Construction with Physical version of Quad-Edge Data Structures

Ergun Akleman\textsuperscript{a}, Shenyao Ke\textsuperscript{a}, You Wu\textsuperscript{a}, Negar Kalantar\textsuperscript{a}, AliReza Borhani\textsuperscript{a}, Jianer Chen\textsuperscript{a}

\textsuperscript{a}Texas A\&M University

Abstract

In this work, we demonstrate that quad-edge mesh data structure can efficiently be used to construct large shapes. We have developed a system to unfold any polygonal mesh based on quad-edge structure. Using our system, any shape can be constructed by using laser-cut developable panels. This approach is particularly suitable to construct complicated sculptural and architectural shapes from anisotropic materials that can only be bent in one direction.

Keywords: Shape Modeling, Physical Construction.

1. Introduction

In this paper, we present our current progress for realization of complex 3D surfaces and structures that are assembled from 2D planar shapes. Our goal is to construct physical structures that can be raised and formed in 3-space using minimal amount of planar materials. Current construction methods assume the use of isotropic materials such as paper or thin metal that can provide forming any developable shape. On the other hand, not all materials are isotropic and there is now a strong interest in construction with anisotropic active materials that can change their shapes when they are activated [1, 2]. These materials have now been applied to medical, aerospace, and automotive applications in the engineering realm [3].

There is one significant restriction with some of these active materials such as smart laminates made from shape memory alloys (SMAs). They can not provide all possible developable structures since they are anisotropic; i.e. they can bend only one (or few directions) direction and almost rigid in the perpendicular direction (or other directions) [4]. In other words, they can provide only generalized cylinders and cannot provide general developable structures such as cones. Therefore, the panels we can use can have only a single bending direction. Moreover, since such active materials are much more expensive than paper, it is also important to reduce waste materials. In conclusion, there is now a need for a general framework that can facilitate economic construction with such anisotropic planar materials. In this paper we present such a framework for modeling the geometry of planar panels that can self-fold into a desired 3D shape and we demonstrate that these elements can be folded to create desired shapes.

In our framework, we start with a polygonal mesh approximation of the desired 3D shape. Then, we unfold it into multiple planar panels, which are simply 2D polygons. By physically constructing the original shapes, we demonstrate the feasibility of this approach (see Figures 1 and 2), which is well-suited for the consideration of deforming materials in which sharp folds or creases are not physically feasible. Our work in this paper can be summarized as follows:

(1) We have developed a new multi-panel unfolding approach to unfold any given polygonal mesh surface into a set of planar pieces that can efficiently be packed since they consist of a series of trapezoids without even using simple rectangular packing algorithms [5, 6]; (2) We have developed and implemented a method to construct physical shapes that approximate original polygonal meshes by connecting and folding the planar pieces; (3) We have constructed a few examples of physical shapes using planar pieces to demonstrate the feasibility of the approach.

2. Motivation for Multi-Panel Unfolding and Construction

With the design and construction of more and more unusually shaped buildings, the computer graphics community has started to explore new methods to reduce the cost of the physical construction for large shapes. Most
of currently suggested methods either focus on reduction of the number of differently shaped components to reduce fabrication cost using mass production [7, 8, 9] or identify shapes that can be built using components that are already mass produced [10, 11, 12].

Fortunately, mass production is not a strong requirement for developable components [13]. Developable components can be fabricated as planar panels, which is economical even if each component is different. Such panels can be individually manufactured in a fairly inexpensive way by cutting large sheets of thin metals or thin paper using laser-cutters, which are now widely available. Then, they can be used for physical construction of large shapes to be used as developable components.

One of the biggest expenses for construction of large shapes comes from packing large number of planar panels. If the panels have irregular shapes, it will be harder to pack them efficiently. If expensive anisotropic materials, which have a preferred bending direction, need to be used, packing become a bigger issue since we cannot freely rotate shapes for more efficient packing. This restriction can cause a significant amount waste material. In this paper, we present a new multi-panel unfolding technique [14, 15] that provides efficient packing with anisotropic materials. Our technique turns each edge of initial planar mesh into a flat shaped panel, which we call quad-edge panels. The quad-edge panels only need to be bent along parallel fold lines – called scores in laser cutting terminology – forming generalized cylinders in the construction of large shapes. This property of quad-edge panels makes them suitable to be printed using anisotropic materials, which already have a preferred bending direction. The quad-edge panels can be packed efficiently along their preferred bending direction that is given by a set of parallel lines. Moreover, a single quad-edge panel can fit inside of a single rectangle without significant wasted area. Therefore, we can obtain efficient packing even with anisotropic materials.

Another important problem is handling and assembling the large number planar panels. This problem is like putting pieces of a large puzzle together. However, unlike puzzles we do not want the construction process to be challenging to the construction people. Instead, we want to simplify the construction process in such a way that the components can be easily assembled by novices given a minimum set of instructions. Quad-edge data structure is particularly suitable for construction, since it provides a well-defined assembly process. The con
Figure 2: A genus-3 structure constructed with our method. Since quad-edge panels consist of a series of trapezoids and parallelograms and their sizes are not significantly different, they can be packed reasonably well even without using sophisticated rectangular packing algorithm. This particular construction uses approximately 200 quad-edge panels. The construction took five hours to assemble for two people. We used the same type of paper and fasteners used to construct Bunny (3 sheets and 2 boxes of fasteners).

Figure 3: Details of laser cut panels that have not yet separated from underlying substrate showing all scores (i.e. fold lines) are parallel to each other forming a series of trapezoids. These images also shows vertex and face identification numbers printed during laser cutting process.

Construction workers can simply assemble quad-edge panels by checking identical corners, which are given by a pair of face and vertex identification numbers.

Multi-panel unfolding also allows us to circumvent the theoretical problems associated with single panel unfolding [16, 17, 18]. In fact, even intuitively simple looking problems in single panel unfolding are still open. For instance, it has not yet been proven whether every convex polyhedron can be cut along its edges and unfolded flat to a single, non-overlapping simple polygon (See problem 9 at [16]).

As an additional property of our multi-panel unfolding, it is possible to construct a single flattened surface without worrying about overlaps. When only quad-edge panels that correspond to the edges of the same face are assembled together, they continue to be planar by forming a single flattened surface – since non-zero curvature exists only in the regions that correspond to the vertices of original planar polygonal mesh. Self-intersection with such single planar unfolding does not create a problem since planar components are not truly 2D and they can go top of each other in physical space in overlapping regions. This property is important, in particular, for automatic formation of 3D structures. To obtain such a single flattened surface is not difficult. All we need to do is to form a spanning tree over the graph.
that represents dual mesh, where each face becomes a vertex. If construction people only connect the quad-edge panels given by a selected spanning tree, the whole system of quad-edge panels can be unfolded to a plane. Our quad-edge panels can be pre-assembled as a single overlapping planar structure and to be transported to remote places. If each quad-edge panel is cut using anisotropic shape memory materials that can be bend only one direction, 3D structure can remotely be formed by simply activating shape memory materials [15].

![Cube](a) Octahedron (b) Toroid

Figure 4: Examples of physical realization of graphs embedded on surfaces that represent cube, octahedron and toroid respectively. In this case, each paper strip is a planar panel entity that corresponds to an edge of graph. Fasteners that connect paper-strips are physical connectors that correspond to vertices of the graph.

3. Polygonal Mesh Data Structures for Multi-Panel Unfolding and Construction

Multi-panel unfolding and construction require to have a 2-manifold mesh representation that provides a well-defined cellular decomposition of any given orientable surface\(^1\). All orientable manifold-mesh data structures are based on the graph rotation systems (GRS), which is a powerful tool for guaranteeing topological consistency [21]. A graph embedded in an orientable surface corresponds to a rotation system, namely, the one in which the rotation at each vertex is consistent with the cyclic order of the neighboring vertices in the embedding [21, 22]. GRS has explicitly been used as a physical mesh data structure as shown in Figure 5 to construct large shapes [14]. GRS has also been implicitly used in computer graphics for representing and manipulating orientable (and non-orientable) 2-manifold surfaces in the guise of various data structures, such as half-edges [23], quad-edges [24], and winged-edges [25]. In this section, we demonstrate that these data structures, in particular quad-edge data structures, are also useful to construct physical structures. Moreover, some of the existing mesh data structures in computer graphics are recently re-discovered by mathematicians and toy designers and used as geometric toys [26]. In the rest of this section, we provide such examples of shape construction with existing mesh data structures. In all these cases, physical connectors correspond to links that connect the entities to each other and entities correspond to planar panels [26].

![Vertex components](a) Fasteners (b) Examples of Archimedean polyhedron.

Figure 5: Examples of construction with Graph Rotation Systems. Process photographs in (d) are courtesy of Gabriel Esquivel. In this case, each vertex component is a planar panel entity that corresponds to an edge of graph. Fasteners are physical connectors that correspond to edges that link vertices to each other.

3.1. Construction with Graph Rotation Systems

One of the implications of GRS is that these physical mesh data structures can be considered instances of thickened graphs embedded on orientable 2-manifold surfaces. Figure 4 shows paper strip sculptures that are used to represent graphs embedded on surfaces [27]. The graph \( G \) can be “thickened” in the surface \( S \) by enlarging each vertex of \( G \) to a small polygonal disk called a vertex-band and widening each edge to a narrow rectangular edge-band that joins the vertex-bands at its endpoints [21, 27]. The union of the vertex-bands and the

---

\(^1\)In this section, we only cover multi-panel surface construction methods that do not alter the input polyhedron for consistency of the presentation. We, therefore, do not include works that allows altering underlying mesh representations such as [19]. We neither include methods that can provide shapes beyond surfaces, such as ZomeTool [20] that can provide representations of 4D regular polytops.
edge-bands is topologically a surface $T$ with boundary, called a *thickening of the graph $G$*, and also called a *band decomposition*.

Figure 5 shows examples of construction with GRS. Figure 5(a) is an example of a laser cut vertex component that provides rotation around a vertex, Figure 5(a) also shows elements of a vertex component, which includes a convex shape that corresponds to 2D thickening of the vertex and a set of quadrilateral shaped "flaps" that correspond to 2D thickening of the edges. The flaps are connector parts that are connected by using fasteners that are shown in Figure 5(b). Figure 5(c) shows examples of Archimedean polyhedra constructed using vertex components while Figure 5(d) provides a series of photographs taken during the process of assembling a more complicated structure using vertex components with fasteners.

![Figure 5](image)

**Figure 5:** Examples of models that correspond to half-edge mesh data structures that represent (a) a shape made from five cubes, (b) a dodecahedron, and (c) an octahedron. In this case, slits are physical connectors that correspond to half-edges that link faces to each other.

3.2. Construction with Half-Edges

In half-edge data structures, each edge consists of two half-edges corresponding to two sides of the edge. Faces are defined by cyclically ordered sets of half-edges [23]. They are linked with each other using these the half-edges of the edges on the face boundary. Half-edges, therefore, play the connector role in physical versions. Although a variety of physical versions of half-edge structures are possible, all existing physical versions of half-edge data structures that we know are some type of slitted structures, in which slitted parts play the role of connector (i.e. half-edge) [26].

Slitted structures were first introduced by George Hart as slide-together structures for platonic solids [28]. Jace Miller independently discovered a general version by showing these slitted polygons can be used to construct any physical shape as interlocking-slitted structure [29] (See Figure 6). Figure 7 shows two types of recently discovered “slitted” geometric toys that also correspond to half-edge data structures. “Space chips”, introduced by Architect Dick Esterle in 2011, are rigid circular structures with slits [30]. “Itsphun” is another slitted geometric toy introduced in 2015 Bridges Art and Math conference [31]. In both geometric toy cases, slits again play the role of half-edges that allow to connect edges.

We observe that one main issue with slits as connectors is that the individual panels in constructed shapes using “Space chips” or “Itsphun” can twist in tandem. This is appropriate property for toys since it can help to separate pieces easily. On the other hand, for more permanent constructions slits need to be replaced by stronger connectors.

3.3. Construction with Quad-Edges

Since we focus on quad-edge based physical structures in this paper, we will provide a detailed description to relate theoretical quad-edge data structure to the physical structures. Note that the concept of quad-edge emerges from the fact that the edges are the only standard elements in a 2-manifold mesh consisting of two edge-ends and two half-edges [24]. Any edge can therefore be easily thickened to a rectangular *edge-band* as discussed earlier and shown in Figure 4.

Quad-edge entities are usually depicted as crosses as shown in Figure 8(a) [24]. We encapsulated the cross shape inside of a square to show that each quad-edge entity is actually a quadrilateral (as the name quad edge suggests). This additional depiction shows that the corners of the quadrilaterals can be considered as links. In quad-edge data structure, quad-edge entities that are linked together describe faces and vertices of 2-manifold meshes as shown in Figure 8(b). Since these are topological entities, it is also possible to stretch...
link regions to depict a representation of tetrahedron as shown in Figure 8(c).

As it can be seen in examples in Figures 8(b) and (c), closed cycles of directed links form boundaries for faces and vertices. The face and vertex regions are shown as two white squares and six yet-uncompleted cycles in Figure 8(b). Since there are uncompleted cycles, Figure 8(b) is not a valid 2-manifold mesh representation. Figure 8(c) shows a valid representation that describe mesh topology of a tetrahedron. In this case, the face and vertex regions are depicted as white circular regions. Note that in Figure 8(c), there is an additional outside cycle that corresponds to one of the triangular faces of a tetrahedron. Also note that these regions of faces and vertices are always organized in a checkerboard pattern; i.e. the neighbors of any given vertex cycle are face cycles and vice versa. This is a well-known property that directly results in thickening the graph embedded on a surface [15]. In this paper, we exploit this checkerboard property to obtain panels that can only be bent in one direction.

Based on this discussion, we can classify any multi-panel construction as quad-edge based, if the panels and construction satisfy the following three conditions:

1. Panels must have exactly four corner connections;
2. Only two panels must be connected by each connection; and
3. Every connection must be used.

Figure 9 shows a multi-panel construction toy that corresponds to quad-edge data structure, which was discovered by toy designers in Japan. These toys are marketed under the name of Flexeez in USA and Wammy in Japan and UK. Figure 9 also shows examples of structures that can be constructed by Flexeez. As shown in the figure, Flexeez demonstrates that using a quad-edge structure enables the ability to construct a wide variety of shapes using a single template building block. This comes from the fact that the edges are the only standard elements in a 2-manifold mesh. If we use vertex or face, we automatically need more variety of shapes to include all possible valences for vertices and all possible sides for faces.

David Reimann independently discovered another physical version of quad-edge data structures in 2015 by demonstrating that a wide variety of Platonic and Archimedean polyhedra can be constructed using only squares cut from anisotropic materials [32] (See Fig-
ure 10). To obtain these structures, underlying polyhedral edges are replaced by squares made from laser-cut paper-backed veneer (walnut or bamboo) and connected at their corners with brass split-pin fasteners. The faces and vertices of the underlying polyhedron are transformed into open negative space. Reimann observed that half of the open spaces must be planar. Since he used an anisotropic material, namely veneer, he needed to carefully put them together to create a consistent structure. This is an expected configuration based on the recent theoretical work in [15]. Using the face defect [27], it is possible to show that the open spaces can be classified as non-zero and zero curvature regions. Open spaces that correspond to original planar faces must—again expectedly—always have zero-curvature based on properties of underlying mesh [14].

Reimann’s work provided the basic inspiration for our approach. It was immediately clear that his approach is not restricted to Platonic and Archimedean polyhedra and can be generalized to any planar polygonal based on [14, 15]. In this paper, we present a method to unfold edges into quad-edge panels.

Figure 10: David Reimann’s expanded models of the Platonic solids after a nonplanar expansion of the edges by replacement with squares. He constructed these models using laser-cut paper-backed bamboo veneer squares that are connected using split-pin fasteners. Solid models of the base polyhedra are shown below the expanded models (Courtesy of David Reimann).

As far as we know, there is no physical counterpart of winged-edge data structure, which is one of the oldest formalized data structures that supports manifold surfaces [25, 33].

3.4. Comparison

It is possible to unfold any polygonal mesh into laser cut panels to construct large shapes using any one of these structures. To compare these physical structures, we have also developed a mesh unfolding system that can produce planar panels for each one of these physical data structures. We have also tried winged-edge representation. We expected because of its wings, it can provide a strong structure. However, as soon as we build one, we realized that it is hard to put together. Another problem is that winged-edge panels do not pack well because of its wings (see Figure 11).

![Figure 11: Examples of models that corresponds to winged-edge mesh data structures. (a) shows flat and bent single panels and (b) shows a shape constructed by using such winged-edge panels and its detail.](image)

Among all these structures, physical versions of quad-edge data structure [24] seems to be most useful for large scale construction using anisotropic materials. The structures shown in Figures 2 and 12 are constructed using unfolded quad-edge panels created using our system. Based on our analysis, construction with quad-edges has three advantages over the others: (1) The quad-edge panels pack densely without leaving much unused material; (2) Every connection is made by only one other panel; which makes connection thickness uniform, and (3) They can be built using anisotropic materials to provide extra stability. The only problem using quad-edge panels is that the number of components is larger than vertex-components in [14, 15]. This is because the number of quad-edge panels equals to the number of edges in the original mesh, which is approximately the sum of the number of vertices and faces for low genus shapes based on Euler characteristics. This makes construction longer since the number of connections that must be done during construction increases. However, we think this is a fair price to pay given this method’s advantages over the others.
4. Theoretical Framework for Construction with Quad-Edges

Our initial goal was to identify a basic mathematical framework for converting any given polygonal mesh surface into a quad-edge physical structure that is easy to construct. The concept of the “band decomposition” from Topological Graph Theory provides such a basic mathematical framework that helped us to develop a simple quad-edge unfolding method. Band decomposition is obtained by 2D-thickening the graph within the surface and is always contractible to the original graph. In a 2D-thickening, each vertex thickens to a polygon (or a disk) and each edge thickens to a band (See Figure 13). Thus, each polygon corresponds to a vertex and each band corresponds to an edge that connects vertex regions. Note that each band is a quadrilateral with curved boundaries. It consists of two types of boundaries: (1) Face-sides that correspond to half-edges and (2) vertex-sides that correspond to edge-ends. We will use this information later.

In the quad-edge method, we convert edges of graphs to bands, which correspond to quad-edges. Using this approach, we can unfold any given polygonal mesh into a set of planar panels, each coming from an edge of the planar polygonal mesh. We then assemble the panels by connecting the corresponding quad-edges to build large structures. In other words, the quad-edge unfolding method simply converts each edge of a graph that is embedded on a surface into a planar panel. By connecting and folding these panels one can construct a close approximation of the desired surface.

With multi-panel unfolding, our goal is to automatically create quad-edge panels that can easily be assembled from any given planar manifold mesh. Each quad-edge panel, which we also call an edge component, becomes a quad shaped polygon with two straight and two curved boundaries that corresponds to the cyclic permutation of the edge-ends and half-edges incident on edge e (See Figure 14(a)). As shown in this figure, unlike general embedding, face-sides can be guaranteed to be straight line since original faces are planar. On the other hand, vertex-sides may not necessarily be straight.

Each quad-edge includes three types of information as shown in Figure 14(b): (1) Four regions of physical links, (2) Face identification numbers, and (3) Physical links. These links and identifiers are used to physically link the panels together to create the final structure.
links with holes to connect (or link) a quad-edge panel to a corresponding quad-edge panel. Each region has at least two holes to create a rigid connection or link. (2) Two face identification numbers that are written close to half-edge boundaries and (3) two vertex identification numbers that are written close to edge-end boundaries. The physical links are used to assemble the final shape by connecting quad-edge panels with each other. For connecting two quad-edge panels that share a corner, we simply join them in their corresponding physical link regions. These can be done in many different ways such as by gluing flaps or by connecting with fasteners as shown in Figure 5(b). To find corresponding physical link regions, we use the vertex and face identification number pairs at two sides of physical link region. Two corresponding physical link regions will have the same identification number pairs. We engrave face and vertex identification numbers onto each panel using the laser cutter. Note that when we print these two identification numbers, we actually define a collection of rotations, one for each corner. This is formally called a pure rotation system of a graph.

Assembly of these panels to construct large structures is based on the fundamental Heffer-Editions theorem of GRS [34, 35]. The theorem asserts that there is a bijective correspondence between the set of pure rotation systems of a graph and the set of equivalence classes of embeddings of the graph in orientable surfaces. As a direct consequence of the theorem, to assemble the structure, the builder must simply attach the corresponding edge-ends of vertex components. Once all the components are attached to each other, the whole structure will correctly be assembled.

5. Rectangles vs. Parallelograms

Our panels consist of a series of parallelograms or trapezoids as shown in Figure 14. This is useful to preserve the local Gaussian curvature characteristics of the original surface, resulting in a close approximation of the shape of the surface. Gaussian curvature is an extremely useful measure for shape and structure of surfaces since it relates topology and geometry of the surface through the Gauss-Bonnet Theorem, which implies that for a manifold mesh, the total Gaussian curvature must be equal to $4\pi$ times the Euler characteristic of the corresponding surface [36, 37]. For instance, for a genus-0 manifold mesh, this sum must be equal to $4\pi$, and for genus-1 surfaces, the Euler characteristic is zero; therefore, this sum must be equal to 0.

Since the structure is composed of only developable panels, Gaussian curvature is zero everywhere on the solid parts. The Gaussian curvature happens only in open spaces that are determined uniquely. If we can correctly form Gaussian curvature of open spaces, the structures will always be raised and formed in 3-space, closely approximating the overall shape of the initial surface. The resulting open spaces in sculptures represent faces and vertices of the original planar mesh. We, therefore, can classify open spaces in two categories.

- (1) Face-holes: These are the open-spaces whose boundary are face-cycles.
- (2) Vertex-holes: These are open space whose boundary are vertex-cycles.

Both of these open spaces, i.e. face-holes and vertex-holes, can be considered faces, in which Gaussian curvature can be defined using their boundary angles assuming that the boundary has zero curvature. Discrete Gaussian curvature for such an open space is known as a face defect [37, 27], and is defined as

$$\Phi = \sum_{j=0}^{n} \phi_j - (n-2)\pi$$

(5.1)

where $\phi_j$ is the angle at corner $j$ of a hole (i.e. open space) when corner is flattaned and $n$ is the number of the sides of the hole (i.e. open space). Since we convert planar meshes, face-defect of face-holes must be zero. On the other hand, face-defect of vertex-holes must be the same as the Gaussian curvature of the vertices. In planar polygonal meshes, for curvature we can use vertex defect, which is the discrete version of Gaussian curvature. For every vertex of piecewise planar meshes [38, 39, 40], vertex defect is defined as

$$\Theta = 2\pi - \sum_{i=0}^{n-1} \theta_i$$

(5.2)

where $\theta_i$ is the angle at corner $i$ of the vertex and $n$ is the valence of the vertex.

To make Gaussian curvature of every vertex-hole $\Phi$ equal to $\Theta$ of its corresponding vertex in the original mesh is conceptually straightforward. Let $\phi_i$ and $\theta_i$ be two corresponding corners of a vertex-hole in the sculpture and its corresponding vertex in the original mesh. If we choose $\phi_i = 2\pi - \theta_i$, then $\Phi = \Theta$ for such corresponding vertex-hole & vertex pair. Unfortunately, this is not always possible when we use rectangular panels since opposite angles are interconnected as shown in Figure 15. In fact, David Reimann’s construction with square(rectangular panels can cause negative curvature
in some of the open spaces corresponding to vertices in realization of convex Archimedean shapes such as snub cube or triakis octahedron (dual of truncated cube) (See page 7 in [32]). In other words, by using squares or rectangles it is not possible to guarantee that the Gaussian curvature of any vertex-hole corresponds to discrete Gaussian curvature of its corresponding vertex in the original polygonal mesh. This problem, in general, is hard. On the other hand, when we have a well-defined planar mesh, it is still straightforward to identify parallelograms that can provide desired local curvatures. In the next section, we present our algorithm to produce panels that are parallelograms or trapezoids.

### 6. Algorithm for Unfolding Quad-Edge Panels

We have developed a software system that automatically creates and draws the shapes of assembly-ready quad-edge panels from any given polygonal mesh. Our goal is to obtain developable or piecewise planar 3D vertex components such that they can be unfolded to a plane with no distortion. In this section, we present the algorithm we used to unfold a planar manifold mesh to obtain quad-edge panels. Our algorithm requires an initial mesh with planar and convex faces. Therefore, the first step is to convert an initial polygonal mesh into a mesh with planar and convex faces. Such a mesh can always be obtained, in the worst case, by triangulating all the faces of the initial mesh.

Our algorithm uses a 2D thickening of graph embedded on planar mesh surface. There are three advantages of this approach: (1) The process always results in a convex planar polygon that is obtained by scaling the original faces of planar mesh (See Figure 17(a) and (b)). (2) Edge thickening always result in (a) planar parallelogram when we use same scaling factor for both neighboring faces and a planar trapezoid when we use different scaling factors for neighboring faces (See Figure 17(c) and (d)). (3) The shapes of the parallelograms preserve the local curvature by producing exact angles. The process is closely related to a corner-cutting operation [41], which is the remeshing scheme of Doo-Sabin subdivision [42]. In this case, we use scaling instead of Doo-Sabin coefficients to guarantee to obtain initial thickening of edges as planar parallelograms or trapezoids. This is used for the construction of Stanford Bunny and genus-3 structures shown in Figures 1 and 2. One advantage of this operation is that it provides a nice smoothed version of the original mesh as if the original mesh is a control mesh of a corner-cutting subdivision scheme as shown in these two examples.

![Figure 15: Comparison of rectangle and parallelogram panels in terms of controlling angles. When we use rectangles, and choose one angle, the opposite angle cannot be controlled, which restricts the flexibility in achieving desired local curvature. Using parallelograms provides additional flexibility since the two angles can be independent.](image1)

![Figure 16: The basic algorithm for 2D thickening for a planar mesh is a corner cutting algorithm: (a) shows a toroidal mesh surface M. (b) shows faces obtained by scaling the original faces of the polygonal mesh. (c) shows the inserting edges between scaled faces. (d) shows the result that corresponds to 2D thickening of graph. The white faces in (d) are 2D thickened vertices and blue faces are 2D thickened edges and yellow regions corresponds to faces.](image2)
we construct a simple rational Bezier surface that provides smooth bending as shown in Figure 17(c) and (d). 3D quad-edge panels and their physical links are produced from these guide shapes as shown in Figure 18. Although we will not show it here the smoothing process continue to preserve local Gaussian curvature. To obtain planar quad-edge panels, we simply unfold each Bezier connector into a plane. Based on the resolution we use to digitize the Bezier curve, this process results in a set of parallelograms or trapezoids as shown in Figure 3. Each of these planar quad-edge panels can still be cut with laser cutter and to facilitate bending, scores are also added. Panels that are used for construction of structures shown in Figure 12 are created using Bezier connectors. In this case, the face-defect equation (the Equation 5.1) cannot be used since the boundaries of open-space is initially created by smooth curves. Based on our results, our conjecture is that this operation still preserves local Gaussian curvature. To resolve this issue, there is a need for a more general definition for quantifying face defects.

Figure 17: Vertex and edge thickening process. (a) shows scaled faces to be used for thickening vertices and edges. Note that scaling guarantees production of edges that are parallel to each other and original edges. (b) shows simple edge thickening that is obtained by connecting the two parallel edges of scaled faces. Note that this operation creates sharp folds. (c) shows the control polygon of the quadric Bezier connector. (d) provides the thickening we use: a quadric non-rational Bezier surface. This connector does not have sharp folds. Moreover, it is developable and has zero Gaussian curvature.

In summary, there are three properties of our quad-edge panels that allow one to obtain a developable sculpture that is a close approximation of the initial manifold mesh surface:

(1): The quad-edge panels are guaranteed to be developable and have a preferred bending direction. Therefore, they can easily be unfolded and folded. Once folded, quad-edge panels become stronger in the perpendicular direction.

(2): Although we did not provide a proof, our conjecture is that the method preserves the local Gaussian curvature. This means that after construction, Gaussian curvature is zero everywhere except at vertex holes. This property guarantees that the shape of the sculpture will resemble the shape of the original polygonal mesh.

(3): Since Gaussian curvature directly comes from the original polygonal mesh, the total discrete Gaussian curvature over the whole sculpture is equal to total discrete Gaussian curvature of the original polygonal mesh, which is the product of the Euler characteristics with $2\pi$. For the sculpture to be closed without any deformation, we need this property.
Using our custom software, we have first converted simpler polygonal meshes into paper sculptures as shown in Figure 12 trying a wide variety of genus-0 shapes. We have then constructed two complicated structures: a simplified version of Stanford Bunny and a genus-3 structure (See Figures 1 and 2). Our software is also used in a Freshman Architecture Studio to construct large shapes. After our two hour presentation, all students were able to construct a wide variety of sculptures using quad-edge panels (See Figure 21. These construction experiments demonstrated that it is not difficult to construct these structures. The main issue for builders is the locating desired quad-edge panels from a large number of pieces. A significant amount of construction time is spent in finding and organizing panels. Therefore, there is a need for strategies for easily finding corresponding pieces among a large number of quad-edge panels.

7. Conclusion and Future Work

In this paper, we have demonstrated that it is possible to form desired shapes by using quad-edge panels. By combining this result with the earlier work in [15], which demonstrated that it is possible to obtain desired smooth bending appropriately heating planar single panel sheets made from shape memory alloy-based laminates, we can claim that it is possible to form desired shapes automatically from single planar structures constructed by quad-edge panels. There is, of course, significant future work needed to be done in order to effectively synthesize complex structures from programmable self-folding active materials for real-world applications. In this paper, we have only shown small size examples, however, quad-edge construction method can be especially effective in constructing much larger shapes when using stronger materials. The structures obtained by this method can also be used as molds to cast large plaster or cement sculptures. Since holes corresponding to faces are already flat, it will be easy to cover them.

An interesting research issue is reduction of the overall footprint of single flattened surface. In some cases, it may be wise to increase amount of overlapping panels to reduce the total footprint of unfolded panels. On the other hand, allowing significant amount of overlapping panels can require more complicated folding procedures and heating strategies. In such cases, development of assembly planning will be an important issue [43].

In this work, we used an extremely simple rectangle packing algorithm since we do not seem to gain much using better algorithms for the shapes we deal with. On the other hand, if the number of pieces becomes significantly large and the difference between the sizes of panels becomes significant, better packing algorithms could be needed [5, 6]. In particular, it can be important to focus on packing parallelograms and trapezoids that could better represent the shapes of quad-edge panels.

In this work, we have also ignored the thickness of the materials. In practical applications, we may not be able to ignore the thickness. Fortunately, there is a simple solution that can be obtained using two sandwiched-layers of panels as shown in Figure 19. With such sandwiched-layers, it is also possible to obtain desired bending changing the relative sizes of two panels. We are also planning to include this property in our implementation.

![Figure 19: Obtaining uniform thickness everywhere using two layers of panels.](attachment:image)

The users have a significant amount of control to obtain final shapes of quad-mesh panels. For instance, they can change scaling parameters, they can change the sizes of physical links, they can even change resolution of Bézier surface (See Figure 20). These choices effects shapes and sizes of both quad-edge panels and open spaces. We have realized that in construction of small structures, it is better to have larger open space, which can allow fingers reach the back side of the structure. On the other hand, for construction of large shapes using large panels it could be possible reduce the relative sizes of open spaces.

One the main issues with our multi-panel unfolding method is that it depends on initial mesh structure and if the number of edges are very high and it can be difficult to locate corresponding panels for construction people. Therefore, there is a strong need to develop strategies for finding corresponding panels. We are planning to segment the shapes into small regions and print each region separately to speed up search process. Another related problem is that identification numbers engraved by laser cutters are not easy to read. Moreover, engraving takes a significant amount of time during laser cutter.
We tried to simplify numbers for faster printing, but it did not really help. During more streamlined process those numbers have to be printed by a regular printer.

A simple way of reducing the number of panels is to use a mesh simplification algorithm [44, 45]. The Stanford Bunny shown in Figure 1 was simplified to 600 edges and 162 faces with a mesh simplification algorithm to speed up construction time. To identify strategies to make construction more efficient, we are planning to build a series of Bunny structures with increasing resolution as architecture group projects. From their inputs, we expect we will develop better strategies for handling large number of panels.

References

[31] Itsphun, geometric construction kit.
[34] L. He, "Über das problem der nachbargebiete, Mathematische Annalen 38 (4) (1891) 477–508.
Figure 21: Examples of quad-Edge sculptures designed and constructed by students in a beginning level architecture studio. The name of the studio and the names of students will be provided later.

Graphical Abstract