Shape-morphing mechanical metamaterials

Caigui Jiang[1], Florian Rist[1], Hui Wang[1], Johannes Wallner[2,*], Helmut Pottmann[1]

Abstract

Small-scale cut and fold patterns imposed on sheet material enable its morphing into three-dimensional shapes. This manufacturing paradigm has been receiving much attention in recent years and poses challenges in both fabrication and computation. It is intimately connected with the interpretation of patterned sheets as mechanical metamaterials, typically of negative Poisson ratio. We here present an affirmative solution to a fundamental geometric question, namely the targeted programming of a shape morph. We use optimization to compute kirigami patterns that realize a morph between shapes, in particular between a flat sheet and a surface in space. The shapes involved can be arbitrary; in fact we are able to approximate any mapping between shapes whose principal distortions do not exceed certain bounds. This amounts to a solution of the so-called inverse problem for kirigami cut and fold patterns. The methods we employ include a differential-geometric interpretation of the morph, besides drawing on recent progress in geometric computing.

Keywords: kirigami, auxetic materials, metamaterials, shape morphing, computational fabrication,

1. Introduction

In recent years there has been growing interest in what we would like to call *geometric materials*, namely mechanical metamaterials capable of shape morphing. One particular way these can be achieved is via folding flat sheet material [1, 2, 3, 4, 5, 6]. Another route is offered by cut and fold patterns imposed on flat sheets, dissecting them into small parts connected by hinges, with subsequent actuation causing the morph. Cuts in addition to folds (*kirigami*) considerably add to the degrees of freedom that are available for morphing. It is this idea that is studied in the present paper – see Fig. 1 for an example. This approach to generating 3D shapes is particularly valuable in situations where this shape is to be endowed with technology more easily printed onto a 2D sheet. A prime example is electronics, where cutting and folding is an alternative to other ways of manufacturing three-dimensional circuit carriers like molding [7]. For this reason, folding has been proposed for pop-up robots on Mars [8].

A typical feature of geometric metamaterials is their auxetic behaviour [9, 10, 11], with expansion under tension being caused by gaps opening when micro-elements are rotating. Analysis of this behaviour is the key for treating metamaterials by differential-geometric methods as we are going to do in this paper.

The cut and fold patterns involved in the geometric materials we have in mind are repetitive, following a periodic tesselation of the plane by triangles [12, 13, 14], quadrilaterals [15], hexagons

*Corresponding Author
[1]King Abdullah University of Science and Technology
[2]Graz University of Technology

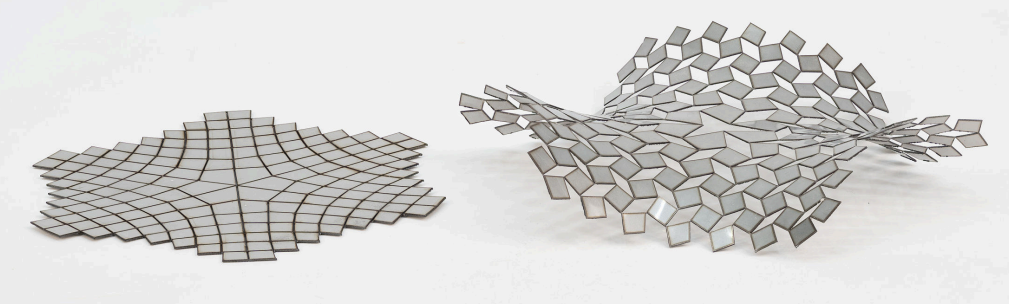


Figure 1: For a given surface in space, we can design a cut pattern in the plane that deploys continuously to that surface.

[16, 17], or more complex periodic patterns [18]. The modulation of strict periodicity to achieve a morph to a double-curved shape is called *programming shape* or *programming curvature* [12, 13, 14, 15, 19, 20]. Figure 2 shows an example of this.

While our geometric considerations mostly refer to metamaterials composed of rigid elements, auxetics have been made in many different ways, from the molecular level [21] to laser cutting elastic sheets [22]. The details of deployment are diverse and include thermal actuation [23], in-plane expansion to the geometrically possible maximum [13], and out-of-plane buckling induced by tension [24]. Further, there is a broad range of shape morphs achieved by materials of higher elasticity, e.g. flexible tiles [25], or inflatables with an embedded kirigami pattern [26]. In this paper actuation is mostly steered by the limits of geometry: we mostly move from a closed state to a fully open one, or vice versa. We also investigate *bistable* metamaterials permitting a snap-through morph [14, 27, 18], see e.g. Figure 6.

Problem statement.

A major challenge in this area is to make programming shape computationally effective. This so-called *inverse problem* can be formulated as follows:

> *Given a shape, can we design a flat cut and fold pattern that after deployment assumes that shape?*

We can ask the same question also for a non-flat initial surface. A more general question is the following:

> *If a mapping to a target surface is given, can we design a cut and fold pattern such that the shape morph defined by that pattern approximates the given mapping?*

This question has already been treated thoroughly for the special case of conformal mappings [12, 13, 14]. The answer in general is affirmative, within limits. If the principal distortions of mappings are within certain boundaries, mappings can be reproduced by the morph of a cut and fold pattern. We can effectively compute morphs from a flat state to a target shape in space (Fig. 1); and likewise for the reverse, namely a morph from a three-dimensional shape to a flat state (Fig. 2). Existence and computation of a geometric material tailored to the desired shape morph is the main contribution of this work.

The solution is based on aligning cut and fold patterns with the *principal stretch directions* of mappings between surfaces. We also believe that the geometric principles at work here may open up avenues of promising future work on the geometry side.

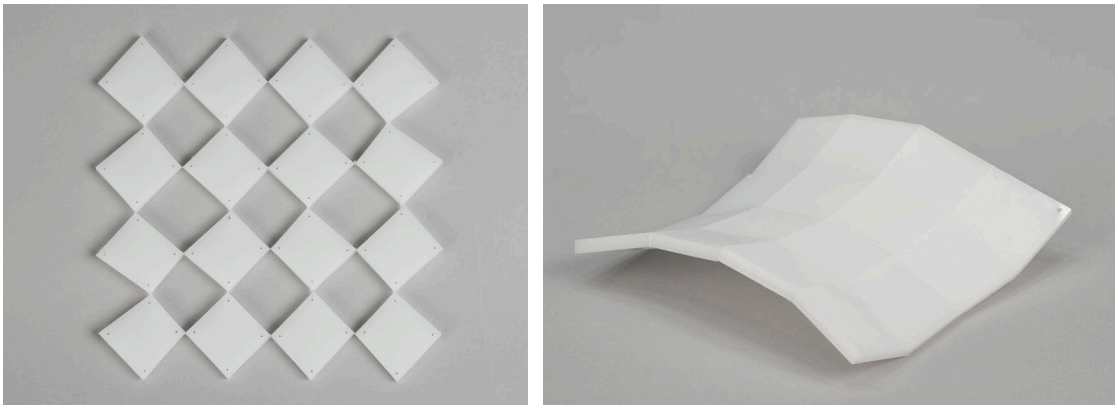


Figure 2: Programming curvature into an almost regular pattern. Here an almost regular open 2D pattern closes to become a surface in space.

Contributions

On a technical level, our contributions are the following.

— We show how a cut and fold pattern that effects a desired shape morph can be computed in a multi-stage process involving optimization. Since the combinatorics of the mesh layout are part of the solution, this optimization is very difficult unless an informed proposal yields initial values to start from. That proposal is based on differential-geometric relations between the desired morph on the one hand, and local cut and fold patterns on the other hand.
— We are able to leverage recent progress in the area of kinematic models of surfaces, which already were successfully employed to numerically treat isometric mappings [28].
— The shape transformation power of mechanical meta-materials is demonstrated by means of a universality result on bistable auxetics. We show that they can mimick deformations between surfaces which are arbitrary except their principal stretches must obey certain bounds.

2. Discretizing Deformations of Continua by Kirigami Patterns

Our treatment of geometric metamaterials is based on the idea that their morph approximates the deformation of a continuum. The material itself is composed of elementary *cells* to which geometric primitives are attached. The morph deforms cells but lets the primitives move as rigid bodies. The shapes of cells and geometric primitives are modelled after ideal shapes in a periodic planar arrangement. It is exactly in the deviations from this ideal shape where curvature is programmed into the pattern.

We start our exposition by describing the ideal shape corresponding to a simple case, where four rectangular primitives of the same size are connected with hinges and are positioned around a rectangular cell. They form a mechanism moving from the fully opened state (Fig. 3, left) to the closed state (Fig. 3, right).

2.1. Constant mean stretch deformations

We consider the transformation of cells $C_1 \rightarrow C_0$ shown by Fig. 3 as deformation of a continuum. Cells C_1, C_0 are rectangular of size $w \times w$ and $a \times b$, resp. The deformation of a continuum mentioned here is modelled as a differentiable mapping ψ from one surface $s(u, v)$

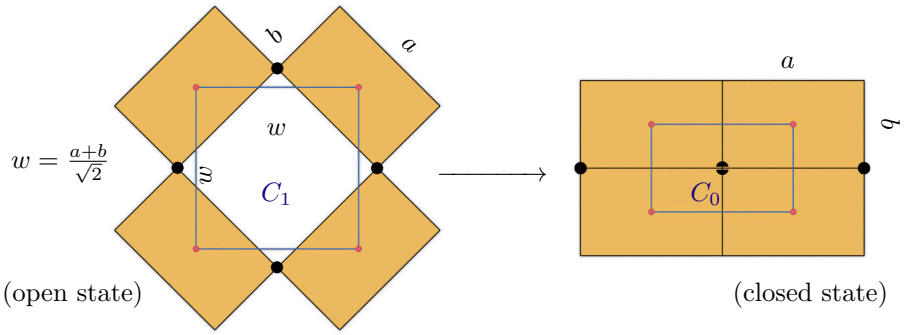


Figure 3: This metamaterial has a square basic cell to which four rigidly moving rectangles of size $a \times b$ are attached. The closing movement causes each rectangle to rotate by 45 degrees and deforms the cell into a rectangular shape. The deformation of cells enjoys constant mean stretch.

to another surface $\bar{s}(u, v)$. In each position (u, v) it is linearized by the linear mapping $d\psi_{(u,v)}$ which maps partial derivatives onto each other:

$$d\psi(s_u) = \bar{s}_u, \quad d\psi(s_v) = \bar{s}_v.$$

The dependence of $d\psi$ on u, v is not indicated in this notation. By the singular value decomposition, one finds an orthonormal basis b_1, b_2 in the tangent plane of the first surface, an orthonormal basis $\bar{b}_1, \bar{b}_2$ in the tangent plane of the second surface, and nonnegative factors λ_1, λ_2 such that

$$d\psi(b_1) = \lambda_1 \bar{b}_1, \quad d\psi(b_2) = \lambda_2 \bar{b}_2, \quad \text{where w.l.o.g., } \lambda_1 \geq \lambda_2 \geq 0.$$

Thes vectors b_1, b_2 and $\bar{b}_1, \bar{b}_2$ indicate the principal stretch directions, while λ_1, λ_2 are called principal stretches. This singular value decomposition of $d\psi$ is also commonly used to visualize the distortions of map projections in cartography. There, one considers the image of the unit circle under the mapping $d\psi$. It is an ellipse called the Tissot indicatrix [29] whose axes are indicated by vectors $\bar{b}_1, \bar{b}_2$ and whose dimensions are given by λ_1, λ_2. The factors λ_1, λ_2 are called the principal stretches, see [30] or [31]. The mapping ψ is *conformal*, if and only if principal stretches are equal in every point,

$$\lambda_1(u, v) = \lambda_2(u, v), \quad \text{for all}, u, v.$$

In this special case the basis vectors b_1, b_2 are not unique. Otherwise they are unique up to multiplication with -1 (of either b_1, or b_2, or both).

The motion of an elementary cell as defined by Figure 3 causes a cell of size $w \times w$ to change to a cell of size $a \times b$. Thus, such a motion approximates a deformation with principal stretches

$$\lambda_1 = \frac{a}{w}, \quad \lambda_2 = \frac{b}{w}, \quad \text{where } w = \frac{a+b}{\sqrt{2}}.$$

The sum of principal stretches is independent of the dimensions of cells, since

$$\lambda_1 + \lambda_2 = \frac{a}{w} + \frac{b}{w} = \sqrt{2}. \tag{1}$$

This is significant, not because of potential implications on finite strains, but because of connections to differential geometry.

Section 3 will show how, by varying the local geometry, but such that quads are still approximately rectangular, we are able to program curvature into the pattern and achieve a continuous morph between two surfaces (Fig. 4). In view of the previous paragraph, the morph from the open state S_1 to the closed state S_0 is a discrete approximation of a deformation where the mean principal stretch, $\frac{1}{2}(\lambda_1 + \lambda_2)$, is constant.

The idea to employ special deformations to describe morphing metamaterials is not new. E.g., conformal deformations can be realized by auxetic linkages modelled after the triangular *kagome* lattice if and only if $1 < \lambda_{1,2} < 2$ [12, 13, 14]. Also non-triangular, quad-based patterns have been treated by means of conformal mappings [20]. However, conformal mappings are not well suited to model the degrees of freedom inherent in quad-based patterns. Better results can be achieved when using a class of mappings where the principal stretches do not have to be equal. We are going to use "constant mean stretch" deformations to initialize cut and fold patterns whose elementary cells are modelled after Fig. 3.

Remarkably, the class of constant mean stretch deformations occurs also in the differential geometry of soap bubbles. The mean stretch of the mapping from the surface of the bubble to the unit sphere via the unit normal vectors is known as mean curvature. It equals $\gamma \cdot \Delta p$, with γ as surface tension and Δp as pressure difference [32]. To the best of our knowledge, constant mean stretch mappings between arbitrary surfaces have not yet been systematically studied from the viewpoint of differential geometry.

Discrete surfaces composed of rigid pieces were systematically studied by R. Sauer, cf. the textbook [33]. Unfortunately there does not seem to exist a comprehensive English language treatment of kinematic models, even if they are an ongoing topic of investigation in discrete

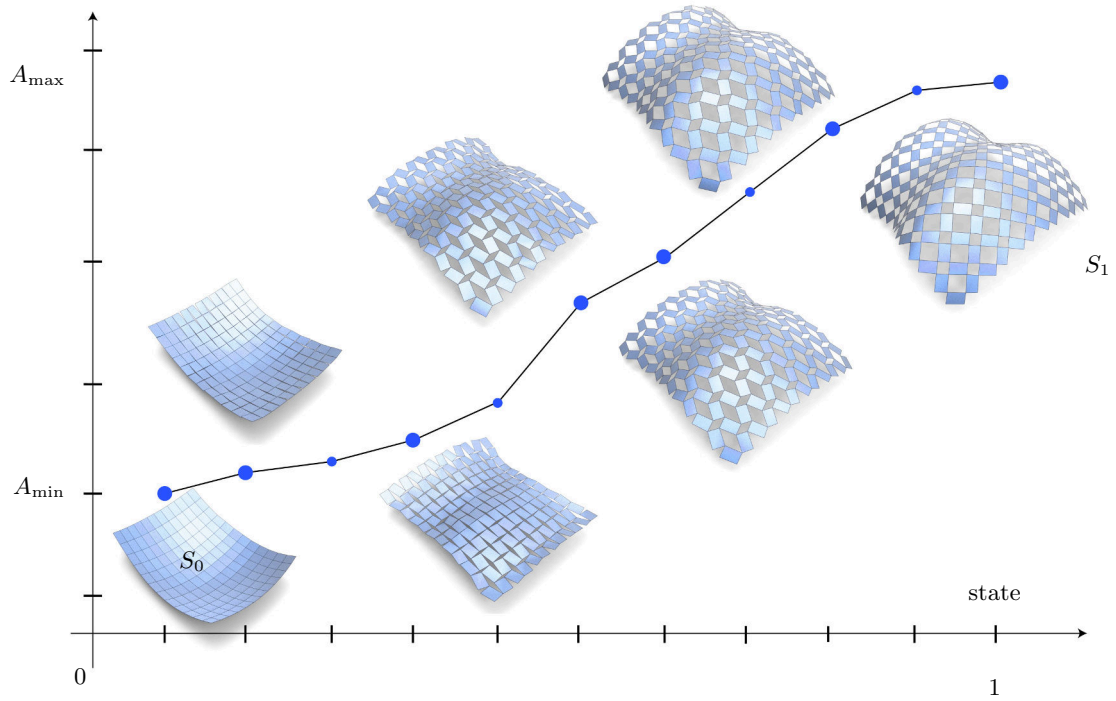


Figure 4: A surface kirigami based on the local configuration of Fig. 3 morphs between the fully closed state S_0 and the fully open state S_1. This morph is a discrete approximation of a continuous deformation of surfaces. This diagram also shows the area expansion on the y axis.

differential geometry — see e.g. [34] or [35].

The design principle of §2.1 can be applied to different and more general quad-based geometric metamaterials. One possible generalization concerns quad-based cut patterns that do not open fully. Rather, each incision opens to a hole that is roughly diamond-shaped, with the degree of opening depending on the location. In this way more general deformations between surfaces can be simulated. The basic idea of Fig. 3 has to be modified in obvious ways. We will briefly treat such generalizations in § 4.1.

2.2. Bistable quad-based auxetics

The principle of programming curvature can be applied to any quad-based regular planar pattern. We demonstrate how to treat a recently proposed bistable metamaterial [18] and show that it is capable of realizing more arbitrary morphs between 3D shapes than the constant mean stretch case discussed above.

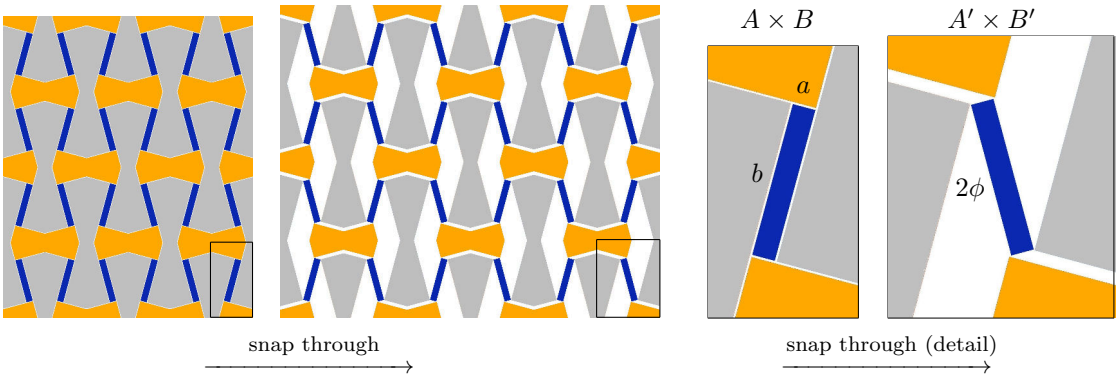
Consider a rectangular cell of size $A \times B$ as shown by Fig. 5. It is dissected into smaller parts by line segments forming the angle ϕ with the sides of the rectangle. By rotating the dark blue rectangle by the angle 2ϕ and translating the other parts we obtain a configuration fitting into a larger rectangle of size $A' \times B'$. Reflection and translation of the base cells creates two arrangements of rectangles and nonconvex 8-gons that form two stable configurations of a 2D periodic metamaterial. Snapping from one configuration to the other causes expansion. The principal stretches are

$$\lambda_1 = \frac{A'}{A} = 1 + 2\frac{b}{A}\sin\phi, \quad \lambda_2 = \frac{B'}{B} = 1 + 2\frac{a}{B}\sin\phi. \tag{2}$$

Given A, B we may choose the angle ϕ and the size $a \times b$ of the dark blue rectangle arbitrarily, only subject to the condition that the edges emanating from the blue rectangle always intersect the 'right' edges of the cell, preventing the grey and yellow 8-gons from becoming disconnected. Assuming symmetry, this leads to

$$a < A\cos\phi - B\sin\phi, \quad b < B\cos\phi - A\sin\phi. \tag{3}$$

The choice of a, b, ϕ, subject to (3), represents the degrees of freedom available for geometric modeling. The actual construction of bistable auxetic metamaterials requires optimization.



$$A' = A + 2b\sin\phi, \quad B' = B + 2a\sin\phi$$

Figure 5: Bistable auxetic metamaterial consisting of an arrangement of grey and yellow nonconvex 8-gons and blue rectangles. The pattern is generated by reflection and translation of a quadrilateral cell. It can snap between two stable states.

Figure 6: A 2D bistable auxetic metamaterial based on an underlying quad pattern.

2.3. Differential-geometric considerations

A shape morph according to § 2.1 between a fully closed state and a fully open state (see Figures 2 and 7) discretizes a mapping between surfaces which has the special property of constant mean stretch. In addition, the cells are aligned with the principal stretch directions. Another property, not mentioned so far, is the following. Since the elementary cells in the fully open state are near-squares, the collection of cells is a near-conformal image of the standard square grid. The continuous analogue of this statement is that the network of principal stretch lines is the conformal image of the horizontal and vertical lines a 2D domain. It is entirely unclear under what circumstances this property is already a consequence of the constant mean stretch property, and to our knowledge the answer to this question is not known. Our computations do not depend on an answer to this question, and they were successful in any case.

A related problem in classical differential geometry is the question whether the network of principal curvature lines is a similar conformal image. This is known to be true for the class of *isothermic* surfaces, which e.g. contains the surfaces of constant mean curvature (soap bubbles) already mentioned in §2.1 above [36].

The bistable case is different: Elementary cells are still aligned with principal stretch directions, but apart from inequalities (3) there are no restrictions on the shape of cells. Therefore, as long as the principal stretches are expressible in terms of a, b, A, B, ϕ according to Figure 5, we expect a mapping between surfaces to be realizable as a bistable auxetic metamaterial. This can be considered a *universality result*.

3. Computing Geometric Metamaterials

The actual computation of the geometric metamaterials described above is based on optimization, and leverages recent progress in geometric computing to handle isometric correspondences [37]. We employ a multi-stage procedure described in detail below.

- Stage 1 starts with two reference surfaces, given as triangle meshes, and establishes a deformation of the first to the second which approximates the desired morph of the metamaterial.

- Stage 2 uses a Levenberg-Marquardt algorithm [38] to optimize these so as to have the correct dimensions.
- Stage 3 assigns geometric primitives to cells and performs another round of optimization. Corresponding primitives in the closed and the open state must be congruent, which is enforced as a hard constraint.

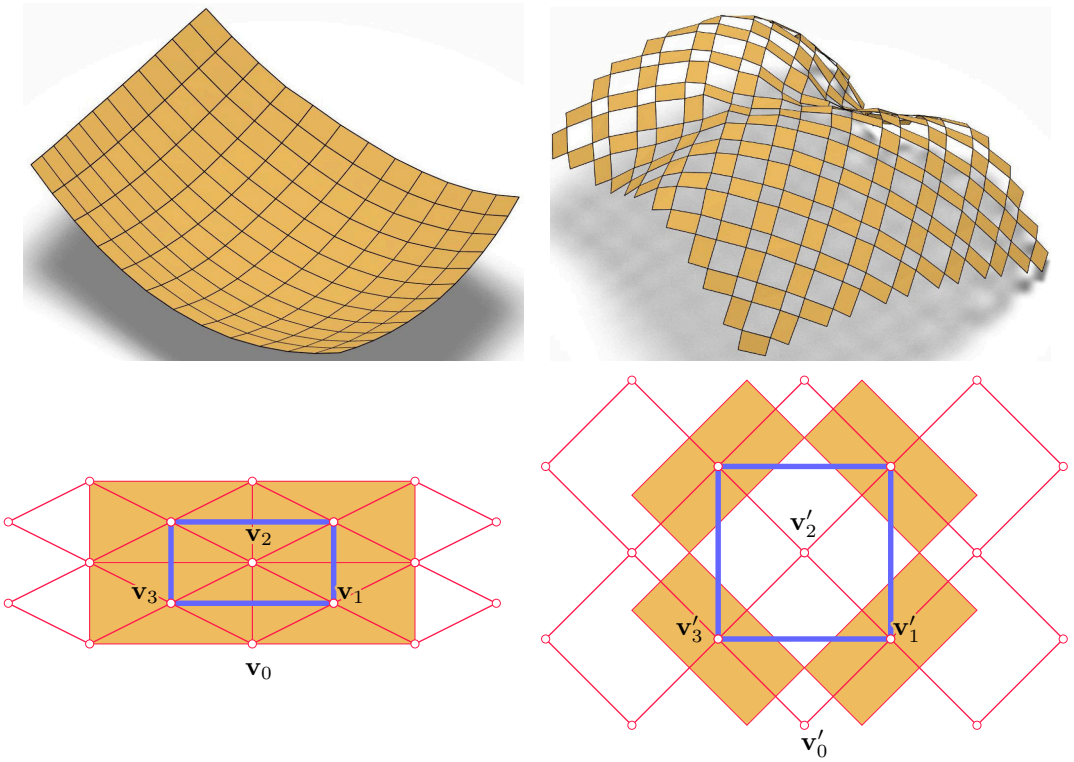


Figure 7: "Control" meshes (thin red) govern the movement of a geometric metamaterial whose basic cells (bold, blue) consist of diagonals of the control mesh. The individual pieces of the geometric metamaterial (orange) in both the closed and open state are derived from the two control meshes.

3.1. Stage 1: Initializing Continuous Deformations

The movement of geometric metamaterials as described by § 2 approximates a continuous deformation. To find a material that matches two given surfaces S and S' we start by finding that deformation, which is a thoroughly studied topic of Computer Graphics and Geometry Processing. As one of several possible alternatives we choose to work with the recently proposed checkerboard pattern method [37]. The deformation is represented by a pair of combinatorially equivalent quad meshes with vertex sets V, V' contained in S resp. S'. Following the checkerboard paradigm, the material's basic cells are formed by one half of the diagonals of faces in this mesh, see Fig. 7.

The constraints we impose on meshes V, V' depend on the particular nature of the geometric metamaterial we wish to generate. The example of Figure 3 calls for the correspondence $V \to V'$ to enjoy the constant mean stretch property. This is eventually achieved by the optimization of Stage 2 below, but to initialize that we first find a pair of meshes V_0, V'_0 in the plane that trivially

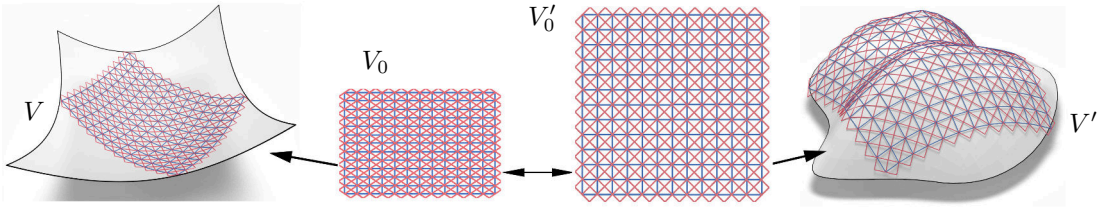


Figure 8: Control meshes V, V' in surfaces S, S' as found by optimization stage 1. Stage 2 is to find a constant mean stretch structure. We prepare for this by first setting up geometrically regular meshes V_0, V_0' enjoying a correspondence $V_0 \to V_0'$ with constant mean stretch, and subsequently determining V, V' such that correspondences $V_0 \to V$ and $V_0' \to V'$ are as isometric as possible.

have this property. The actual principal stretches of the mapping $V_0 \to V_0'$ are defined by the user, subject to Equ. (1). We then map V_0, V_0' into surfaces S resp. S' in an as-rigid-as-possible manner [39, 40]. This generates meshes V, V' that have still almost orthogonal diagonals and enjoy the desired property in an approximate way (Fig. 8).

3.2. Stage 2: Optimizing control meshes

The meshes obtained in Stage 1 do not yet enjoy the required properties and must undergo optimization, letting meshes V, V' glide along the reference surfaces S resp. S'. We illustrate this procedure first by means of constant mean stretch deformations. One property to be established is that cells are as rectangular as possible. It has proved efficient to achieve this by requiring that each face of V, V' has orthogonal diagonals, only half of which are used for cells [37]. In the notation of Fig. 7, this amounts to constraints

$$c_{\text{orth}}(f) = (\mathbf{v}_1 - \mathbf{v}_3) \cdot (\mathbf{v}_2 - \mathbf{v}_0) = 0,$$

for all faces $f = (\mathbf{v}_0, \ldots, \mathbf{v}_3)$ and analogous constraints for faces $f' = (\mathbf{v}_0', \ldots, \mathbf{v}_3')$ of the target mesh V'. We further need faces of V' to be approximately square. Corresponding faces f, f' should enjoy constant mean stretch λ. Both conditions are expressed in terms of edge lengths as

$$c_{\text{square}}(f') = \|\mathbf{v}_0' - \mathbf{v}_2'\|^2 - \|\mathbf{v}_1' - \mathbf{v}_3'\|^2 = 0, \tag{4}$$

$$c_{\text{stretch}}(f, f') = \|\mathbf{v}_1 - \mathbf{v}_3\| + \|\mathbf{v}_0 - \mathbf{v}_2\| - \lambda(\|\mathbf{v}_1' - \mathbf{v}_3'\| + \|\mathbf{v}_0' - \mathbf{v}_2'\|) = 0, \tag{5}$$

for all faces f and corresponding faces f'.

To ensure that meshes V, V' remain close to the reference surfaces S resp. S', we employ the points $\bar{\mathbf{v}}_i$ in S closest to vertex $\mathbf{v}_i$ and the normal vector $\mathbf{n}_i$ there. These data are recomputed in each round of our iterative optimization procedure, using fine mesh representations of S, S' and the method of *kd-trees*. The condition

$$c_{\text{prox}}(\mathbf{v}_i) = (\mathbf{v}_i - \bar{\mathbf{v}}_i) \cdot \mathbf{n}_i = 0$$

constrains the movement of $\mathbf{v}_i$ to a path tangential to S. Analogous constraints work on V'. The efficiency of this approach has been argued by means of Taylor expansions of the surfaces' squared distance field [41].

Finally, we impose soft constraints which ensure that the quad meshes our computations are based on can be interpreted as a discrete version of a smooth parametrization of a surface. Away from singularities, a quad mesh can be seen as the image of a mapping from the square grid $\mathbb{Z}^2$ (contained in the u, v domain) to space. A sequence of vertices in the integer grid which is lying

on a line parallel to either the u axis or the v axis is mapped to a sequence of vertices on the mesh which represents a discrete parameter line of a surface. We call these the *mesh polylines*. We impose a certain kind of fairness on them be requiring that for each triple $\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k$ of consecutive vertices on such a mesh polyline, the 2nd difference

$$c_{\text{fair}}(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k) = (\mathbf{v}_i - \mathbf{v}_j) - (\mathbf{v}_j - \mathbf{v}_k) \tag{6}$$

is as small as possible.

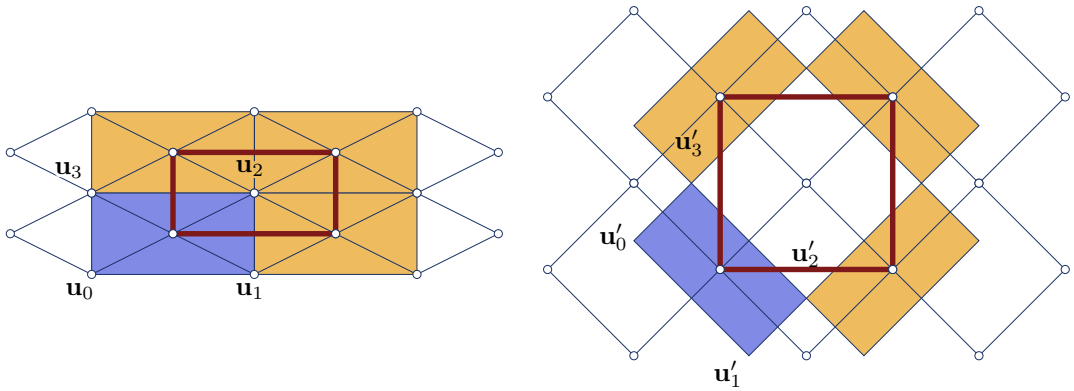


Figure 9: Deriving the vertices $\mathbf{u}_j, \mathbf{u}'_j$ of a geometric primitive from the control mesh to initialize stage three of optimization. The basic cells are highlighted in red.

Constraints c_{square} and c_{stretch} are specific to the constant mean stretch case, while the others apply more generally. In any case we form an objective function as a weighted sum of squares of all applicable constraints:

$$E_{\text{orth}} + E_{\text{geom}} + E_{\text{prox}} + E_{\text{fair}},$$

where

$$E_{\text{orth}} = w_{\text{orth}} \sum_{f \in F} c_{\text{orth}}(f)^2, \qquad E_{\text{geom}} = w_{\text{geom}} \sum_{f \in F} (c_{\text{square}}(f')^2 + c_{\text{stretch}}(f, f')^2),$$
$$E_{\text{prox}} = w_{\text{prox}} \sum_{v \in V} c_{\text{prox}}(v)^2, \qquad E_{\text{fair}} = w_{\text{fair}} \sum_{\text{triples}} c_{\text{fair}}(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k)^2.$$

Here the face f' is seen as a function of the face f, so summation is still over all faces $f \in F$.

The minimization of the target functional, using the coordinates of vertices as variables, is performed by a standard Levenberg-Marquardt algorithm [38] and yields the control meshes V, V'.

The choice of weights w_{orth}, w_{geom}, w_{prox}, w_{fair} is important: They must, for instance, compensate the different magnitudes of individual contributions to the target functional which are caused by the choice of unit length. The soft fairness constraints must not dominate the other hard constraints, while still ensuring that mesh polylines can be interpreted as discrete derivatives. In our academic implementation, the weights are chosen by hand in each individual example. Two different rules of thumb were used for the examples shown in this paper, and the fact that both succeeded indicates that the method is robust w.r.t. a change in weights. Detailed statistics on performance of optimization and the choice of weights are given in § 3.5 below.

3.3. Stage 3: Optimizing geometric microstructure

In this third stage we derive the individual pieces of the material's microstructure (the "geometric primitives") from the control meshes V, V' and apply a final pass of optimization.

We first describe the constant mean stretch materials according to Figure 3. Figure 9 illustrates the location of vertices $\mathbf{u}_j, \mathbf{u}'_j$ relative to the control meshes for the constant mean stretch case of Figures 3, 4. Stage 2 produces control meshes V, V' whose faces are not exactly rectangular, so in general the rules for defining vertices $\mathbf{u}_i, \mathbf{u}'_i$ from V, V' have to take some deformation into account. Accuracy is not an issue here, since we are going to perform further optimization anyway.

The most important constraint to be achieved is identity of dimensions of the individual pieces in the closed state (vertices $\mathbf{u}_i$) and the open state (vertices $\mathbf{u}'_i$). This is expressed by constraints

$$c_{\text{isometry}}(i,j) = \|\mathbf{u}_i - \mathbf{u}_j\|^2 - \|\mathbf{u}'_i - \mathbf{u}'_j\|^2 = 0, \tag{7}$$

where indices i, j run through *all* pairs of vertices that belong to the same piece, for all pieces. Vertices belong to different pieces that have identical position in the closed state are handled by a constraint of the form

$$c_{\text{match}}(i,j) = \|\mathbf{u}_i - \mathbf{u}_j\|^2 = 0. \tag{8}$$

Closeness of vertices to reference surfaces is dealt with as in Stage 2, as is fairness. For vertices $\mathbf{u}'_i$ however, the expression (6) is not appropriate, since even in the most regular cases, three consecutive vertices $\mathbf{u}'_i\mathbf{u}'_j\mathbf{u}'_k$ in a mesh polyline will be related by

$$c_{\text{fair},s}(i,j,k) = (\mathbf{u}'_i - \mathbf{u}'_j) - s(\mathbf{u}'_j - \mathbf{u}'_k) = 0, \tag{9}$$

where $s \neq 1$ (see Fig. 10). However, with the appropriate value of s (which depends on the concrete choice of geometric metamaterial), we use $c_{\text{fair},s}$ as a soft fairness constraint on vertices $\mathbf{u}'_i$. A weighted sum of squares of constraints now forms a target function that undergoes minimization by a standard Levenberg-Marquardt method in the same way as described in Stage 2. Section 3.2 applies (with the obvious substitutions), with isometry constraints being hard constraints, and fairness constraints being soft.

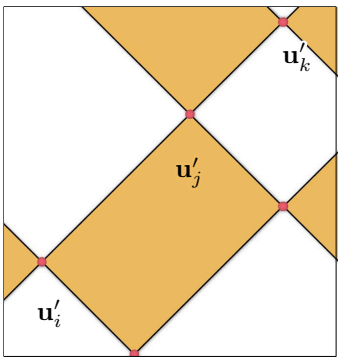


Figure 10: Illustration of the modified fairness term (9).

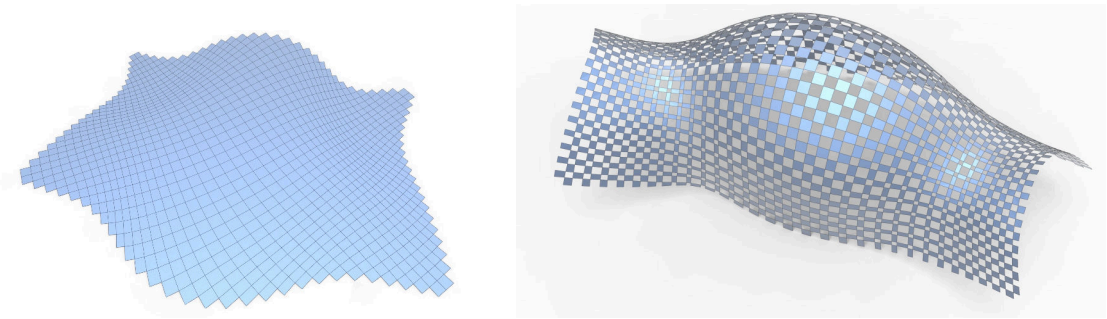


Figure 11: Example of a quad-based metamaterial where a planar closed state morphs to a maximally open spatial state. The computation according to stages 1–3 involves a constant mean stretch mapping from a planar domain to the given design surface at right.

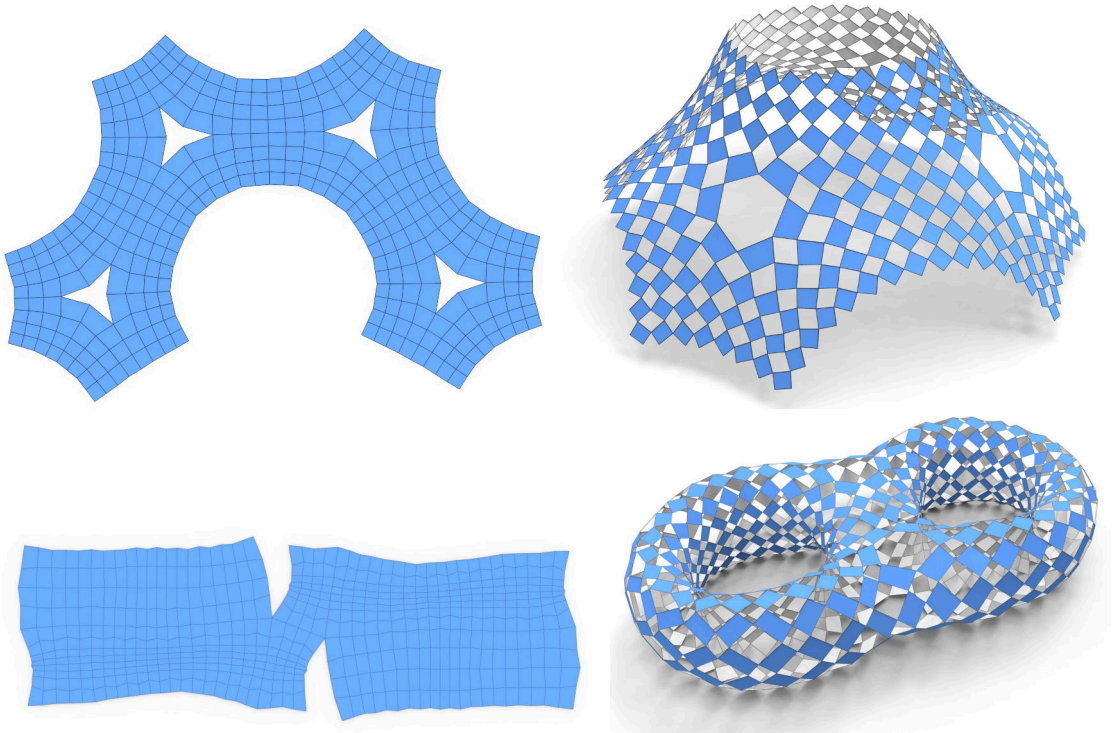


Figure 12: Examples of constant-mean-stretch metamaterials exhibiting combinatorial singularities. To model such significant changes in shape, it may be necessary to apply user-defined cuts.

Figure 13: For a given surface in space, we can design a cut pattern in the plane deploying continuously to that surface (figures at left). The expansion can also go the other way — on a given surface we design a cut pattern that causes deployment to a planar domain (figures at right).

3.4. Computations in the bistable case

We here briefly describe in which ways the computations in the bistable case differ.

- Stage 1: The bistable materials of Fig. 5 are capable of representing more arbitrary deformations. We could, in theory, use any of the available methods of Geometry Processing to find a deformation of the surface S to the surface S' that is as rigid as possible up to a scaling factor > 1. In a second step we compute its principal directions (by singular value decomposition of its differentials), and use known techniques to find meshes V, V' that align with the cross field of principal directions in S resp. S' [42]. In practice this procedure runs into difficulties since it is highly likely that for some pair of corresponding quads we will not be able to fit a configuration according to Fig. 5 to it.

  Rather, we adopt a pragmatic approach and use the idea of Fig. 8. The difference to the previous case is that the principal stretches of the mapping $V_0 \to V_0'$ are computed with Equ. (2). The parameters a, b, A, B, ϕ needed here are taken from a single model instance.
- Stage 2: For the bistable case, stage 2 is analogous to what is described above: the constraint c_{square} defined by Equ. (4) does not occur. Constraint c_{stretch} is modified to fit the stretch factors λ_1, λ_2 computed by Equ.(2):

$$\begin{aligned} c_{\text{stretch},1}(f, f') &= \|\mathbf{v}_0 - \mathbf{v}_2\|^2 - \lambda_1^2 \|\mathbf{v}_0' - \mathbf{v}_2'\|^2 = 0, \\ c_{\text{stretch},2}(f, f') &= \|\mathbf{v}_1 - \mathbf{v}_3\|^2 - \lambda_2^2 \|\mathbf{v}_1' - \mathbf{v}_3'\|^2 = 0. \end{aligned} \tag{10}$$

- Stage 3: Initializing the final pass of optimization is analogous but a bit more complex, since we have to map the rectangular model configuration of Fig. 5 to quadrilaterals that are rectangular only in an approximate way. This is done by a standard least-squares approximation, taking care to match neighbours. As to optimization, we impose isometry constraints between corresponding pieces analogous to Equ. (7) as hard constraints, we impose proximity to the reference surface as a soft constraint, and we do not impose fairness.

3.5. Performance of optimization

We exemplarily show how optimization progresses by means of a typical example, namely the one of Fig. 14, bottom. Its control mesh has 2362 vertices, 4614 edges and 2253 faces. Weights corresponding to soft constraints (proximity to reference and fairness) were set to $w_{\text{prox}} = w_{\text{fair}} = 0.01$, while weights of hard constraints were set to $w_{\text{orth}} = w_{\text{geom}} = 1$. During the 10 iterations we performed, the fairness and closeness energies behave as follows.

iterations	0	1	2	3	4	5	6	7	8	9
$10^4 \cdot E_{\text{fair}}$	4.9	12.4	10.3	10.2	10.1	10.0	9.5	9.5	9.3	9.2
$10^4 \cdot E_{\text{prox}}$	89k	137	0.34	0.075	0.072	0.072	0.071	0.070	0.070	0.069

E_{fair} successfully acts as a regularizer. It cannot achieve zero residual (that would only be the case if mesh polylines are straight). Hard constraints are c_{square}, c_{stretch}, and c_{orth} whose sums of squares yield energies E_{geom} and E_{orth}. Their values decrease in the following way:

iterations	0	1	2	3	4	5	6	7	8	9
$10^4 \cdot E_{\text{geom}}$	2100k	26k	37	1.3	1.2	1.1	1.1	1.0	1.0	1.0
$10^4 \cdot E_{\text{orth}}$	1020k	16k	42	1.0	1.2	1.2	1.2	1.2	1.1	1.1

The total energy thus was reduced from 321.5 to 0.0011 in the last stage of iteration for Stage 2 in a total runtime of 1.1 seconds.

Iteration for Stage 3 is analogous. We have energies E_{close} as in stage 2, while E_{fair} is composed from the fairness constraint of Equ. (9). The weights for soft proximity and fairness constraints are set to 0.01. Isometry and matching constraints according to Equ. (7) and (8) are seen as hard constraints and are used to build energies E_{isom} and E_{match}. Those are given weight 1. The behaviour of these energies over the first 10 iterations is as follows:

iterations	0	1	2	3	4	5	6	7	8	9
$10^4 \cdot E_{\text{prox}}$	7.8	1.3	0.20	0.15	0.13	0.11	0.09	0.08	0.06	0.05
$10^4 \cdot E_{\text{fair}}$	382	31	21	17	15	13	12	11	9.8	9.1
$10^4 \cdot E_{\text{isom}}$	2060k	62k	22	21	9.0	5.9	4.4	3.1	2.1	1.3
$10^4 \cdot E_{\text{match}}$	0	0.03	0.006	0.003	0.002	0.002	0.002	0.001	0.001	0.001

The runtime was 5.5 seconds. It can be clearly observed that the fairness energy does not converge to zero residual. In order to overcome this deficiency we add another 10 iterations, with the weights of soft constraints set to 0. They need an additional 1.7 seconds of runtime and achieve the following values:

iterations	10	11	12	13	14	15	16	17	18	19
$10^4 \cdot E_{\text{isom}}$	0.83	$8{\cdot}10^{-3}$	$3{\cdot}10^{-4}$	$2{\cdot}10^{-5}$	$2{\cdot}10^{-6}$	$7{\cdot}10^{-7}$	$3{\cdot}10^{-7}$	$2{\cdot}10^{-7}$	$1{\cdot}10^{-7}$	$7{\cdot}10^{-8}$
$10^4 \cdot E_{\text{match}}$	$9{\cdot}10^{-4}$	$8{\cdot}10^{-8}$	$7{\cdot}10^{-9}$	$1{\cdot}10^{-9}$	$1{\cdot}10^{-10}$	$2{\cdot}10^{-11}$	$6{\cdot}10^{-12}$	$2{\cdot}10^{-12}$	$8{\cdot}10^{-13}$	$4{\cdot}10^{-13}$

For other examples in this paper (Figures 13, 14, 15, and 16) the weights were chosen as described above. The examples are scaled so that the average edge length is 1. On the other hand the examples of Fig. 12 are scaled such that the bounding box diameter is 1, the weights of hard constraints are set to 1, while the weights of soft fairness constraints are set to 0.005 and phase out towards zero as the optimization progresses. We did not observe that either of these two regimes performs better than the other one.

4. Results

We performed numerical experiments and actually manufactured several geometric metamaterials to show the capabilities of the two cases treated in this paper.

4.1. Results of numerical optimization

Stages 1–3 as described above can be directly employed to compute a morph between a flat state and a three-dimensional state, see Figure 11. The procedure still works in the presence of combinatorial singularities (see Figure 12), and morphing capabilities are expanded if cuts are applied.

There are no theoretical restrictions as to the direction of the morph — opening a cut pattern can transform a 2D domain to a 3D surface and also vice versa (Fig. 13).

Shape morphs between a curved and a flat state may be more relevant for applications, but we also applied the procedure to mappings from one double-curved surface to another. Figure 14 shows two examples.

The algorithms laid out in detail in Section 3 can be modified in some obvious ways. When removing the constraints of proximity to the target surface in our optimization, we can perform experimental form-finding, and examples like the one of Figure 15 emerge. Obviously, this result can be unique only up to isometric deformations, and it is not clear a priori to which of the many

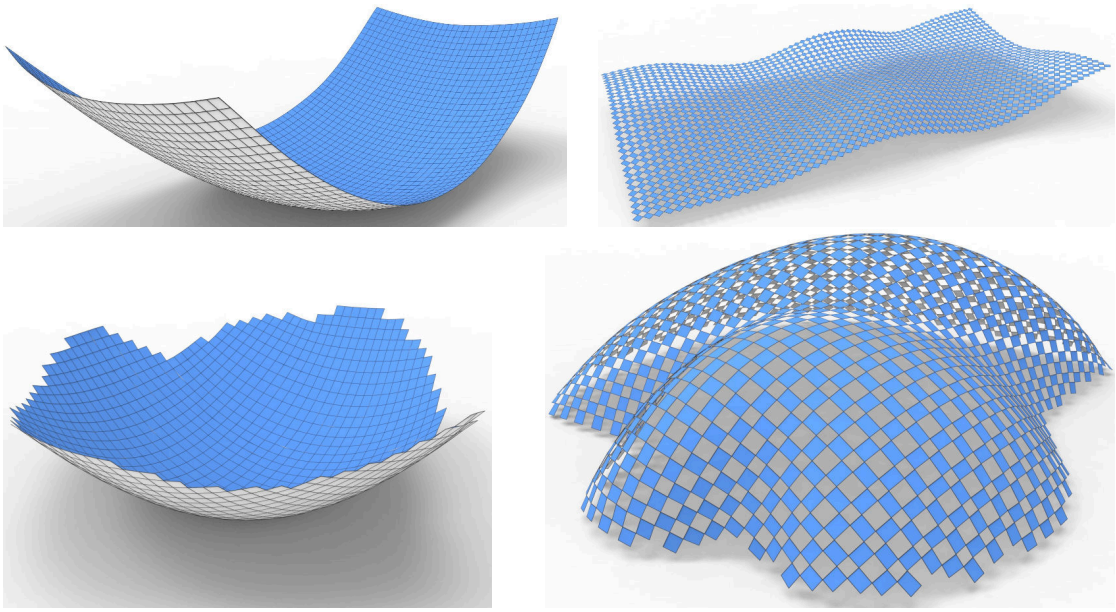


Figure 14: Here we designed a mechanical metamaterial which morphs a double-curved surface (a paraboloid) to a given target surface. The example at the top is the surface of the roof of the Islamic Arts exhibition in the Louvre, Paris.

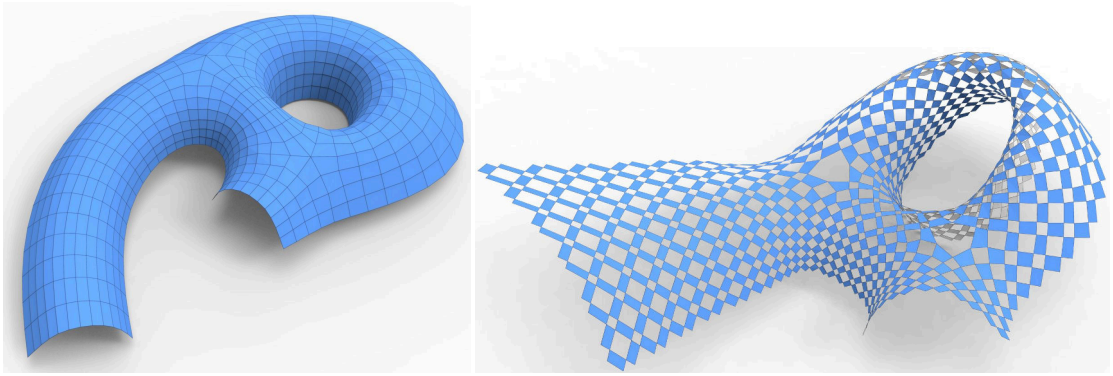


Figure 15: Form-finding. Here a cut pattern has been drawn on a given surface. When it opens fully, the example at right emerges. The computation uses the procedures of §3, but without the constraint that the result has to be close to a given target surface.

possible isometric solutions the optimization converges. It would not be difficult to combine our optimization with user-guided isometries, as recently proposed by [43].

Another modification allows us to generalize the morph between elementary cells. Figure 3 describes a motion of rigid elements between a fully closed and a fully open state. When we allow a variable opening, we are capable of reproducing more general deformations between surfaces. Stages 1–3 of the algorithm remain almost unchanged: The initialization is from general deformations (like in the bistable case), and in order to allow incomplete opening, one simply disregards the fairness terms c_{fair} $c_{\text{fair},s}$. Figures 1 and 16 show examples.

As to *bistable* metamaterials, Figure 17 shows an example which we initialized via a mapping

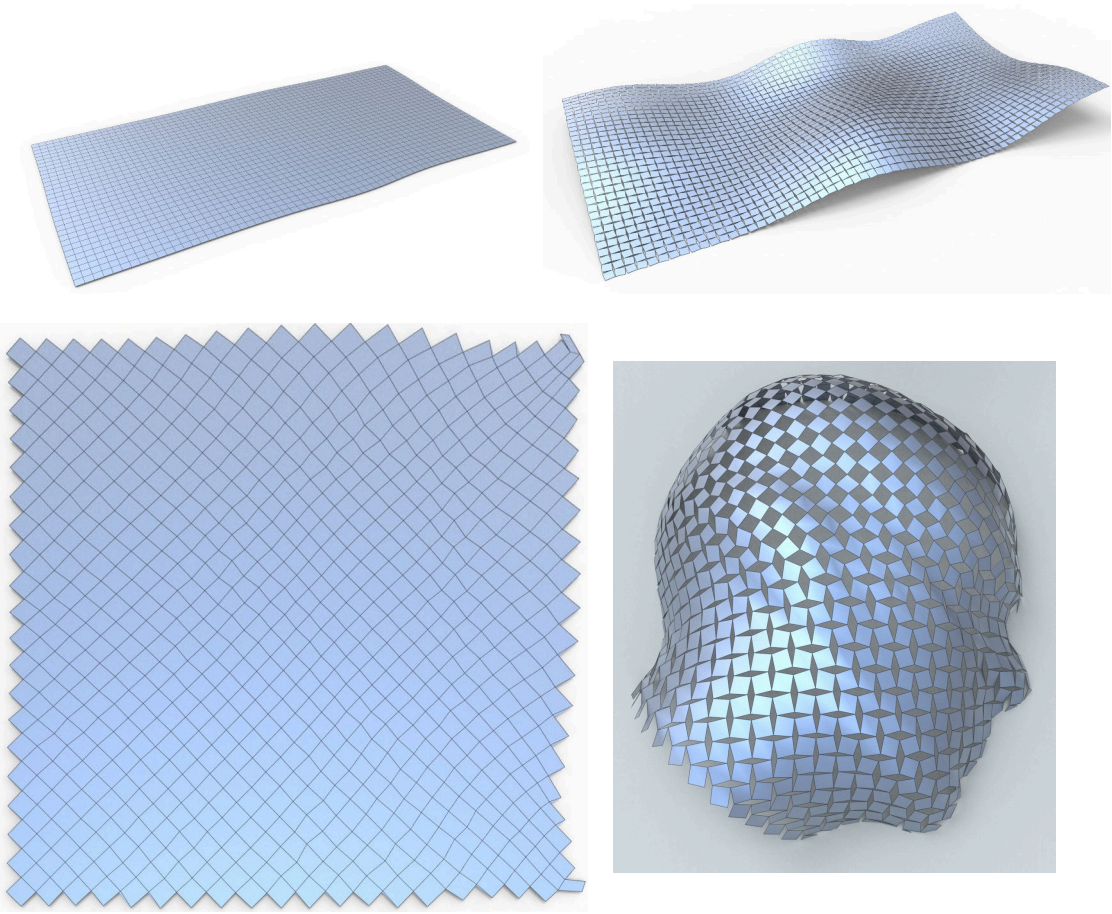


Figure 16: These examples show the capabilities of quad-based cut patterns that are similar to the constant-mean-stretch case, but exhibit a variable degree of expansion depending on the location.

from a sphere to a target surface.

So far we have treated the rigid constituent elements of geometric metamaterials as two-dimensional objects. This can be the case only approximately. In fact the two examples of Figure 18 show that three-dimensional effects can be quite noticeable and must not be summarily neglected. For each individual rigid member M_j our optimization yields an initial position and a final position which arises from the initial position by applying a rotation matrix A_j and a translation. If pieces M_j, M_k are connected, their relative movement is described by the rotation matrix $A_j^{-1}A_k$ or $A_k^{-1}A_j$, depending on which piece is the observer. The axis of either of these two relative rotations indicates the direction of the hinge that can exist between pieces M_j, M_k. Hinges computed in this way are visualized by Fig. 19. We return to the topic of hinges below, in our discussion on fabrication.

4.2. Fabrication

We experimentally confirm the viability of the proposed geometric metamaterials by building models a person can handle with their hands — see videos

- https://vimeo.com/524221493/b7d951751c (corresponding to Fig. 2),

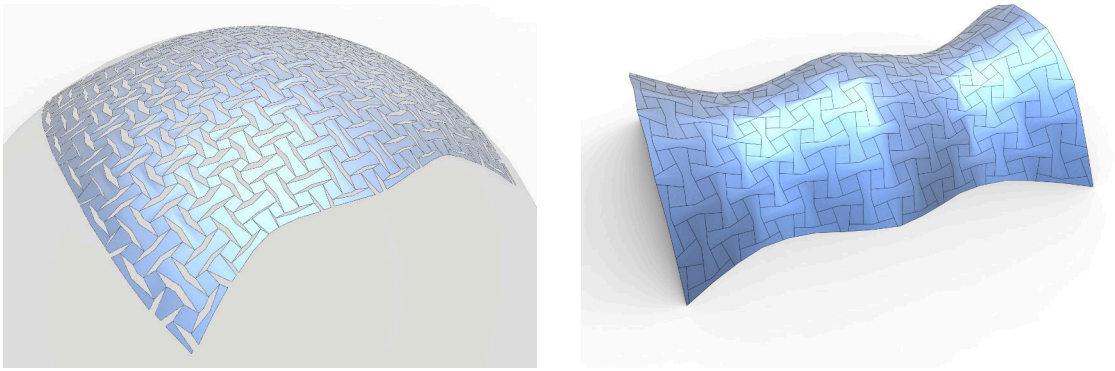


Figure 17: A hypothetical bistable material that snaps between two surfaces.

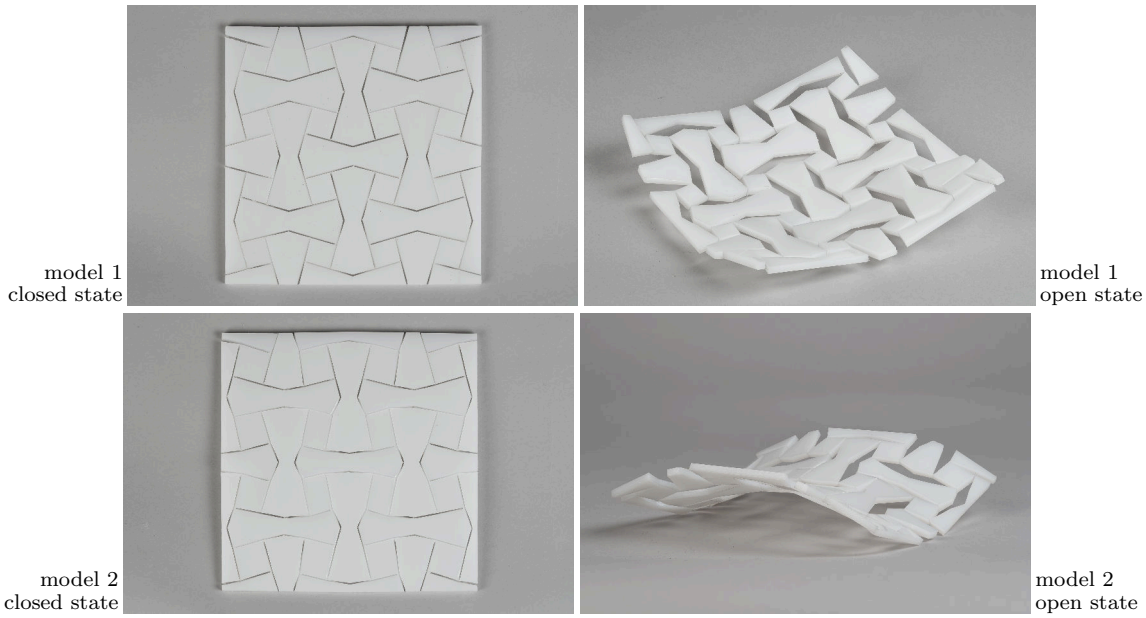


Figure 18: These bistable geometric materials snap between a flat and a spatial state, demonstrating that the curvature programmed into a pattern might not be visible to the naked eye. The example on top has positive curvature, the example at the bottom enjoys negative curvature. The actual manufacturing from a flat sheet has to take the spatial rigid body motion of geometric primitives into account: During the shape morph, each primitive is rotating against its neighbours, and the thin hinge holding two individual pieces together must be placed along a correctly computed axis of rotation (cf. Fig. 19 below).

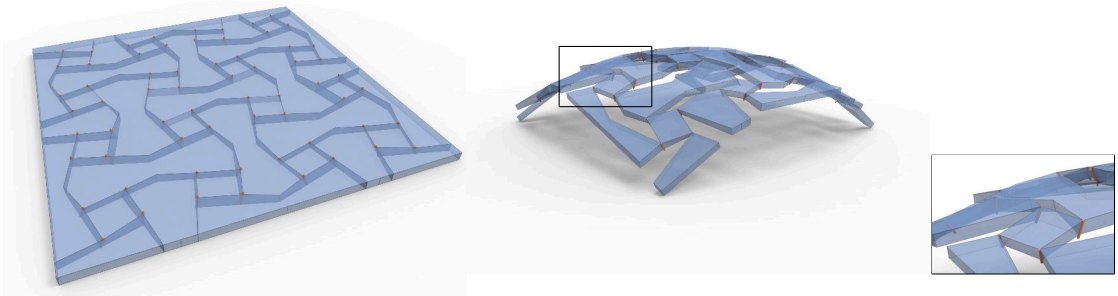


Figure 19: The axes of rotation between connected primitives occurring in a shape morph are here highlighted in red. This rendering corresponds to Fig. 18, top.

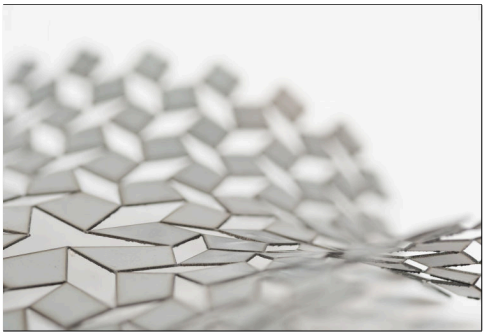
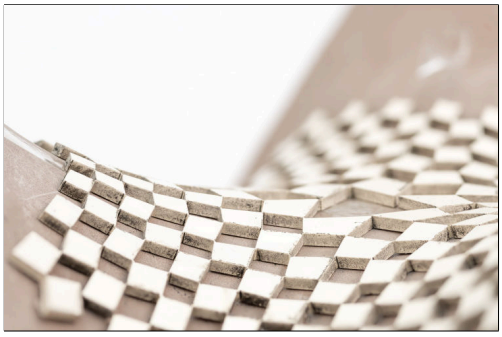



Figure 20: We experimented with materials allowing the manufacturing of geometric metamaterials. Besides Nylon 6, which was used for the examples of Figures 6 and 18, we employ stainless steel for the example at left (see also Fig. 1). Nitrile rubber, here sitting on a PU foam mold, was used for the example at right.

- https://vimeo.com/524221586/a53a1f1064 (corresponding to Fig. 18, top),
- https://vimeo.com/524221545/c6b63f9480 (corresponding to Fig. 18, bottom).

We restricted ourselves to cases where either the closed or the open state is flat. We employed plastic, metal and rubber (Fig. 20). In detail, we worked as follows.

- a 5mm "Nylon 6" (polycaprolactam) sheet was used for the examples of Figures 6 and 18. We employed 5-axis milling to cut it, using a 0.5mm cylindrical tool.
- "304 stainless steel" of 0.5mm thickness has been employed for the model shown by Fig. 1. It was cut by a Q-switched Yb:YAG Laser.
- We also experimented with acrylonitrile butadiene rubber (Nitrile rubber), see Fig. 20, right. It was cut by a 200 Watt CO_2 laser.

A shape morph in space engenders a rotating motion between any two primitives that are connected. In manufacturing, this hinge is simply realized as a thin piece of material. Its actual dimensions are crucial for striking a balance between avoiding fatigue, avoiding stress concentration, and ease of the shape morph, depending on the application at hand. For sheets of a certain thickness, e.g. those of Fig. 18, it is important that for any two primitives connected by a hinge, that hinge is positioned in the axis of rotation (Fig. 19) Otherwise failure must be expected. For the models we built, the details are as follows:

- When milling Nylon 6, hinges are produced by a 0.5mm radius cylindrical tool, which gives the lowest radius of curvature; they have a minimum thickness of 0.5mm. Increasing the length by 1mm makes the model too loose and already prevents the desired snapping in the bistable cases.
- In the stainless steel example of Fig. 1 hinges have overall dimensions 0.5mm×0.5mm×0.5mm. Extra T-shaped cuts reduce stress concentration.

4.3. Conclusion

We have demonstrated the capabilities of *kirigami* cut and fold patterns for shape-morphing metamaterials. The state of the art in Geometry Processing and mathematical optimization allows us to realize both continuous and snap-like deformations between shapes. It is noteworthy that these deformations are not limited by the shapes themselves (which can be arbitrary) but

by their maximum and minimum principal stretches. In particular our method enables the programming of curvature into almost-regular flat patterns. Our method is based on mappings between surfaces (shape morphs). In the process of optimization, this mapping emerges, and elementary material cells align with its principal stretch directions.

A limitation of our work is the need to find a deformation whose principal stretches lie between 1 and the allowable maximum. In our implementation we did not make full use of previous work, e.g. [39, 44].

Future work includes a thorough differential-geometric analysis of the questions mentioned in §2.3. On the practical side, it would not be difficult to implement a more explicit relationship between mappings on the one hand, and metamaterials on the other hand. This would require remeshing along principal stretch directions and is treated e.g. by [45]. Future work also includes extending the capabilities of our limited implementation, e.g. regarding mappings between surfaces which respect their boundaries.

Acknowledgments

This work was supported by the Austrian Science Fund via grants I2978 (SFB-Transregio programme Discretization in geometry and dynamics), F77 (SFB grant Advanced Computational Design); further by the Vienna Science and Technology Fund (WWTF) under grant ICT15-082. C. Jiang, F. Rist, and H. Wang were supported by KAUST baseline funding.

References

[1] Z. Wei, Z. V. Guo, L. Dudte, H. Liang, L. Mahadevan, Geometric mechanics of periodic pleated origami, Phys. Rev. Lett. 110 (2013) 215501:1–5.

[2] L. H. Dudte, E. Vouga, T. Tachi, L. Mahadevan, Programming curvature using origami tessellations, Nature Materials 15 (2016) 583–588.

[3] S. M. Felton, M. T. Tolley, B. Shin, C. D. Onal, E. D. Demaine, D. Rus, R. J. Wood, Self-folding with shape memory composites, Soft Matter 9 (2013) 7688–7694.

[4] T. Tachi, Generalization of rigid foldable quadrilateral mesh origami, in: Proc. IASS Symposium, U Politècnica València, 2009, pp. 2287–2294.

[5] C. D. Santangelo, Extreme mechanics: Self-folding origami, Ann. Rev. Condensed Matter Phys. 8 (2016) 165–183.

[6] T. Mukhopadhyay, J. Ma, H. Feng, D. Hou, J. M. Gattas, Y. Chen, Z. You, Programmable stiffness and shape modulation in origami materials: emergence of a distant actuation feature, Appl. Mater. Today 19 (2020) 100537:1–7.

[7] A. Islam, H. N. Hansen, P. T. Tang, J. Sun, Process chains for the manufacturing of molded interconnect devices, Int J. Adv. Manuf. Technol. 42 (2009) 831–841.

[8] J. T. Karras, C. L. Fuller, K. C. Carpenter, A. Buscicchio, D. McKeeby, C. J. Norman, C. E. Parcheta, I. Davydychev, R. S. Fearing, Pop-up Mars rover with textile-enhanced rigid-flex PCB body, in: IEEE Int. Conf. Robotics & Automation, 2017, pp. 5459–5466.

[9] J. Grima, K. Evans, Auxetic behavior from rotating squares, J. Mat. Sc. Lett. 19 (2000) 1563–1565.

[10] J. Grima, A. Alderson, K. Evans, Auxetic behaviour from rotating rigid units, Physica Status Solidi B 242 (2005) 561–575.

[11] J. Grima, E. Manicaro, D. Attard, Auxetic behaviour from connected different-sized squares and rectangles, Proc. Royal Soc. A 467 (2011) 439–458.

[12] M. Konaković, K. Crane, B. Deng, S. Bouaziz, D. Piker, M. Pauly, Beyond developable: Computational design and fabrication with auxetic materials, ACM Trans. Graph. 35 (4) (2016) 89:1–11.

[13] M. Konaković-Luković, J. Panetta, K. Crane, M. Pauly, Rapid deployment of curved surfaces via programmable auxetics, ACM Trans. Graph. 37 (4) (2018) 106:1–13.

[14] T. Chen, J. Panetta, M. Schnaubelt, M. Pauly, Bistable auxetic surface structures, ACM Trans. Graph. 40 (4) (2021) 39:1–9.

[15] F. Wang, X. Guo, J. Xu, Y. Zhang, C. Q. Chen, Patterning curved three-dimensional structures with programmable kirigami designs, J. Appl. Mech. 84 (2017) 061007:1–7.

[16] T. Castle, Y. Cho, X. Gong, E. Jung, D. M. Sussman, S. Yang, R. D. Kamien, Making the cut: Lattice kirigami rules, Phys. Rev. Lett 113 (2014) 245502,1–5.

[17] T. Castle, D. M. Sussman, M. Tanis, R. D. Kamien, Additive lattice kirigami, Sci. Advances 2 (2016) e1601258:1–11.

[18] A. Rafsanjani, D. Pasini, Bistable auxetic mechanical metamaterials inspired by ancient geometric motifs, Extreme Mechanics Lett. 9 (2016) 291–296.

[19] S. Callens, A. Zadpoor, From flat sheets to curved geometries: Origami and kirigami approaches, Mater. Today 21 (2018) 241–264.

[20] G. Choi, L. Dudte, L. Mahadevan, Programming shape using kirigami tessellations, Nature Materials 18 (2019) 999–1004.

[21] K. E. Evans, Auxetic polymers: a new range of materials, Endeavour 15 (4) (1991) 170–174.

[22] L. Mizzi, E. Salvati, A. Spaggiari, J.-C. Tan, A. M. Korsunsky, Highly stretchable two-dimensional auxetic metamaterial sheets fabricated via direct-laser cutting, Int. J. Mechanical Sc. 167 (2020) 105242:1–14.

[23] Y. Tang, Y. Li, Y. Hong, J. Yin, Programmable active kirigami metasheets with more freedom of actuation, Proc. Nat. Acad. Sci. U.S.A. 116 (2019) 26407–26413.

[24] N. An, A. G. Domel, J. Zhou, A. Rafsanjani, K. Bertoldi, Programmable hierarchical kirigami, Adv. Functional Mater. 30 (6) (2020) 1906711.

[25] P. Celli, C. Mcmahan, B. Ramirez, A. Bauhofer, C. J. Naify, D. C. Hofmann, B. Audoly, C. Daraio, Shape-morphing architected sheets with non-periodic cut patterns, Soft Matter 14 (48) (2018) 9744–9749.

[26] L. Jin, A. E. Forte, B. Deng, A. Rafsanjani, K. Bertoldi, Kirigami-inspired inflatables with programmable shapes, Adv. Mater. (2020) 2001863.

[27] A. Rafsanjani, A. Akbarzadeh, D. Pasini, Snapping mechanical metamaterials under tension, Adv. Mater. 27 (2015) 5931–5.

[28] C. Jiang, C. Wang, F. Rist, J. Wallner, H. Pottmann, Quad-mesh based isometric mappings and developable surfaces, ACM Trans. Graph. 39 (4) (2020) 128:1–13.

[29] P. Laskowski, The traditional and modern look at Tissot's indicatrix, The American Cartographer 16 (2) (1989) 123–133.

[30] K. Strubecker, Differentialgeometrie: Theorie der Flächenmetrik, Vol. 1179 of Sammlung Göschen, De Gruyter, Berlin, 1969.

[31] M. Lai, E. Krempl, D. Ruben, Introduction to Continuum Mechanics, 4th Edition, Butterworth-Heinemann, 2010.

[32] G. K. Batchelor, An Introduction To Fluid Dynamics, Cambridge University Press, 1967.

[33] R. Sauer, Differenzengeometrie, Springer, 1970.

[34] W. Schief, A. Bobenko, T. Hoffmann, On the integrability of infinitesimal and finite deformations of polyhedral surfaces, in: A. Bobenko, et al. (Eds.), Discrete differential geometry, Vol. 38 of Oberwolfach Seminars, Springer, 2008, pp. 67–93.

[35] K. Sharifmoghaddam, G. Nawratil, A. Rasoulzadeh, J. Tervooren, Using flexible trapezoidal quad-surfaces for transformable design, in: S. A. Behnejad, et al. (Eds.), Proc. IASS Annual Symposium 2020/21 and the 7th Int. Conf. on Spatial Structures, 2021.

[36] A. Bobenko, Y. Suris, Discrete differential geometry: Integrable Structure, American Math. Soc., 2009.

[37] C.-H. Peng, C. Jiang, P. Wonka, H. Pottmann, Checkerboard patterns with black rectangles, ACM Trans. Graph. 38 (6) (2019) 171:1–13.

[38] K. Madsen, H. B. Nielsen, O. Tingleff, Methods for non-linear least squares problems, 2nd Edition, Technical Univ. Denmark, 2004.

[39] O. Sorkine, M. Alexa, As-rigid-as-possible surface modeling, in: Proc. Symp. Geometry Processing, 2007, pp. 109–116.

[40] L. Liu, L. Zhang, Y. Xu, C. Gotsman, S. Gortler, A local/global approach to mesh parameterization, Comput. Graph. Forum 27 (5) (2008) 1495–1504.

[41] H. Pottmann, Q.-X. Huang, Y.-L. Yang, S.-M. Hu, Geometry and convergence analysis of algorithms for registration of 3D shapes, Int. J. Computer Vision 67 (3) (2006) 277–296.

[42] D. Bommes, H. Zimmer, L. Kobbelt, Mixed-integer quadrangulation, ACM Trans. Graph. 28 (3) (2009) 77:1–10.

[43] C. Jiang, H. Wang, V. Ceballos Inza, F. Dellinger, F. Rist, J. Wallner, H. Pottmann, Using isometries for computational design and fabrication, ACM Trans. Graph. 40 (4) (2021) 42:1–12.

[44] L. Liu, L. Zhang, Y. Xu, C. Gotsman, S. J. Gortler, A local/global approach to mesh parameterization, Computer Graphics Forum 27 (5) (2008) 1495–1504.

[45] C. Jiang, F. Rist, H. Pottmann, J. Wallner, Freeform quad-based kirigami, ACM Trans. Graph. 39 (6) (2020) 209:1–11.