Spring 2016

Programming Assignment 1 - Unfolding heuristics

Yunjoo Park *Computer Aided 3D Artifact Fabrication*

1 Unfolding polytope

Let P be a polytope, bounded polyhedron. A simple polygon without overlapping is called *net* N of P. Each of polyhedra has multiple nets. The set of cut edges for the net is a spanning tree T, the *cut tree*, of *G(P)*. An *unfolding* is an isometric mappig Ø: $F(P) \rightarrow \mathbb{R}^2$ of the facets of P to the Euclidean plane, such that for all $(f_1, f_2) \in D(P), \varphi(f_1) \cap \varphi(f_2)$ is an edge of both $\varphi(f_1)$ and $\varphi(f_2)$. $N(P,T) = \varphi(F(P))$ is the unfolding of P *induced by* T. In this project, I will provide two implementation of unfolding heuristics from 'Schlickenrieder, Wolfram. "Net of Polyhedra." Master's Thesis, Technishe Universität Berlin (1997)' [1].

2 Summary of four methods

In this section, I will summarize four methods:

- Steepest Edge Unfolding (SEU)
- Flat Edge Unfolding (FEU)
- Greatest increasing Edge Unfolding (GIU)
- Rightmost ascending Edge Unfolding (REU)

Two methods, SEU and FEU, are given methods. I implement GIU and REU.

An objective function c is used in optimization; minimize or maximize.

2.1 Steepest Edge Cut Tree

If c is an objective function in general position, let v_- and v_+ be the bottom and top vertex of P with regard to c. For each vertex except v_+, we choose the *steepest ascending edge* at that vertex as a cut edge. This lead to straight paths from v_- to v_+.

Algorithm 1 STEEPEST-EDGE-UNFOLD (P, c)

INITIALIZE $T = \emptyset$
for all vertices $v \in P,\ v \neq v_+$ **do**
  Compute the steepest ascending edge incident to v (w.r.t. c),
   i.e. the edge e = (v,w), for which $\frac{\langle c, w-v \rangle}{\|w-v\|}$ *is maximal.*
  $T = T \cup \{e\}$
end for
return CUT (P, T)

2.2 Flat Edge Unfolding

Let c be a given vector. We join facets along *flat* edges, which are as perpendicular as posible to the given vector c. Assign *weights* to each edge $e^* \in D(P)$ where $e = (p, q)$ is the corresponding primal edge of e^*. Then compute a *minimum spanning tree* T $D(P)$ with regard to these weights, and use this tree as the join tree. For each vertex, we choose the *steepest ascending edge* at that vertex as a cut edge.

Algorithm 2 FLAT-SPANNING-TREE-UNFOLD (P, c)

```
INITIALIZE T = ∅
for all edges e = (p,q) ∈ P do
    let e* be the corresponding dual edge of e
    let weights(e*) = | ⟨c,p−q⟩ / (||c|| ||p−q||) |
end for
compute a minimal spanning tree T in D w.r.t. weights
return JOIN (P,T)
```

2.3 Greatest Increase Cut Tree

Let c be an objective function in general position and v_+ the highest vertex. For each vertex except v_+, among all ascending edges incident to v, we choose the edge with *greatest increasing* with regard to c.

Algorithm 3 GREATEST-INCREASE-UNFOLD (P, c)

```
INITIALIZE T = ∅
for all vertices v ∈ P, v ≠ v+ do
    Compute the edge with greatest increase (w.r.t. c) incident to v,
      i.e. the edge e = (v,w), for which ⟨c, w − v⟩ is maximal.
    T = T ∪ {e}
end for
return CUT (P,T)
```

2.4 Rightmost Ascending Cut Tree

Let c be a vector in general position and v_- the lowest and highest vertex of P with regard to c. For each vertex $v \in V(P)$ except v_-, we ccompute the *rightmost ascending edge* incident to v.

The barycenter of P is the centroid of a polygon P. The centeroid of P with uniform density can be computed as the weighted sum of the centroids of a partition of the polygon into triangles.

First Triangulate a polygon P.

Second Compute the centroid of P.

1. Form a sum of the centroids of each triangle, weighted by the area of each triangle.
2. Nomalize the whole sum by the total polyhedron.

Third Create disks

Algorithm 4 Rightmost-Ascending-Edge-Unfold (P, c)

Initialize v_{-} = the minimal vertex of P w.r.t. c
b = the barycenter of P
$T = \emptyset$
for all vertices $v \in P,\ v \neq v_{-}$ **do**
let $e = (v, w)$ *be the rightmost ascending edge incident to* v,
i.e. the edge, for which $\frac{\langle c, w-v \rangle}{\|w-v\|\|c\|} > 0$, *and* $det(c, \frac{v-b}{\|v-b\|}, \frac{w-v}{\|w-v\|})$ *maximal.*
$T = T \cup \{e\}$
end for
return Cut (P, T)

3 Results

I test four heuristics methods on 5 convex models and 5 non-convex models.

3.1 Convex Models

Shape of the models is as follows.

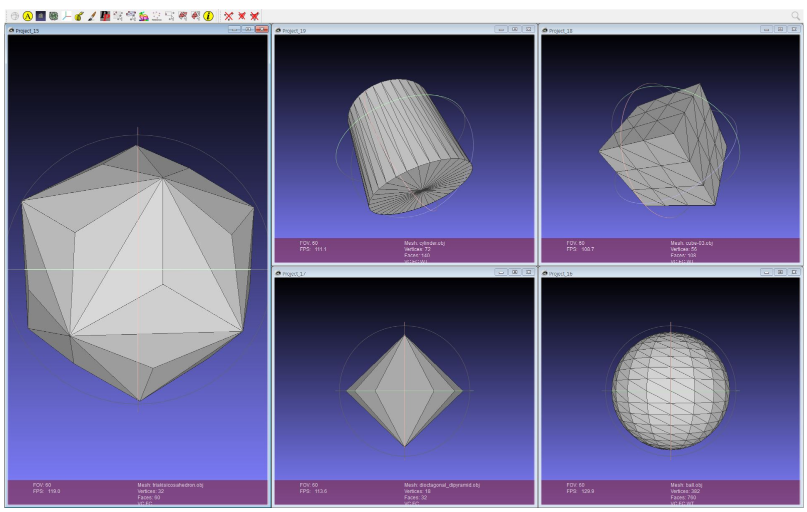
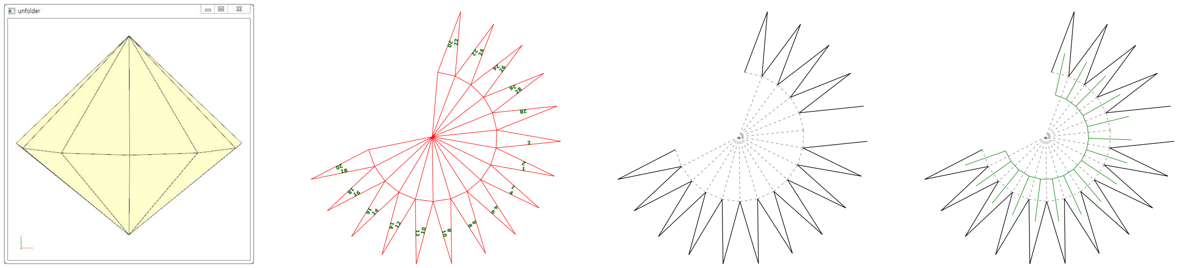


Figure 1: The shape of convex models

The number of faces is as follows.

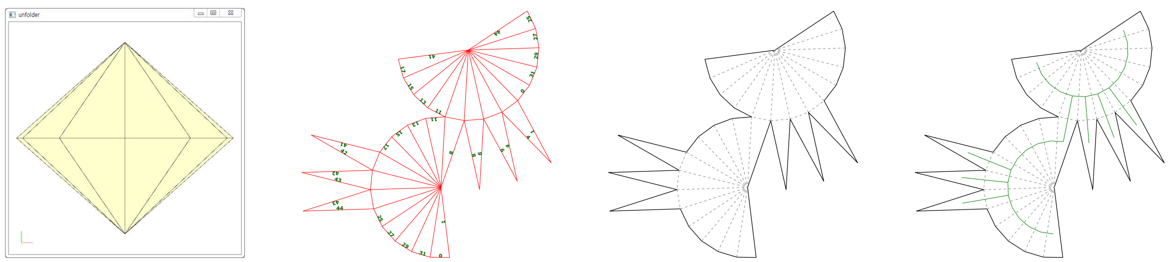
Table 1: 5 convex models

	dioctagonal pyramid	tirakisicosahedron	cylinder	cube-03	ball
Faces	32	60	140	108	760



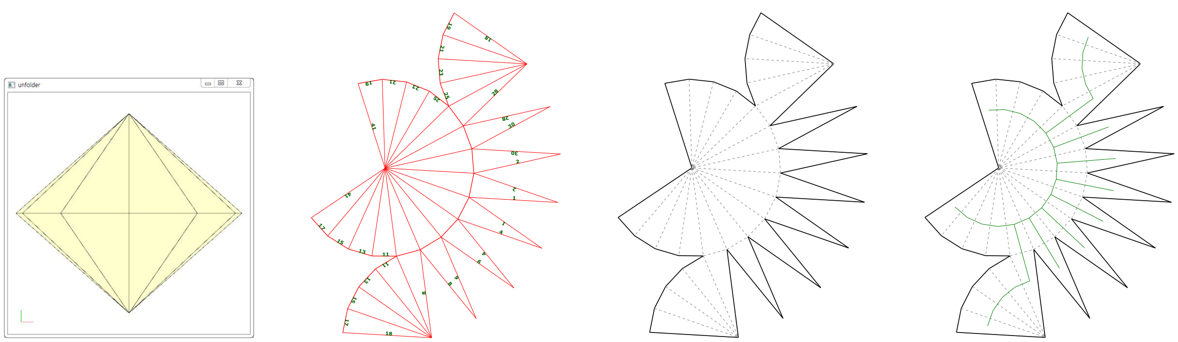
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 2: The Steepest Edge Unfold



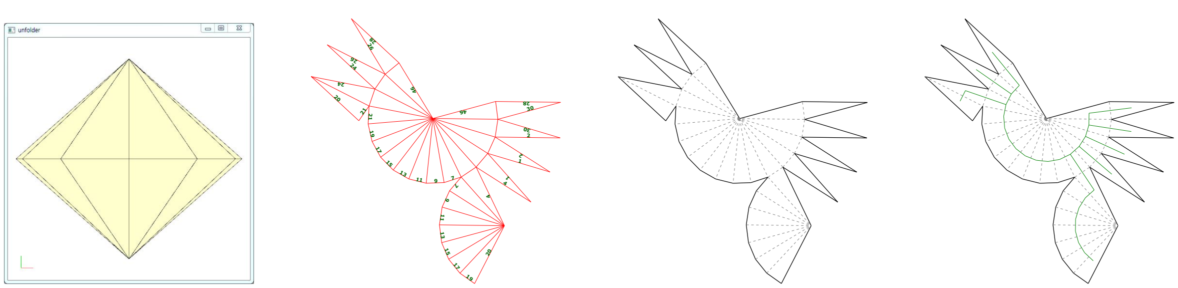
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 3: The Flat Edge Unfold



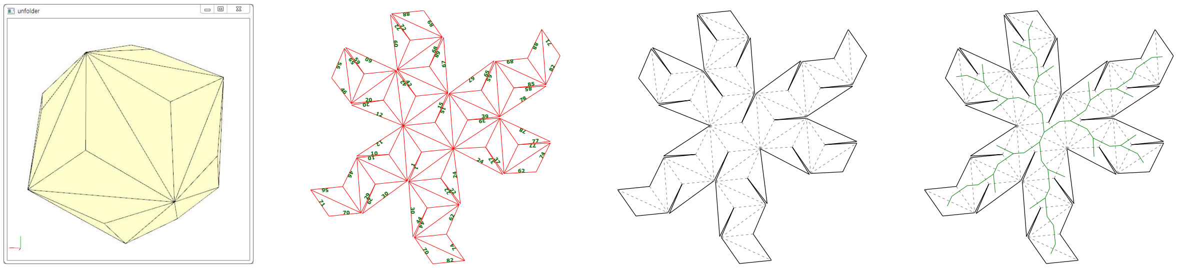
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 4: The Greatest Increase Unfold



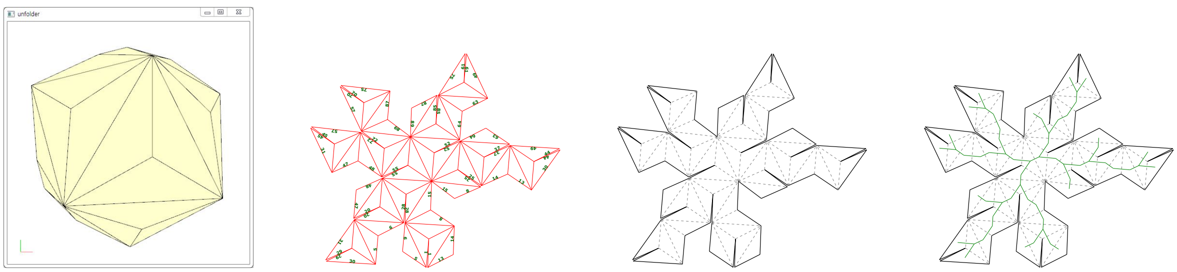
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 5: The Rightemost Ascending Edge Unfold



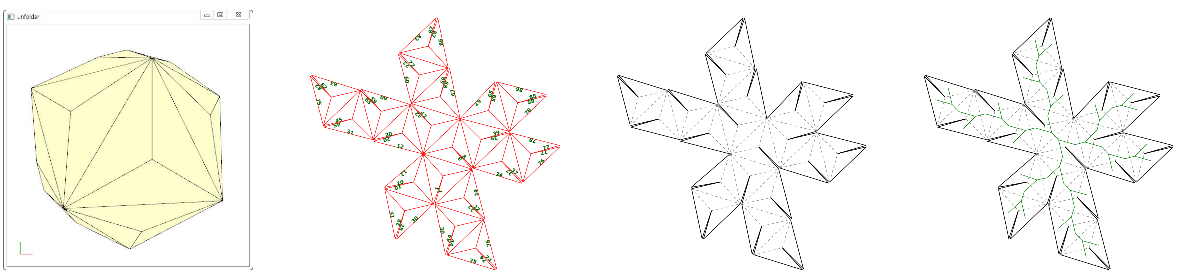
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 6: The Steepest Edge Unfold



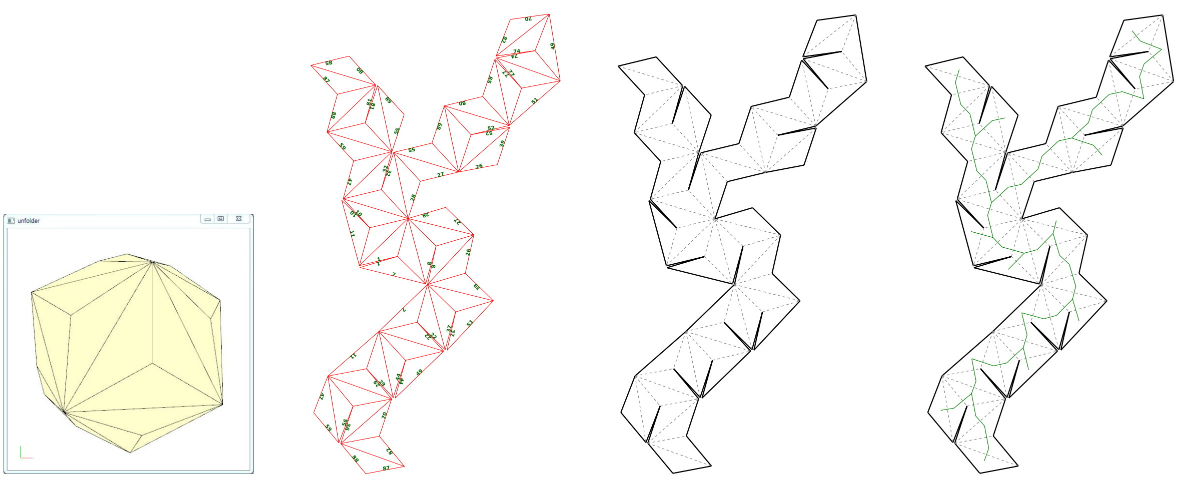
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 7: The Flat Edge Unfold



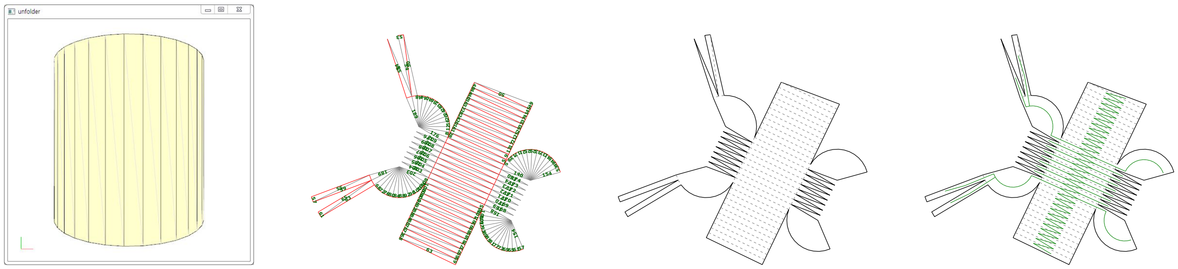
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 8: The Greatest Increase Unfold



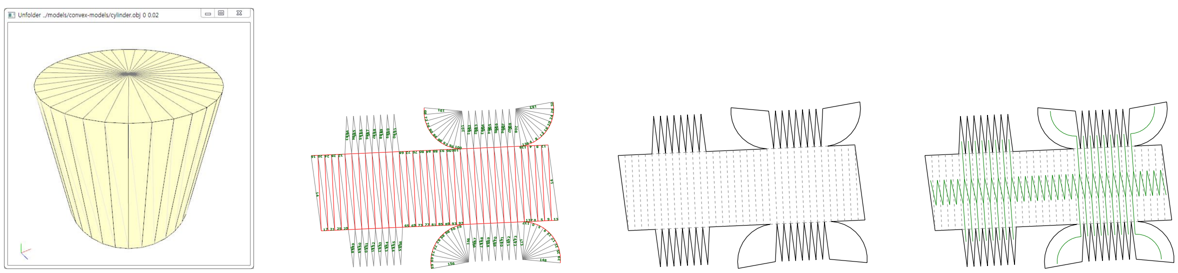
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 9: The Rightemost Ascending Edge Unfold



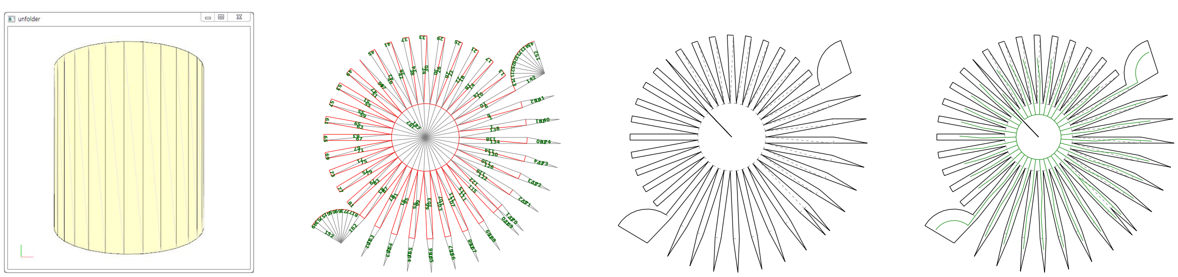
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 10: The Steepest Edge Unfold



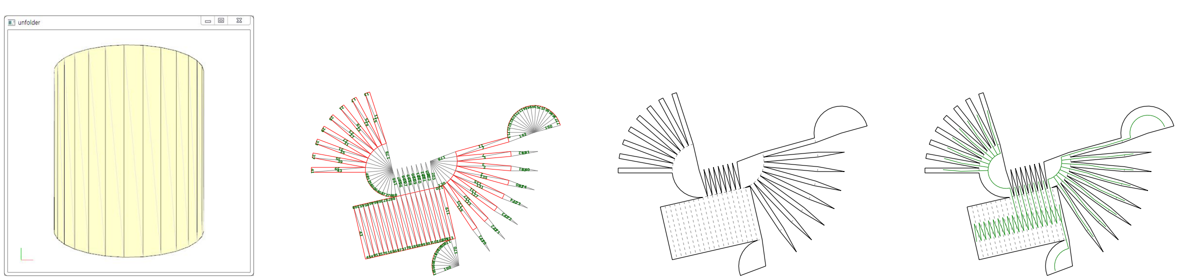
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 11: The Flat Edge Unfold



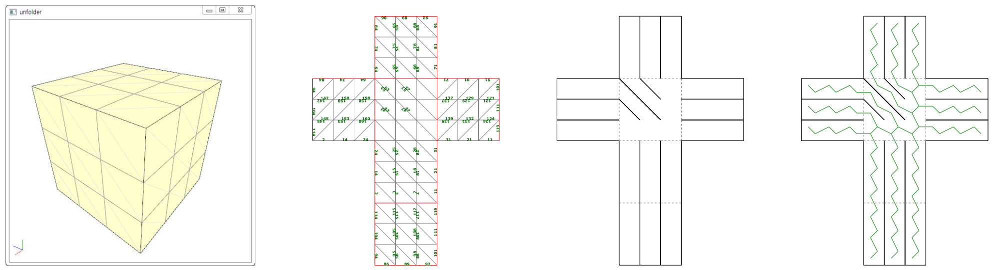
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 12: The Greatest Increase Unfold



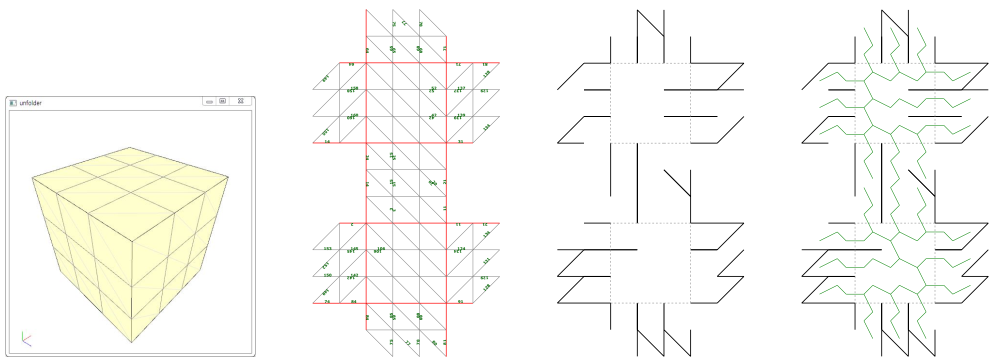
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 13: The Rightemost Ascending Edge Unfold



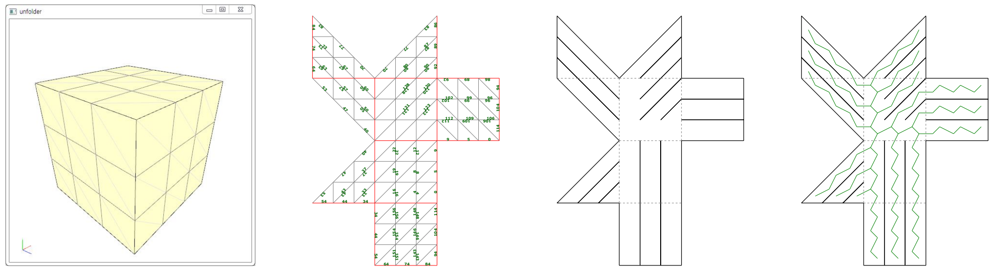
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 14: The Steepest Edge Unfold



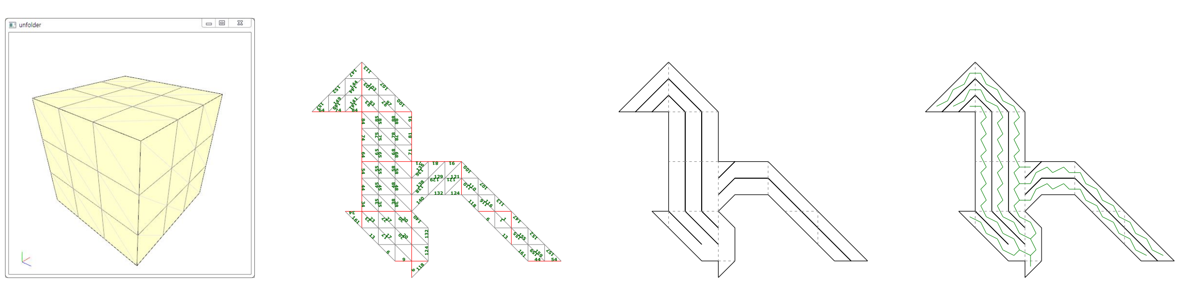
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 15: The Flat Edge Unfold



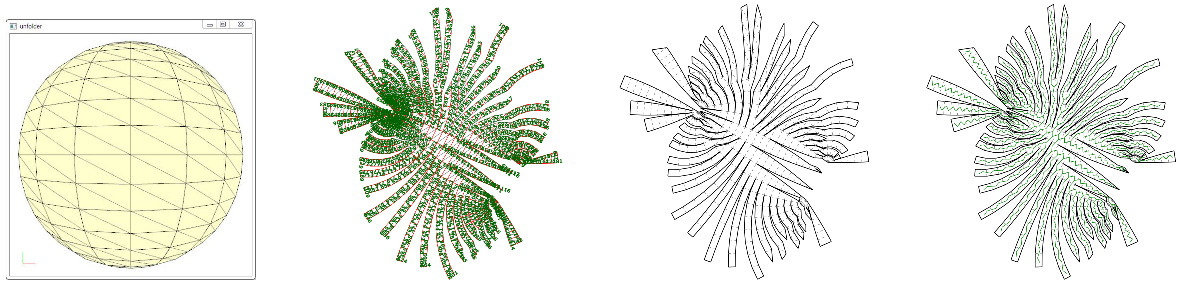
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 16: The Greatest Increase Unfold



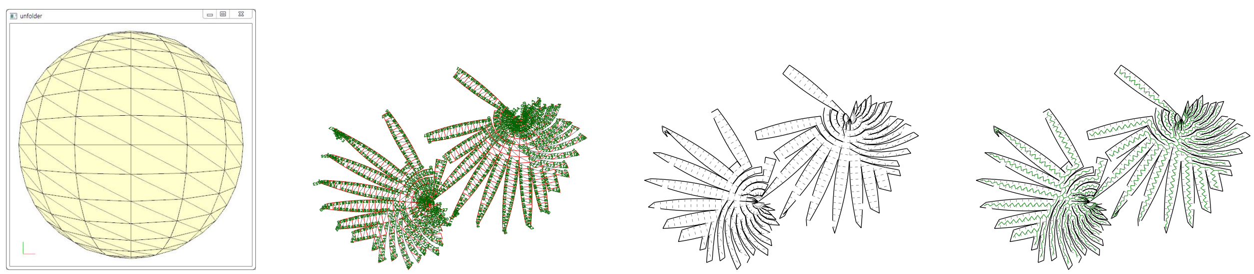
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 17: The Rightemost Ascending Edge Unfold



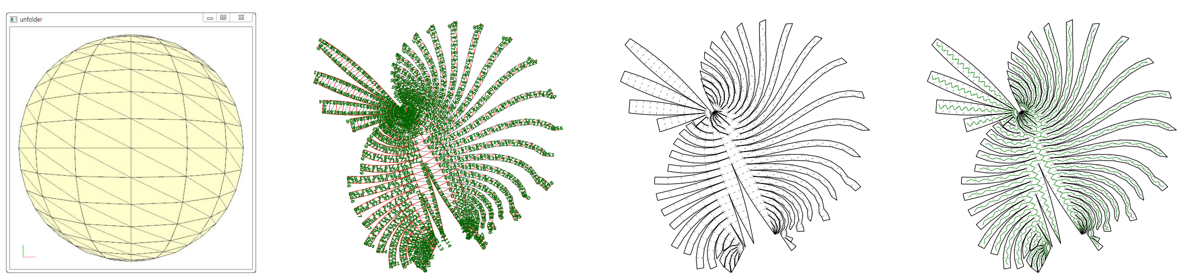
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 18: The Steepest Edge Unfold



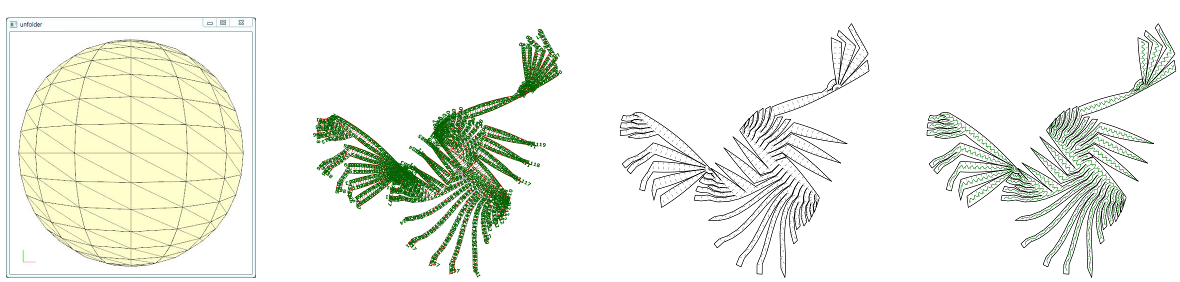
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 19: The Flat Edge Unfold



(a) model (b) unfolded (c) cut (d) spanning tree

Figure 20: The Greatest Increase Unfold



(a) model (b) unfolded (c) cut (d) spanning tree

Figure 21: The Rightemost Ascending Edge Unfold

3.2 non-Convex Models

Shape of the models is as follows.

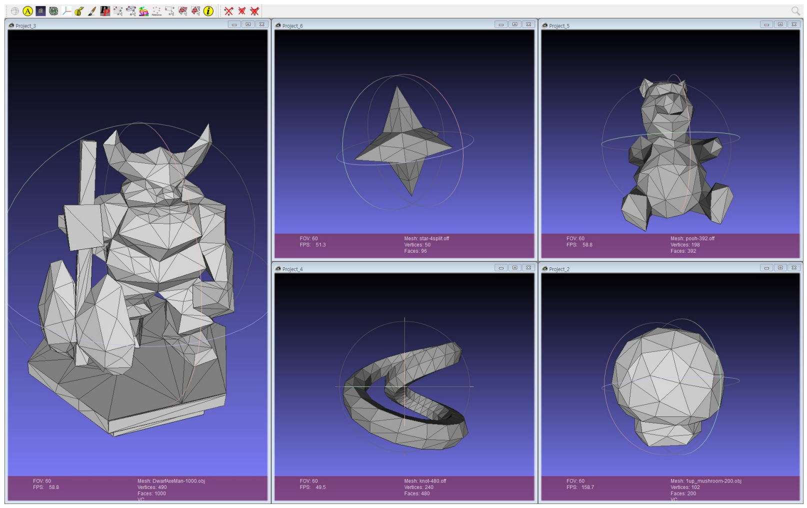


Figure 22: The shape of non-convexx models

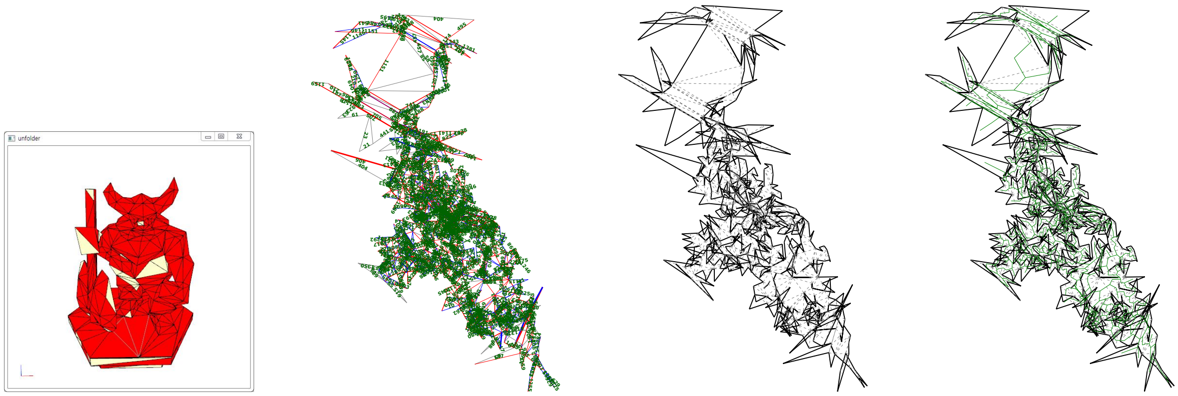
The number of faces is as follows.

Table 2: 5 non-convex models

	DwarfAxeMan	knot	pooh	mushroom	star-4split
Faces	1000	480	392	200	96

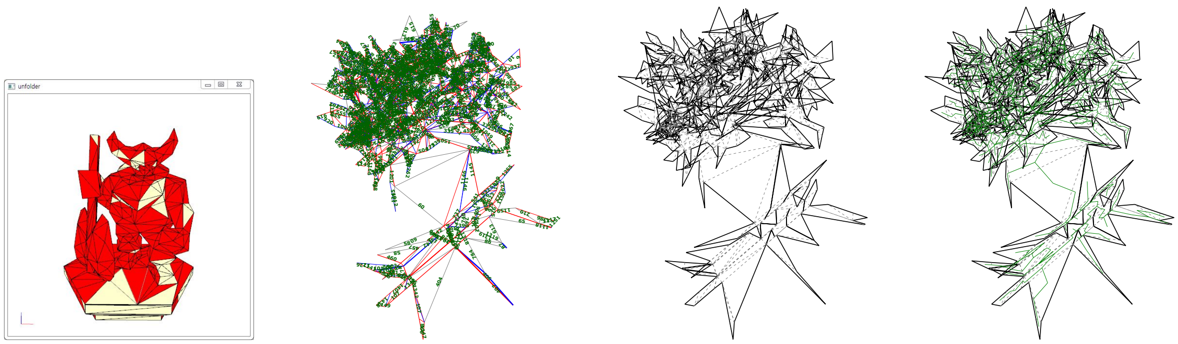
Unlike testing on convex models, unfolding a non-convex model requires many attempts. There is a $-r\ \ n$ option. It implies that it attempts at most n times until the mesh is unfolded without overlapping. Especially, I tried 1000 times.

- Dwarf Axe Man
  All mehtods have similar number of overlapping. Compared to other three methods, the rightmost ascending edge unfolding method takes more time.
- Knot
  All methods cannot generate a unfolded polygon without overlapping. They take similar time and generate similar number of overlappings.
- Pooh
  The rightmost ascending edge unfolding method takes a little more time. All methods fail to generate unfolded polygon without overlappings.
- Mushroom
  The steepest edge unfolding methods has the smallest number of overlappings in $min/avg/max$ overlaps.
- Star-4split
  Only the rightmost ascending edge unfolding method generates unfolded polygon without overlapping. The method completed unfolding *star* model with only 7 times.



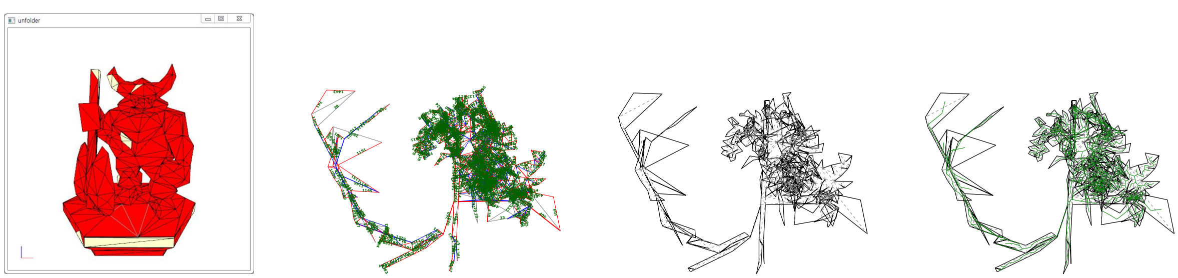
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 23: The Steepest Edge Unfold



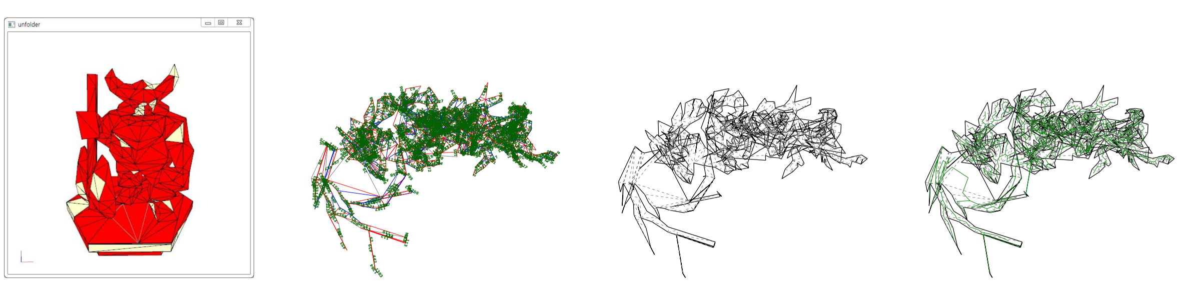
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 24: The Flat Edge Unfold



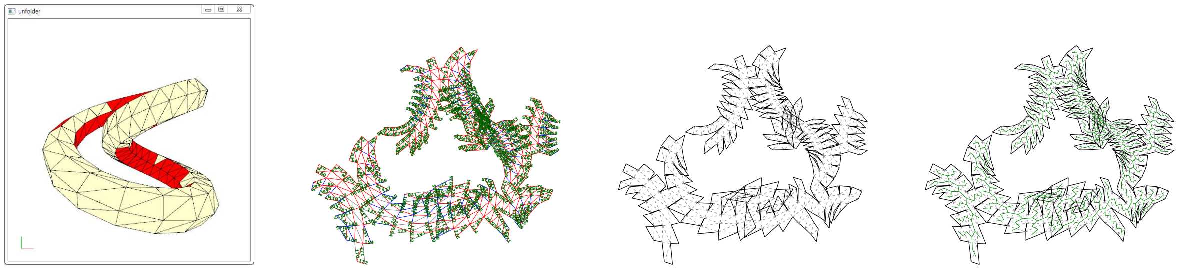
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 25: The Greatest Increase Unfold



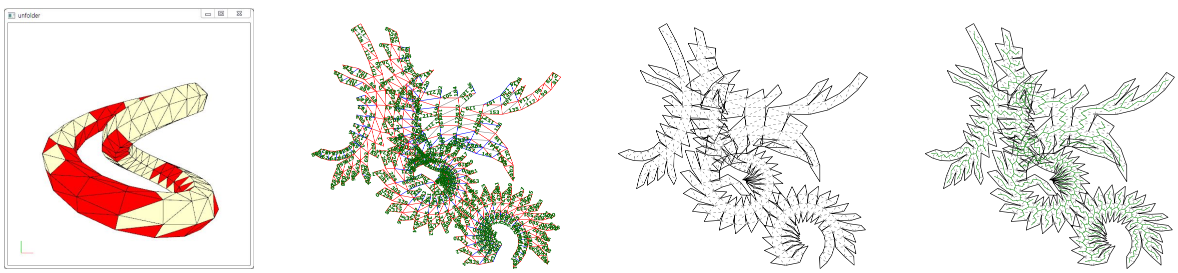
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 26: The Rightemost Ascending Edge Unfold



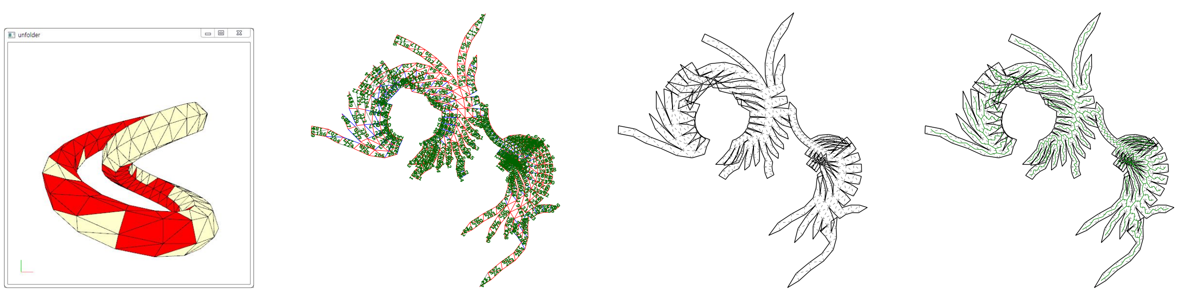
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 27: The Steepest Edge Unfold



(a) model (b) unfolded (c) cut (d) spanning tree

Figure 28: The Flat Edge Unfold

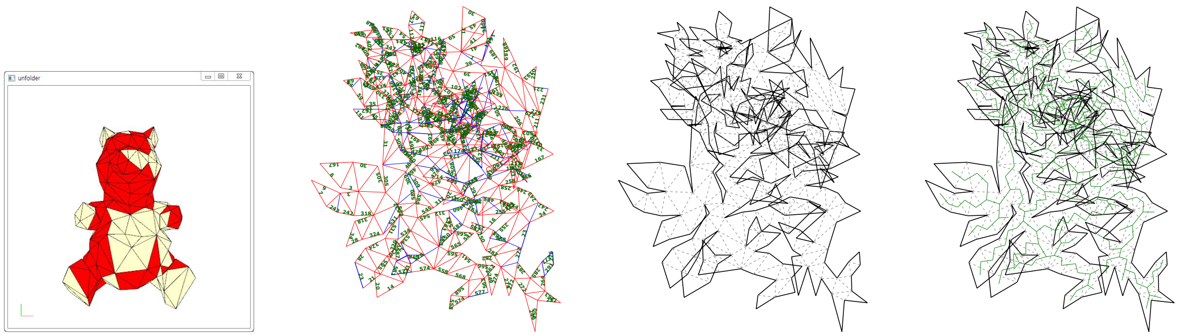


(a) model (b) unfolded (c) cut (d) spanning tree

Figure 29: The Greatest Increase Unfold

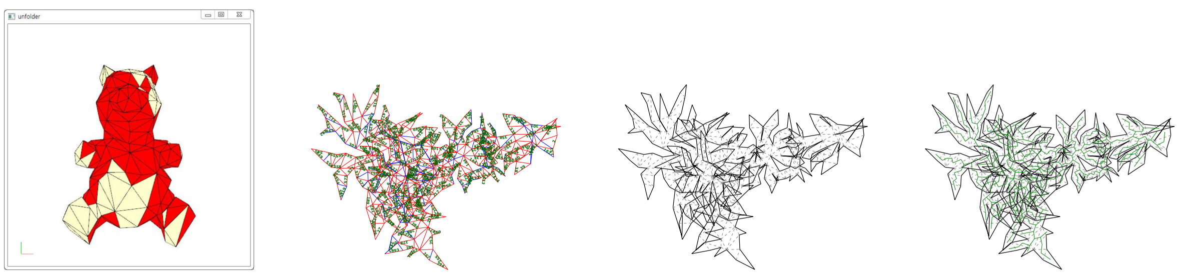
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 30: The Rightemost Ascending Edge Unfold



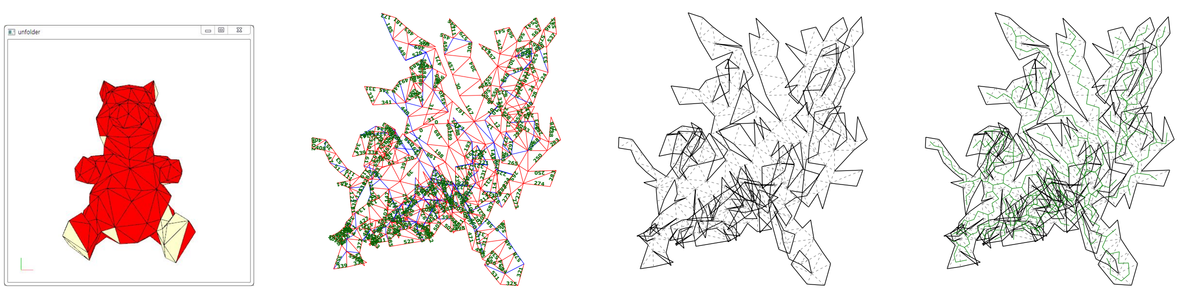
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 31: The Steepest Edge Unfold



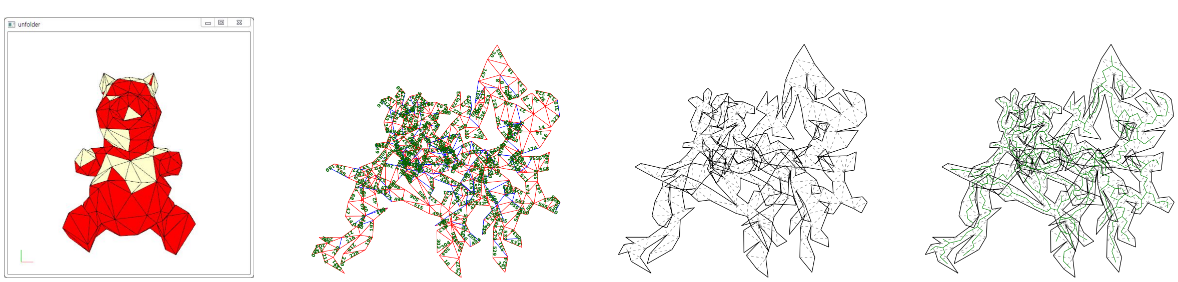
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 32: The Flat Edge Unfold



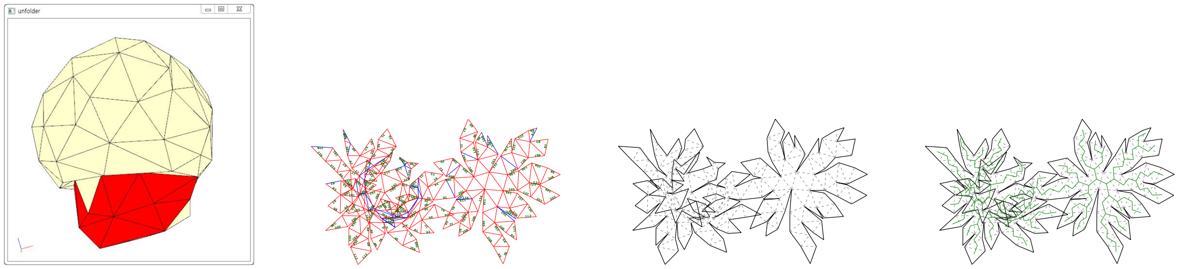
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 33: The Greatest Increase Unfold



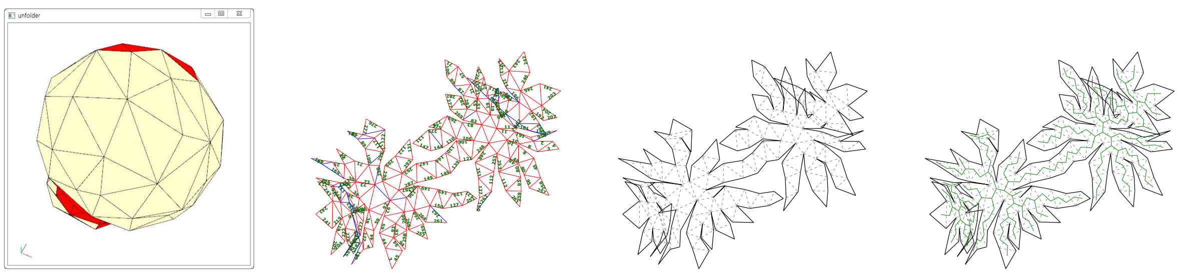
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 34: The Rightemost Ascending Edge Unfold



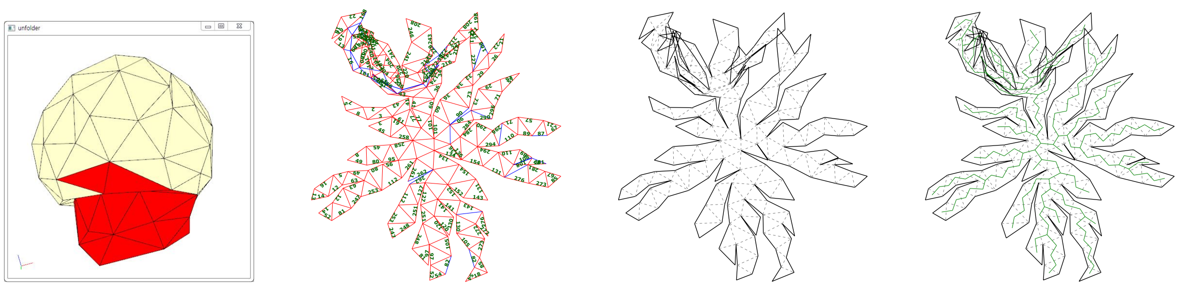
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 35: The Steepest Edge Unfold



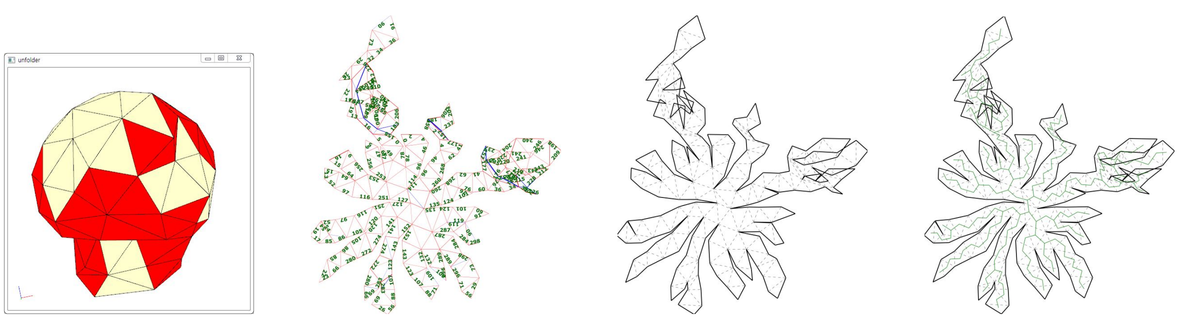
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 36: The Flat Edge Unfold



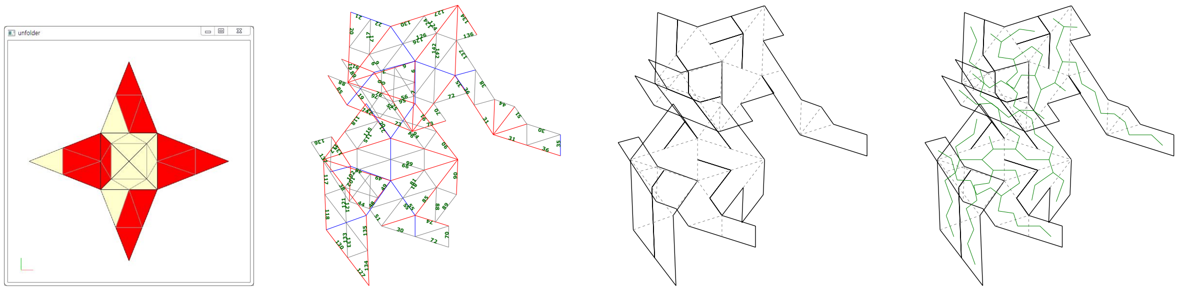
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 37: The Greatest Increase Unfold



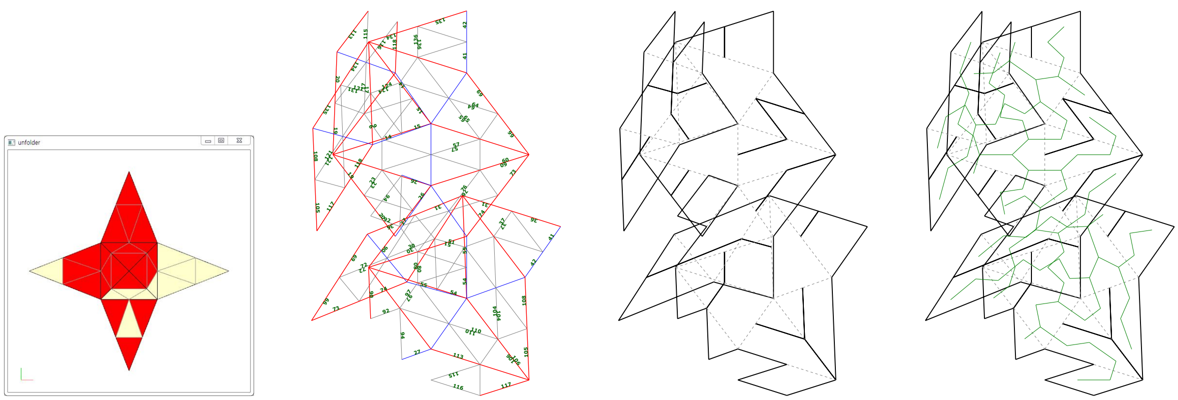
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 38: The Rightemost Ascending Edge Unfold



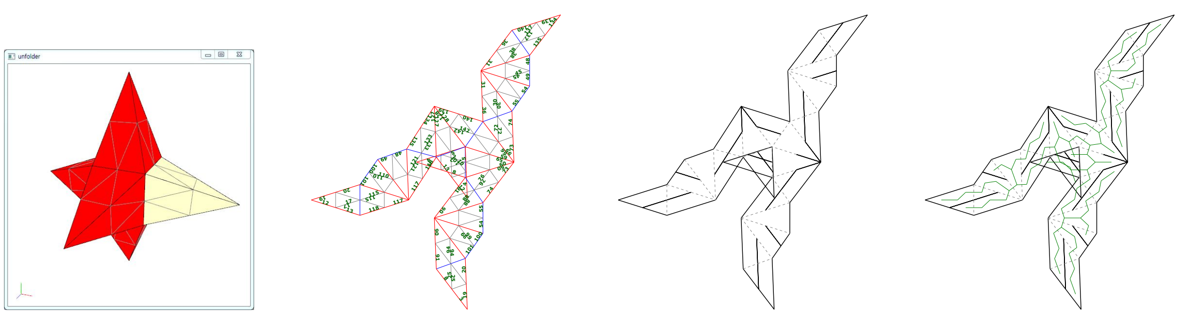
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 39: The Steepest Edge Unfold



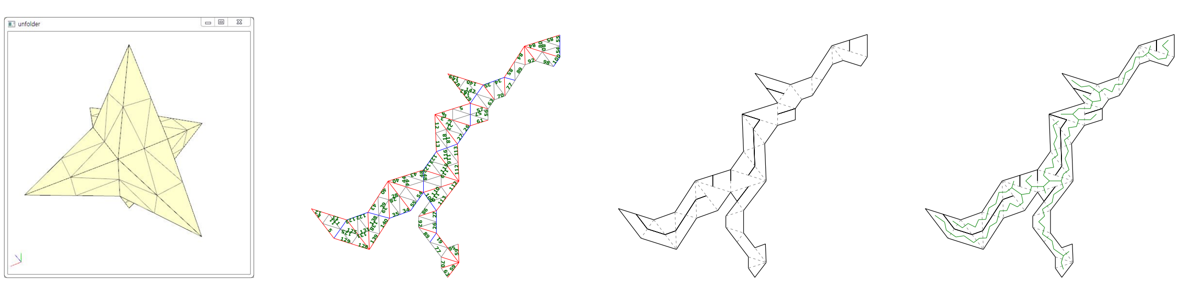
(a) model (b) unfolded (c) cut (d) spanning tree

Figure 40: The Flat Edge Unfold



(a) model (b) unfolded (c) cut (d) spanning tree

Figure 41: The Greatest Increase Unfold



(a) model (b) unfolded (c) cut (d) spanning tree

Figure 42: The Rightemost Ascending Edge Unfold

4 Foundings

Since I use a random vector c with $\|c\| = 1$, I do not care about length. Although four methods do not work well for non-convex model, they successfully generate unfolded polygons for convex models.

- Time
  All methods take similar amount of time. However, the rightmost ascendiing edge unfold method takes more time with a complicated model.
- Overlapping
  The flat edge unfold method has much more than the others.
- Height
  The height of tree of the rightmost ascending edge is higher than the others.

5 GUI Options

It is possible to change back color of unfolded polygons.

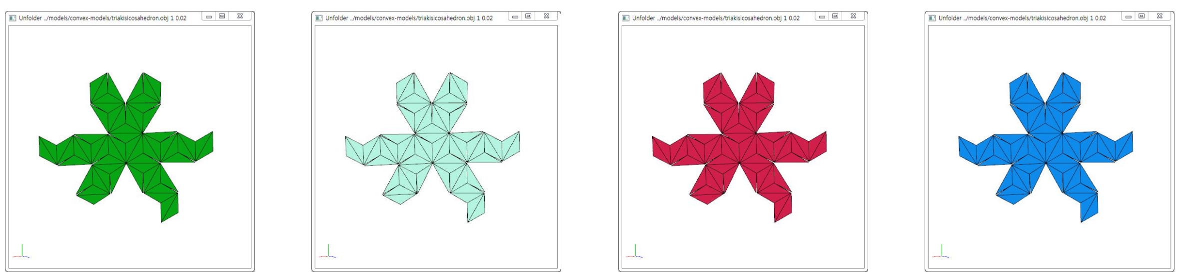


Figure 43: Triakisicosahedron using the steepest edge unfold method

It is possible to generate a unfolded polygon randomly.

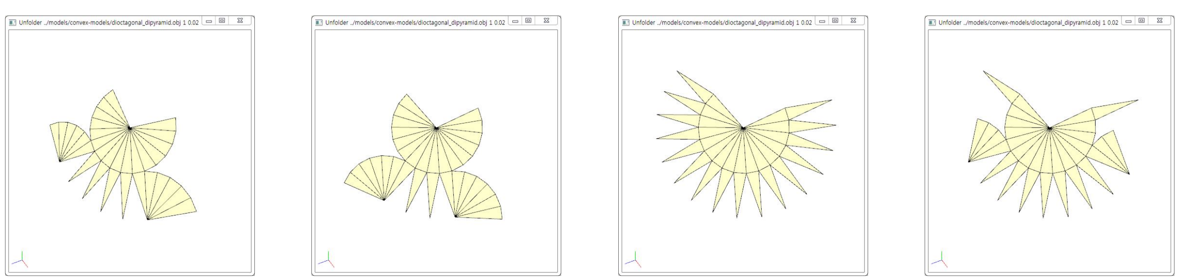


Figure 44: Dioctagonal pyramid using the steepest edge unfold method

References

[1] Wolfram Schlickenrieder. Nets of polyhedra. Master's thesis, Technische Universitat Berlin, 1997.