# Modeling D-Forms

Özgür Gönen, Ergun Akleman and Vinod Srinivasan
Visualization Sciences Program, Department of Architecture,
College of Architecture, Texas A&M University
ozgur@viz.tamu.edu, ergun@viz.tamu.edu, vinod@viz.tamu.edu

### Abstract

In this paper, we present a computational method for modeling D-forms. These D-forms can directly be designed using our software. We unfold designed D-forms using a commercially available software. Unfolded pieces are later cut using a laser cutter. We obtain the physical D-forms by gluing the unfolded paper pieces together. Using this method we can obtain complicated D-forms that cannot be constructed without a computer.

## 1 Introduction

Developable surfaces are particularly interesting for sculptural design. It is possible to find new forms by physically constructing developable surfaces. Recently, very interesting developable sculptures, called D-forms, were invented by the London designer Tony Wills [22] and first introduced by John Sharp to the art and math community [18]. D-forms are created by joining the edges of a pair of sheet metal or paper shapes with the same perimeter [18, 22].

Despite its power to construct unusual shapes easily, there are two problems with physical D-form construction. First, the physical construction is limited to only two pieces. It is hard to figure out the perimeter relationships if we try to use more than two pieces. The second problem with D-form construction is that until we finalize the physical construction of the shape we do not exactly know what kind of shape will be obtained.

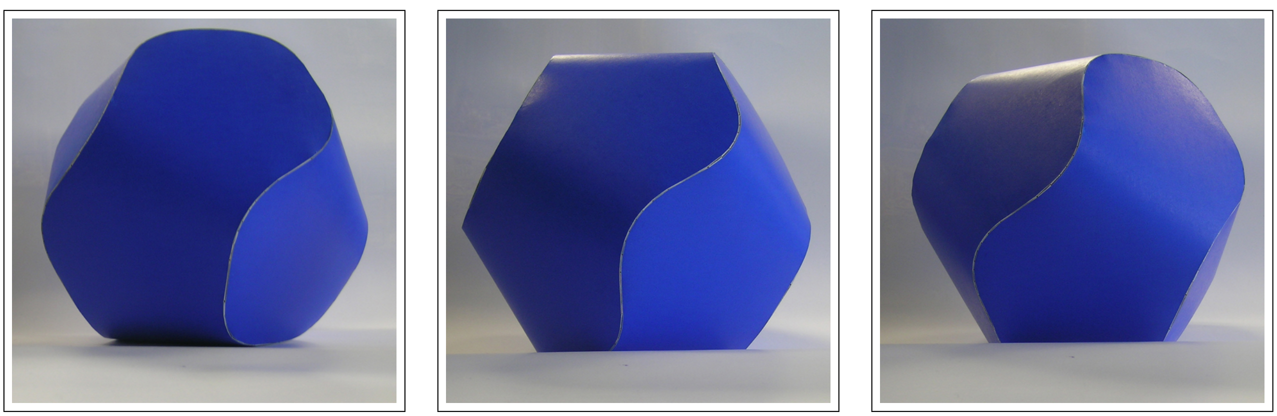

**Figure 1**: *Three views of a D-form constructed using our method starting from a dodecahedron. This shape is designed using our software by Ergun Akleman. This D-form consists of two pieces. The computer designed and unfolded versions of this D-form are shown in Figure 8. Jonathan Penney combined the unfolded pieces to create final physical D-forms.*

In this paper we introduce a computation method that provides an alternative to physical D-form construction. Our implementation allows the user to design D-forms directly in software. Our D-forms can consist of more than two pieces (see Figure 8). Another advantage of our method is that the user can visualize the final shape before physical construction of the shape. Our computer-designed D-forms can be unfolded using Pepakura, a commercially available polygonal unfolding software [21]. Once unfolded, the pieces can be cut using a laser cutter and glued together to create physical D-forms. Using this method we have also created new D-forms that were not known before.

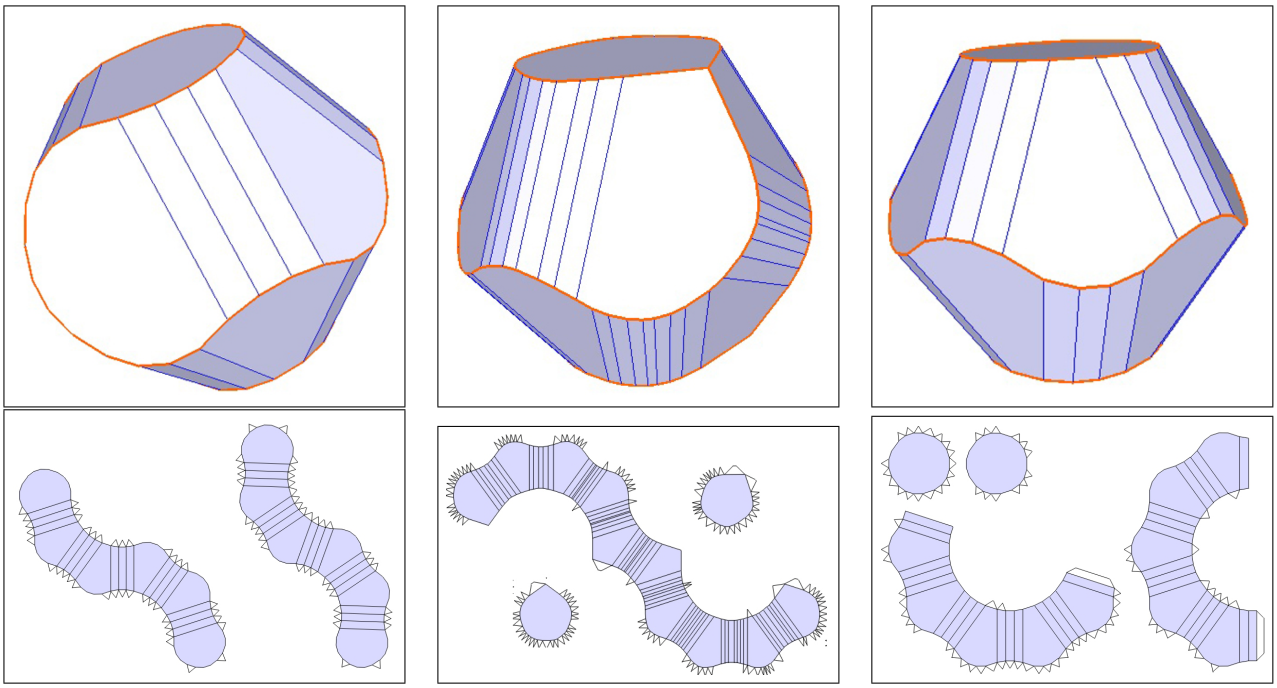


**Figure 2**: *Three D-forms constructed using our method starting from a dodecahedron. The top row shows computer generated D-forms. The second row shows unfolded versions of the same D-forms. Note that each of these D-forms are different. The one on the left consists of only two pieces. The one in the middle consists of three pieces and the one on the right consists of four pieces.*

## 2 Previous Work

In architectural and sculptural modeling, we want eventually to construct the shapes that we have designed. The recently introduced concept of conical meshes [12] provides a framework to model constructible shapes. In this paper we introduce a method to design D-forms with conical meshes using valence-3 planar meshes. Since valence-3 planar meshes always guarantee conical mesh property, the shapes created by using our methods can physically be constructed.

Planar meshes are useful for surface representation. A planar mesh is said to have conical property if and only if all vertices in the polygonal mesh have the property that offsetting all the face planes incident with the vertex by a constant distance leads to planes which intersect again in a common point [12]. This is equivalent to the property that the planes, consistently oriented via the connectivity of the mesh, are tangent to an oriented cone of revolution. The most common planar meshes in computer graphics, triangular and quadrilateral meshes, do not guarantee conical property. Wallner [12] introduced a method for approximating smooth surfaces with valence-4 planar quadrilateral meshes that satisfy conical property. In this paper, we propose a tool to directly model D-forms with conical property by using only valence-3 vertices. Valence-3 vertices are very common in nature. The set operations over randomly oriented valence-3 planar meshes usually result in valence-3 planar meshes. Valence-3 structures are observed in natural formations such as rocks, trees, any type of cracks on planar surfaces and even on crumpled paper. One use of valence-3 planar meshes is to approximate developable surfaces with planar strips.

Developable surfaces are defined as surfaces on which the Gaussian curvature is 0 everywhere [20]. Developable surfaces are useful since they can be made out of sheet metal or paper by rolling a flat sheet of material without stretching it [15]. Most large-scale objects such as airplanes and ships are constructed using un-stretched sheet metal. In ship or airplane design, the problems usually stem from engineering concerns and in engineering design there has been a strong interest in developable surfaces. For instance, modeling packages such as Rhino provide developable surface analysis [15, 16].

Although it is easy to physically construct developable surfaces using sheet metal or paper, it is not that easy to provide computational models to represent developable surfaces. Sun and Fiume developed a technique for constructing developable surfaces [19], but their method is useful only to represent ribbons and is hard to use to represent general developable surfaces. Chu and Sequin introduced developable Bézier patches [7]. Haeberli recently introduced a method to represent a shape with piecewise developable surfaces and implemented it in his Lamina Design Software [11]. The current results seem to be limited but Haeberli's approach has great potential for developable surface design. Mitani and Suzuki introduced a method to approximate any given shape using developable surfaces to create paper models [13]. Because of the approximate nature of their models, there exist gaps between individual pieces and therefore, their method is not suitable for engineering application.

Developable surfaces are frequently used by contemporary architects to design new forms. However, the design and construction of large-scale shapes with developable surfaces requires extensive architectural and civil engineering expertise. Only a few architectural firms such as Gehry Associates have been able to take advantage of the current graphics and modeling technology to construct such revolutionary new forms [9]. Gehry Partners and Schlaich Bergermann and Partners [10] argue that freeform glass structures with planar quadrilateral facets are preferable over structures built from triangular facets or non-planar quads and also show a few simple ways to construct quad meshes with planar faces.

One of the main usages of planar meshes is in developable surfaces, represented by an arrangement of thin planar quadrilaterals in a single row. In particular, D-forms can be approximated using thin planar quadrilaterals with valence-3 vertices. The research community has also been exploring D-forms. For instance, Pottman and Wallner introduced two open questions involving D-forms [14, 8]. Sharp introduced anti-D-forms that are created by joining the holes [17]. Ron Evans invented another related developable form called Plexagons [6]. Paul Bourke has recently constructed computer generated D-forms and plexagons [5, 6].

In this paper, we present a method that allows the user to approximate developable surfaces with valence-3 planar meshes. One important usage of our method is to design a large variety of D-forms [22].

## 3 Methodology

The fundamental idea behind our computational method is to slice a planar mesh with planes. Our method is inspired by traditional sculpture techniques. It is based on a planar truncation operation which simply slices a vertex or an edge by intersecting the mesh with a planar surface. This operation always guarantees planarity and is conceptually similar to ""truncation" or "beveling" operations in shape modeling. However, the classical truncation or beveling operations do not guarantee that the resulting faces will be planar [3].

Our planar truncation operation can work as either vertex, edge, or face truncation. The only difference between these three cases is in how we define the slicing planes. The slicing planes are given by two parameters; a normal vector $n$ that is perpendicular to the plane and a point $p$ that is on the plane. For this paper, edge truncation is the key operation. The default parameters for edge truncation guarantee to provide smoothness with successive iterations. In other words, when planar truncation is applied to an edge consecutively, it can smooth the edge by creating a nice curved developable surface. With default parameters smoothing can result in a quadric profile like Chaitkin's algorithm. For detailed discussion of Chaitkin's algorithm see [3].

Figure 3 shows smoothing of an edge by consecutive application of planar truncation. As can be seen in this figure, with each application of the cut operation, the resulting surface approaches a developable surface. Moreover, the application of a planar truncation operation creates valence-3 vertices as seen in the figure. If we apply the edge-cut operation to four edges of a cube consecutively, we can eventually create a D-form which is similar to the D-form that is created from to ellipsoid as shown in Figure 4.

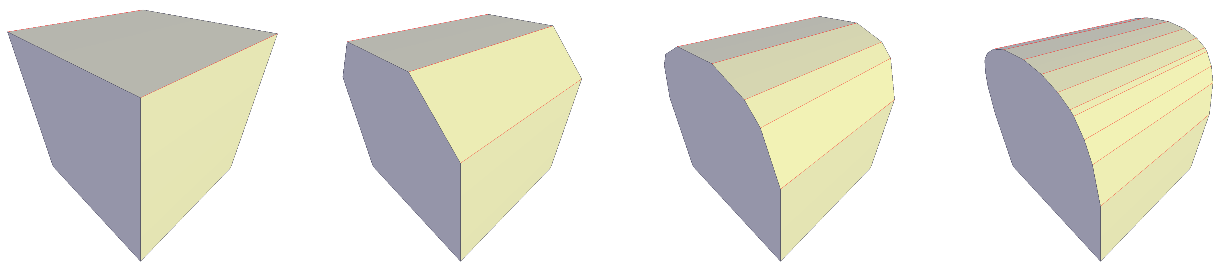

**Figure 3**: *If we apply cut operations to 4 edges of a cube, the shape eventually approaches to a D-form.*

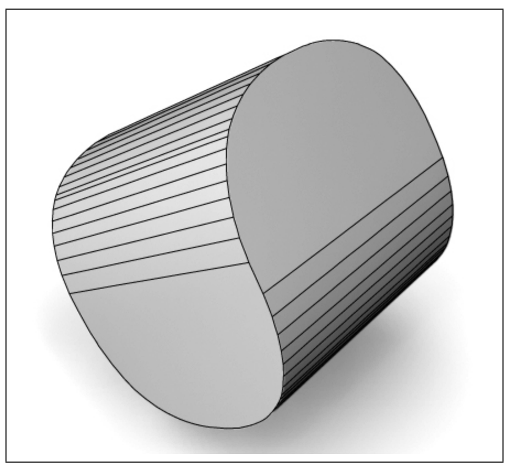

**Figure 4**: *Using the procedure in Figure 3, we can create a D-form that resembles one of the most well-known D-forms. Note that if we unfold this structure, the resulting two pieces will not exactly be ellipsoids.*

Note that since slicing is done with an intersection operation, one planar truncation operation can remove more than one vertex or one edge. In other words, our planar truncation operation slices all the edges intersected by the given slicing plane and gets rid of the portion of the mesh which remains in the positive side of the slicing plane. Therefore, the method works best for converting convex shapes to D-forms. To avoid cutting the whole mesh globally, we also provide a "local planar truncation" operation that traverses all the edges of the mesh starting from the marked element, until the slicing plane was hit. Using local planar truncation it is possible to remove only a part of the shape without touching the rest.

It is also possible to apply the cut operation to multiple vertices, edges or faces. We compute one slicing plane for each selected entity and then apply the cut operation. All slicing planes are computed before any cut operation is performed, since a selected entity could potentially be modified by a cut operation.

## 4 Implementation and Results

We have implemented our planar truncation operation in a topological mesh modeling software, TopMod [1, 2]. We provide three different tools – *Cut by Edge*, *Cut by Vertex*, and *Cut by Face*. Users can adjust the default parameters of the slicing planes.

Figures 1 and 8 show D-forms sculpted out of a dodecahedron by using our planar truncation operation. The designed D-forms are later unfolded by using Pepakura [21]. A screen-shot of the Pepakura user interface is shown in Figure 5. The unfolded pieces can be cut either using a pair of scissors or a laser cutter. Once the pieces are cut, they are glued together to create the final D-form sculpture.

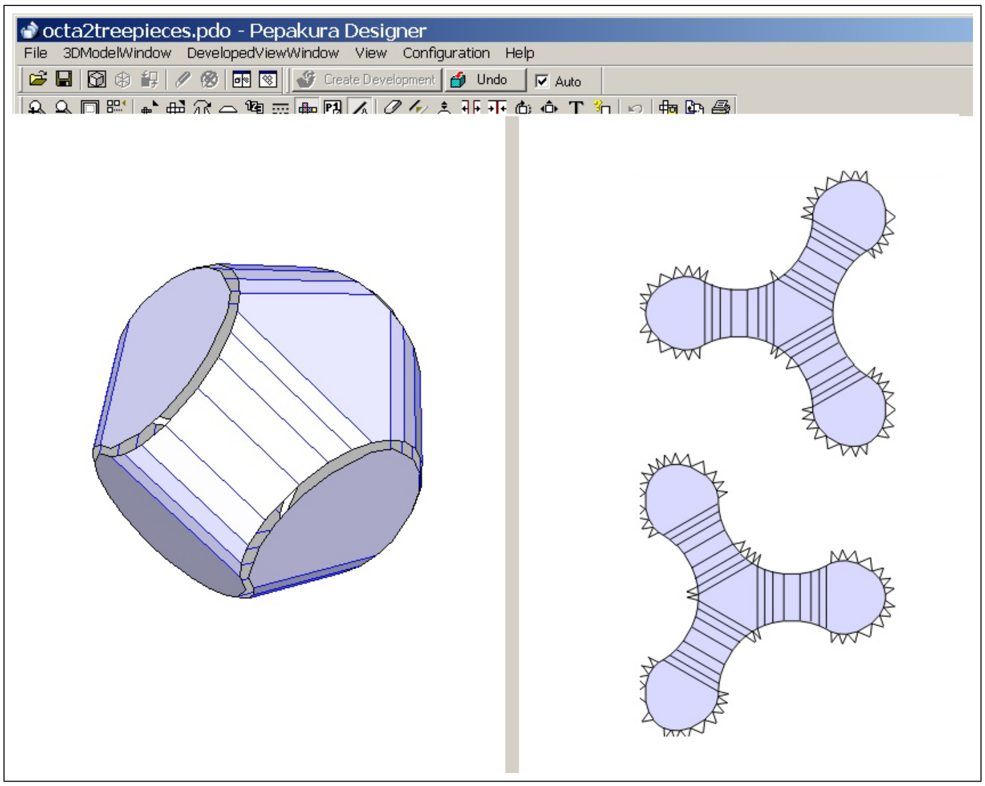


**Figure 5**: *Unfolding a D-form in Pepakura. This D-form is obtained from an octahedron. We first truncate vertices to obtain a truncated octahedron. Then, we creat the D-form with successive edge truncations. Note the* **Y** *shape of unfolded pieces.*

The three piece case in Figure 8 is particularly interesting since the long piece touches itself. This suggests that it may be possible to construct a D-form using only one piece, although we have not been able to find one. The D-form in Figure 5 is also interesting in the sense that both unfolded pieces have a **Y** shape. The Figure 6 shows some other unusual D-forms that consist of more than two pieces. The planar truncation operation can also be used to create interesting patterns that can be built as freeform glass structures as shown in Figure 7.

## 5 Conclusions and Future Work

In this paper, we have presented a computational method for modeling D-forms as conical meshes. Our method provides an alternative to physical D-form construction. Our computer generated D-forms can be

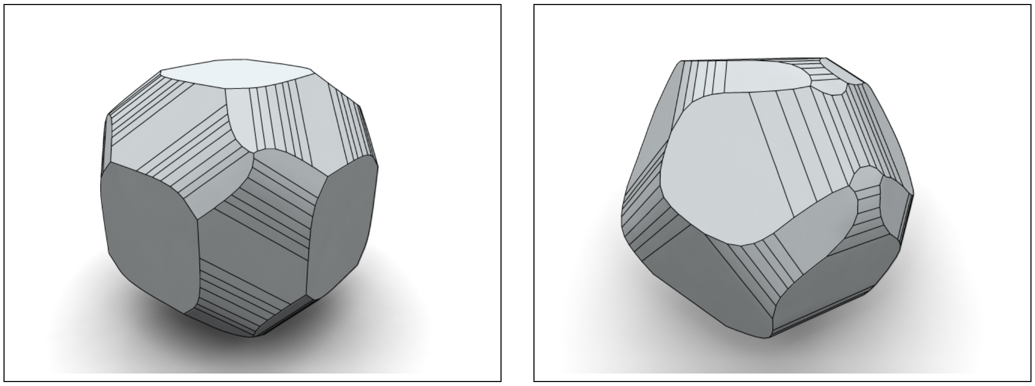

**Figure 6**: *Some unusual D-forms that consist of more than two pieces.*

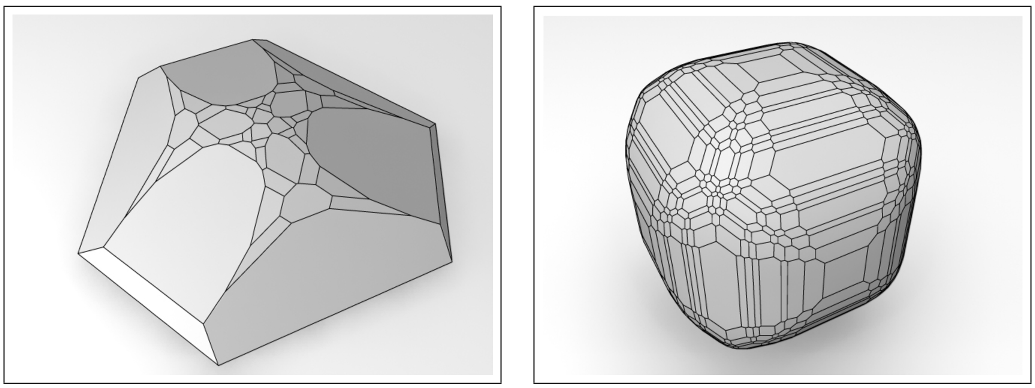

**Figure 7**: *The planar truncation operation can also be used to create interesting patterns that can be built as freeform glass structures*

unfolded using commercially available software and cut using a laser cutter. Physical D-forms can be obtained by putting the unfolded metal or paper pieces together. Using this method it is possible to create complicated D-forms that cannot be constructed without a computer. One of the major advantages of our D-forms is that they are created as conical meshes and can therefore be constructed at larger scales even from thick and planar materials like glass or sheet metal.

Currently our method works only for convex shapes. We are look at ways of generalizing our method to non-convex shapes and shapes with saddle points.

## References


[1] TopMod website: http://www-viz.tamu.edu/faculty/ergun/research/topology/

[2] E. Akleman, J. Chen, and V. Srinivasan. A minimal and complete set of operators for the development of robust manifold mesh modelers. *Graphical Models Journal, Special issue on International Conference on Shape Modeling and Applications 2002*, 65(2):286–304, 2003.

[3] E. Akleman, P. Edmundson and O. Ozener, A Vertex Truncation Subdivision Scheme to Create Intriguing Polyhedra, Proceedings of Bridges 2004, Winfield, Kansas, August 2004.

[4] Kenneth A. Brakke, Surface Evolver, http://www.susqu.edu/facstaff/b/brakke/evolver/evolver.html

[5] Paul Bourke, D-Forms, http://astronomy.swin.edu.au/ pbourke/surfaces/dform/

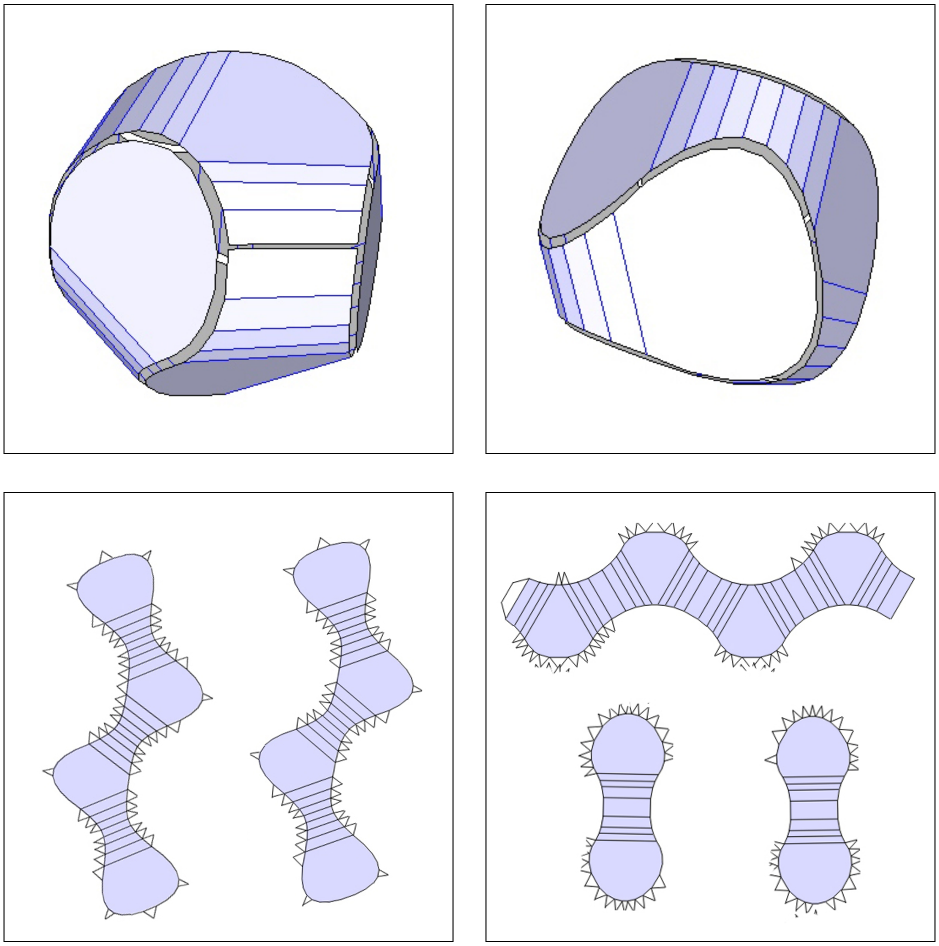



**Figure 8**: *Two D-forms constructed using our method starting from an octahedron. The top row shows computer generated D-forms. The second row shows unfolded versions of the same D-forms. Note that these D-forms are different. The one on the left consists of only two pieces and the one on the right consists of three pieces and the piece on the top is a closed loop.*


[6] Ron Evans, Plexons created by Paul Bourke, http://astronomy.swin.edu.au/ pbourke/geometry/plexagon/

[7] Chu, C. H., and Sequin, C. 2002. Developable Bezier patches: properties and design. Computer-Aided Design 34, 511-528.

[8] Erik D. Demaine and Joseph O'Rourke, "Open Problems from CCCG 2002," in Proceedings of the 15th Canadian Conference on Computational Geometry (CCCG 2003), pp. 178-181, Halifax, Nova Scotia, Canada, August 11-13, 2003.

[9] Frank Gehry, http://www.gehrytechnologies.com/

[10] Glymph, J., Shelden, D., Ceccato, C., Mussel, J., and Schober, H. 2002. A parametric strategy for freeform glass structures using quadrilateral planar facets. In Acadia 2002, ACM, 303-321.

[11] Paul Haeberli, http://laminadesign.com/index.html

[12] Y. Liu, H. Pottmann, J. Wallner and Y. Yang Tand W. Wang. Geometric Modeling with Conical Meshes and Developable Surfaces. ACM SIGGRAPH, pp. 681 - 689, 2006.

[13] Jun Mitani and Hiromasa Suzuki, “Making Papercraft Toys From Meshes Using Strip-Based Approximate Unfolding. ACM Trans. Graph. 23(3). pp. 259-263, 2004.

[14] Helmut Pottmann and Johannes Wallner, “Computational Line Geometry”, Springer-Verlag, 4, p. 418, 2001.

[15] http://www.rhino3.de/design/modeling/developable/index.shtml

[16] http://www.rhino3.de/design/modeling/developable/marine_design.shtml

[17] John Sharp, D-forms: Surprising new 3D forms from flat curved shapes,Tarquin 2005.

[18] John Sharp, Presentation of ”D-forms and Developable Surfaces”, in Proceedings of Bridges 2005, pp. 121-128, Banff, Canada, 2005.

[19] Meng Sun and Eugene Fiume, A technique for constructing developable surfaces, Proceedings of the conference on Graphics interface ’96, Toronto, Ontario, Canada, pp. 176 - 185, 1996.

[20] Eric W. Weisstein. ”Developable Surface.” From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/DevelopableSurface.html.

[21] ”Pepakura Designer” http://www.tamasoft.co.jp/pepakura-en/

[22] Tony Wills, D-forms, in Proceedings of Bridges 2006, London, 2006.