

## 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 mapping  $\varphi: F(P) \rightarrow \mathbb{R}^2$  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, Technische 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 possible 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*  $T$  in  $D(P)$  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  $T = \emptyset$ 
for all edges  $e = (p, q) \in P$  do
    let  $e^*$  be the corresponding dual edge of  $e$ 
    let  $\text{weights}(e^*) = \left| \frac{\langle c, p-q \rangle}{\|c\| \|p-q\|} \right|$ 
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  $T = \emptyset$ 
for all vertices  $v \in P, v \neq v_+$  do
    Compute the edge with greatest increase (w.r.t.  $c$ ) incident to  $v$ ,
    i.e. the edge  $e = (v, w)$ , for which  $\langle c, w - v \rangle$  is maximal.
     $T = T \cup \{e\}$ 
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 compute the *rightmost ascending edge* incident to  $v$ .

The barycenter of  $P$  is the centroid of a polygon  $P$ . The centroid 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. Normalize 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.

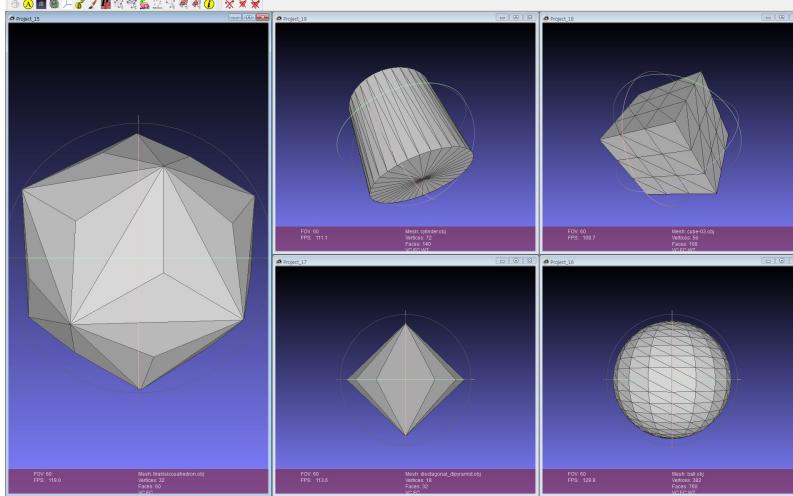


Figure 1: The shape of convex models

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

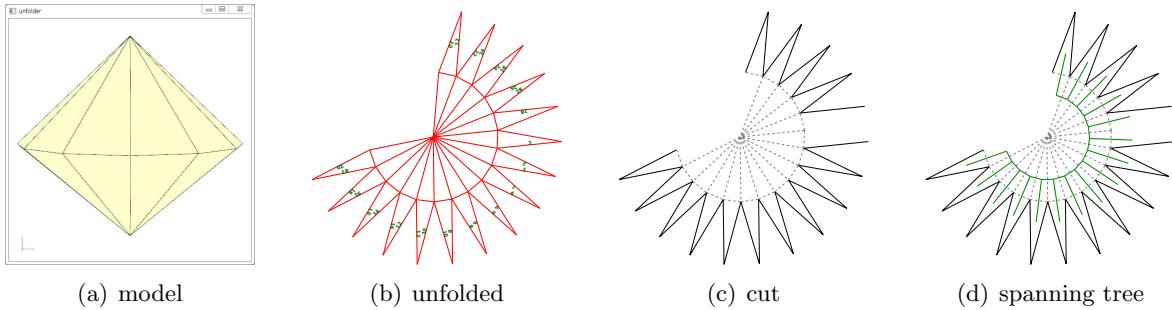


Figure 2: The Steepest Edge Unfold

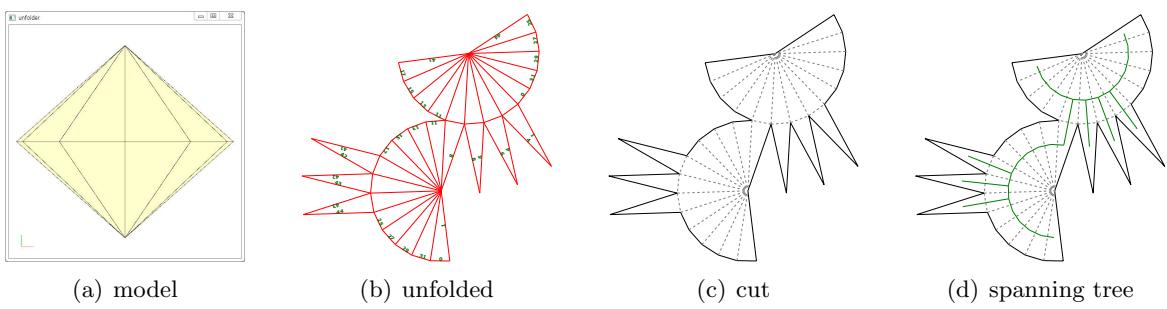


Figure 3: The Flat Edge Unfold

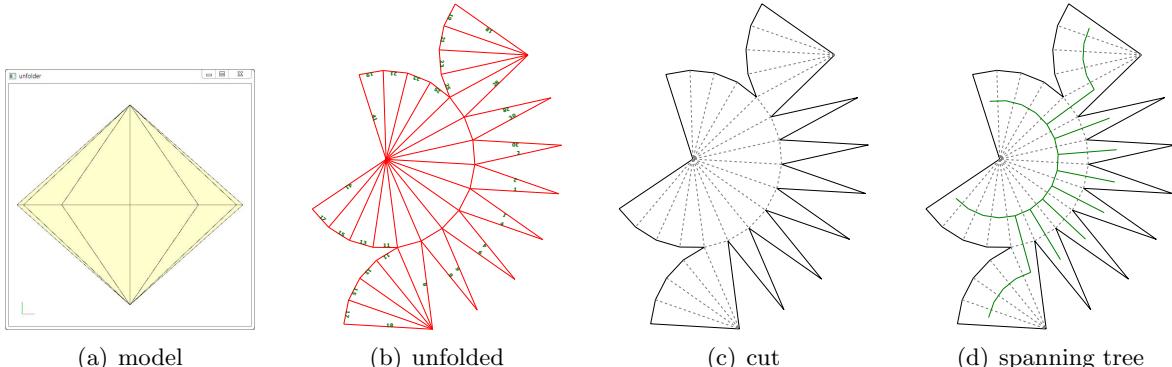


Figure 4: The Greatest Increase Unfold

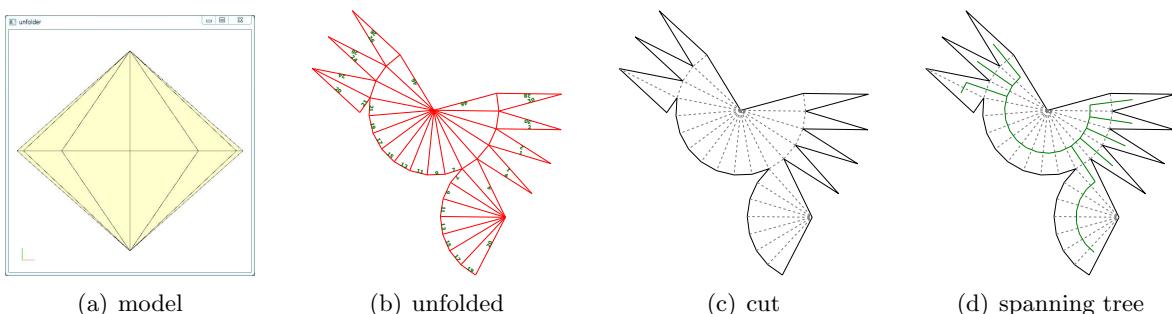


Figure 5: The Rightemost Ascending Edge Unfold

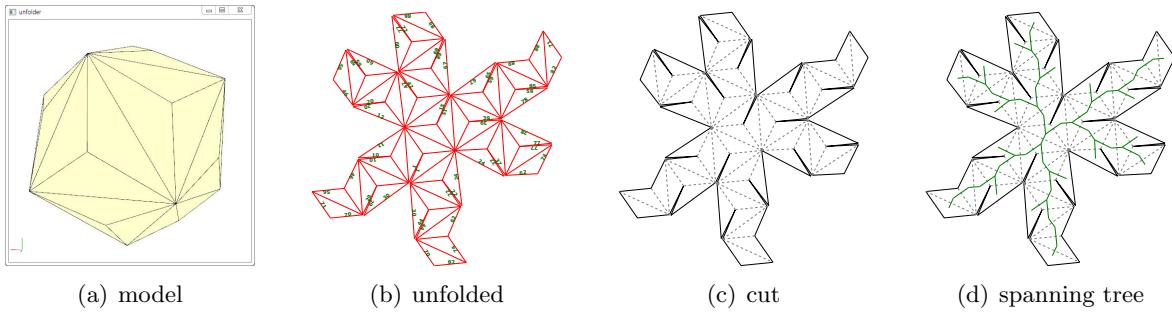


Figure 6: The Steepest Edge Unfold

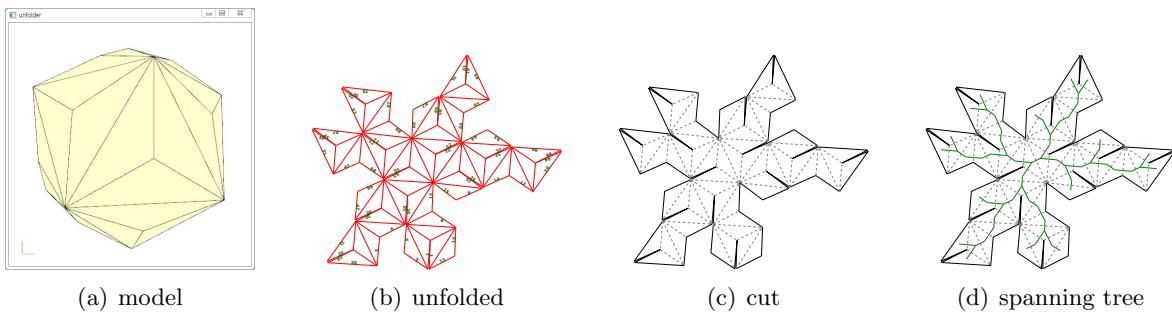


Figure 7: The Flat Edge Unfold

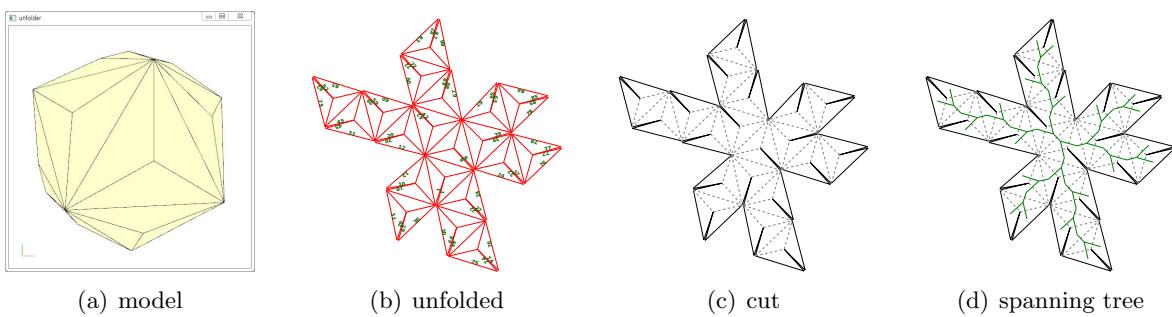


Figure 8: The Greatest Increase Unfold

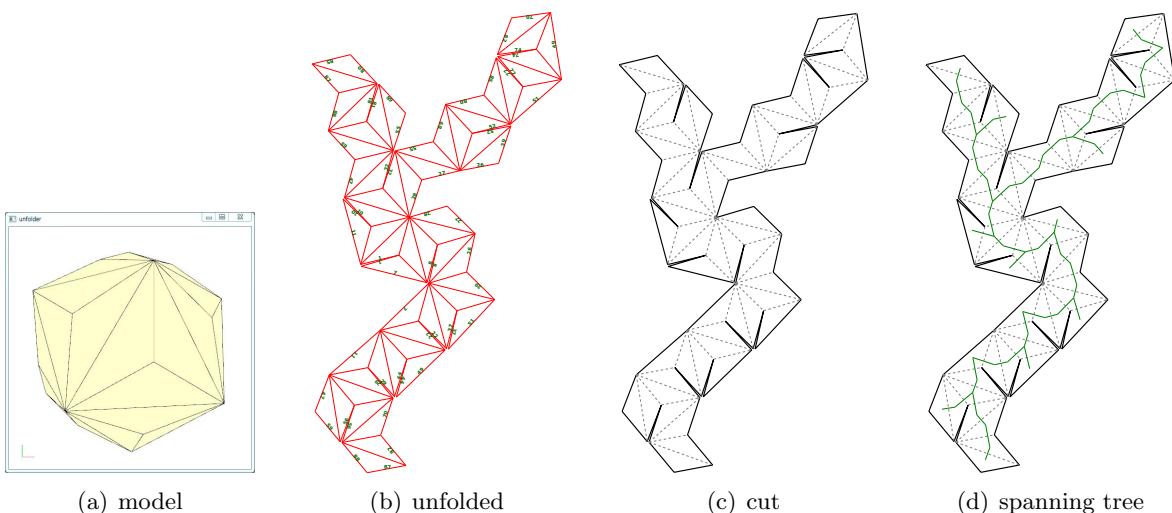


Figure 9: The Rightmost Ascending Edge Unfold

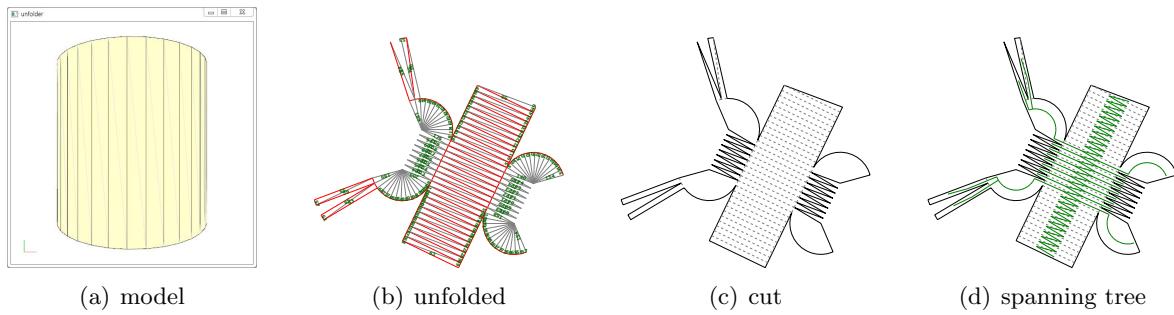


Figure 10: The Steepest Edge Unfold

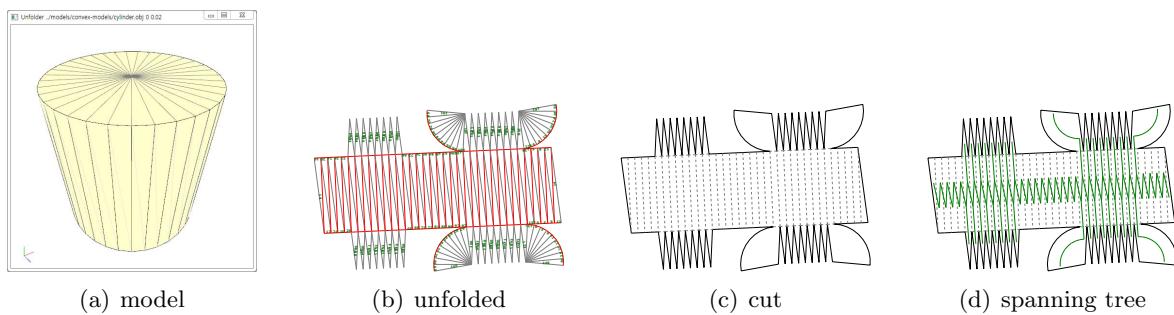


Figure 11: The Flat Edge Unfold

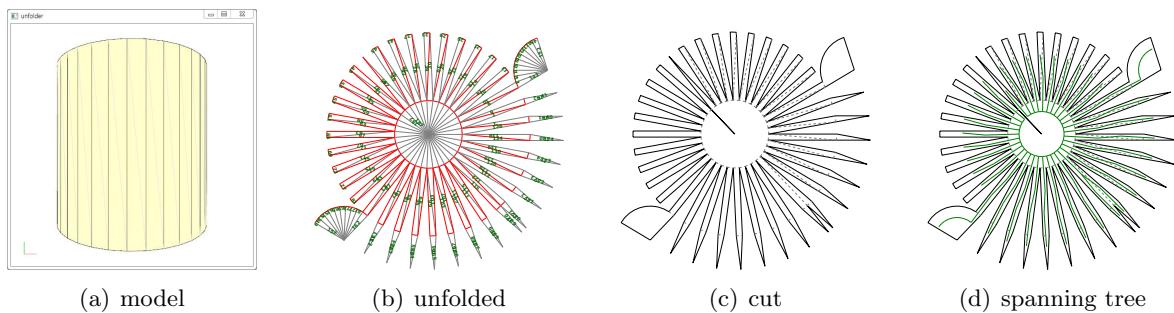


Figure 12: The Greatest Increase Unfold

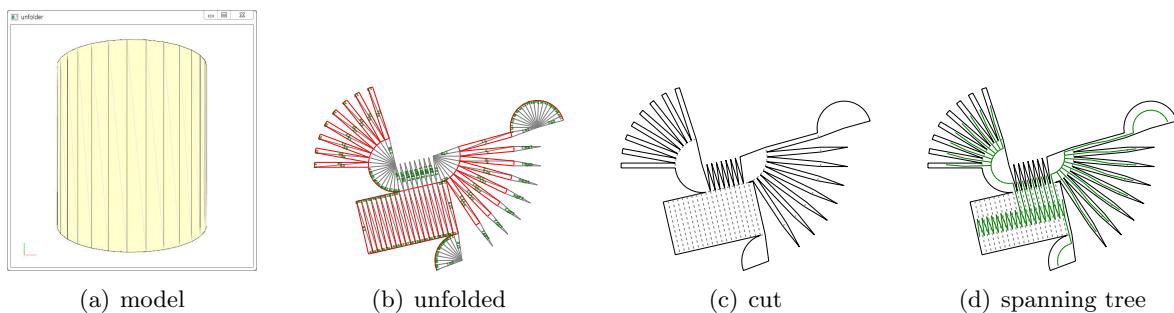


Figure 13: The Rightmost Ascending Edge Unfold

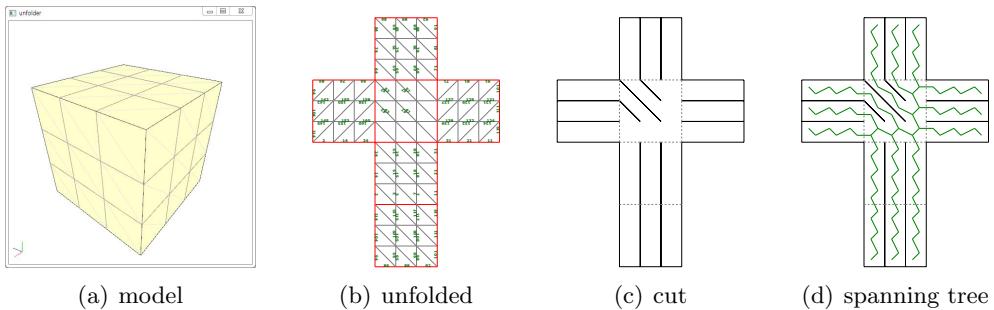


Figure 14: The Steepest Edge Unfold

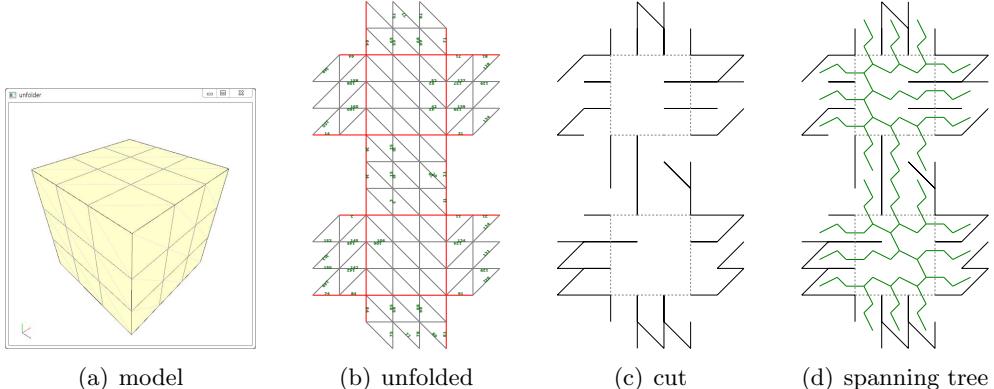


Figure 15: The Flat Edge Unfold

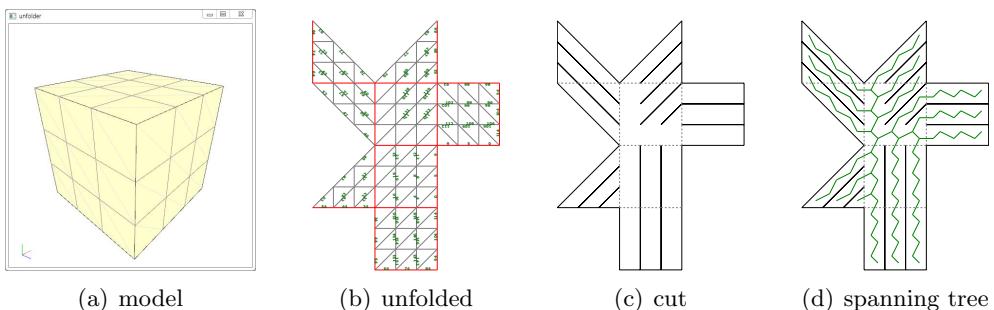


Figure 16: The Greatest Increase Unfold

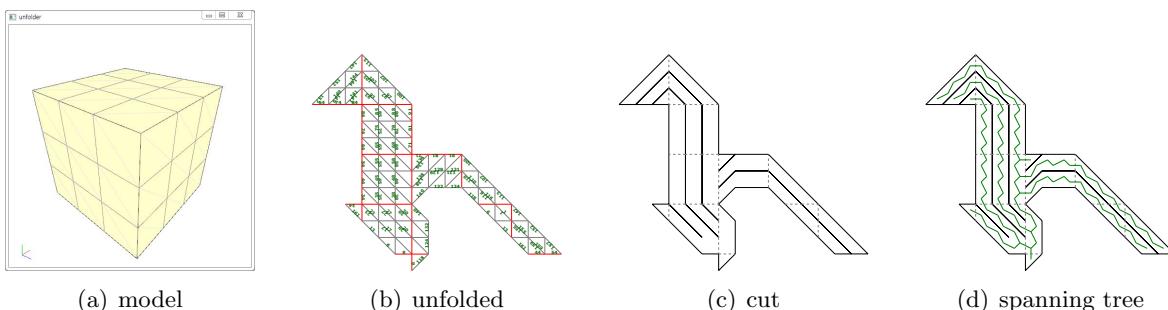


Figure 17: The Rightemost Ascending Edge Unfold

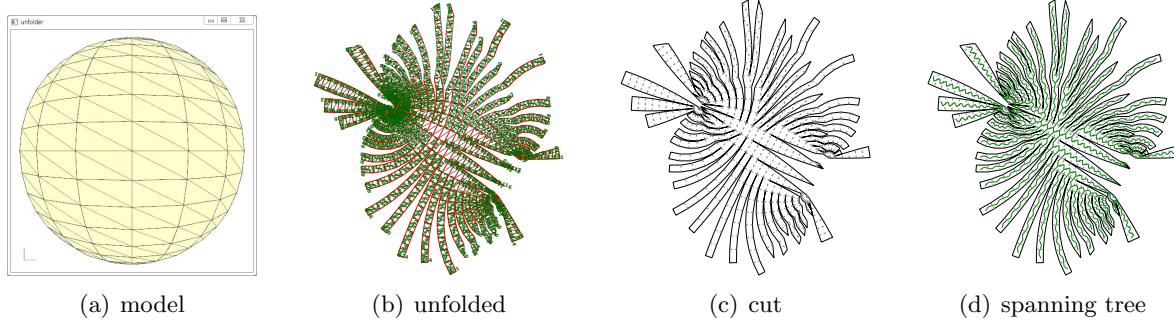


Figure 18: The Steepest Edge Unfold

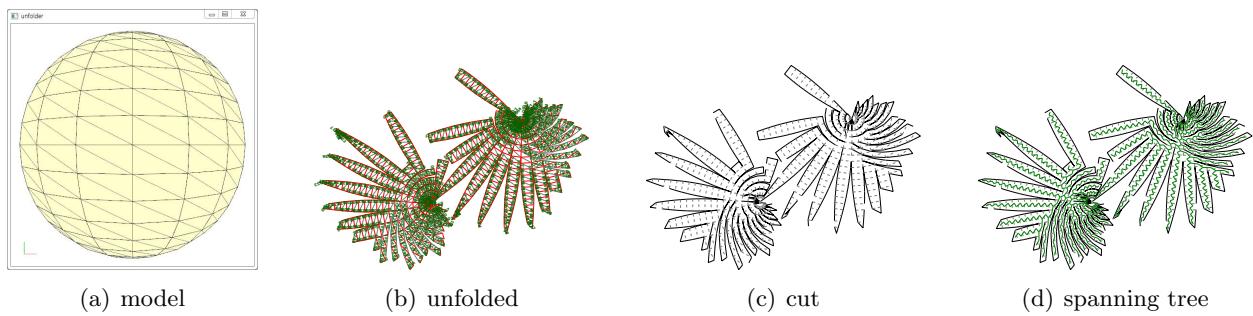


Figure 19: The Flat Edge Unfold

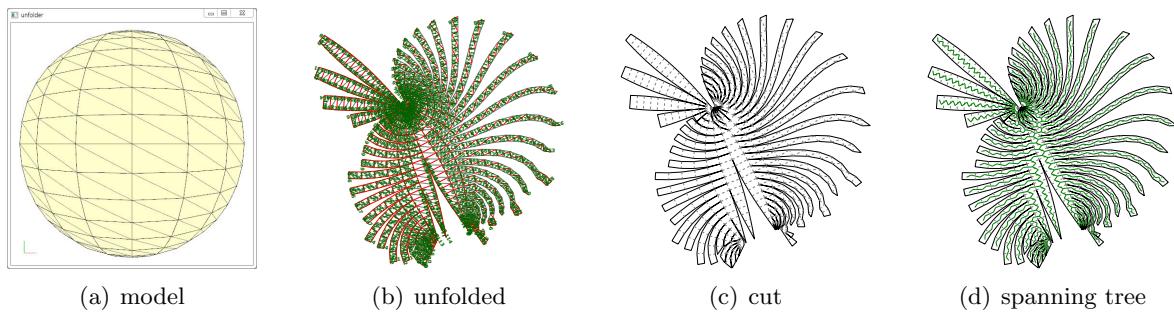


Figure 20: The Greatest Increase Unfold

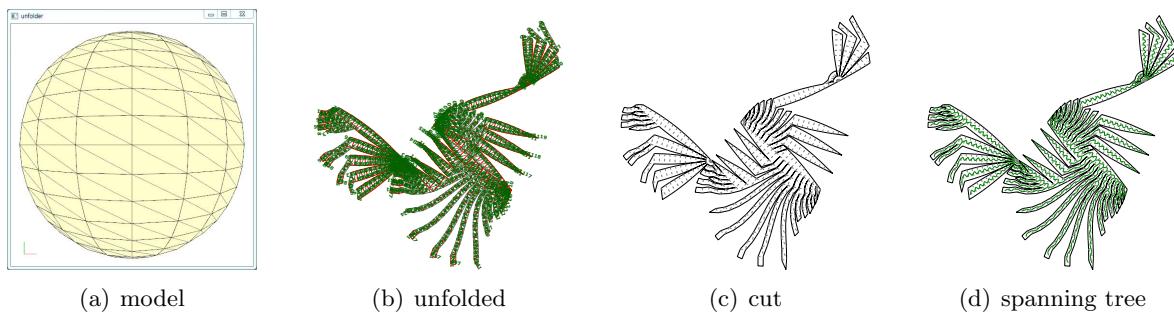


Figure 21: The Rightemost Ascending Edge Unfold

### 3.2 non-Convex Models

Shape of the models is as follows.

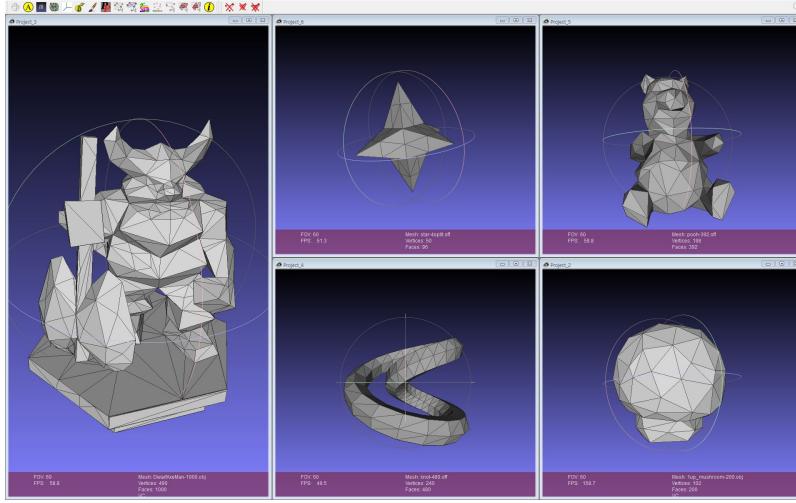


Figure 22: The shape of non-convexx models

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  $\min/\text{avg}/\max$  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.

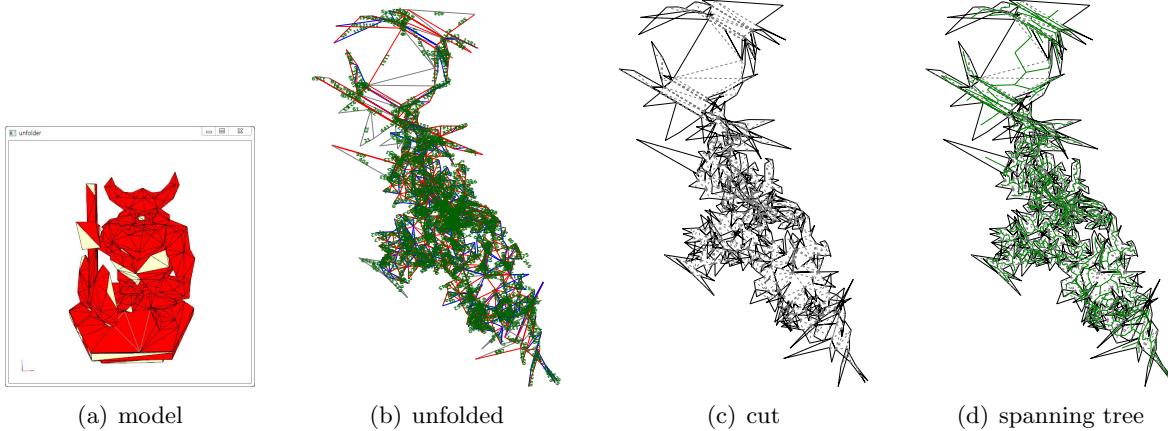


Figure 23: The Steepest Edge Unfold

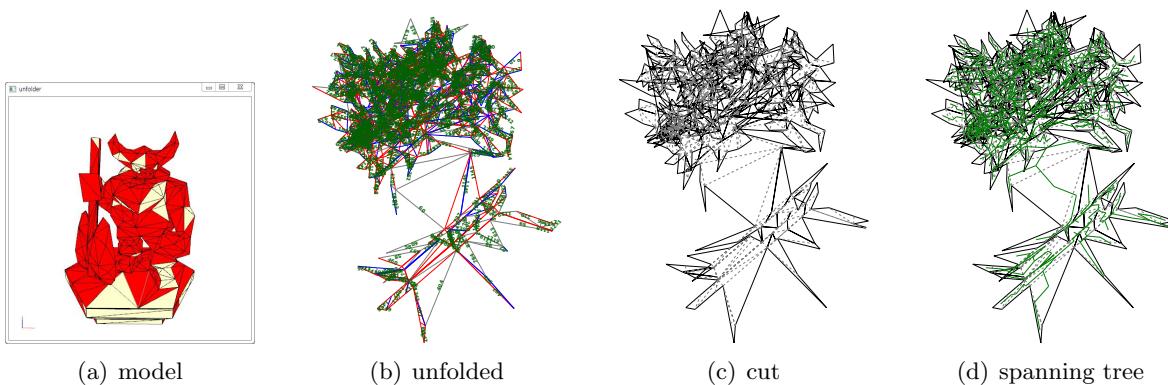


Figure 24: The Flat Edge Unfold

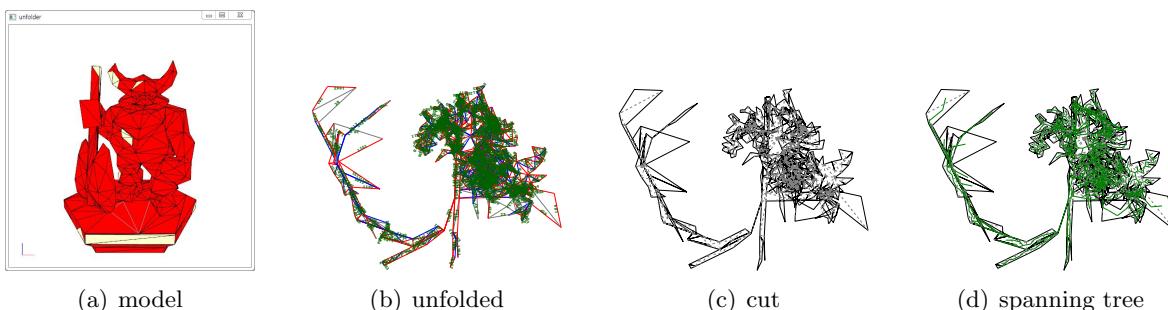


Figure 25: The Greatest Increase Unfold

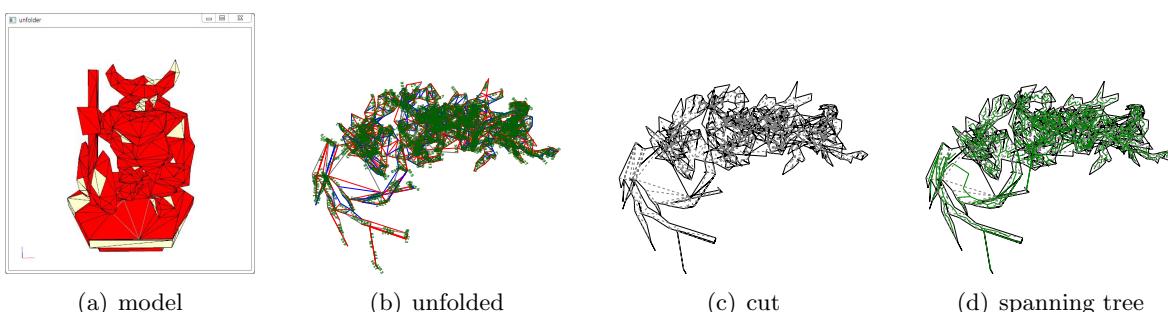


Figure 26: The Rightemost Ascending Edge Unfold

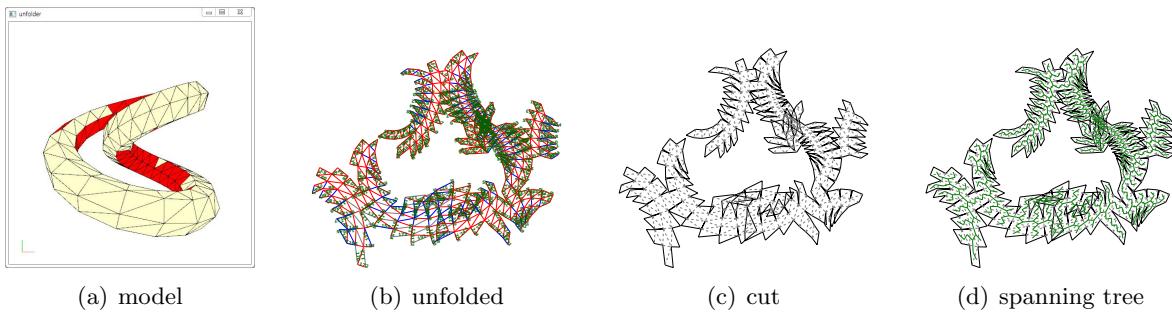


Figure 27: The Steepest Edge Unfold

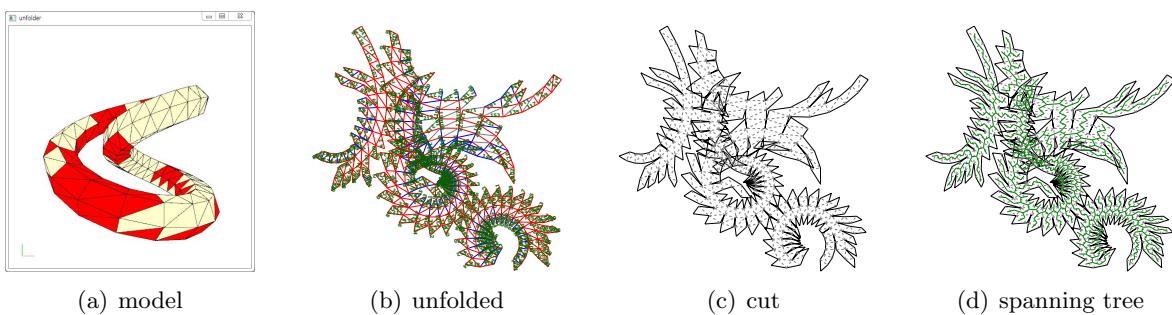


Figure 28: The Flat Edge Unfold

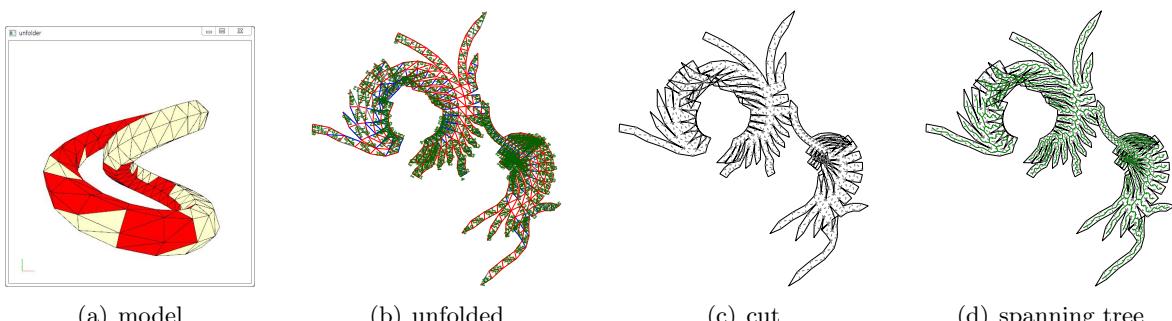


Figure 29: The Greatest Increase Unfold

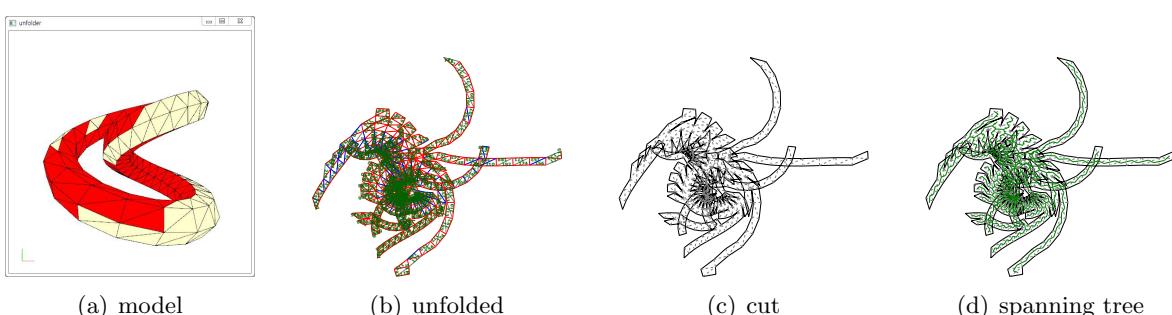


Figure 30: The Rightemost Ascending Edge Unfold

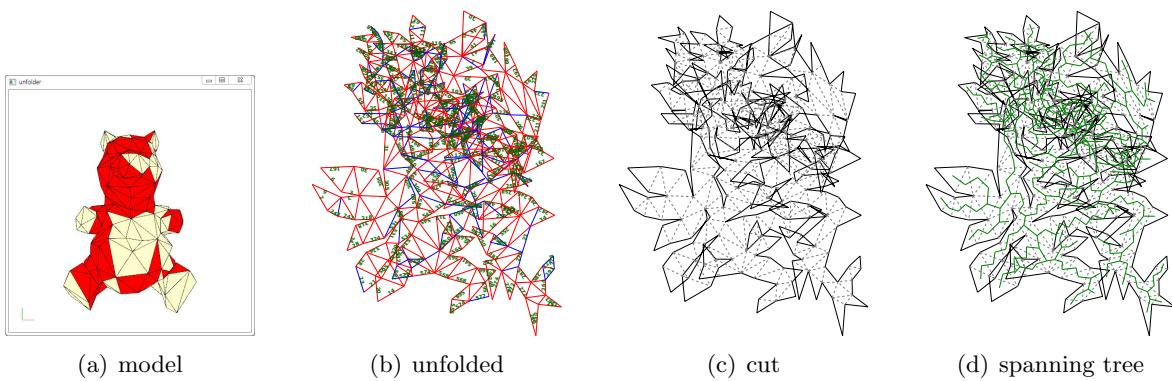


Figure 31: The Steepest Edge Unfold

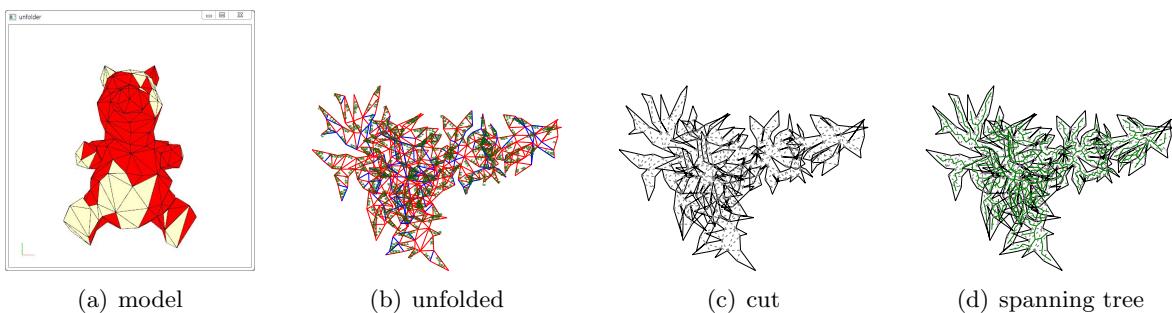


Figure 32: The Flat Edge Unfold

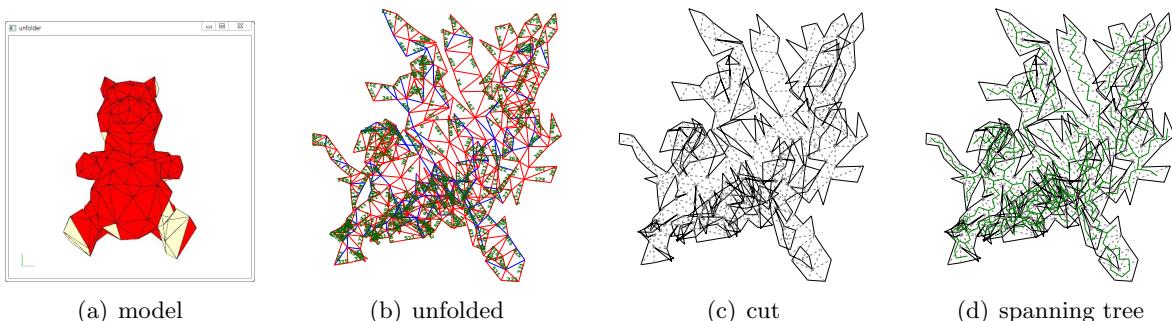


Figure 33: The Greatest Increase Unfold

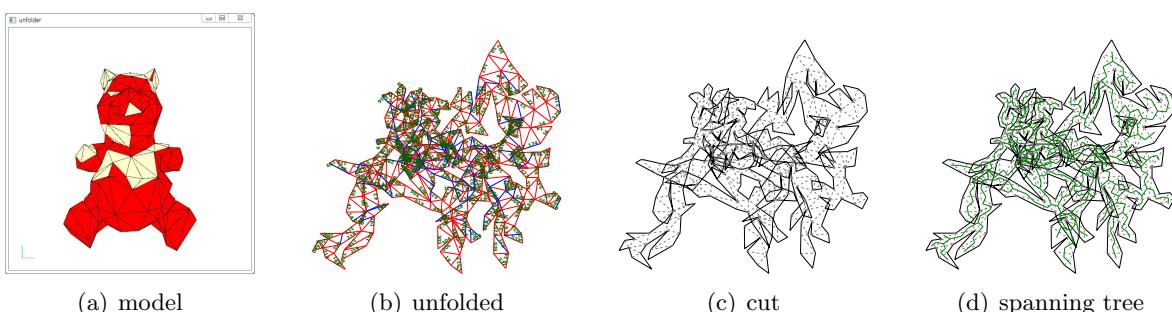


Figure 34: The Rightemost Ascending Edge Unfold

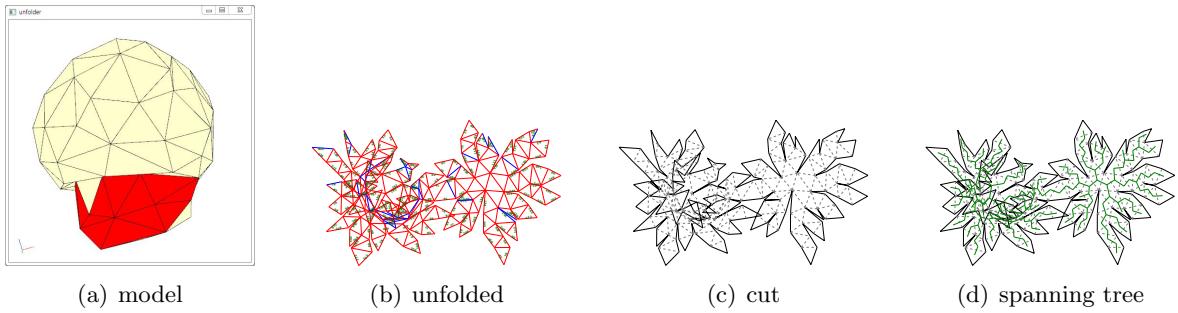


Figure 35: The Steepest Edge Unfold

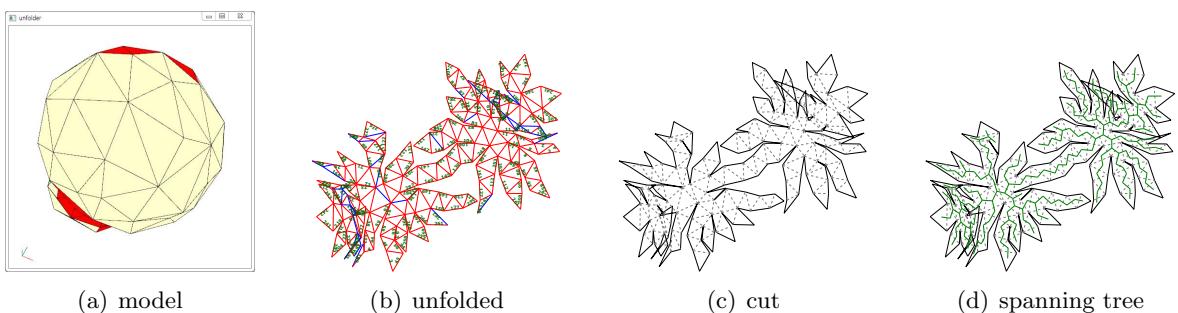


Figure 36: The Flat Edge Unfold

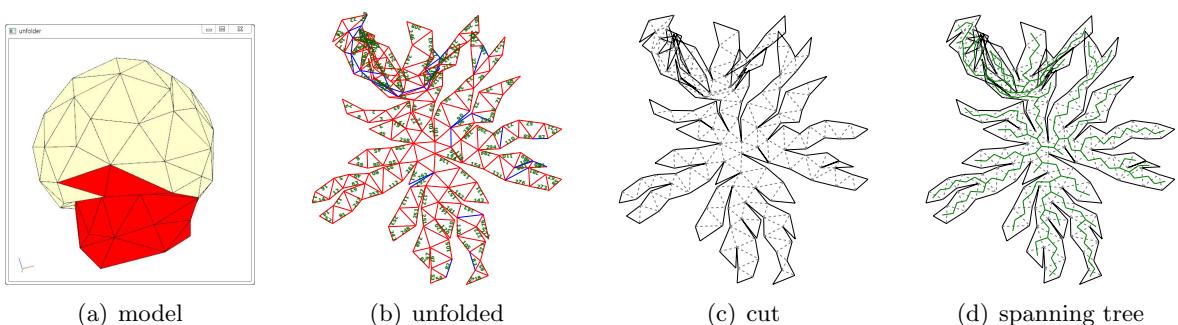


Figure 37: The Greatest Increase Unfold

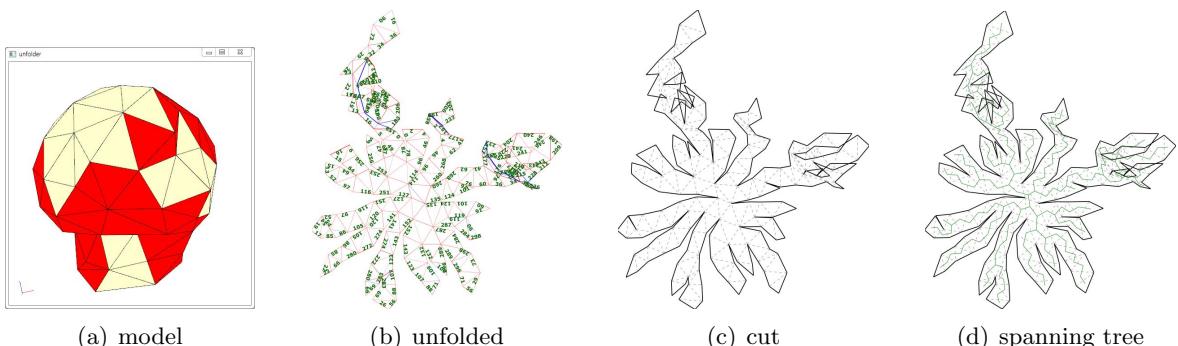


Figure 38: The Rightemost Ascending Edge Unfold

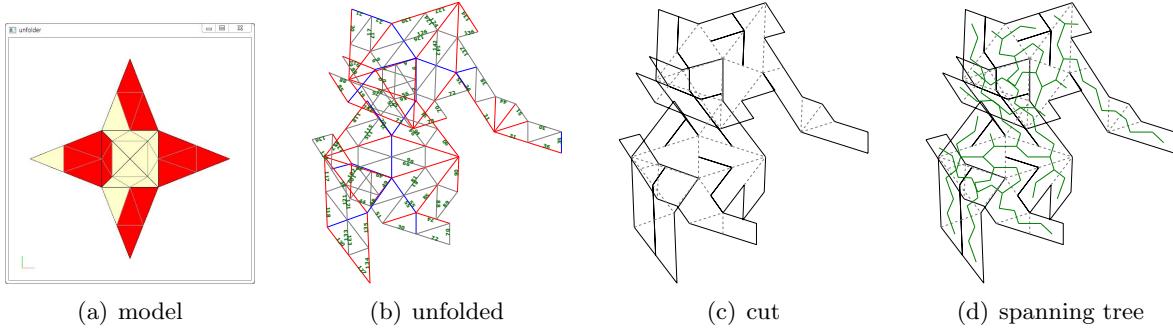


Figure 39: The Steepest Edge Unfold

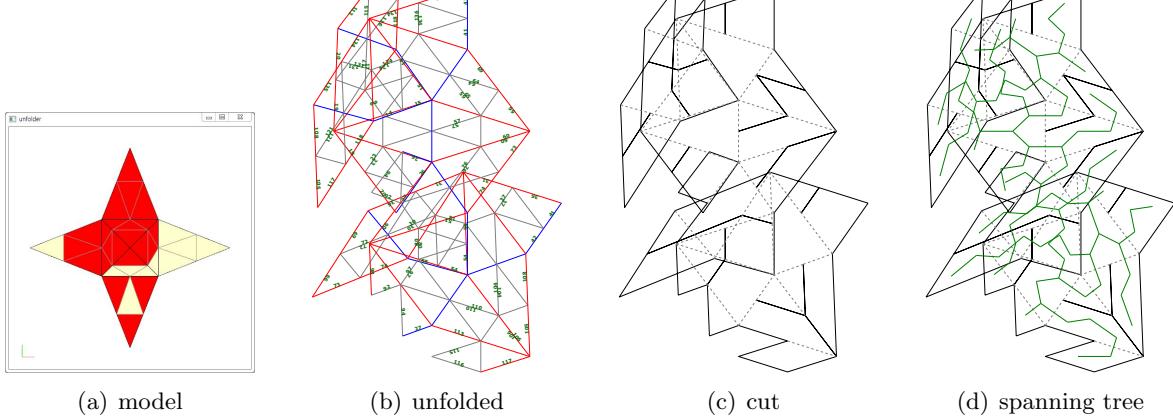


Figure 40: The Flat Edge Unfold

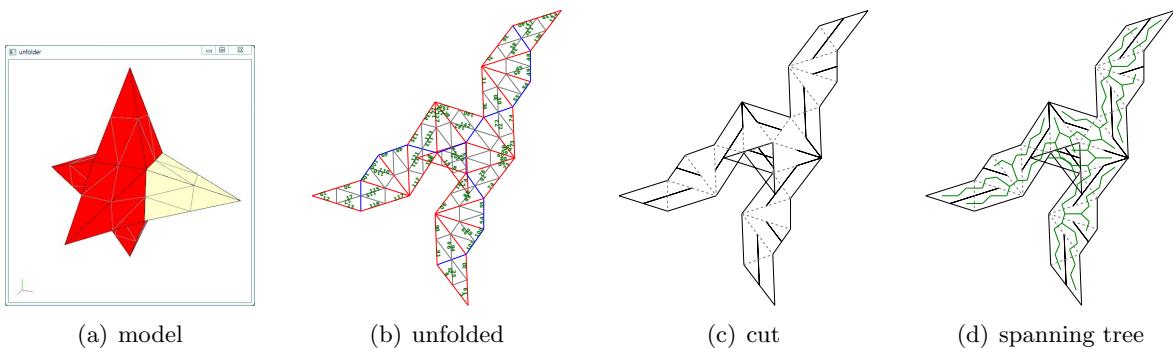


Figure 41: The Greatest Increase Unfold

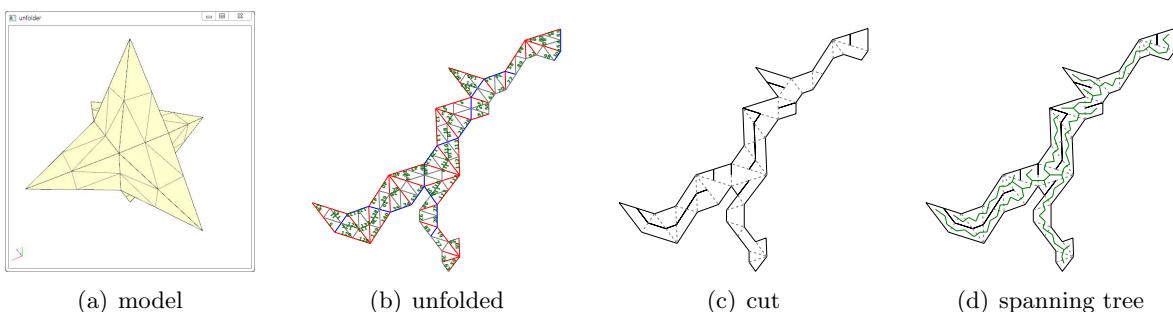


Figure 42: The Rightmost Ascending Edge Unfold

## 4 Foundations

Although four methods do not work well for non-convex model, they successfully generate unfolded polygons for convex models. The program attempts at most 1000 times for non-convex models. As a result, only *the rightmost ascending edge unfold* method can generate a unfolded polygon for *star-4split* model.

## References

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