

# Computational Design of Iris Folding Patterns

Yuki Igarashi<sup>†1</sup> Takeo Igarashi<sup>2</sup> and Jun Mitani<sup>3</sup>

<sup>1</sup>Meiji University, Japan

<sup>2</sup>The University of Tokyo, Japan

<sup>3</sup>University of Tsukuba, Japan

## Abstract

*Iris folding is a motif consisting of layered strips of paper, forming a spiral pattern behind an aperture, which can be used to make cards and gift tags. This paper describes an interactive computational tool to assist in the design and construction of original iris folding patterns. The design of iris folding patterns is formulated as the calculation of a circumscribed polygonal sequence around a seed polygon. We analyzed the problem and found that it is possible to compute the positions of vertices analytically for regular polygon; however, it is not straightforward to do so for irregular polygons, and we used a numerical method for irregular polygons, which can be applied to arbitrary convex seed polygons. The user can quickly experiment with various patterns using the system prior to constructing the motif.*

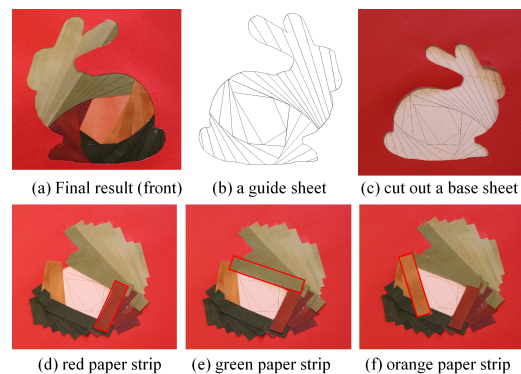
Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques

## 1. Introduction

Iris folding is a motif consisting of layered paper strips that form a spiral pattern behind an aperture, as shown in Figure 1. Iris folding is a simple and fun paper-folding technique that can be used to make greeting cards and gift tags. The center of the design forms an iris-like pinhole at the center, similar to that of an eye or a camera lens. In a typical workflow, a person first prepares a guide sheet (usually taken from a book), base sheet, and paper strips. She then cuts the base sheet making a hole as shown in the guide sheet. She places the base sheet on the guide sheet and pastes the paper strips on the base sheet following the instruction shown in the guide sheet. After pasting all the paper strips, she turns the base sheet over to obtain the final result.

There are several interesting geometric properties of iris folding patterns, which make the design of such patterns an intriguing mathematical problem. Although it can be easy to design basic iris folding patterns, even for novices, many of the more complex geometrical patterns must satisfy certain geometric constraints, as shown in Figure 3. It is typically difficult for novices to design such patterns. If the geometric constraints are not satisfied, waste of paper increases because it cannot use the width of paper strips effectively. In addition, it takes more time in manual construction as the number of required polygons increases.

This paper presents an interactive computational system to assist in the design and construction of original iris folding patterns that satisfy geometric constraints. Figure 2 shows an overview of the

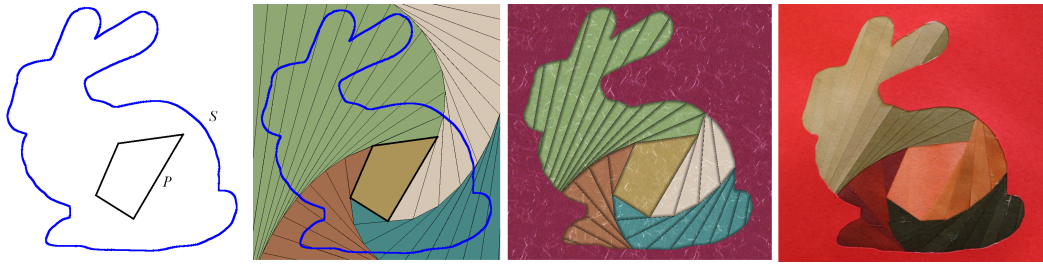


**Figure 1:** (a) The end result viewed from the front side. (b) a guide sheet. (c – f) The construction process viewed from the back side. (c) The user first cuts a paper and turns over the paper on a guide sheet. (d – f) Then the user pastes colored paper strips around the central polygon.

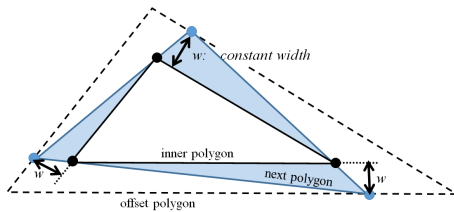
process. The user first inputs an outline and a seed polygon inside. Then it automatically computes an iris folding pattern, i.e., a series of strips around the seed polygon. The user can experiment with arbitrary polygons and color combinations prior to completing the pattern.

The design of iris folding patterns requires the computation of a series of strips of equal width around the polygon. As shown in Figure 3, we formulate this design problem as the calculation of an offset polygonal sequence around an inner polygon, which satisfies

<sup>†</sup> yukim@acm.org



**Figure 2:** An overview of our system. (a) The user first inputs an outline  $S$  (blue line) and a seed polygon  $P$  (black line). (b) Then the system automatically computes a pattern for iris folding and (c) applies texture rendering. (d) The user finally creates the physical pattern.



**Figure 3:** The problem is to calculate a next polygon circumscribed to the inner polygon whereby the distance from the vertices of the polygon to the corresponding internal polygon edge is constant.

the following constraint: the distance from the vertices of the polygon to the corresponding internal edge of the polygon is constant (i.e., the width of paper strips is constant). In this manner, the system computes iris folding patterns around an arbitrary convex seed polygon.

## 2. Related Work

Coahran and Fiume [CF05] reported a sketch-based design for a specific quilting art form, i.e., Bargello patterns. They described an algorithm that transforms sketched input data into graceful Bargello curves. Patchy [IM15] is an interactive system that can assist the design of original patchwork patterns. The user designs original patchwork strokes, and can quickly experiment with various patterns prior to sewing.

The pursuit curve of a polygon forms a geometric pattern that is similar to iris folding. It is defined as the trajectories of vertices starting from the corners of the polygon, and moving inwards pursuing the neighboring vertices [Pet01]. In the case of a triangle, three pursuit curves converge to a point known as the Brocard point. Although pursuit curves form similar patterns to those seen in iris folding, the definitions differ, and the analysis of pursuit curves cannot be applied directly to our problem.

## 3. User Interface

### 3.1. User Inputs

The user first inputs an outline polygon  $S$  and an internal convex seed polygon  $P$  inside  $S$ . The user can change the width  $w$  of the

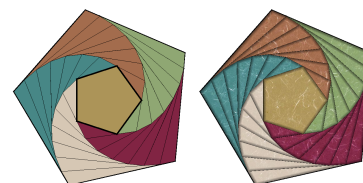
paper strips. The system provides two methods to input the outline  $S$ : the user can either draw a stroke line directly, or can load a black-and-white bitmap image and have the system trace the contour line using the marching-squares algorithm [LC87].

The user draws a polygon  $P$ , as shown in Figure 2(a). Arbitrary polygons can be designed by clicking on the vertex position of the polygon. The user can also draw a regular polygon by choosing one from a menu bar. The system automatically places the polygon at the center, and the user can drag it to a desired position. It is also possible to scale the polygon  $P$ . The default width of the paper strips is 10 mm; however, this can be changed using the menu. The user can change orientation of the strips using the menu.

### 3.2. Interactive Design

Once the outline  $S$ , the seed polygon  $P$  and the strip width  $w$  are determined, the system automatically generates a pattern for iris folding. When the user drags the vertices of the internal polygon  $P$ , the system updates the pattern in real time. The user can set the color of the strips using a paint tool. By default, the system shows a design with  $n$ -sided polygons with  $n$  colors, as shown in Figure 2(b).

To create a shaded image, the system may also switch to visualization with shading, considering the layers of paper strips, as shown in Figure 4. Regions of the same color are grouped for previewing, and the system calculates a rendered image for each group. The user can preview the resulting image using a texture image of paper strips. This cannot be manipulated interactively while previewing the texture image, however, and the system switches to normal coloring mode while the user performs a dragging operation.



**Figure 4:** (left) The fill tool, and (right) texture rendering.

### 3.3. Construction Guide

The system ultimately outputs a pattern for use in construction. The user pastes paper strips onto the back side of the main piece of paper, as shown in Figure 5. Therefore, the resulting guide sheet is flipped horizontally. The system also shows a step-by-step graphical presentation to aid the construction.

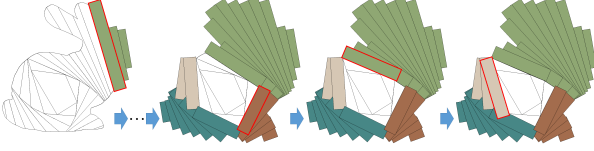


Figure 5: A step-by-step construction guide.

### 4. Theoretical Analysis

This section describes a theoretical analysis of the geometric problem. The algorithm used in the current implementation is described in the next section. First we define the problem, and then we describe an analysis of regular seed polygons, followed by the treatment of irregular seed polygons.

#### 4.1. Definitions

The seed polygon  $P$  (input by the user) is defined as  $P^0$ . The polygons formed using paper strips are defined as  $P^m$  ( $m = 1, 2, \dots$ ). We consider the problem as determining the  $P^{m+1}$  from  $P^m$  for a given strip width  $w$ . The system first computes an offset polygon  $P^{m'}$ , i.e., the polygon that is obtained by offsetting each edge of  $P^m$  by the strip width  $w$ . Then it searches for the  $P^{m+1}$  that satisfies the following geometric constraints: 1) The vertices of the next polygon  $P^{m+1}$  are located on the edge of the offset polygon  $P^{m'}$ ; 2) The edges of the next polygon  $P^{m+1}$  pass through the vertices of the polygon  $P^m$ .

#### 4.2. Analytical Solutions for Regular Polygons

With a regular  $n$ -polygon  $P^m$ , it is possible to analytically calculate the angle of rotation and the scaling ratio that maintain the next polygon  $P^{m+1}$  as a  $n$ -polygon. As shown in Figure 6, for an interior angle of the  $n$ -polygon  $\alpha$  and strip width  $w$ , the edge length  $l$  can be found. The system computes the angle of rotation  $\theta$  whereby the vertices of the polygon  $P^{m+1}$  lie on the edges of the strip.

The geometrical relations shown in Figure 6 can be expressed as follows:

$$\frac{L}{\sin\alpha} = \frac{l}{\sin\theta} = \frac{k}{\sin(\pi - \alpha - \theta)} \quad (1)$$

Equation(1) can be rewritten as

$$\begin{aligned} w &= k \sin\theta \\ &= \frac{\sin(\pi - \alpha - \theta)}{\sin\alpha} L \sin\theta \\ &= L \frac{\sin(\alpha + \theta)}{\sin\alpha} \sin\theta \end{aligned} \quad (2)$$

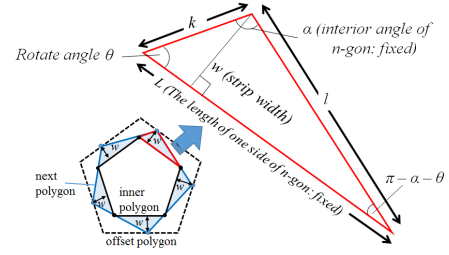


Figure 6: The iris folding geometry for a regular  $n$ -polygon.

When  $L = 1$ , we obtain the angle of rotation  $\theta$  as follows:

$$\begin{aligned} \theta &= 2\pi n - 2 \tan^{-1} \left( \frac{\sqrt{-4w^2 + 4w \cot \alpha + 1}}{2w} \right) \\ &\quad - \frac{\sqrt{\frac{\sqrt{-4w^2 + 4w \cot \alpha + 1}}{w^2} + \frac{1}{w^2} + \frac{2 \cot \alpha}{w}} + \frac{1}{2w}}{\sqrt{2}} \end{aligned} \quad (3)$$

where  $\frac{\alpha}{2\pi} - \frac{1}{2} \notin \mathbb{Z}$  and  $w \tan(\frac{\alpha}{2}) \neq 0$ . The length of a single side of polygon  $P^{m+1}$  is given by  $L_{m+1} = l + k$ .

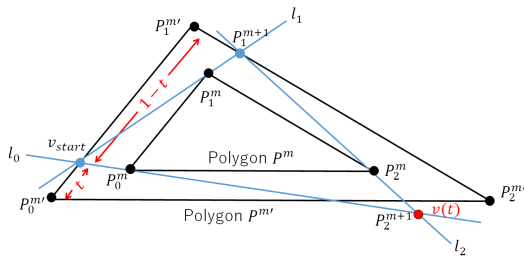
#### 4.3. Numerical Analysis for Arbitrary Polygons

Now we describe numerical analysis for an arbitrary  $n$ -polygon  $P^m$ . As shown in Figure 7, the vertices of an  $n$ -polygon  $P^m$  are defined as  $P_0^m, P_1^m, \dots, P_{n-1}^m$  in clockwise order. We put a point  $v_{\text{start}}$  on the edge  $P_0^m P_1^m$  by varying the parameter  $t$  in the range  $[0, 1]$ . The straight line  $l_1$  passes through  $v_{\text{start}}$  and  $P_1^m$ , and intersects the edge  $P_1^m P_2^m$  at point  $P_1^{m+1}$ . The straight line that passes through  $P_1^{m+1}$  and  $P_2^m$  is line  $l_{i+1}$ , which intersects the edge between  $P_{i+1}^m P_{i+2}^m$  at  $P_{i+1}^{m+1}$ . The straight line that passes through  $v_{\text{start}}$  and  $P_0^m$  is line  $l_0$ . The point where lines  $l_0$  and  $l_{n-1}$  intersect is  $v(t)$ . The intersection between the trajectory of  $v(t)$  and the edge  $P_0^m P_{n-1}^m$  becomes  $P_{n-1}^{m+1}$ .

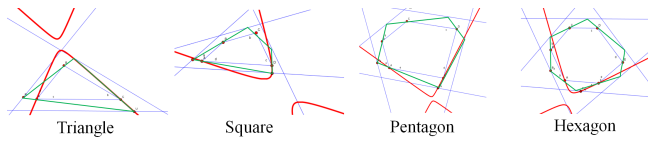
We tried to obtain an explicit formula to compute  $P^{m+1}$  from  $P^m$ , but found that it is overly complicated and difficult. So, we analyzed the problem numerically using an interactive geometry software (Cinderella [cin98]). By varying  $t$  in the range 0 to 1, the trajectory of  $v(t)$  can be found, as shown in Figure 8. We observe that the trajectories are conic curves, and therefore we can calculate the value  $t$  for which the point  $v(t)$  is located on an edge of  $P^{m'}$  using a bisection search. In Appendix, we show that such trajectory is a conic curve (quadratic equation) when the seed polygon is a triangle.

### 5. Algorithm

We calculate the next polygon  $P^{m+1}$  from polygon  $P^m$  using a bisection search. The system computes the signed distance from  $v(t)$  to the corresponding edge  $P_0^m P_{n-1}^m$ . When the user chooses clockwise pattern, we search for the parameter  $t$  between  $[0, 0.5]$ . When the user chooses counterclockwise pattern, we search for



**Figure 7:** Analysis of iris folding with arbitrary polygons. The aim is to find  $t$  such that  $v(t)$  is located on the edge  $P_0^{m+1}P_{n-1}^{m+1}$ .



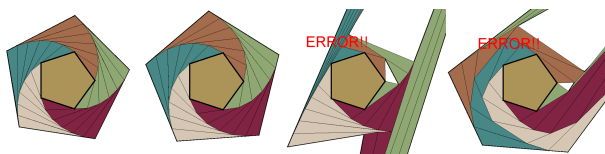
**Figure 8:** By varying  $t$  from 0 to 1, the trajectory  $v(t)$  was found (red line).

the parameter  $t$  between  $[0.5, 1]$ . After identifying the value of  $t$ , the system generates  $P^{m+1}$  using the value. This is repeated until  $P^m \supset P(S)$ , where  $P(S)$  includes the polygon of the inputted outline  $S$ .

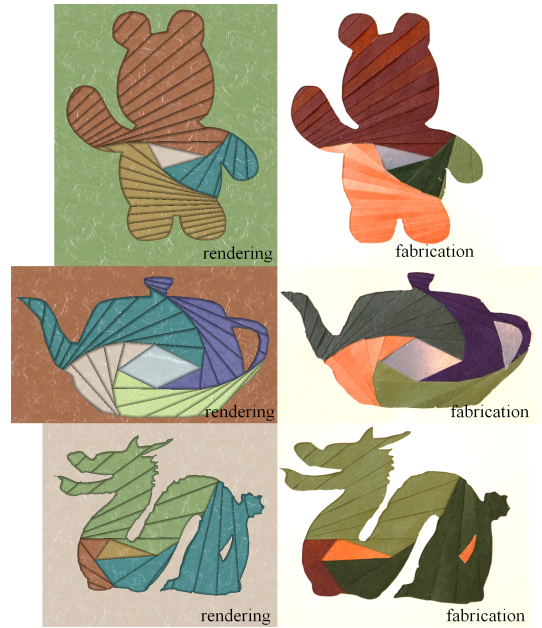
Because the trajectory is conic curve, a solution does not always exist. The existence of a solution depends on the ratio of the strip width to the edge length of the polygon (see Figure 9). If a solution cannot be found, the system does not draw a pattern, and the user searches for another solution by moving the vertex of the internal polygon or changing the strip width  $w$ . If a solution cannot be found, the system show the user to the fact that the problem was not solved.

**6. Results**

A prototype of the system was implemented using Java™ running on a laptop computer with a 1.2 GHz processor and 2 GB RAM. We used this system to create designs for iris folding pattern, as shown in Figure 10. Users can experiment with various patterns using the system prior to beginning construction. A design session typically takes approximately 10 – 20 min, and construction takes about 1 h.



**Figure 9:** The existence of a solution depends on the ratio of the strip width to the length of the edges of the polygon. Here the ratio  $w/l$  was varied from 0.29 to 0.33, 0.37, and 0.41 (left to right.) For  $w/l = 0.37$  and 0.41, a pattern could not be formed.



**Figure 10:** Design examples formed using our system.

**7. Conclusion and Future Work**

We have described an interactive design system for generating iris folding patterns. Users can quickly experiment with various patterns using the system before beginning the practical work.

As part of future work, we plan to develop a method to handle multiple internal polygons. We also plan to investigate patterns formed with other constraints, such as symmetry and similarity with the original polygon. We also plan to investigate conditions for the presence of the polygon  $P^{m+1}$  which circumscribes polygon  $P^m$  and inscribes polygon  $P^{m'}$ . It is also interesting problem to analyze a requirement about the shape of the internal polygon.

**References**

[CF05] COHRAN M., FIUME E.: Sketch-Based Design for Bargello Quilts. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2005), Jorge J. A. P., Igarashi T., (Eds.), The Eurographics Association. doi:10.2312/SBM/SBM05/165-174. 2

[cin98] Cinderella. the interactive geometry software, 1998. <http://cinderella.de/>. 3

[IM15] IGARASHI Y., MITANI J.: Patchy: An interactive patchwork design system. In *ACM SIGGRAPH 2015 Posters* (New York, NY, USA, 2015), SIGGRAPH '15, ACM, pp. 10:1–10:1. URL: <http://doi.acm.org/10.1145/2787626.2792601>, doi:10.1145/2787626.2792601. 2

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.* 21, 4 (Aug. 1987), 163–169. URL: <http://doi.acm.org/10.1145/37402.37422>, doi:10.1145/37402.37422. 2

[Pet01] PETERSON I.: 2001. <https://www.sciencenews.org/article/pursuing-pursuit-curves.2>