Universal Hinge Patterns for Folding Orthogonal Shapes

Nadia M. Benbernou [*] Erik D. Demaine[*†]
Martin L. Demaine[*] Aviv Ovadya[*‡]

1 Introduction

An early result in computational origami is that every polyhedral surface can be folded from a large enough square of paper [DDM00]. A recent algorithm for this problem even attains practical foldings [DT10]. But each polyhedral surface induces a completely different crease pattern. Is there a single hinge pattern for which different subsets fold into many different shapes?

Our motivation is developing programmable matter out of a foldable sheet [HAB$^+$10]. The idea is to statically manufacture a sheet with specific hinges that can be creased in either direction, and then dynamically program how much to fold each crease in the sheet. Thus a single manufactured sheet can be programmed to fold into anything that the single hinge pattern can fold.

We prove a universality result: an $N \times N$ square tiling of a simple hinge pattern can fold into all face-to-face gluings of $O(N)$ unit cubes (polycubes). Thus, by setting the resolution N sufficiently large, we can fold any 3D solid up to a desired accuracy.

The proof is algorithmic: we present the *Cube Extrusion Algorithm* which converts a given polycube into a crease pattern (a subset of the universal hinge pattern) and a 3D folded state in the shape of that polycube, with seamless faces. Figure 1 shows a simple example.

At the core of our algorithm is the notion of a *cube gadget*, which folds a cube in the middle of a sheet of paper. Such foldings of a single cube

[*]MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA, {nbenbern,edemaine,mdemaine,avivo}@mit.edu

[†]Partially supported by NSF CAREER award CCF-0347776, DOE grant DE-FG02-04ER25647, and AFOSR grant FA9550-07-1-0538.

[‡]Corresponding Author

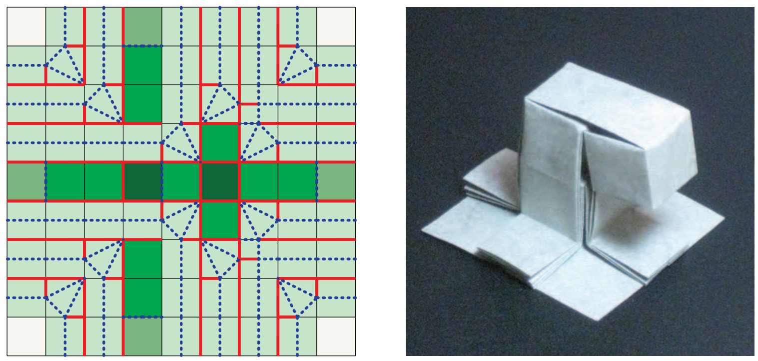

Figure 1: Folding a bend-shaped polycube with a square base via the Cube Extrusion Algorithm. For simplicity, the mountain-valley pattern in this and other figures does not exactly show the reflected creases when multiple layers are folded.

have been independently developed by origamists over the years; the first documented design we are aware of was created by David A. Huffman in 1978.[1] The novelty is the way we combine multiple cube gadgets to form a desired polycube. We present three different cube gadgets, one of which is the gadget independently created by Huffman, each with its own advantages and disadvantages when combined to fold a polycube.

We also describe an implementation of the algorithm which can be used to automate experimentation and design of geometric origami using a cutting plotter or laser cutter to score the paper.

2 Definitions

We start with a few definitions about origami, specified somewhat informally for brevity. For more formal definitions, see [DO07, ch. 11].

For our purposes, a *piece of paper* is a connected collection of flat polygons in 3D joined along shared edges (a polyhedral complex; note that we can have multiple polygons in the same place but with different connections, and with a specified stacking order). A notable special case is a single $m \times n$ rectangle of paper for integers m and n (but in general, a piece of paper does not have to be flat; for example, a polyhedron is a piece of paper). We index the unit squares of such a rectangle in the style

[1] Personal communication with the Huffman family. The fourth author independently developed this gadget in middle school circa 2000.

of matrices: $s_{i,j}$ refers to the unit square in the ith row and jth column, and $s_{1,1}$ is in the upper-left corner.

A *hinge* is a line segment drawn on a piece of paper which is capable of being creased in either direction. A *hinge pattern* is a collection of hinges drawn on a piece of paper. The hinge patterns we consider in this paper are all based on subdivisions of the unit-square grid, adding a finite number of hinges within each unit square. The unit squares of the hinge pattern correspond to the unit squares of a rectangle of paper.

An example of a hinge pattern is the *box-pleated pattern* (known in geometry as the *tetrakis tiling*) which is formed from the unit-square grid by subdividing each square in half vertically, horizontally, and by the two diagonals, forming eight right isosceles triangles. The upper-left corner of Figure 3 shows an example for four unit-squares.

An *angle pattern* is a hinge pattern together with an assignment of a real number in $[-180°, +180°]$ to each hinge, specifying a fold angle (negative for valley, positive for mountain). We allow a hinge to be assigned an angle of 0, in which case we call the hinge *trivial*, though we do not draw trivial hinges in most figures. A hinge with a nonzero angle is called a *crease*. The *crease pattern* is the subgraph of the hinge pattern consisting of only the creases.

An angle pattern determines a 3D geometry called the *folded geometry*, which maps each face of the crease pattern to a 3D polygon via a Euclidean isometry (by the composition of rotations at creases). More explicitly, a folded geometry is a map from all points of the piece of paper to $\mathbb{R}^3$ that satisfies constraints as specified in [DO07, ch. 11]—and there is an obvious mapping from angle patterns to folded geometries.

A *folded state* consists of such a folded geometry together with an ordering λ, which is a partial function over the touching points in the folded geometry, in our case describing the stacking relationship among polygons of the crease pattern that touch in the folded geometry. Define the *starting sheet* of a folded state to be the original piece of paper, that is, the domain of the folded geometry. [2] A *folding sequence* is a sequence of folded states $F_1, F_2, \ldots, F_k$ from the same starting sheet. The last folded state in a folding sequence is called the *final folded state*.

Define the *number of layers at a point q* to be the number of noncrease points in the piece of paper that get mapped to q by the folded geometry. The *number of layers of a folded state* is the maximum number of layers over all points.[3]

Next we define a notion of "coalescing" which lets us ignore certain

[2]Note that "sheet" is not to suggest that the piece of paper needs to be flat; it can be any polyhedral complex.

[3]This measure is a simple way to bound the effect of paper thickness, but in practical origami there are other quantities that could be measured.

details of a folded state. A *coalesce folded state* is a folded state augmented with a *coalesce set* which is a subset of the starting sheet. If we take the starting sheet and identify (glue together) all pairs of points in the coalesce set that are collocated by the folded geometry, then we obtain a metric space called the *coalesce result*. This coalesce result is also a piece of paper under our definition and therefore can be the domain of a new coalesce folded state. A *coalesce sequence* is a sequence $C_1, C_2, \ldots, C_k$ of coalesce folded states, where each C_k is a folding of the coalesce result of C_{k-1}.

One can generate a folding sequence from a coalesce sequence by letting $F_1 = C_1$ and $F_k = F_{k-1} \circ C_k$, and then composing the geometry and ordering functions in the obvious way. Note that the starting sheet of each F_k is the starting sheet of C_1, while the starting sheets of the other C_k's can be any shape folded from that starting sheet. The *final folded state of a coalesce sequence* is the final folded state of the generated folding sequence. We say that a folding sequence or coalesce sequence *folds a piece of paper* π *into a shape* σ if the starting sheet of the first folded state is the piece of paper π and the image of the last folded geometry is the shape σ.

In this paper we allow all but the last folded state in a folding sequence or coalesce sequence to have crossings. By [DDMO04], all folded states are reachable from the starting sheet by the continuous folding motion, so the final folded state is still reachable. We use folding sequences as a tool to construct the final folded state, not as instructions for folding.

Now that we can describe how to fold a shape, we define our target shapes. A *polycube* P is a union of unit cubes on the unit-cube lattice with a connected *dual graph*; the dual graph has a vertex for each unit cube and an edge between two vertices whose corresponding cubes share a face. The *faces* of the polycube are the (square) faces of the individual cubes that are not shared by any other cubes.

A *folding of a polycube* is a folded state that covers all faces of the polycube, and nothing outside the polycube. In fact, some of our foldings of polycubes will also include the internal squares, the faces shared by multiple cubes, and some of our foldings will not put anything else interior to cubes, but in general we do not require either property. A face of a folded polycube is *seamless* if the outermost layer of paper covering it is an uncreased unit square of paper. Our foldings will generally be seamless.

3 Cube Gadgets

We now introduce the notion of a cube gadget; refer to Figure 2. For positive integers r and c, an $[r, c]$-*cube gadget* is a method of extruding a cube from a rectangular piece of paper at a specified location. The input to the cube gadget is an $m \times n$ rectangle of paper, for integers $m > 2r$ and

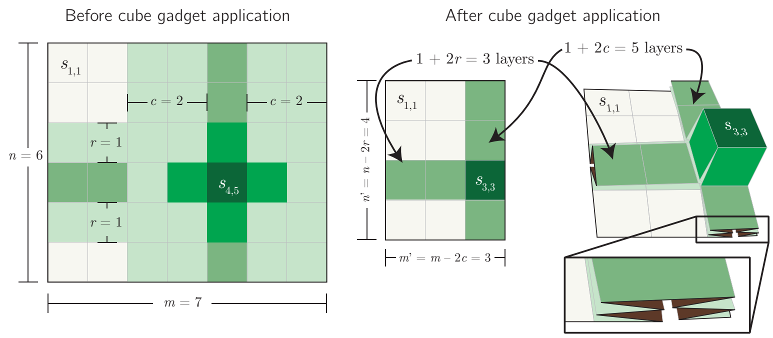

Figure 2: Abstract effect of applying a $[1, 2]$-cube gadget at square $s_{4,5}$ of a 6×7 rectangle of paper. The two leftmost diagrams are top views before and after folding. The right diagram is a stylized perspective view of the folded state.

$n > 2c$, as well as a unit square $s_{i,j}$ on the paper, where $r < i < m - r$ and $c < j < n - c$. The output of the cube gadget is a folding of the $m \times n$ rectangle into the shape of a cube sitting on a smaller, $(m - 2r) \times (n - 2c)$ rectangle of paper. The cube sits on the square $s_{i-r,j-c}$ in the smaller sheet of paper. All six faces of the cube are seamless except for the bottom face. The top face of the cube is covered by square $s_{i,j}$ from the original piece of paper. The boundary of the original $m \times n$ rectangle paper is mapped onto the boundary of the smaller $(m - 2r) \times (n - 2c)$ rectangle.

The cube gadgets in this paper achieve the folding by making horizontal pleats in the r rows above and below row i, and making vertical pleats in the c columns left and right of column j. The pleats are called *half-square pleats* because they are composed of unit squares folded in half.

Each pleat adds two layers to the row or column it is under, so the number of layers of the folded state is at least $1 + 2 \max\{r, c\}$ (one for the row or column plus two for each pleat). Another property of our foldings is that all folding is within the $2r + 1$ rows and $2c + 1$ columns surrounding square $s_{i,j}$. Thus, the quadrant of paper consisting of rows $< i - r$ and columns $< j - c$ is not folded and is incident to the top-left corner of the cube, and similarly for the other four quadrants.

In this paper, we give three different cube gadgets based on three different hinge patterns, as shown in Figure 3. The three cube gadgets are based, respectively, on the box-pleated pattern, the slit pattern, and the arctan $\frac{1}{2}$ pattern.

The arctan $\frac{1}{2}$ gadget and slit gadget are $[1, 1]$-cube gadgets, while the box-pleated gadget is a $[1, 2]$-cube gadget. The advantage of the box-pleated and slit gadgets is that the hinge pattern is simpler: box pleating

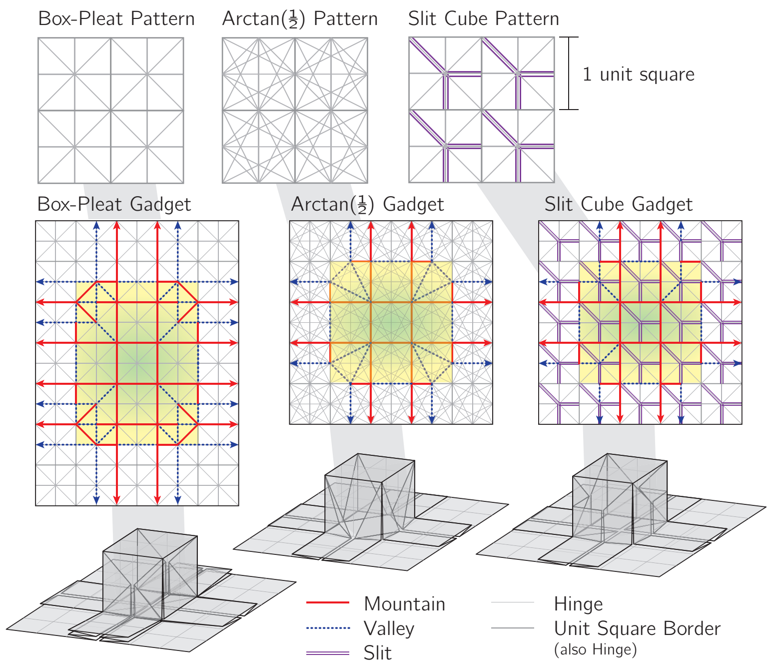

Figure 3: The hinge patterns (top), mountain-valley patterns (middle), and semitransparent folded states (bottom) for the three cube gadgets. The highlighted region of each mountain-valley pattern has dimensions $2r+1 \times 2c+1$ and is the region used to actually fold the cube (half-square pleats extend out from those regions).

has all creases with angles at integer multiples of 45°. The slit gadget attains higher efficiency than seems possible with regular box pleating by adding a regular pattern of slits in the paper. The arctan $\frac{1}{2}$ gadget attains higher efficiency using more hinges some of which are at angles of arctan $\frac{1}{2}$. We use the arctan $\frac{1}{2}$ gadget in all figures for consistency.

Next, we show how a cube gadget can be used to modify an existing folding, which will be the key construction in our folding of general polycubes. Figure 4 provides some intuition for how existing cubes move as new cubes are folded and Figure 5 provides a formal example of the lemma below.

Lemma 1 (Gadget Application) *Let C^P be a coalesce sequence for a polycube P from an $m \times n$ rectangle of paper. Let f be a face of the polycube P that is seamless in the final folded state of C^P. Then there is a coalesce*

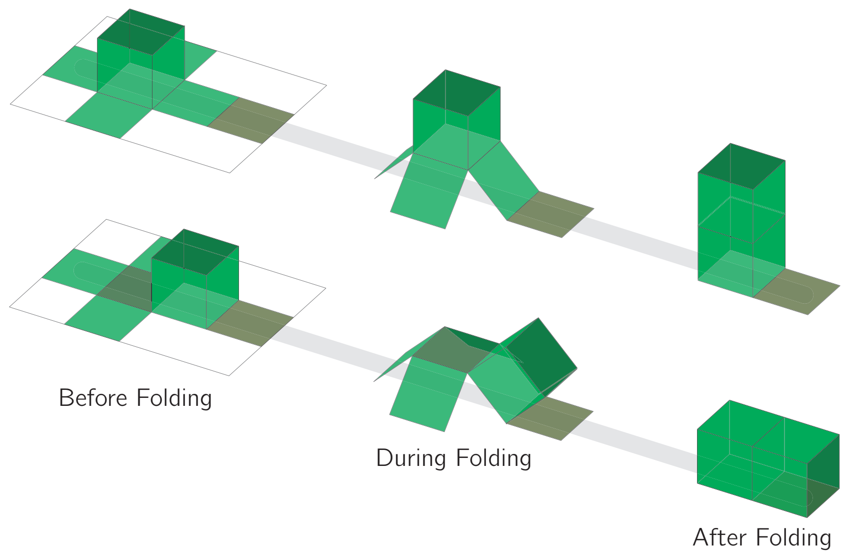

Figure 4: Given a piece of paper with a cube already folded on it, this diagram shows in an abstract manner how the paper moves when a new cube is folded depending on whether the old cube is on the top face (above), or a side face (below) of the new cube.

sequence $C^{P'}$ *for the polycube* P'*, consisting of* P *plus a cube extruded from face* f*, from an* $(m+2r) \times (n+2c)$ *rectangle of paper. The construction is parameterized by a cube gadget.*

Proof: Let $C^P = C_1^P, C_2^P, \ldots, C_q^P$, and let F_q^P be the final folded state (for P). Let $s_{i,j}$ be the square in the starting sheet of F_q^P that is mapped to f via F_q^P. We use σ to refer to this square in an abstract sense—when we add rows or columns to the start sheet σ will move with the insertions—so the square coordinates referred to by σ may change.

We construct a new coalesce folded state $C_1^{P'}$ that we will prepend to C^P. Define $C_1^{P'}$ to have the same starting sheet as that of F_q^P, except that we insert r rows above σ, r rows below σ, c columns left of σ, and c columns right of σ. So the starting sheet is a rectangle of paper of size $(m+2r) \times (n+2c)$ and σ refers to the square $s_{i+r,j+c}$ in this enlarged sheet of paper. Define this entire enlarged rectangle to be the coalesce set of $C_1^{P'}$. Now we define the folded state of $C_1^{P'}$ to be the given cube gadget applied at σ. The result looks like a cube sitting on the square $s_{i,j}$ of an $m \times n$ rectangle of paper. (The paper does have additional layers in some

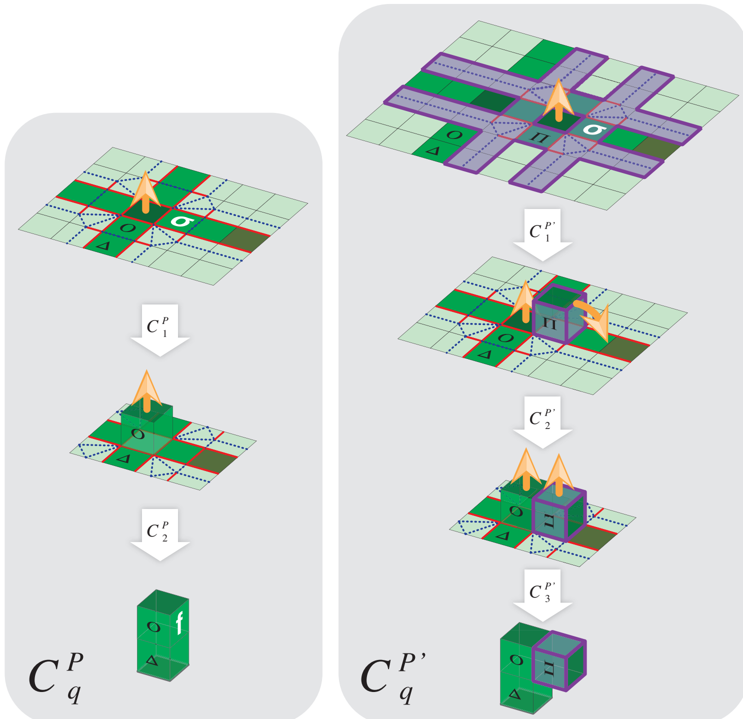

Figure 5: Given a coalesce sequence C^P, shows the application of Lemma 1 to generate $C^{P'}$ with an additional cube. The purple regions show where additional rows and columns are inserted. We use the arctan $\frac{1}{2}$ cube gadget in all figures from here onwards for consistency.

places from the pleats, but it folds flat except at the cube, and these layers are all coalesced because the entire sheet is in the coalesce set.)

Now, for each coalesce folded state C_k^P, we create a modified coalesce folded state $C_{k+1}^{P'}$ with σ replaced by a cube of paper. Here we use the fact that σ never gets folded throughout the sequence (since it is always the face of a cube), and thus corresponds to a seamless square of paper in the starting sheet of each coalesce folded state C_k^P. Note that we add the five new faces of the cube to the starting sheet, but we do not add these faces to the coalesce set of $C_{k+1}^{P'}$; the latter will remain a rectangle. We also add any polygons of paper internal to the cube that appear in $C_1^{P'}$, in the same orientation. Because the coalesce result of $C_k^{P'}$ is the starting sheet of $C_{k+1}^{P'}$, we have thus generated a new coalesce sequence $C^{P'} = C_1^{P'}, C_2^{P'}, \ldots C_q^{P'}, C_{q+1}^{P'}$.

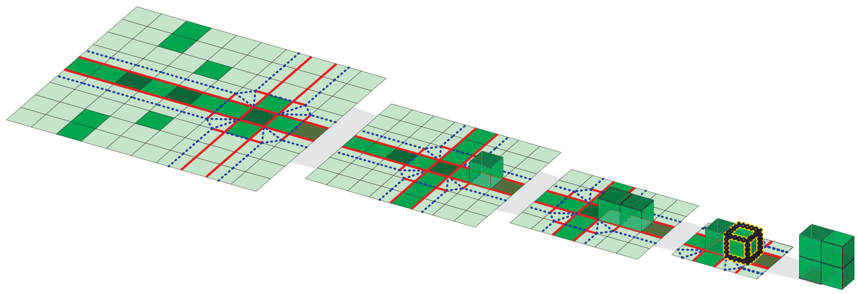

Figure 6: This sequence of folded states for a particular coalesce sequence shows an example of self intersection in part of a coalesce folding sequence. The self intersection occurs at the highlighted cube and is resolved in the final step. The self intersection can be avoided in this case by making additional simple folds.

Note that these added cubes may create intersections in the coalesce folded states $C_k^{P'}$ (as mentioned in Section 2). However, the final folded state $F_{q+1}^{P'}$ of $C^{P'}$ (as well as $C_{q+1}^{P'}$ itself) is guaranteed not to have intersections. This follows because F_q^P had no self intersections, the application $C_1^{P'}$ of the cube gadget has no self intersections, and adding the cube of paper to make P into P' cannot create intersections.

□

4 Folding Polycubes

Our *Cube Extrusion Algorithm* for folding any polycube is parameterized by an arbitrary cube gadget, and consists of repeated application of the gadget according to Lemma 1. We describe the recursive algorithm by way of an inductive proof:

Theorem 2 (Cube Extrusion Algorithm) *Any polycube of N cubes can folded with all faces seamless from a $(2rN+1)\times(2cN+2)$ rectangle of paper by a sequence of N applications of an $[r,c]$-cube gadget plus one additional fold.*

Proof: We prove by induction that any polycube P' of N cubes can be folded seamlessly by a coalesce sequence from a $(2rN+1)\times(2cN+2)$ rectangle of paper. Arbitrarily choose a "bottom face" f_b of P', and let b be the unique (bottom) cube having f_b as a face.

The base case is $N = 1$, when P' consists of the single cube b. We can use the cube gadget directly at square $s_{r+1,c+1}$ to obtain a folding of the single cube from a $(2r+1) \times (2c+2)$ rectangle of paper. The folded state is a cube next to a (pleated) unit square of paper. Note that the bottom face of the cube corresponds to f_b, and is adjacent to the square of paper. By definition of cube gadgets, all faces of the cube except the bottom face are seamless. We fold the extra square of paper over to seamlessly cover the bottom face, thus making a seamless one-cube polycube. The resulting folded state forms the first and only step in a coalesce sequence.

It remains to prove the inductive step. Let T be a spanning tree of the dual graph of P'. Because every tree has at least two leaves, T has a leaf corresponding to a cube $l \neq b$. Let u be the unique cube sharing a face with l, and let f_{ul} be the face shared by u and l.

Now consider the polycube $P = P' \setminus \{l\}$, with $N-1$ cubes. Because $l \neq b$, f_b remains a face of P. By induction, there is a coalesce sequence C^P that folds a $(2r(N-1)+1) \times (2c(N-1)+2)$ rectangle of paper into P. By Lemma 1, we extrude from f_{ul} to obtain a new coalesce sequence $C^{P'}$ for P' from a rectangle of size $((2r(N-1)+1)+2r) \times ((2c(N-1)+2)+2c) = (2rN+1) \times (2cN+2)$.

The final folded state of the inductively obtained coalesce sequence is the desired folded state from the rectangle of paper into the polycube. $\square$

Without the concern for a seamless bottom face, we can reduce the $+2$ in the rectangle bound down to $+1$.

The algorithm runs in polynomial time. The bottleneck is in converting the coalesce sequence into its final folded state. Each of the N cube gadgets causes the creation of at most $O(N)$ creases, because the piece of paper at that point has size $O(N) \times O(N)$ with $O(1)$ existing creases per square.

The size bound of an $O(N) \times O(N)$ rectangle of paper is tight up to constant factors for square paper. Specifically, folding a $1 \times 1 \times N$ tower of cubes requires starting from a square of side length N in order to have diameter N, as folding can only decrease diameter.

4.1 Hinge Pattern Completeness

Next we show that the Cube Extrusion Algorithm does not create creases that stray from the given hinge pattern.

For a rectangular piece of paper, the *tile* $t_{i,j}$ of a crease pattern is the set of creases within the unit square $s_{i,j}$. The *tile set* of a cube gadget is the set of all distinct tiles that can be generated by the cube gadget. The hinge pattern *generated* by a tile is the result of replicating the tile in a unit-square grid.

Proposition 3 *Given a cube gadget with a finite tile set and half-square* *pleats as the only folded structure outside of the cube, if we add to an empty tile a hinge for every crease of each tile in the tile set for every 2D orthogonal orientation (rotations and reflections), then the resulting tile generates the hinge pattern required to fold a polycube with the Cube Extrusion Algorithm.*

This proposition is nontrivial as it is possible that some combination of cube gadgets would create new tiles that are not present in any single gadget which are thus not in the tile set.

Proof: We prove that no other hinges are needed beyond those found in the constructed generator tile.

There are two types of folded tiles used by cube gadgets to make polycubes as described in the proposition: *inner tiles* which make up the non-visible parts inside the cubes and *pleat tiles* that make up the non-visible part of the pleats (outside of the cube). Inner tiles are never folded once they are made part of a cube as our foldings never modify the inner structure of an existing cube, so they do not require any additional hinges beyond those in the tile set. Pleat tiles may be folded again — but they are all half-square pleats, which means that they simply reflect a crease along the midline of the tile — but each of the reflected halves would already have existed in reflections of that tile of the cube gadget, so this also does not create additional hinges. □

4.2 Paper Dimensions

We now show the specific bounds on the dimensions of the required rectangle of paper for each of the three cube gadgets considered in this paper.

Corollary 4 ($\arctan \frac{1}{2}$ **Universality Lemma)** *Any polycube of N cubes can be folded with all faces seamless from an* $\arctan \frac{1}{2}$ *hinge pattern on a* $(2N+1)\times(2N+2)$ *rectangle of paper using the Cube Extrusion Algorithm.*

Proof: The $\arctan \frac{1}{2}$ gadget has $r = 1$ and $c = 1$. Plugging these constants into Theorem 2 yields a process for folding the polycube from a rectangle of paper of size $(2N + 1) \times (2N + 2)$. □

Corollary 5 (Slit Universality Lemma) *Any polycube of N cubes can be folded with all faces seamless from a slit hinge pattern on a* $(2N + 1) \times (2N + 2)$ *rectangle of paper using the Cube Extrusion Algorithm.*

Proof: Same as Corollary 4. □

The previously discussed gadgets create foldings from a rectangle of paper that is within an additive constant of being square. As we show now, directly applying the Cube Extrusion Algorithm with the tetrakis cube gadget generates a folding with a ratio within a constant of 1×2, but a slight modification allows us to use an approximately square sheet of paper.

Corollary 6 (Box-Pleated Universality Lemma) *Any polycube P of N cubes can be folded with all faces seamless from a box-pleated (tetrakis) hinge pattern on a $(2N+1)\times(4N+2)$ rectangle of paper using the Cube Extrusion Algorithm. A slight modification of the Cube Extrusion Algorithm uses a $(3n+1)\times(3n+2)$ rectangle of paper for even N and a $(3N)\times(3N+3)$ rectangle of paper for odd N.*

Proof: The box-pleated gadget has $r = 1$ and $c = 2$. Plugging these constants into Theorem 2 yields a process for folding P from a rectangle of paper of size $(2N + 1) \times (4N + 2)$.

Now we describe the modified approach. Define the *transpose of a cube gadget* to be the cube gadget with r and c interchanged, so that now we insert r columns to the left and right of the column and c rows below and above the specified row. We alternate the box-pleated gadget and its transpose for a polycube of N cubes such that the box-pleated gadget is applied $\lceil N/2\rceil$ times and its transpose is applied $\lfloor N/2\rfloor$ times. This yields a final folded state F_n with a starting sheet of size $(2r \cdot \lceil \frac{N}{2} \rceil + 2c \cdot \lfloor \frac{N}{2} \rfloor + 1) \times (2c \cdot \lceil \frac{N}{2} \rceil + 2r \cdot \lfloor \frac{N}{2} \rfloor + 2)$ which simplifies slightly to $(2 \cdot \lceil \frac{N}{2} \rceil + 4 \cdot \lfloor \frac{N}{2} \rfloor + 1) \times (4 \cdot \lceil \frac{N}{2} \rceil + 2 \cdot \lfloor \frac{N}{2} \rfloor + 2)$. For even N, this bound is $(3N + 1) \times (3N + 2)$, and for odd N, it is $(3N) \times (3N + 3)$. $\square$

4.3 Number of Layers

Proposition 7 *For any polycube of N cubes, the Cube Extrusion Algorithm produces a folding that uses $O(N^2)$ layers.*

Proof: The folded state produced by Theorem 2 has a starting sheet of size $(2rN + 1) \times (2cN + 2)$, which is clearly $O(N^2)$ for constants r and c. And because each square of a hinge pattern contains $O(1)$ hinges (by definition), there can be at most $O(N^2)$ layers in the folding (even if we folded the paper up into the smallest unit of area allowable by the hinge pattern). $\square$

Unfortunately, this quadratic bound on the number of layers is tight in the worst case:

Proposition 8 *For any N and any cube gadget G, there exists a polycube of N cubes for which the Cube Extrusion Algorithm yields a folding requiring $\Omega(N^2)$ layers.*

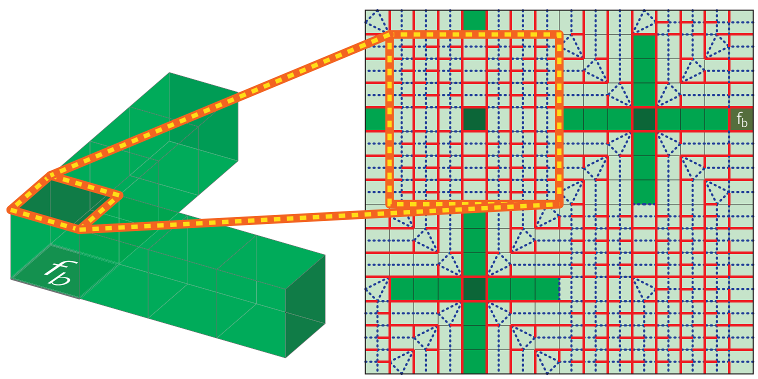

Figure 7: This horizontal L-shaped polycube uses $\Omega(N^2)$ layers when folded by the Cube Extrusion Algorithm.

Proof: Without loss of generality, assume that N is odd. The example we use is a horizontal L-shaped polycube, as shown in Figure 7. To construct it, take a single cube b and two faces which share an edge of the cube. Extrude $(N-1)/2$ cubes from each of the faces. (If N were even, we would extrude $(N/2)-1$ cubes from one of the faces and $N/2$ cubes from the other face).

We now show that our folding algorithm would construct a folding having $\Omega(N^2)$ layers.

Let the bottom cube of the folding algorithm be b, and take the bottom face f_b to be one of the faces parallel the plane spanned by the legs of the L.

Now consider the face in the resulting folded state opposite to f_b: it is seamless, but hidden beneath it are pleats from prisms of both legs of the L. There are $\Omega(N)$ pleats from each leg, and the pleats are orthogonal to each other. This results in $\Omega(N^2)$ layers. □

5 Implementation

A simplification of this algorithm was implemented in Ruby. It allows the user to define a polycube through extrusions and displays the corresponding folding sequence to the screen as a series of angle patterns, but it does not generate a final folded state. It can also show a *simplified composition of the folding sequence* which can be saved to a PostScript file. The simplified composition just shows the original angle pattern for each square tile on the full sheet of paper (that is, the angle pattern from the step when the

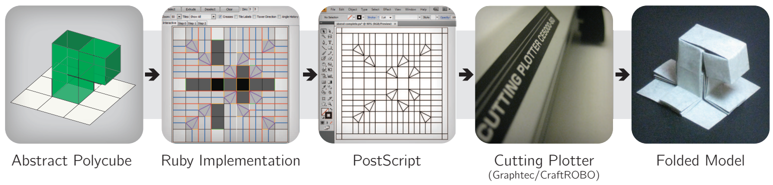

Figure 8: The process by which paper origami can be constructed using the described algorithms.

square was inserted as part of a row or column), and does not take into account any non-orthogonal creases made as a result of later steps.

The PostScript file can be sent to a cutting plotter, which can etch the simplified composition onto a sheet of paper. Alternatively, one can use a laser cutter to score the paper. See Figure 8. Only a single cube gadget is foldable at each step when folding along visible etchings of the simplified composition, so one can simply fold cube gadgets until there are no visible etchings.

If the foldings generated by the Cube Extrusion Algorithm are applied to a larger sheet of paper, it appears that the polycubes are "lying of top of" or "extruded" from the sheet. Artistically this effect is generally preferable to having a polycube closed off by a small flap. In particular, it prevents the pleats from expanding as much, and in some cases enables multiple polycubes to be extruded from a single sheet.

6 Rigid Foldability and Self-Folding Sheets

The Cube Extrusion Algorithm can be used to make artistic paper origami, but to get from one folded state to the next may require curving the paper or introducing temporary nonhinge creases. This becomes an issue for controlling a self-folding programmable matter sheet, where the polygons of a hinge pattern are (nearly) rigid.

It is an open question which polycubes are rigidly foldable from a particular cube gadget, though it seems through simple empirical testing that our cube gadgets fold rigidly in isolation (when making one-cube polycubes). Polycubes with more than one cube may nonetheless not be rigidly foldable, and polycubes with intermediary folded states that self-intersect are almost certainly not rigidly foldable. Our intuition suggests that the slit pattern has the broadest potential for programmable matter.

Acknowledgments

A. Ovadya would like to thank a host of friends who have not complained about his incessant folding of cubes, and in particular Simone Agha, Tucker Chan, Lyla Fischer, Andrea Hawksley, Robert Johnson, and Maria Monks for being sounding boards, folding, and/or commenting. E. Demaine thanks the Huffman family—Elise, Linda, and Marilyn—for access to David A. Huffman's notes, which include the arctan $\frac{1}{2}$ cube gadget. We also thank Jason Ku and Scott Macri for assisting with Figure 3 and Figure 4 respectively.

References

[DDM00] Erik D. Demaine, Martin L. Demaine, and Joseph S. B. Mitchell. Folding flat silhouettes and wrapping polyhedral packages: New results in computational origami. *Computational Geometry: Theory and Applications*, 16(1):3–21, 2000.

[DDMO04] Erik D. Demaine, Satyan L. Devadoss, Joseph S. B. Mitchell, and Joseph O'Rourke. Continuous foldability of polygonal paper. In *Proceedings of the 16th Canadian Conference on Computational Geometry*, pages 64–67, Montréal, Canada, August 2004.

[DO07] Erik D. Demaine and Joseph O'Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge University Press, July 2007.

[DT10] Erik D. Demaine and Tomohiro Tachi. Origamizer: A practical algorithm for folding any polyhedron. Manuscript, 2010.

[HAB+10] E. Hawkes, B. An, N. M. Benbernou, H. Tanaka, S. Kim, E. D. Demaine, D. Rus, and R. J. Wood. Programmable matter by folding. *Proceedings of the National Academy of Sciences of the United States of America*, 2010. To appear. http://www.pnas.org/cgi/doi/10.1073/pnas.0914069107.