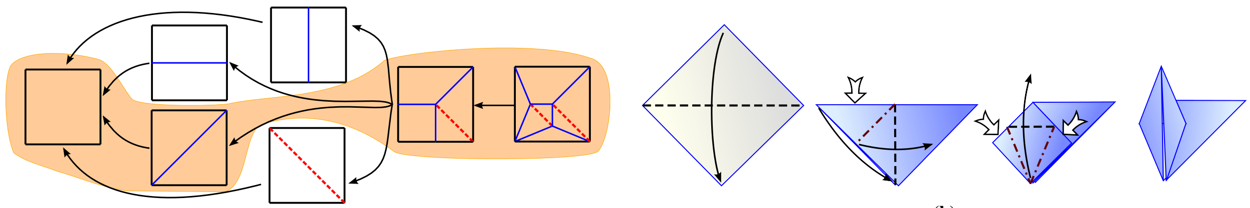
Generating Folding Sequences from Crease Patterns of Flat-Foldable Origami

Hugo A. Akitaya*
University of Tsukuba

Jun Mitani†
University of Tsukuba / JST ERATO

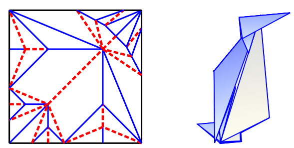
Yoshihiro Kanamori†
University of Tsukuba

Yukio Fukui†
University of Tsukuba



(a) (b)

Figure 1: *(a) Example of a step sequence graph containing several possible ways of simplifying the input crease pattern. (b) Results obtained using the path highlighted in orange from Figure 1(a).*



(a) (b)

Figure 2: *(a) Crease pattern and (b) folded form of the "Penguin". Design by Hideo Komatsu. Images redrawn by authors.*

1 Problem and Motivation

The most common form to convey origami is through origami diagrams, which are step-by-step sequences as shown in Figure 1b. They are composed by two graphic elements: images of the folded state of the paper and symbols, mostly lines and arrows indicating the position of the folds and the movement of the paper. Although the origami diagrams are relatively easy to follow, their production is very time-consuming, since it requires the drawing of several individual images to represent each step.

With the development of modern techniques of origami design, the complexity of achievable shapes have increased drastically and the crease pattern (the pattern of creases left on the paper after folding an origami model) has gained importance as an efficient method of documenting origami [Lang 2012]. A crease pattern shows the entire origami structure of folds in just one picture and, therefore, is easier to draw than the origami diagrams for the same model. In an origami technical design, it is common to draw the crease pattern before even folding the model in paper. An example of crease pattern and its folded form is shown in Figure 2.

The creases inside a crease pattern can be of two different types. A mountain crease is created by a fold that forms a ridge, making the paper convex in the process. A valley crease is created by a fold that forms a through, making the paper concave in the process. The type of a crease is relative, since a mountain crease is actually a valley when looked from the opposite side of the paper and vice versa. In this work, mountain creases are represented as red dashed lined and valley creases as blue solid lines.

The disadvantage of crease patterns is that it is difficult to use them to fold the paper into the origami design, since crease patterns show only where each crease must be made and not folding instructions. In fact, folding an origami model by its crease pattern may be a difficult task even for people experienced in origami [Lang 2012]. According to Lang, a linear sequence of small steps, such as usually portrayed in traditional diagrams, might not even exist and all the folds would have to be executed simultaneously.

We introduce a system capable of identifying if a folding sequence exists using a predetermined set of folds by modeling the origami steps as graph rewriting steps. It also creates origami diagrams semi-automatically from the input of a crease pattern of a flat origami. Our system provides the automatic addition of traditional symbols used in origami diagrams and 3D animations in order to help people who are inexperienced in folding crease patterns as well as automatizing almost completely the time-consuming task of drawing diagrams.

2 Background and related work

Origami mathematics is a subset of mathematics that investigates the mathematical properties of origami [Lang 2013]. Thinking about origami in the mathematical perspective was crucial to the development of origami as both art and science. Although origami can transform a 2-dimensional material (such as paper, metal sheets, leather, etc.) into a 3-dimensional form, most crease patterns are representations of **flat** origami. In many cases, the folded form described by the crease pattern is not the final model, but is the basic geometrical shape used to represent the desired subject that is called an *origami base*. A more detailed and expressive model can be obtained by adding 3-dimensional folds to the base, "sculpting" the paper into the desired form.

Because origami bases are usually flat, many researches have investigated the properties of flat origami [Hull 1994; Bern and Hayes 1996; Demaine and O'Rourke 2008]. A crease pattern is called *flat-foldable* if it can be flattened in its folded form without the addition of any extra crease. There are two famous necessary conditions of flat-foldability called *Maekawa*'s and *Kawasaki*'s theorems. The first states that, among the creases that intersect at a vertex, the sum of mountain creases minus the sum of valley creases must be either +2 or -2. The second condition enunciates that the alternate angles around a vertex must sum up to π. Both theorems, as well as the discussion about sufficiency, are addressed by Tomas Hull [Hull 1994] and Erik Demaine [Demaine and O'Rourke 2008].

Arkin et al. [Arkin et al. 2004] investigated simple folds in orthog-

*e-mail:hugoakitaya@npal.cs.tsukuba.ac.jp
†e-mail:{mitani, kanamori, fukui}@cs.tsukuba.ac.jp

onal grid of creases. A *simple fold* is a rigid rotation of some layers of paper around a single axis, which is the folding line. The simple fold can only be applied to flat folded states and map to another flat state, as shown in Figure 3a.

Some computer applications were developed using mathematical theories for flat origami such as ORIPA [Mitani 2007]. ORIPA is dedicated to editing origami crease patterns and calculating their folded form. It uses brute force to determine how the layers of paper overlap, which is a NP complete problem [Bern and Hayes 1996]. An application called Rigid Origami Simulator [Tachi 2009] animates a virtual paper, folding all the creases at the same time, until the folded state of the input crease pattern is reached. The faces of paper are considered as rigid panels and their positions are controlled by affine transformations [Belcastro and Hull 2002].

Origami mathematical theories are also widely used in engineering. From small stent grafts for medical usage [Kuribayashi et al. 2006] to huge solar panels used in space [Nishiyama 2012], origami based structures can be used as an efficient method to create transformable surfaces. Many of these researches also rely on crease patterns to represent folded states. The National Science Foundation (NSF) of the US have awarded, in 2012, 15 grants for projects in origami design in engineering (`http://www.nsf.gov/eng/efri/fy12awards_ODISSEI.jsp`).

3 Uniqueness of the approach

Although most of the research about crease patterns focus on mathematical properties or design, the folding process of a crease pattern still remain a hard task that is performed without practically any automation. In our approach, we consider that the input is a crease pattern of a flat foldable origami. The folded form of this crease pattern is called the *completed form*. The diagrams must show the paper in its initial state (without any folds) and successively show intermediary states until the completed form is reached. In origami diagrams, the intermediary states are obtained after the execution of folds. In our approach, we consider that the folds transform a flat folded form into another flat state. In order to find a sequence of folding steps that results in the completed form, we simplify the input gradually, using rewriting rules, until the initial state of the paper is reached. The folding sequence can then be obtained by inverting the order of the sequence generated by the simplification process. In other words, this is a reverse engineering approach that starts from the final step of the diagrams and generates an unfolding sequence that goes until the initial step, as shown in Figure 1a.

3.1 Extended crease pattern

When we analyze the changes that a simple fold makes in the crease pattern, it is clear that a single fold can produce several creases when the fold is applied to more than one layer. A simple fold always divides the faces where it is applied. Consequently, it will add as many creases to the crease pattern as many faces are divided. An example of simple fold and the changes it makes in the crease pattern is shown in Figure 3. A *complex fold* is a folding technique that performs more than one fold simultaneously. The folds that compose a complex fold cannot be executed independently and will be called *sub-folds* in this paper. An example of complex fold is shown in Figure 6. Each one of the sub-folds will also produce multiple creases if applied to multiple layers of paper.

In order to model folds in terms of graph theory, we propose a new graph structure to represent crease patterns: the *extended crease pattern*. This is a key part of our contribution. Due to space restrictions, this subsection will present a rough explanation and not the exact algorithm used to obtain this graph. Figure 4 shows the extended crease pattern of the crease pattern shown in Figure 3c.

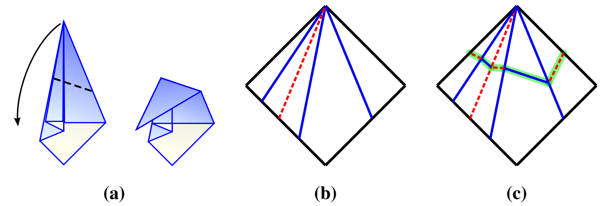


Figure 3: *(a) Example of a simple fold. (b) Crease pattern before and (c) after the simple fold is executed. The creases highlighted with green background compose a reflection path.*

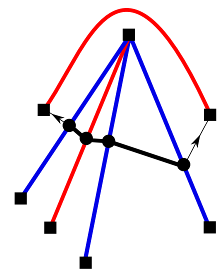


Figure 4: *The extended crease pattern for the example shown in Figure 3c.*

Our goal is to group creases that are originated from the same fold while preserving the topology of the original crease pattern. This will allow us to model origami folds as graph rewriting, as shown in Subsection 3.3. We identify creases that lie in the same position when the origami is folded based on Kawasaki's theorem. From this group of creases, based in Maekawa's theorem, we extract paths of creases that can be removed from the crease pattern without affecting the flat-foldability conditions of the vertices through which the path crosses. These paths are called *reflection paths*. In Figure 3c, the path highlighted in green is a reflection path. Each one of the other creases is a reflection path by itself.

The extended crease pattern is a directed graph, as can be seen in Figure 4. Edges that do not have an arrow to indicate their direction are actually bidirectional, i.e., they represent a combination of two edges with opposite directions. As can be seen in the picture, there are colored edges and black edges. The colored ones are called *active* and the others *inactive*. Active edges represent reflection paths, i.e., creases created from the same fold. Inactive edges mark the position of creases in the interior part of the reflection path maintaining the original topology. Vertices that lie on the edge of the paper are called *border vertices* and are represented as small squares. The other vertices are called *internal vertices* and are represented as small circles. The details on the algorithm and properties of the extended crease pattern can be found in [Akitaya 2014].

3.2 Crease pattern complexity

A group of creases created by the same fold is represented as just one edge in the extended crease pattern. We can, therefore, use them to analyze the occurrence and position of **folds** rather than just **creases** as shown in crease patterns. This allow us to identify the occurrence of simple and complex folds, as shown in Subsection 3.3. In this Subsection, we try to define an objective measurement for the complexity of a crease pattern based on its extended crease pattern.

We define that the complexity attributed to a simple fold must be zero. On the other hand, complex folds should increase the complexity of the crease pattern. At first, we look at each vertex of the extended crease pattern individually. The complexity, defined only for internal vertices, is the number of active edges going out of a vertex minus two. The sum of complexities of all internal vertices is then defined as the complexity of the crease pattern.

It is possible to prove that, using the above definition, if a crease pattern is foldable only using simple folds (*simply-foldable*) it has complexity equal to zero [Akitaya 2014]. Therefore, this definition can be useful to evaluate whether an origami is simply-foldable. The determination of simply-foldability can be useful in applications in sheet metal and paper product manufacturing [Arkin et al. 2004], since robotic folding is usually restrained to simple folds [Balkcom 2004].

Additionally, the execution of a certain complex fold alters the complexity of the crease pattern in a systematic way. For example, the execution of a complex fold called *squash fold* always adds two to the complexity of the crease pattern. An example of squash fold is given in the second step of the diagrams shown in Figure 1b and in Figure 6. The definition of complexity is another main contribution of our research. It is used in the simplification process to assure that the sequence is approaching the unfolded paper.

3.3 Unfolding with graph rewriting

Using the extended crease pattern, it is possible to describe the basic folds that are used in diagrams in terms of graph rewriting. Since the goal is to simplify the crease pattern, the rewriting rules actually models the *unfolding* action.

Let us now focus on the simplest form of fold: the simple fold. The creases created by a simple fold, such as the one shown in Figure 3a, create a reflection path connecting two border vertices. In terms of an extended crease pattern, the execution of a simple fold adds an active edge that connects two squares. The unfolding rewriting rules for the simple fold can be defined as shown in Figure 5a.

The rewriting rules are represented with undirected graphs with colored nodes in order to explicit the difference between the nodes. As in the extended crease pattern, nodes can be internal (circles) or border (square) nodes. There are two types of edges, one represented with solid lines and the other with dotted lines, to represent the two types of creases (mountain/valley). Instead of the color code, a different representation was used because these two types of folds are relative to which side of the paper is being viewed, as said in Section 1, and the same rule must be able to handle both cases.

An example of matching and rewriting of a simple fold is given in Figure 5c. Since the graphs of the extended crease pattern and the unfolding rewriting rules are different from each other, we need to define an isomorphism function in order to identify matchings and then perform the rewriting. The following list gives the most important characteristics of this isomorphism (a full description can be found in [Akitaya 2014]). (i) The left-hand side of the rule (*pattern graph*) must be matched with a subgraph of the extended crease pattern. (ii) Extended crease patterns are embedded graphs and, therefore, the order in which the creases occur around a vertex must be the same in both graphs. (iii) Inactive edges cannot be matched with edges of the pattern graph, because they do not represent an entire fold but only one of the creases created by a fold.

After the matching is identified, the isomorphic subgraph of the extended crease pattern must be rewritten according to the rewriting rule as shown in Figure 5c. Once the simplified graph is obtained, it is necessary to convert it to a crease pattern. All active edges must be mapped again to reflection paths. In the case of the example in Figure 5c, all remaining active edges are preexistent to the rewriting. Therefore, all creases of the simplified crease pattern will lie on the same positions as before. Therefore, the extended crease pattern shown in the right side of Figure 5c corresponds to the crease pattern shown in Figure 3b.

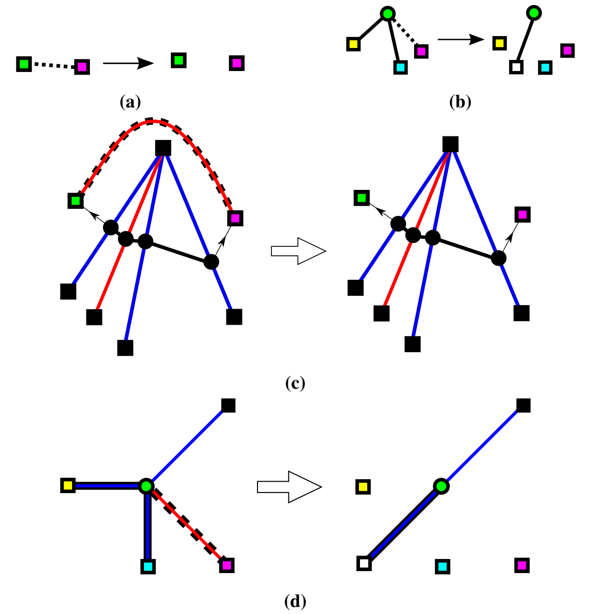


Figure 5: *Unfolding rewriting rules for (a) the simple fold and (b) the squash fold. (c) Graph rewriting corresponding to the unfolding of a simple fold. (d) Graph rewriting corresponding to the unfolding of a squash fold.*

The rewriting process that corresponds to the unfolding of a complex fold happens in a similar way as in the simple fold case. To exemplify a rewriting rule of a complex fold, we show, in Figure 5b, the rule for the squash fold. In this rule, three active edges parting from an internal vertex are substituted by one, thus, reducing the complexity of the crease pattern by two, as described in Subsection 3.2. A rewriting using this rule is exemplified in Figure 5d.

We identified the rewriting rules of the five most common folds that occur in origami diagrams, including the simple fold and squash fold. Our system can generate folding sequences containing those five folds. Although five folds may sound little, it covers a wide range of traditional, simple and intermediate origami models. Even if a model cannot be simplified with these rules, it is expected that the addition of more rewriting rules can allow the system to compute their folding sequences.

3.4 Step-sequence graph

More than one possibility of simplification may be possible at the same time. Sometimes there are multiples matchings for the same rule and/or matchings for different rules in the same extended crease pattern. In fact, there are usually several ways in which one can fold the same origami model. Therefore, there must be several ways to unfold it. In order to organize the unfolding possibilities, the simplified crease patterns are stored in the *step sequence graph*. An example of this graph is given in Figure 1a. It is expected that the successive simplifications of the crease pattern lead to the unfolded paper.

Any path that connects the input crease pattern with the unfolded paper is a possible unfolding sequence. The choice of an "optimal" path is subjective and depends on the experience of the user. For instance, a novice user might want a long and easy folding sequence while an experienced one prefers a short and efficient sequence. Since the possibilities of folding sequences are very big, due to a combinatorial explosion, even for simple models (the step sequence graph of the model shown in Figure 2b has more than 3,000 nodes) it is hard to show all of them simultaneously. For complex models it can be time-consuming even to compute all possibilities.

Our approach is to display the final model at first and the possible simplifications that can be directly applied to it (the nodes directly connected to it in the step sequence graph). When the user chooses the desired simplification, the system displays the possible simplifications from this chosen state. This approach allows both a personalized choice and a good computing time.

3.5 Diagram generation and 3D animations

In order to show the result of the simplification and export the results as origami diagrams, the system must be able to produce the figure of the folded form and the arrow and lines symbols used in traditional diagrams. To generate the folded form, we used the core of the ORIPA system. ORIPA receives a crease pattern as input and returns the shape and position of each layer of paper as well as their overlapping relation. From this information we are able to construct a vector drawing of the folded form. Additionally, small distortions are added to the position of the vertices of the folded origami in order to slightly show hidden layers and, thus, creating a more natural drawing.

The proposed system is able to produce some of the symbols that are commonly used in origami diagrams. We implemented three of the most common of them: folding lines, folding arrows and pushing arrows. Example of all these three elements can be found in the diagrams shown in Figure 1b.

Folding lines are the lines that indicate where the paper should be folded to complete the step. In this work, valley folds are marked with black dashed lines and mountain fold with dark red dot-dot-dash lines. Folding arrows are slim black arrows that indicate the movement of the paper. They are obtained by comparing the actual position of the each vertex between two sequential forms. Pushing arrows are fat white arrows used to indicate where the paper must be pushed. Usually, pushing a region of the paper aims at unfolding an existing fold. Therefore, this symbol is generated when a new crease appears in the crease pattern after the simplification.

Our system is also capable of animating the transitions form one folded state to another, based on the crease pattern of the two states. Our approach is very similar to the Rigid Origami Simulator [Tachi 2009]. The faces of the origami are considered initially rigid, but, unlike previous approaches, the system forces some faces to be stationary while permitting some deformation in order to animate the moving layers. Since some folds cannot be achieved with rigid folding, there may be self-intersections. The result is not mathematically accurate, but it portrays the folding process more naturally. Stills extracted from the animation of the second step of the diagrams shown in Figure 1b (squash fold) are shown in Figure 6. The animations can also help in the production of origami diagrams, since frames of the animation can be inserted between steps of the diagrams to provide additional detail (Figure 7b).

4 Results and contributions

The proposed method was implemented in a system that could generate semi-automatically diagrams for several traditional and some modern origami models. The result obtained by choosing the path highlighted in orange in Figure 1a is shown in Figure 1b. Using the crease pattern shown in Figure 2a as input, our system generated the diagrams shown in Figure 7a. Diagrams for the traditional paper crane are shown in Figure 7b.

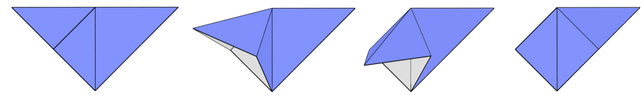


Figure 6: *Frames from the animation of a squash fold. The corresponding diagram is the second step shown in Figure 1b.*

The output of diagrams is in a vector drawing file format which allows easy manipulation and personalization. The symbols in the diagrams shown in Figure 7b were edited as follows. Using a third-party application, the arrow indicating that the origami was turn around was added after steps 3, 8, 12 and 19. Some folding arrows were deleted and 3D animation stills were added. These diagrams were created in approximately 3 minutes with just some mouse clicks to choose the folding sequence and the use of a vector graphics application to edit the final results as described.

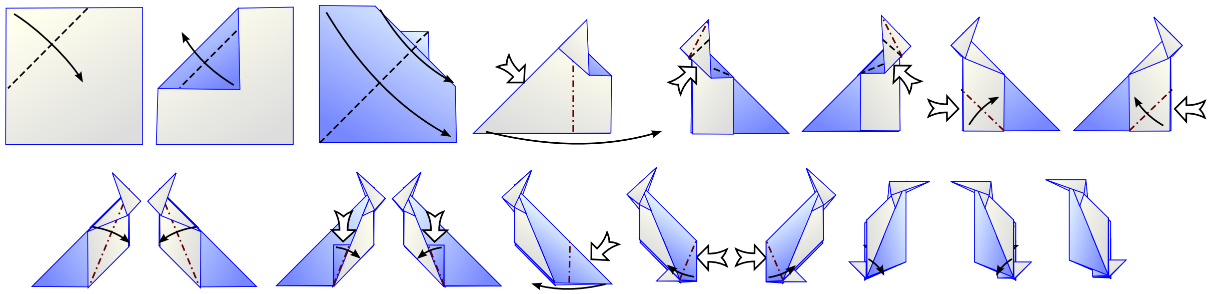
The implemented system used five simplification rules (simple fold and the four complex folds) as mentioned in Section 3.3. Depending on the complexity of the input model, the software is not able to simplify the crease pattern until it becomes the unfolded paper. If there are no matchings for any unfolding rewriting rule used by the system, the algorithm would stop. However, even in those cases it is capable of outputting diagrams for the final part of the folding sequence, connecting the state in which the algorithm stopped and the completed form of the origami (given by the input crease pattern).

The input of new unfolding rewriting rules can increase the range of crease pattern inputs that can be fully simplified using the proposed method. With more rewriting rules, the matching probability increases. By now, new rules can be registered in a text file. The implementation of a graphic interface, that allows this input, is a future work.

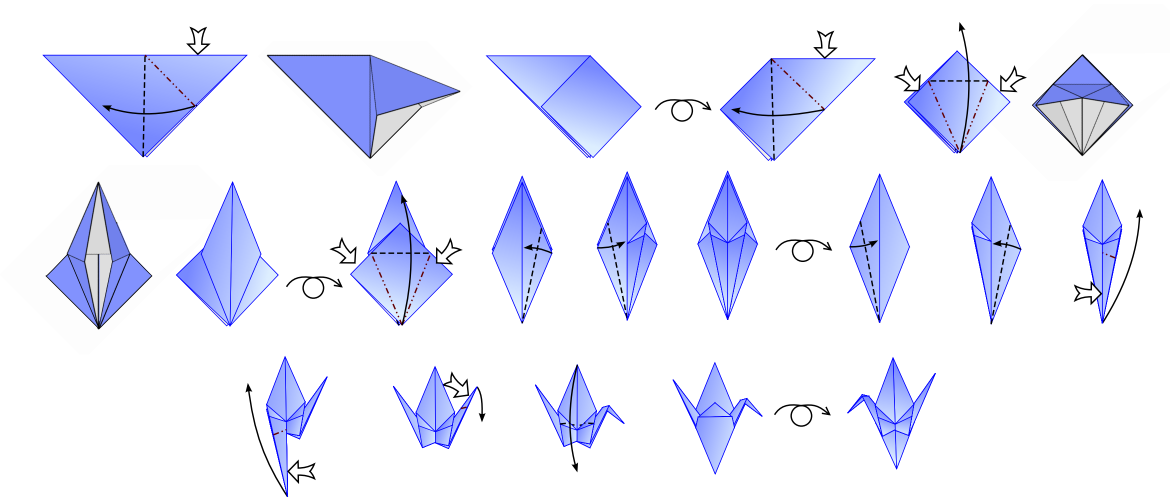
Although the simplified crease patterns are usually flat-foldable, there are some exceptions. The main reason is that Maekawa's and Kawasaki's theorems are necessary but not sufficient to guarantee the flat foldability. This drawback is addressed by checking the foldability with ORIPA. The wrong results are simply discarded.

The contributions of this work are organized in the following list. (i) The proposed system accelerates the time-consuming task of producing origami instructions with a novel method for generating diagrams, including the automatic addition of symbols that are commonly used. (ii) Our method helps finding a folding sequence to an origami pattern using a predefined set of origami techniques and, therefore, it can also be used to determine the feasibility of the pattern regarding the set of techniques. (iii) A new graph representation is proposed for crease patterns that allows better understanding of the relation between folds and creases. (iv) The proposed model of rewriting rules also helps to understand and catalog traditional and recurrent new folding techniques. (v) An objective method to quantify the complexity of origami models is proposed. (vi) Our method offers 3D animations for each step in the diagrams that can also be used in the creation of diagrams.

Our system is scheduled to be released publically soon and we expect that the addition of rewriting rules by users may allow the automatic generation of diagrams for complex origami designs.



(a)



(b)

Figure 7: *(a) Output for the penguin's folding sequence. The crease pattern shown in Figure 2a is used as input. (b) Diagrams for the traditional paper crane. The folding sequences were chosen by the authors.*

References

AKITAYA, H. A. 2014. *Generating Origami Folding Sequences from Flat-Foldable Crease Patterns*. Master's thesis, University of Tsukuba.

ARKIN, E. M., BENDER, M. A., DEMAINE, E. D., DEMAINE, M. L., MITCHELL, J. S., SETHIA, S., AND SKIENA, S. S. 2004. When can you fold a map? *Computational Geometry 29*, 23–46.

BALKCOM, D. 2004. *Robotic Origami Folding*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

BELCASTRO, S., AND HULL, T. 2002. Modelling the folding of paper into three dimensions using affine transformations. *Linear Algebra and its Applications 348*, 13, 273 – 282.

BERN, M., AND HAYES, B. 1996. The complexity of flat origami. In *Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, SODA '96, 175–183.

DEMAINE, E. D., AND O'ROURKE, J. 2008. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*, reprint ed. Cambridge University Press, New York, NY, USA.

HULL, T. 1994. On the mathematics of flat origamis. *Congressus Numerantium 100*, 215–224.

KURIBAYASHI, K., TSUCHIYA, K., YOU, Z., TOMUS, D., UMEMOTO, M., ITO, T., AND SASAKI, M. 2006. Self-deployable origami stent grafts as a biomedical application of ni-rich tini shape memory alloy foil. *Materials Science and Engineering: A 419*, 12, 131 – 137.

LANG, R. J. 2012. *Origami Design Secrets: Mathematical Methods for an Ancient Art. Second Edition*. CRC Press.

LANG, R. J., 2013. Origami mathematics. http://www.langorigami.com/science/math/math.php, Accessed in Nov 2013.

MITANI, J. 2007. Development of origami pattern editor (ORIPA) and a method for estimating a folded configuration of origami from the crease pattern. *IPSJ Journal 48*, 9 (sep), 3309–3317.

NISHIYAMA, Y. 2012. Miura folding: Applying origami to space exploration. *International Journal of Pure and Applied Mathematics 79*, 2, 269–279.

TACHI, T. 2009. Simulation of rigid origami. In *Origami 4, Fourth International Conference on Origami in Science, Mathematics, and Education (4OSME)*, A K Peters, 175–187.