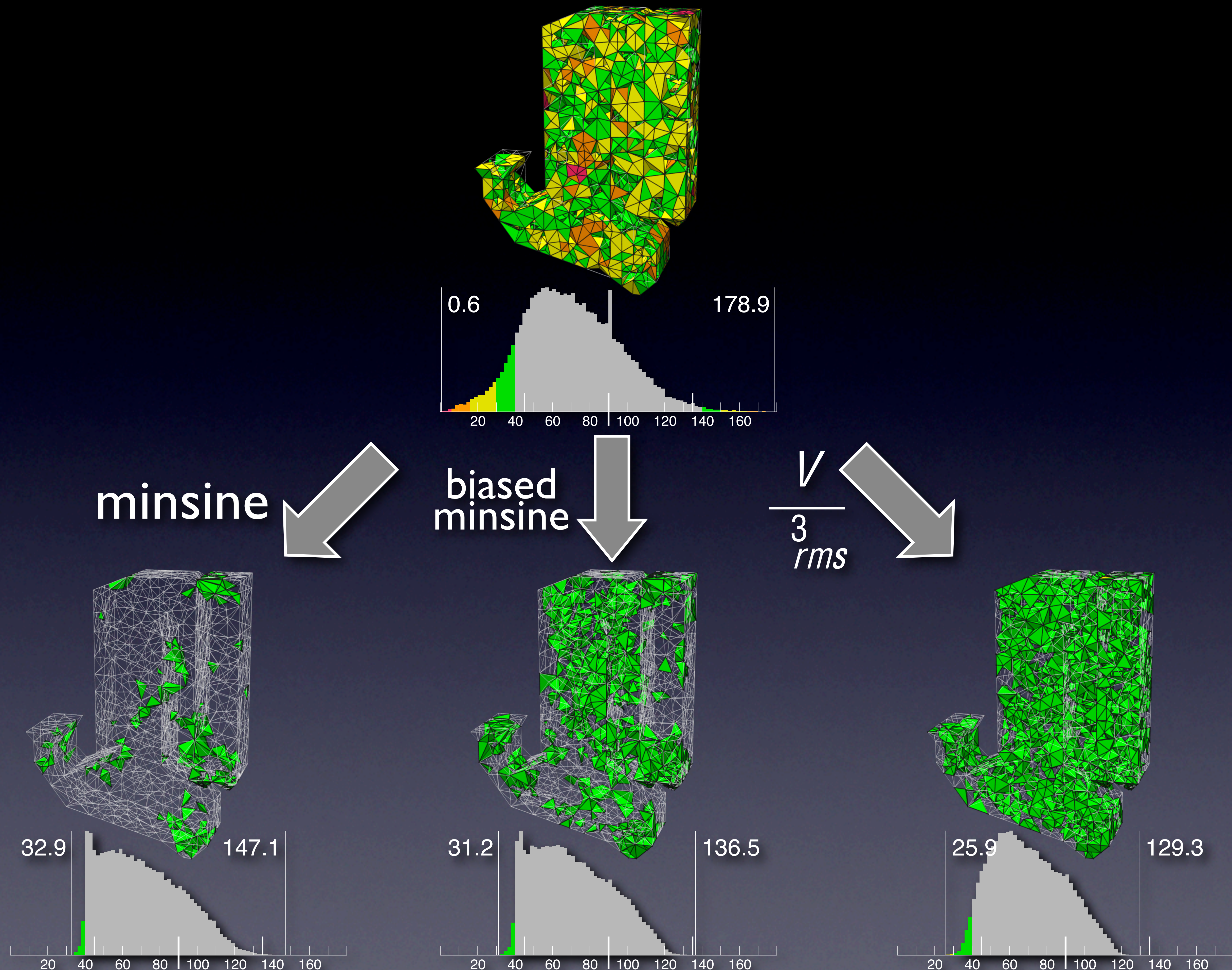


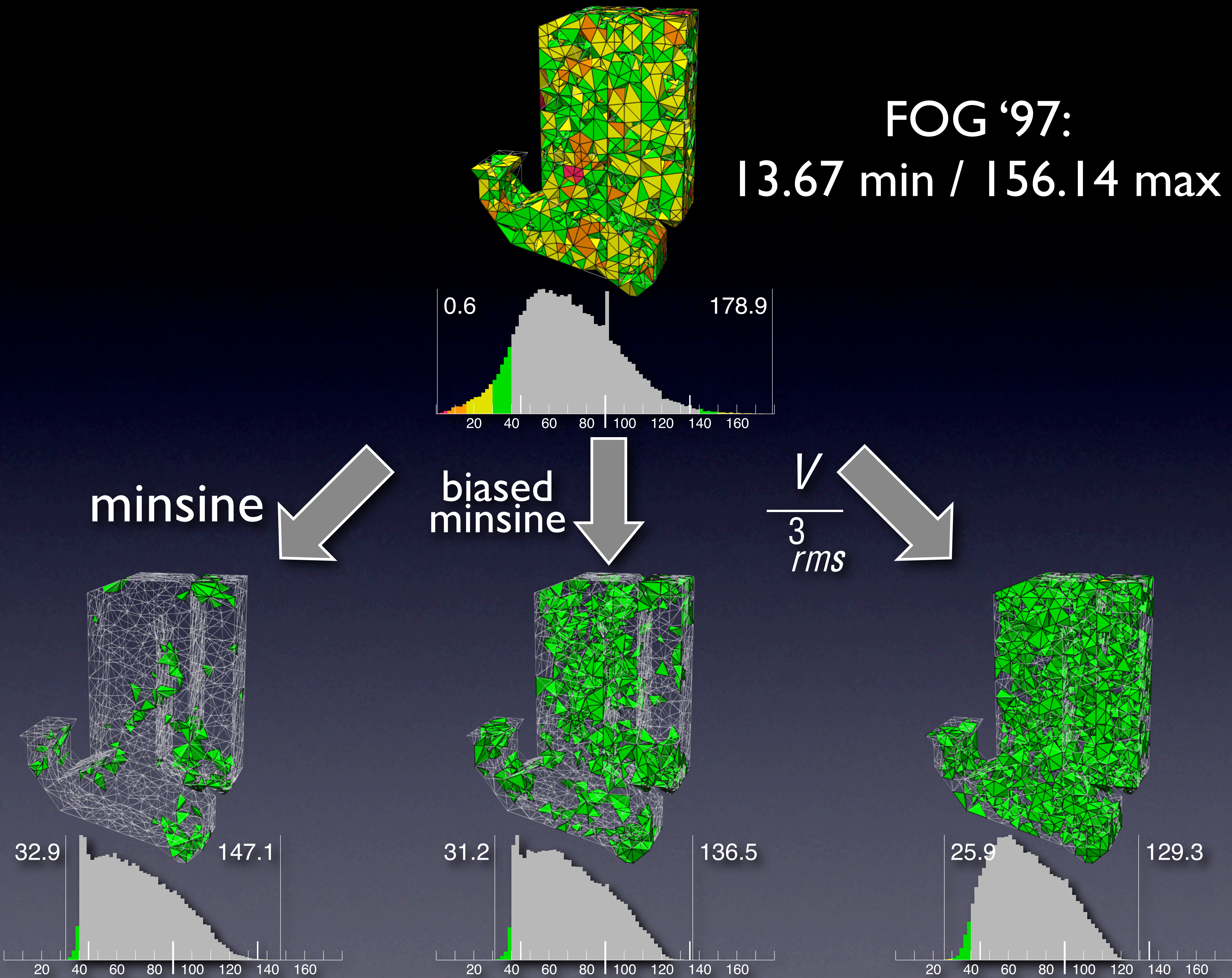
STAYPUFT


< 5 or > 175
< 15 or > 165
< 30 or > 150
< 40 or > 140
better

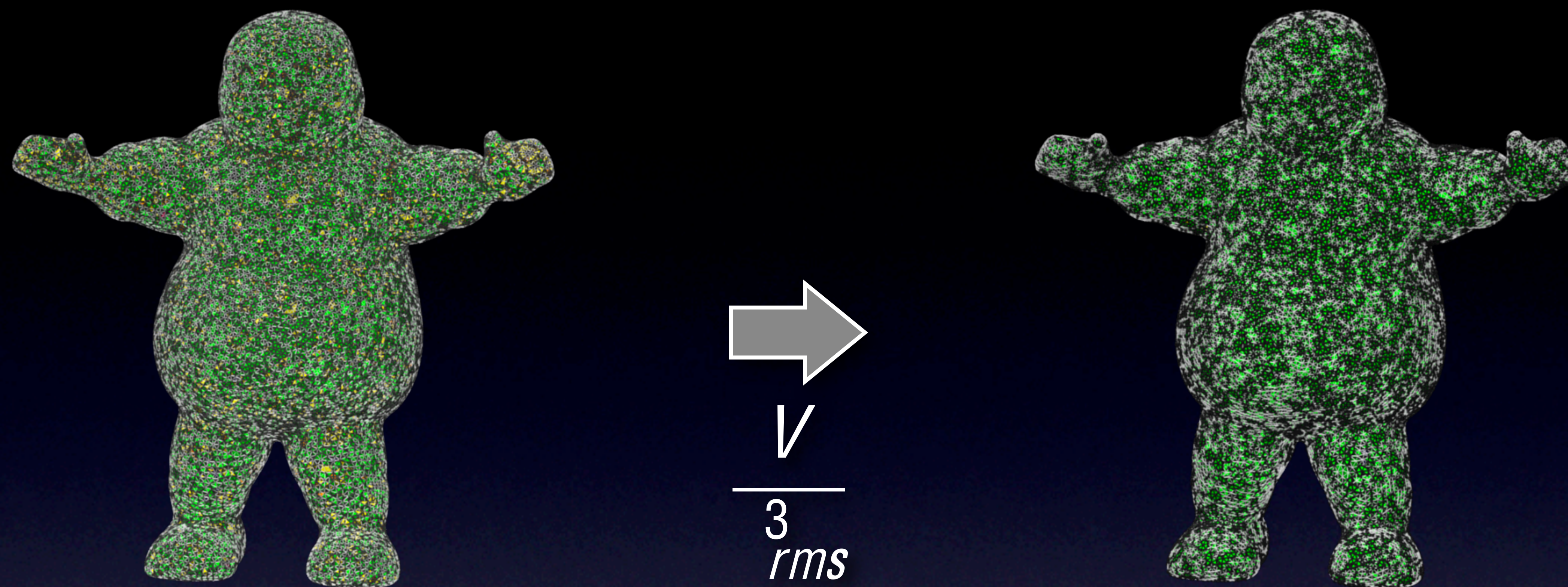


smoothing
0.3
179.0
20 40 60 80 100 120 140 160


0.6
178.9
20 40 60 80 100 120 140 160
minsine
biased minsine
$\frac{V}{\ell_{rms}^3}$
32.9
147.1
31.2
136.5
25.9
129.3

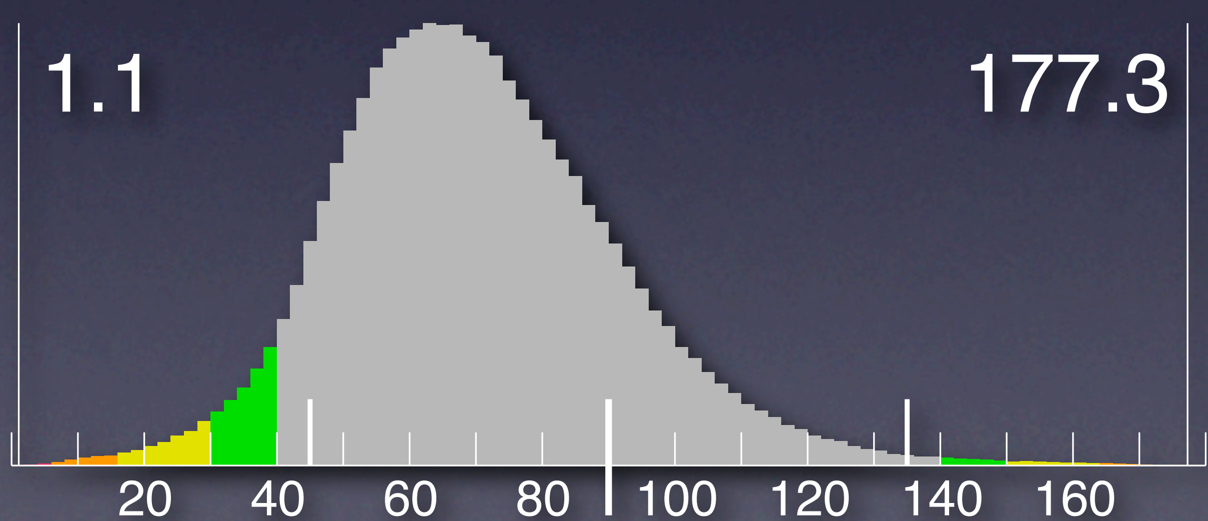

FOG '97:
13.67 min / 156.14 max
0.6
178.9
20 40 60 80 100 120 140 160
minsine
biased
minsine
$\frac{V}{3\,rms}$
32.9
147.1
31.2
136.5
25.9
129.3
20 40 60 80 100 120 140 160
20 40 60 80 100 120 140 160
20 40 60 80 100 120 140 160

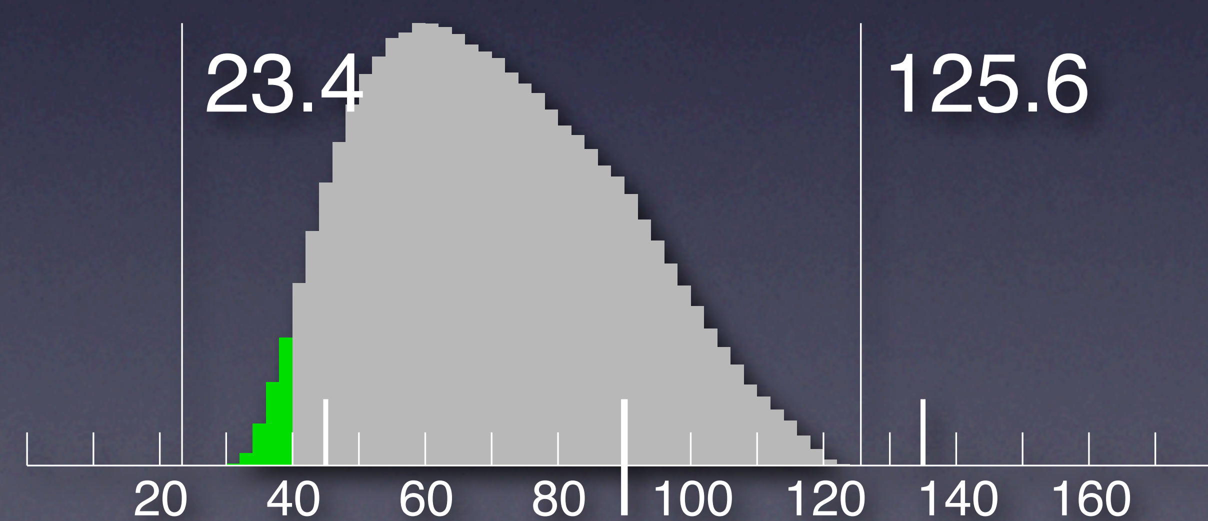
STAYPUFT 14,214 sec



102,393 tetrahedra

130,736 tetrahedra

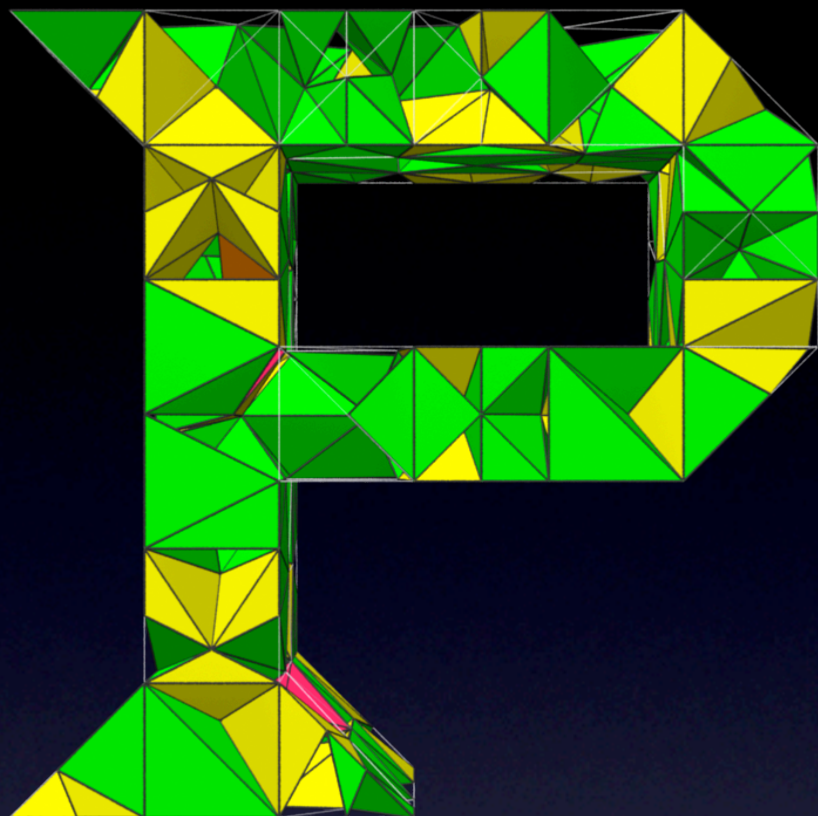




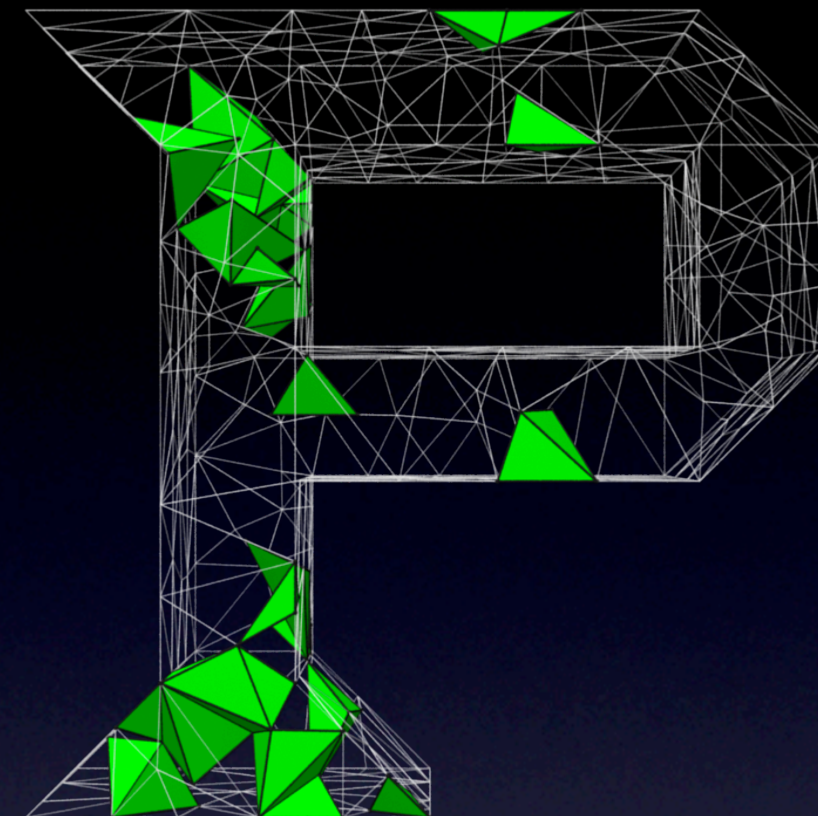
Insertion is slow: 90% of running time

P

24 sec

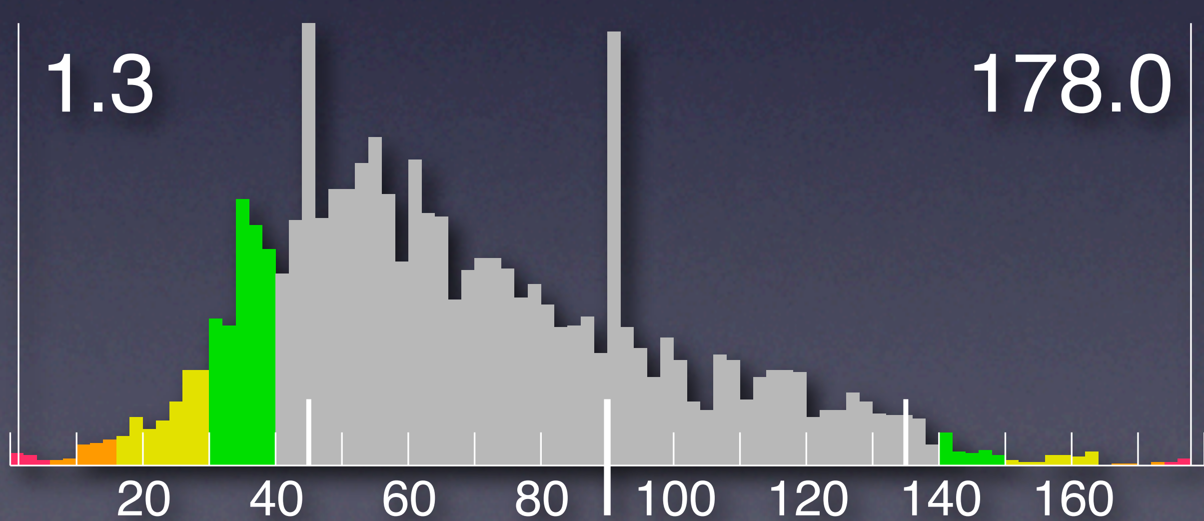


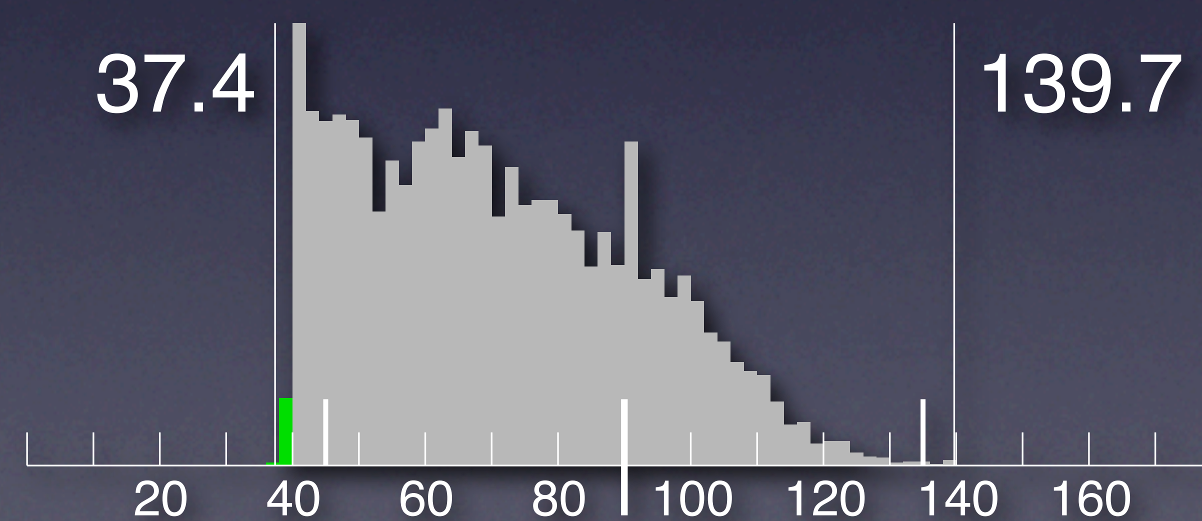
minsine



927 tetrahedra

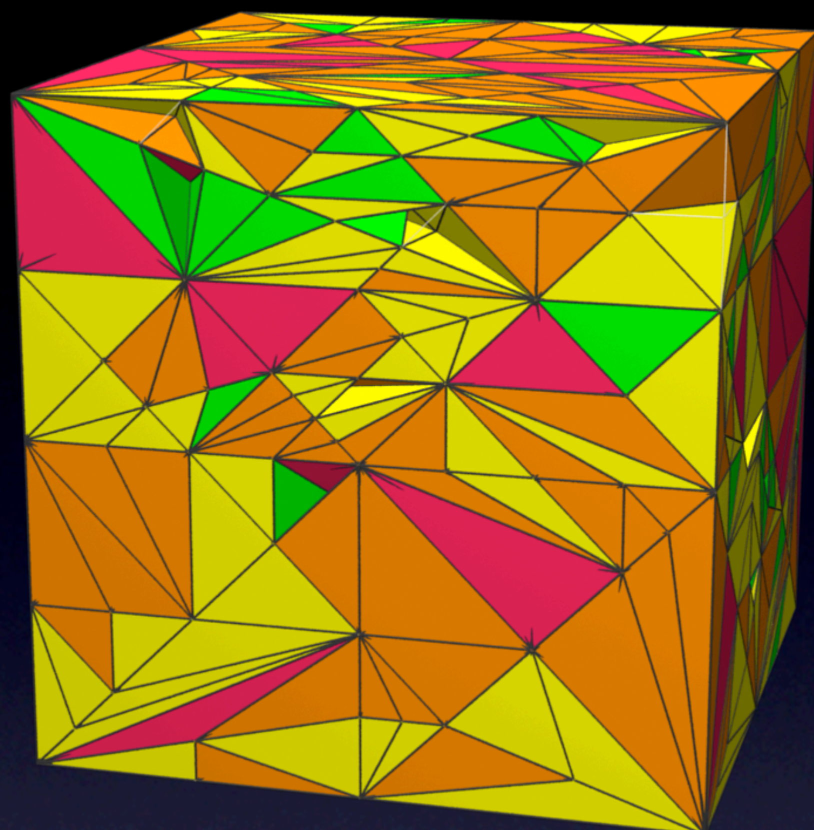
1,261 tetrahedra



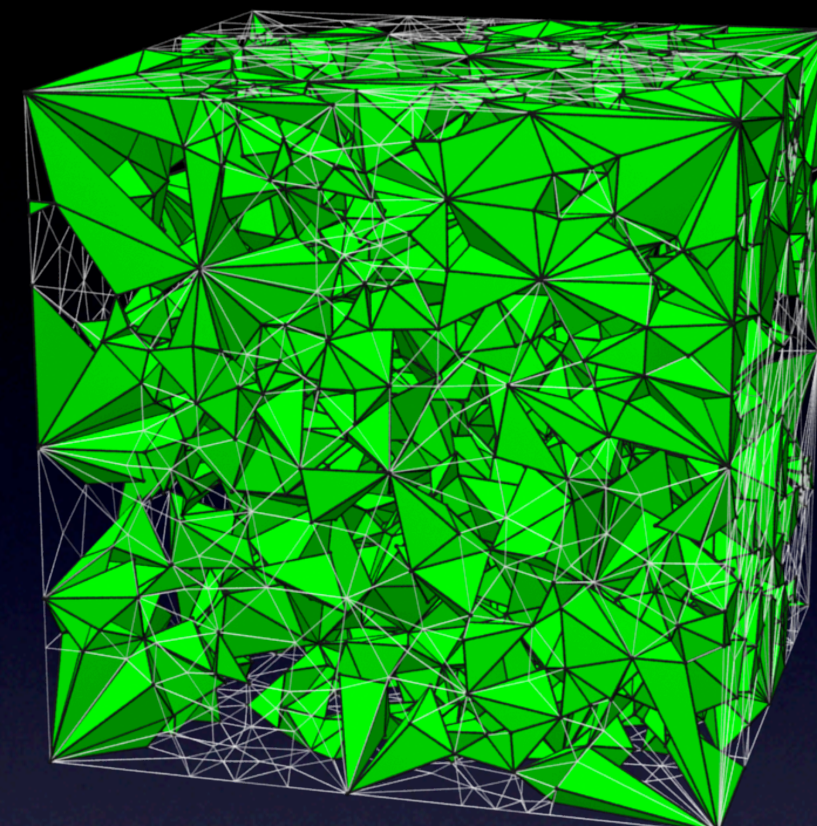


Insertion can make meshes bigger.

RAND2 4,658 sec

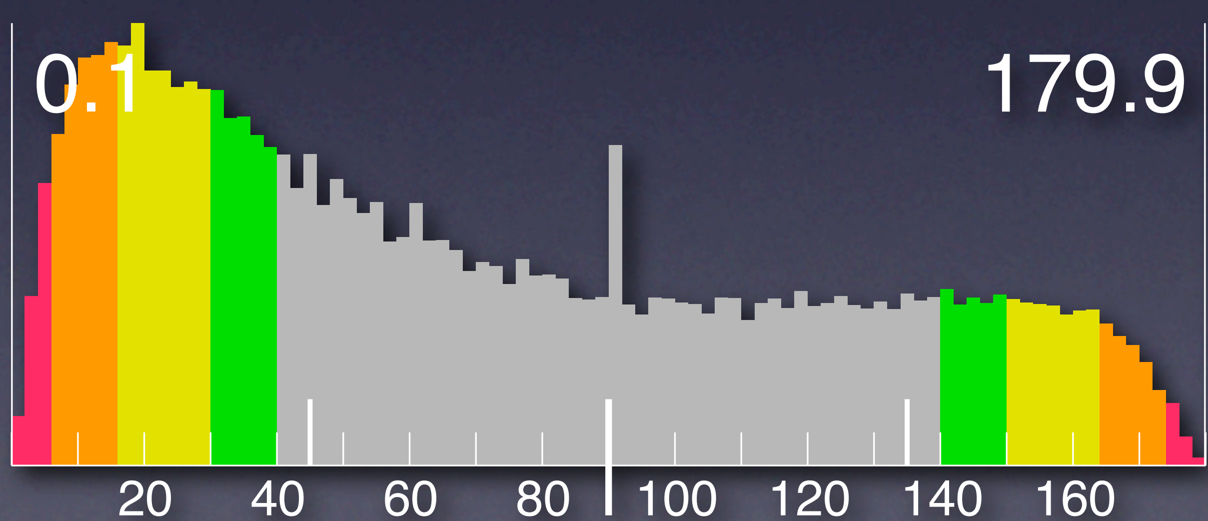


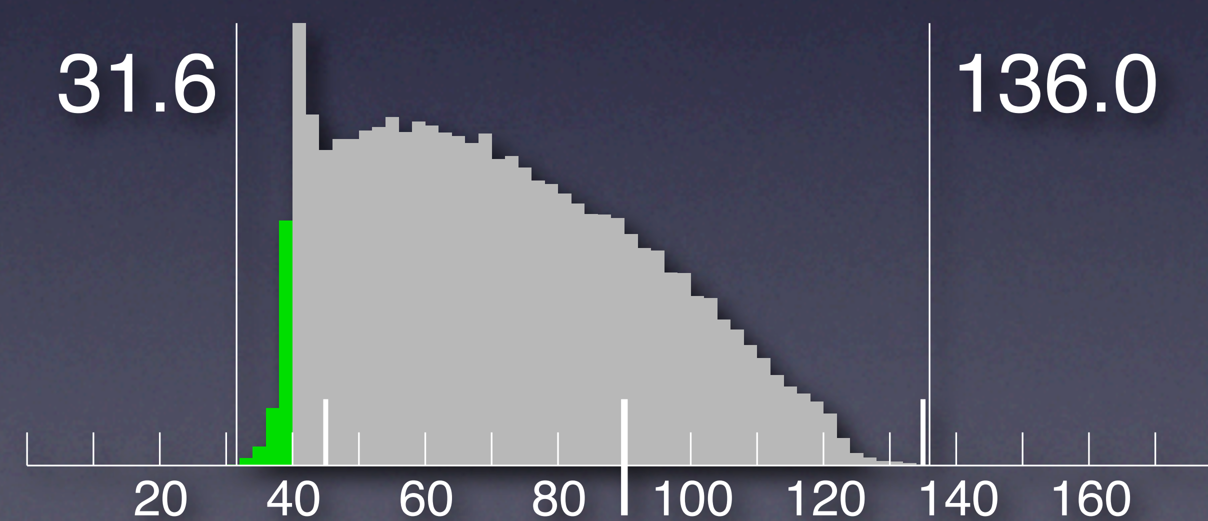
biased minsine



25,705 tetrahedra

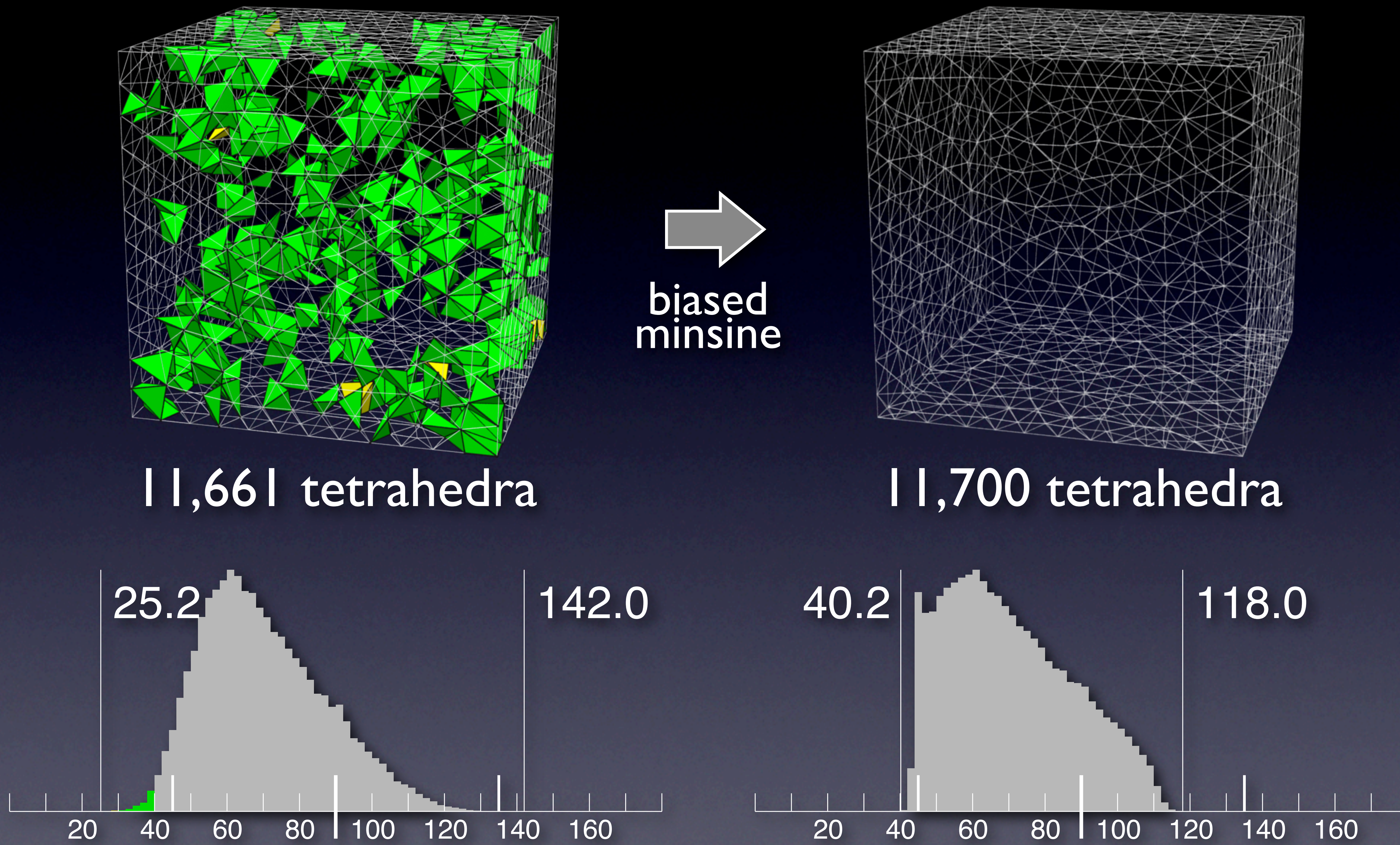
17,527 tetrahedra





Insertion can also make meshes smaller.

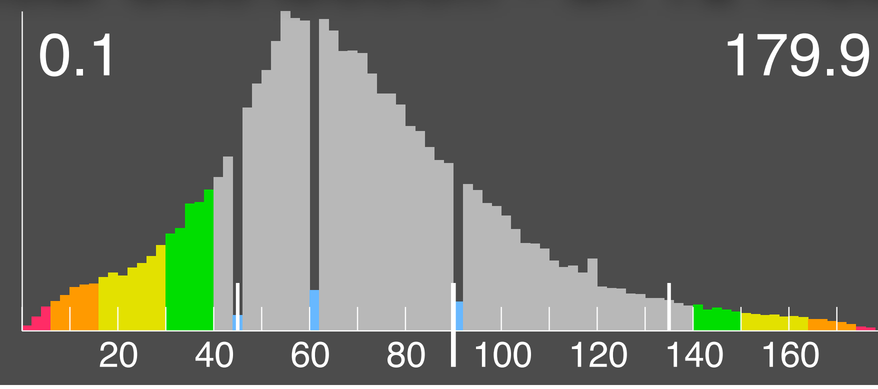
CUBE10K 121 sec



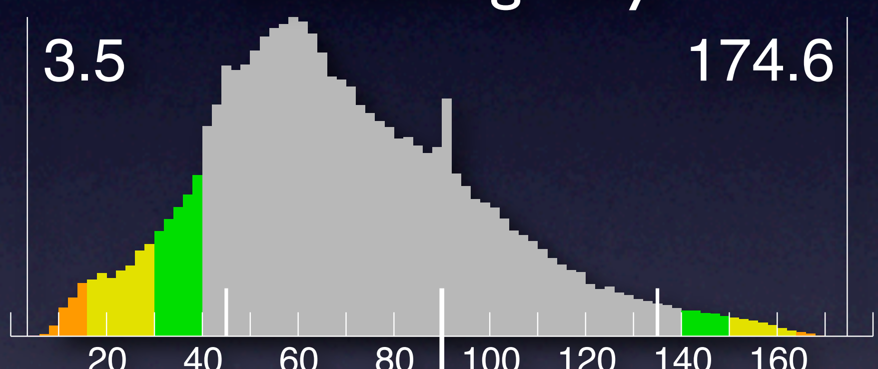
Meshes that start out good run fast and end great.

Adding features

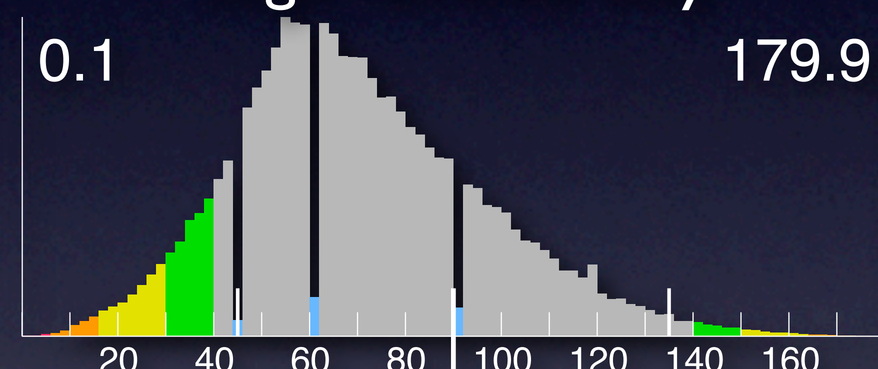
Initial distribution - all 12 meshes



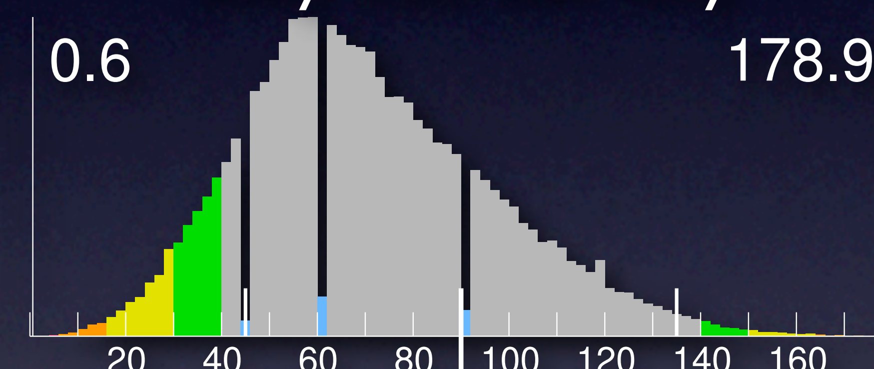
smoothing only



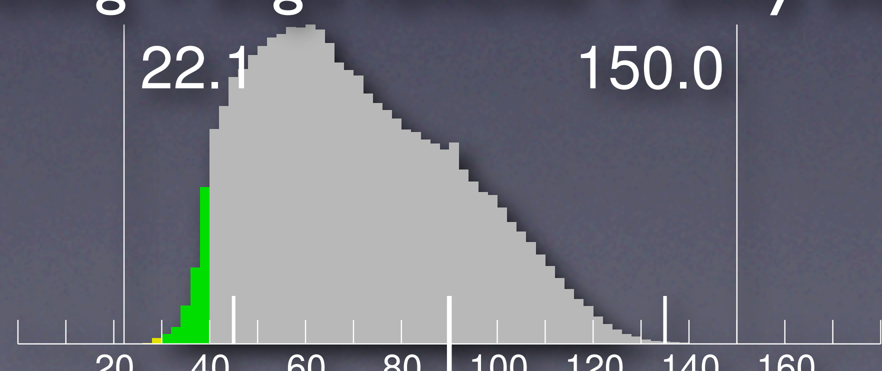
edge removal only



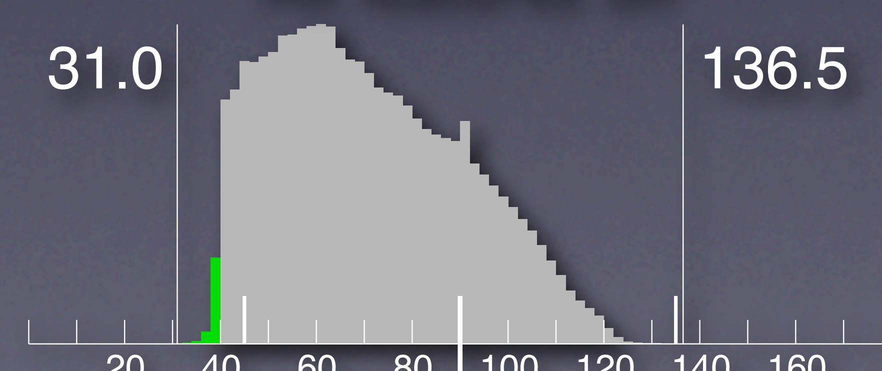
body insertion only



smoothing + edge removal + body insertion

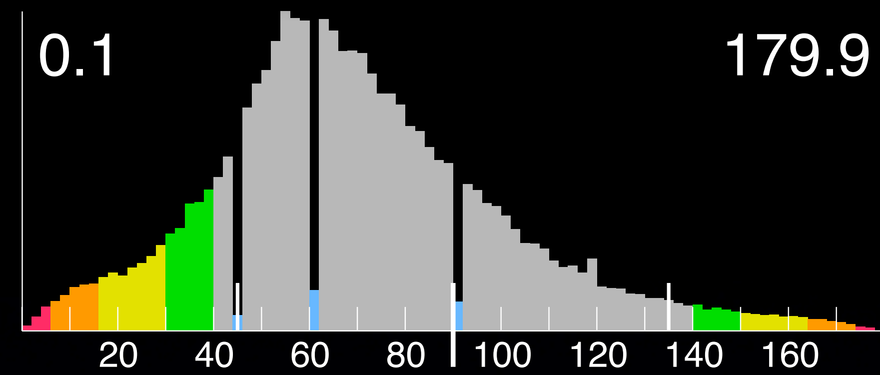


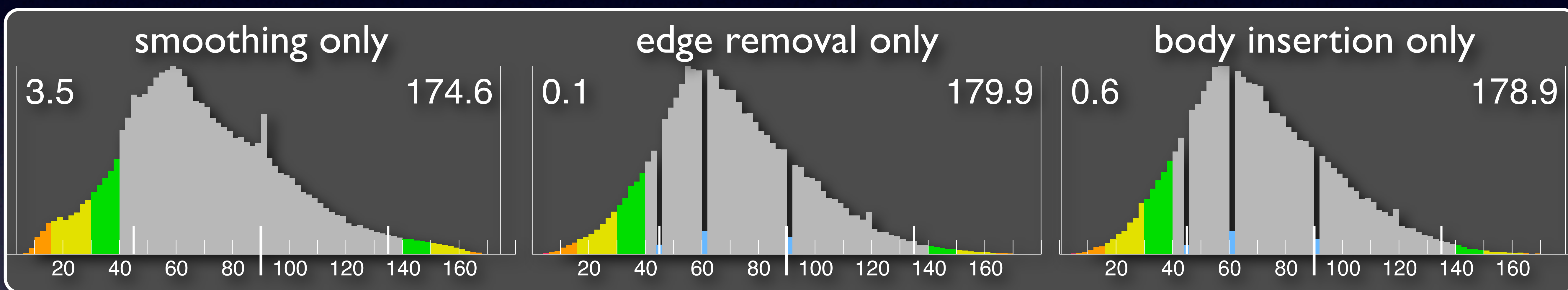
All features on



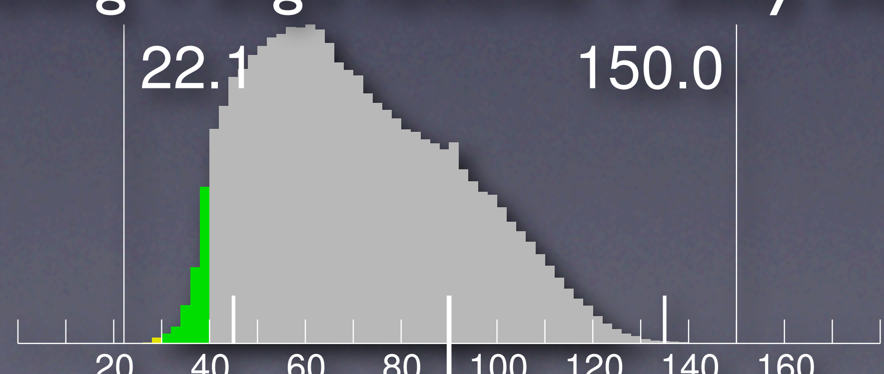
Adding features

Initial distribution - all 12 meshes

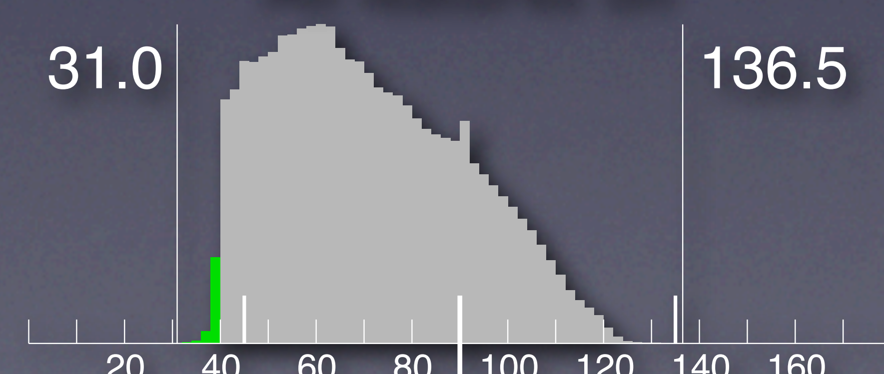




smoothing + edge removal + body insertion

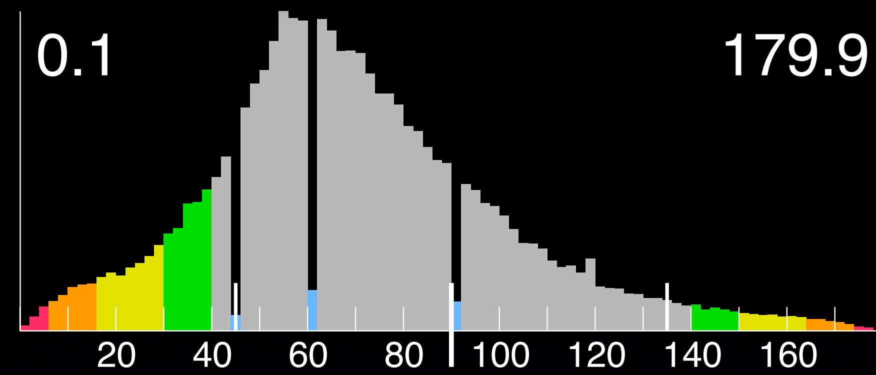


All features on

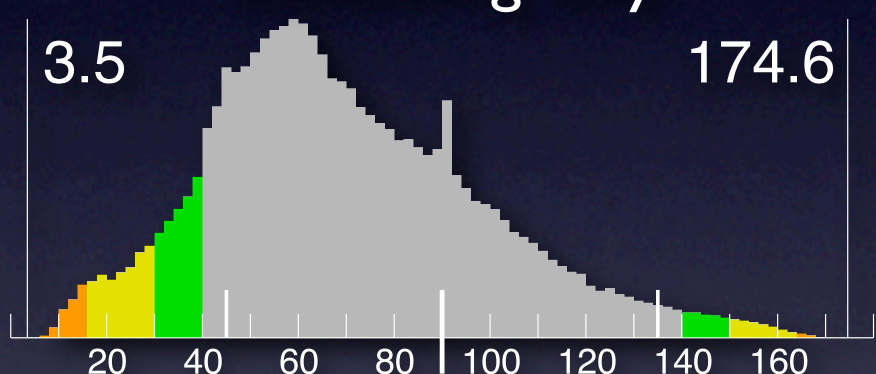


Adding features

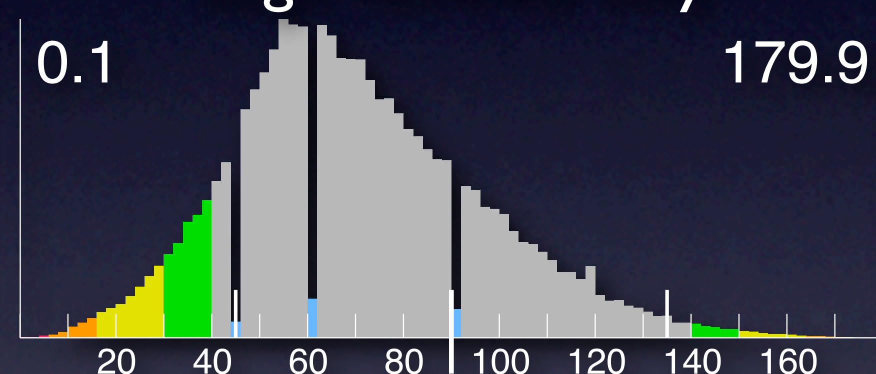
Initial distribution - all 12 meshes



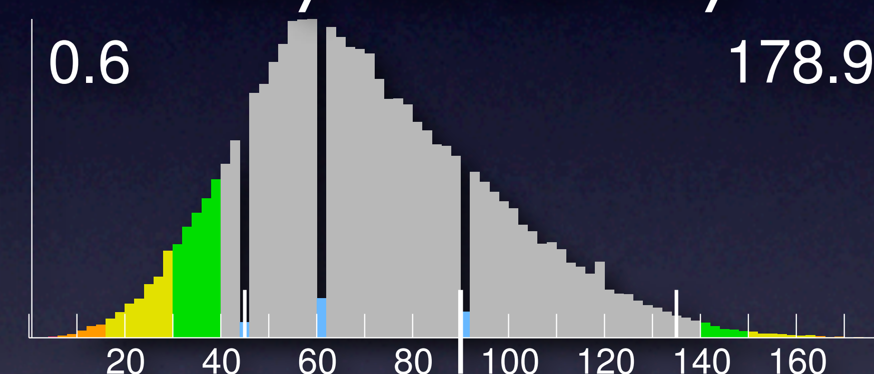
smoothing only



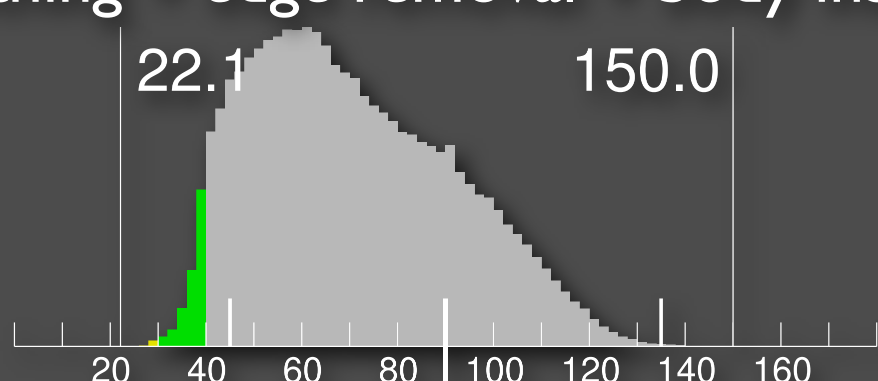
edge removal only



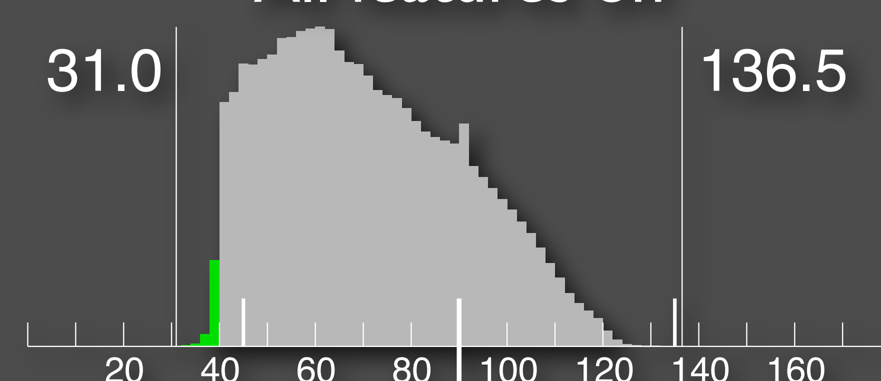
body insertion only



smoothing + edge removal + body insertion

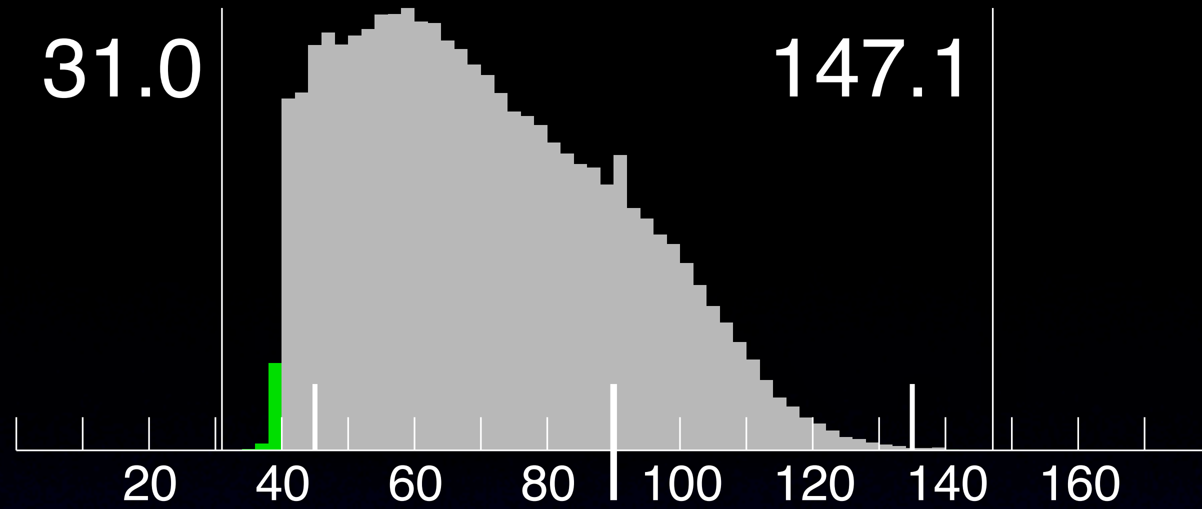


All features on

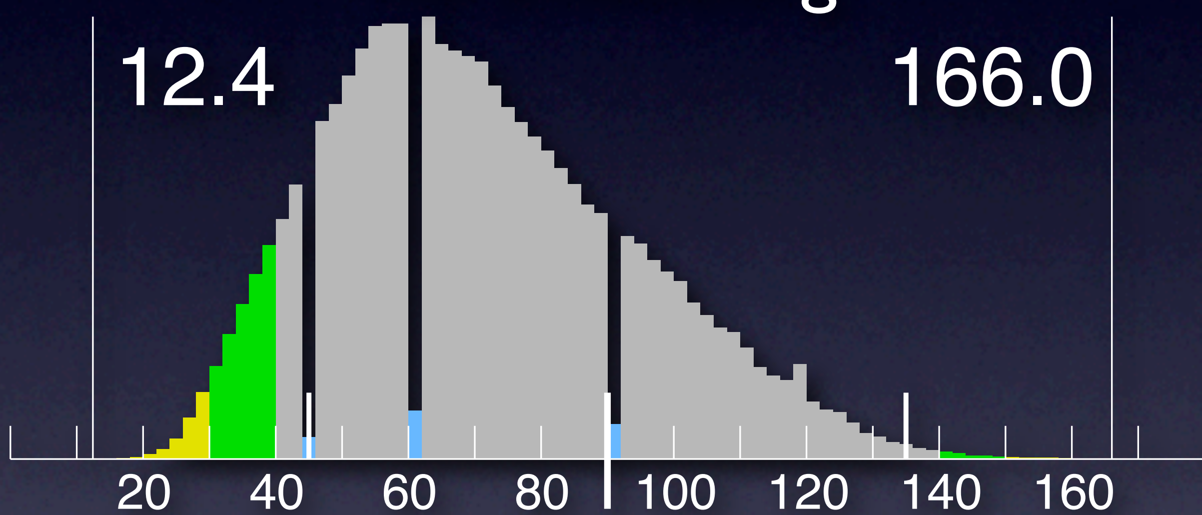


Removing features

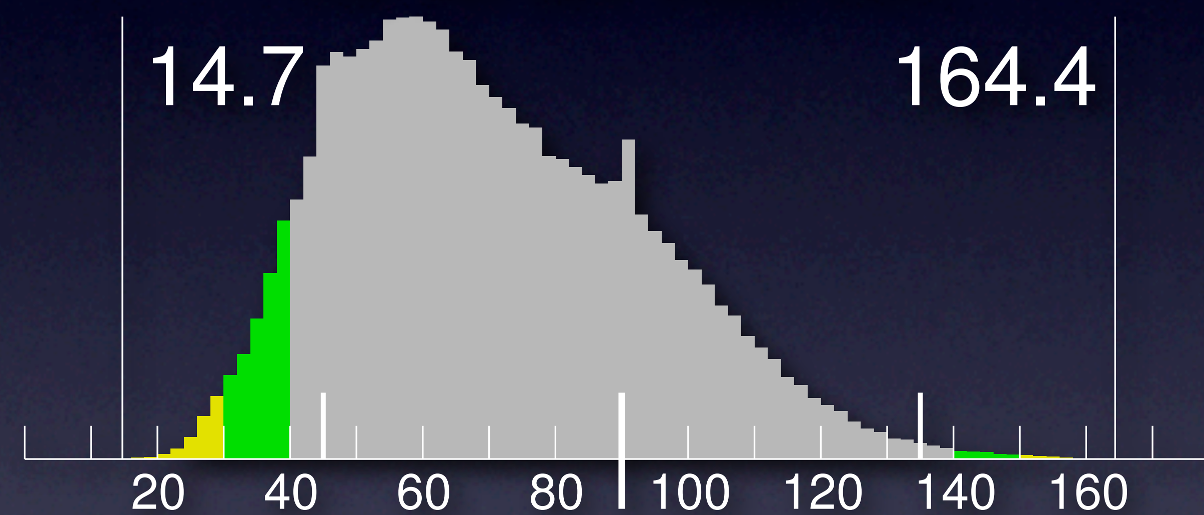
All features on - all 12 meshes



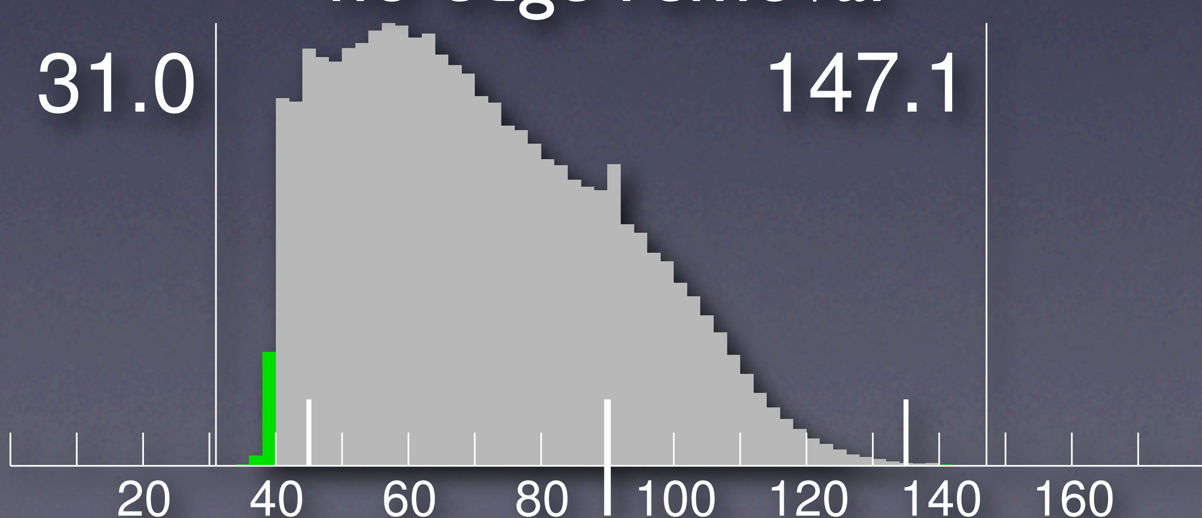
no smoothing



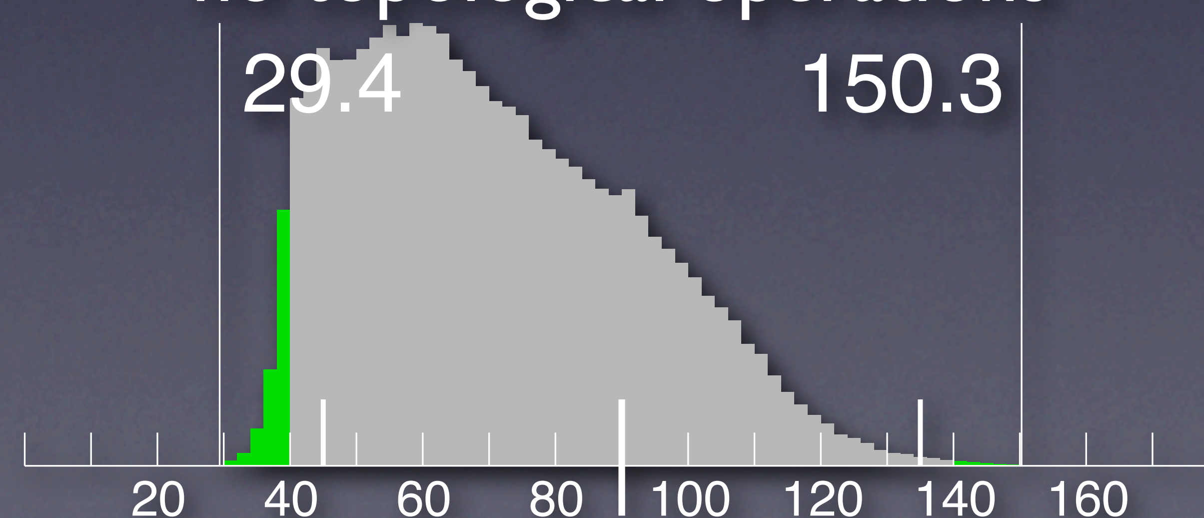
no vertex insertion



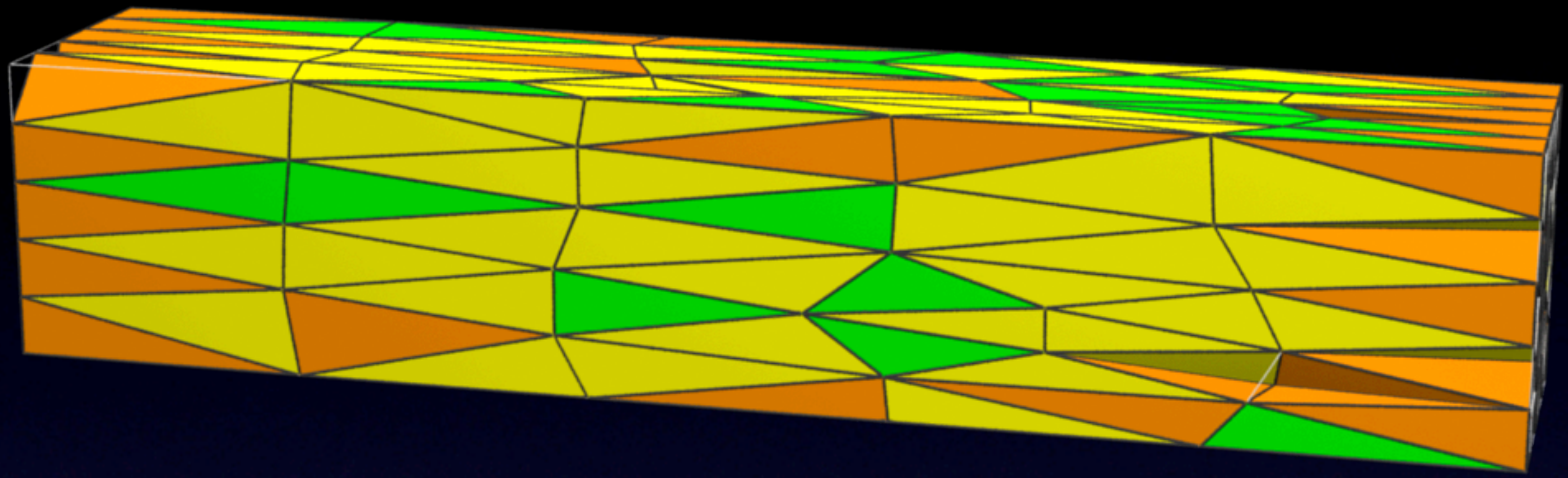
no edge removal



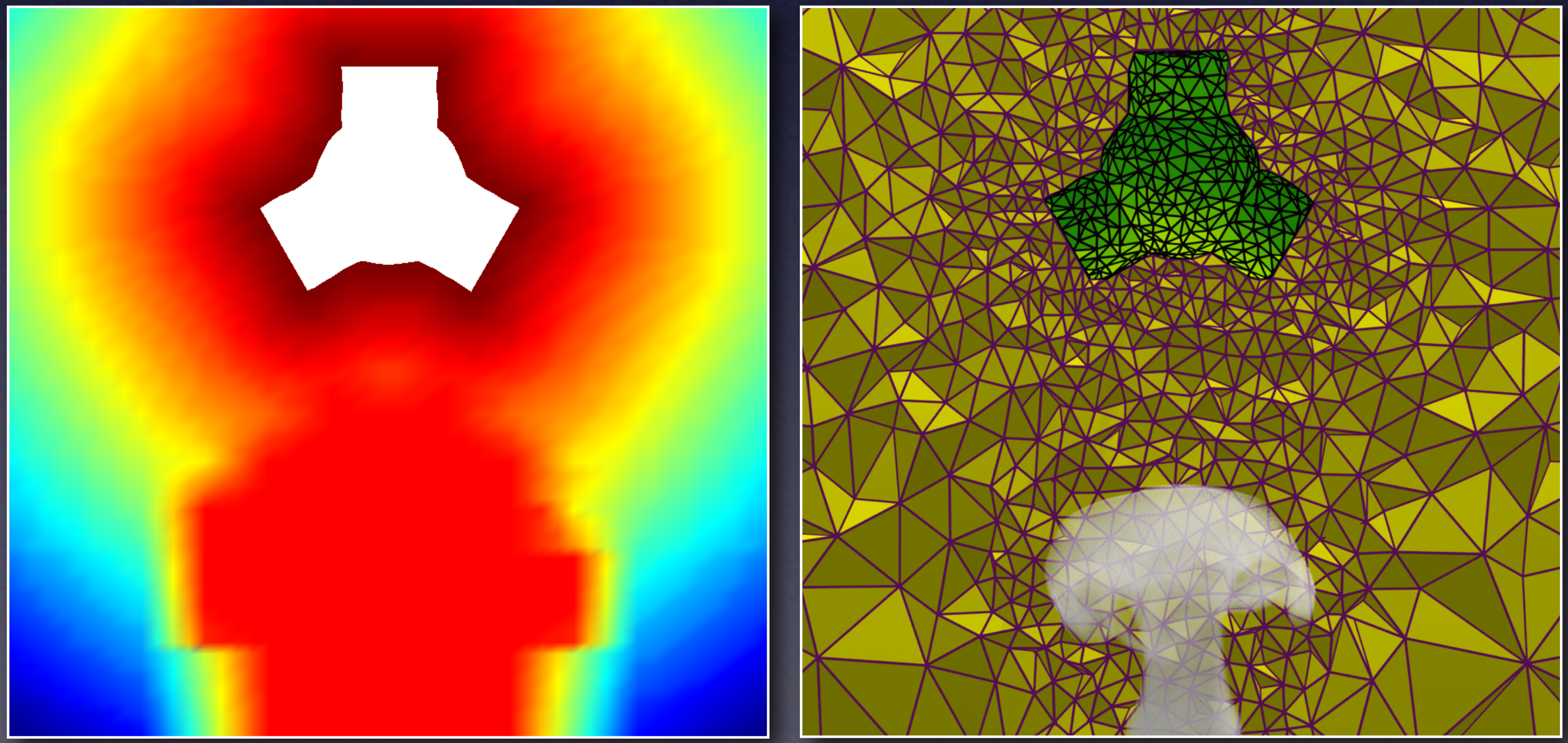
no topological operations



The next steps



anisotropy



adaptivity