Sketch-Based Design of Foundation Paper Pieceable Quilts

Mackenzie Leake
MIT CSAIL
Cambridge, Massachusetts, USA
leake@mit.edu

Gilbert Bernstein
UC Berkeley
Berkeley, California, USA
gilbo@berkeley.edu

Maneesh Agrawala
Stanford University
Stanford, California, USA
maneesh@cs.stanford.edu

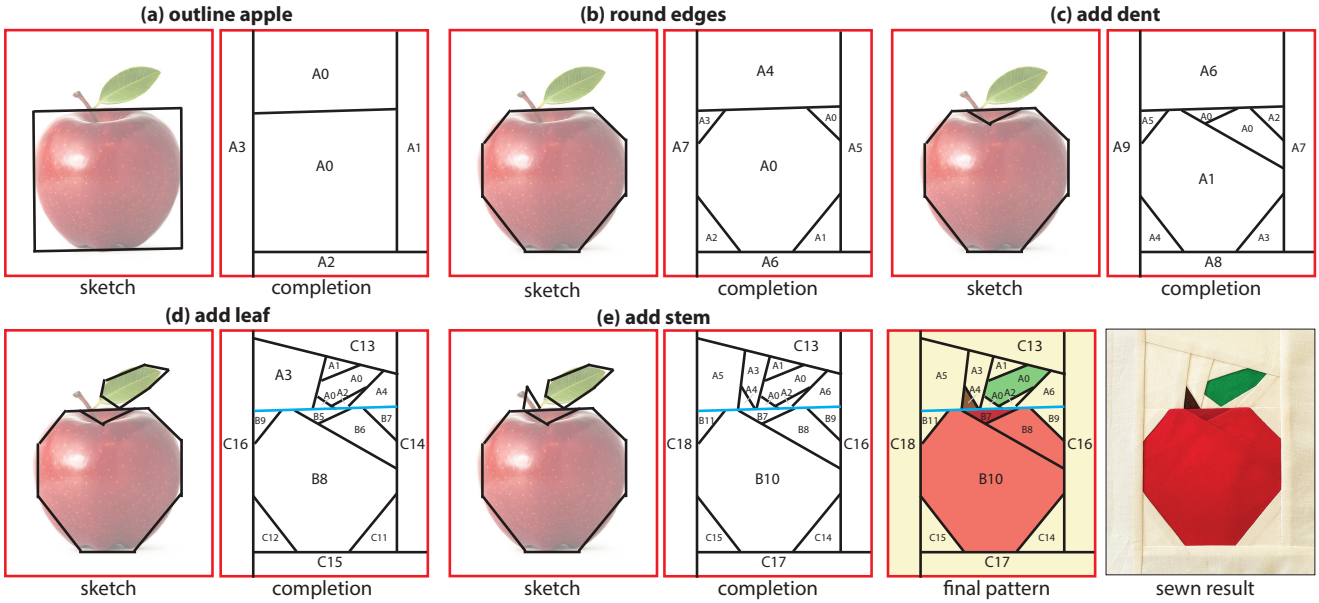


Figure 1: Our quilt design tool fits the iterative workflow of quilt designers. A designer sketches a partial design as a set of edges representing outlines and details of the foreground shapes (a-e left images). Our completion algorithm converts the sketch into a foundation paper pieceable pattern (a-e right images), extending the sketched edges and breaking the design into independently paper pieceable *sections* as necessary. A blue section edge for this design is marked in iterations (d) and (e).

ABSTRACT

Foundation paper piecing is a widely used quilt-making technique in which fabric pieces are sewn onto a paper guide to facilitate construction. But, designing paper pieceable quilt patterns is challenging because the sewing process imposes constraints on both the geometry and sewing order of the fabric pieces. Based on a formative study with expert quilt designers, we develop a novel sketch-based tool for designing such quilt patterns. Our tool lets designers sketch a partial design as a set of edges, which may intersect but do not have to form closed polygons, and our tool automatically completes it into a fully paper pieceable pattern. We contribute a new sketch-completion algorithm that extends the input sketched edges into a planar mesh composed of closed polygonal faces representing fabric pieces, determines a paper pieceable sewing order for the faces, and breaks complicated sketches into independently paper pieceable *sections* when necessary. A partial input design often admits multiple visually different completions. Thus, our tool lets designers specify completion heuristics, which are based on current quilt design practices, to control the appearance of the completed quilt. Initial user evaluations with novice and expert quilt designers suggest that our tool fits within current design workflows and greatly facilitates designing foundation paper pieceable quilts by allowing users to focus on the visual design rather than tedious constraint checks.

CCS CONCEPTS

• **Computing methodologies** → **Graphics systems and interfaces**.

KEYWORDS

quilting, design tools, craft, fabrication

ACM Reference Format:
Mackenzie Leake, Gilbert Bernstein, and Maneesh Agrawala. 2022. Sketch-Based Design of Foundation Paper Pieceable Quilts. In *The 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22), October 29-November 2, 2022, Bend, OR, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3526113.3545643

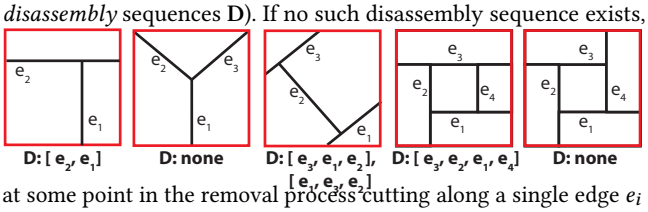
UIST '22, October 29-November 2, 2022, Bend, OR, USA

ACM ISBN 978-1-4503-9320-1/22/10.
https://doi.org/10.1145/3526113.3545643

1 INTRODUCTION

Foundation paper piecing (FPP) is a very popular technique for constructing quilts, especially among beginners, because it involves sewing fabric to a paper *foundation*. The paper provides stability during construction and is printed with the geometric pattern design to serve as a visual guide for sewing the seams precisely. The popularity of foundation paper piecing is reflected by the thousands of websites and videos available online describing the construction process and offering paper pieceable pattern designs.

But, while foundation paper piecing facilitates quilt construction, designing a new paper pieceable pattern is extremely challenging because the sewing process constrains both the geometry and sewing order of the quilt's fabric pieces. Leake et al. [27] have recently shown that a *complete* quilting pattern (i.e., a fully connected, non-degenerate, hole-free partition of a bounded planar region into polygonal faces) is foundation paper pieceable if and only if it is possible to sequentially remove each face from the input design by cutting along a single edge e_i of the face. The inline figure below contains five complete pattern designs with their edge removal (or *disassembly* sequences **D**). If no such disassembly sequence exists,



at some point in the removal process cutting along a single edge e_i must either remove more than one face or not remove a face at all. Leake et al. [27] present an algorithm that takes a complete pattern as input and determines a paper pieceable sewing order for it (as the reverse of the disassembly sequence), if it exists. However, their algorithm does not fit the workflow of quilt designers.

In a formative study of current foundation paper pieceable quilt design practice (Section 2), we observe that expert quilt designers establish the geometry of the pattern iteratively. They sketch multiple options in partial form as a set of edges rather than a complete non-degenerate planar mesh, iteratively adding detail toward a complete design [44, 48]. Periodically throughout this iterative process, the designers manually check whether their partial design admits a foundation paper pieceable sewing order. If not, they consider breaking the pattern into independently paper pieceable *sections*. Our observations suggest that for most of the design process the pattern is a partial sketch rather than a complete planar mesh. Since Leake et al.'s [27] algorithm only works for complete meshes, it can only be applied at the very end of the iterative design process. Moreover, if their algorithm reports the design is not paper pieceable, it is up to the designer to manually adjust the pattern geometry or manually break it into *sections* until the algorithm reports the adjusted pattern and each of its sections is fully paper pieceable.

In this work we address the challenges of helping quilt designers develop and adjust the geometry of a quilt pattern to ensure paper pieceability, even while it is in partial sketch form (Figure 1). Specifically, we develop a new sketch-completion algorithm that can take as input a partial design sketch (i.e., a set of edges lying within a convex bounded region) and automatically generate as output a complete foundation paper pieceable pattern. The output patterns extend the partial input edges to (1) form a complete, fully connected, non-degenerate planar mesh comprising closed polygonal faces, (2) provide a valid paper pieceable sewing order, and (3) break complicated input sketches into a minimal number of independently paper pieceable sections if necessary. Because a partial design often admits multiple visually different completions, designers can further specify completion heuristics that are based on current quilt design practices to control the appearance of the completed quilt. By selecting different heuristics, designers can explore different visual completion options at any stage of their iterative design process or continue adjusting edges in their partial design to further refine its visual appearance. Evaluations with novice quilt designers suggest that our sketch-based tool fits the quilt design workflow very well and makes it much easier for designers to focus on the visual appearance of the quilt, rather than worrying about whether the design adheres to the constraints of paper pieceability.

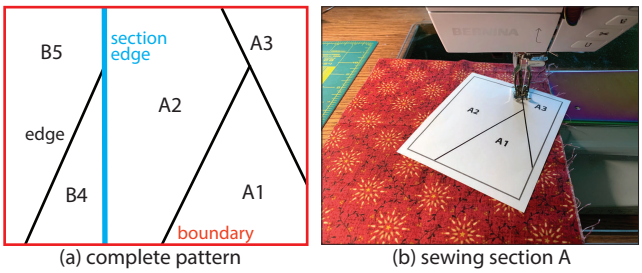


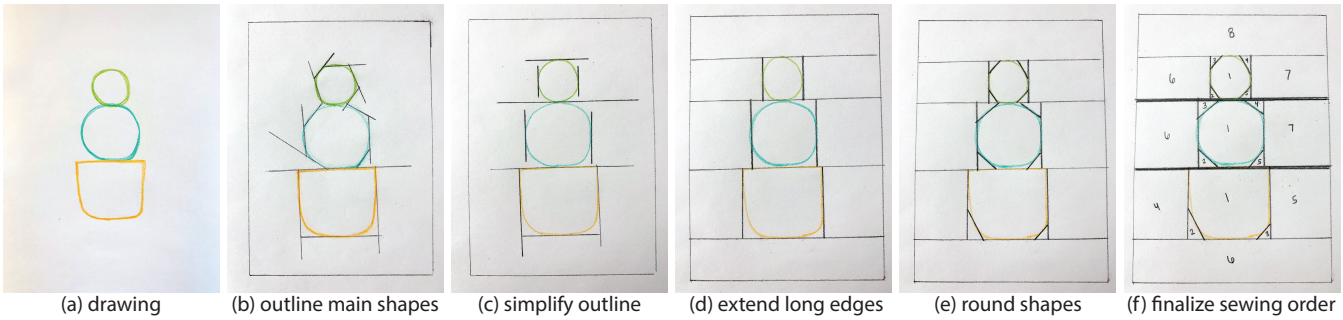
Figure 2: A complete quilt pattern (left) is constructed by layering and sewing fabric pieces at their seams to the paper in the order indicated by face labels (e.g., A2 is sewn to A1). Each section is sewn on a separate sheet of paper, and the sections are sewn together along a section edge.

2 BACKGROUND ON FPP QUILT DESIGN

Numerous books, tutorials and videos describe the sewing process for constructing foundation paper pieceable quilts [2, 8, 18, 25, 34, 44, 48]. A complete quilt pattern contains closed polygonal faces representing pieces of fabric with the *edges* between them representing seams where the pieces are sewn together (Figure 2). *Section edges* break the design into independently paper pieceable subdesigns, and letters indicate the subdesign each face belongs to. The numbering indicates the sewing order within each section. To sew the pattern, each section is printed on a separate piece of paper. The quilter then layers and sews the fabric pieces directly to the paper guide in the sewing order. The paper provides a stable base and serves as a visual guide when sewing the seams. The quilter can then remove the paper foundation, leaving a precisely sewn design. After sewing each section, the quilter must manually sew together the subdesigns at section edges without the benefits of the paper foundation. Thus, quilt designers often minimize the number of section edges in the design [48].

As noted in Section 1, the sewing process imposes constraints on the geometry of foundation paper pieceable quilt designs. To better understand how quilters design paper pieceable quilts subject to these constraints, we reviewed quilt design websites [7, 10, 15, 25, 48] and observed the sketching process of expert quilt designers.

Figure 3 illustrates a typical quilt design process based on an observation of an experienced quilter sketching an original design.



(a) drawing (b) outline main shapes (c) simplify outline (d) extend long edges (e) round shapes (f) finalize sewing order

Figure 3: To manually create a quilting pattern for this cactus drawing (a), the designer first over-sketched outlines of the shapes (b). They then mentally checked if extending the sketched edges in the clockwise order they drew them would yield a paper pieceable design. They realized that this approach would require many sections and some of the section edges would cut through the cactus. They simplified the outline, creating longer horizontal edges (c), and then extended these longest edges first to create a design with three sections (d). They thought it might be possible to use even fewer sections but that this design seemed workable. Next, they added detail to round the shapes (e) and finally determined a sewing order for the design (f).

Quilters often pick a photograph or drawing as the basis for their design (a). They start by sketching edges along the polygonal outlines of the foreground shapes and then iteratively add, delete, and reposition edges to adjust the level of detail in the design. Here the designer considers different outlines of the cactus (b, c), extends the lines creating major sections (d), adds detail (e), and finally determines the sewing order for the faces in each section, as shown by the numbering (f). Through much of the iterative process, design sketches do not form a complete planar mesh and often contain edges that intersect or do not form closed polygons.

Between iterations the designer of the cactus in Figure 3 mentally checked whether extending the edges to form a complete design would yield a paper pieceable sewing order, explaining that this was often done using heuristics like extending the edges in the order they were drawn or extending the longest edges first. When determining sectioning edges, they also considered roughly balancing the number of faces that would fall into each resulting section. They said that performing these checks and breaking the design into sections were the most challenging aspects of designing the quilt. This approach of starting with outlined shapes, producing sections, and then minimizing the number of sections is consistent with the process mentioned by other quilters [7, 10, 48].

3 RELATED WORK

Computational support for craft and fabrication has been explored using many different materials, as detailed in Bickel et al.'s [6] recent survey. For example, researchers have developed tools for knitting [19, 24, 35, 38], weaving [1, 40], cutting and folding paper [13, 26, 36, 50], as well as sewing cloth to make garments [3, 5, 21, 46, 47] and soft objects [37].

Sewing foundation paper pieced designs is similar to other fabrication processes in which materials are attached to scaffolds or jigs as part of the construction process. Prior work has explored how to use scaffolds to design stencils for graphic artwork [22], jigs for wrapped wire jewelry [20, 45], chain-based scaffolds for assembling self-supporting masonry models [14], and laser cut wooden scaffolds for 3D wire meshes [17]. Like in these applications, the paper foundations in foundation paper piecing guide the placement of materials throughout the construction process. Our design tool helps designers convert their partial sketches into complete patterns that serve as paper scaffolds for quilt construction.

Quilting is a domain with many specialized techniques. For instance, free motion quilting is a technique that uses continuous stitching curves, and researchers have developed methods for generating the stitching procedurally [9, 30] or based on photographs [31]. Pieced or patchwork quilts combine fabric pieces in different ways, and researchers have developed tools to help designers create and visualize various types of quilt layouts [11, 23, 28]. Similarly, commercial quilt design tools largely focus on letting users draw, color, and arrange traditional geometric block patterns [12, 16, 41, 42]. While these commercial tools offer some support for foundation paper piecing, none of these prior tools fully account for the constraints imposed by the paper piecing construction process. In fact, those constraints have only recently been formalized by Leake et al. [27], who also present a method for checking if a complete planar mesh input design is single section paper pieceable. But, their approach cannot handle partial input design sketches and cannot break the design into sections. In this work we handle both of these issues, which arise commonly in the quilt design process.

Our work is inspired by tools designed to guide and autocomplete freeform user input in the early stages of visual design. For example, ShadowDraw [29] suggests object contours as a user is sketching, while Rivers et al. [43] similarly guide users as they are sculpting 3D objects via a projector camera setup. Xing et al. [49] present a tool for autocompleting hand-drawn animations by predicting future sketches from previous ones. Liu et al. [32] develop methods to clean up raw sketches by automatically aggregating strokes. In contrast to these methods, we focus on the problem of completing partial quilt design sketches into complete foundation paper pieceable patterns.

4 METHOD

Given a partial pattern design sketch as input, our goal is to generate one or more complete paper pieceable designs as output, breaking the pattern into sections as necessary. Our approach builds on the mathematical theory of single section paper pieceability of Leake et al. [27]. They assume the input design consists of a non-degenerate planar mesh and show that such a design is single section paper

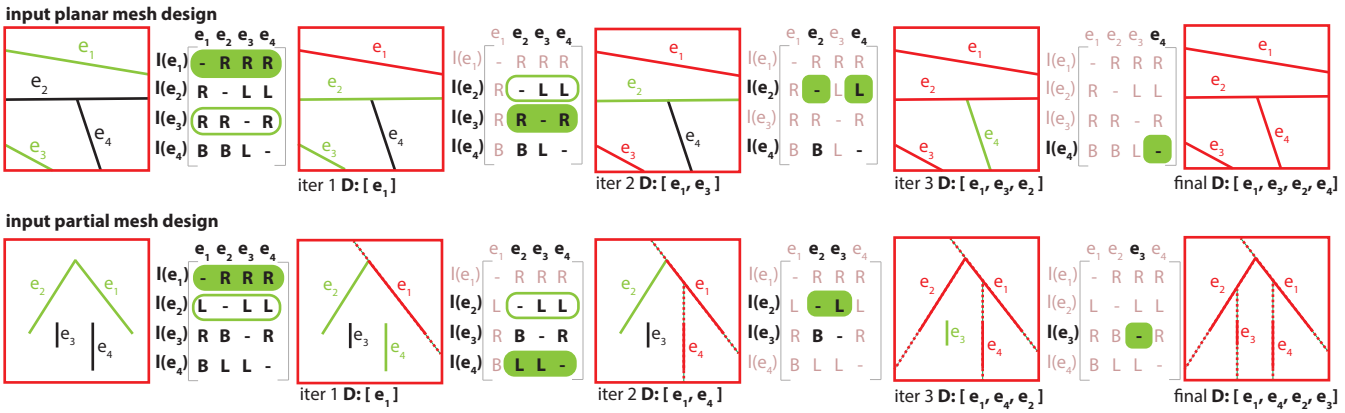


Figure 4: Given an input planar mesh design (top) or partial mesh design (bottom), our algorithm computes a disassembly ordering D of the interior edges $e_i \in I$. In both cases (top and bottom) an edge e_i is cuttable if and only if all other edges e_j lie on on the same side of $l(e_i)$, the line containing e_i. In step 1 our algorithm builds the sidedness graph S for the input design where entry $s_{i,j}$ is set to L, R, or B, depending on whether e_j lies to the left, the right, or on both sides of $l(e_i)$. In step 2 our algorithm iteratively identifies *cuttable edges* (green) by finding rows in S containing only L's or only R's. It chooses one cuttable edge to cut (green filled), removes its row and column from S, moves it from I to the set of boundary edges B (red), and adds it to the disassembly sequence D. If every edge can be removed in this manner, the design is single section paper pieceable, and reversing the D gives a valid sewing order for its edges.

pieceable if and only if it is possible to sequentially remove each face from the input design by cutting along a single edge of the face. Here, we call such an edge a *cuttable edge*. If a complete *disassembly sequence* exists (i.e., one can sequentially remove all faces), reversing the sequence gives a valid sewing order for constructing the design using foundation paper piecing. If no such disassembly sequence exists, at some point in the sequence cutting along a single edge always removes more than one face, and the design is *not* single section paper pieceable.

We advance Leake et al.'s approach in three significant ways. First, we simplify their approach to focus entirely on analyzing the geometry of the edges in a complete planar mesh design rather than analyzing its faces using a hypergraph representation (Section 4.1). We then extend our edge-based approach to handle partial input design sketches (Section 4.2). The resulting algorithm completes the partial sketch into a fully connected planar mesh and generates a sewing order for its faces if it is single section paper pieceable. Our third advance is a new optimization-based approach for hierarchically breaking an input design into multiple paper pieceable sections when it is not single section paper pieceable (Section 4.3). As we present our edge-based algorithm, we show that it is equivalent to constructing an Binary Space Partition (BSP) tree [4, 39], where the edges serve as partitions, and the resulting tree is as *unbalanced* as possible.

4.1 Edge-Based Disassembly

One key insight of our approach is that if we limit the input designs to having convex boundaries, we can identify its *cuttable edges* by examining only the spatial relationships between the edges and without considering any faces. Consider a convex bounded planar mesh input design (Figure 4, top row). We assume the edges on the boundary of the planar mesh are marked as *boundary* edges $e \in B$ and that they form a convex hole-free polygon containing a fully connected set of faces, with the edges of these faces marked as *interior* edges $e_i \in I$ if they lie inside the boundary. An interior edge e_i is cuttable if and only if all other interior edges e_j lie on the same side of $l(e_i)$, the line containing e_i. If there are interior edges on both sides of e_i, then it is not a cuttable edge because the other interior edges must break both sides of e_i into more than one face; cutting along e_i would remove more than one face from the design. Thus, we start by building a graph representing this sidedness relationship between all pairs of interior edges in the design and then iteratively remove cuttable edges using the graph.

Step 1: Build sidedness graph. We build the sidedness graph S for the input design as an adjacency matrix (Figure 4). For each pair of interior edges e_i and e_j in the design, we set entry $s_{i,j}$ of the matrix to a value of 'L' if e_j lies to the left of line $l(e_i)$, to 'R' if e_j lies to the right of $l(e_i)$, or to 'B' if e_j lies on both sides of $l(e_i)$. Note that while the orientation of $l(e_i)$ must remain consistent within each row of S, we do not require any form of orientation consistency between different rows of S. In Figure 4, $l(e_1)$ and $l(e_3)$ are treated as having opposite orientations, and therefore, for both lines all of the other internal edges lie to the right.

Step 2: Iteratively remove cuttable edges. Using the sidedness graph, we identify the set of cuttable edges $C \subseteq I$ by finding rows of S which contain only L's or only R's (Figure 4, green edges). We choose one such cuttable edge $e_i \in C$, and cut it from the design by removing the row and column for e_i from the sidedness graph S, moving e_i from the set of interior edges I to the set of boundary edges B, and then adding e_i to a disassembly ordering sequence D. Note that since removing a cuttable edge amounts to eliminating a row and column from S, once an edge e_i becomes cuttable, it must remain cuttable even if another cuttable edge is removed before it. We iteratively repeat this removal process of identifying cuttable edges and cutting one of them until the set of cuttable edges C is empty. If the iterative process eventually cuts every one of the interior edges, the input design is single section paper pieceable and

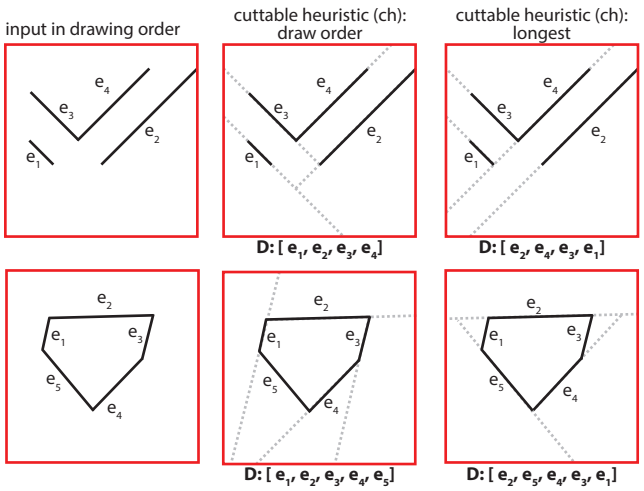
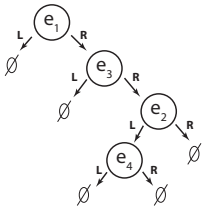


Figure 5: Completing a partial input design produces different looks, depending on the order in which the cuttable edges are removed from the design. Here, edges of the input designs (left) are given in drawing order—i.e., e_2 was the second edge drawn. We provide cuttable edge heuristics (ch), for choosing which cuttable edge to remove in each iteration of our disassembly algorithm. Heuristics include *draw order* (middle) and *longest* (right).

reversing the disassembly sequence D provides a paper pieceable sewing order for the edges. Otherwise, the iteration ends with $C = \emptyset$, but with uncut interior edges remaining in the design, and we report that the input design is not single section paper pieceable.



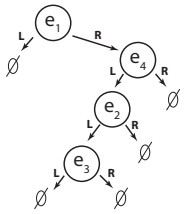
This iterative edge-based disassembly process is equivalent to building a BSP tree where the first edge in the disassembly sequence is the root partition and each subsequent edge partitions either the right or left halfspaces of its parent, depending on which side of the parent edge it falls on. Moreover, because each edge is cuttable when it is removed, one of its child halfspaces must always be empty, thereby producing BSP tree that is as unbalanced as possible. The inset figure (right) shows the BSP tree corresponding to the input design in Figure 4 (top).

4.2 Completing a Partial Input Design Sketch

We observe that because our edge-based disassembly algorithm (Section 4.1) does not rely on any information about faces, we can apply it directly to a partial input sketch, consisting of a set of edges rather than a fully connected planar mesh. In this setting, the input sketch may contain interior edges that intersect or that do not form closed faces (Figure 4, bottom row). Nevertheless, an interior edge e_i is cuttable if and only if all other interior edges e_j lie on the same side of $l(e_i)$. Without loss of generality, suppose all other interior edges e_j lie to the right of $l(e_i)$ (e.g., in Figure 4 bottom row, e_1 in input design). Cutting along $l(e_i)$ all the way to the nearest boundary edges B, is guaranteed to generate a single removable polygonal face to the left of the cut line. If however, there are interior edges on both sides of e_i, neither side of $l(e_i)$ can contain a single removable polygon. Instead both sides will either contain multiple polygons or other disconnected interior edges; either way, e_i is not a cuttable edge. Thus, we can apply the disassembly algorithm of Section 4.1 to generate a single section paper pieceable sewing order if it exists. Importantly, if such an ordering is found, we can complete the partial input design sketch into a fully-connected planar mesh by modifying step 2 as follows:

Modifying step 2 to complete a partial design. Recall that in step 2 of the disassembly algorithm we choose a cuttable edge e_i, remove it from S, and then move it from I to B. For a partial input design, we first extend the cuttable edge e_i outwards in both directions to the closest boundary edges before moving it from I to B. Extending the cuttable edge in this manner allows us to explicitly represent the removable face, and the collection of removed faces partitions the plane into a fully connected planar mesh.

Modifying step 2 to explore design completions. With a partial input design sketch, the order in which cuttable edges are removed from the design can impact the visual look of the completed design. In particular, when we extend a cuttable edge e_i to the current set of boundary edges, the length of the extension may differ depending on which edges were cut before it (Figure 5). To explore these design options, we let users pick from a set of heuristics for selecting among the cuttable edges in step 2. Our cuttable edge selection heuristics (ch) are based on our observations of the heuristics expert quilt designers use when choosing how to extend edges in their manual design process (Section 2). Specifically, these heuristics include; (1) *draw order*, which chooses the edge drawn earliest by the designer, (2) *longest*, which chooses the longest cuttable edge (pre-extension) and (3) *shortest*, which chooses the shortest cuttable edge (Figure 5).



If our modified algorithm runs to completion, the disassembly ordering remains equivalent to a completely unbalanced BSP tree, as the modifications simply extend the edges and do not affect their space partitioning properties. The inset figure (right) shows the BSP tree corresponding to the input design in Figure 4 (bottom).

4.3 Breaking the Design into Sections

If a partial design sketch is not single section paper pieceable, we must break it into multiple independent paper pieceable sections. Following the practice of expert quilt designers (Section 2), our algorithm aims to minimize the number of sections in the resulting design in order to reduce the complexity of constructing the quilt. But, the placement of section edges can also affect the visual aesthetics of the design, as they may break faces that should appear homogeneous (e.g., use a single piece of patterned fabric), into separate pieces. As noted at the end of Section 4.2, our approach assumes that quilt designers specify their aesthetic intent via the set of edges they provide in their partial input design sketch. Thus, our sectioning algorithm is also subject to the constraint that section edges may only appear along lines containing edges of the input sketch. Our algorithm cannot add new edges; it can only extend those that already exist.

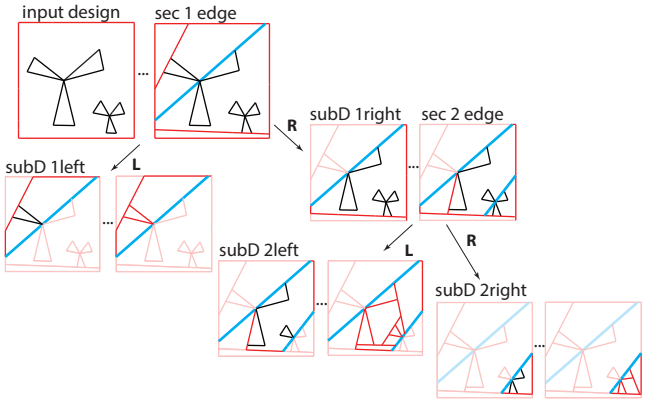
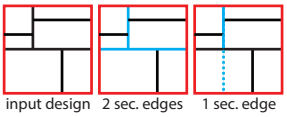


Figure 6: Our hierarchical sectioning algorithm generates a design tree. Starting from the input design, we iteratively remove cuttable edges until we need to split the design at section edge 1 (blue) and form subdesign 1left and subdesign 1right. We iteratively remove cuttable edges from both subdesigns. Subdesign 1right eventually needs to split into two subdesigns at section edge 2 (blue). The highlighted portions of each design show how the different pieces are completed in our algorithm. The resulting multi-section paper pieceable design is shown in Figure 9.

Hierarchical sectioning to form a design tree. Our approach is to build sections hierarchically by recursively splitting the design into two sections (or subdesigns) as necessary. Specifically, if step 2 of our disassembly algorithm stops early, with no cuttable edges $C = \emptyset$ but with edges remaining in $I \neq \emptyset$, we split the design along one of the remaining interior edges. We choose a *section edge* $e_i \in I$, extend it to the current set of boundary edges, and then form two subdesigns, one containing all remaining interior edges that lie to the left of $l(e_i)$, and the other containing those that lie to the right of $l(e_i)$. Any interior edge that intersects $l(e_i)$ is split into two sub-edges at the intersection point, and each sub-edge is placed in the corresponding subdesign. Both subdesigns inherit all of the boundary edges B of the parent design. We then recursively apply our edge-based algorithm to compute a disassembly order for each of the subdesigns. This recursive approach may further split the subdesigns and thereby generate a tree of subdesign sections. We call this hierarchy a *design tree* and treat the input partial design sketch as its root (Figure 6).

We note that even with the introduction of section edges, our algorithm is equivalent to generating a BSP tree, but in this case, some of the internal partitions – namely those corresponding to section edges – include non-empty left and right children. Specifically, to form the BSP tree, we consider the design tree produced by our hierarchical sectioning algorithm (Figure 6), and we build a BSP tree for each section subdesign, treating its parent section edge as the root and adding each cuttable edge in the diassembly ordering for the subdesign as a parition.

Choosing section edges. The choice of which of the interior edges to set as the section edge can significantly impact the total number of sections needed to construct the design as well as the visual aesthetics of the design. In this example (inset), if we pick the horizontal edge as the first section edge, we eventually have to pick another section edge (inset middle). Picking a vertical section edge instead does not require another sectioning (inset right).



Our approach is to focus first on identifying design trees that minimize the number of sections and then letting users choose among these low construction complexity designs to select one that best meets their aesthetic intent. Given an input partial design, we can exhaustively build all possible design trees (e.g., for every possible choice of interior edge whenever a section is needed), and keep track of the design trees with the minimal number of sections. While this approach is guaranteed to find design trees with the minimal number of sections, it can also become combinatorially expensive for designs containing lots of interior edges. However, we have found that using a branch-and-bound strategy to exit early from building out a design tree whenever it contains more sections than a design tree we already built earlier makes this approach efficient enough to run on a wide variety of partial input designs. We discuss this point in more detail in Section 5.

To further facilitate exploration of the space of minimal section design trees, we let users specify a section edge heuristic (sh) for selecting an interior edge for sectioning. Our section edge heuristics are similar to our cuttable edge selection heuristics and are also based on our observations of expert quilt design practices (Section 2). They include selecting the section edge based on (1) *draw order*, (2) *longest* length, and (3) *shortest* length. In addition we include a *balance* heuristic, which selects the edge that splits the design into subdesigns that contain the most balanced number of interior edges, i.e., those that produce the smallest difference between the number of edges remaining on the left and right. While the first three heuristics are aimed at producing different visual looks, the goal of the *balance* heuristic is to produce sections that have similar numbers of faces and therefore require similar amounts of sewing to piece together. Designers can choose which of these heuristics to apply, and our branch-and-bound algorithm then finds the design tree with the minimal number of sections that also best meets the chosen heuristic (Figure 1). If no heuristic is chosen, the algorithm simply outputs all design trees with the minimum number of sections.

4.4 Visualizing & Constructing the FPP Pattern

Traditionally a single section foundation paper pieceable quilt design numbers the faces of the design according the sewing order in which they should be sewn onto the paper. A multi-section design usually includes a letter indicating which section the face is part of. However, our edge-based algorithm generates a sewing order for the edges in the completed design, not the faces. Figure 7 describes our approach for converting the edge-based sewing order into face labels representing the section each face belongs to and the order in which the face should be sewn into the quilt. Constructing a multi-section quilt according the hierarchical design trees we produce involves first sewing the leaf level sections independently, joining them at their parent section edge, and then treating the resulting joined sections as a single face that can be paper pieced into the next higher subdesign in the tree. For the carrot quilt design in Figure 7, we would paper piece all the faces labeled A using one

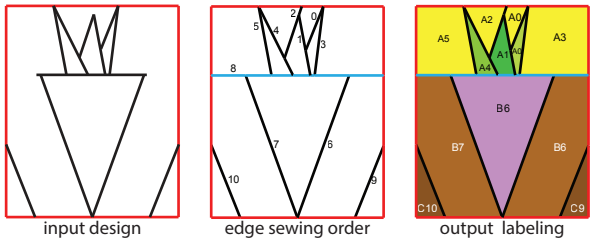


Figure 7: Our completion algorithm computes a sewing order for completed edges (middle). To order the completed faces, we choose the minimum sewing order value of the perimeter edges of each face, as that is the edge that first attaches the face into the section. To compute the alphabetic section labels, we do a post-order traversal of the design tree. Since each subdesign corresponds to a section, we also apply consecutive letters to indicate the faces that belong to each subdesign. In some sections a pair of adjacent faces are assigned the lowest numbering (e.g., here in section A there are two faces labeled A0, section B contains two B6's, etc.). These are the first two that must be attached in the section.

sheet of paper, all the faces labeled B using another sheet, then manually join the sections at the section edge without paper, and finally use a third sheet of paper to join the faces labeled C.

5 RESULTS

We have applied our algorithm to complete a variety of partial input design sketches (Figures 8–11). For the partial sketches in Figures 8 and 9, we explored different heuristics settings for selecting the cuttable edges and section edges to extend in each iteration of our disassembly algorithm. Note that we added colors to the faces of the patterns after they were completed by our algorithm to make it easier to visually distinguish foreground and background elements. Quilters often visualize patterns in this way.

Figure 8 shows six completion possibilities for the milk carton input design. All six contain the minimum number of two section edges. However, these completions differ in the lengths of edges and shapes of faces – the first four from left to right vary the section edge heuristic and differ in the placement of those edges, while the last two vary both heuristics, and the cuttable edges near the outer boundary of the design vary. The differences can impact the visual aesthetics of the resulting quilt; designers may wish to keep certain regions edge-free to maintain a visual seamlessness (e.g., the main faces of the milk carton), or maintain certain symmetries between faces (e.g., the background faces surrounding the milk carton).

Figure 9 shows one completion possibility for each input sketch, but with different completion heuristics for each one. The windmills, for example, extended the shortest cuttable edges first (i.e., black edges always extend shorter edges of windmill blades), but also the longest sectioning edges first (i.e., blue edges extend the longest edges of windmill blades). The balance heuristic for section edges, as used in the teapot and ghost designs, chooses sectioning edges that balance the number of faces within the sections on either side of the edge as much as possible.

Figures 10 and 11 walk through examples of patterns created using an interactive sketch-based quilt design tool we developed to support quilt designers (Figure 12). In both cases we start with a basic sketch of the foreground elements of the design, examine how it might be completed into a paper pieceable design, and then refine the design to improve its visual aesthetics. This workflow matches the way quilt designers often work, starting from a rough sketch and iteratively refining it, but with the added benefit of being able to see possible completions at any point in the process. As shown in our supplemental video, the tool also lets designers adjust the design and/or select different completion heuristics and regenerate the completion to explore different possibilities easily and quickly.

5.1 Evaluations

Running time. Our unoptimized JavaScript implementation of our algorithm required 0.5 to 28 seconds on a 2018 MacBook Pro, 2.9 GHz Intel i9 to complete the partial input designs we have tried. The running time is dominated by our sectioning approach, which searches over the space of design trees to find the ones containing the minimal number of sections. As noted in Section 4.3, even with our branch-and-bound early exit strategy, the search can be combinatorially expensive when the input design contains many edges and their geometric configuration requires many sections. However, in practice, because quilt designers usually aim to produce repeating blocks of moderate construction complexity, most patterns contain 10-25 edges. It is very rare to see patterns with more than 40 edges. As shown in Figure 9, we find that at this scale of input edges, the combinatorics of our branch-and-bound approach are manageable. The longest running example we have tried contained 47 input edges and 4 section edges and took 28 seconds to complete.

User evaluation. We invited three quilters to design foundation paper pieceable quilts using our sketch-based tool. All three were experienced at sewing quilts from a paper pieceable pattern and were somewhat familiar with the challenges of designing such quilts, but none had created a complete paper pieceable pattern on their own. Figures 1 and 13 show their iterative design processes.

One quilter based their sketches on a photo of an apple (Figure 1). They first drew a square outline around the main part of the fruit (a) and then added edges to round the corners (b). Next, they added detail, including the dent at the top of the apple (c), the leaf (d), and finally the stem (e). They considered adding the highlight on the left side of the apple but ultimately decided not to break up the interior of the apple, as the pattern might become too complicated. They set both the cuttable edge and section edge heuristics to *draw order* throughout the design process.

Another quilter worked from a photo of a blue gill fish (Figure 13, top). They outlined the silhouette of the fish and its largest fins (a). They completed the sketch and saw that the resulting design included two sections forming a T-junction. They thought this might be difficult to sew without the benefit of a paper guide, and so they tried using the *longest* section heuristic (b) but only eliminated the T-junction when they tried the *shortest* section heuristic (c). Finally, they tried different cuttable edge heuristics to adjust the shapes of the faces around the front of the fish, until they reached a configuration that they thought looked best (d).

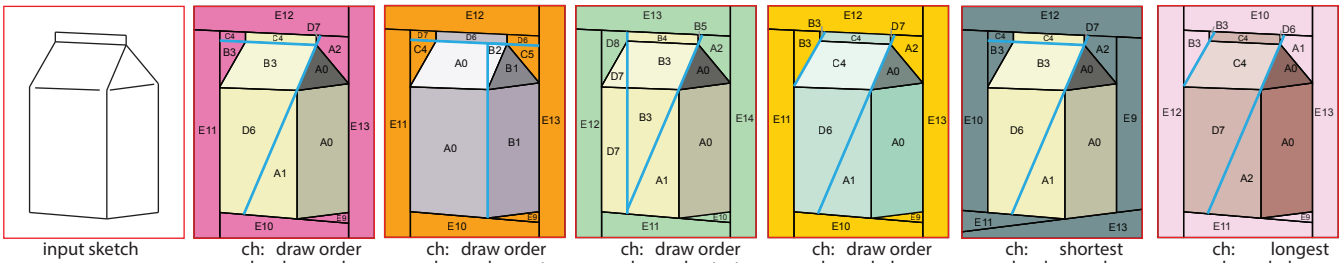


Figure 8: This input sketch of a milk carton contains 14 edges, and all 6 completions contain the minimum number of 2 section edges (blue), but each one uses a different pair of completion heuristics, denoted (ch) for cuttable edge heuristic and (sh) for section edge heuristic. The heuristics cause the lengths of edges and the shapes of faces to vary among the 6 completions.

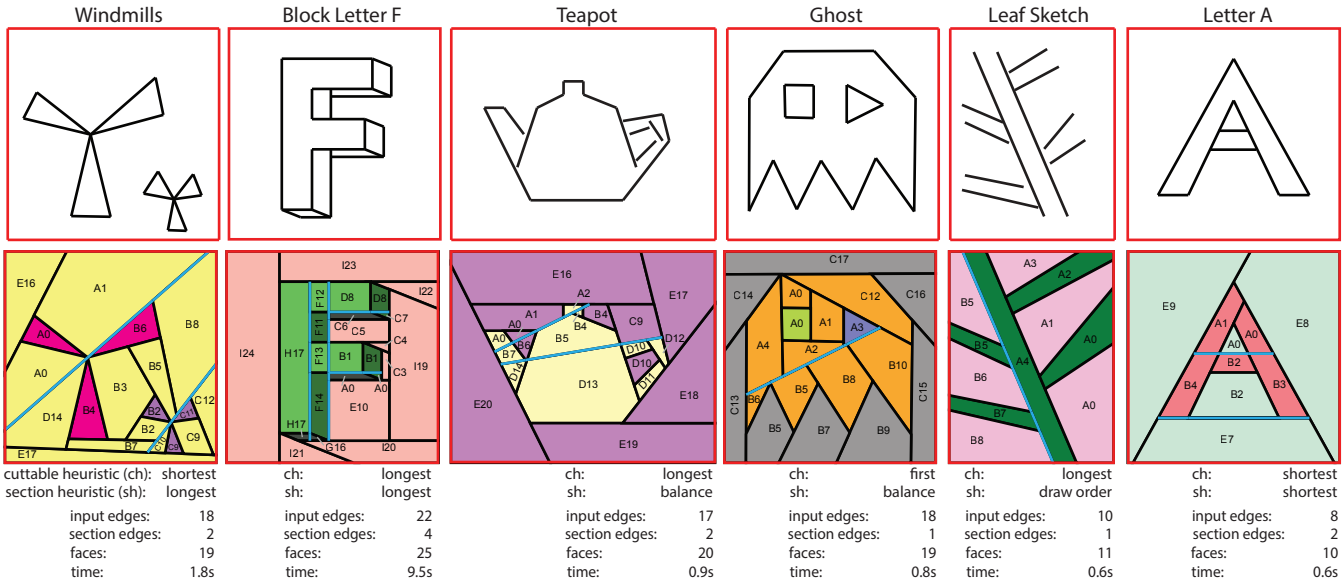


Figure 9: A variety of partial input design sketches and corresponding completions generated with our algorithm. The completions minimize the number of section edges (blue) required for paper pieceability and show one choice of completion heuristics. The total running time for these designs depends on the number and geometric configuration of the input edges.

The third quilter sketched over a photo of a landscape (Figure 13, bottom). They first drew two edges for the ground and asked the tool to generate the completion (a). They then added the mountain tops on the right and paused to complete the design (b), double checking that the design did not introduce a section and that none of the faces were getting too small to sew easily. Next, they added the mountain on the left (c) and changed the cuttable edge heuristic to *shortest* to adjust the edge extensions in the sky region (d).

All three quilters rated our sketch-based design tool using a 5-point Likert scale very positively. Specifically they found the tool helpful in creating their designs (M = 4.7). They also thought using our tool would be preferable to designing patterns manually (M = 5). One participant shared, *"Having to handle the ordering myself would feel like a roadblock to my creative process. I want to do the creative bits and let the tool handle the ordering. I really appreciate this [tool]."* Another participant shared, *"This tool is so much more approachable as someone who has sewn FPP blocks but has never drafted one. This gets me thinking about all of the farm blocks I could make – how I'd do the chickens and the barn and the fence."* Another participant said, *"I have a hard time visualizing the end product for a design so seeing the completion helps."* They also thought our tool would be preferable to the tool of Leake et al. [27] (M = 5). All three noted that our sketch-based interface was less restrictive than theirs and allowed for easy experimentation at early stages of the design process when the pattern is not yet complete.

Participants also shared a few suggestions for the tool. Two participants suggested adding a scale bar to the interface to make it easier to determine if a design was going to require pieces that would be too small to sew easily. One also suggested that users should be able to specify faces that should not be split by extended edges or section edges in the completion.

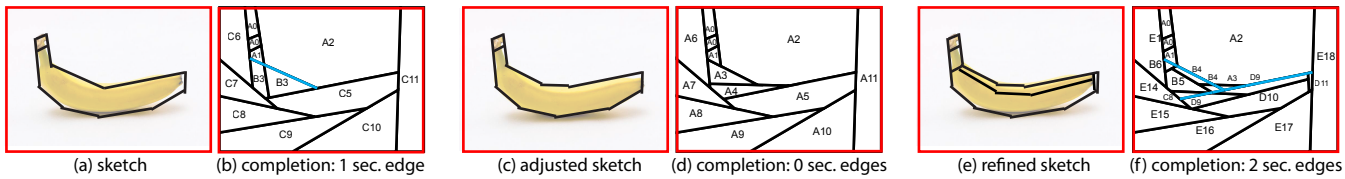


Figure 10: For this banana quilt, we start with a photograph of a banana and sketch edges along its silhouette to establish the main banana shape (a). Our interactive tool reports that this initial design requires a section edge (blue) (b). We try adjusting the edges near the section edge, looking for and eventually finding a design that eliminates the sections (c, d). We then add more shading detail inside the banana by sketching edges along the interior contour (e). The final pattern (f) requires 2 section edges, making it more complicated to construct than the earlier single section design (d), but it provides better visual detail. All completions in this case use *draw order* for both the cuttable edge heuristic (ch) and the section edge heuristic (sh).

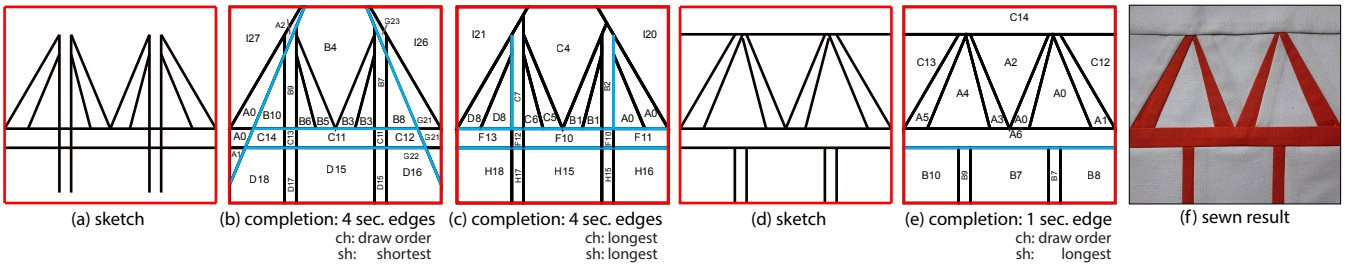


Figure 11: For this bridge quilt, we start with a rough sketch of its primary shapes (a). The initial completion (with heuristics set to cuttable edge heuristic (ch): draw order, section edge heuristic (sh): shortest) extends the angled section edges (blue) from the long suspension cables all the way through the roadway, breaking it into small pieces (b). Switching to (ch: longest, sh:longest) eliminates this problem, but the resulting design is complex with 4 section edges (c). We iteratively adjust our sketch and eventually create a simpler design with shorter supports and a flat top (d). The paper pieceable completion contains one section edge and is visually less cluttered with small pieces (e). We sew this design (f).

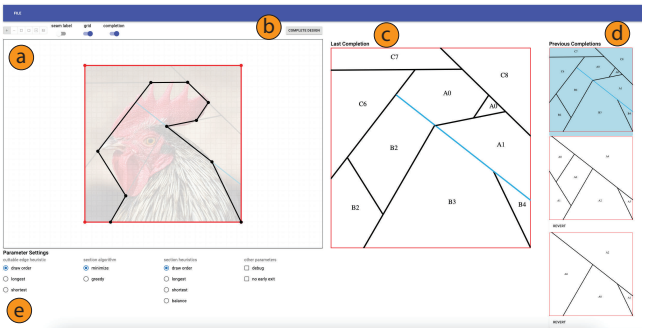


Figure 12: Our interactive tool lets users sketch a partial design (a). Clicking "complete design" (b) produces a completion (c) labeled with the sewing order. Users can add, remove, or reposition edges, revert to previous completions (d), change heuristics settings (e), and complete the sketch again.

5.2 Discussion

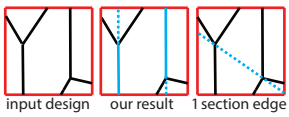
Convex boundary assumption. While our approach can complete partial input designs containing only edges, it does rely on one important assumption; the boundary of the input design is assumed to be convex. This assumption is essential to our disassembly approach (Section 4.1), where we build and process the sidedness graph that maintains the relationships between pairs of edges. In practice, quilt designers almost always design patterns as convex blocks, most often rectangular blocks, but sometimes hexagonal. We also note that Leake et al.'s [27] approach based on analyzing faces in an input planar mesh design can handle non-convex boundaries.

Hierarchical sectioning. Our algorithm generates hierarchical completions in which the leaf level sections must be paper pieced individually, joined at their section edge, and then treated as a single face in the parent design. But traditional foundation paper pieceable quilt designs are not hierarchical. Instead, each section is treated as a completely independent piece. In such non-hierarchical designs, the section edges always partition the design all the way to its input boundaries and can therefore extend section edges much farther than absolutely necessary. In contrast our hierarchical approach can yield shorter section edges and thereby better reflect the input design as specified by the designer.

6 LIMITATIONS AND FUTURE WORK

While our completion algorithm enables creative exploration of paper pieceable quilt designs, it does have some limitations that provide directions for future work.

Unconstrained section edges. Our sectioning algorithm (Section 4.3) is constrained to create section edges by extending edges that appear in the input design. This constraint gives designers some control over the visual aesthetics of the completed design, as they can be sure that the completed design cannot generate edges outside the lines of the input design. However, there are designs for which this constraint makes it impossible to find the globally minimum number



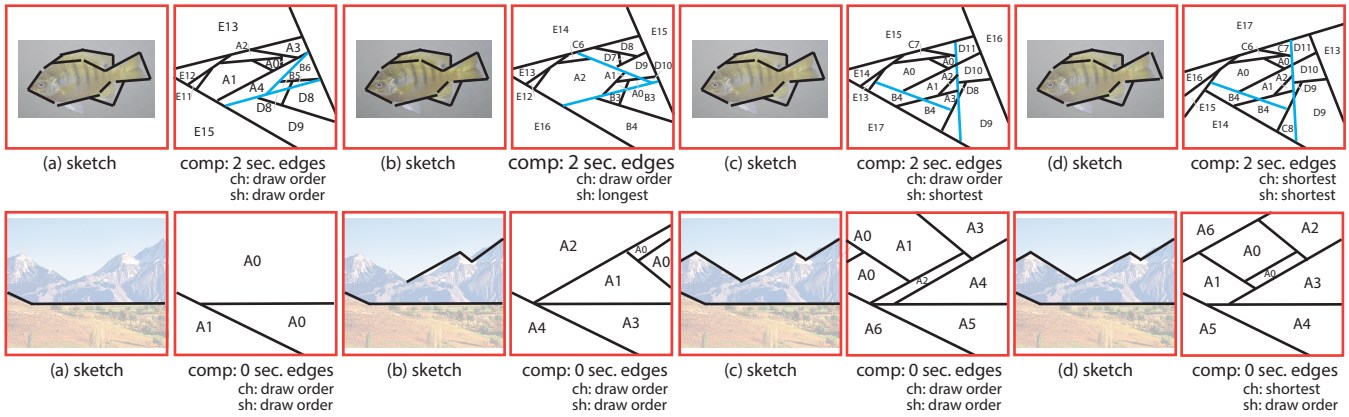


Figure 13: Novice quilt designers used our sketch-based tool to design patterns based on a fish (top) and a landscape (bottom). For the fish the quilt designer sketched the silhouette of the fish and its major fins (top, a) and then explored different section edge and cuttable edge heuristics to produce a design with sections they thought would be easiest to join (top, b-d). For the landscape the designer started by sketching the ground (bottom, a) and then added mountains a few lines at a time (bottom, b, c). They tried different cuttable edge heuristics to produce a pattern that they thought split the sky into pieces nicely (bottom, d).

of sections. In the inset example, when section edges are constrained to lie on existing edges, the minimum number of section edges (blue) is two (here our algorithm uses the *longest* section edge heuristic). But as shown, the globally minimum number of sections edges is one. We note that the aesthetics of these two designs are different, with the latter design introducing long new edges, not present in the input. We leave it to future work to consider placing section edges that lie beyond those in the input design and find the globally minimum number of section edges.

Reasoning about heuristics. Most foundation paper piecing quilters are primarily experienced at constructing such quilts rather than designing them. When we ask quilters to use our design tool, they generally try out the different heuristics until they reach a configuration they like. Since thinking about paper piecing design is largely new to them, sometimes the heuristics are somewhat difficult to reason about by name alone. The current UI allows users to try different heuristics quickly and view them in the history panel. It is not clear from our sessions with users which heuristics are preferred. Generating all of the options would allow users to select visually pleasing designs, but this would come at the expense of speed on larger designs, which could hinder rapid sketch iteration.

Iterative refinement of sketches. Our focus for this implementation was to develop an automated approach to extending edges to form paper pieceable designs. Some quilters frequently use the complete button to build patterns. This process of incrementally checking a design is quite different from existing manual approaches to FPP design, which often have more of a delay between sketching and checking the design. Our tool lets users sketch designs and try different heuristics, but there is currently no direct way to preserve the edges extended by our completion algorithm between iterations. To "lock" a particular edge, users can draw over a completed edge if they want to make sure it appears in the final result.

Application to other fabrication domains. While our disassembly and sectioning algorithms are based on the constraints imposed by foundation paper piecing, there may be other domains with similar constraints. For example, in prior work BSP trees have been used to partition a 3D model into parts than can easily be 3D printed within a small printing volume [33]. This approach considers volume fit and easy assemblability as the primary constraints. In contrast, our disassembly algorithm is based on the idea of cutting through an input design along a straight line *cuttable edge* and then sewing the resulting pieces back together. Similarly, with a crystalline material like ice, it may be easier to make planar cuts all the way through the material and fuse resulting pieces back together than to precisely stop a cut in the middle of the material. Hot wire cutting through foam material may also impose similar constraints that make it easier to make planar cuts all the way through the material and glue the pieces together than to stop a cut in the middle of the foam. Following these analogies, it may be possible to adapt our algorithms to the contexts of 3D ice or foam sculpting or other related domains.

7 CONCLUSION

We have presented a tool for turning a partial quilt design sketch into a complete foundation paper pieceable pattern. The algorithm is designed to fit into an iterative workflow so that even at the earliest stages, when designers are sketching out the shapes that should appear in the design, they can apply the algorithm to get a sense for how the sketch might be completed into a pattern. While foundation paper piecing is a very popular quilting technique, today most quilters focus on applying the technique to construct patterns designed by others. We believe that our approach can open the door to quilt design for many more would-be designers.

ACKNOWLEDGMENTS

We thank the Silicon Valley Modern Quilt Guild members and Stanford Engineering librarians who shared their insights about foundation paper piecing with us. M.A. dedicates this work to his late mother, Radhika Agrawala who patiently taught him to appreciate the craft of quilt-making.

REFERENCES

[1] Lea Albaugh, James McCann, Lining Yao, and Scott E. Hudson. 2021. Enabling Personal Computational Handweaving with a Low-Cost Jacquard Loom. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 497, 10 pages. https://doi.org/10.1145/3411764.3445750

[2] Ingrid Alteneder. 2020. *Adorable Animal Quilting*. Page Street.

[3] Aric Bartle, Alla Sheffer, Vladimir G. Kim, Danny M. Kaufman, Nicholas Vining, and Floraine Berthouzoz. 2016. Physics-Driven Pattern Adjustment for Direct 3D Garment Editing. *ACM Trans. Graph.* 35, 4, Article 50 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925896

[4] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. 1997. Computational geometry. In *Computational geometry*. Springer, 1–17.

[5] Floraine Berthouzoz, Akash Garg, Danny M. Kaufman, Eitan Grinspun, and Maneesh Agrawala. 2013. Parsing Sewing Patterns into 3D Garments. *ACM Trans. Graph.* 32, 4, Article 85 (July 2013), 12 pages. https://doi.org/10.1145/2461912.2461975

[6] Bernd Bickel, Paolo Cignoni, Luigi Malomo, and Nico Pietroni. 2018. State of the Art on Stylized Fabrication. *Computer Graphics Forum* 37 (2018). http://vcg.isti.cnr.it/Publications/2018/BCMP18

[7] Ruth Blanchet. 2022. *Designing a Foundation Pieced Block from a Photograph*. https://arbeedesigns.com/blogs/news/designing-a-foundation-pieced-block-from-a-photograph

[8] Katy Cameron. 2012. *Foundation Paper Piecing For The Terrified*. www.the-littlest-thistle.com/2012/02/foundation-paper-piecing-for-terrified_20.html

[9] Christopher Carlson, Nina Paley, Theodore Gray, et al. 2015. Algorithmic quilting. In *Proceedings of Bridges 2015: Mathematics, Music, Art, Architecture, Culture*. Tessellations Publishing, 231–238.

[10] Sheri Cifaldi-Morrill. 2018. *Behind the scenes: Designing the Butterfly Bunch quilt pattern*. https://blog.wholecirclestudio.com/?p=2379

[11] Marge M Coahran and Eugene Fiume. 2005. Sketch-Based Design for Bargello Quilts.. In *SBM*. 165–174.

[12] Arnout Cosman. 2012. Quilt Assistant v2.24. https://quiltassistant.com/

[13] Erik D Demaine and Martin L Demaine. 2002. Recent results in computational origami. In *Origami3: Third International Meeting of Origami Science, Mathematics and Education*. 3–16.

[14] Mario Deuss, Daniele Panozzo, Emily Whiting, Yang Liu, Philippe Block, Olga Sorkine-Hornung, and Mark Pauly. 2014. Assembling self-supporting structures. *ACM Trans. Graph.* 33, 6 (2014), 214–1.

[15] Carol Doak. 2003. Piecing on paper: Designing your own blocks. *Quilter's Newsletter Magazine* (July/August 2003), 44–75.

[16] ElectricQuilt. 2017. Electric Quilt 8 (EQ8). https://electricquilt.com/

[17] Akash Garg, Andrew O Sageman-Furnas, Bailin Deng, Yonghao Yue, Eitan Grinspun, Mark Pauly, and Max Wardetzky. 2014. Wire mesh design. *ACM Transactions on Graphics* 33, 4 (2014).

[18] Hayley Grzych. 2018. *Beginner-friendly foundation paper piecing*. https://weallsew.com/beginner-friendly-foundation-paper-piecing

[19] Megan Hofmann, Lea Albaugh, Ticha Sethapakadi, Jessica Hodgins, Scott E. Hudson, James McCann, and Jennifer Mankoff. 2019. KnitPicking Textures: Programming and Modifying Complex Knitted Textures for Machine and Hand Knitting. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 5–16. https://doi.org/10.1145/3332165.3347886

[20] Emmanuel Iarussi, Wilmot Li, and Adrien Bousseau. 2015. WrapIt: Computer-Assisted Crafting of Wire Wrapped Jewelry. *ACM Trans. Graph.* 34, 6, Article 221 (Oct. 2015), 8 pages. https://doi.org/10.1145/2816795.2818118

[21] Takeo Igarashi and John F Hughes. 2002. Clothing manipulation. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*. 91–100.

[22] Yuki Igarashi and Takeo Igarashi. 2010. Holly: A drawing editor for designing stencils. *IEEE Computer Graphics and Applications* 30, 4 (2010), 8–14.

[23] Yuki Igarashi and Jun Mitani. 2015. Patchy: An interactive patchwork design system. In *ACM SIGGRAPH 2015 Posters*. 1–1.

[24] Alexandre Kaspar, Liane Makatura, and Wojciech Matusik. 2019. Knitting skeletons: A computer-aided design tool for shaping and patterning of knitted garments. In *proceedings of the 32nd annual ACM symposium on user interface software and technology*. 53–65.

[25] Lily Kerns. 2020. *Designing a Foundation Pieced Block from a Photograph*. https://www.academyofquilting.com/library/free-lessons/designing-a-foundation-pieced-block-from-a-photograph/

[26] M. Kilian, S. Flöry, Z. Chen, N. J. Mitra, A. Sheffer, and H. Pottmann. 2008. Curved Folding. *ACM Transactions on Graphics* 27, 3 (2008), #75, 1–9.

[27] Mackenzie Leake, Gilbert Bernstein, Abe Davis, and Maneesh Agrawala. 2021. A Mathematical Foundation for Foundation Paper Pieceable Quilts. *ACM Trans. Graph.* 40, 4, Article 65 (jul 2021), 14 pages. https://doi.org/10.1145/3450626.3459853

[28] Mackenzie Leake, Frances Lai, Tovi Grossman, Daniel Wigdor, and Ben Lafreniere. 2021. *PatchProv: Supporting Improvisational Design Practices for Modern Quilting*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3411764.3445601

[29] Yong Jae Lee, C Lawrence Zitnick, and Michael F Cohen. 2011. Shadowdraw: real-time user guidance for freehand drawing. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 1–10.

[30] Yifei Li, David E Breen, James McCann, and Jessica Hodgins. 2019. Algorithmic Quilting Pattern Generation for Pieced Quilts. (2019), 9.

[31] Chenxi Liu, Jessica Hodgins, and James McCann. 2017. Whole-cloth quilting patterns from photographs. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering - NPAR '17*. ACM Press, Los Angeles, California, 1–8. https://doi.org/10.1145/3092919.3092925

[32] Chenxi Liu, Enrique Rosales, and Alla Sheffer. 2018. StrokeAggregator: Consolidating Raw Sketches into Artist-Intended Curve Drawings. *ACM Trans. Graph.* 37, 4, Article 97 (jul 2018), 15 pages. https://doi.org/10.1145/3197517.3201314

[33] Linjie Luo, Ilya Baran, Szymon Rusinkiewicz, and Wojciech Matusik. 2012. Chopper: Partitioning models into 3D-printable parts. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 1–9.

[34] Nancy Mahoney. 2016. *Learn to Paper Piece: A visual guide to piecing with precision*. Martingale.

[35] James McCann, Lea Albaugh, Vidya Narayanan, April Grow, Wojciech Matusik, Jennifer Mankoff, and Jessica Hodgins. 2016. A compiler for 3D machine knitting. *ACM Transactions on Graphics* 35, 4 (July 2016), 1–11. https://doi.org/10.1145/2897824.2925940

[36] Jun Mitani and Hiromasa Suzuki. 2004. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM transactions on graphics (TOG)* 23, 3 (2004), 259–263.

[37] Yuki Mori and Takeo Igarashi. 2007. Plushie: an interactive design system for plush toys. In *ACM SIGGRAPH 2007 papers*. 45–es.

[38] Vidya Narayanan, Lea Albaugh, Jessica Hodgins, Stelian Coros, and James McCann. 2018. Automatic machine knitting of 3D meshes. *ACM Transactions on Graphics (TOG)* 37, 3 (2018), 1–15.

[39] Bruce Naylor. 1990. Binary space partitioning trees as an alternative representation of polytopes. *Computer-Aided Design* 22, 4 (1990), 250–252.

[40] Daniëlle Ooms, Nick Voskuil, Kristina Andersen, and Hanna Ottilia Wallner. 2020. Ruta, a Loom for Making Sense of Industrial Weaving. In *Companion Publication of the 2020 ACM Designing Interactive Systems Conference* (Eindhoven, Netherlands) *(DIS' 20 Companion)*. Association for Computing Machinery, New York, NY, USA, 337–340. https://doi.org/10.1145/3393914.3395815

[41] PreQuilt. 2020. PreQuilt. https://prequilt.com/

[42] Quiltster. 2020. Quiltster | Digital Quilt Planner. https://www.quiltster.com/

[43] Alec Rivers, Andrew Adams, and Frédo Durand. 2012. Sculpting by numbers. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 1–7.

[44] Sarah Elizabeth Sharp. 2018. *Adventures in Paper Piecing & Design*. Stash Books.

[45] Cesar Torres, Wilmot Li, and Eric Paulos. 2016. ProxyPrint: Supporting Crafting Practice through Physical Computational Proxies. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems (DIS '16)*. Association for Computing Machinery, New York, NY, USA, 158–169. https://doi.org/10.1145/2901790.2901828

[46] Nobuyuki Umetani, Danny M Kaufman, Takeo Igarashi, and Eitan Grinspun. 2011. Sensitive couture for interactive garment modeling and editing. *ACM Trans. Graph.* 30, 4 (2011), 90.

[47] Katja Wolff and Olga Sorkine-Hornung. 2019. Wallpaper Pattern Alignment along Garment Seams. *ACM Trans. Graph.* 38, 4, Article 62 (July 2019), 12 pages. https://doi.org/10.1145/3306346.3322991

[48] Linda Worland. 2020. *Designing Paper Pieced Patterns*. https://www.paperpanache.com/designing-paper-piecing

[49] Jun Xing, Li-Yi Wei, Takaaki Shiratori, and Koji Yatani. 2015. Autocomplete hand-drawn animations. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–11.

[50] Jie Xu, Craig S. Kaplan, and Xiofeng Mi. 2007. Computer generated paptercutting. In *15th Pacific Conference on Computer Graphics and Applications*. 343–350.