# Designing Planar Deployable Objects via Scissor Structures

Ran Zhang, Shiwei Wang, Xuejin Chen, Chao Ding, Luo Jiang, Jie Zhou, and Ligang Liu

**Abstract**—Scissor structure is used to generate deployable objects for space-saving in a variety of applications, from architecture to aerospace science. While deployment from a small, regular shape to a larger one is easy to design, we focus on a more challenging task: designing a planar scissor structure that deploys from a given source shape into a specific target shape. We propose a two-step constructive method to generate a scissor structure from a high-dimensional parameter space. Topology construction of the scissor structure is first performed to approximate the two given shapes, as well as to guarantee the deployment. Then the geometry of the scissor structure is optimized in order to minimize the connection deflections and maximize the shape approximation. With the optimized parameters, the deployment can be simulated by controlling an anchor scissor unit. Physical deployable objects are fabricated according to the designed scissor structures by using 3D printing or manual assembly. We show a number of results for different shapes to demonstrate that even with fabrication errors, our designed structures can deform fluently between the source and target shapes.

**Index Terms**—Computer graphics, digital fabrication

## 1 Introduction

Deployable structures have been widely applied in society, as well as in many fields of industry due to their space-saving capabilities [1], [2], [3]. By changing their shapes, a compact form can be utilized for transportation or storage, while an expanded form can be unfurled for the final use. A variety of space-saving objects (e.g., umbrella, scissor lift) have been engineered in biological science, aerospace structures, architecture, toy design, etc.

The scissor structure is one of the most widely used deployable structures [5], [6]. Fig. 1 shows two well-known planar deployable objects that employ scissor structures. The expandable barricades shown in Fig. 1a are commonly used in traffic stops, store entrances, etc. The expanding circle shown in Fig. 1b deploys a small circle to a big circle, and vice versa [4].

In these existing deployable objects, only regular scissor structures are used to generate the deformation of similar shapes in different scales [7], [8]. Designing a new deployable scissor structure with complex shapes requires knowledge of specific mechanisms and plenty of professional experience. Even for a professional artist, the design process requires a great expenditure of time and effort. It becomes much more challenging when both the compact form and the expanded form are specified. This is a non-trivial problem even for simple 2D curves which are curves without self-intersection. In this paper we focus on designing scissor structures for planar, simple curves. Since scissor structures for 3D cases are more challenging, we will address this in future studies.

Automatically producing a scissor structure given a pair of any 2D simple curves is non-trivial in two folds. The first distinction, which is in contrast to existing work on scissor engineering, reveals that the design process is an inverse problem. This requires that the two states of the scissor structure match the two given shapes. Secondly, the scissor structure contains multifarious degrees of freedom (DOF). While the number of scissor units within a scissor structure formation can widely vary, it is extremely difficult to find a feasible solution. This problem could has potential to become more troublesome if the two given shapes are both closed.

In this paper, we present a two-step system that allows common users to design their own planar deployable shapes using scissor structures. The first step is *topology construction*, which approximates the given shapes in their two different states by determining the number and the shape of the scissor units needed. The second step is *geometry optimization*. This revises the geometry of the scissor units in order to minimize the connection deflections during deployment, and maximize the shape approximation in its two states.

The main contributions of this paper are:

1) Different from the existing scissor structures that only focus on the deployed form, our system generates a deployable scissor structure formation that considers both the source shape and the target shape. Using our two-way design system, the source shape can be expanded to the target shape, and vice versa.

- *R. Zhang, X. Chen, and J. Zhou are with the CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System, University of Science and Technology of China, Hefei, Anhui, China 230026. E-mail: {cuminflea, zjsist}@gmail.com, xjchen99@ustc.edu.cn.*
- *S. Wang, C. Ding, L. Jiang, and L. Liu are with the School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui, China 230026. E-mail: {shiweiw7, dc322, jluo}@mail.ustc.edu.cn, ligang.liu@gmail.com.*

*Manuscript received 27 Oct. 2014; revised 12 Mar. 2015; accepted 7 Apr. 2015. Date of publication 5 May 2015; date of current version 2 Jan. 2016.*
*Recommended for acceptance by A. Tal.*
*For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.*
*Digital Object Identifier no. 10.1109/TVCG.2015.2430322*

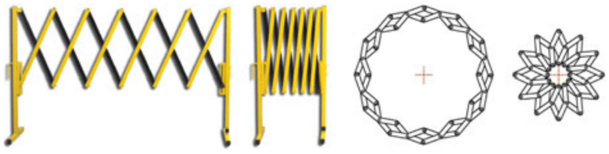



Fig. 1. Two representative deployable objects using regular scissor structures for space-saving. (a) Portable and deployable barricades. (b) The expanding circle [4]. Each object's expanded form is shown on the left and the compact form at right.

2) Given two planar shapes, our algorithm generates a feasible solution in a suggestive fashion in a parameter space with multifarious DOF. The generated scissor structure approximates the given shapes at two states and has few deflections during deployment.

To the best of our knowledge, our system is the first one that allows non-expert users to easily design deployable objects with two specified shapes using scissor structures. Our system is able to generate deployable scissor structures for arbitrary 2D simple curves, as shown in Figs. 13, 14, and 15.

## 2 Related Work

*Deployable structures* are a type of structures that can transform themselves from a small shrunken configuration to a larger expanded configuration. Due to their space-saving properties, deployable structures are frequently studied in the mechanisms field [9]. Metamorphic mechanisms are the generic shape changing mechanisms [10], [11]. For space-saving, there are two typical categories of deployable structures: struts structures and surface structures. The first category includes the scissor-hinged, lazy tong mechanism, and the sliding, or umbrella mechanism. The second category is folded, inflatable or telescopic, etc.

*Scissor structures* are typical samples of deployable structures [5]. They have been used by many notable researchers to design deployable systems. The Hoberman sphere and the Iris Dome [4] are two representative deployable scissor structures designed by Chuck Hoberman. The authors in [7], [8] provide an overall description of scissor units and the global scissor structure, except for closed shapes. Structural analysis and case studies are featured in their research, however, they focus more on analysis than design. In our system, we analyze the geometry properties of scissor structures and propose a method of designing when given two complex shapes. Rosenberg et al. [12] investigates the existed scissor structures and designs a scheme for morphing shapes with pin sliding scissor structures to provide more DOF. In comparison, our system generates more stable deformation based on one-DOF scissor structures to morph between two complex shapes.

*Design of mechanical structure* is attracting a lot of interests in geometry processing. Both static and dynamic mechanical structures have many applications in architecture or animation. Several interactive systems are presented to allow common users to design static structures given the expected geometry [13], [14], [15], [16], [17], [18]. For dynamic mechanical structures, an automatic approach is presented for generating visualization of animating mechanical assemblies [19]. User-specified motion is used to guide the generation of a mechanism assembly in [20]. This system allows users to specify the geometry, time-varying rotation and translation of each rigid component. Non-expert users can easily create mechanical characters using gears and linkages for the defined motion curve in [21]. An interactive system [22] is proposed for designing linkage-based characters, which allows users to browse different linkage topology options, optimize the parameters to improve motion quality, and safeguard against singularities. Mechanical automata can be designed and fabricated from motion capture sequences [23]. In this paper, we employ scissor structures to construct deployable objects that can physically deform between two given shapes.

Geometry, fabrication cost, and workload should all be taken into account during the design phase. With the development of 3D printing techniques, fabricating objects using a 3D printer has become a popular area for research. A data-driven method for designing 3D models that can be fabricated [24] has been proposed. A simple interface allows novice users to choose template parts from a database. Then the system can automatically position, align, and connect the parts, ensuring that the created models can be fabricated. Many algorithms have been proposed to reduce the material cost in printing static models [25], [26]. For malleable object fabrication, elastic materials are used in [27], [28]. However, elastic materials are very expensive for common users and the elastic models are not stable. Instead of elastic materials, joint structures can be used for articulated objects with strategies for fabricating the adaptable model [29], [30]. Furthermore, a method for transforming 3D objects to a cube or a box has been proposed in [31]. It produces a single, connected object that can be physically fabricated and folded from one shape to another by finding appropriate voxelization, tree structure, and a non-intersecting folding sequence. In comparison, we build a very flexible framework that allows different fabrication requirements in order to design deployable objects using scissor structures.

## 3 Scissor Structure

Before the design process, we first describe the geometric representation of a 2D scissor structure in detail. Though only 2D shapes are handled in this paper, the geometric representation of 2D scissor structure can be used for constructing 3D deployable as well. Extending our scheme for 3D cases is a possible future work.

The symbols used in our paper are listed in Table 1. Details of the symbols are explained in the following sections.

### 3.1 Definitions

*Scissor unit*. A scissor unit, also called a scissor-like element, consists of two rigid arms connected by an intermediate hinge allowing a relative rotation [7], as Fig. 2 shows. A rigid arm $A = \{\mathbf{p}^l, \mathbf{o}, \mathbf{p}^r\}$ consists of two straight beams $\mathbf{op}^l$ and $\mathbf{op}^r$ (with lengths $b^l$ and $b^r$, respectively) connected at $\mathbf{o}$ with a fixed angle $\varphi$. A scissor unit is denoted as $X = \{A^1, A^2\} = \{\mathbf{p}^{1,l}, \mathbf{o}^1, \mathbf{p}^{1,r}; \mathbf{p}^{2,l}, \mathbf{o}^2, \mathbf{p}^{2,r}\}$. $\mathbf{o}^1 = \mathbf{o}^2$ is called the revolute joint and $\{\mathbf{p}^{1,l}, \mathbf{p}^{1,r}; \mathbf{p}^{2,l}, \mathbf{p}^{2,r}\}$ are called the pin nodes. The two arms are coplanar. Each $X$ has six parameters $Q(X) = \{b^{1,l}, \varphi^1, b^{1,r}; b^{2,l}, \varphi^2, b^{2,r}\}$, which are of six DOF.

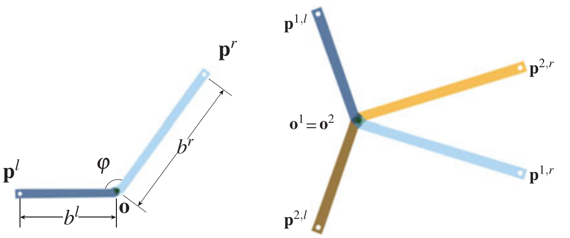
TABLE 1
Table of Symbols Used in the Paper

| Symbol | Meaning |
|---|---|
| $X$ | A scissor unit (SU) |
| $A_1, A_2$ | The two arms of a scissor unit $X$ |
| $\mathbf{p}^l, \mathbf{p}^r$ | The left and right pin nodes of an arm $A$ |
| $\mathbf{o}$ | The revolute joint of an arm $A$ |
| $\varphi$ | The angle of two beams $\mathbf{op}^l$ and $\mathbf{op}^r$ of an arm $A$ |
| $b^l, b^r$ | The length of two beams of $A$, i.e., $b^l = \Vert\mathbf{op}^l\Vert, b^r = \Vert\mathbf{op}^r\Vert$ |
| $l_A$ | The total length of the arm $A$ |
| $u^l(X)$ | The lengths of the left unit line of $X$ |
| $Q(X)$ | Six parameters of $X$, i.e., $Q(X) = \{b^{1,l}, \varphi^1, b^{1,r}; b^{2,l}, \varphi^2, b^{2,r}\}$ |
| $\mathbf{S}$ | The scissor segment $\mathbf{S} = \{X_i\}_{i=1}^n$ consists of $n$ scissor units |
| $\mathscr{S}$ | The scissor structure $\mathscr{S} = \{\mathbf{S}_i\}_{i=1}^m$ with $m$ scissor segments |
| $\mathbf{D}(\mathscr{S}')$ | The connection deflections in the deployed structure $\mathscr{S}'$ |
| $P^0, P^1$ | The two user-specified shapes |

*Unit line*. The line connecting two pin nodes at the same side is called the unit line. The shape of $X$ can be deformed via controlling the length of one of its unit lines, e.g., $u^l(X) = \Vert\mathbf{p}^{1,l}\mathbf{p}^{2,l}\Vert$. During the deployment of a scissor unit, the left beams, $\mathbf{o}^1\mathbf{p}^{1,l}$ and $\mathbf{o}^1\mathbf{p}^{1,r}$ are always on the two opposite sides, as are the right beams, $\mathbf{o}^2\mathbf{p}^{2,l}$ and $\mathbf{o}^2\mathbf{p}^{2,r}$. The deployment will be locked when the two beams meet.

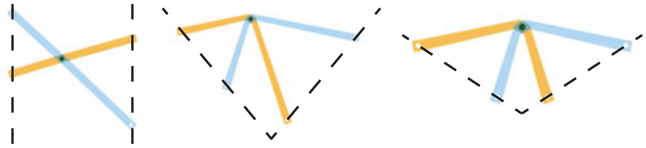
*Types of scissor units*. To approximate various shapes, we use three particular types of scissor units to construct the scissor structure. They are classified by the angles between their two unit lines, as Fig. 3 shows. A *parallel unit* has two unit lines that keep parallel during deployment by using two straight arms with $b^{1,l}b^{2,r} = b^{1,r}b^{2,l}$. An *isogonal unit* keeps the same angle between the two unit lines during deployment by $b^{1,l} = b^{2,l}$ and $b^{1,r} = b^{2,r}$. A *symmetric unit* consists of two symmetric arms, i.e., $b^{l,1} = b^{2,r}, b^{1,r} = b^{2,l}, \varphi_1 = \varphi_2$. The angle between two unit lines changes during its deployment, while the lengths of the two unit lines are always equal to each other.

We use these particular units to construct the topology and geometry of the scissor structure for two given shapes. In order to permit deployment between a wide variety of shapes, the initial scissor units are optimized to generalized scissor units, which are composed of arbitrary lengths and angles of the four beams.




(a) Notations of an arm (b) Notations of a scissor unit

Fig. 2. A scissor unit $X$ consists of two rigid arms $A_1 = \{\mathbf{p}^{1,l}, \mathbf{o}^1, \mathbf{p}^{1,r}\}$, $A_2 = \{\mathbf{p}^{2,l}, \mathbf{o}^2, \mathbf{p}^{2,r}\}$. The two blue beams form a rigid arm $A_1$ and the two yellow beams form the other rigid arm $A_2$.



(a) Parallel unit (b) Isogonal unit (c) Symmetric unit

Fig. 3. Three types of scissor units (unit lines are shown in dashed lines).

*Scissor segment*. A scissor segment $\mathbf{S} = \{X_i\}_{i=1}^n$ is defined as a sequence of $n$ coplanar scissor units connected at their pin nodes by hinges, indicating that

$$\mathbf{p}_i^{1,r} = \mathbf{p}_{i+1}^{2,l},\ \mathbf{p}_i^{2,r} = \mathbf{p}_{i+1}^{1,l}, \quad i = 1, 2, \ldots, n-1. \tag{1}$$

*Joint unit line*. The first scissor unit $X_1$ and the last scissor unit $X_n$ are respectively called the *left joint unit* and the *right joint unit* of $\mathbf{S}$. The unit lines $\mathbf{p}_1^{1,l}\mathbf{p}_1^{2,l}$ of $X_1$ and $\mathbf{p}_n^{2,r}\mathbf{p}_n^{1,r}$ of $X_n$ are *joint unit lines*, where $\mathbf{S}$ can be connected with another scissor segment.

*Scissor structure*. A scissor structure $\mathscr{S} = \{\mathbf{S}_j\}_{j=1}^m$ is composed by a set of $m$ scissor segments $\mathbf{S}_j = \{X_{j,i}\}_{i=1}^{n_j}$ connected at their joint unit lines. The total number of scissor units in $\mathscr{S}$ is $N = \sum_{j=1}^m n_j$. There are generally $6N$ DOF for a scissor structure consisting of $N$ scissor units.

The deployable scissor structure for a 2D shape is composed of a sequence of scissor segments connected by their joint unit lines. To represent a 2D curve, the scissor segments in the scissor structure are connected sequentially as

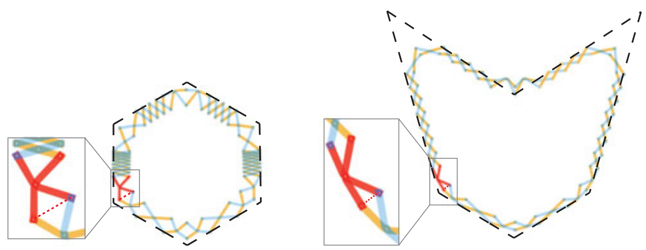
$$\mathbf{p}_{j,n_j}^{1,r} = \mathbf{p}_{j^*,1}^{2,l}, \qquad \mathbf{p}_{j,n_j}^{2,r} = \mathbf{p}_{j^*,1}^{1,l}, \quad j = 1, 2, \ldots, m^*, \tag{2}$$

where $j^* = (j+1) \bmod m$, $m^* = m$ if the scissor structure is closed, or $m^* = m - 1$ if the scissor structure is non-closed.

*Anchor unit*. Since all the scissor units are connected, the deployment of $\mathscr{S}$ from a compact shape to a deployed shape is one DOF. If the length of any of the unit lines in a scissor unit is changed as the driving motion, this scissor unit is called the *anchor unit*. By default, we choose $X_{1,1}$ as the anchor unit and change the length of its left unit line, $u^l(X_{1,1})$, to deploy $\mathscr{S}$, as shown in Fig. 4.

### 3.2 Simulation

Given a scissor structure $\mathscr{S}$ with parameters $Q(\{X_{j,i}\})$ of all the scissor units, we develop a simulation process for computing the geometry of the deployed $\mathscr{S}'$ when we change $u^1(X_{1,1})$. A straightforward way to simulate the deployment is to compute the new positions of all the pin nodes and revolute joints sequentially from the anchor unit $X_{1,1}$ to the last



(a) $t = 0$ (b) $t = 1$

Fig. 4. A scissor structure for two 2D curves in two states. The anchor unit and the anchor unit line are shown in red.

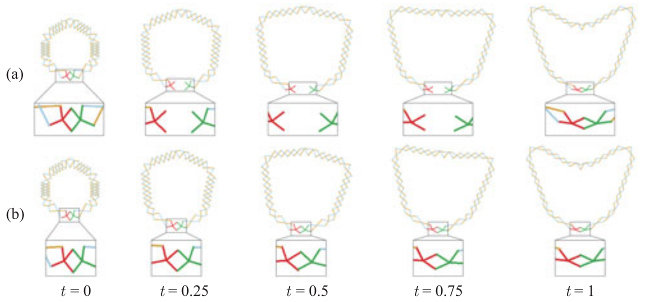



Fig. 5. Deflections of a closed scissor structure during its deployment. (a) Deflections between the anchor unit and the last unit when we propagate the change of the anchor unit line sequentially on each scissor unit in the scissor structure; (b) Least-squares solutions to compute the positions of all the pin nodes during deployment.

scissor unit. For non-closed scissor structures, the positions of all the scissor units can be uniquely computed given the new $u^1(X_{1,1})$. However, for most closed scissor structures, this straightforward propagation can not guarantee the closure of $\mathscr{S}'$ at any time during its deployment, even though it is ideally closed at $t = 0$ and $t = 1$, as Fig. 5a shows.

Instead of directly computing the positions of pin nodes and revolute joints of $\mathscr{S}'$, we compute a set of rigid transformations $\{\mathbf{M}^1_{j,i}, \mathbf{M}^2_{j,i}\}$ of the scissor units $\{X_{j,i}\}$ in $\mathscr{S}$. In this process, $\mathbf{M}^1_{j,i}, \mathbf{M}^2_{j,i}$ respectively transform the two arms, $A^1$ and $A^2$, of $X_{j,i}$ into the counterparts in $\mathscr{S}'$.

*Constraints.* According to the connection topology of $\mathscr{S}$, when the two pin nodes $\mathbf{p}^{1,l}_{1,1}, \mathbf{p}^{2,l}_{1,1}$ of the anchor unit are deployed to new positions $\tilde{\mathbf{p}}^1, \tilde{\mathbf{p}}^2$

$$\mathbf{M}^1_{1,1}\mathbf{p}^{1,l}_{1,1} = \tilde{\mathbf{p}}^1, \qquad \mathbf{M}^2_{1,1}\mathbf{p}^{2,l}_{1,1} = \tilde{\mathbf{p}}^2, \tag{3}$$

the following constraints at the connections must be satisfied. For the two revolute joints of each scissor unit $X_{j,i}$:

$$\overline{\mathbf{o}}_{j,i} \equiv \mathbf{M}^1_{j,i}\mathbf{o}^1_{j,i} - \mathbf{M}^2_{j,i}\mathbf{o}^2_{j,i} = \mathbf{0}, \tag{4}$$

For each pair of adjacent scissor units $X_{j,i}$ and $X_{j,i+1}$ in each scissor segment $\mathbf{S}_j$, $j = 1, 2, \ldots, m$, $i = 1, 2, \ldots, n_j - 1$:

$$\begin{aligned} \overline{\mathbf{p}}^1_{j,i} &\equiv \mathbf{M}^1_{j,i}\mathbf{p}^{1,r}_{j,i} - \mathbf{M}^2_{j,i+1}\mathbf{p}^{2,l}_{j,i+1} = \mathbf{0}, \\ \overline{\mathbf{p}}^2_{j,i} &\equiv \mathbf{M}^2_{j,i}\mathbf{p}^{2,r}_{j,i} - \mathbf{M}^1_{j,i+1}\mathbf{p}^{1,l}_{j,i+1} = \mathbf{0}. \end{aligned} \tag{5}$$

For each pair of connected joint units $X_{j,n_j}$, $X_{j^*,1}$ for adjacent scissor segments $\mathbf{S}_j$ and $\mathbf{S}_{j^*}$, as defined in Eq. (2):

$$\begin{aligned} \overline{\mathbf{d}}^1_j &\equiv \mathbf{M}^1_{j,n_j}\mathbf{p}^{1,r}_{j,n_j} - \mathbf{M}^1_{j^*,1}\mathbf{p}^{2,l}_{j^*,1} = \mathbf{0}, \\ \overline{\mathbf{d}}^2_j &\equiv \mathbf{M}^2_{j,n_j}\mathbf{p}^{2,r}_{j,n_j} - \mathbf{M}^1_{j^*,1}\mathbf{p}^{1,l}_{j^*,1} = \mathbf{0}. \end{aligned} \tag{6}$$

*Equation systems.* If the scissor structure is non-closed, the equation system consisting of Eqs. (3,4,5,6) has $(4 + 2N + \sum_{j=1}^{m} 4(n_j - 1) + 4(m-1) = 6N)$ equations and thus $\mathscr{S}'$ can be obtained by directly solving this equation system.

*Least-squares solution.* However, if the scissor structure is closed, i.e., the first scissor segment $\mathbf{S}_1$ and the last one $\mathbf{S}_m$ are connected by a joint unit line, the equation system is *over-constrained*. A least-squares solution for $\mathscr{S}'$ is solved. Denote $\mathbf{D}$ as the vector consisting of all the deflections, i.e., $\mathbf{D} = [\{\overline{\mathbf{o}}_{j,i}\}, \{\overline{\mathbf{p}}^1_{j,i}, \overline{\mathbf{p}}^2_{j,i}\}, \{\overline{\mathbf{d}}_j\}]^T$. The transformations of all arms can be obtained by minimizing the connection deflections of all pin connections and revolute joints in $\mathscr{S}'$ as a shape matching problem which minimizes the difference between corresponding positions of pin connections and revolute joints that similar to [32]:

$$\begin{aligned} &\min_{\{\mathbf{M}^1_{j,i}, \mathbf{M}^2_{j,i}\}} \quad \mathbf{D}^T\mathbf{D} \\ &\text{s.t.} \qquad\qquad Eq.\ (3). \end{aligned} \tag{7}$$

Fig. 5b shows the least-squares solutions for the deployed scissor structure at different times. Compared with the large deflection between the anchor unit and the last unit in Fig. 5a, the least squares solution distributes the deflections to all the pin connections. Therefore, the closure of the scissor structure is well preserved during its deployment. Considering the fabrication error in pin connections, the least squares solution gives more accurate simulations for closed scissor structures.

## 4 Problem and Design Overview

*Problem.* Suppose we are given two planar shapes $P^0$ and $P^1$. Our goal is to design a scissor structure $\mathscr{S}$ which approximates the source $P^0$ and can be deployed to the target $P^1$. The deployment of $\mathscr{S}$ represents a physical morphing between $P^0$ and $P^1$.

*Challenges.* This problem is very challenging for two reasons. Firstly, our perspective differs from existing research in that we regard scissor design as an *inverse* problem. Therefore, the two states of the scissor structure must match the two given shapes. Secondly, there are a wide range of DOF within the scissor structure since the number of scissor units in a scissor structure formation can vary extensively. This problem can become more troublesome if the two given shapes are both closed because there are more constraints for $\mathscr{S}$.

*Factors.* There are two factors in determining $\mathscr{S}$: *topology* and *geometry*. Topology is defined as the number of scissor units used in $\mathscr{S}$ and how they are organized. Geometry refers to the parameters of all scissor units. To get a good scissor structure, we first construct a feasible topology, and then optimize its geometry.

*System overview.* We consider both topology and geometry in order to develop a computational approach for generating a scissor stucture $\mathscr{S}$. This structure has the capability to transform from the shape $P^0$, into another shape, $P^1$. Fig. 6 shows the pipeline of our algorithm, which consists of two main steps: topology construction and geometry optimization.

In the first step, we construct the topology of the scissor structure $\mathscr{S}$, which approximates $P^0$ and $P^1$ at two states respectively. Specifically, we use particular types of scissor units to compose $\mathscr{S}$, and determine its topology and parameters based on the input shapes. If one of the two shapes is *non-closed*, $\mathscr{S}$ gives the exact deployment from $P^0$ to $P^1$. If both $P^0$ and $P^1$ are *closed*, we expect that the scissor structure is always closed. However, the closure of $\mathscr{S}$ might not be guaranteed during deployment even though the initial state and the final state of $\mathscr{S}$ match $P^0$ and $P^1$ respectively.

Although it is difficult to achieve, in the second step we optimize the geometry of $\mathscr{S}$ to ensure its closure at any time during its deployment. We sample the deployment process

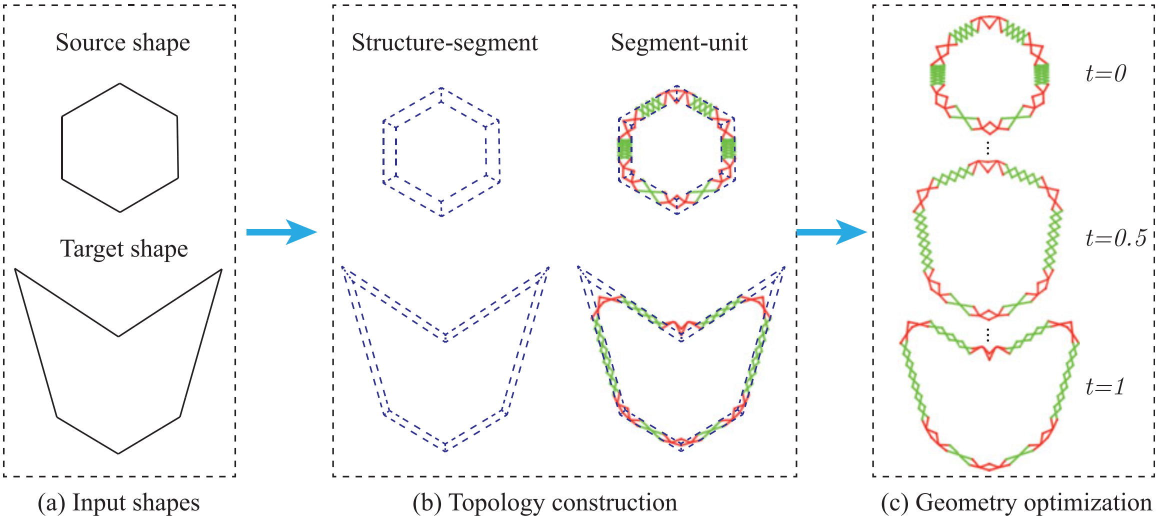



Fig. 6. Overview of our algorithm. (a) Given the source and target 2D curves, (b) in the topology construction step, we first decompose the scissor structure into segments given shape correspondences. For each scissor segment, joint units are first placed at each vertex, then parallel units along the edge. (c) Parameters of all the scissor units are finally optimized in order to better approximate the input shapes and minimize the connection deflections at all pin nodes at any time $t \in [0, 1]$.

and minimize the node deflections at the sampled timing. Meanwhile, the change of its geometry is also constrained to approximate the given shapes.

## 5 Topology Construction

The deployment of the scissor structure $\mathscr{S}$ between the two shapes $P^0$ and $P^1$ is guaranteed by its topology consistency, i.e., the number and the connection of all scissor units keep the same. To approximate the input shapes, we first compute the vertex and edge correspondences between them. We adopt the method of [33] to generate the correspondence between the two 2D curves. Thus both the two shapes have $K$ vertices with correspondences, i.e., $P^0 = \{V_j^0\}_{j=1}^K$ and $P^1 = \{V_j^1\}_{j=1}^K$. Our system also allows the user to manually specify or modify the vertex correspondences.

Given the vertex and edge correspondences, we construct a scissor structure $\mathscr{S}$ that closely approximates $P^0$ and $P^1$ when $t = 0$ and $t = 1$ respectively. Because each scissor unit is deployed from the source shape to the target shape, each edge in $P^0$ must be shorter than its corresponding edge in $P^1$. The user can modify the size of the target shape, or our system generates a feasible size for the target shape.

Because of the huge number of DOF in both topology and geometry of the scissor structures, we use a constructive scheme to generate a reasonable initial solution of $\mathscr{S}$. Firstly, we construct the topology of $\mathscr{S}$ by decomposing it into a set of connected scissor segments, called the *structure-segment topology*. Each scissor segment is constructed to deploy an edge at $t = 0$ to the edge at $t = 1$. Secondly, we construct the *segment-unit topology* to determine how each scissor segment is composed by scissor units. The parameters of each unit are initialized to approximate the shapes in both the compact form and the expanded form.

### 5.1 Structure-Segment Topology Construction

Given two curves $P^0 = \{V_j^0\}_{j=1}^K$ and $P^1 = \{V_j^1\}_{j=1}^K$ associated with vertex and edge correspondences, we use a sequence of scissor segments $\{\mathbf{S}_j\}_{j=1}^m$ as defined in Eq. (2) to approximate the two curves by the envelope of $\mathscr{S}$, as shown in Fig. 6b. $m = K$ if the two curves are both closed. Otherwise, $m = K - 1$.

Each edge between two adjacent vertexes at two states $V_j^0, V_{j+1}^0$, $V_j^1, V_{j+1}^1$ is represented by the scissor segment $\mathbf{S}_j = \{X_i\}_{i=1}^{n_j}$ (we will omit subindex $j$, $j+1$ for simplicity next). Generally, the joint unit lines $\mathbf{p}_1^{1,l}\mathbf{p}_1^{2,l}$ and $\mathbf{p}_n^{1,r}\mathbf{p}_n^{2,r}$ can be any direction in the range between the two edges at each vertex. Denote the angle at each vertex between two edges as $\hat{\theta}$, where $\hat{\theta} \in [0, 2\pi]$, as shown in Fig. 7. The difference between two angles at $t = 0$ and $t = 1$ is denoted as $\delta\theta = |\hat{\theta}^0 - \hat{\theta}^1|$, where $\delta\theta \in [0, 2\pi]$. For a single scissor unit, the angle difference between two unit lines at two states belongs to the interval $[0, \pi]$. Therefore, we use two scissor units at each vertex to obtain a large range of angle alternations. We then set angle's bisector at each vertex as the joint unit line direction of each pair of adjacent scissor segments.

### 5.2 Segment-Unit Topology Construction

Scissor segment-unit topology construction is used to generate a number of scissor units $\{X_i\}_{i=1}^n$ and their parameters. This process also serves to approximate the shapes in both the compact form and deployed form as accurately as

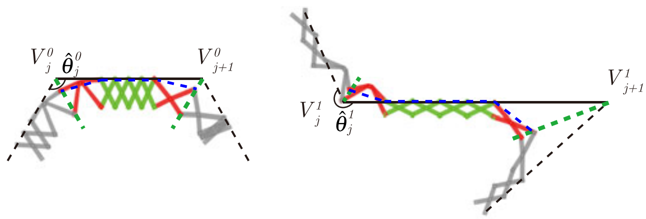
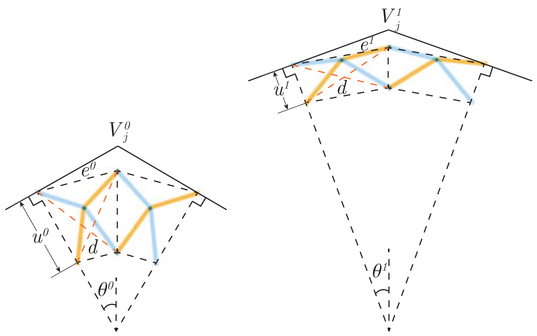



Fig. 7. A scissor segment for an edge shape with two symmetric joint units (red) and a group of parallel units (green). The joint unit lines (green dashed lines) are the bisectors of the two vertexes. The edge (black line) is approximated by the envelope of the scissor segment (blue dashed line).




(a) Source shape (b) Target shape

Fig. 8. Two symmetric scissor units are used at a vertex.

possible. We select particular types of units and compute their parameters according to the input shapes.

According to the properties of each type of scissor unit described in Section 3, the symmetric unit is uniquely suited for deploying a large range of angles and to keep the two unit lines with the equal length. We select symmetric units as the joint units $X_1$, $X_n$ in each scissor segment. The two joint unit lines lie on the bisectors of the two vertexes, and the other unit lines are perpendicular to the edge in both states. The parallel unit is selected for $\{X_2, \ldots, X_{n-1}\}$, as Fig. 7 shows.

### 5.2.1 Construction of Joint Scissor Units

The parameters of the two joint units $X_1$ and $X_n$ are first computed. We will take $X_1$ as an example for explanation. As Fig. 8 shows, a symmetric unit forms an isosceles trapezoid that denoted by its four pin nodes $\mathbf{p}^{1,l}, \mathbf{p}^{2,l}, \mathbf{p}^{2,r}, \mathbf{p}^{1,r}$. The left unit line lengths are shown as $u^0$,$u^1$, the angles as $\theta^0$, $\theta^1$, and the lengths of $\mathbf{p}^{1,l}\mathbf{p}^{2,r}$ as $e^0$, $e^1$ at $t=0$, $t=1$, respectively. $\theta^0$ and $\theta^1$ are defined as $\theta^0 = \frac{\pi-\hat{\theta}^0}{2}$ and $\theta^1 = \frac{\pi-\hat{\theta}^1}{2}$, where $\hat{\theta}^0$ and $\hat{\theta}^1$ are the angles at the vertex at $t=0$ and $t=1$. Its diagonal length $d$ can be computed as

$$d = \sqrt{u^2 \cos^2\frac{\theta}{2} + \left(e - u \sin\frac{\theta}{2}\right)^2} \tag{8}$$

given $u, e, \theta$ at any time $t$.

Given $u^0$, $e^0$, and $\theta^0$, $d$ can be computed first. Then the four pin nodes at $t=0$ can be determined from $d, u^0, e^0, \theta^0$. The four pin nodes at $t=1$ are determined by computing $e^1$ given $d, u^1, \theta^1$ as

$$e^1 = \sqrt{d^2 - \left(u^1 \cos\frac{\theta^1}{2}\right)^2} + u^1 \sin\frac{\theta^1}{2}. \tag{9}$$

The revolute joints $\mathbf{o}^0$ and $\mathbf{o}^1$ are determined by solving the four equations $b^{1,l} = b^{2,r}|_{t=0}$, $b^{1,l} = b^{2,r}|_{t=1}$, $b^{1,l}|_{t=0} = b^{1,l}|_{t=1}$, $b^{1,r}|_{t=0} = b^{1,r}|_{t=1}$, given the four pin nodes at $t=0$ and $t=1$. When we get multiple solutions for the revolute joint from the quadric equation system, we choose the one that is closer to the center of the trapezoid.

### 5.2.2 Construction of Parallel Units

After determining the two joint units, we construct the scissor units for the remaining part of the edge. Because the right unit line of $X_1$ and left unit line of $X_n$ in the scissor segment are both perpendicular to the edge in two states, we use a sequence of parallel scissor units. In order to deploy an edge $\mathbf{p}_a\mathbf{p}_b$ from length $L^0$ to $L^1$, a single scissor unit might not be able to deploy the initial edge to the target edge given $u^0, u^1$. Fig. 9 shows an edge that was constructed with differing numbers of scissor units. A longer edge can be deployed from the same initial edge if more scissor units are used. However, the integral number of the same parallel scissor units may not be precisely deploy the edge from $L^0$ to $L^1$ given $u^0$, $u^1$.

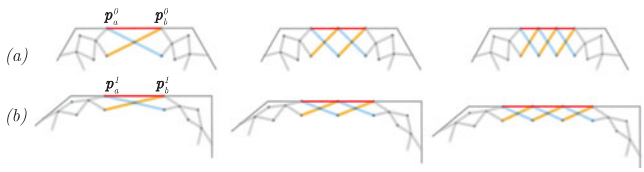



Fig. 9. An edge is represented using different numbers of scissor units. Given $u^0, u^1$, the edge length at $t=1$ is larger and contains more scissor units (b) from the same edge at $t=0$ (a).

Therefore, we first determine the minimum number of parallel scissor units needed as, $n^* = \lceil\sqrt{\frac{(L^1)^2-(L^0)^2}{(u^0)^2-(u^1)^2}}\rceil$. If $n^* < 1$, we reduce the size of the source shape to guarantee the deployment to the target shape. Given $n^*$, we compute the parameters of the parallel scissor units, as Fig. 10 shows. The first scissor unit is a parallel unit with $a = b^{1,l} = b^{2,l}$, $b = b^{2,r} = b^{1,r}$, the last scissor unit is symmetric to the first scissor unit with $a = b^{2,r} = b^{1,r}$, $b = b^{1,l} = b^{2,l}$, and the middle parallel units have four beams with the same length $b$. The scissor units can be determined by solving $\phi^0, \phi^1, a, b$ from the following equation system

$$\begin{cases} u^0 & = & 2a\cos\phi^0 \\ u^1 & = & 2a\cos\phi^1 \\ 2(n^*-1)b\cos\phi^0 & = & L^0 - 2a\sin\phi^0 \\ 2(n^*-1)b\cos\phi^1 & = & L^1 - 2a\sin\phi^1. \end{cases} \tag{10}$$

In the topology construction, the lengths $u^0, u^1, e^0$ play important roles in the generated scissor units. Typically $u^0 = \lambda u^1$, where $\lambda$ is the deployment ratio of the unit lines. $u^0 = e^0 = \mu L_{min}$, where $L_{min}$ is the minimum edge length in $P^0$. We set $\lambda = 2, \mu = 0.2$ in our system. However, due to a large variety of input shapes, these settings might casually lead to interpenetration between scissor units. The scissor units must be large enough in order to allow easy fabrication. Therefore, we provide a very simple way for the user to adjust two sliders to set $u^0$ and $e^0$, and show the simulated results of the generated scissor structure.

After we construct the scissor segment for each edge, we simply connect the adjacent scissor segments according to

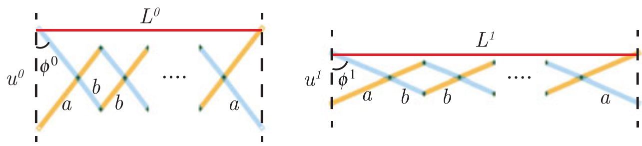



Fig. 10. A sequence of parallel units for an edge from $L^0$ to $L^1$.

the structure-segment topology, thus generating the complete scissor structure $\mathscr{S}$.

## 6 Geometry Optimization

The generated scissor structure in topology construction approximates the source shape and the target shape at $t = 0$ and $t = 1$ respectively. However, the closure of $\mathscr{S}$ is not guaranteed during deployment. To this end, we allow minor deflections at the connection nodes (both the revolute joints and pin nodes) to ensure the closure of $\mathscr{S}$ at any time. As we will discuss later, this is a reasonable course of action since fabrication errors often occur during construction. We optimize the parameters of $\mathscr{S}$ to minimize the total connection deflections during deployment, and approximate to $P^0$ and $P^1$ as much as possible.

*Connection deflections.* The lengths of the anchor unit line of $\mathscr{S}$ at $t = 0$ and $t = 1$ are $u^0$ and $u^1$ respectively. For any time $t \in [0, 1]$, the anchor unit line length is computed by $u(t) = (1-t)u^0 + tu^1$. Ideally, the deflections $\mathbf{D}(\mathscr{S}; u(t)) = \mathbf{0}$ for any $t \in [0, 1]$. However, it is not always possible. Therefore, we optimize the geometry of $\mathscr{S}$ to achieve the minimum deflections of all the nodes during the entire deployment

$$E_{conn} = \int_0^1 \mathbf{D}^T\big(\mathscr{S}; u(t)\big)\mathbf{D}\big(\mathscr{S}; u(t)\big)dt. \tag{11}$$

We uniformly sample $K$ moments ($K = 100$ in our implementation) in $[0, 1]$ and approximate the integral with the sum

$$\hat{E}_{conn} = \frac{1}{K}\sum_{k=0}^{K} \mathbf{D}^T\big(\mathscr{S}; u(t_k)\big)\mathbf{D}\big(\mathscr{S}; u(t_k)\big)\Delta t. \tag{12}$$

*Remark.* The solution minimizing the discrete form $\hat{E}_{conn}$ still works for the continuous form $E_{conn}$ because $E_{conn} - \hat{E}_{conn}$ is bounded by a small value. In this way, the deflections are limited in order to ensure the successful closure of $\mathscr{S}$. For detailed proof, please refer to the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2015.2430322 available online.

*Shape approximation.* The envelope of the pin nodes in $\mathscr{S}$ is expected to be close to the input shapes. We define the shape approximation error as

$$D(\mathscr{S}, P) = \frac{1}{2N}\sum_{i=1}^{N} \big(d(\mathbf{p}_i^{1,l}, P) + d(\mathbf{p}_i^{2,r}, P)\big), \tag{13}$$

where $d(\mathbf{p}, P)$ is the minimum distance from a pin node $\mathbf{p}$ to the polygonal shape $P$. The total shape approximation error of the scissor structure to the two given shapes is defined as

$$E_{shape} = D(\mathscr{S}^0, P^0) + D(\mathscr{S}^1, P^1). \tag{14}$$

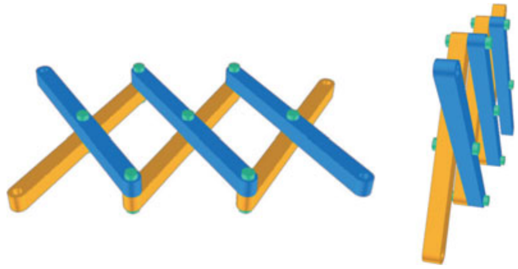


Fig. 11. All the $A_1$ arms are on the top of $A_2$ arms in the scissor structure for fabrication. Two different views are shown.

*Optimization.* Considering both the shape approximation error and the connection deflections, we look for the optimal parameters of $\mathscr{S}$ to

$$\min_{Q_\mathscr{S}} E = \hat{E}_{conn} + \lambda E_{shape}, \tag{15}$$

where $Q_\mathscr{S} = \{Q(X) : X \in \mathscr{S}\}_{i=1}^N$ are all the geometric parameters to be optimized and $\lambda$ is used to to balance the two terms ($\lambda = 0.001$ in our system). $\lambda$ does not affect the optimization for non-closed shapes since the deflections are always zero. For closed shapes, the deflections are desired to be as small as possible for fabrication.

We estimate $\frac{\partial E}{\partial Q_\mathscr{S}}$ using finite differences. This requires us to instantiate the scissor structure with different parameters $\widetilde{Q}_\mathscr{S}$, and compute the value of the objective function $E$. To improve the convergence rate, the BFGS quasi-Newton method is used to estimate the approximate Hessian $\frac{\partial^2 E}{\partial^2 Q_\mathscr{S}}$.

## 7 Fabrication

In order to validate the scissor structures generated by our system, we fabricate them using 3D printing or wood models. However, physical constraints and easy assembly are two issues to be aware of during the design process. We will discuss some fabrication issues in this section.

With the thickness of the materials in fabrication, the two arms of a scissor unit are not ideally coplanar. We set the arm $A_1$ is on the top layer and $A_2$ on the bottom layer for each unit. By connecting two adjacent units with $A_1$ is on the top of $A_2$, each segment can be coplanar, as shown in Fig. 11. For 2D shapes, the scissor structure can be physically deployed by connecting the joint units of adjacent segments, the same as adjacent scissor units. The scissor structure is still a planar shape that represents planar curves.

Regular scissor units are preferred in manual assembly due to their low cost and easy composition. In comparison, 3D printing does not require any constraints on the scissor unit shape. The advantage of our system is its flexibility in solving design problems with different requirements on the scissor unit parameters. If straight arms are desired for manual manufacturing, the parameter optimization will be solved with $\varphi_i^1 = \pi, \varphi_i^2 = \pi$ in a smaller parameter space. Fig. 12 compares the generated scissor structures with generalized scissor units, and with scissor units of straight arms.

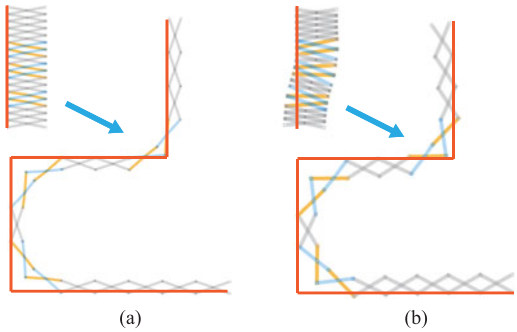



Fig. 12. Scissor structures generated using scissor units with different shapes. (a) Generalized units are used. The colored scissor units have non-straight arms. (b) Only scissor units with straight arms are used. With stricter shapes for the scissor units, the scissor structure does not resemble the source shape as closely.

## 8 Results and Discussions

We tested our algorithm to generate deployable scissor structures on a number of shapes. A group of simulated results is shown first in Fig. 13, followed by a group of fabricated objects using a 3D printer, as shown in Fig. 14 and using wood, as shown in Fig. 15. We will discuss how the factors in our algorithm affect the final results.

### 8.1 Results

Our system has proven to be very flexible on a wide range of shapes, including concave and convex 2D curves, as well as non-closed and closed curves.

Since the complex shapes are difficult to fabricate using our current fabrication method, we demonstrate our algorithm works well for complex shapes by simulating results, as shown in Fig. 13, ignoring the additional error caused by fabrication. The generated scissor structures closely resemble the input shapes, while the deployment process prevents deflections from occurring at all nodes. The simulation results demonstrate that our system is very effective for designing deployable scissor structures for a wide range of 2D complex shapes.

When given a generated scissor structure, physical objects can be created by 3D-printing or manual assembly. 3D printing has greater accuracy and fabrication flexibility, while manual assembly of freeform shapes requires much more manual work.

Five 2D objects are printed out using the 3D printer ProJet® 3500 HDMax, as shown in Fig. 14, including both closed shapes and non-closed shapes. Figs. 14a, 14b, and 14c show the transformations between three non-closed shapes, while Fig. 14d, 14e show the transformations between two closed shapes.

Two 2.5D objects are fabricated using wood to demonstrate the practicability of our system, as shown in Fig. 15a and 15b. A chair is deployed from a short line by two duplicated scissor structures connected by metal bars. As shown in Fig. 15a. After reinforcing it with a support bar, the chair is able to hold objects like a large teddy bear. Another example is a wood shelf that is deployed from a short line. After connecting the two scissor units at the two ends, a stable shelf is ready for hanging things, as Fig. 15b shows. The chair and the star shelf can be assembled in six hours for cutting the straight arms and compose the arms.

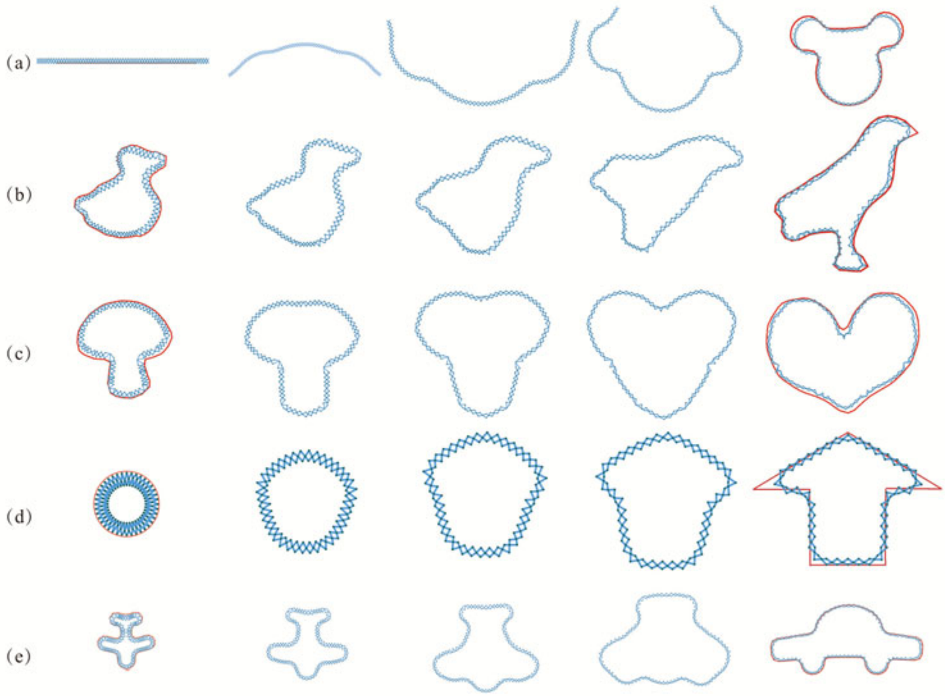



Fig. 13. Simulation for complex shapes. Our system is very flexible on a wide range of shapes. From left to right: the source shape and the scissor structure at $t = 0$; the scissor structure at the sampled times $t = 0.25$, $t = 0.5$, $t = 0.75$ during deployment; and the scissor structure at $t = 1$ approximating the target shape. From top to bottom: 2D cases including (a) Line-Micky, a non-closed shape to a closed shape. Four pairs of 2D closed shapes: (b)Duck-bird; (c)Mushroom-heart; (d) Circle-house; (e) Airplane-car. Note the transformations between concave and convex angles in 2D scissor structures, especially in (c,d). Our geometric optimization guarantees the closure of the scissor structure for closed shapes.

(a)
(b)
(c)
(d)
(e)

Fig. 14. Photos of the real objects fabricated by 3D printing with SLA material. From left to right: $t = 0$; $t = 0.25$; $t = 0.5$; $t = 0.75$; $t = 1$. (a,b,c) The deployment of three objects between three pairs of *non-closed* shapes, which are Capricorn-Libra and letters S-G, respectively. (d,e) Two deployable objects between two *closed* shapes, circle-star, square-car.

## 8.2 Discussions

Our system is very effective for generating superior scissor structures with acceptable deflection. The statistics of all the concerned factors are listed in Table 2, including the number of scissor units, the iteration number $N_{iter}$ in the geometry optimization, time (msec) used to generate each scissor structure, the maximum length $L^{arm}_{max}$ (measured in mm) of the arms, the overall errors $E^{top}_{overall}$, the maximum deflections $D^{top}_{max}$ and $D^{opt}_{max}$ (measured in mm), the shape error $E^{top}_{shape}$ (measured in mm), the connection deflection $\hat{E}^{top}_{connection}$ (measured in mm$^2$) of the scissor structure generated from the topology construction, and the corresponding $E^{opt}_{overall}$, $D^{opt}_{max}$, $E^{opt}_{shape}$, $\hat{E}^{opt}_{connection}$ after the geometry optimization.

*Number of scissor units.* For the two given shapes, we currently construct the topology of the scissor structure using as few scissor units as possible with empirical values of $u^0, u^1$. Smaller $u^0, u^1$ result in more scissor units in the scissor structure, but it can more closely resemble the given shapes. In contrast, bigger $u^0, u^1$ results in fewer scissor units. However, the generated scissor structure does not well approximate to the given shapes. Fig. 16 shows a comparison. For complex shapes, more units are preferred to construct a scissor structure that better approximates the given shapes.

*Time.* The time used to generate the scissor structure depends heavily on the shape's degree of complexity. For non-closed shapes, the design process is very fast (less than one second) because only the shape error needs to be optimized while the connection deflection is always zero. Only a few iterations are required in the geometry optimization steps. For closed shapes, the connection deflections at each sampled time should be computed for the geometric optimization. The geometry optimization requires more iterations

(a)
(b)

Fig. 15. Photos of the real objects fabricated by a carpenter with wood. From left to right: $t = 0$; $t = 0.25$; $t = 0.5$; $t = 0.75$; $t = 1$; deployed states with actual functionalities. (a) A wooden chair deployed from a short line shape. (b) A wooden shelf in a star shape deployed from a short line shape. Note the space-saving compact forms of the deployable scissor structures generated by our system.

TABLE 2
Statistics of Experiment Results

| | Number | $N_{iter}$ | Time (ms) | $L_{max}^{arm}$ | $E_{overall}^{top}$ | $E_{overall}^{opt}$ | $D_{max}^{top}$ | $D_{max}^{opt}$ | $E_{shape}^{top}$ | $E_{shape}^{opt}$ | $\hat{E}_{connection}^{top}$ | $\hat{E}_{connection}^{opt}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Fig. 13a | 71 | 29 | 112,759 | 54.7516 | 0.0068 | 0.0054 | 0 | 0 | 6.7577 | 5.4417 | 0 | 0 |
| Fig. 13b | 71 | 112 | 724,843 | 71.3421 | 0.1088 | 0.0020 | 0.1219 | 0.0122 | 8.3685 | 1.5689 | 0.1004 | 0.0005 |
| Fig. 13c | 73 | 107 | 745,261 | 80.2232 | 0.1325 | 0.0041 | 0.4548 | 0.0197 | 9.5027 | 1.8530 | 0.1230 | 0.0022 |
| Fig. 13d | 41 | 105 | 420,280 | 63.3478 | 0.0149 | 0.0088 | 0.0375 | 0.0351 | 6.179 | 5.7345 | 0.0087 | 0.0025 |
| Fig. 13e | 93 | 113 | 953,330 | 58.175 | 0.0567 | 0.0037 | 0.0598 | 0.0187 | 2.8914 | 1.6946 | 0.0538 | 0.0021 |
| Fig. 14a | 23 | 5 | 23 | 52.345 | 0.0123 | 0.0095 | 0 | 0 | 12.3372 | 9.4453 | 0 | 0 |
| Fig. 14b | 30 | 15 | 620 | 57.822 | 0.0035 | 0.0033 | 0 | 0 | 3.5322 | 3.3310 | 0 | 0 |
| Fig. 14c | 21 | 4 | 17 | 50.173 | 0.0108 | 0.0066 | 0 | 0 | 10.7834 | 6.5467 | 0 | 0 |
| Fig. 14d | 51 | 105 | 522,793 | 62.1563 | 0.0034 | 0.0012 | 0.0168 | 0.0134 | 0.1286 | 0.4167 | 0.0033 | 0.0008 |
| Fig. 14e | 61 | 114 | 625,302 | 61.1327 | 0.0048 | 0.0014 | 0.0175 | 0.0135 | 0.3491 | 0.4503 | 0.0045 | 0.0009 |
| Fig. 15a | 15 | 3 | 17 | 200 | 0.0395 | 0.0375 | 0 | 0 | 39.5304 | 37.464 | 0 | 0 |
| Fig. 15b | 12 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

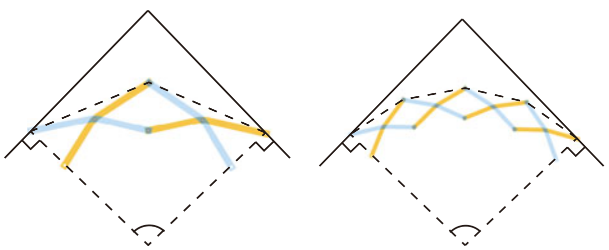
to converge. It usually takes $2 \sim 20$ minutes to obtain a suitable scissor structure for closed shapes, as listed in Table 2.

*Connection deflection.* For 2D closed shapes, our topology construction generates reasonable scissor structures. However, the deflections are still much larger compared with the level of precision in current 3D printers. The overall deflections decrease notably after geometric optimization. At each pin connection, the deflection is much smaller than with 3D printing, as compared in Table 2. Generally, for a 2D scissor structure with about $5\,\mathrm{cm}$ arms in 2D cases, there is only 0.02 mm of deflection, which is quite smaller than the precision of most available 3D printers.

Table 2 shows that the node deflection is zero for non-closed shapes. The shape errors are smaller after optimization, which means that the scissor structures are closer to the input shapes. For closed shapes, the connection constraints are more important for simulation and fabrication. Therefore, the shape errors do not decrease by much after optimization. However the deflections are markedly smaller.

## 9 Conclusion

We present an automatic system that allows non-expert users to easily design planar deployable objects with scissor structures. Given two 2D shapes, a scissor structure can be generated to approximate the input shapes in its two states. The geometry is then optimized to generate minimum deflections during deployment. The following simulated process generates a low-error deployable scissor structure for both closed and non-closed shapes. With 3D printing or wood fabrication, the generated scissor structure can be easily made into real objects. The results demonstrate that our system works well for a variety of shapes. With the flexibility of designing deployable scissor structures for space-saving, our system has innumerable applications in convertible furniture, architecture, etc.



(a) Two scissor units at a vertex (b) Four scissor units at a vertex

Fig. 16. Using more scissor units, the generated scissor structure gets closer to the input shape (black line).

*Limitations and future work.* Though our system solves the challenging problem of generating scissor structures for a wide range of shapes, limitations exist in the current system. After the fabrication step is complete, a few highly complicated models may have difficulties deploying fluidly due to fabrication errors or physical constraints. In the future, we plan to take the physical transmission of scissor units into account for more robust deployment results.

Stability of fabricated models is not analyzed in our current system. Additional inside linkages can be used to strengthen the stability of complex scissor structures. For example, the chair model shown in Fig. 15b requires an additional piece of wood as support. An interactively editing interface, such as [34], for the physically supported scissor structure can be a future topic as well.

In the current system, the number of scissor units is determined in topology construction and does not change in the optimization step. Smaller numbers of units may be desired in scissor structure composition to allow for ease in manufacturing. Therefore, optimization of the number of scissor units is another potential direction for future work.

For 3D cases, the existed framework use a set of uniform scissor units to form a regular shape, like spheres, cubes, etc. General 3D cases are more complicated, since large amounts of closed loops exist in 3D cases. It would be a challenge to handle the deflection problem in a general 3D case. However, this poses an intriguing question for future research on how to design deployable structures for arbitrary 3D shapes.

## Acknowledgments

The authors would like to thank the reviewers for their generous reviews that shaped the direction of the paper. Ran Zhang and Shiwei Wang both contributed equally to this paper and should be considered equal first authors. This work was supported by the National Natural Science Foundation of China under Nos. 61472377, 61331017, 61222206 and partially supported by the Fundamental Research Funds for the Central Universities under No. WK2100060011 as well as the One Hundred Talent Project of the Chinese Academy of Sciences. X. Chen is the corresponding author.

## REFERENCES


[1] P. Mollerup, *Collapsibles: A Design Album of Space-Saving Objects*. Thames & Hudson, 181A High Holborn, London, United Kingdom, 2001.

[2] S. Pellegrino, *Deployable Structures*. New York, NY, USA: Springer, 2001.

[3] C. H. H. Merchan, "Deployable structures," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 1987.

[4] C. Hoberman, [Online]. Available: http://www.hoberman.com

[5] J.-S. Zhao, F. Chu, and Z.-J. Feng, (2009). The mechanism theory and application of deployable structures based on SLE. *Mechanism and Machine Theory* [Online] vol. 44, no. 2, pp. 324–335. Available: http://www.sciencedirect.com/science/article/pii/S0094114X08000815

[6] F. Maden, K. Korkmaz, and Y. Akgün, "A review of planar scissor structural mechanisms: geometric principles and design methods," *Archit. Sci. Rev.*, vol. 54, no. 3, pp. 246–257, 2011.

[7] N. De Temmerman, "Design and analysis of deployable bar structures for mobile architectural applications," Ph.D. dissertation, Department of Architectural Engineering Sciences, Vrije Univertiteit Brussel, Brussels, Belgium, 2007.

[8] L. Alegria Mira, "Design and analysis of a universal scissor component for mobile architectural applications," Ph.D. dissertation, Department of Architectural Engineering Sciences, Vrije Univertiteit Brussel, Brussels, Belgium, 2007.

[9] A. Hanaor and R. Levy, "Evaluation of deployable structures for space enclosures," *Int. J. Space Structures*, vol. 16, no. 4, pp. 211–229, 2001.

[10] L. Zhang and J. Dai, "An overview of the development on reconfiguration of metamorphic mechanisms," in *Proc. ASME/IFToMM Int. Conf. Reconfigurable Mechanisms Robots*, 2009, pp. 8–12.

[11] M. A. Puchetaa, A. Buttib, V. Tamellinib, A. Cardonaa, and L. Ghezzi, "Topological synthesis of planar metamorphic mechanisms for low-voltage circuit breakers," *Int. J. Mechanics Based Des. Structures Mach.*, vol. 40, pp. 453–468, 2012.

[12] D. Rosenberg, "Designing for uncertainty: Novel shapes and behaviors using scissor-pair transformable structures," Ph.D. dissertation, Massachusetts Institute of Technology. Dept. of Architecture, 77 Massachusetts Avenue - Cambridge, MA 02139, Massachusetts Inst. Technol., 2009.

[13] E. Vouga, M. Höbinger, J. Wallner, and H. Pottmann, "Design of self-supporting surfaces," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 87:1–87:11, Jul. 2012.

[14] D. Panozzo, P. Block, and O. Sorkine-Hornung, "Designing unreinforced masonry models," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 91:1–91:12, Jul. 2013.

[15] P. Song, C.-W. Fu, P. Goswami, J. Zheng, N. J. Mitra, and D. Cohen-Or, "Reciprocal frame structures made easy," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 94:1–94:13, Jul. 2013.

[16] T. Tachi, "Interactive freeform design of tensegrity," in *Proc. Adv. Archit. Geometry*, 2013, pp. 259–268.

[17] H. Zimmer and L. Kobbelt, "Zometool rationalization of freeform surfaces," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 10, pp. 1461–1473, Oct. 2014.

[18] T.-V. Le-Nguyen, K.-L. Low, C. Ruiz Jr., and S. N. Le, "Automatic paper sliceform design from 3D solid models," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 11, pp. 1795–1807, Nov. 2013.

[19] N. J. Mitra, Y.-L. Yang, D.-M. Yan, W. Li, and M. Agrawala, "Illustrating how mechanical assemblies work," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 58:1–58:12, Jul. 2010.

[20] L. Zhu, W. Xu, J. Snyder, Y. Liu, G. Wang, and B. Guo, "Motion-guided mechanical toy modeling," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 127:1–127:10, Nov. 2012.

[21] S. Coros, B. Thomaszewski, G. Noris, S. Sueda, M. Forberg, R. W. Sumner, W. Matusik, and B. Bickel, "Computational design of mechanical characters," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 83:1–83:12, Jul. 2013.

[22] B. Thomaszewski, S. Coros, D. Gauge, V. Megaro, E. Grinspun, and M. Gross, "Computational design of linkage-based characters," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 64:1–64:9, Jul. 2014.

[23] D. Ceylan, W. Li, N. J. Mitra, M. Agrawala, and M. Pauly, "Designing and fabricating mechanical automata from mocap sequences," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 186:1–186:11, Nov. 2013.

[24] A. Schulz, A. Shamir, D. I. W. Levin, P. Sitthi-amorn, and W. Matusik, "Design and fabrication by example," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 62:1–62:11, Jul. 2014.

[25] L. Luo, I. Baran, S. Rusinkiewicz, and W. Matusik, "Chopper: Partitioning models into 3D-printable parts," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 129:1–129:9, Nov. 2012.

[26] W. Wang, T. Y. Wang, Z. Yang, L. Liu, X. Tong, W. Tong, J. Deng, F. Chen, and X. Liu, "Cost-effective printing of 3D objects with skin-frame structures," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 177:1–177:10, Nov. 2013.

[27] B. Bickel, M. Bächer, M. A. Otaduy, H. R. Lee, H. Pfister, M. Gross, and W. Matusik, "Design and fabrication of materials with desired deformation behavior," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 63:1–63:10, Jul. 2010.

[28] M. Skouras, B. Thomaszewski, S. Coros, B. Bickel, and M. Gross, "Computational design of actuated deformable characters," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 82:1–82:10, Jul. 2013.

[29] M. Bächer, B. Bickel, D. L. James, and H. Pfister, "Fabricating articulated characters from skinned meshes," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 47:1–47:9, Jul. 2012.

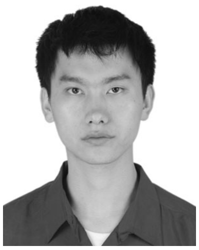
[30] J. Calì, D. A. Calian, C. Amati, R. Kleinberger, A. Steed, J. Kautz, and T. Weyrich, "3D-printing of non-assembly, articulated models," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 130:1–130:8, Nov. 2012.

[31] Y. Zhou, S. Sueda, W. Matusik, and A. Shamir, "Boxelization: Folding 3D objects into boxes," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 71:1–71:8, Jul. 2014.

[32] M. Botsch, M. Pauly, M. Gross, and L. Kobbelt, "Primo: Coupled prisms for intuitive surface modeling," in *Proc. 4th Eurographics Symp. Geometry Process.*, 2006, pp. 11–20

[33] T. B. Sebastian, P. N. Klein, and B. B. Kimia, "On aligning curves," *IEEE Trans. Pattern Recognit. Mach. Intell.*, vol. 25, no. 1, pp. 116–125, Jan. 2003.

[34] N. Umetani, T. Igarashi, and N. J. Mitra, "Guided exploration of physically valid shapes for furniture design," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 86:1–86:11, Jul. 2012.


**Ran Zhang** received the BSc degree from Tianjin University in 2011 and the MSc degree from the University of Science and Technology of China in 2014. He is currently working towards the PhD degree in IST Austria. His research interests include digital fabrication and mechanical structure design.



**Shiwei Wang** received the BSc degree in 2013 from Zhejiang University. He is currently a master's student at the University of Science and Technology of China.



**Xuejin Chen** received the BSc degree in 2003 and the PhD degree in 2008 from the University of Science and Technology of China (USTC). She worked in Microsoft Research Asia (MSRA) as an joint PhD student between USTC and MSRA from 2004 to 2008. From 2008 to 2010, She conducted research as a postdoctoral scholar in the Department of Computer Science at Yale University. In August 2010, She joined USTC as an associate professor. Her research interests include 3D modeling and geometry processing.

**Chao Ding** received the BSc degree in 2013 from the University of Science and Technology of China. He is currently a master's student at the University of Science and Technology of China.

**Luo Jiang** received the BSc degree in 2013 from the Huazhong University of Science and Technology, China. He is currently working towards the PhD degree at the University of Science and Technology of China.

**Jie Zhou** received the BSc degree from the University of Science and Technology of China, in 2012. He is currently working towards the PhD degree at the University of Science and Technology of China. His research interests include indoor scene analysis and 3D modeling.

**Ligang Liu** received the BSc degree in 1996 and the PhD degree in 2001 from Zhejiang University, China. He is a professor at the University of Science and Technology of China. Between 2001 and 2004, he was at Microsoft Research Asia. Then he was at Zhejiang University during 2004 and 2012. He paid an academic visit to Harvard University during 2009 and 2011. His research interests include geometric processing and image processing. His research works could be found at his research website: http://staff.ustc.edu.cn/ lgliu