# Distance between Folded Objects

Chen Gu* Leonidas Guibas†


## Abstract

Geometric folding problems have recently attracted much attention in both mathematics and theoretical computer science. In this paper, we study the following basic problem: given a set of folded conformations of a unit-length rope in 1D, how do we decide which are more similar or less similar to each other? We first define a distance function between flat folded states that incorporates both the geometry and the overlap order. Then we do some computational experiments clustering random folded ropes with this distance metric. Finally, we generalize our results for 1D folded ropes to flat folded papers (origami) in 2D.


## 1 Introduction

We start by introducing the folding model and relevant definitions from [3]. The notion of folding ropes is very intuitive. Mathematically, we imagine the rope as a one-dimensional line segment that has zero thickness. A folding motion is a continuous motion of the rope from one configuration to another that does not cause the rope to stretch, tear, or self-penetrate. A snapshot of this motion at a particular time is called a folded state.

Since ropes are not rigid, all folded conformations (final folded states) are flat foldings. A flat folding has the property that it lies in the same space as the unfolded line segment. However, there can be multiple layers of the segment at a point, so the folding really occupies a finite number of infinitesimally close copies of the one-dimensional space. Formally, a flat folded state can be specified by a function mapping all points to their folded positions, together with a partial order that specifies their overlap order. For every pair of distinct noncrease points that are mapped into the same position, this partial order should specify which one lies above the other.

In this paper, we consider a new folding problem: determine which folded conformations are more similar or less similar to each other. This problem is very fundamental in the area of geometric folding and fairly natural from a practical point of view. Traditionally, there are many distance functions that can measure the similarity between different geometric shapes [5]. However, the difficulty of comparing folded conformations is that we cannot extract from the geometry the relative stacking order of portions of the line segment that come into contact as multiple overlapping layers. This lack of information makes it impossible to identify different folded states. Thus, we need to define a distance function that incorporates both the geometry and the ordering.

There have been several related studies on folding one-dimensional strings in recent years. For example, Arkin et al. solved the 1D flat-foldability problem that characterize which crease patterns and mountain-valley assignments have flat folded states [1], and Cardinal et al. investigated the folding complexity about how to quickly fold a paper strip to obtain a desired mountain-valley pattern of equidistant creases [2]. In particular, we will use the results of the 1D flat-foldability problem to design a sampling algorithm in our experiment of clustering random folded ropes with this distance metric.

## 2 Distance between folded ropes in 1D

In this section, we define a distance function to measure the similarity between different folded conformations of a unit-length rope, and develop an efficient algorithm for the distance computation.

### 2.1 Distance function

In order to compare flat foldings, we discretize the rope into $2n$ points, uniformly distributed on both sides of the rope (see Figure 1). We use a plus sign superscript to denote the points on the top side, and a minus sign superscript to denote the points on the bottom side. Now, consider the two conformations shown in Figure 2: suppose we fold the rope at position 3, but in two different directions. In 2(a), point 4 is connected to point $2^-$, but not point $2^+$; In 2(b), point 4 is connected to point $2^+$, but not point $2^-$. Therefore, although the geometric positions of the two discretized point sets are identical, we could distinguish these two conformations from their different topological connections.

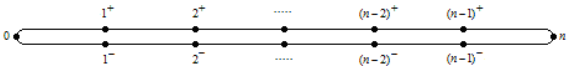


Figure 1: Discretization

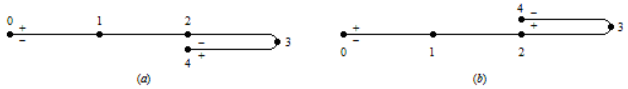


Figure 2: Mountain/Valley fold at position 3

*Institute for Computational and Mathematical Engineering, Stanford University, guc@stanford.edu

†Department of Computer Science, Stanford University, guibas@cs.stanford.edu

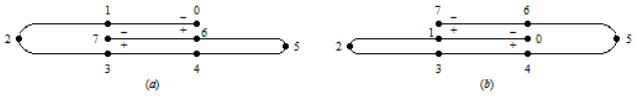

Figure 3: Two conformations with different overlap orders

We define the distance function using the distance root mean squared error (dRMS). dRMS is a metric of distance between two point sets with known correspondence, which is computed by comparing all internal pairwise distances of the two point sets:

$$dRMS^2(P,Q) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(||p_i - p_j|| - ||q_i - q_j||\right)^2$$

For simplicity, we assume that $n$ is large enough so that we only fold the rope at integer sample points $\{1, 2, ..., n-2, n-1\}$ (or we can approximate all folding operations at these positions). Given two folded conformations, we compute their distance using dRMS, where $p_i, p_j, q_i, q_j \in \{0, (1/2)^{\pm}, 1^{\pm}, (3/2)^{\pm}, 2^{\pm}, ..., (n-1)^{\pm}, (n-1/2)^{\pm}, n\}$. (We define $(i+1/2)^{\pm}$ as the midpoint of $i^{\pm}$ and $(i+1)^{\pm}$.) The internal distance $||p_i - p_j||$ is defined as the geodesic distance $d(p_i, p_j)$ from $p_i$ to $p_j$ along the rope. For two points that come in contact in the folded conformation, we consider them to be connected, so that the distance between two contacted points is 0. For example, consider the two conformations shown in Figure 3: we denote $h = 1/n$ as the length between two consecutive integer sample points along the rope. In 3(a), since 0 and $6^-$ are connected, we have $d(0, 6^-) = 0$; However, in 3(b), $d(0, 6^-) = 2h$ since the shortest path from 0 to $6^-$ along the rope is $0 \to 1^-(7) \to 6^-$. Intuitively, the two conformations in Figure 3 have very similar geometric shapes, however, it might follow a long trajectory if we want to deform from one shape to another since they have different overlap orders. Using this distance function, we can check that the distance increases monotonically as we increase the overlap of the top two layers.

## 2.2 Algorithm

To compute internal pairwise distances, we build a graph of $4n$ nodes corresponding to all sample points $\{0, (1/2)^{\pm}, 1^{\pm}, (3/2)^{\pm}, 2^{\pm}, ..., (n-1)^{\pm}, (n-1/2)^{\pm}, n\}$. For any two consecutive sample points along the rope, we connect them by an edge with weight $h/2$. Thus, the graph of an unfolded rope is simply a cycle of $4n$ nodes. For two sample points that come in contact by a folding operation, we add an edge between them with weight 0. Then, the internal distance between any two sample points along the rope is the same as the shortest path distance between their corresponding nodes in this graph.

This graph has a property that it is very sparse. In fact, each node has exactly 2 incident edges of weight $h/2$ and at most 2 incident edges of weight 0: for any sample point, it belongs to at most 2 unit segments on the rope (a unit segment has length $h$ with two endpoints at integer positions). For each segment, there can only be one new segment lying immediately above/below (depending on its side) it in the folded conformation. Therefore, each sample point is connected to at most 2 other sample points with edges of weight 0 from these new segments. As a result, the degree of each node in this graph is at most 4, which means the total number of edges $m = O(n)$.

For shortest paths computation, since there are only two types of different weights in the graph (0 and $h/2$), the shortest paths distances between all pairs of nodes are multiples of $h/2$ between 0 and 1. From this observation, when we implement the Dijkstra's algorithm from a single source, we can simply build an array of $2n+2$ cells, with cell $i$ pointing to a linked list of nodes that have current upper bound on distance equal to $ih/2$. Initially, the source is linked to cell 0, while all other nodes are linked to cell $2n+1$ (which serves as infinity). Each time we update the upper bound on distance of some node by relaxing an edge, this node moves left to a linked list of nodes that have smaller distance. Using this data structure, the total amount of time is bounded by $O(m)$ for time we spent on relaxing edges plus $O(n)$ time we spent moving right in the array, looking for the next non-empty cell. Thus, the total running time of Dijkstra's shortest path algorithm is $O(m)$ in this case [4], instead of $O(m + n \log n)$ using Fibonacci heaps. Finally, since $m = O(n)$, we can implement a single source Dijkstra's algorithm in $O(n)$ time, and compute all internal pairwise distances in $O(n^2)$ time.

## 2.3 Symmetries

There are eight types of symmetries for folding ropes (see Figure 4). The two conformations in each column are identical, which only differ by a rotation of 180 degrees. Type (a) and type (b) differ by a horizontal reflection: if we can swap the top/bottom sides of the rope, these two types become identical. Type (a) and type (c) differ by a vertical reflection: if we can number the sample points in a reverse order from right to left, these two types become identical. Finally, type (d) contains both a horizontal and a vertical reflection from type (a).

When we do not distinguish the two ends of the rope, the conformations in type (a) and type (d) are considered as the same state. In this case, we can define the distance between two conformations $C_1$ and $C_2$ as $\min\{d(C_1, C_2), d(C_1, C_2^{rot})\}$, where $C_2^{rot}$ corresponds to a type (d) rotation on $C_2$. Then the new distance function would be invariant under rotations. Furthermore, if we also want to include the symmetry of reflections, we can define the distance between two conformations as the minimum of four distances under all possible rotations/reflections.

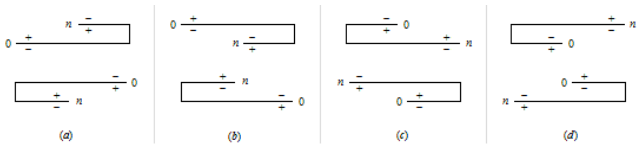

Figure 4: Symmetries

## 3 Experiment: clustering folded ropes

In this section, we test the performance of the distance function we defined in Section 2. We first design a sampling algorithm to generate random folded conformations of a rope, and then use this distance metric to cluster similar conformations together.

### 3.1 Generating random folded conformations

In our experiment, we generate random folded conformations of a rope from 1D crease patterns. The characterization of flat-foldable crease patterns has been studied extensively in origami science [3]. In 1D, a crease pattern is a set of prescribed crease points labeled with mountain or valley directions on a line segment. In the model of simple foldings, a flat folding is made by a sequence of simple folds, each of which folds one or more layers of the line segment. In [1], Arkin et al. showed that a 1D mountain-valley pattern is flat foldable if and only if it can be folded by a sequence of crimps and end folds.

Formally, let $c_1$,...,$c_n$ denote the creases on a line segment, oriented from left to right. In addition, let $c_0$ denote the left end and $c_{n+1}$ denote the right end. First, a pair $(c_i, c_{i+1})$ of consecutive creases is crimpable if $c_i$ and $c_{i+1}$ have opposite directions and $|c_{i-1} - c_i| \geq |c_i - c_{i+1}| \leq |c_{i+1} - c_{i+2}|$. Crimping such a pair corresponds to fold $c_i$ and then fold $c_{i+1}$, using one-layer simple folds. Second, $c_0$ is a foldable end if $|c_0 - c_1| \leq |c_1 - c_2|$, and $c_{n+1}$ is a foldable end if $|c_{n-1} - c_n| \geq |c_n - c_{n+1}|$. Folding such an end corresponds to perform a one-layer simple fold at its nearest crease. In the 1D flat-foldability test, we only need to search for one of these two local operations to perform. Moreover, the sequence can be found in any greedy way.

Using this result, we can generate random folded conformations as follow: we first randomly generate a crease pattern by sampling folding positions and mountain/valley directions uniformly on the rope, and then run the flat-foldability test to check whether this crease pattern can be folded flat. We repeatedly search for a local minimum segment and test whether we can fold it using crimp or end fold, until we reach the final flat folding or we know it cannot be folded flat. When the flat-foldability test fails, we simply generate a new crease pattern and test it again.

Notice that some crease patterns may have multiple flat folded states (for example, see Figure 3). The reason is that their sequences of crimps and end folds are different so that they have different overlap orders in the fold states. From this observation, we randomly select an allowable crimp or end fold operation in each step during the flat-foldability test. Using this sampling technique, we would be able to generate all valid crease patterns and folded conformations. Finally, if we want to avoid duplicated folded conformations, we can use our distance function to check whether the distance between the new folded conformation and any previous conformation is equal to 0.

### 3.2 Clustering result

Using the dRMS metric, each conformation maps to a point in dimension $N^2$ whose coordinates are its pairwise internal distances. Thus, we can apply standard clustering algorithms to group similar conformations together. Notice that when we allow rotations or reflections in the distance computation, each conformation may correspond to multiple points in the dRMS space. In this case, we can use clustering algorithms which do not require coordinate representations.

In our implementation, we set $n = 20$ and generate 40 random conformations by at most 5 folds. We do not distinguish the two ends of the rope so that two conformations are considered as the same state if they only differ by a type (d) rotation (see Figure 4). For clustering, we select cluster centers using the farthest-first traversal algorithm, which gives a 2-approximation solution for the k-center problem by repeatedly picking a new center with maximum distance to its nearest center in the last iteration.

Figure 5 shows the experiment result with 10 clusters. We see that it works fairly well for clustering similar folded conformations. Generally, our distance function will ignore small crimps and end folds, and compare the folding structures of long segments. If two conformations have the same overlap order with similar lengths on corresponding long segments, or different overlap orders but the overlap length is small, then their distance should be small; Otherwise, they are more likely to be in different clusters.

## 4 Extension to flat folded papers in 2D

In this section, we generalize our results for 1D simple folding to orthogonal folding in 2D. In the orthogonal folding model, we only fold along horizontal or vertical directions on a square (or rectangular) piece of paper, where horizontal and vertical are defined by the sides of the paper. In origami science, this model is also called the map folding.

Given a square paper in 2D, we discretize it by using a uniform grid on both sides of the paper (see Figure 6(a)). Similarly, we can define the distance between two flat folded papers using the dRMS distance function (under rotations/reflections if necessary). In particular, we define the internal distance between two sample points by their geodesic Manhattan distance along the paper. For example, the distance between $p^+$ and $q^+$ is $2h$ ($h = 1/n$), however, the distance between $p^+$ and $q^-$ is $4h$ since we need to go across the boundary of the paper. Again, for two points that come in contact in the folded conformation, we consider them to be connected, so that their internal distance along the paper is 0.

To compute internal pairwise distances, we build a graph consists of all sample points at $(ih/2, jh/2)^{\pm}$ ($n + 1$ integer positions and $n$ midpoints along each axis) on both sides of the paper. Each sample point is connected to its 4 neighbors in the grid by edges of weight $h/2$, and at most 4 contacted points by edges

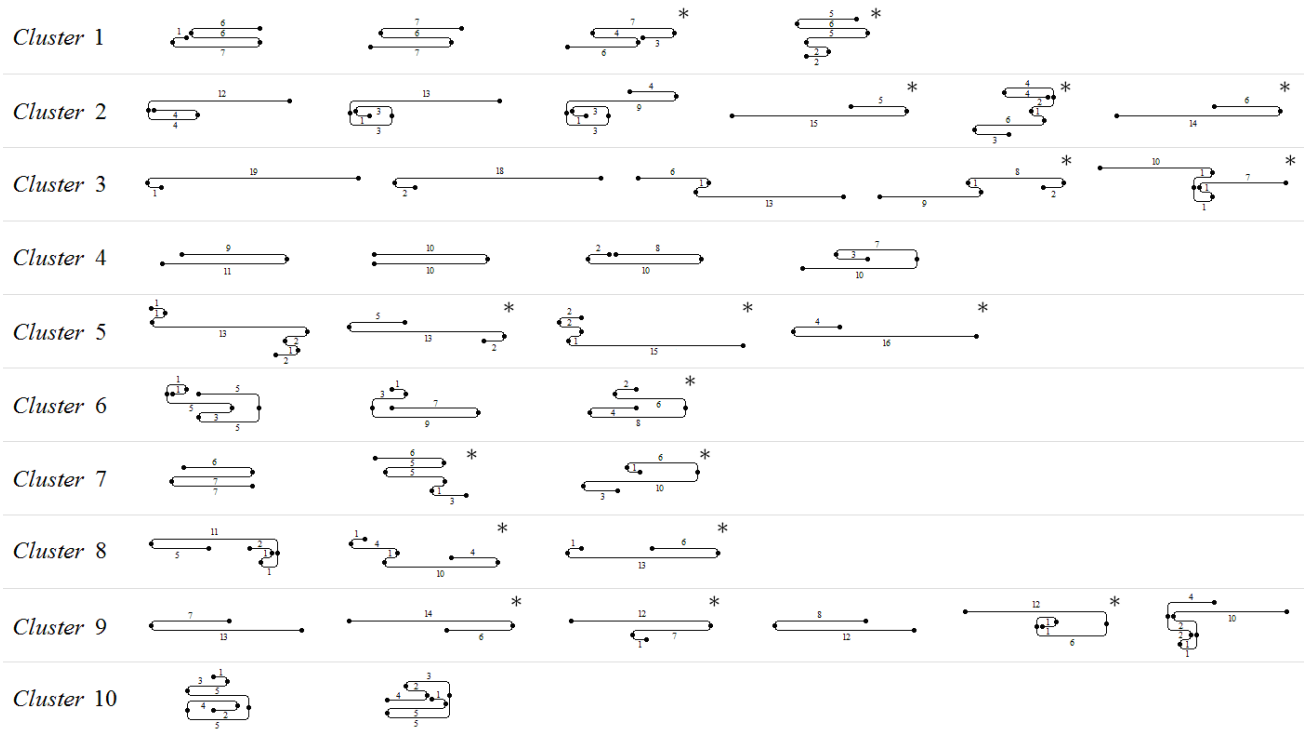


Figure 5: Experiment result: in each cluster, the first conformation represents the cluster center computed by the farthest-first traversal algorithm, and all other conformations are sorted in ascending order by their distances to this cluster center. (Conformations marked with a '*' are rotated by 180 degrees when compared with the cluster centers.)

of weight 0. (we check the 4 unit squares around a sample point to see if there are any unit squares lying immediately above/below (depending on their sides) them in the folded conformation, see the shaded areas around $p$ in Figure 6(a)). Therefore, the degree of each node in this graph is $O(1)$, and the total number of edges $m = O(n^2)$. Using a similar data structure from the 1D algorithm, we can implement a single source Dijkstra's algorithm in $O(n^2)$ time, and compute all internal pairwise distances in $O(n^4)$ time.

In addition, we can also allow folding operations to be axis-parallel plus at a 45-degrees angle (see Figure 6(b)). In this case, we can still use the geodesic Manhattan distance for internal distances computation. The only difference is that we may need to check at most 8 unit triangles around a sample point to build edges for connected points in the folded conformation.

When we allow folding operations at arbitrary directions, the geodesic Manhattan distance does not work because the grid may not be aligned after a simple fold. In this case, we may define the internal distance between two sample points as their geodesic Euclidean distance along the paper. However, since the shortest path connecting two points along the paper might be at any direction, we need to build a complete graph of all sample points, and the computation for their geodesic Euclidean distance might be very complex in a complicated folded conformation. The case of arbitrary folding in 2D might be a topic for our future research.

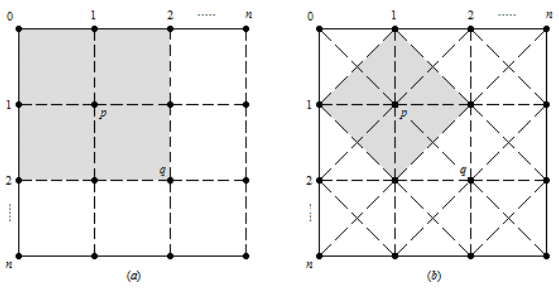


Figure 6: 2D folding (a) orthogonal (b) diagonal

## Acknowledgments

This research was supported by NSF grant IIS-0914833.

## References

[1] E. Arkin, M. Bender, E. Demaine, M. Demaine, J. Mitchell, S. Sethia, and S. Skiena. *When can you fold a map?* Computational Geometry: Theory and Applications, 29(1): 23-46, 2004.

[2] J. Cardinal, E. Demaine, M. Demaine, S. Imahori, S. Langerman, and R. Uehara. *Algorithmic folding complexity.* In Proceedings of the 20th Annual International Symposium on Algorithms and Computation, Lecture notes in Computer Science, volume 5878, pages 452-461, 2009.

[3] E. Demaine and J. O'Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra.* Cambridge University Press, 2007.

[4] H. Gabow. *Scaling algorithms for network problems.* Journal of Computer and System Sciences, 31:148-168, 1985.

[5] L. Kavraki. *Molecular distance measures.* Connexions, http://cnx.org/content/m11608/1.23/, 2007.