Department of Computer Science
George Mason University Technical Reports

4400 University Drive MS#4A5
Fairfax, VA 22030-4444 USA
http://cs.gmu.edu/ 703-993-1530

Folding and Unfolding Origami Tessellation by Reusing Folding Path

Zhonghua Xi
zxi@gmu.edu

Jyh-Ming Lien
jmlien@cs.gmu.edu

Technical Report GMU-CS-TR-2015-2

Abstract

Recent advances in robotics engineering have enabled the realization of self-folding machines. Rigid origami is usually used as the underlying model for the self-folding machines whose surface remains rigid during folding except at joints. A key issue in designing rigid origami is *foldability* that concerns about finding folding steps from a flat sheet of crease pattern to a desired folded state. Although recent computational methods allow rapid simulation of folding process of certain rigid origamis, these methods can fail even when the input crease pattern is extremely simple. In this paper, we take on the challenge of planning folding and unfolding motion of origami tessellations, which are composed of repetitive crease patterns. The number of crease lines of a tessellation is usually large, thus searching in such high dimensional configuration space with the requirement of maintaining origami rigidity is nontrivial. We propose a motion planner that takes symmetry into consideration and reuses folding path found on the *essential crease pattern*. Both of these strategies enable us to fold large origami tessellation much more efficiently than the existing methods. Our experimental results show that the proposed method successfully folds several types of rigid origami tessellations that the existing methods fail to fold.

1 Introduction

Rigid origami has been a fundamental model in many self-folding machines [1] that are usually composed of mechanical linkage of flat rigid sheets joined by hinges, such as the micro-thick folding actuators [2]. In the past, people have enjoyed many practical uses of rigid origami, ranging from folding maps and airbags to packing large solar panel arrays for space satellites and folding space telescope. In the near future, rigid origami

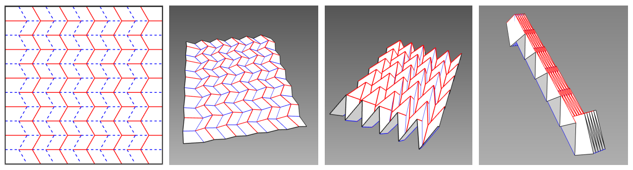


Figure 1: Folding process of a 11×11 Miura crease pattern (DOF = 220) produced by the motion planner proposed in this paper.

will take the form of self-folding machines and provide much broader applications, such as in minimally invasive surgery, where there is a need for very small devices that can be deployed inside the body to manipulate tissue [3]. Examples that illustrate the ability of transforming rigid origami from a shape to another can be found in Fig. 1 and Fig. 7, where a large flat sheet can be folded into a compact stick or to a tube.

A key issue in designing rigid origami is *foldability* that determines if one can fold a given origami form one state to another. Researchers in computational origami have attempted to simulate or plan the folding motion [4, 5, 6, 7, 8]. These existing methods, however, are known to be restricted. For example, the work by Miyazaki et al. [6] only allows bending, folding-up, and tucking-in motions. Balkcom's method [9] cannot guarantee the correct mountain-valley assignment for each crease. The well-known Rigid Origami Simulator by Tachi [8] may sometimes produce motion with self-intersection and can be trapped in a local minimum. One of the main difficulties of planning origami folding motion comes from its highly constrained folding motion in high dimensional configuration space. For example, there are 100 closed-chain constraints in the 11×11 Miura origami shown in Fig. 1. These constraints make most (if not all) existing motion planners impractical, especially for folding large origami tessellations.

Moreover, it is known to the community that given a crease pattern and a rigid goal configuration, the existence of *continuous rigid folding motion* is not guaranteed in general [10]. Unfortunately, there is no known criteria for determining whether a crease pattern or its tessellation can be folded between two rigid configurations without violating the rigidity constraint. In practice, when a crease pattern is designed, it usually requires its designer to create a physical copy to verify that a rigid folding motion does exist to bring the crease pattern to a rigid goal configuration. This process can often be costly and time consuming.

This paper models rigid origami as a *kinematic system with closure constraints*. Our ideas for addressing both of these rigid foldability issues include: *adaptive randomized search* and *folding path reuse*. Specifically, we propose a deformation bounded folding planner (in Section 4) that can ensure the rigidity of the origami during continuous folding motions; such planning has not been achieved before in the community. Given a tessellation formed with repetitive crease patterns, we further take advantage of its symmetry to reduce the degrees of freedom (DOF). Our experimental results show strong evidences that this strategy can significantly speed up the computation (in Section 5). We further propose the idea of *essential crease pattern* in Section 5.2. Fig. 1 shows a folding sequence of a 11×11 Miura origami (220 DOF, with $<1\%$ deformation) found by the proposed method within 1.7 seconds[1]. Examples and results of folding larger tessellations can be found in Section 5.

Our planner requires the crease pattern for computing the folding map. In Section 6 we propose a novel algorithm to obtain the crease pattern of a rigid folded origami from an unknown crease pattern. This allows the input of our system from crease patterns extended to rigid origamis in arbitrary configurations.

2 Related Work

Planning under closure constraints. There have been many methods proposed to plan motion for articulated robots under closed-chain constraints [11, 12, 13, 14]. Interestingly, we see many similar ideas used in both closed-chain systems and origami folding. For example, gradient decent was used by [8] for rigid origami simulation and by [11] for generating valid configuration of a closed-chain system. Another example is inverse kinematics, which plays the central role both in Balkcom's simulator [9] and in constructing the so-called *kinematic roadmap* [13, 15] for capturing the topology of free configuration space. Tang et al. [16] proposed an efficient sampling-based planner for spatially constrained systems. By sampling in the reachable distance space in which all configurations lie in the set of constraint-satisfying subspaces and using a local planner, they can significantly reduce the computation time for finding a path.

Planning and simulating origami motion. Miyazaki et al. [6] simulated origami folding by a sequence of simple folding steps, including bending, tucking in, and folding up in 1996. It is easy to reconstruct an animation from a sheet of paper to the final model. However, the simplicity of folding steps limits the types of origami models that could be represented in the system. Consequently, this method is not suitable for many complex origami models whose folding process cannot be represented as simple folding steps such as the Miura pattern shown in Fig. 5(a). Song et al. [17] presented a PRM based framework for studying folding motion. However, their kinematic representation of origami is a *tree-structure model* whose folding angle of each crease line is independent of other crease lines. Although a tree-structure model greatly simplifies the folding map that can be easily defined along the path from base to each face, this model is not applicable to represent the majority of origamis, such as the one shown in Fig. 5(a), due to their *closure constraints*. Balkcom [9] proposed a simulation method based on the ideas of virtual cutting and combination of forward and inverse kinematics using a rigid origami model. Although this approach is computational efficient, the correctness of mountain-valley assignment for each crease is not guaranteed, i.e., a mountain fold can become a valley fold or vice versa. Tachi [8] proposed an interactive simulator for rigid origami model (known as Rigid Origami Simulator (ROS)) which generates folding motion of origami by calculating the trajectory by projection to the constrained space based on rigid origami model, global self-intersection avoidance and stacking order problems are not considered in his work. An et al. [2] proposed a new type of self-reconfiguration system called self-folding sheet. They first construct the corresponding folded state for a given crease pattern and angle assignment then continuously unfold the paper using local repulsive energies (via a modification of ROS [8]). By reversing the unfolding sequence, they obtained the path starting from a flat sheet and ending with the desired folded state. Akitaya et al. [18] proposed a method for generating folding sequences of origami, however, their system can only handle flat-foldable origami. More recently, Xi and Lien [19] proposed a randomized search algorithm via nonlinear optimization to find the intermediate folding steps which guarantees self-intersection free, however, the motion it found can lead to arbitrary deformation.

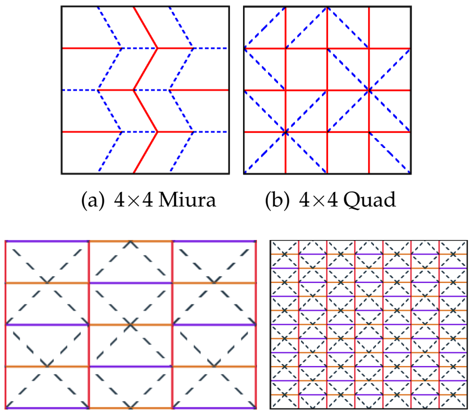
[1]All timing data reported in this paper are collected on a 2012 Macbook Pro laptop with a 2.9GHz Intel Core i7 CPU and 16GB RAM.

3 Preliminaries: Rigid Origami Model

3.1 Crease Pattern

In this paper, we use crease pattern, a straight-edged graph embedded in the plane, to represent the rigid origami model. Fig. 2 shows the crease patterns of the origami tessellations used in our experiments (in Section 5). An edge of this graph correspond to the location of a crease line in an unfolded sheet. A crease line can be either *mountain folded* or *valley folded*. A mountain fold forms a convex crease at top with both sides folded down. On the other hand, a valley fold forms a concave crease.

Real & Virtual Vertices Vertices in crease pattern can be categorized into two groups: real vertices and virtual vertices. Vertices on the boundary of a pattern are considered as virtual vertices and they cannot act as *witness vertices* for the purpose of computing *folding map*. Using the folding map of a given configuration, we can *instantaneously* fold a crease pattern to a folded shape. All other vertices are considered as real vertices. For example, vertices v_1, v_2, v_3 and v_4 are the only real vertices in Fig 5(a) and all the other vertices are virtual vertices.


(a) 4×4 Miura (b) 4×4 Quad

(c) 4×6 Waterbomb (d) 12×14 Waterbomb

Figure 2: Crease patterns used in the experiments. The mountain creases are shown as solid lines in red, valley creases are show as dashed lines in blue.

Crease Lines We use $l_{(i,j)}$ to denote the crease line that connects vertex v_i and v_j in which at least one vertex should be real. Boundary edges in the crease pattern are not considered as crease lines. Each crease line $l_{(i,j)}$ is associated with a *plane angle* $\alpha_{(i,j)}$ which is the angle between $\overrightarrow{v_i v_j}$ and $[1, 0]^T$ (x-axis) and a *folding angle* $\rho_{(i,j)}$ which equals to π minus the dihedral angle between two faces sharing the crease line $l_{(i,j)}$. The value of ρ is bounded in [-π, π] to avoid adjacent faces penetrating each other.

Faces We use $F_{(i,j,...)}$ to refer to a face in the crease pattern, where $\{v_i, v_j, ...\}$ are its vertices. The crease line $l_{(i,j)}$ belongs to two faces $F_{(i,j,...)}$ and $F_{(j,i,...)}$.

For a non-triangular crease pattern, we will triangulate it first, newly added diagonals are called *virtual edges* whose folding angles should always be zero otherwise the panel will be bended.

3.2 Configuration

We use the folding angles of all crease lines to represent the configuration of an origami model. For an origami with n crease lines, its configuration is represented as $\mathcal{C} = [\rho_{(i_1,j_1)}, \rho_{(i_2,j_2)}, \cdots, \rho_{(i_n,j_n)}]^T$. Given a configuration $\mathcal{C}$, we can classify $\mathcal{C}$ according to its *foldability* and *feasibility*.

Foldability For a real vertex v_i in a multi-vertex crease pattern, let A_i be the 4×4 matrix which translates a point in $\Re^3$ by v_i. Let $B_{(i,j)}$ be the 4×4 matrix in homogeneous coordinates which rotates around z-axis for plane angle $\alpha_{(i,j)}$, and let $C_{(i,j)}$ be the 4×4 matrix in homogeneous coordinates which rotates around x-axis for folding angle $\rho_{(i,j)}$. Then the 4×4 folding matrix of counter-clock-wisely crossing crease line $l_{(i,j)}$ with witness vertex v_i is $\chi_{((i,j),i)} = A_i B_{(i,j)} C_{(i,j)} B^{-1}_{(i,j)} A^{-1}_i$.

Let $\{l_{(i,j_1)}, l_{(i,j_2)}, \dots , l_{(i,j_{c_i})}\}$ be the crease lines incident to v_i, ordered by their plane angles $\alpha_{(i,j)}$, where c_i is the number of crease lines incident to v_i. If we pick $F_{(i,j_{c_i},...)}$ as F_0 and fix it in the xy-plane, we define the local foldability matrix for real vertex v_i as $L(v_i) = \prod_{t=1}^{c_i} \chi_{((i,j_t),i)}$. Finally, the necessary condition of foldability is:

$$L(v_i) = I, \forall v_i \tag{1}$$

This condition for multi-vertex rigid origami was first discovered by Balcastro and Hull in 2002 [7].

Feasibility There are several properties that an origami rigid folding should have: (1) unstretchable, (2) flat (planar) for all faces, and (3) free of self intersection. A foldable configuration only guarantees the first two properties. In order to check if $\mathcal{C}$ is free of self intersection, we need a folding map for each face. A *folding map* is a function that maps a point in $\Re^2$ to the corresponding point of folded state in $\Re^3$ for a given foldable configuration.

4 Folding via Adaptive Randomized Search

Searching for a valid folding motion of an origami tessellation is difficult because of its highly constrained nature and high dimensional configuration space. In particular, there are n closed-chain constraints for an origami with n real vertices. These constraints make most (if not all) existing probabilistic motion planners impractical. In [20] we show that for rigid origami with closure constraint, the portion of free space is near to zero even certain amount of deformation is allowed.

In this paper, we extend FROCC [19] which uses an adaptive randomized search with nonlinear optimization. FROCC samples a random configuration $\mathcal{C}_{rand}$ around current configuration $\mathcal{C}_\tau$ and pushes $\mathcal{C}_{rand}$ to a foldable configuration $\mathcal{C}_\Delta$ via nonlinear optimization (NLOpt). If $\mathcal{C}_\Delta$ is feasible and closer to the goal, it then replaces $\mathcal{C}_\tau$ with $\mathcal{C}_\Delta$ and keep doing so until goal is reached. FROCC works well in practice, however, it also has several issues that we are trying to address in this paper.

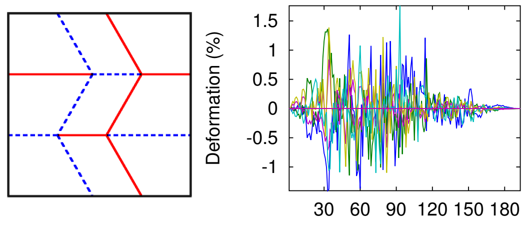
Objective Function Intuitively, because each real vertex of a foldable configuration must satisfy the constraint in Eq. (1), for a given real vertex v_i, we want the local foldability matrix $L(v_i)$ to be as close to an identity matrix I as possible. However, the objective function $F(\mathcal{C}) = \sum_i |L(v_i) - I|$ used by FROCC could be easily trapped at local minima. In this paper, we updated the objective function to Eq. (2).

$$F(\mathcal{C}) = \max_i |L(v_i) - I| \tag{2}$$

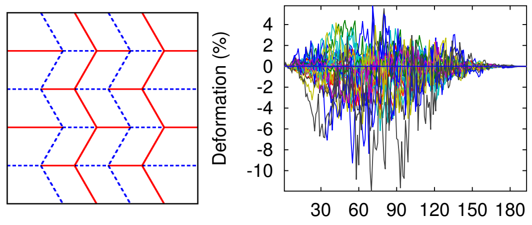
If $L(v_i) \neq I$, deformation will be introduced, in Eq. (2) we try to minimize the maximum deformation which works better than the original one.

Deformation Bounded Search (DBS) During randomized search, NLopt finds an optimal configuration C around $\mathcal{C}_\tau$, but the value of $F(C)$ in Eq. (2) may be none zero. This is because local-foldable configuration might not exist around $\mathcal{C}_\tau$ or NLopt is not able to find it within given iterations. Consequently, none-zero $F(C)$ leads to deformation in folded oragami. However, directly bounding $F(C)$ [19] does not give us a quantitative rigidity measure. To illustrate this, in Fig. 3, we show deformation measured in terms of the stretch and shrinkage of edges given that $F(C) < 0.1$. We can see that with the increase of size of the crease pattern, though their folding paths still look identical (with the naked eye), the edge deformation is quite dramatic (increased from $\approx$1.5% in 3×3 Miura to $\approx$10% in 5×5 Miura).

Thus, we propose a deformation bounded search (DBS) that checks the maximum amount of deformation measured by the change of edge length including virtual edges which is defined as $(||e_{folded}|| - ||e_{org}||)/||e_{org}||$. In DBS, we use the same objective function in Eq. (2) but only accept configurations that are within the deformation bound given by the user. The folding path found by DBS is guaranteed to be deformation bounded and self-intersection free, which has not been achieved before in the community. Folding paths for the 3x3 Miura crease pattern with different deformation bounds found by the proposed method are shown in Fig. 4. We can see that there are huge differences between assigned folding angles and measured ones due to deformation. Some virtual edges have more than 15° folding angles which means some panels have been bended in order to reduce the deformation which is not tolerable in practice. Theoretically, they should be the same if the configuration is foldable and the origami will be deformation free. Within 1% deformation, they become identical (see Fig. 4(d) and Fig. 4(c)). Note that, instead of simply filtering out configurations with large deformation, we have also tried incorporating maximum deformation in the objective function, but the optimization process is often trapped in local minima due to the higher complexity of the objective function.



(a) 3×3 Miura

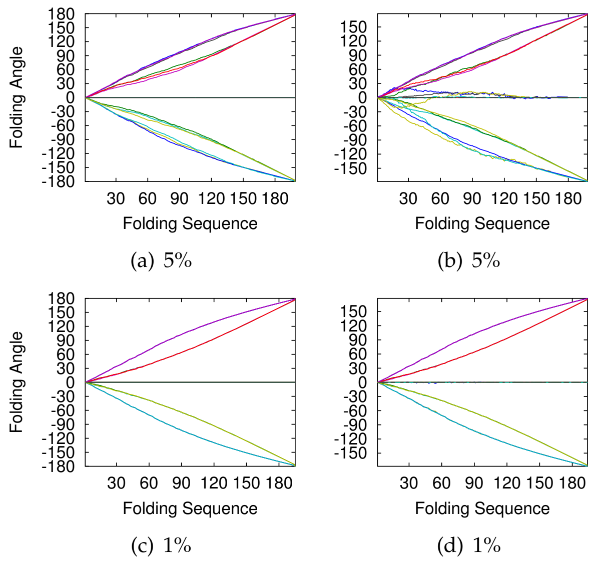


(b) 5×5 Miura

Figure 3: Edge deformations during the folding process by requiring $F(C) < 0.1$ for Miura crease patterns. **Left:** Crease patterns. **Right:** Edge deformations.

Running times against various deformation upper bounds are shown in Table 1. As we can see, when lower the deformation tolerance, our method takes longer time to find a valid path which is expected. The main computation time is from the increase in the number of iterations needed to find an accurate enough foldable configuration in NLopt.

Large Origami Tessellation Though FROCC works well in lower dimensional space (<10), with the increasing of the complexity of the crease pattern (e.g., large origami tessellation), it becomes harder for FROCC to find a valid path. Detailed discussion regarding this issue will be given in Section 5.



(a) 5% (b) 5%

(c) 1% (d) 1%

Figure 4: Assigned and measured folding angles under different deformation upper bound for folding a 3x3 Miura crease pattern. **Left:** Assigned folding angles computed by NLopt. **Right:** Measured folding angles. These are the folding angles measured on the origami after folded with the assigned angles.

Table 1: Planning Time v.s. Deformation Upper Bound

Model	DOF	DUB	MI	Time (s)
4×4 Miura	24	10%	100	0.14
		5%	100	0.18
		2%	1000	1.51
		1%	1000	3.17
4×4 Quad	40	10%	200	0.34
		5%	200	0.49
		2%	1000	2.42
		1%	2000	12.41

DUB=Deformation Upper Bound, MI=Maximum Iteration for NLopt. The maximum iteration for NLopt is set to the minimum value that could make the planner achieve over 90% success rate.

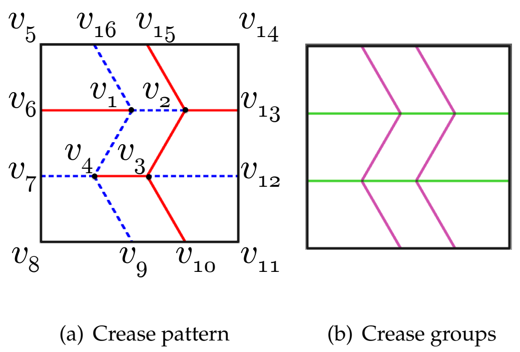
5 Folding Large Origami Tessellation

A tessellation is a type of crease pattern that can usually be viewed as an arrangement of smaller repetitive crease patterns. As a result, the degrees of freedom of a tessellation is usually very large (758 for a 12×22 Waterbomb and 1680 for a 24×24 Miura fold). Finding valid folding motion for such as tessellation can be extremely time consuming. In order to speed up the motion planner, we propose the idea of *crease group* and *essential vertex* by exploiting symmetry in the tessellation in Section 5.1.

Computation reuse is a widely used technique to improve the performance of a robotic system [21, 22]. In Section 5.2, we propose the idea of reusing folding path found on the *essential crease pattern* to fold large origami tessellation.

5.1 Crease Group and Essential Vertex

Given a large crease pattern (tessellation), crease lines can be gathered into groups naturally due to symmetry property. We say that a set of crease lines are in one crease group if the absolute value of their folding angles trace out the same folding trajectory. Given the crease groups, we define *essential vertices* as a set of real vertices whose incident crease lines collectively *cover* all the crease groups. The smallest essential vertices can be found by solving the *set covering problem*. An example of crease groups is shown in Fig. 5, in which crease lines belong to the same crease group are shown in the same color. From Fig. 5 we can see that the 3×3 Miura crease pattern has only two crease groups: all vertical crease lines are in one group and all horizontal crease lines are in another group, even though they have different type (mountain fold v.s. valley fold). Since any of the real vertices can cover all the crease groups, the 3×3 Miura crease pattern has only one essential vertex which could be v_1 or v_2 or v_3 or v_4.



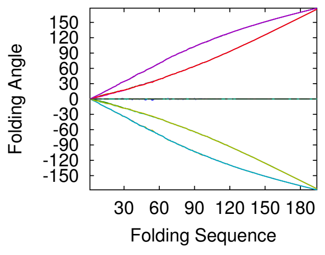
(a) Crease pattern (b) Crease groups

Figure 5: Crease groups of a 3×3 Miura crease pattern. Crease lines belong to the same crease group are shown in the same color.

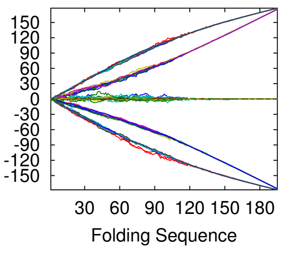
By gathering crease lines from a large crease pattern into crease groups, the DOF of the origami can be reduced from the number of crease lines to the number of crease groups. Moreover, by identifying essential vertices, we only need to check the local foldability (Eq. (2)) on *essential vertices*, a much smaller subset of real vertices than the number of all the real vertices. Table 2 reports the size of crease groups and essential vertices of 6 crease patterns. As we can also see in Table 2, using symmetry and essential vertex significantly reduces the computation time for finding a valid folding motion. We also tested the running with and without collision detection. From Table 2 we can see when we use full DOF for planning, the majority of the time is spent on finding valid configuration, collision detection takes only about 2% of the running time for folding the 5 × 5 Miura crease pattern. However, when we use symmetry property and essential vertex, the running time reduced significantly, collision detection (with almost the same amount of computation) then dominates the running time which takes about 83% on average.

5.2 Reusing Folding Path

Given a crease pattern (tessellation), if this crease pattern is rigid foldable, it is expected that the folding angles of all crease lines in the same crease group remains identical even when planning is done using the full DOF. Further more, the trajectories are expected remain identical when folding a smaller but same type tessellation as shown in Fig. 6.



(a) 3×3 Miura DOF=12



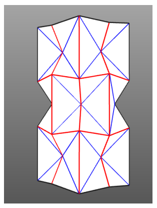
(b) 5×5 Miura DOF=40

Figure 6: Folding paths found without using symmetry information.

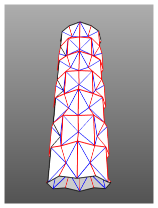
This give us the idea of reusing the folding path found on smaller crease pattern to fold the large one which is much more computational efficient. With the crease group and essential vertex in mind, here we define the concept of *essential crease pattern* which is the smallest crease pattern that contains all essential vertices as real vertices. We first find a folding path whose deformation is enough low on the essential crease pattern that could satisfy the deformation criteria when folding the larger pattern with it since the deformation will be amplified with the increase of size of the crease pattern. Then the folding path is applied to the original crease pattern.

An example of reusing folding path for another rigid-foldable crease pattern Waterbomb is shown in Fig. 7. The configuration of the folded tube is from [23] in which the authors showed that the tube is in fact continuous rigid foldable and our method confirms that the folding process is indeed rigid. As we can see from Fig. 7, though the deformation by reusing folding path is about 10x larger than the one on the essential crease pattern, it is still within the user given deformation upper bound (1%).

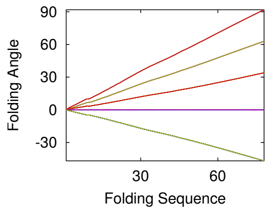
Note that, when reusing folding path, the deformation will be scaled up according to crease pattern size. We observe that this increasing in deformation is much more



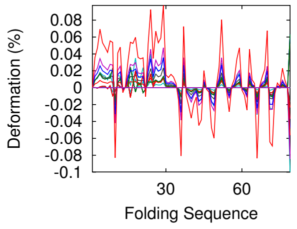
(a) 4×6



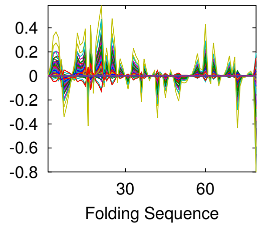
(b) 12×14



(c) 4×6



(d) 4×6



(e) 12×14

Figure 7: Reusing folding path. (a) Folded shape of 4×6 waterbomb crease pattern shown in Fig. 2(c). (b) Folded shape of 12×14 waterbomb crease pattern shown in Fig. 2(d) by reusing folding path (c). (c) Folding path found on the essential crease pattern Fig. 2(c). (d) Edge deformation when folding Fig. 2(c). (e) Edge deformation when folding Fig. 2(d) by reusing the folding path (c).

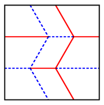
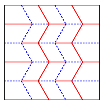
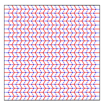
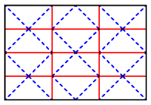
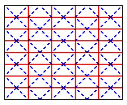
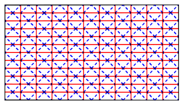
dramatic for non-rigid-foldable crease pattern. An example of non-rigid-foldable crease pattern is shown in Fig. 8. Folding a 24×24 Quad crease pattern shown in Fig. 8(a) by reusing a 0.1% deformation upper bound folding path found on its essential crease pattern shown in Fig. 8(b) gives us a large ($>$ 400%) deformation shown in Fig. 8(g) due to folding map inconsistency. Thus, we can determine the rigid-foldability of a given symmetric crease pattern by reusing path found for its essential crease pattern.

Proposition 5.1 *If a crease pattern with a goal configuration is symmetrically rigid foldable, then the folding process of that crease pattern is deformation bounded when it is folded by reusing an arbitrary deformation bounded folding path found on its essential crease pattern.*

6 Unfolding Rigid Folded Shapes

Note that the folding map is applied on the crease pattern, thinking about the scenario that given a 3D shape that was rigid folded from some unknown crease pattern. Without knowing the crease pattern beforehand, our planner is not able to fold or unfold the origami. To address this problem, we propose a novel algorithm to

Table 2: Path Planning Time using Symmetry.

Model	RV/EV	SYM	EV	DOF	MI	Time (sec)	CD (%)
3×3 Miura	4/1	×	×	12	25	0.037	27.03
		∘	×	2	5	0.016	56.25
		∘	∘	2	5	0.014	64.29
5×5 Miura	9/1	×	×	40	500	1.681	2.02
		∘	×	2	5	0.098	81.63
		∘	∘	2	5	0.082	85.37
24×24 Miura	529/1	×	×	1680	N/A*	N/A*	N/A*
		∘	×	2	100	35.278	78.32
		∘	∘	2	100	32.402	99.51
4×6 Waterbomb	15/3	×	×	50	5	0.040	77.50
		∘	×	4	5	0.037	81.08
		∘	∘	4	5	0.034	88.24
8×10 Waterbomb	63/3	×	×	182	5	0.406	79.80
		∘	×	4	5	0.332	90.36
		∘	∘	4	5	0.310	96.45
12×22 Waterbomb	231/3	×	×	758	5000	499.048	11.27
		∘	×	4	5	3.893	91.75
		∘	∘	4	5	3.160	97.59

Note that the running time were obtained under 5% deformation upper bound. RV=Real Vertex, EV=Essential Vertex, SYM=Symmetry, MI=Maximum Iteration, CD=Time cost for Collision Detection. The symbols ∘ and ×, in the columns of SYM and EV, indicate if symmetry and essential vertex are used or not. *The planner failed to find a valid path within the time limit due to high DOF.

unfold a rigid folded 3D shape to its crease pattern as shown in Algorithm 1, which can unfold the 3D shape *instantaneously* while the intermediate motion remains unknown. The target folding angle for each crease line can be measured from the folded shape. An example of unfolded Yoshimura crease pattern is shown in Fig. 9(f) which is unfolded from a half-folded shape shown in Fig. 9(d).

Proposition 6.1 *Any rigid folded shape can be flattened to its crease pattern instantaneously.*

7 Compare with Existing Works

Although there have been several existing works on simulating or planning motion of rigid origami [6, 9, 2], most of these works are only applicable to specific type of rigid origami. Tachi's Rigid Origami Simulator (ROS) [8] provides the most general solution so far and is the only publicly available software the we are aware of. Consequently, we have tested ROS extensively using the crease patterns shown in the paper. However, we found that it is difficult to provide a meaningful comparison to our methods due to that both approaches focus on different objectives. The main objective of this paper is to find rigid folding path from one configuration to

Algorithm 1 Unfold Rigid Folded Shape to Crease Pattern

```
Input: Rigid folded shape $\mathcal{S}$ (triangular mesh)
Output: Crease pattern $\mathcal{CP}$ of $\mathcal{S}$
 1: Pick an arbitrary face from $\mathcal{S}$ as $F_0$
 2: Place $F_0$ onto $xy$-plane arbitrarily
 3: $\mathcal{CP} \leftarrow \{F_0\}$
 4: while not all faces of $\mathcal{S}$ were attached to $\mathcal{CP}$ do
 5:     Pick a face $F$ from $\mathcal{S}$, s.t. at least one edge of $F$ was attached to $\mathcal{CP}$
 6:     if One edge of $F$ was attached to $\mathcal{CP}$ then
 7:         Attach $F$ to $\mathcal{CP}$ without overlapping $\mathcal{CP}$   ▷ Two ways to attach $F$ to $\mathcal{CP}$, one of them causes overlapping.
 8:     else if Two edges of $F$ was attached to $\mathcal{CP}$ then
 9:         Attach the last edge of $F$ to $\mathcal{CP}$
10:     else
11:         Do nothing   ▷ All three edges of $F$ were attached to $\mathcal{CP}$, position of $F$ is determined.
12:     end if
13:     $\mathcal{CP} \leftarrow \mathcal{CP} \cup F_k$
14: end while
15: return $\mathcal{CP}$
```

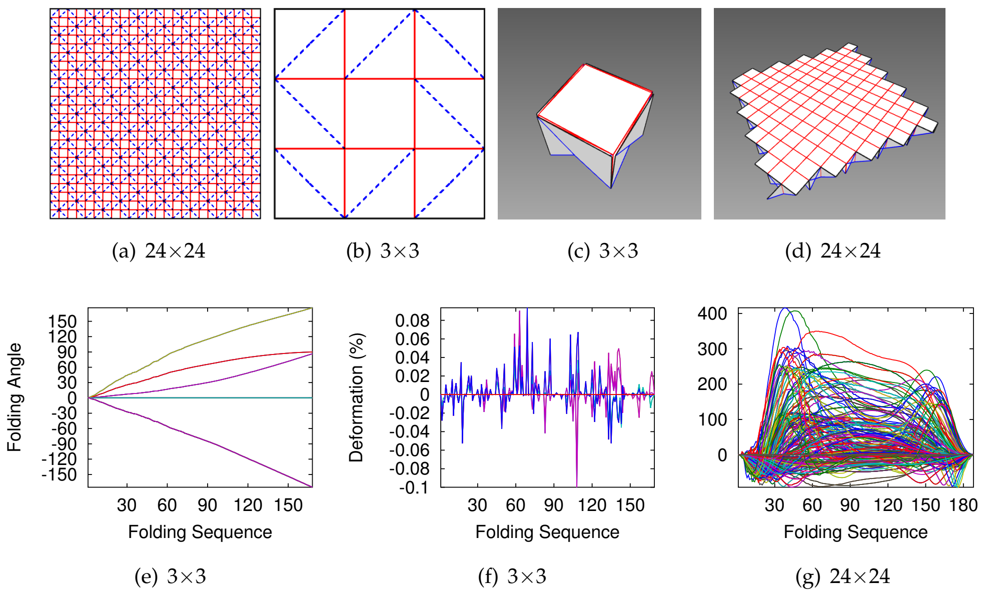


Figure 8: Non-rigid-foldable crease pattern. (a) 24×24 Quad crease pattern. (b) Essential crease pattern of (a). (d) Folded shape of (a). (c) Folded shape of (b). (e) Folding path found on (b). (f) Edge deformation when folding (b). (g) Edge deformation when folding (a) by reusing folding path (e).

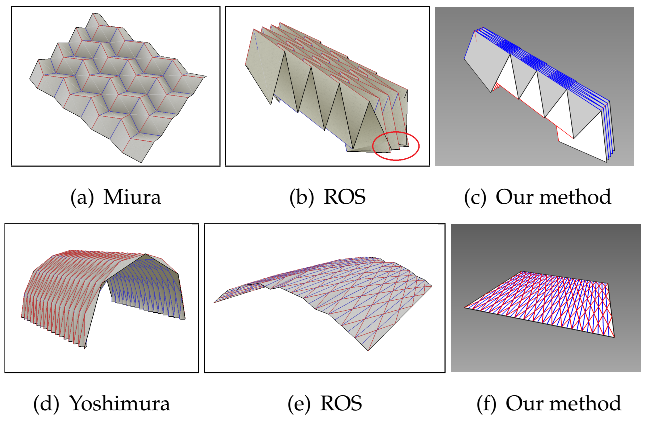


Figure 9: Comparisons to ROS. **Top:** (a) Half-folded state of Miura crease pattern. (b) Maximum folded state from ROS. This configuration found by ROS is not collision free. (c) Folded by our method. **Bottom:** (d) Half-folded state of Yoshimura crease pattern. (e) Maximum unfolded state from ROS. (f) Unfolded by our method.

another while ROS focused on folding a crease patten as much as possible (and usually this means as flat as possible). Moreover, ROS does not guarantee that the folding motion is rigid and collision free. Visual comparisons with results obtained from ROS are shown in Fig. 9. Self-intersection can be found in Fig. 9(b).

8 Conclusions

In this paper, we proposed a randomized approach for planning the motion of rigid origami. We used a nonlinear optimization method to find a valid (deformation bounded and collision free) configuration around a given sample configuration. The experimental results shows that our planner could efficiently and effectively find valid path for various types of rigid origami that existing tools fail to fold. Taking symmetry into consideration and reusing folding path found on the essential crease pattern enable us to fold large origami tessellation efficiently.

The proposed randomized rigid origami folding method is designed to assist the foldability analysis of self-folding origami. Self-folding origami using active-materials usually have many kinematic and dynamic constraints, such as maximum folding angles, and may often requires *multiple folding phases* in order to fold itself to the desired state, see details in [24]. User defined motion criteria can be easily introduced into the proposed framework. For example, our method supports multi-phase folding by given a sequence of valid intermediate configurations $\{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_n\}$.

Limitations and Future Work Even through our preliminary results are encouraging, our method still has much room for improvement and many open questions to be answered. For example, given a crease pattern without a goal configuration, how to determine its crease groups. Given a desired deformation upper bound of a large crease pattern, how can one determine the deformation upper bound required for its essential crease pattern.

References

[1] S. Felton, M. Tolley, E. Demaine, D. Rus, and R. Wood, "A method for building self-folding machines," *Science*, vol. 345, no. 6197, pp. 644–646, 2014.

[2] B. An, N. Benbernou, E. D. Demaine, and D. Rus, "Planning to fold multiple objects from a single self-folding sheet," *Robotica*, vol. 29, no. 1, pp. 87–102, 2011.

[3] L. Swanstrom, M. Whiteford, and Y. Khajanchee, "Developing essential tools to enable transgastric surgery," *Surgical endoscopy*, vol. 22, no. 3, pp. 600–604, 2008.

[4] T. Hull, "On the mathematics of flat origamis," *Congressus numerantium*, pp. 215–224, 1994.

[5] M. Bern and B. Hayes, "The complexity of flat origami," in *Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms*, pp. 175–183, Society for Industrial and Applied Mathematics, 1996.

[6] S. Miyazaki, T. Yasuda, S. Yokoi, and J.-i. Toriwaki, "An origami playing simulator in the virtual space," *Journal of Visualization and Computer Animation*, vol. 7, no. 1, pp. 25–42, 1996.

[7] S.-M. Belcastro and T. Hull, "A mathematical model for non-flat origami," in *Origami3: Proc. the 3rd International Meeting of Origami Mathematics, Science, and Education*, pp. 39–51, 2002.

[8] T. Tachi, "Simulation of rigid origami," in *Origami4: Proceedings of The Fourth International Conference on Origami in Science, Mathematics, and Education*, 2009.

[9] D. J. Balkcom and M. T. Mason, "Robotic origami folding," *The International Journal of Robotics Research*, vol. 27, no. 5, pp. 613–627, 2008.

[10] T. Tachi, "Designing freeform origami tessellations by generalizing resch's patterns," *Journal of Mechanical Design*, vol. 135, no. 11, p. 111006, 2013.

[11] J. H. Yakey, S. M. LaValle, and L. E. Kavraki, "Randomized path planning for linkages with closed kinematic chains," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 951–958, 2001.

[12] J. Cortes, T. Simeon, and J. P. Laumond, "A random loop generator for planning the motions of closed kinematic chains using PRM methods," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 2141–2146, 2002.

[13] L. Han and N. M. Amato, "A kinematics-based probabilistic roadmap method for closed chain systems," in *Robotics:New Directions*, (Natick, MA), pp. 233–246, A K Peters, 2000. Book containts the proceedings of the International Workshop on the Algorithmic Foundations of Robotics (WAFR), Dartmouth, March 2000.

[14] J. Cortes and T. Simeon, "Sampling-based motion planning under kinematic loop-closure constraints," in *Proc. Int. Workshop Alg. Found. Robot.(WAFR)*, 2004. To appear.

[15] D. Xie and N. M. Amato, "A kinematics-based probabilistic roadmap method for high dof closed chain systems," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 1, pp. 473–478, IEEE, 2004.

[16] X. Tang, S. Thomas, P. Coleman, and N. M. Amato, "Reachable distance space: Efficient sampling-based planning for spatially constrained systems," *The international journal of robotics research*, vol. 29, no. 7, pp. 916–934, 2010.

[17] G. Song and N. M. Amato, "A motion-planning approach to folding: From paper craft to protein folding," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 1, pp. 60–71, 2004.

[18] H. A. Akitaya, J. Mitani, Y. Kanamori, and Y. Fukui, "Generating folding sequences from crease patterns of flat-foldable origami," in *ACM SIGGRAPH 2013 Posters*, p. 20, ACM, 2013.

[19] Z. Xi and J.-M. Lien, "Folding rigid origami with closure constraints," in *International Design and Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE)*, (Buffalo, NY), ASME, Aug. 2014.

[20] Z. Xi and J.-M. Lien, "Plan folding motion for rigid origami via discrete domain sampling," Tech. Rep. GMU-CS-TR-2015-4, Department of Computer Science, George Mason University, 4400 University Drive MSN 4A5, Fairfax, VA 22030-4444 USA, 2015.

[21] J.-M. Lien and Y. Lu, "Planning motion in similar environments," in *Proceedings of the Robotics: Science and Systems Conference (RSS)*, (Seattle, Washington), Jun 2009.

[22] D. Berenson, P. Abbeel, and K. Goldberg, "A robot path planning framework that learns from experience," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 3671–3678, IEEE, 2012.

[23] J. Ma and Z. You, "Modelling of the waterbomb origami pattern and its applications," in *International Design and Engineering Technical Conferences*

and Computers and Information in Engineering Conference (IDETC/CIE), (Buffalo, NY), ASME, Aug. 2014.

[24] S. Ahmed, C. Lauff, A. Crivaro, K. McGough, R. Sheridan, M. Frecker, P. von Lockette, Z. Ounaies, T. Simpson, J.-M. Lien, and R. Strzelec, "Multi-field responsive origami structures: Preliminary modeling and experiments," in *Proceedings of the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, August 2013.