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

Plan Folding Motion for Rigid Origami via Discrete Domain Sampling

Zhonghua Xi
zxi@gmu.edu

Jyh-Ming Lien
jmlien@cs.gmu.edu

Technical Report GMU-CS-TR-2015-4

Abstract

Self-folding robot is usually modeled as rigid origami, a class of origami whose entire surface remains rigid during folding except at crease lines. In this work, we focus on finding valid folding motion that brings the origami from the unfolded state continuously to the 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 but with *implicit folding orders*. Moreover, due to the rigidity requirement, the probability of generating a valid configuration via uniform sampling is zero, which greatly hinders that applicability of traditional sampling-based motion planners. We propose a novel sampling strategy that samples in the discrete domain. Our experimental results show that the proposed method could efficiently generate valid configurations. Using those configurations, the planners successfully fold several types of rigid origamis that the existing methods fail to fold and could discover multiple folding paths for Multi-DOF origamis.

1 Introduction

In recent year, we have witnessed the acceleration in the development of *self-folding origami* or *self-folding machines* [1] due to the advances in robotics engineering and material science. These self-folding origami can fold itself into a desired shape via the micro-thick folding actuators [2] or by reacting to various stimuli such as light [3], heat and magnetic fields [4]. Although the development is still in its early stage, there have already been many applications, such as surgical instruments for minimally invasive surgery, where there is a need for very small devices that can be deployed inside the body to manipulate tissue [5].

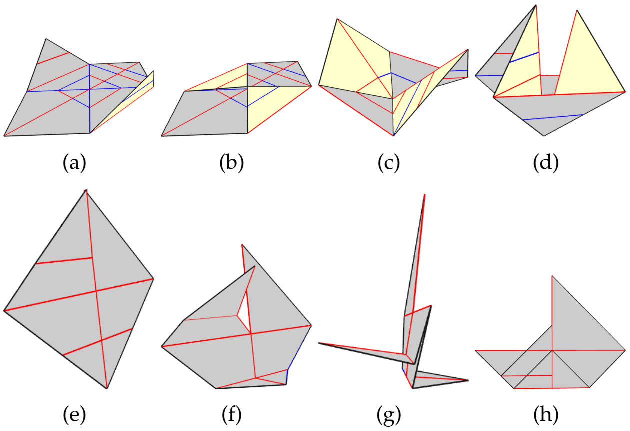


Figure 1: Folding sequence (from (a) to (h)) of a rigid sailboat origami produced by the proposed planner which samples in the discretized configuration space of the sailboat.

Designing self-folding origami that can resume or approximate a single or multiple target shapes requires careful *foldability* analysis. The foldability problem can usually be addressed in two steps. First, does there exist an angle assignment for the crease pattern that maps an unfolded sheet to the target shape? Second, if such a mapping does exist, what is the folding process, i.e., a sequence of angle assignments, that brings the crease pattern from the unfolded state continuously to the folded state. In this paper, our focus is on the second question: planning folding motion of rigid origami. Fig. 1 shows an example of folding motion generated by the motion planner proposed in this paper.

The challenges of planning motions for rigid origami stem from two main sources: the high dimensionality in the search space and highly constrained kinematic system resulted from the rigidity requirement. Consequently, we have experienced great difficulty in applying traditional sampling-based methods such as PRM

[6] and RRT [7] directly. This is due to the fact that the valid configurations of many origami usually form a manifold in lower dimensional space, thus the probability of generating a valid configuration is zero via uniform sampling. This phenomenon can be observed from Waterbomb and Miura origami in Fig. 2, where the valid configurations form 1D curves in the projected 2D configuration spaces.

In this paper, we propose a sampling-based motion planner that generates configurations using only a small set of folding angles, such as those found in the initial and final configurations and some commonly used angles such as $\frac{\pi}{2}$ and π. Given the simplicity of the proposed method, it provides many advantages comparing with a strategy that samples from the continuous space [8, 9]. First, our method can find more valid configurations in shorter sampling time. Second, our method can quickly discover *implicit folding order* that provides critical information to guide the folding process of many origami, such as the sailboat in Fig. 1. It should be noted that, finding the implicit folding order, that requires the crease lines to be folded in a very specific order, can be viewed as the notorious "narrow passage problem" in sampling-based motion planners. Finally, contrary to the existing methods that only report a single folding path, our method can provide multiple folding paths in different homotopic classes (see Fig. 7).

The rest of the paper is organized as the follows. We survey related work on simulating and planning motion for origami in Section 2. Rigid origami model used in our work is provided in Section 3. In Section 4, we describe our method to fold rigid origamis in details. Experimental results and discussions are given in Section 5. Finally, Section 6 concludes the paper.

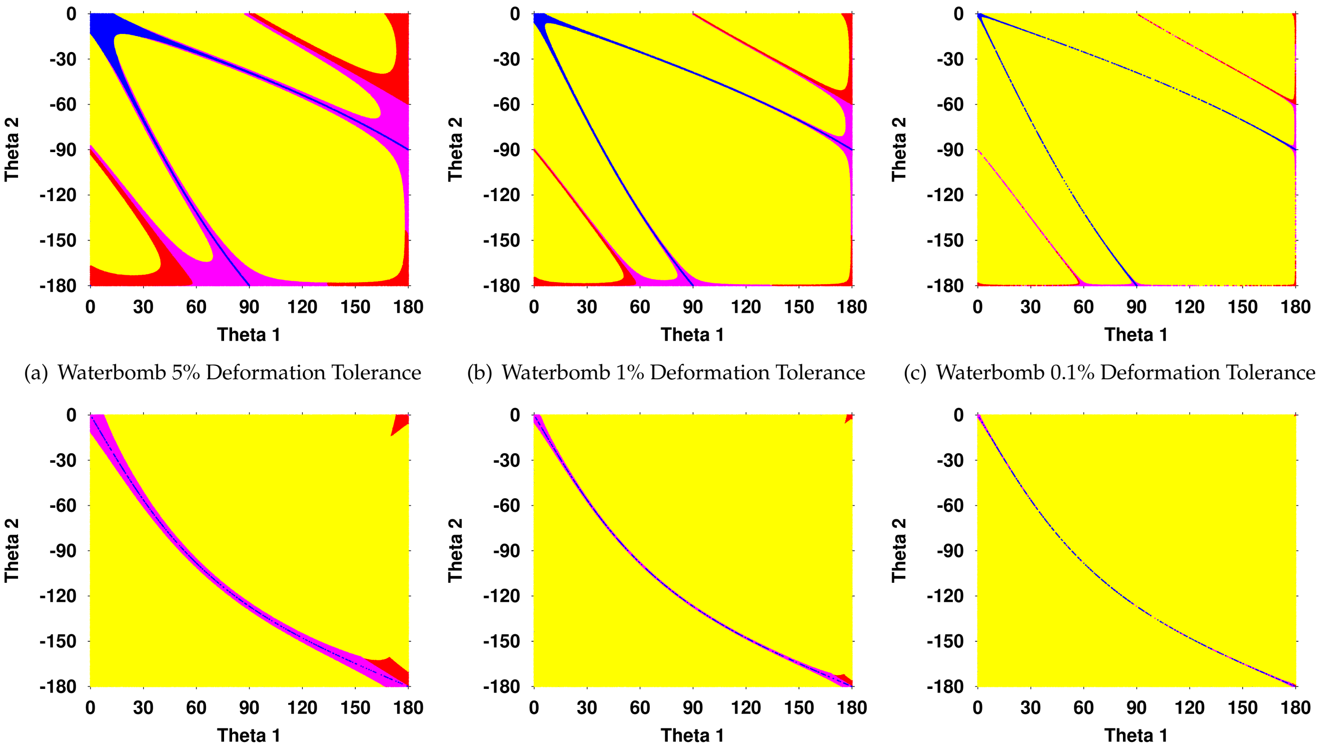
2 Related Works

Although there has been many works on computational origami in the past few decades, few of them focused on planning and simulating origami motion. In this section, we will review some of these works. In 1996, Miyazaki et al. [10] simulates origami folding by a sequence of simple folding steps, including bending, folding up, and tucking in. 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 crease pattern. shown in Fig. 3(c). Song et al. [11] 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 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 the origami, due to their *closure constraints*. Balkcom [12] 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, it cannot guarantee the correct mountain-valley assignment for each crease, i.e., a mountain fold can become a valley fold or vice versa. Recently, 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. [13] proposed a method for generating folding sequences of origami, however, their system can only handle flat-foldable origami. More recently, Xi and Lien [9] proposed an randomized search algorithm via nonlinear optimization (FROCC) to find the intermediate folding steps which guarantees the rigid foldability and self-intersection free.

3 Preliminaries: Rigid Origami Model

3.1 Crease Pattern

We use a crease pattern to represent the rigid origami model. A crease pattern is a straight-edged graph embedded in the plane. 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. Examples of crease pattern are shown in Fig. 3. For a crease pattern with non-triangular faces, we will triangulate it first before planning the motion, newly added diagonals are called *virtual edges* whose folding angles should always be zero.



(a) Waterbomb 5% Deformation Tolerance (b) Waterbomb 1% Deformation Tolerance (c) Waterbomb 0.1% Deformation Tolerance

(d) Miura 5% Deformation Tolerance (e) Miura 1% Deformation Tolerance (f) Miura 0.1% Deformation Tolerance

Figure 2: Random sample one million configurations uniformly for a Waterbomb crease pattern (top) and a Miura crease pattern (bottom) under different deformation tolerances. The crease patterns of these origami can be found in Fig. 3. Red: has self-intersection, invalid. Yellow: deformation is larger than tolerance, invalid. Magenta: within deformation tolerance but actual folding angles are different from assigned ones, invalid. Blue: valid.

3.2 Configuration

We use the folding angles of crease lines to represent the configuration of an origami model. For an origami with n crease lines, its configuration is represented as $\mathcal{C} = [\rho_1, \rho_2, \cdots, \rho_n]^T$ where ρ_i is the folding angle of the i-th crease line.

3.3 Folding Map

Folding map [14] is function that defined on each face which maps a point in that face from $\Re^2$ (on the crease pattern plane) to the corresponding point of folded state in $\Re^3$. For a given foldable configuration $\mathcal{C}$, we can compute the folding map, and with the folding map, the crease pattern can be folded to the 3D shape represented by the configuration $\mathcal{C}$ *instantaneously* [9]. However, the intermediate motion remains unknown.

3.4 Valid Configurations

Given a configuration $\mathcal{C}$, $\mathcal{C}$ can be classified according to its *foldability* and *feasibility*.

Foldability For a vertex v in the crease pattern, we use $\mathcal{C}_v = [\rho_{v_1}, \rho_{v_2}, \cdots, \rho_{v_k}]^T$ to denote a configuration of v where ρ_{v_j} is the folding angles of the j-th crease line incident to v. It is obvious that $\mathcal{C}_v$ is a subset of $\mathcal{C}$. The necessary condition for guaranteeing foldability requires every vertex v in the crease pattern to satisfy the closure constraint [14]

$$F(v, \mathcal{C}_v) = \prod_{i=1}^{k} \chi(\rho_i) = I \tag{1}$$

where $\chi(\rho_i)$ is a folding matrix for folding the i-th crease line by ρ_i. Due to page limit, please refer to [9] for details about the folding matrix χ.

During the folding process, [9] tries to minimize the objective function $|F(v, \mathcal{C}_v) - I|$ to find a valid configuration. However, computing the error between the mapping function and identity matrix usually does not give us a quantitative measure of how much an invalid configuration deviates from the manifold of foldable configurations [15]. Thus, in this paper, we propose two new metrics: deformation and angle inconsistency (discussed below). With these new metrics, we say that a configuration is rigid foldable if both deformation and angle inconsistency of the folded shape are under user specified tolerances.

Deformation A rigid foldable configuration should be deformation free. Deformation is measured on every edge including virtual edges and it is defined as $(||e_{folded}|| - ||e_{org}||)/||e_{org}||$. If the maximum deformation of a shape folded by apply the folding map of a configuration $\mathcal{C}$ is larger than a user given deformation tolerance, we say that $\mathcal{C}$ is non-foldable.

Angle inconsistency We measure the folding angles of each crease line on the folded shape and compare it to the assigned angle in the configuration. The angle inconsistency is defined as $|\rho_{folded} - \rho_{assigned}|$. In most of the cases, angle inconsistency is caused by deformation which means the configuration is non-foldable. However, sometimes we noticed that there are huge differences between two angles even though the folded shape is deformation free. Vertices shared by faces have multiple folding maps, whose coordinates may be override later depends on the order of applying the folding maps. For a configuration $\mathcal{C}$, if a vertex was mapped to different positions and its final position happen to be valid and deformation free, angle inconsistency can be used to find out those vertices, and $\mathcal{C}$ will be regarded as invalid.

Feasibility There are several properties that a rigid origami should have during folding: (1) unstretchable, (2) flat (planar) for all faces, and (3) free of self intersection. However, a foldable configuration only guarantees the first two properties. Collision detection needs to be applied on the folded shape to determine the feasibility.

Finally, we classify a configuration into one the following 4 categories according to deformation and angle inconsistency tolerances:

1. Invalid: deformation is larger than tolerance.

2. Self-intersected: deformation is within the tolerance but self-intersection occurs.

3. Inconsistent: deformation is within the tolerance but folding angle is inconsistent.

4. Valid: otherwise.

3.5 Continuous Foldability

Given two valid configuration $\mathcal{C}_1$ and $\mathcal{C}_2$, it is important to determine whether or not $\mathcal{C}_1$ can be continuously folded to $\mathcal{C}_2$. We say that $\mathcal{C}_1$ can be continuously folded to $\mathcal{C}_2$ if a set of valid intermediate configurations (within user given resolution) exists.

4 Folding Multi-Dof Rigid Origami

We say an origami is Multi-DOF if there exists a configuration that under which one or more crease lines can be folded/unfolded independently, i.e. its rigidity can be maintained without folding other crease lines.

4.1 Sampling In Discrete Domain

Traditional sampling strategies have difficult to effectively generate valid samples in the configuration space for rigid origami with closure constraints even in lower dimensional space. Some crease patterns have been shown to be 1-DOF mechanism such as the Miura crease pattern [16] which means the valid configurations form a curve in the configuration space, thus the probability of a random configuration to be valid is zero. Although we could tolerant certain amount of deformation, the configuration space is still mostly occupied by "obstacles" as shown in Fig. 2, only 0.044% of the configuration space is valid under 0.1% deformation tolerance for the Miura crease pattern with number of crease lines reduced to 2 by taking symmetry into consideration [15]. And situation will become even worse in higher dimensional configuration space.

To address this problem, instead of sampling in the continues domain with zero probability to generate a valid configuration, we propose the idea of sampling in the discrete domain. For a crease line with target folding angle ρ, we only sample the folding angle from its *important angle set*: $\{0, \pi, \rho\}$ which are corresponding to the flat state, the fully folded state and its target state. The total number of unique configurations for the origami with n crease lines is 3^n. For 1-DOF origami, usually it has only two continuous foldable configurations in the discrete domain which represent the initial state and the target state. And for Multi-DOF origami we expect to find more valid and continuous foldable configurations.

4.2 Connecting Two Valid Configurations

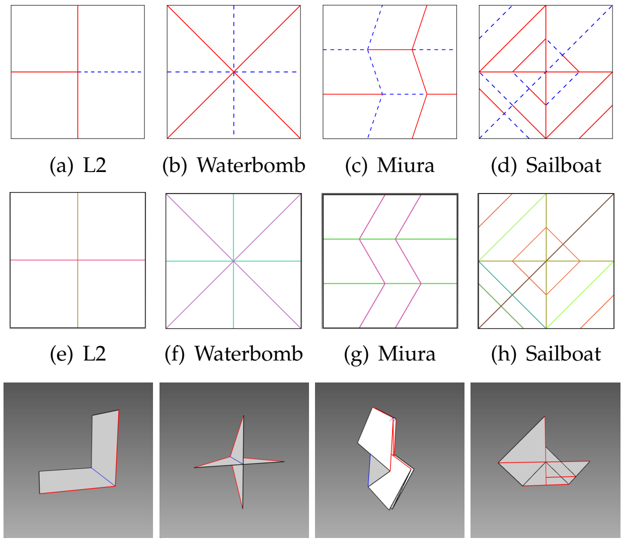
Given two valid configurations, it is usually unknown whether a rigid foldable and collision free path exist or not due to closure constraints which usually result in highly nonlinear path. In order to connect two configurations, we employ two connection methods: linear connection and nonlinear connection.

Linear connection An intuitive but turns out the most efficient way to connects two valid configurations is by linearly interpolating the intermediate configurations. Two configurations are rigid foldable to each other if all interpolated intermediate configurations are rigid foldable and collision free.

Nonlinear connection If linear connection failed to connect two configurations which means the path has to be nonlinear or even does not exist. We use a randomized search method proposed in [9] to connect two valid configurations, which could find a nonlinear, rigid foldable and collision free path. A folding path for the Waterbomb crease pattern found by [9] is shown in Fig. 7(c), from which we can see that the entire path is nonlinear.

4.3 Path Planning

We propose a folding path planner for a Multi-DOF origami under the Lazy-PRM framework [17]. First, we sample configurations in the discrete domain by adding valid configurations to the roadmap. We then connect all pairs of the configurations initially and add the edges to the roadmap. Then a graph query is answered to find a path from start node to target node. Connectivity checking will be applied only on the consecutive nodes in the path. If two nodes cannot be connected, i.e., they are not continuous foldable to each other, their corresponding edge is removed from the roadmap and a new path is extracted. We repeat this process until all edges that connect consecutive nodes in the path are validated. Finally, the rigid foldable and self-intersection free path is obtained by combining all the path segments.



(a) L2 (b) Waterbomb (c) Miura (d) Sailboat

(e) L2 (f) Waterbomb (g) Miura (h) Sailboat

(i) Target shapes

Figure 3: **Top**: Crease patterns used in our experiments. Mountain creases are shown as solid lines in red, valley creases are show as dashed lines in blue. **Middle**: Crease patterns with crease lines in groups. crease lines in the same group are shown in the same color. **Bottom**: Target shapes of above crease patterns.

5 Experiments And Discussions

5.1 Experiment Setup

We implemented the proposed method in C++, which will be opened to the public after this work is published. All data reported in this paper were collected on a MacBook Pro with a 2.9 GHz Intel Core i7 CPU and 16GB Memory running Mac OS X Yosemite. Crease patterns used in the experiments are shown in Fig. 3 with their target shapes. In the following experiments, without notice we use 1% as the deformation tolerance, 0.0174 rad ($\approx 1^\circ$) as folding angle inconsistency tolerance. A configuration is regarded as valid if it falls into the "Valid" category.

5.2 Visualization of the Configuration Space

We observed that some crease lines will have almost the same folding angles during entire folding process. For example, in the folding path of the Waterbomb crease pattern as shown in Fig. 7(c), 8 trajectories of folding angles overlapped into 2 groups. This is due to those crease lines are symmetric to each other in the crease pattern. By assuming corresponding crease lines have the same motion in the folding process, we can gather them into groups. As shown in Fig. 3, crease lines in the same group are displayed in the same color. By using symmetry property of the crease pattern, dimensionality of the configuration space could be reduced significantly [15], and visualization of the configuration space become possible.

Configuration spaces of Waterbomb and Miura crease patterns under different deformation tolerances are shown in Fig. 2, from which we can see that the majority of the configuration space is invalid since configurations that do not satisfy closure constraints will generate large deformation which makes the origami become unfoldable. Also we can see from Fig. 2 that there are curves that connect the initial state $[0, 0]^T$ and some extreme states (one or more crease lines/groups are fully folded). For the Waterbomb crease pattern, we can clearly see that there are two extreme states, the one with the configuration $[\pi, -\pi/2]^T$ represents the target shape of the Waterbomb crease pattern as shown in Fig. 3(i), another one with the configuration $[\pi/2, -\pi]^T$ happens to represent the bi-stable state [18] of Fig. 3(i). Shapes that those two extreme states represent to are shown in Fig. 4.

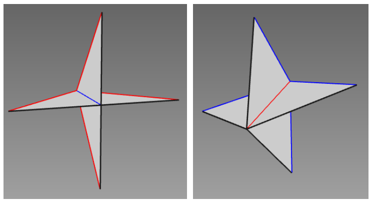


Figure 4: Shapes represented by the extreme states of Waterbomb.

5.3 Continuous V.S. Discrete Sampling Strategy

In order to evaluate our method, we conduct an experiment on the crease patterns shown in Fig. 3. The number of crease lines n in the crease pattern we used are from 2 to 12 shown in Table 1, which equal to the dimensionality of the configuration space.

We uniformly sample one million random configurations in the configurations space and compare the number of valid samples and their running time.

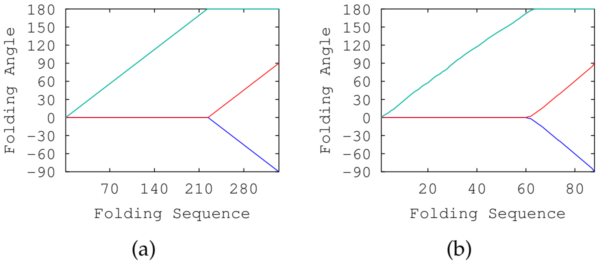
Sampling in continuous domain As we can see from Table 1, even in lower dimensional space (e.g., 2D) it can generate only a few valid configurations, for Waterbomb and Miura crease pattern in 2D, the valid configurations are only about 1.02% and 0.13% respectively. With the increase of dimensionality, it failed to find any valid configuration even though the origami is Multi-DOF due to closure constraints.

Sampling in discrete domain For the proposed method, folding angles are sampled only from each crease line's *important angle set*: $\{0, \pi, \rho\}$ as **Discrete3**. For comparison we also sampled from another angle set: $\{0, \pi/2, \pi, \rho\}$ as **Discrete4**. From Table 1 we can see that, this strategy finds several intermediate configurations efficiently since we can filter out duplicated configurations in constant time using a hashtable, and effectively as we will see later those intermediate configurations are very important.

In this experiment we show that sampling in discrete domain is a powerful strategy to generate valid samples for rigid origami with closure constraints. This strategy works even when the sampling domain is small (in our case only three values) and enables us to discover foldable states while sampling in continuous domain was not able to find any.

5.4 Linear Connection V.S. Nonlinear Connection

Let us look at the L2 crease pattern shown in Fig. 3(a) first. L2 crease pattern has 4 crease lines, two vertical crease lines in the middle are both mountain creases which need to be fold first since the horizontal ones have opposite folding angles ($\pi/2$ and $-\pi/2$). L2 has two folding steps hidden in the crease pattern, we say it has implicit folding orders. Using the proposed method, we are able to discover one valid intermediate state, which makes the connections from start to the intermediate state and from the intermediate state to goal both become linear, though the entire folding process is nonlinear. Folding path found by the proposed method are shown in Fig. 5(a). In this case, FROCC was also able to find the folding path as shown in Fig. 5(b). Both paths look identical to each other except the path find by nonlinear connection has a little turbulence. If no deformation is allowed in the system, the only valid folding path has to be Fig. 5(a) exactly. The total running time for proposed method to find the path is 11.784ms, in which sampling takes 0.971ms to generate 3 valid configurations among 192 random samples in the discrete domain and connecting takes 10.582ms. While FROCC costs 209.747ms which is about 18x slower than the proposed method, since it uses nonlinear connection, the NLopt library it used will generate much more samples in order to find valid ones.



(a)

(b)

Figure 5: Folding paths of L2 crease pattern. (a) Linear connection with one intermediate state. (b) Nonlinear connection.

In this experiment we show that sampling in the discrete domain allow us to discover some intermediate states of the origami, which may make one or more path segments become linear. Thus we could find the folding path much faster and more accurate than using nonlinear connection alone.

5.5 Alternative Paths for Multi-DOF Origami

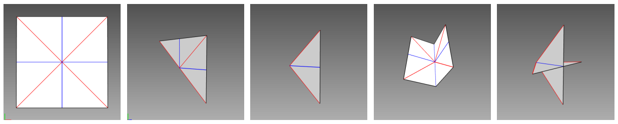


Figure 6: Continuous foldable shapes folded from Waterbomb crease pattern.

Table 1: Comparison Between sampling strategies

Model	n	Continuous		Discrete3		Discrete4	
		Valid	Time	Valid	Time	Valid	Time
Waterbomb*	2	10161	12.10	5	0.35	6	0.42
Miura*	2	1305	17.48	3	0.36	3	0.43
L2	4	1	10.31	5	0.45	6	0.52
Sailboat*	6	0	33.70	48	0.57	118	0.74
Waterbomb	8	0	10.97	71	0.73	114	0.89
Miura	12	0	17.35	7	0.94	7	7.13

Running time is measured in second. * indicates that symmetry property is used.

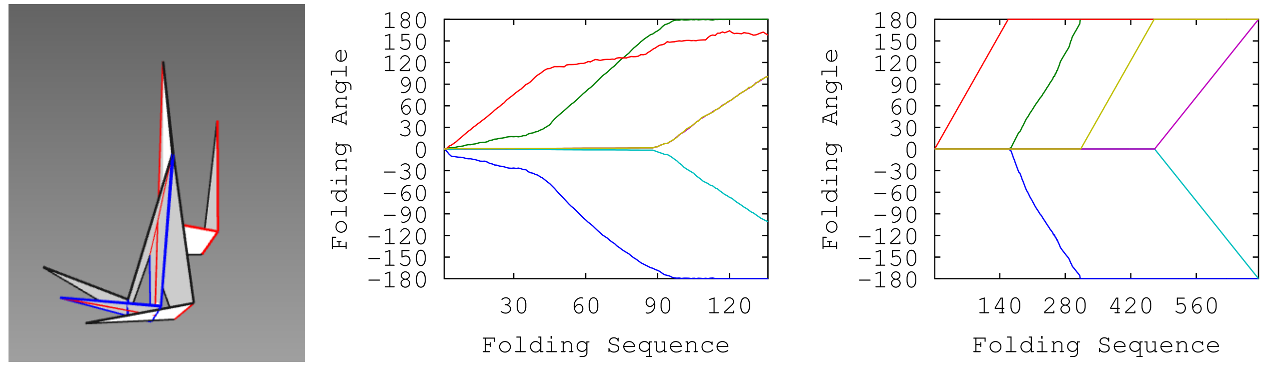
For 1-DOF origamis such as the Miura crease pattern shown in Fig. 3(c), sampling in the discrete domain may not help since typically they have only two continuous foldable configurations which represent the initial state and goal state and nonlinear connection has to be used to connect those two configurations. However, for Multi-DOF origamis, at least one valid intermediate configuration can be sampled using the proposed method, and we can expect much more. Here we use the Waterbomb crease pattern shown in Fig. 3(b) as an example. The Waterbomb crease pattern has 8 crease lines: 4 mountain creases and 4 valley creases, and it is a Multi-DOF origami. Previous methods like ROS or FROCC could find at most one folding path even though the origami is a Multi-DOF system and has many folding paths. The folding path found by FROCC for the Waterbomb crease pattern is shown in Fig. 7(c), and the folding process is shown in Fig. 7(a). Using the proposed method, we found 71 valid configurations on the Waterbomb crease pattern. In Fig. 6, we show 5 continuous foldable shapes in which 3 three of them represent intermediate states. Via intermediate configurations, we found alternative folding path for the Waterbomb crease pattern, folding path and folding process are shown in Fig. 7(b) and Fig. 7(d) respectively. As we can see from Fig. 7(b), the origami folds to goal state via an intermediate state and one of the path segments is linear.

In this experiment we show that intermediate states of Multi-DOF origamis sampled from the discrete domain allow us to capture the connectivity of the configuration space of the origami and a roadmap could be built and queries can be answered. The proposed method could find multiple paths from start to goal, user can choose their favorite one(s) according to the applications. Such as finding the path that has the minimum energy requirement to actuate the origami from flat to folded state.

5.6 Global Connectivity

Local Minima and Self-Intersection Avoidance Previous methods like ROS or FROCC could find only one folding path even though the origami is a Multi-DOF system and has many folding paths. And both methods are deterministic even though FROCC employs some randomness, however, its greedy choice property dominants the path globally, randomness only impacts the path locally (small turbulences). Single path and determinacy make pervious methods easily trapped at local minima or lead to self-intersection and would not able to fold any further. An interesting example is to use FROCC to fold sailboat crease pattern as shown in Fig. 3(d) without specified intermediate configurations, it folds the origami to the very end but it can not fold any further, since the next optimal, closer to goal and foldable configuration it tries to reach has self-intersection as shown in Fig. 8(a), and the folding path is shown in Fig. 8(b). On contrast, sampling in the discrete domain gives us many valid intermediate configurations which enables us to capture the global connectivity of the configuration space. And we can easily fold the origami from flat sheet to target shape without being trapped at local minima and could avoid self-intersection regions. The folding path found by the proposed method is shown in Fig. 8(c) from which we can see that the origami fold from start to goal via 3 intermediate states.

Connectivity of the Configuration Space With the sampled valid configurations by the proposed method, we could capture the connectivity of the configuration space by checking the connectivity of all pairs of configurations. A visualization of the connectivity of the Waterbomb crease pattern is shown in Fig. 9, in which each node represents a valid configurations, and an edge connects two configurations if they are directly connected (either via linear or nonlinear connection). There are 71 nodes and 412 edges in the graph, 180 linear paths and 232 nonlinear paths. Nonlinear paths are shown as bolded lines. Node 0 represents the initial state while Node 70 represents the target state. As we can see from Fig. 9, initial state and goal state are connected via a nonlinear path (the only path FROCC can find), however, many alternative paths exist. One thing need to be noticed is that there are two connected components in the graph, which means there are a set of nodes (5 of them are shown in Fig. 10(a)) that the shapes they represent are not able to be continuously folded from flat sheet while they can be folded to each other continuously (via



(a) FROCC failed to fold sailboat due to self-intersection

(b) FROCC

(c) Proposed method

Figure 8: Folding paths of the sailboat crease pattern.

zero or more intermediate configurations).

5.7 Non-continuous Foldable Configurations

By sampling in discrete domain, we would able to discover many valid intermediate configurations, however, some of them may not able to be folded from flat sheet or folded to goal state while they can fold to other configurations continuously as we can see from Fig. 9. We call those configurations as non-continuous foldable configurations from flat sheet. More detailed discussion about continuous foldability can be found in [19]. Here we show some non-continuous foldable shapes folded from valid configurations in Fig. 10, from which we can see all of the folded shapes are rigid, deformation and self-intersection free. Some of them (e.g., the last three shapes of Miura crease pattern) are even continuous foldable if mountain/valley creases are not restrictedly enforced. Although they cannot be folded from flat sheet, unique shapes that they can reach, their flat-foldability, etc. will be interesting to explore and remain as the future work.

5.8 Limitation and Future Work

Even though we sample from a finite set, the configuration space is still exponential to the number of crease lines. Thus, discrete sampling doesn't make the problem much easier, and it remains challenging to fold complex crease patterns.

During the experiment we notice that different configurations may represent the same shape. These shapes can be perfectly aligned via rigid transformation. How those "duplicated" configurations can be removed while still maintaining the connectivity remains as the future work.

6 Conclusion

Instead of uniformly sampling in the continuous domain that has zero probability of generating a valid configuration for rigid origami with closure constraints, in this paper, we proposed a novel method that samples in the discrete domain: folding angle of a crease line is sampled from the finite angle set of that crease line. Our experimental results show that the proposed method is effective and efficient to generate important intermediate configurations for origamis, in particular for those Multi-DOF with closure constraints. With these valid intermediate configurations, the proposed motion planner has the following advantages: (1) Globally nonlinear path could be replaced by the combination of linear path segments and nonlinear ones, which significantly speed up the path finding process, (2) Multiple folding paths could be found for Multi-DOF origamis, and (3) Local minima and self-intersection regions could be avoided.

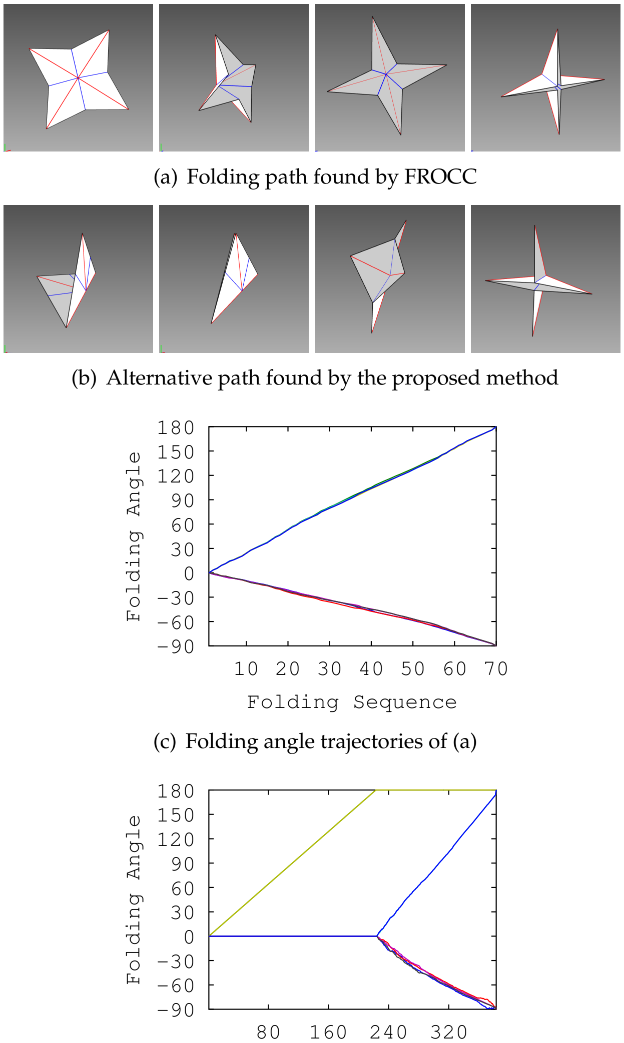
References

[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] J. Ryu, M. D'Amato, X. Cui, K. N. Long, H. J. Qi, and M. L. Dunn, "Photo-origami—bending and folding polymers with light," *Applied Physics Letters*, vol. 100, p. 161908, 2012.

[4] S. Ahmed, C. Lauff, A. Crivaro, K. McGough, R. Sheridan, M. Frecker, P. von Lockette, Z. Ounaies,



(a) Folding path found by FROCC

(b) Alternative path found by the proposed method

(c) Folding angle trajectories of (a)

(d) Folding angle trajectories of (b)

Figure 7: Valid states and folding process of a Waterbomb crease pattern.

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.

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

[6] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 566–580, 1996.

[7] S. M. LaValle and J. J. Kuffner Jr, "Rapidly-exploring

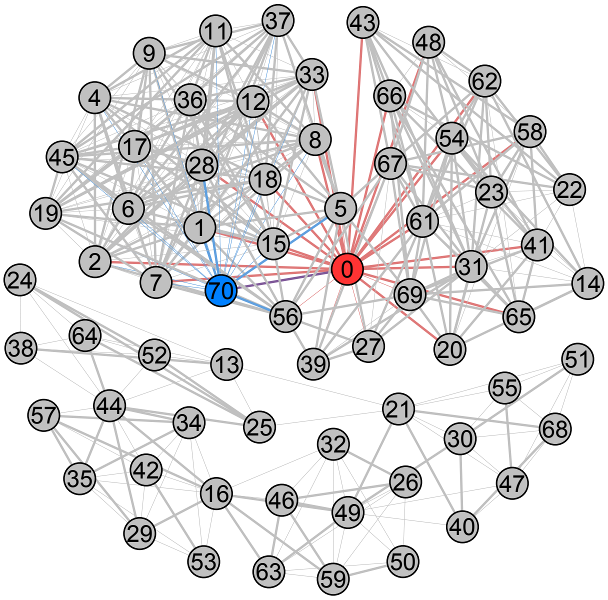


Figure 9: Connectivity graph of the Waterbomb crease pattern. Each node represents a valid configuration in which Node 0 shown in red represents the initial state (flat sheet) and Node 70 shown in blue represents the target state. Two node are connected by an edge if they are continuous foldable to each other. Nonlinear connected edges are bold.

random trees: Progress and prospects," 2000.

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

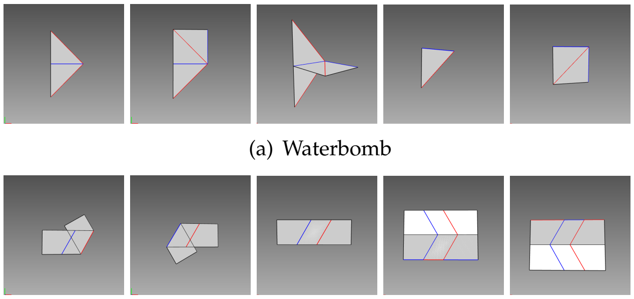
[9] Z. Xi and J.-M. Lien, "Folding rigid origami with closure constraints," in *International Design and Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE)*, (Buffalo, NY), ASME, Aug. 2014.

[10] 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.

[11] 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.

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

[13] 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.



(a) Waterbomb

(b) Miura

Figure 10: Shapes that are non-continuous foldable from flat sheet.

[14] 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.

[15] Z. Xi and J.-M. Lien, “Folding and unfolding origami tessellation by reusing folding path,” Tech. Rep. GMU-CS-TR-2015-2, Department of Computer Science, George Mason University, 4400 University Drive MSN 4A5, Fairfax, VA 22030-4444 USA, 2015.

[16] K. MIURA, “Proposition of pseudo-cylindrical concave polyhedral shells,” *ISAS report*, vol. 34, no. 9, pp. 141–163, 1969.

[17] R. Bohlin and E. Kavraki, “Path planning using lazy prm,” in *Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on*, vol. 1, pp. 521–528, IEEE, 2000.

[18] L. Bowen, M. Frecker, T. W. Simpson, and P. von Lockette4, “A dynamic model of magneto-active elastomer actuation of the waterbomb base,” in *International Design and Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE)*, (Buffalo, NY), ASME, Aug. 2014.

[19] Z. Xi and J.-M. Lien, “Determine distinct shapes of rigid origami,” in *The 6th International Meeting on Origami in Science, Mathematics and Education (6OSME)*, (Tokyo, Japan), Aug. 2014.