

Rethinking Origami: A Generative Specification of Origami Patterns with Shape Grammars[☆]



Ying Yu^{a,b}, Tzu-Chieh Kurt Hong^a, Athanassios Economou^a, Glaucio H. Paulino^{a,*}

^a Georgia Institute of Technology, USA

^b Shantou University, China

ARTICLE INFO

Article history:

Received 1 April 2020

Received in revised form 25 October 2020

Accepted 13 March 2021

Keywords:

Origami
Shape grammar
Pattern Design

ABSTRACT

As a ubiquitous paper folding art, origami has promising applications in science and engineering. Many software and parameterized methods have been proposed to draw, analyze and design origami patterns. Here we focus on the shape grammar formalism and the Shape Machine, a shape grammar interpreter that has managed to automate the seamless shape calculations that the shape grammar formalism advocates. Different from other origami pattern generation methods, shape grammars generate origami patterns through recursive applications of shape rewriting rules using lines and curves. Based on this concept, the transformations between some common origami patterns are reorganized following visual cues and reasoning. Four examples of generating origami pattern are presented to show the capability of Shape Machine in origami design, including construction, modification and programming of an origami pattern. The new origami designs inspired by this work prove that shape grammars and Shape Machine provide a perspective and modeling technique for creating origami tessellation patterns.

© 2021 Elsevier Ltd. All rights reserved.

1. Introduction

Non-representational origami, e.g. origami tessellation, has a deep history in various fields including art [1], design [2], architecture [3], computational geometry [4] and mathematics [5]. Classic origami tessellations, such as the Miura-ori [6], the Resch pattern [7] and so forth, routinely appear in several pattern books showcasing focused studies in the architecture of form. A sample of such origami tessellations is shown in Fig. 1 all generated within the Shape Machine, the formal modeling system discussed in this paper.

The scientific study of the origami tessellation can be traced back to the study of the origami design, itself kicked off after the pioneering work of the computer scientist and artist Ron Resch [7,8] who began designing and folding paper forms using mathematical and computational algorithms. Since then, scientists and engineers have been following his paradigm and have established origami research as a growing body of research with its own structures, forms and mathematical laws governing origami design [4,5,9,10] as well as a growing number of novel applications in various fields and primarily in science and engineering. Examples of such applications of the geometry of origami includes the design of metamaterials [11,12], the DNA folding in nanotechnology [13,14], origami acoustic metamateri-

als [15], graphene folding in nanotechnology [16], deformable optoelectronic and thermoset shape-memory polymer applications [17,18], tunable structures [15,19], and deployable structures [20,21]. As the applications of origami design get broader exposure, advances in origami mathematics and origami computation technologies bring the contemporary design of such patterns to a new level of complexity.

Current software developed for origami design and analysis include the *Treemaker* [22], *Freeform Origami* [23], and *Merlin* [24], among others. The software *TreeMaker* is based on the powerful design techniques of circle-river packing. It can construct the full crease pattern for a wide variety of origami bases. The patterns constructed in *TreeMaker* are commonly the most efficient solutions possible for a given tree figure. *Freeform Origami* enables a freeform variation of rigid-foldable structure by introducing bidirectionally flat-foldable planar quadrilateral (BDFFPQ) meshes. It can also generate the initial pattern suited for almost any target 3D form [10]. *Merlin* is a software for nonlinear structural analysis of both rigid and non-rigid origami assemblages. Based on the nonlinear mechanics, it uses the bar-and-hinge model for origami discretization.

Powerful as they may be, these formal representations of origami models are primarily analytical or numerical, and they do not lend themselves easily in the creation of origami designs or the comparative generation and evaluation of classes of origami designs. Take for example, the criteria for folding of a flat pattern without self-intersections: Such rigidly flat foldable tessellations

[☆] This paper has been recommended for acceptance by M. Sitharam.

* Corresponding author.

E-mail address: paulino@gatech.edu (G.H. Paulino).

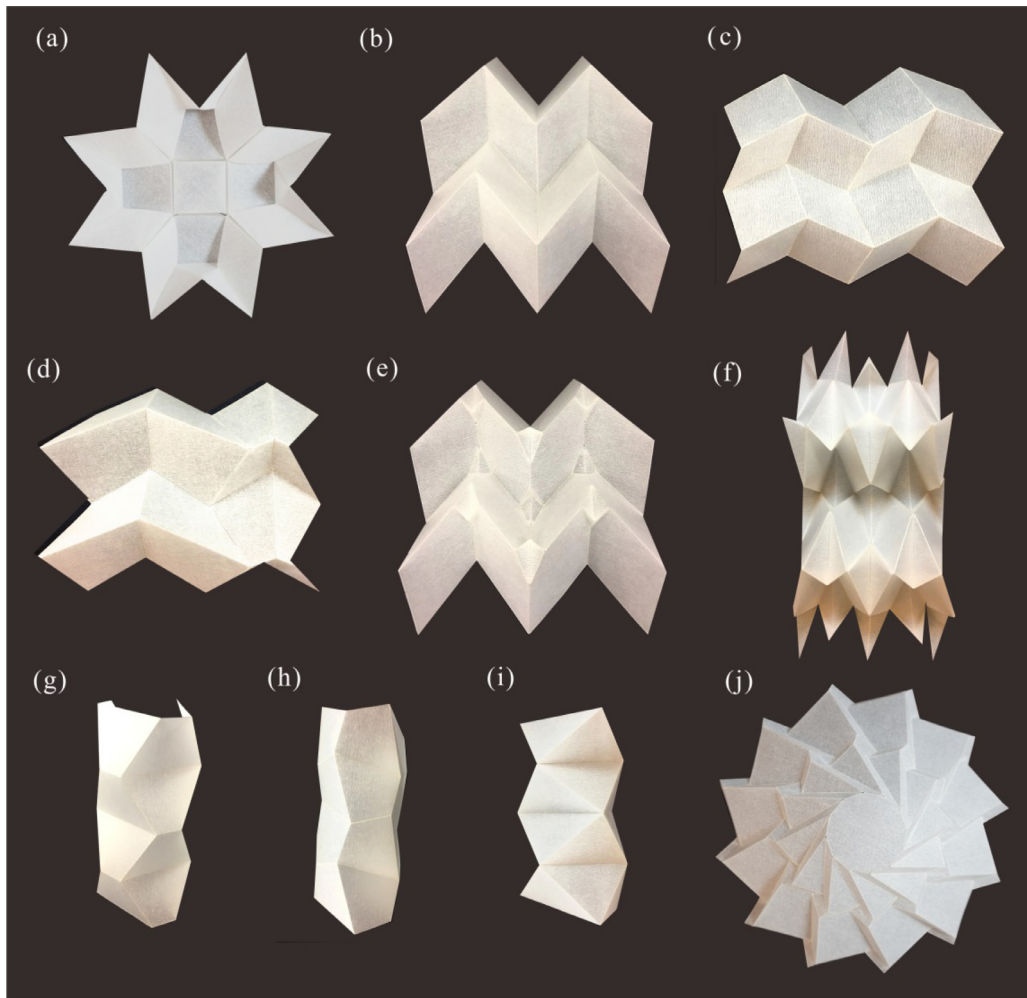


Fig. 1. Physical models of origami tessellations designed in the Shape Machine. (a) Flower pattern; (b) Miura-ori pattern; (c) Barreto pattern; (d) Quadrilateral mesh pattern; (e) Miura-ori pattern with gadget; (f) Gadget pattern; (g) Arc pattern or Chicken-wire pattern; (h) Huffman grid pattern; (i) Yoshimura pattern; (j) Dodecagon ring pattern.

can be evaluated using several theorems including the Maekawa–Justin Theorem, the Kawasaki–Justin theorem, the Even Degree Theorem, the Local Min Theorem and the fold-angle multiplier, all used to evaluate the foldability of a single vertex [4]. For a simple degree-4 vertex, all possible assignments of mountain and valley creases can be enumerated and tested if the given assignment is valid. However, the situation becomes much more complicated when the networks of creases consist of multiple vertices. Because different vertices may place contradictory conditions upon other creases, several collisions between layers of paper may happen. In this case, additional mathematical arguments such as the Justin Isometry Theorem, the Justin Non-twist Theorem, vector formulations and the local Flat-foldability Graph [5], may need to be deployed to decide whether locally flat foldability can be obtained. And still, there is no guarantee that the entire creased pattern folds flat without self-intersection. As the origami folding process involves non-rigid deformation and curve creases, the mathematical laws become much more complicated, and most of the times, no final analytical results can be found.

A very different approach emphasizing the constructive specification of origami designs employs symbolic grammars or rewrite production systems, such as the L-system (Lindenmayer system) [25], a production system that captures the generation of plant cells and self-similar fractals and periodic topology [26]. Specific classes of origami patterns, for example, the Heighway dragon curve origami pattern, can be generated by a set of parallel

rewrite rules in the manner of an L-system simulating a fractal-like folding of a strip of paper [27]. Still, useful as this specification may be, it requires facility with computer programming and remains inaccessible to those who are not familiar with Python or other programming languages.

A different approach focuses on the comparative study of the geometry of classes of origami designs and tessellations [28]. This approach gives an organization of several rigidly foldable origami tessellations and examines how they are related. Because of the periodicity of origami tessellation, only one or two basic units of an origami pattern need to be illustrated and the rest can be derived from them: ‘Vertex mirror symmetry’, ‘Vertex inversion’ or ‘collapse to degree six vertices’ are frequently mentioned to transform one pattern to another [28]. Still, to designers without an engineering background, it is difficult to clearly understand the transformation through geometric modification. A typical transformation from an Arc pattern to a Miura-ori pattern is shown as an example, in Fig. 2. Both patterns are well defined in origami mathematics to ensure the rigidly flat foldability of the plane [9]. Based on the given geometry, the injunction ‘Add vertices inversion’ means that the points A, B and C in the Arc pattern should be moved to the points A’, B’ and C’, which are symmetric about an axis MM’. This also means that the corresponding mountain and valley assignments have to be changed. In all, the representation, interpretation and evaluation of such new origami designs through this geometric method still require a deep sophistication

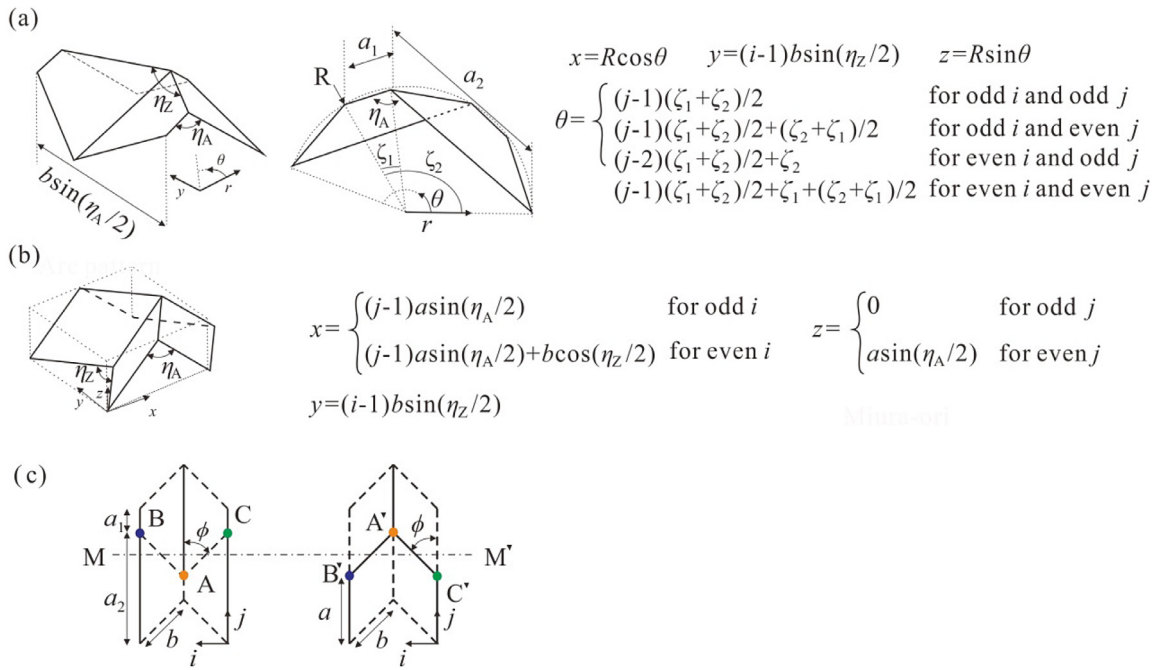


Fig. 2. Transformation from Arc patterns to Miura-ori patterns (a) Geometry of Arc pattern; (b) Geometry of Miura-ori; (c) Add 'Vertex inversion' from Arc pattern to Miura-ori. Source: Credit: Parts (a) and (b) of the figure are adapted from Gattas et al. [9].

in geometric modeling techniques and the validity of the results is not guaranteed. We need an alternative perspective to rethink origami pattern.

The methods mentioned above generate origami patterns, mathematically and parametrically. These methods are powerful and efficient, but require complex modeling methods and a steep learning curve to understand the appropriate tools and methodologies. More detrimentally, any changes and modifications to the original design require a complete restructuring of the whole pattern. As the origami system becomes non-flat foldable and non-rigid, the design process becomes cumbersome, difficult and uncertain. Still, since rational thought cannot predict everything in advance, seeing and drawing may work perfectly for origami pattern generation. In other words, if we consider origami patterns as shapes (arrangements of lines and arcs), the description, interpretation and evaluation of origami design can all be done by visual reasoning. Here, shape grammars [29–33] are adopted to design origami patterns.

Shape grammars are production systems using shape rewriting rules to perform computations with shapes. Their uniqueness with respect to all formal approaches in computational design is that they operate exclusively with shapes rather with some other symbols i.e. numbers, text, or symbolic instructions in some programming language. Their formidable formalism relying on the algebras of shapes U_{ij} , the algebras of labeled shapes V_{ij} , and the algebras of weights W_{ij} , for $i \leq j$ and $j \leq 3$, and the intuitive construct of a visual rule in the form of a pair of shapes, labeled shapes or weights defined in any of the algebras above, have provided a strong foundation for formal studies in design, and an unwavering commitment to visual reasoning. Their very reliance on visual aspects of form – and the ways these enter in visual calculations – has made them one of most accessible and intuitive formalisms to use in formal studies in design. Several applications have been developed in various fields over the years including architectural design [34], landscape architecture [35], engineering [36], painting [37], furniture design [38], ornamental design [39], origami design [40], and others. Still, this resolute commitment of shape grammars in shapes and shape rules does

not come without its toll: shapes consisting of lines, planes and solids fuse every time they are combined in a calculation to produce new and unexpected results and despite a long and sustained effort to produce a shape grammar interpreter to perform in a computer system the shape calculations that are possible with a pencil and a paper, all efforts have been inconclusive.

A sophisticated, but ultimately severely constrained, approach using a combinatorial calculation of the boundaries of the shapes to predict emergent shapes, has provided some useful applications, albeit all failing to provide a general solution to the problem [41]. Still, the situation is not as grave as it may seem. The single major obstacle to take on is the implementation of shape embedding [42], that is, the implementation of the mathematical concept of the “part relation” between two shapes, and it appears that the riddle has finally been solved: the very first software technology that has successfully managed to put the formalism in practice is the Shape Machine [43], a new computational technology that features a new way to specify the way geometric shapes are digitally represented, indexed, queried and operated upon. In this work, existing and new origami patterns are modeled, modified and programmed in Shape Machine and along the way some preliminary thoughts are discussed on the future of this technology in origami design.

2. Shape grammars and Shape Machine

A shape grammar performs computations by applying shape rules. A computation in a shape grammar begins with a starting shape called the initial shape. A shape rule has the form $A \rightarrow B$, which means that a shape A is rewritten as a shape B . A shape rule $A \rightarrow B$ is applied to a shape C , when there is a geometric transformation t that makes the shape $t(A)$ part of the shape C – or alternatively, when there is a transformation t that embeds the shape $t(A)$ in C . The resulting computation identifies the instance of the shape $t(A)$ in the shape C and replaces it with the corresponding instance of the shape $t(B)$ to generate a new shape $C' = C - t(A) + t(B)$. A shape computation is a sequence of shapes

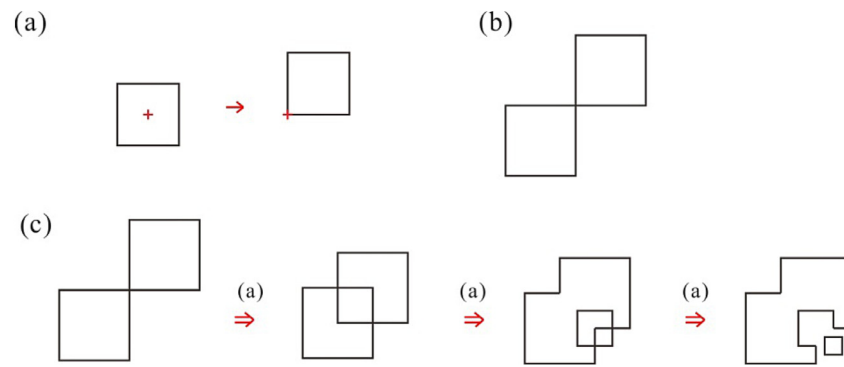


Fig. 3. Shape rules and shape computation. (a) Shape rule; (b) Initial shape; (c) Shape computation.

in which each shape, except for the first shape, is generated from the previous shape by shape rules.

A shape computation based on one shape rule applied multiple times is shown in Fig. 3. The rule in Fig. 3(a) specifies that a square A is rewritten as a translated square B along a distance equal to the half of its diagonal $\sqrt{2}/2$. The initial shape C in Fig. 3(b) consists of two squares in a spatial relation sharing a vertex along their colinear diagonals. The shape rule applies in Fig. 3(c) under an isometric transformation t_i to pick up the upper left square $t_i(A)$ and replace it with the square $t_i(B)$ that lies in a distance $\sqrt{2}/2$ towards its lower left side of the original square $t_i(A)$. The new shape C' consists now of three squares, the original two plus a new emergent smaller square in the center. The shape rule applies again in C' but this time around under a similarity transformation t_s to pick up the emergent smaller central square $t_s(A)$ and replace it with the square $t_s(B)$ that lies in a distance $\sqrt{2}/2$ towards the lower right side of the square $t_s(A)$. The new shape C'' consists now of two squares, the smaller one that emerged after the first application of the rule (a) and an even smaller square that emerged after the second application of the rule (a) — along with some more exotic shapes including an octagonal concave polygon and an L -shape among others. The shape rule applies yet again in the shape C'' under a similarity transformation $t_s(A)$ to pick up the emergent smaller lower right square $t_s(A)$ and replace it with the square $t_s(B)$ that lies in a distance $\sqrt{2}/2$ towards the lower right side of the square $t_s(A)$ and to produce a shape C''' where all squares, except the very last $t_s(B)$, have vanished. Clearly, there are several options about the specific transformation under which the rule may apply and this choice is typically made by the user during the design process. In shape grammar productions when a specific result is targeted, the shapes are typically labeled in specific ways to disambiguate their symmetries and the ways they interact, and the shape calculations are made in algebras of labeled shapes [44].

Note that during the shape computation in Fig. 3, new shapes are created ad hoc and they can all be found and transformed by applying a shape rule during the computation. As in visual computation any combination of shapes produces a complexity of different topologies and shapes that were not used in the original combination. Although it is straightforward to accomplish the shape computation by human vision, it is very challenging to implement shape grammar by machine. Finding the embedded shape and enabling parametric rules are two fundamental problems. Specific software designed to deal with these issues include the *SGI* [45], *GEdit* [46], *Grape* [47], and *Sortal* [48]. However, all these shape grammar interpreters are relying on commercial software for representing the data structure of the shape and because of the multidimensional data structures of these libraries, the programming of shapes in terms of these alternative data structures is not straightforward and not conducive to a general

solution for shape computation. This problem has been taken into account in Shape Machine, featuring a new data structure fully supporting shape recognition for all Euclidean transformations.

Shape rules can be generalized to parametric shape rules and even more to rule schemata of the form $x \rightarrow y$ consisting of predicates, variables and sets of variables [29]. The significance of predicates and variables becomes evident when a rule is given in a recursive form. More specifically, if the variables of the parametric shapes x and y are associated through some design operation f , then y becomes a function $f(x)$ and the rule schema can be rewritten in the form $x \rightarrow f(x)$ for f any operation in design. Examples of design operators include the transformation operation T , the boundary operation B , the part operator P , the division operator D and so forth, and they can all be combined with one another through compositions and products to yield complex symbolic schemata capturing compositional processes in design [32]. There is a nice symmetry between the two approaches: for example, the rule (a) that substitutes a square A with a square B in Fig. 3 can be understood as a shape rule cast in the rule schema $x \rightarrow T(x)$ for T a translation. Note though that the transformation T in the rule schema is different from the transformation t under which a rule applies.

3. Rethinking origami pattern using shape grammars and Shape Machine

There exists a wide audience of designers who are eager to engage with the expressive medium of folding and origami construction. Here, a visual recursive approach based on shape grammars is suggested to provide an alternative method for constructing origami designs that potentially can fill in the gap between visual design and mathematical analysis and seamlessly integrate computer models with evaluation modules in mathematical analysis. Visual reasoning cannot guarantee the rigidity or flat foldability of a surface, but the patterns generated in the shape machine can be readily exported and tested in origami software and imported back again in the shape machine for a new design cycle of generation and evaluation. Both non-rigid and curve folding may be involved in those new patterns. In a way, what is proposed here is an alternative construction of the 'learning by doing' approach [8], whereas existing and new origami designs can be described, interpreted and evaluated. A series of brief modeling studies follows below to begin to explore this territory.

3.1. Modeling an origami tessellation in Shape Machine

The first study takes on the formal specification of an existing origami tessellation in terms of shapes found in the tessellation and spatial relations between these shapes. The given tessellation,

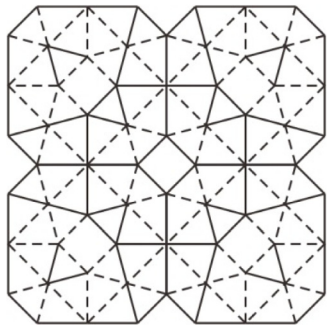


Fig. 4. An unfolded origami tessellation.

when folded, evokes the geometry of a flower and consists of a set of solid and dotted lines simulating the characteristic features of mountains and valleys of an origami construction. The geometry of this foldable origami tessellation is given in Fig. 4.

The formal specification of this origami tessellation is given in the form of a shape grammar modeled after of the ice-ray grammar [49], a type of shape grammars that have been designed to capture the structure of traditional Chinese lattice designs [50]. The original ice-ray grammar generates a periodic structure consisting of squares, each exhibiting alternating dihedral symmetry of order 2, and then modifies this structure with a spatial motif. Here, this ice-ray grammar is reworked to produce a regular lattice design to comply within the conventions of origami design. In Fig. 5, the labeled shape rules (1)–(5) for the addition of the lattice squares remain identical with the initial ice-ray grammar while the initial labeled shape of a 4×4 grid is slightly smaller than the original one. The new rule in the grammar is rule (6), a rule that complies with the condition of foldability of origami design featuring two lines on the RHS of the rule drawn as dash lines to fulfill the ‘Maekawa–Justin Theorem’, ‘Kawasaki–Justin theorem’, and the ‘Even Degree Theorem’. The two different types of the labels of the origami grammar are used in the shape rules to ensure the desired transformation under which the rules apply and the right ordering of the execution of the rules. Applying shape rules (1) to (5) in the schema $x \rightarrow x + T(x)$, that is, applying rules one by one adding each time to the square (x) of the left-hand-side (LHS) of the rule a reflected copy $T(x)$ to the right-hand-side (RHS) of the rule, produces a periodic design constructed in a boustrophedon manner from left to right in the first row, right to left in the second row, left to right in the third row, and finally right to left in the last row. The rule (6) applies in the schema $x \rightarrow \sum T(x)$ to the complete set of available labeled squares in the lattice and generates the complete design.

Note that the application of rule (6) in the schema $x \rightarrow x + T(x)$, could not produce this design because the structure of the lattice of the squares would have been changed after the first application of the rule and it would have been impossible to continue the execution of the labeled shape rules. Still, despite the apparent visual interest of the generated lattice, the particular design generated by rule (1) to (6) does not qualify yet as an origami design because as is, it cannot be folded. In origami design, the single vertex foldability does not guarantee global foldability. The assignment of mountains and valleys is a ‘NP-hard’ problem [4]. Rule (7) is introduced to change some assignments of the mountain and valley labels in the design and to produce the origami design which can be folded to a flower-like configuration. The shape rules, initial shape and the production of the flower origami tessellation are given in Fig. 5. A physical model of this origami flower is shown in Fig. 1(a).

3.2. Modeling classes of origami tessellations in Shape Machine

The second study takes on the formal specification of classes of origami tessellations through the comparative deformations of their geometrical periodic units [28]. These comparative specifications include transformations of one origami to another through a ‘vertex mirror symmetry’, ‘vertex inversion’, ‘collapse to degree six vertices’ and more. A series of origami tessellations featuring such invariant periodic units is shown in Fig. 6. It is proposed here that these origami patterns can be generated using only visual reasoning without explicit mathematical formulae, and even more, that these origami patterns can all be visually generated from transformed versions of the ice-ray lattice grammar used in the first study.

The Miura-ori and the Arc pattern can be generated by a straight forward variation of the original ice-ray lattice grammar featuring a different labeling scheme of the basic structural shape rules for the generation of the underlying square lattice and different terminating shape rules. The shape rules, initial shape, and the production of these two origami patterns are given in Fig. 7. In this new grammar, the labeling of the square reduces its symmetry to 1 – the identity symmetry element of the cyclic group of order 1. This reduction of symmetry is achieved here by the simple translation of the label in any other portion of the shape of the square such that it will not overlap to any of the 8 elements of the symmetry group of the square [51]. The resulting effect is that the symmetry of the labeled square becomes equal to 1. The rest of the rules of the shape grammar follow the same structure as the initial one enumerating all possible ways of adding a reflected copy $T(x)$ of the square x on the LHS of the rule to the RHS of the rule and erasing the terminating label of the initial shape. The two key rules that capture the structures of the Miura-ori and the Arc pattern are the rules (6) and (7) in Fig. 7(a), respectively. Note that the shape rules of the grammar (1)–(5) apply as before in the schema $x \rightarrow x + T(x)$ while the rules (6) and (7) apply in the schema $x \rightarrow \sum T(x)$ to the complete set of available labeled squares in the lattice. The physical models of the Miura-ori and the Arc pattern are shown in Fig. 1(b) and (g), respectively.

For the Barreto pattern, as many as four labels have been used. The rules of the Barreto pattern grammar are necessarily more than the previous ones to capture both the emergent spatial relations between the labeled squares and the different terminating spatial motifs, here consisting of one pair of squares and one pair of rhombi. As before, the shape rules of the grammar (1)–(7) apply as before in the schema $x \rightarrow x + T(x)$ while the rules (8)–(10) apply in the schema $x \rightarrow \sum T(x)$ to the complete set of available labeled squares in the lattice. The shape rules, initial shape and the production of the Barreto pattern are given in Fig. 8(a), (b) and (c), respectively. The physical model of this pattern is shown in Fig. 1(c).

The quadrilateral mesh origami grammar can be generated along the same lines of the Barreto pattern using a variation of the original ice-ray lattice grammar including four different labeling schemes for the basic structural shape rules and four different terminating shape rules. The four labels of the squares here are disposed in yet one more setting to account for the emergent symmetry properties of the tessellation and for the bookkeeping of the portions of the shapes to be deleted. The terminal rules for the four labeled squares here feature four distinct quads with specific geometric features to account together for the specific properties of the emergent translated concave decagon upon a pair of squares. As before, the shape rules of the grammar (1)–(9) apply as before in the schema $x \rightarrow x + T(x)$ while the rules (10)–(14) apply in the schema $x \rightarrow \sum T(x)$ to the complete set of available labeled squares in the lattice. The shape rules, initial

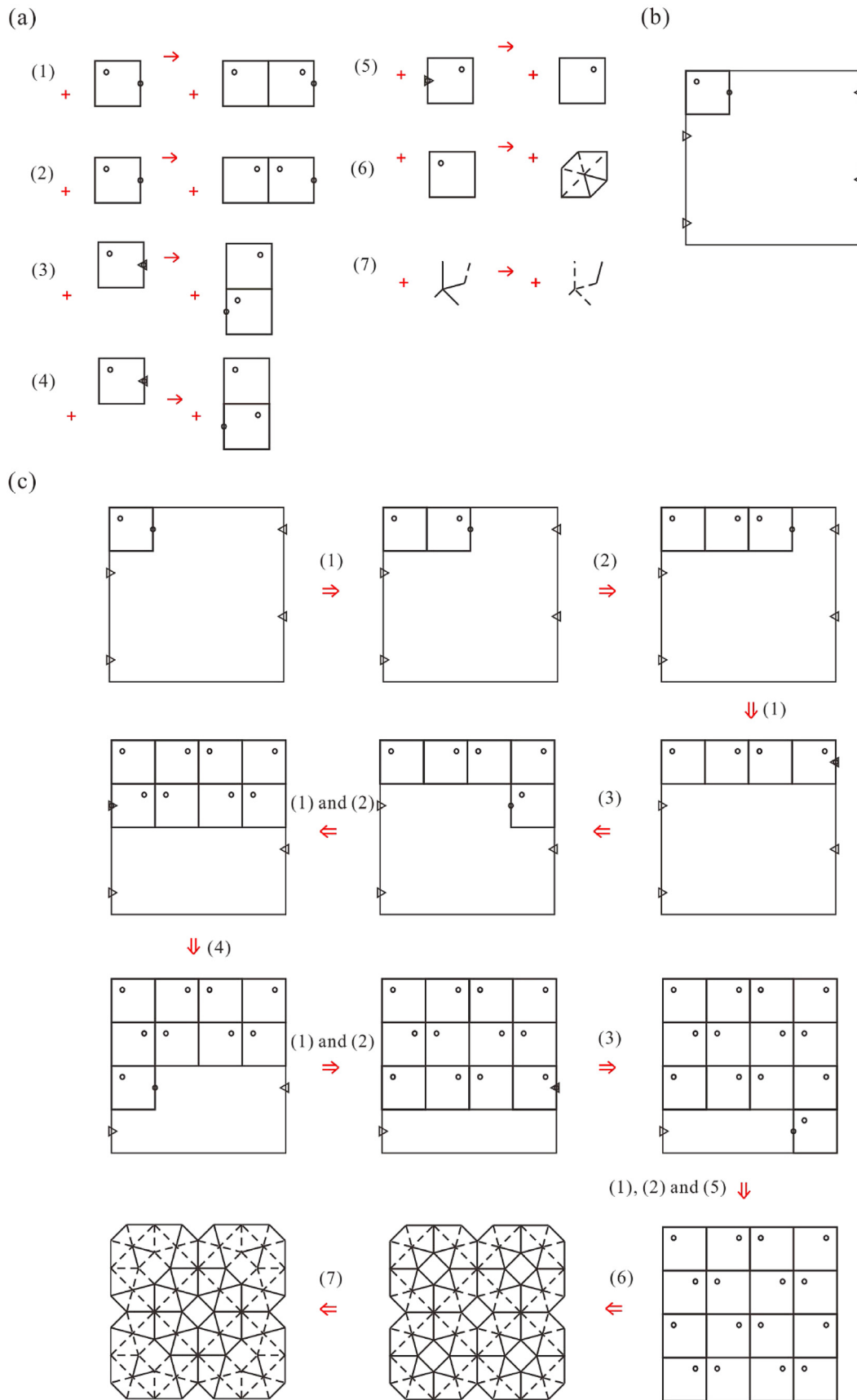


Fig. 5. An origami flower. (a) Shape rules (1) to (5) generate the underlying grid, and shape rules (6) and (7) replace the grid with the new motifs; (b) Initial shape; (c) Production of the flower pattern.

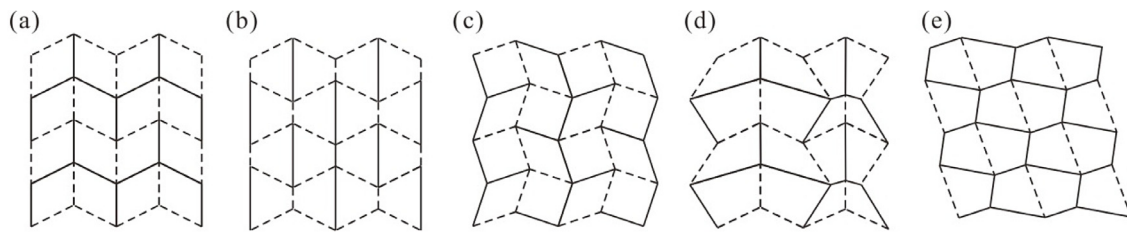


Fig. 6. Five origami tessellations featuring transformed variations of an underlying quadrilateral. (a) Miura-ori pattern; (b) Arc pattern; (c) Barreto pattern; (d) Quadrilateral mesh pattern; and (e) Huffman grid pattern.

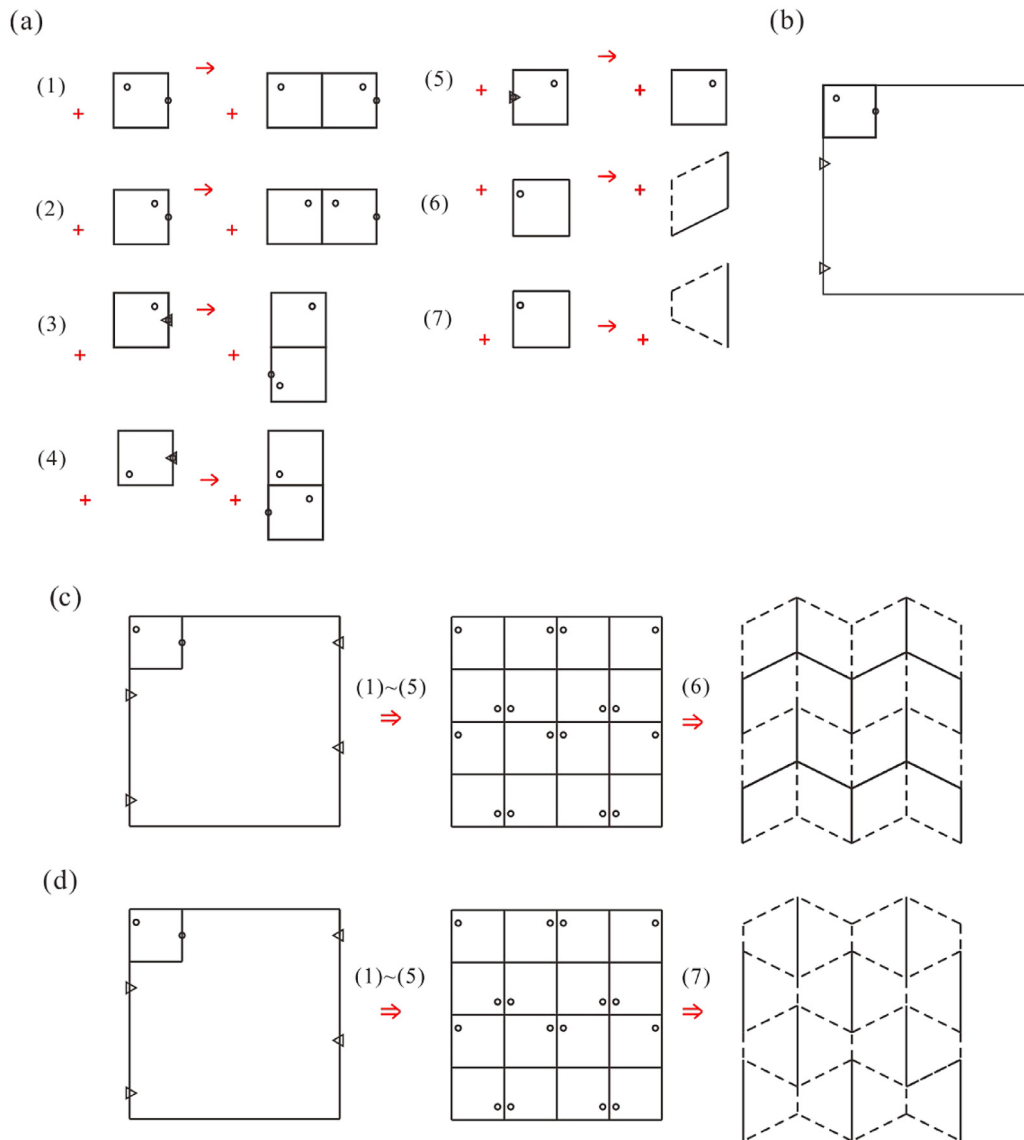


Fig. 7. Two origami tessellations featuring an identical configuration of an underlying lattice square grid. (a) Shape rules (1) to (5) generate the underlying grid, and shape rules (6) and (7) replace the grid with the new motifs; (b) Initial shape; (c) Production of Miura-ori pattern and Arc pattern.

shape and the production are given in Fig. 9. The physical model of the quadrilateral mesh pattern is shown in Fig. 1(d).

The Huffman grid pattern presents a more interesting problem in the sense that the glide reflectional symmetries of the original underlying square lattice need to be dropped to accommodate the rotational repositioning of the four versions of the quadrilaterals that all have symmetry of 1 – the identity symmetry element of the cyclic group of order 1. In this sense, the recursive generation of a labeled square lattice to provide a labeled configuration for

the application of terminating labeled shape rules would not do it. Instead, we need a different underlying lattice – here, a parallelogram lattice or a rhombic lattice that will be labeled in such a way so that the reflections of the rhombi will be dropped and the overall pattern will show local and global half-turn rotations. This transformed version of the original ice-ray grammar suggests a whole new way to look at the recursive generation of origami tessellations by combining lessons learned from the geometric transformations between origami patterns with the symmetry

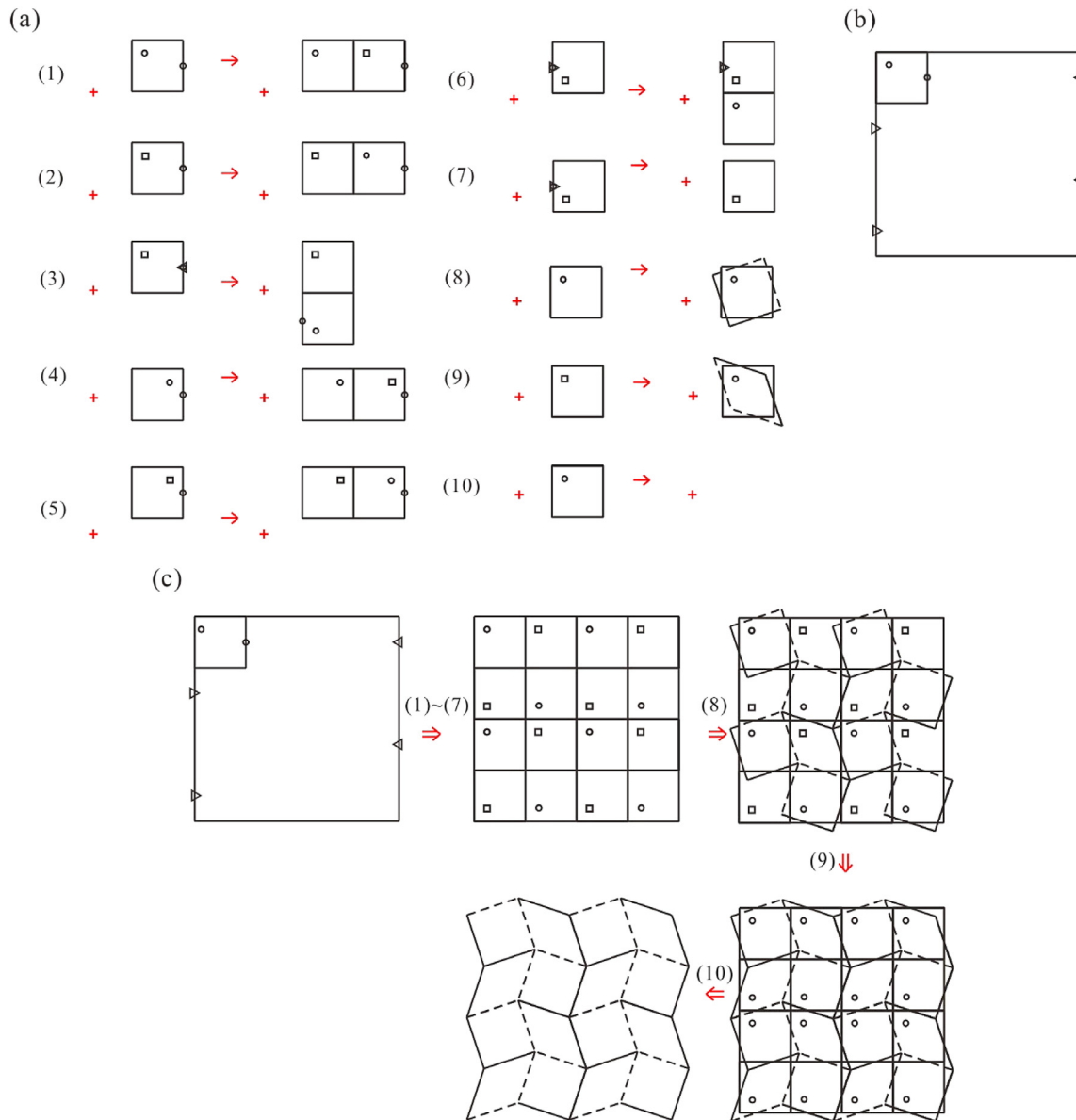


Fig. 8. The shape grammar for generating Barreto pattern. (a) Shape rules (1) to (7) generate the lattice square grid; shape rules (8) and (9) replace the grid with new motifs and shape rule (10) deletes the underlying grid; (b) Initial shape; (c) Production of the Barreto pattern.

theory of wallpaper configurations and even more with the five regular systems of lines [52] and the corresponding five Dirichlet domains of the two-dimensional Euclidean space [53]. The transformational ice-ray grammar, initial shape and production of the Huffman grid pattern are shown in Fig. 10. The physical model of the Huffman grid pattern is shown in Fig. 1(h).

3.3. Transforming origami designs using Shape Machine

The real power of a shape grammar specification of an origami tessellation in the Shape Machine appears in the editing of the model in a CAD system. Currently, any modification of an origami model in a CAD system is bounded by the parametric definition of the model itself. Any modification of an element or a feature of the model not specified in the system requires the redefinition of the complete model. Although a human can use visual reasoning to see emergent possibilities in a design process, it is impossible for a machine to do so with a pre-defined data structure. Shape Machine is a geometric modeling system that can achieve these

changes of data in design flows and execute commands without pre-definition of these parts of the shape in the initial structure. Moreover, Shape Machine can bundle any series of executable shape rules effectively proposing a completely new programming language to generate new shapes that can readily be tested for compatibility in origami design engineering. A series of three computations is given below to show the usage of Shape Machine in the editing of origami designs in an increasing degree of complexity.

The first example takes on the possibility of defining an origami pattern in terms of an existing one. This modeling technique differs significantly from the previous ones discussed so far that produced origami patterns from scratch in terms of an underlying labeled tessellation and substituting shapes consisting of solid and dotted lines to produce foldable designs. Moreover, this transformative modeling is not trivial because the transformation of an origami to another one often requires a change of the topology of the corresponding underlying patterns of the origami designs – a formidable task for current origami modelers if these instructions are not explicitly encoded in the system. An

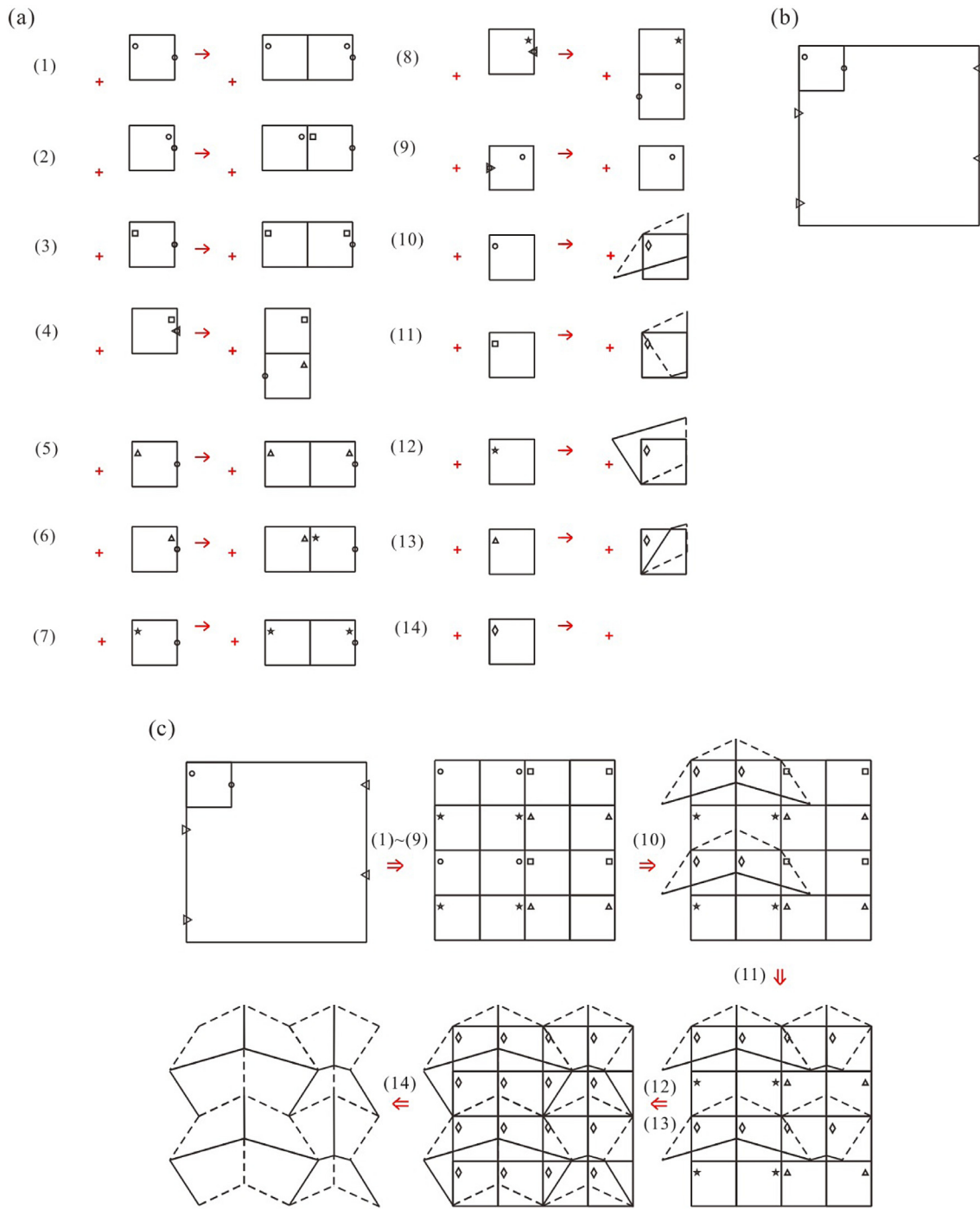


Fig. 9. The shape grammar for generating quadrilateral mesh pattern. (a) Shape rules (1) to (9) generate the lattice-square grid, shape rules (10) to (13) replace the grid with new motifs, and shape rule (14) deletes the underlying grid; (b) Initial shape; (c) Production of the quadrilateral mesh pattern.

example to illustrate this seamless transformation of the topology generation in Shape Machine is given below in Fig. 11. In this example, the Arc pattern is used to generate in a transformational and generative way the Yoshimura pattern. As mentioned above, Evan et al. [28] describe this process as ‘collapse to degree six vertices’, which means that a four-crease vertex becomes a six-crease vertex. Applying the shape rule (1) shown in Fig. 11(a) to the Arc pattern in Fig. 11(b) in the schema $x \rightarrow \sum T(x)$ replaces the set of quadrilaterals in the Arc pattern with the corresponding set of isosceles triangles to produce the Yoshimura pattern. The prototype of Yoshimura pattern is shown in Fig. 1(i).

The challenge of defining an origami design from one type of pattern to another becomes significantly harder when the new transformed origami pattern has points and edges that are not registered in the database of the original one. Two examples are given below to show how an existing origami design can be seamlessly modified without interfering with the underlying structure of the origami model.

The first computation proposed is based on the existing definition of an “origami gadget” as a localized section of crease pattern that can replace an existing patch to add functionality or otherwise modify the pattern [28]. The geometric conditions

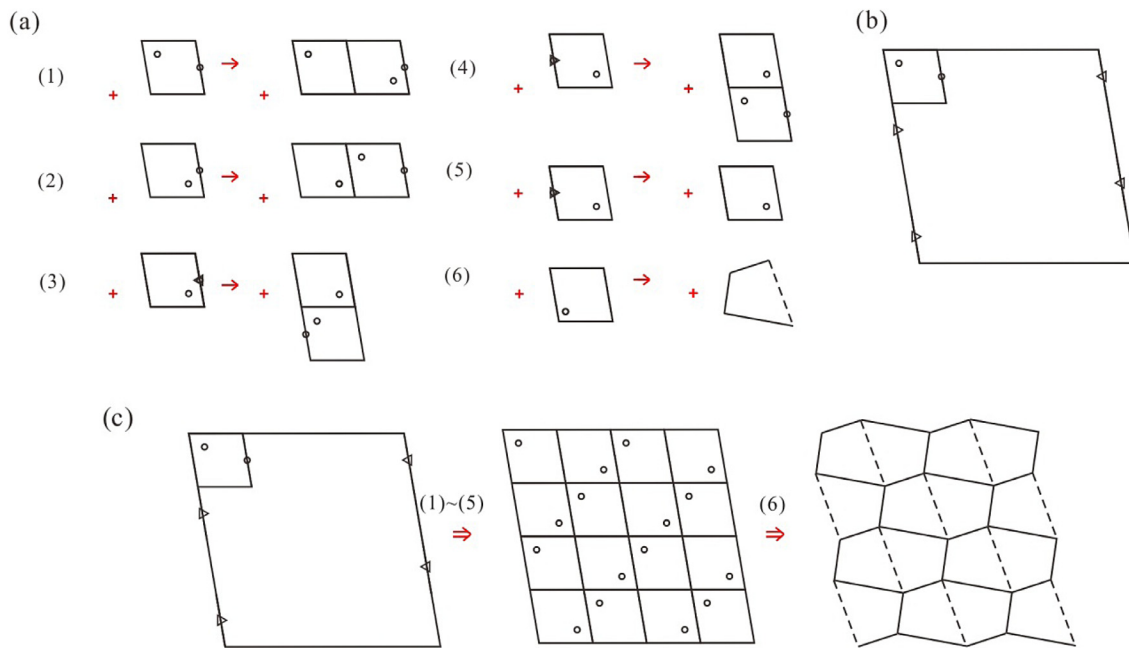


Fig. 10. The shape grammar for generating the Huffman grid pattern. (a) Shape rules (1) to (5) generate the rhombus-grid, and shape rule (6) replaces the grid with a new motif; (b) Initial shape; (c) Production of the Huffman grid pattern.

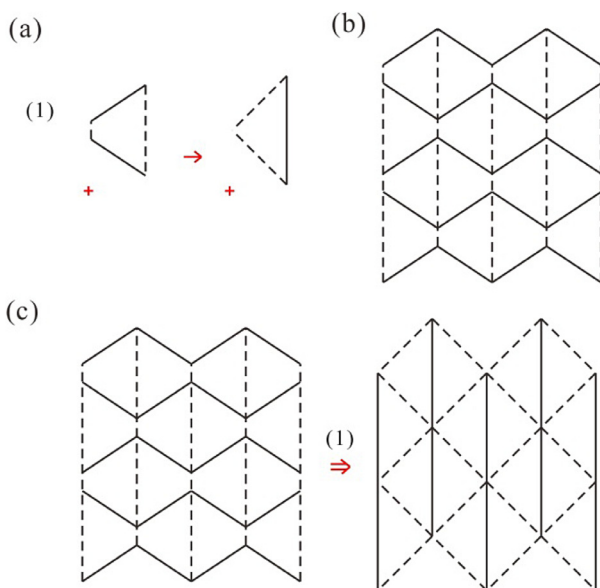


Fig. 11. Shape computation from Arc pattern to Yoshimura pattern. (a) Shape rule; (b) Initial shape; (c) Production of Yoshimura pattern.

shown in Fig. 12(a) preserve the flat-foldability and deployability of the original pattern after adding the gadget. Adding gadgets to an existing Miura-ori pattern requires that the crease in the previous pattern be split. Reconstruction of the whole structure is unavoidable in the design process. The visual computation in Shape Machine bypasses all this restructuring of the parametric model and instead replaces a whole unit with the Miura-ori pattern with gadgets. The local crease pattern can be straightforwardly embedded in the Miura-ori pattern in the form of two shape rules to account for the different orientations of the crease. Both rules apply in the schema $x \rightarrow \sum T(x)$ to the complete sets of the available types of the Ψ -shapes shown in Fig. 12(c). Shape Machine will find the specific positions of the embedded shapes

and make the change at all matching positions. In this way, the remodeling of the whole pattern is avoided. The physical model of Miura-ori pattern with gadget is shown in Fig. 1(e).

The second computation takes on the type of transformation of movement between the two patterns and significantly adds an expressive power in the transformative generation of new origami patterns from existing ones. In the previous example, the gadget adding in the Miura-ori pattern is a rigidly foldable gadget and will not modify the motion of the Miura-ori pattern. By observing the crease pattern of Miura-ori with gadget and playing with the physical model, new questions readily arise: What would happen if the crease lines of the Miura-ori pattern are removed? What happens if the gadgets connect to each other? Will these processes form a new pattern? Is the pattern foldable? It is suggested here that a new origami pattern can be generated by removing the crease lines between gadgets. Because the motion of each embedded gadget is coordinated with the original Miura-ori unit, the creases between the gadgets remain parallel to each other in two directions. Removing these creases introduces translation transformations other than rotation to each gadget and the motion coordination is not broken. Fig. 13 shows the shape computation of a new pattern with only gadgets, named as “gadget pattern”. The initial shape is still a Miura-ori pattern. To connect the gadgets together, the original Miura-ori pattern has to be elongated comparing to the initial shape in Fig. 12(d). Otherwise, more shape rules will be needed. This transformation is encoded in terms of two pairs of shape rules that both apply in the schema $x \rightarrow \sum T(x)$ to change the topology of the pattern in its body and its boundaries respectively. Rule (1) and rule (2) change the topology of the vertices at the second to the fourth rows into gadgets; Rules (3) and (4) change the topology of the vertices at the first and the fifth rows into gadgets. Lines between the gadgets disappear and a new topology around the vertices appears. Although the gadget in this pattern is generated from the Miura-ori, there is no Miura-ori structure in this pattern anymore. A parallelogram is formed inside four connected gadgets, which increases the complexity of the crease pattern. The new gadget is flat foldable and has a negative Poisson’s ratio. Its physical model is shown in Fig. 1(f).

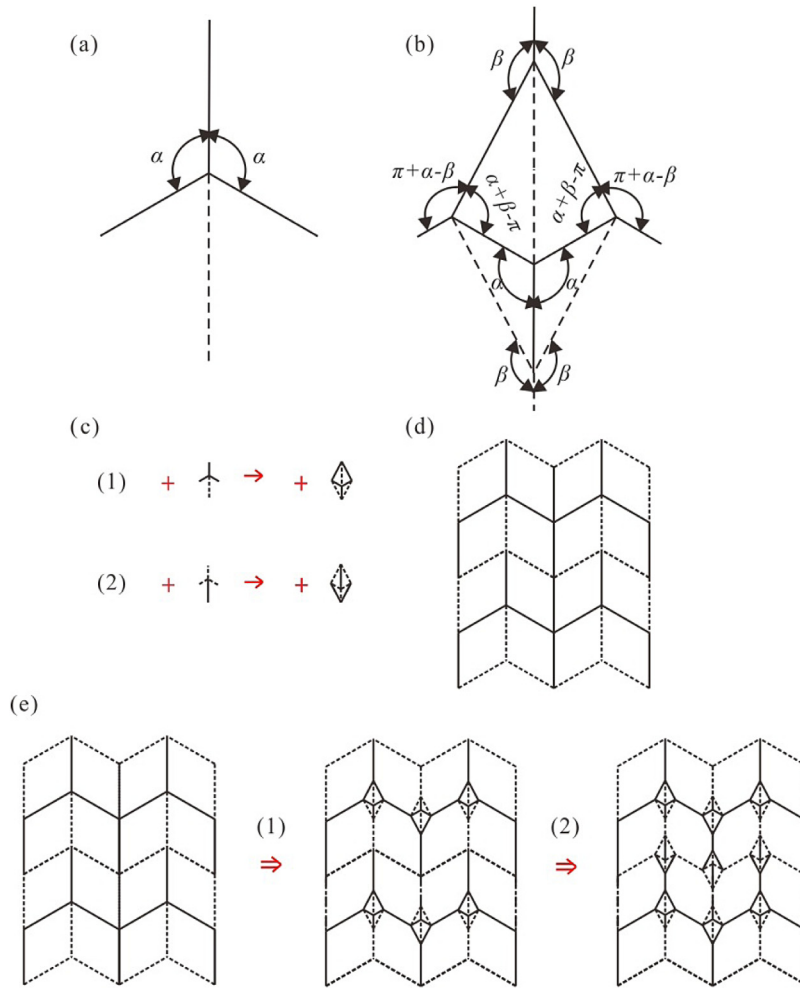


Fig. 12. Shape computation of the Miura-ori pattern with symmetric corner gadgets. (a) Original Miura-ori vertex; (b) A gadget vertex is flat-foldable and deployable for $\pi/2 < \alpha < \beta < \pi$; (c) Shape rules; (d) Initial shape; (e) Production of the Miura-ori pattern with gadget.

4. Discussion

In this work, a formal description of origami tessellation patterns is presented using the shape grammar formalism implemented within the Shape Machine interpreter. We consider the origami pattern as a combination of different shapes in spatial relations and we construct and modify origami patterns visually by instantiating, modifying and applying shape rules defined directly in terms of shapes and spatial relations. A general way of constructing an underlying square tessellation is presented and various families and types of origami tessellations are produced by alternative labelings of the square cells and the shapes that substitute them. Moreover, the power of creating origami tessellations in terms of two-dimensional visual rules is contrasted with current approaches in mathematical modeling of origami and the emphasis is given in the computations in Shape Machine that allow seamless translations between geometries and topologies of origami patterns without requiring any reworking on the representation of the spatial elements that constitute the patterns.

All examples have been given so far in the form of a shape grammar with a set of shape rules and an initial shape upon which the shape rules apply to make the production of the design. Still, there is no need for the production to require always an initial shape. In fact, the initial conditions for the setup of the production can be encoded in the shape rules themselves and the process can start from scratch. An example inspired by a 17-gon

ring pattern [54] is given below to illustrate the serial application of shape rules starting with a shape rule that features an empty shape on its LHS. Any type of regular polygons on the RHS would do. In this example a 12-gon origami ring pattern is generated. The complete set of shape rules of the 12-gon origami ring pattern is shown in Fig. 14. Rule (1) specifies the initial shape of the computation, a 12-gon; its LHS is the empty shape and its RHS is the instance of the 12-gon. Rule (2) offsets a 12-gon to define the outer 12-gon of the origami. The first rule applies in the rule schema $x \rightarrow y$, for x the empty shape and y the 12-gon; the second rule applies in the rule schema $x \rightarrow x + T(x)$, for x the 12-gon and $T(x)$ its enlarged, offset copy. Rule (3) subdivides the region between the two 12-gons into 12 wedges. Rules (4) and (5) specify the skeleton outline of the pattern and rules (6) and (7) specify the second round of the pattern. Shape rules (3)–(7) are applied in the schema $x \rightarrow \sum T(x)$ to all twelve parts of the design. The physical model of the 12-gon ring pattern is shown in Fig. 1(j).

Significantly, all examples discussed in the paper are given in three rule schemata, namely, $x \rightarrow y$, $x \rightarrow x + T(x)$, and $x \rightarrow \sum T(x)$ applying to all possible matching parts of the design. In either case and for any shape rules defined within these three schemata, the shape rule executes till there are no other matching parts in the design and then the production continues to the next rule till all the shape rules in the shape grammar have been executed to make the design. In other words, every grammar produces a single design; a change in one of the shape rules of

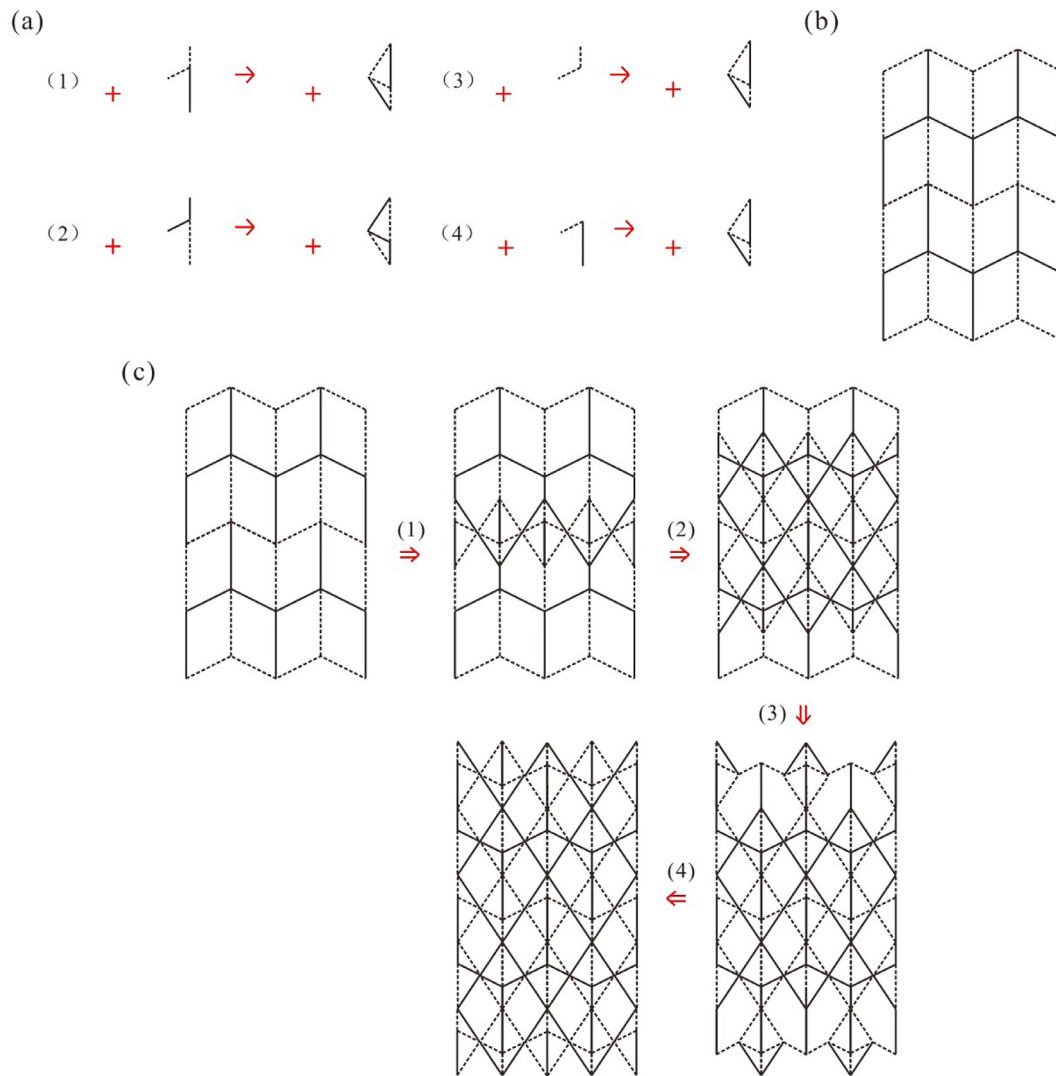


Fig. 13. Shape computation of the Gadget pattern. (a) Shape rules; (b) Initial shape; (c) Production of the gadget pattern.

the grammar readily produces a new instance of a design. This type of design inquiry is straightforwardly generalized in the form of programming language whereas the collections of shape rules can be compiled in a shape script that automates the production of an origami tessellation in a single execution of the complete program.

Additional fronts to tackle are readily within reach: New classes of origami patterns generalizing the flower-like pattern in Fig. 1(a) and the gadget pattern in Fig. 1(j); additional rule schemata, including divisional operators and boundary operators [25,29] to allow for a greater expressiveness in the specification of the shape rules; additional transformations under which shape rules apply, including affine and linear transformations, to allow for even more flexibility and expressiveness in the modeling and modification of shape rules; additional programming constructs in a software package implementation module to allow for iterative loops, conditions and so forth for more variability in the specification of the origami patterns at each stage of a rule set; and a closer integration with current mathematical origami modelers to allow for the seamless evaluation and analytical analysis and insight of origami characteristics within the Shape Machine itself. The designs presented in this work are meant to draw attention to a different way of thinking and designing origami pattern. It is suggested that shape grammars and Shape

Machine can provide a new perspective of constructing, editing and ultimately designing origami tessellation patterns.

CRedit authorship contribution statement

Ying Yu: Designed research, Performed research, Contributed new tools, Analyzed data, Wrote the paper. **Tzu-Chieh Kurt Hong:** Designed research, Performed research, Contributed new tools, Analyzed data, Wrote the paper. **Athanasios Economou:** Designed research, Performed research, Contributed new tools, Analyzed data, Wrote the paper. **Glaucio H. Paulino:** Designed research, Performed research, Contributed new tools, Analyzed data, Wrote the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

G.H.P. and Y.Y. acknowledge the endowment provided by the Raymond Allen Jones Chair at the Georgia Institute of Technology.

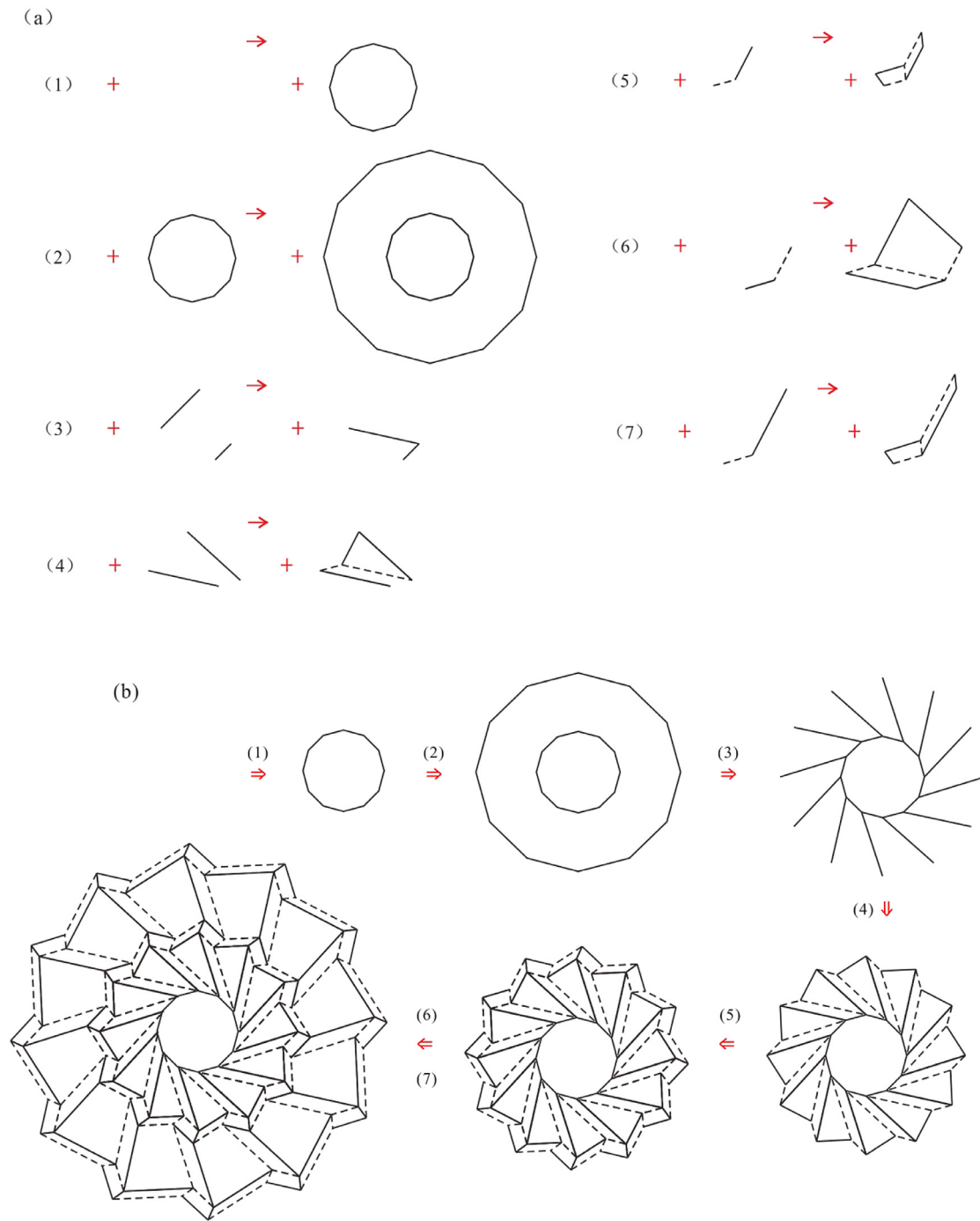


Fig. 14. Shape computation to generate an origami 12-gon ring pattern; (a) Shape rules (b) Shape computation.

References

- [1] McArthur M. *New expressions in origami art: masterworks from 25 leading paper artists*. Tuttle Publishing; 2017.
- [2] Jackson P. *Folding techniques for designers: from sheet to form (how to fold paper and other materials for design projects)*. Laurence King Publishing; 2011.
- [3] Sorğuç AG, Hagiwara I, Selçuk S. Origamics in architecture: a medium of inquiry for design in architecture. *MetuJfa* 2009;2:235–47. <http://dx.doi.org/10.4305/METUJFA.2009.2.12>.
- [4] Demaine ED, O'Rourke J. *Geometric folding algorithms: linkages, origami, polyhedra*. Cambridge University Press; 2007.
- [5] Lang RJ. *Twists, tilings, and tessellations: mathematical methods for geometric origami*. AK Peters/CRC Press; 2017.
- [6] Miura K. *Method of packaging and deployment of large membranes in space*. Title the institute of space and astronautical science report 618, 1985, p. 1.
- [7] Resch RD. *Self-supporting structural unit having a series of repetitious geometrical modules*. 1968, United States patent 3407558.
- [8] Resch R, Christiansen H. The design and analysis of kinematic folded plate systems. In *Proceedings of IASS symposium on folded plates and prismatic structures* 1970.
- [9] Gattas JM, Wu W, You Z. Miura-base rigid origami: Parameterizations of first-level derivative and piecewise geometries. *J Mech Des* 2013;135. <http://dx.doi.org/10.1115/1.4025380>.
- [10] Tachi T. Designing free form origami tessellations by generalizing Resch's patterns. *J Mech Des* 2013;135. <http://dx.doi.org/10.1115/1.4025389>.
- [11] Filipov ET, Tachi T, Paulino GH. Origami tubes assembled into stiff, yet reconfigurable structures and meta materials. *Proc Natl Acad Sci* 2015;112:12321–6. <http://dx.doi.org/10.1073/pnas.1509465112>.

- [12] Liu K, Tachi T, Paulino GH. Invariant and smooth limit of discrete geometry folded from bistable origami leading to multistable meta surfaces. *Nature Commun* 2019;10:4238. <http://dx.doi.org/10.1038/s41467-019-11935-x>.
- [13] Xu L, Shyu TC, Kotov NA. Origami and kirigami nanocomposites. *ACS Nano* 2017;11:7587–99. <http://dx.doi.org/10.1021/acsnano.7b03287>.
- [14] Tikhomirov G, Petersen P, Qian L. Fractal assembly of micrometre-scale DNA origami arrays with arbitrary patterns. *Nature* 2017;552:67–71. <http://dx.doi.org/10.1038/nature24655>.
- [15] Pratapa PP, Suryanarayana P, Paulino GH. Bloch wave framework for structures with nonlocal interactions: Application to the design of origami acoustic metamaterials. *J Mech Phys Solids* 2018;118:115–32. <http://dx.doi.org/10.1016/j.jmps.2018.05.012>.
- [16] Mu J, et al. Origami-inspired active graphene-based paper for programmable instant self-folding walking devices. *Sci Adv* 2015;1:e1500533. <http://dx.doi.org/10.1126/sciadv.1500533>.
- [17] Lin C-H, et al. Highly deformable origami paper photo detector arrays. *ACS Nano* 2017;11:10230–5. <http://dx.doi.org/10.1021/acsnano.7b04804>.
- [18] Zhao Z, et al. Origami by frontal photo polymerization. *Sci Adv* 2017;3:1602326. <http://dx.doi.org/10.1126/sciadv.1602326>.
- [19] Nauroze SA, Novelino LS, Tentzeris MM, Paulino GH. Continuous-range tunable multilayer frequency-selective surfaces using origami and inkjet printing. *Proc Natl Acad Sci* 2018;115:13210–5. <http://dx.doi.org/10.1073/pnas.1812486115>.
- [20] Tang R, et al. Origami-enabled deformable silicon solar cells. *Appl Phys Lett* 2014;104:083501. <http://dx.doi.org/10.1063/1.4866145>.
- [21] Morgan J, Magleby SP, Howell LL. An approach to designing origami-adapted aerospace mechanisms. *J Mech Des* 2016;138. <http://dx.doi.org/10.1115/1.4032973>.
- [22] Lang RJ. Treemaker (version 5). 2015, <https://langorigami.com/article/treemaker/>.
- [23] Tachi T. Freeform origami (version 0.3). 2017, <http://myehosthttp://origami.c.u-tokyo.ac.jp/fachi/software/>.
- [24] Liu K, Paulino GH. Merlin software. 2018, <http://paulino.ce.gatech.edu/software.html>.
- [25] Lindenmayer A. Mathematical models for cellular interactions in development II. Simple and branching filaments with two-sided inputs. *J Theoret Biol* 1968;18(3):300–15. [http://dx.doi.org/10.1016/0022-5193\(68\)90080-5](http://dx.doi.org/10.1016/0022-5193(68)90080-5).
- [26] Rozenberg G, Salomaa A. *The mathematical theory of L systems*. New York: Academic Press; 1980.
- [27] Bailey S, Kim T, Strichartz RS. Inside the Lévy dragon. *Amer Math Monthly* 2002;109(8):689–703. <http://dx.doi.org/10.2307/3072395>.
- [28] Evans TA, Lang RJ, Magleby SP, Howell LL. Rigidly foldable origami gadgets and tessellations. *R Soc Open Sci* 2015;2:150067. <http://dx.doi.org/10.1098/rsos.150067>.
- [29] Stiny G. *Shape: talking about seeing and doing*. MIT Press; 2006.
- [30] Stiny G. Introduction to shape and shape grammars. *Environ Plann B Plann Des* 1980;7:343–51. <http://dx.doi.org/10.1068/b070343>.
- [31] Knight TW. *Transformations in design: a formal approach to stylistic change and innovation in the visual arts*. Cambridge University Press; 1995.
- [32] Economou A, Kotsopoulos S. From shape rules to rule schemata and back. In: *Proceedings of the Sixth International Conference Design Computing and Cognition (DCC)*. 2014, p. 419–38. http://dx.doi.org/10.1007/978-3-319-14956-1_22.
- [33] Economou A, Grasl T. Paperless grammars. *Comput Stud Cult Var Hered* 2018;139–60.
- [34] Stiny G, Mitchell W. The palladian grammar. *Environ Plann B Plann Des* 1978;5:5–18. <http://dx.doi.org/10.1068/b050005>.
- [35] Stiny G, Mitchell W. The grammar of paradise: on the generation of mughul gardens. *Environ Plann B Plann Des* 1980;7:209–26. <http://dx.doi.org/10.1068/b070209>.
- [36] Pugliese M, Cagan J. Capturing a rebel: Modelling the harley-davidson brand through a motorcycle shape grammar. *Res Eng Des* 2002;13(3):139–56. <http://dx.doi.org/10.1007/s00163-002-0013-1>, (2002).
- [37] Knight TW. Transformations of de stijl art: the paintings of george vantongerloo and fritz glarner. *Environ Plann B: Plann Des* 1989;16:51–98. <http://dx.doi.org/10.1068/b160051>, (1989).
- [38] Knight TW. The generation of hepple white-style chair-back. *Des Environ Plann B Plann Des* 1980;7:227–38. <http://dx.doi.org/10.1068/b070227>.
- [39] Jowers I, Prats M, H Eissa, Lee JH. A sty of emergence in the generation of islamic geometric patterns. In *Dave, A. I.Li, N. Gu, H.-J. Park (eds.), Proceedings of the 15th International Conference CAADRIA*, 2010, p. 39–48.
- [40] Gattas J. A generative shape grammar for piecewise cylindrical surfaces and curved-crease origami. In: *Lang R, Bolitho M, You Z, editors. Origami 7, volume 3: engineering one*. 2018, p. 921–34.
- [41] McKay A, et al. Spatial grammar implementation: From theory to useable software. *Artif Intell Eng Des Anal Manuf* 2012;26(2):143–59. <http://dx.doi.org/10.1017/S0890060412000042>.
- [42] Hong T-CH, Economou, five criteria for shape grammar interpreters. In: *Gero JS, editor. Design computing and cognition DCC'20*. Springer; 2020.
- [43] Economou A, Hong T-CK, Ligler H, Park J. Shape machine: A primer for visual computation. *Cult DNA: Comput Stud Cult Var Hered* 2019;79–102.
- [44] Stiny G. The algebras of design. *Res Eng Des* 1991;2:171–81. <http://dx.doi.org/10.1007/BF01578998>.
- [45] Krishnamurti R, Giraud C. Towards a shape editor: The implementation of a shape generation system. *Environ Plann B: Plann Des* 1986;13:391–404. <http://dx.doi.org/10.1068/b130391>.
- [46] Tapia M. A visual implementation of a shape grammar system. *Environ Plann B Plann Des* 1999;26:59–73. <http://dx.doi.org/10.1068/b260059>.
- [47] Grasl T, Economou A. From topologies to shapes: Parametric shape grammars implemented by graphs. *Environ Plann B: Plann Des* 2013;40:905–22. <http://dx.doi.org/10.1068/b38156>.
- [48] Stouffs R, Hou D. Composite shape rules. In: *International conference on design computing and cognition*. 2018, p. 439–56.
- [49] Stiny G. Ice-ray: A note on the generation of chinese lattice designs. *Environ Plann B Plann Des* 1977;4:89–98. <http://dx.doi.org/10.1068/b040089>.
- [50] Dye DS. *A grammar of chinese lattice*. Harvard University Press; 1949.
- [51] Economou A, Grasl T. Sieve_n: A computational approach for the generation of all partial lattices of two-dimensional shapes with an n-fold symmetry axis. In: *Predicting the future: proceedings of the twentieth-fifth conference of the education and research in computer-aided architectural design in europe (ECAADe)*. 2007, p. 947–54.
- [52] Shubnikov A, Koptsik V. *Symmetry in science and art*. Plenum Press; 1974.
- [53] Devlin K. *Mathematics: the science of patterns : the search for order in life, mind, and the universe*. Scientific American Library; 1994.
- [54] Lang RJ. *Origami design secrets: mathematical methods for an ancient art*. second ed.. AK Peters/CRC Press; 2011.