Proceedings of the ASME 2023
International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC/CIE2023
August 20–23, 2023, Boston, MA

DETC2023-XXXXXX

2.5D SIGNAGE FROM SHEET MATERIAL WITH ORTHOGONAL CUTS AND FOLDS

Erik D. Demaine[1,†,*], Martin L. Demaine[1,†], Satyan L. Devadoss[2,†], Perla Myers[2,†], Alfonso Parra Rubio[3,†]

[1]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA
[2]Department of Mathematics, University of San Diego, San Diego, CA
[3]Center for Bits and Atoms, Massachusetts Institute of Technology, Cambridge, MA

ABSTRACT

Motivated by making signs with depth from bendable but thick sheet material such as metal or plastic, we analyze 2.5D structures manufacturable from a rectangular sheet by orthogonal cuts and folds (parallel to the sheet's edges). We provide a practical universality algorithm for folding arbitrary pixelated 2.5D surfaces using many parallel cuts, which also supports subpixel features in one dimension. We develop two fonts that enable textual signs within this family of designs, and show real-world constructions from aluminum/polypropylene composite panels. We also study what is possible with just one cut, proving a necessary condition and developing a third font under this restriction.

Keywords: origami, kirigami, fonts

1. INTRODUCTION

Origami engineering with thick material is well-studied, with the typical goal of modifying zero-thickness designs to compensate for material thickness [1–4]. In this paper, we explore new origami/kirigami design algorithms that output simple enough patterns that they can be implemented directly in thick material, with little or no compensation. Specifically, we explore the following design goals:

1. The target shape is a ***2.5D surface***, given by a height (z) function over a rectangle in the xy plane. We require that each horizontal facet is realized by material, and that the folded material does not realize any horizontal facets above the target surface. (The vertical facets can be realized or not by material.) In particular, for 2.5D signage, we have in mind that the height function takes on values of just 0 and 1, with height 0 receding in the background and height 1 forming a foreground "outdented" image — or the reverse for an "indented" image.

2. All folds are ***orthogonal***, meaning the creases are parallel to one of the sides of the rectangle of material, and they are always folded by an integer multiple of 90°. Thus each crease is folded either 90° or 180°, and in either the mountain or valley direction.

3. The number of ***layers*** of material at any facet, and especially at any crease, is small — ideally, 1. In particular, we aim for creases to never meet at vertices.

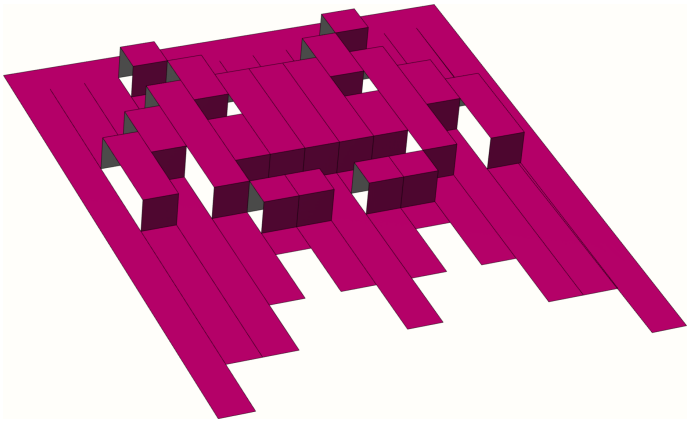
4. To make the above constraints feasible, we allow ***cuts*** or slits in the material. Like creases, we require all cuts to be orthogonal. In addition, for strength, we aim to use relatively few and short cuts.

1.1 Practical Universality

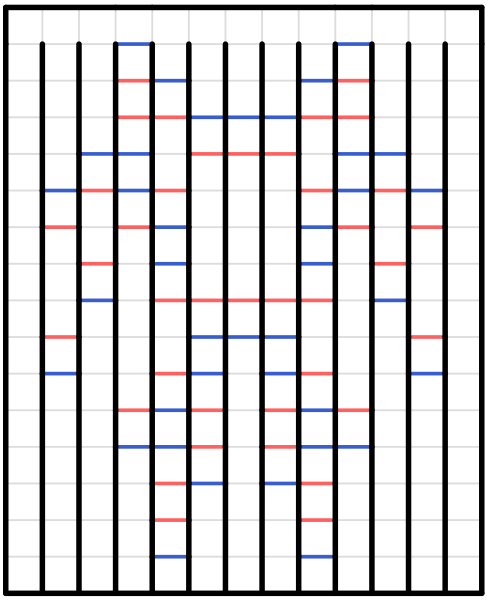
Our main algorithmic result (in Section 2) establishes ***universality***: an arbitrary pixelated 2.5D surface can be realized under the above constraints, with just one layer of material everywhere. The key idea is to make parallel cuts in the shorter direction, leaving strips of material to freely fold along the surface. Figure 1 shows an example. In fact, this method supports subpixel features in one dimension, parallel to the cuts: each strip can fold up and down at arbitrary positions, not just at pixel boundaries. While the algorithm is straightforward, the advantage of this approach is that the designs are very simple, enabling practical construction out of thick material. Figure 4 in Section 2.2 shows real-world examples of signs made from aluminum/polypropylene composite panels using this technique, proving its practicality.

By contrast, past work on folding arbitrary or orthogonal surfaces [5–7] involves more complex, nonorthogonal crease patterns with multiple overlapping layers of material. The major difference is that our solution also involves cuts. One somewhat closer result to ours is that orthogonal surfaces can be made by folding just along orthogonal creases [8], but this solution uses non-rectangular paper and stacks up to four overlapping layers.

[†]Joint first authors
[*]Corresponding author: edemaine@mit.edu



(a) Folded geometry.



(b) Crease–cut pattern. Cuts in black, 90° mountain creases in red, 90° valley creases in blue.

FIGURE 1: OUR 2.5D UNIVERSAL ALGORITHM APPLIED TO A CHARACTER FROM TOMOHIRO NISHIKADO'S *SPACE INVADERS*, RELEASED BY TAITO IN 1978.

To support practical sign making, we develop two fonts that enable multiword messages to be constructed as 2.5D surfaces using our universal algorithm. The first is a 3×5 pixel font, where letters are aligned to an integer grid. The minimal dimensions of this font mean that just *two* cuts in a rectangle suffice to make an individual letter (separating the three columns of pixels). The second font also uses 3×5 letters, but it exploits the subpixel ability of our algorithm, using half-pixels in the vertical dimension to improve readability. Figures 2 and 3 in Section 2.1 show these two font designs and the resulting crease–cut patterns. The pixel font is the basis for the real-world constructions in Figure 4. We also describe strategies for writing multiple rows of text (Section 2.3), hiding information with puzzle fonts (Section 2.4), and practical changeable signs with just 16 universal pieces for writing all letters and digits (Section 2.5). The reader can experiment with these fonts using an interactive web app.[1]

Even within the design constraints of orthogonal creases and cuts, where creases never meet at vertices, sequencing the folds to avoid collisions is known to be strongly NP-complete [9, 10]. The designs output by our universality algorithm, however, have a straightforward sequence of operations.

1.2 Few Cuts

A disadvantage of our universal design algorithm is that it makes nearby parallel cuts, reducing the strength of the construction. For signage with text or other narrow imagery, the cuts are relatively short, so the designs are quite practical. But this issue motivates the study of how few cuts we can use to realize a 2.5D surface.

As an extreme, we study the restriction to at most one orthogonal cut. We provide a full characterization of what 2.5D surfaces are possible with zero cuts: essentially, the surface must be "1.5D" because all 90° folds must be parallel. We then use this to provide a partial characterization of what 2.5D surfaces are possible with a single cut by proving a necessary condition: essentially, any realizable surface must decompose into two 1.5D surfaces. In particular, a 2×2 array of height-1 pixels extruded from a sea of height 0 is impossible to make with just one cut.

On the positive side, we design a font where each letter can be realized with just a single cut. This font design is more involved, requiring the stacking of a few layers of material and folding through up to two layers, but likely achieves higher strength. Figure 10 in Section 3.3 shows the font design, and Figure 11 shows a real-world construction made from thick paper.

2. PRACTICAL UNIVERSALITY VIA PARALLEL CUTS

Define a ***2.5D surface*** to be a height function $h(x, y) \in \mathbb{R}$ defined over a rectangle $(x, y) \in [0, X] \times [0, Y]$. We treat h as being zero outside this rectangle. Call a 2.5D surface ***x-integral*** if it is constant over each integer unit horizontal segment $(i, i + 1) \times \{y\}$, where i is an integer. This condition is weaker than being ***pixelated***, i.e., constant over each integer unit square $(i, i + 1) \times (j, j + 1)$, where i and j are integers. Call a 2.5D surface ***y-varying-by-V*** if it is piecewise constant in any column $\{x\} \times \mathbb{R}$, and the sum of the absolute changes in height between adjacent constant portions is at most V. Note that this definition includes the absolute change between the surrounding field of height 0 and the heights of h defined at the extreme y coordinates 0 and Y.

Theorem 1. *For any x-integral y-varying-by-V 2.5D surface $h(x, y)$ defined over a rectangle $(x, y) \in [0, X] \times [0, Y]$, there is an orthogonal cutting and folding of a rectangle of material $[0, X] \times [0, Y + V + \varepsilon]$ that realizes all horizontal facets of h (those parallel to xy), some vertical facets of h (those parallel to xz), and some more horizontal facets outside the domain of h at height 0, for any $\varepsilon > 0$.*

Proof. Refer to Figure 1, which sets $\varepsilon = 1$. For connectivity, we reserve a row of thickness ε at the top of the material, which will be a small horizontal facet at height $z = 0$ above the top y

[1] https://erikdemaine.org/fonts/cutfold/

coordinate of the domain of h. Then we cut the material into X vertical columns of width 1, cutting at each integer x coordinate between 1 and $X-1$ from just below the top reserved row down to the bottom of the material. We then crease each column as follows: when the height remains constant for a y distance of d, we leave an uncreased portion of the column for a y distance of d; and whenever the height changes by some amount Δ, we add a pair of folds separated vertically by $|\Delta|$. The folds are 90° valley then 90° mountain for $\Delta > 0$, and 90° mountain then 90° valley for $\Delta < 0$. When a column has less variation than V, there will be a horizontal facet extending below the bottom y coordinate of the domain of h. □

These universal designs are easy to execute in a sequence of folds that avoids collisions: for each column in any order, fold each column's creases in order from one end to the other.

One extension to the universality result is that we can make a ***two-sided*** sign, defined by one nonnegative 2.5D surface (the top) and one nonpositive 2.5D surface (the bottom). When each column of material reaches the bottom of the top surface, we can mountain fold it by 180° to go underneath and construct the bottom surface. If both the top and bottom surfaces have faces at height 0, then we will have two layers of material at such faces; otherwise, we will maintain a single layer everywhere.

2.1 Fonts

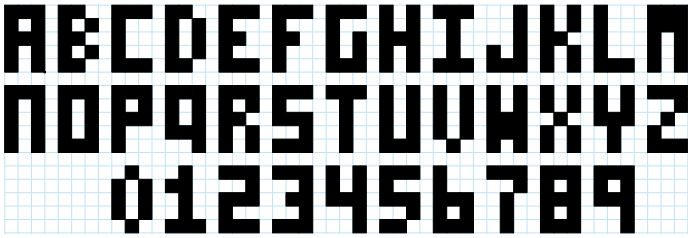
We design two fonts to construct textual messages in 2.5D signs via the universality result of Theorem 1. In both fonts, each letter and digit is 3 units wide and 5 units tall (3×5). The first, the ***pixel font*** in Figure 2a, is *pixelated* as defined above. The second, the ***subpixel font*** in Figure 3a, is merely *x-integral* (as defined above), exploiting the ability to vary the height at half-integer y coordinates.

Each glyph can be interpreted as a 2.5D surface in two different ways. In the ***outdented*** form, the glyph is at height 1 while the background is at height 0. In the ***indented*** form, the glyph is at height 0 while the background is 1 unit high. We can further vary both forms by changing the nonzero height from 1 to any positive real number. Figures 2b and 3b show the resulting crease–cut patterns for the height-1 outdented forms of the pixel and subpixel fonts respectively. (The indented forms just have mountains and valleys flipped.) These patterns were generated automatically by an implementation of the universality algorithm in CoffeeScript.

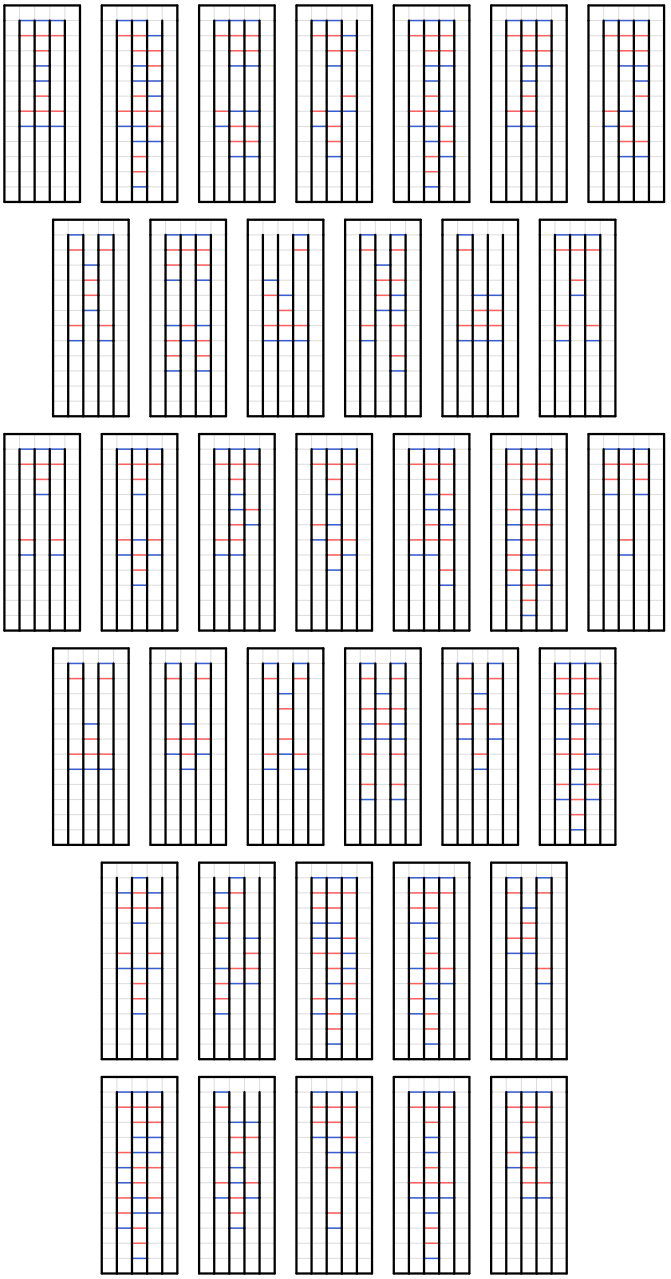
We chose a width of 3 because it means that constructing each glyph by itself requires only two cuts, which is quite small. (Section 3 considers the minimum possibility of one cut.) Of course, when we combine multiple letters together to form a word or row of words, we end up with more cuts, namely 4 per letter as shown in Figures 2b and 3b.

2.2 Experiments

Figure 4 shows the physical results of two experiments constructing real-world signs with the word "FOLD" using the pixel font, in both the outdented and indented forms. (The experiment was done before we designed the subpixel font.)

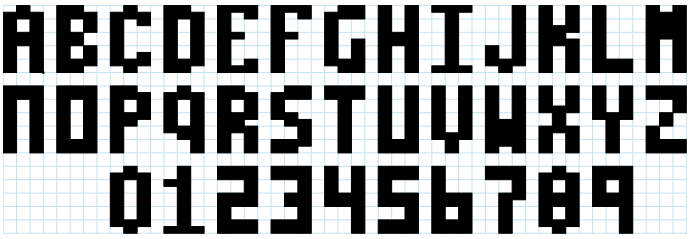


(a) 3 × 5 pixel font design.

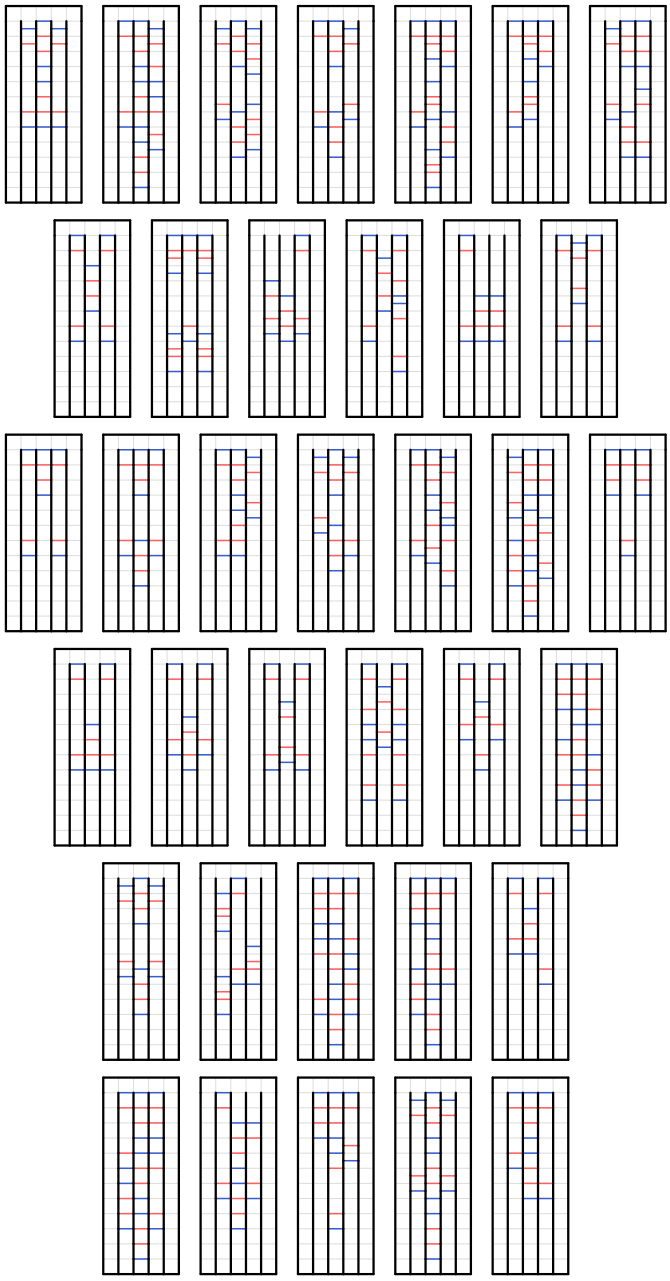


(b) Crease–cut patterns for outdented form, A–Z and 0–9.

FIGURE 2: 3 × 5 PIXEL FONT DESIGN AND HOW TO FOLD IT.

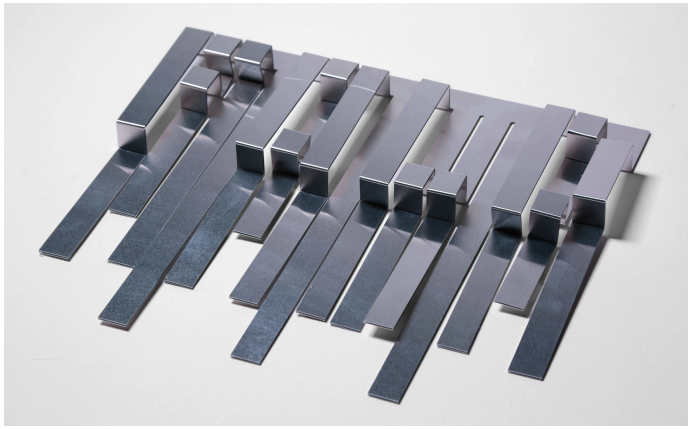


(a) 3×5 subpixel font design, which uses half-squares in the vertical dimension.

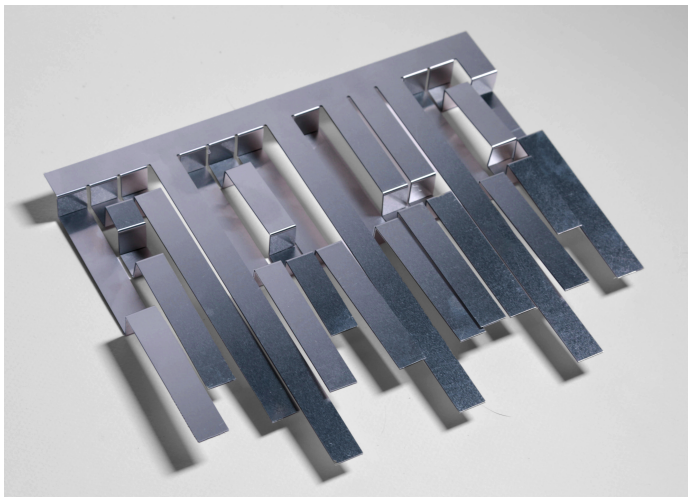


(b) Crease–cut patterns for outdented form, A–Z and 0–9.

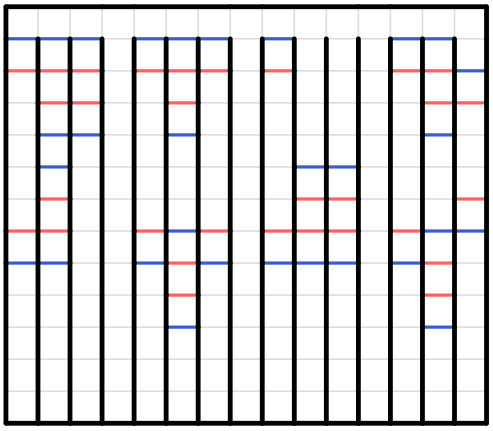
FIGURE 3: 3 × 5 SUBPIXEL FONT DESIGN AND HOW TO FOLD IT.



(a) Outdented.



(b) Indented.



(c) Crease–cut pattern for outdented form.

FIGURE 4: REAL-WORLD CONSTRUCTIONS OF "FOLD" USING OUR UNIVERSAL ALGORITHM AND 3 × 5 PIXEL FONT, IN OUTDENTED AND INDENTED FORMS. THE MATERIAL IS A 1.2MM 3-LAYER COMPOSITE PANEL CONSISTING OF A 0.8MM POLYPROPYLENE CORE SANDWICHED BETWEEN TWO LAYERS OF 0.2MM ALUMINUM (SOLD AS *HYLITE* BY 3A COMPOSITES).

The material is *Hylite* from 3A Composites, a 1.2mm 3-layer composite panel consisting of a 0.8mm polypropylene core sandwiched between two layers of 0.2mm aluminum. We milled the material using a CNC machine (Zund G-3 L-2500) with router module (RM-A) equipped with a 2mm end mill (R502). More specifically, we double face milled the crease pattern, cutting the valleys on top, flipping the material, and then cutting the mountains. (Proper alignment between the two passes is particularly easy on the Zund by cutting registration holes in the first pass of cutting valleys, and orienting relative to these holes in the second pass of cutting reflected mountains.) We engraved the creases to a depth of 0.6mm, which cuts through the front 0.2mm aluminum layer and half of the 0.8mm polypropylene core. The polypropylene core then easily bends into the crease, and the back aluminum layer deforms plastically to hold the shape, with minimal bending strain.

Visually, we find the outdented form to be quite legible, while the indented form is more difficult to read. Finishing, painting, or otherwise distinguishing the raised facets of the sign may improve legibility.

2.3 Multiple Rows of Text

To stack multiple rows of text, we can follow two approaches.

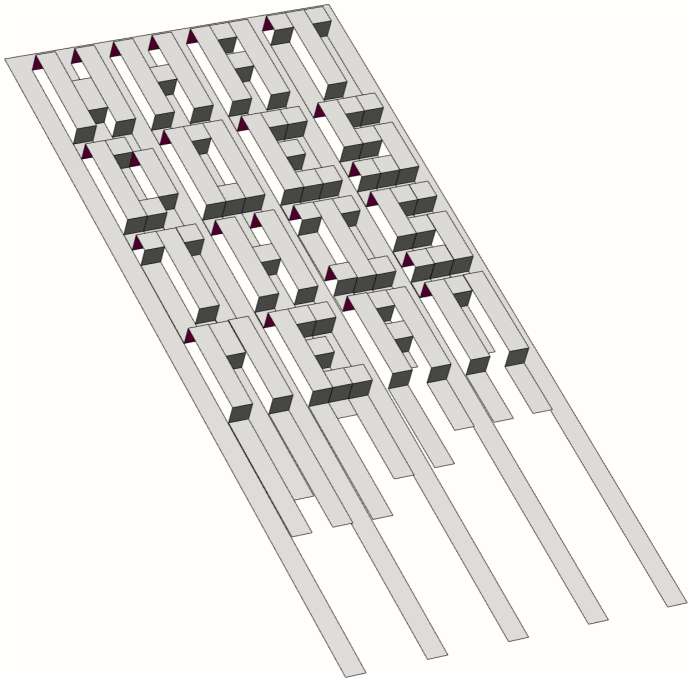
In the first approach, we can apply Theorem 1 directly to the entire 2.5D surface formed by the grid of letters. Figure 5 shows an example (generated by our CoffeeScript implementation). This solution shares cuts between the rows, and may reduce the total material required because multiple worst-case columns (with three raised/lowered bumps, as in the middle columns of "E", "S", and "3") are unlikely to be aligned on a vertical line. But the construction is potentially more fragile: each column of material acts as a cantilever beam, so making it longer might reduce stability. On the other hand, this instability could likely be fixed by fusing the folded material to a backing sheet or frame, at all or some of the facets where the folding returns to height 0.

In the second approach, we can apply Theorem 1 separately to each row, and add a horizontal cut between rows, keeping rows connected together via a leftmost and/or rightmost column of material. Figure 6 shows an example. This solution uses more cuts and potentially more material (always matching the worst-case pattern heights of Figures 2b and 3b), but each vertical cut is shorter (the same as a single row of text), making for shorter cantilever beams which should be more stable in practice.

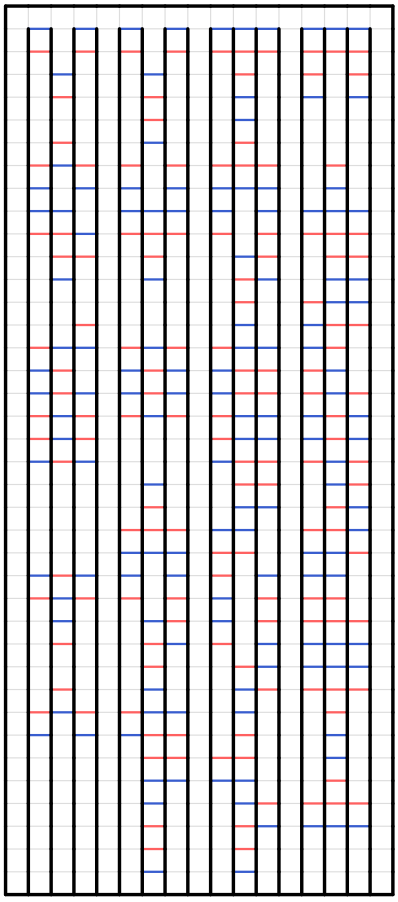
2.4 Puzzle Fonts

As a fun way to communicate messages without making them plainly readable, the crease–cut patterns of Figures 2b and 3b serve as ***puzzle fonts*** [11–13]. These puzzle fonts follow a pattern: the same character is always represented by the same crease–cut pattern (unless we switch between the pixel and subpixel fonts, but then each character is still represented by just two patterns).

We can break this pattern using the first approach to multiple rows of text described in Section 2.3, as in Figure 5b. Now the columns forming each letter have varying vertical shift from column to column, depending on the number of bumps in the text above, removing the obvious visual pattern of repetition (after the first row of text). Furthermore, letters from the same row of

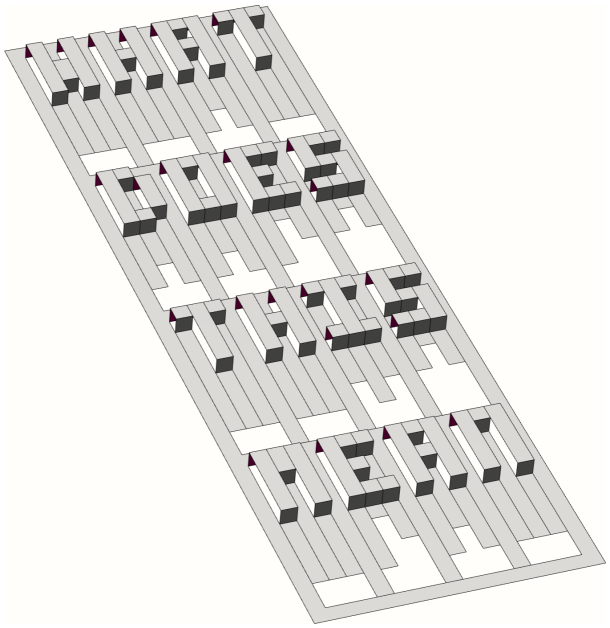


(a) Folded geometry.

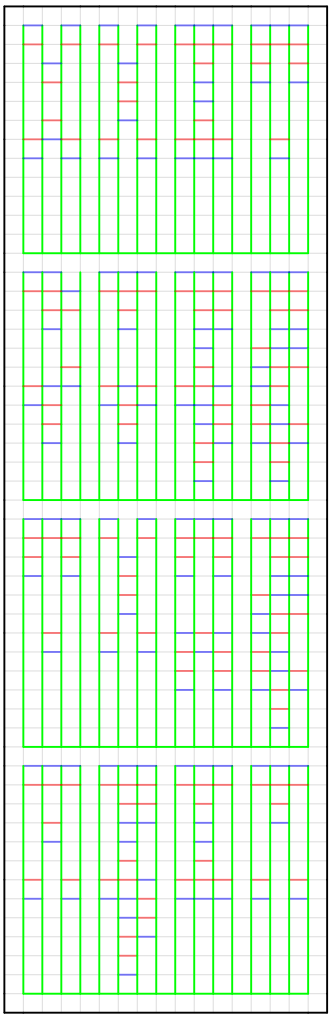


(b) Crease–cut pattern.

FIGURE 5: APPLYING UNIVERSALITY THEOREM 1 DIRECTLY TO MULTIPLE ROWS OF TEXT.



(a) Folded geometry.



(b) Crease–cut pattern.

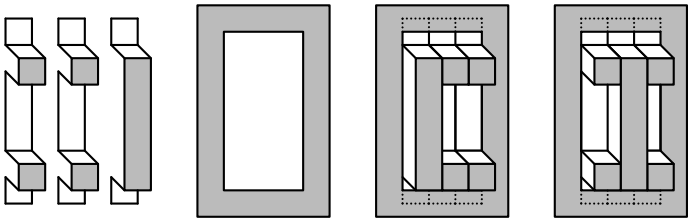
FIGURE 6: APPLYING UNIVERSALITY THEOREM 1 TO EACH ROW OF TEXT INDIVIDUALLY.

text are no longer horizontally aligned, making it difficult even to separate into rows.

2.5 Changeable Signs

Changeable signs (as frequently used in marquee signs above movie theaters and beside churches) are typically made with one physical part per glyph. One problem with this approach is that, with 26 different letters in the alphabet (and 10 digits), it is easy to run out of a particular part.

The pixel font offers a different approach: decompose each glyph into three parts, one per column. Each part is a folded strip of material, making up the five pixels of one column, plus an extra square at either end for sliding into a backing frame. Figure 7 shows a simple example of how such parts can fit together into a frame while supporting reconfiguration between different letters. For multiple letters, the frame can include a separate hole for each letter, or use one long hole for each row.



(a) Parts. (b) Frame. (c) “C” form. (d) “I” form.

FIGURE 7: CHANGEABLE SIGNS WITH ONE PART PER COLUMN INSTEAD OF ONE PART PER LETTER.

Combinatorially, there are $2^5 = 32$ different columns of five pixels. But the pixel font of Figure 2a uses only 21 distinct columns, which is already a savings over the 26 (or 36) if we used a separate part for each glyph. Furthermore, because each column can be turned upside-down, many of these parts are effectively identical. Taking into account this symmetry, the pixel font uses only 16 distinct columns. (In these counts, we ignore the blank column used to separate glyphs, which we assume is part of the backing frame instead of an actual part.) The number of distinct parts is thus far fewer than the number of glyphs.[2]

Table 1 shows the distribution of part usage: the number of times each part is used, and by which letters (in some cases, multiple times in one letter). One part, where every pixel is filled, gets used by every letter except five (“S”, “V”, “X”, “Y”, and “Z”), so it clearly should be very highly stocked. At the other extreme, parts with unique usage point to potential font tweaks that can reduce the number of distinct parts. For example, by modifying the rightmost column of “B” (the only instance of the pattern ▫■■▫) to match the rightmost column of “D” (▫■■■▫), we reduce the number of distinct parts to 15.

The last column of Table 1 shows part frequencies when we weight each letter by its relative usage in English, as measured by Samuel Morse (when inventing Morse code) by counting letters in sets of printers’ type (a kind of changeable sign) [14]. This expected distribution is roughly what we should follow when

[2]The same analysis with the subpixel font of Figure 3a is less impressive, requiring 30 distinct parts.

Pattern	Letter usage	Count	Percent	Morse
	B	1	1.28%	0.50%
	D	1	1.28%	1.38%
	W	1	1.28%	0.63%
	Q	1	1.28%	0.16%
	JM	2	2.56%	1.07%
	ZZ	2	2.56%	0.13%
	GSS	3	3.84%	5.55%
	HKX	3	3.84%	2.38%
	QVV	3	3.84%	0.91%
	BESZ	4	5.13%	6.83%
	KRXX	4	5.13%	2.44%
	PYYY	4	5.13%	2.41%
	AFPR	4	5.13%	5.76%
	CCDEGIIO	8	10.26%	15.07%
	FJLLNTTUV	9	11.54%	13.00%
	AABCDEFGHHIJKLM MNNOOPQRTUUWW	28	35.90%	41.79%

TABLE 1: USAGE OF 1 × 5 PARTS IN THE PIXEL FONT OF FIGURE 2a, UP TO REFLECTION, SORTED BY USAGE. COLUMNS IN ORDER: PART PATTERN ORIENTED TO BE LEXICALLY MINIMUM; LETTERS (WITH MULTIPLICITY) USING THE PART; NUMBER OF SUCH LETTER USES; PERCENT OF TOTAL USES OF PARTS IN LETTERS ($26 \cdot 3 = 78$); AND PERCENT OF USES WEIGHTED BY MORSE'S LETTER FREQUENCY TABLE [14].

manufacturing a universal set of parts, but to account for variance in the distribution of English, we should also stock extras of the infrequent parts.

Our approach to changeable signs inspires another potential puzzle font. Each letter of the 3×5 pixel and subpixel fonts of Figures 2a and 3a uses a unique set of three columns, except for C and I from the pixel font as illustrated in Figure 7. Thus every letter could have their parts randomly permuted, and the intended letters could still be reconstructed by a patient reader.

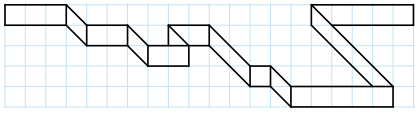
3. THEORY OF A SINGLE CUT

While the universality of Theorem 1 is powerful, the solution makes many cuts. In this section, we study the other extreme: what is possible with at most one cut? We fully characterize what is possible with zero cuts (Section 3.1), and partially characterize what is possible with one cut via a strong necessary condition (Section 3.2). Despite this limitation, we design a font using only one cut per letter (Section 3.3).

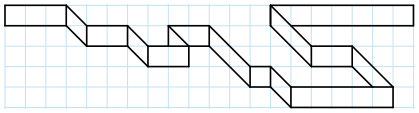
3.1 Zero Cuts

First we characterize what 2.5D surfaces are possible with zero cuts and orthogonal folding; refer to Figure 8a. Define a ***1.5D surface*** to be a 2.5D surface where nonzero height is confined to an orthogonal band in the xy plane (infinite in x or y but not both), and the height function is constant in any orthogonal segment across the finite dimension of the band. In particular, all vertical faces are parallel to each other (either all xz or all yz).

Lemma 2. *For any orthogonal folding of a rectangle of material realizing the horizontal surfaces of a 2.5D surface (and no higher horizontal surfaces), the surface must in fact be a 1.5D surface.*



(a) 1.5D surface (defined by an x-infinite band).



(b) Possible folding.

FIGURE 8: REALIZING 1.5D SURFACES BY ORTHOGONAL FOLDING AND NO CUTTING.

Proof. We can view all 180° folds as being applied to the material first. Because there are no cuts and the paper is rectangular, such folds effectively just shrink the material [9], so can be ignored. Then the 90° folds cannot cross each other, so they must all be parallel. Unlike 1.5D surfaces, such foldings may have overhang. For example, the folding in Figure 8b realizes the 1.5D surface in Figure 8a according to our definitions, because the additional horizontal facets are no higher than the target surface. Taking the upper envelope of horizontal facets from the folding (keeping just the subfacets that are not below other horizontal facets) gives us a 1.5D surface. □

3.2 Limits of One Cut

Building on our characterization of 1.5D surfaces from zero cuts, we next prove a necessary condition on what 2.5D surfaces can be made with a single cut (restricting to orthogonal patterns):

Theorem 3. *Every orthogonal cutting and folding of a rectangle of material with a single partial cut (which does not disconnect the material) realizes the upper envelope of two 1.5D surfaces. Furthermore, the two bands defining the surfaces must intersect in the xy plane.*

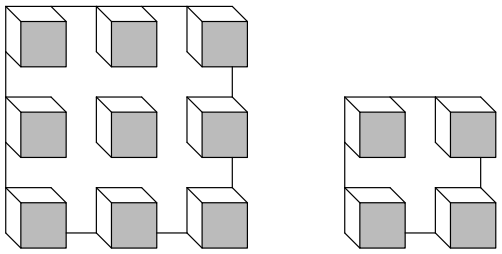
Proof. Assume by symmetry that the cut is parallel to the x axis. Extending this cut to a full x-infinite line ℓ divides the rectangle R of material into two uncut rectangles, R_1 and R_2. Restricting the folding to each of these rectangles, we must have a 1.5D surface by Lemma 2. By our definitions, the realized 2.5D surface ignores lower facets, so it is the claimed upper envelope.

To see that the strips defining the bands must intersect, we need to characterize the 180° folds which might allow R_1 and R_2 to shrink before folding. The key property is that such folds cannot move the boundary edge of R_1 that is shared by R_2 (or vice versa), because the material remains partially connected there (as the cut is partial). Let e denote that connection: the uncut material along the line ℓ. After the 180° folds, the folded rectangles R_1' and R_2' still share an edge, the folded form e' of e. No matter how these rectangles fold into 1.5D surfaces via 90° folds, the underlying strips will overlap at the xy projection of e'. □

Note that the two 1.5D surfaces can be defined by two x-infinite strips, two y-infinite strips, or one x-infinite and one y-infinite strip, and this three-way choice is independent of whether

the cut is parallel to the x or y axis. These choices leave a fair amount of flexibility, which we will exploit in our font design.

Nonetheless, Theorem 3 implies that most 2.5D surfaces cannot be realized with a single cut, when restricting to orthogonal patterns, as detailed in the corollaries below. In particular, these results contrast our universality result with multiple cuts (Theorem 1). More interesting is the contrast with standard results in computational origami that every 3D surface can be made without any cuts [5, 6], but these solutions use non-orthogonal creases. Another contrasting result is that orthogonal surfaces can be made using orthogonal creases [8], but this solution uses non-rectangular paper.



(a) 3×3. (b) 2×2.

FIGURE 9: 2.5D SURFACES THAT CANNOT BE MADE BY ONE ORTHOGONAL CUT AND ORTHOGONAL FOLDS. HORIZONTAL FACES ABOVE HEIGHT 0 ARE SHADED.

Corollary 4. *No orthogonal cutting and folding of a rectangle of material with a single cut can realize the horizontal surfaces (and no higher horizontal surfaces) of the 2.5D surfaces in Figure 9.*

Proof. Figure 9a (and various subsets of this 3×3 grid of bumps, such as the quincunx pattern on the 5 side of a die) cannot be decomposed into two 1.5D surfaces. Essentially, the gaps between rows/columns of bumps force any strip to end there, which confines each strip to either a single row or column, necessitating three strips to cover all bumps.

Figure 9b is more interesting because it *can* be decomposed into two 1.5D surfaces. However, the underlying strips must then form two rows or two columns (one row and one column would leave one bump uncovered), so again there must be a separation between the two strips. This contradicts that the strips must intersect, as guaranteed by Theorem 3. □

Corollary 5. *The probability that a pixelated 2.5D surface over an $n \times n$ square, where each pixel height is selected uniformly at random from $\{0, 1\}$, can be realized by an orthogonal cutting and folding of a rectangle of material with a single cut is $1/2^{n^2-O(n)}$.*

Proof. The number of different $n \times n$ pixel patterns is 2^{n^2}. We argue that very few of them are possible according to Theorem 3. Specifically, the number of possible infinite strips is $O(n^2)$, and the number of possible 1.5D surfaces each such strip can make is 2^n. Thus the number of possible pixel patterns that can be made as the upper envelope of two 1.5D surfaces is $O(n^2 2^n)^2 = O(n^4 4^n)$. The probability is thus $O(n^4 4^n / 2^{n^2}) = 1/2^{n^2-O(n)}$. □

3.3 One-Cut Font

Figure 10 shows a 2.5D font where each letter is made by one orthogonal cut and orthogonal folding. These designs illustrate the surprising versatility of a single cut. The visible seams in Figure 10a help verify that every 2.5D surface can indeed be decomposed into two 1.5D surfaces, with a mixture of x-infinite and/or y-infinite strips.

These designs may also be useful for practical sign making, as they involve less cutting than the two cuts per letter from the fonts in Figures 2 and 3. We also suspect that they lead to much stronger structures, as the various layers help support each other. However, the one-cut designs are more complicated to fold from thick material: multiple layers overlap at some facets, and some folds cross, requiring folds through multiple layers.

3.4 Experiments

Figure 11 shows the results from a simple experiment of folding thick paper into the letters "O", "N", and "E". We printed the crease–cut pattern onto the paper, then cut and folded along the lines by hand. We expect that a similar experiment with metal would require modifying the folds slightly to compensate for material thickness.

4. FUTURE WORK

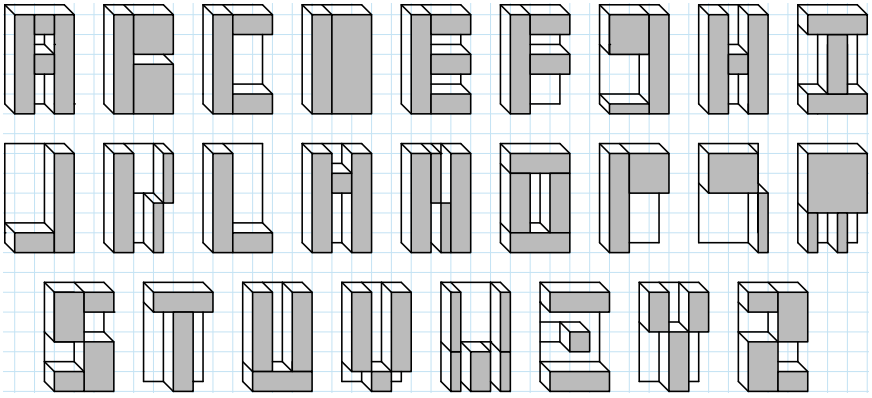
On the mathematical side, the main open problem is to characterize what 2.5D surfaces can be realized by orthogonal folding and cutting with just one cut. Our necessary condition may not be sufficient in particular because of possible collisions between the two 1.5D surfaces.

On the engineering side, we plan to explore slumping plate glass to create glass signage. We believe the universality algorithm and fonts of Section 2 should be quite practical in this context. By building a mold of the desired shape (e.g., by stacking ceramic bricks), a properly heated strip of plate glass should fall into the desired shape. Multiple strips needed for one or more letters could be made and mounted individually, as in Section 2.5, or made from a single rectangle of plate glass that is waterjet with the appropriate cuts.

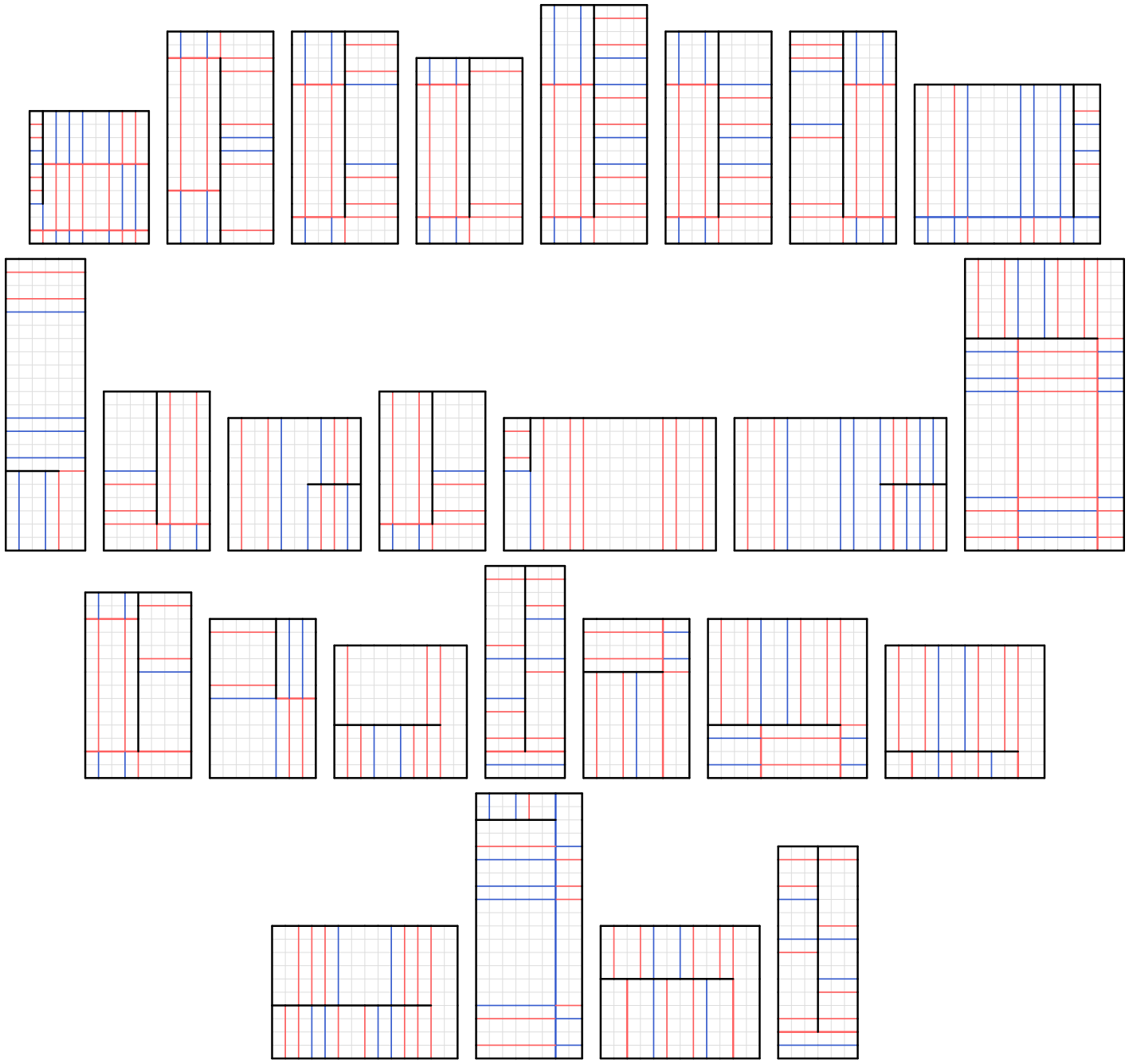
ACKNOWLEDGMENTS

This work was initiated while E. Demaine and M. Demaine were Knapp Chairs of Liberal Arts at University of San Diego in April 2018, collaborating with many Knapp Scholars: Nicholas Bail, Parth Bansal, Eleanor Cecilia Barnhill, MarGhece Barnes, Vayunamu Bawa, Laura Becerra, Melissa Bridges, Hans Broders, Odesma Dalrymple, Kristin Desplinter, Emma Dickson, Monique Farha, Valeriya Fox, Gabriella Goerke, Evan Grattendick, Heather Hodlin, Anelise Hunziker, Tanya Keval, Elizabeth Kresock, Daniel Ley, Daniel Lopez-Perez, Jordan Matuszewski, Madylin Miller, Daniel Myers, Julia Norman, John Pendas, James Ponwith, Jordan Readyhough, Jasmyn Sosa-Houston, Taylor Wong, Kenneth Yee, and Qineng Zeng. We thank these students for inspiration and initial collaboration.

We thank the MIT Center for Bits and Atoms for the use of their Zund CNC machine.



(a) Folded geometry.



(b) Crease–cut patterns, A–Z. Cuts in black, mountain creases in red, valley creases in blue. 180° folds are thicker than 90° folds.

FIGURE 10: FONT DESIGN WITH ONE CUT PER LETTER.

FIGURE 11: REAL-WORLD CONSTRUCTION OF "ONE" USING OUR SINGLE-CUT FONT FROM FIGURE 10. THE MATERIAL IS 98 LB / 160 GSM WATERCOLOR PAPER (SOLD AS *MI-TEINTES* BY CANSON).

REFERENCES

[1] Tachi, Tomohiro. "Rigid-Foldable Thick Origami." *Origami^5: Proceedings of the 5th International Conference on Origami in Science, Mathematics and Education*. A K Peters (2010): pp. 253–264.

[2] Hoberman, Charles. "Folding Structures Made of Thick Hinged Sheets." U.S. Patent No. 7,794,019 (2010).

[3] Chen, Yan, Peng, Rui and You, Zhong. "Origami of thick panels." *Science* Vol. 349 No. 6246 (2015): pp. 396–400. DOI 10.1126/science.aab2870.

[4] Ku, Jason S. and Demaine, Erik D. "Folding Flat Crease Patterns With Thick Materials." *Journal of Mechanisms and Robotics* Vol. 8 No. 3 (2016): pp. 031003–1–6. DOI 10.1126/science.aab2870.

[5] Demaine, Erik D., Demaine, Martin L. and Mitchell, Joseph S. B. "Folding Flat Silhouettes and Wrapping Polyhedral Packages: New Results in Computational Origami." *Computational Geometry: Theory and Applications* Vol. 16 No. 1 (2000): pp. 3–21. DOI 10.1016/S0925-7721(99)00056-5.

[6] Demaine, Erik D. and Tachi, Tomohiro. "Origamizer: A Practical Algorithm for Folding Any Polyhedron." *Proceedings of the 33rd International Symposium on Computational Geometry*: pp. 34:1–34:15. 2017. Brisbane, Australia. DOI 10.4230/LIPIcs.SoCG.2017.34.

[7] Biswas, Amartya Shankha, Demaine, Erik D. and Ku, Jason S. "Efficient Origami Construction of Orthogonal Terrains using Cross-Section Evolution." *Origami^7: Proceedings of the 7th International Meeting on Origami in Science, Mathematics and Education*. Vol. 2. Tarquin, Oxford, England (2018): pp. 631–646.

[8] Benbernou, Nadia M., Demaine, Erik D., Demaine, Martin L. and Lubiw, Anna. "Universal Hinge Patterns for Folding Strips Efficiently into Any Grid Polyhedron." *Computational Geometry: Theory and Applications* Vol. 89 (2020): p. 101633. DOI 10.1016/j.comgeo.2020.101633.

[9] Arkin, Esther M., Bender, Michael A., Demaine, Erik D., Demaine, Martin L., Mitchell, Joseph S. B., Sethia, Saurabh and Skiena, Steven S. "When Can You Fold a Map?" *Computational Geometry: Theory and Applications* Vol. 29 No. 1 (2004): pp. 23–46. DOI 10.1016/j.comgeo.2004.03.012.

[10] Akitaya, Hugo A., Demaine, Erik D. and Ku, Jason S. "Simple Folding is Really Hard." *Journal of Information Processing* Vol. 25 (2017): pp. 580–589. DOI 10.2197/ipsjjip.25.580.

[11] Demaine, Erik D. and Demaine, Martin L. "Fun with Fonts: Algorithmic Typography." *Theoretical Computer Science* Vol. 586 (2015): pp. 111–119. DOI 10.1016/j.tcs.2015.01.054.

[12] Demaine, Erik D. and Demaine, Martin L. "More than Words: Fonts as Generative Art." *Proceedings of the 24th Generative Art Conference (GA 2021)*: pp. 369–378. 2021. Sardinia, Italy.

[13] Demaine, Erik D. and Demaine, Martin L. "Puzzle Fonts About Puzzles." *Exchange Book of the 14th Gathering for Gardner (G4G14)*. Atlanta, Georgia (2022).

[14] Keating, Barry. "The frequency of the letters of the alphabet in English." https://www3.nd.edu/~busiforc/handouts/cryptography/letterfrequencies.html. Lecture notes from Business Forecasting class, University of Notre Dame.