ScrapMap: Interactive Color Layout for Scrap Quilting

Mackenzie Leake
leake@adobe.com
Adobe Research
San Francisco, CA, USA

Ross Daly
rdaly525@cs.stanford.edu
Stanford University
Stanford, CA, USA

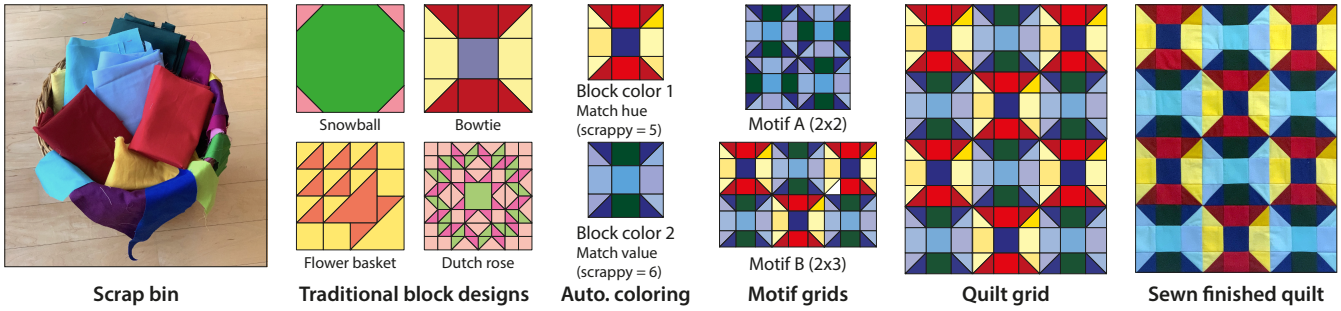


Figure 1: ScrapMap is an interactive design tool for scrap quilts. Users start with their fabric scraps (left). They select a traditional quilt block to make scrappy. Using different objectives, such as matching the hue (Block color 1) or value (Block color 2) of the input block, and a desired scrappiness value, they can explore different block colorings. They then combine these blocks into different motif configurations and visualize the full quilt grid. Finally, they can check if they have enough fabric before sewing the finished quilt in their fabric (right).

ABSTRACT

Scrap quilting is a popular sewing process that involves combining leftover pieces of fabric into traditional patchwork designs. Imagining the possibilities for these leftovers and arranging the fabrics in such a way that achieves visual goals, such as high contrast, can be challenging given the large number of potential fabric assignments within the quilt's design. We formulate the task of designing a scrap quilt as a graph coloring problem with domain-specific coloring and material constraints. Our interactive tool called ScrapMap helps quilters explore these potential designs given their available materials by leveraging the hierarchy of scrap quilt construction (e.g., quilt blocks and motifs) and providing user-directed automatic block coloring suggestions. Our user evaluation indicates that quilters find ScrapMap useful for helping them consider new ways to use their scraps and create visually striking quilts.

CCS CONCEPTS

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

KEYWORDS

Quilting, Fabrication, Graph coloring, Design tools

UIST '24, October 13–16, 2024, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-0628-8/24/10
https://doi.org/10.1145/3654777.3676404

ACM Reference Format:
Mackenzie Leake and Ross Daly. 2024. ScrapMap: Interactive Color Layout for Scrap Quilting. In *The 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24), October 13–16, 2024, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3654777.3676404

1 INTRODUCTION

Quiltmaking is a craft that embraces a wide range of geometric designs. Quilts are often designed hierarchically. In block-based quilts, which are the most common form of quilts, single units, called ***quilt blocks***, are arranged to form ***motifs***, which are repeated to form a ***quilt***. Quiltmaking is widely popular, with an estimated 9-11 million quilters in the United States and Canada [2]. Most quilt designs result in leftover fabrics, which get added to the quilter's ***stash***. Quilters have begun to explore new ways to utilize their leftover materials, and there has been renewed interest in reducing waste and repurposing scraps of fabric.

When people think of quilts, it is common for a ***scrap quilt*** to come to mind. Judy Martin, one of the leading experts in scrap quilts has said, *"Steeped in tradition, totally typical yet absolutely one of a kind, the scrap quilt is the most quilt-like and the best loved of quilts"* [49, p.5]. Scrap quilting is a traditional quiltmaking method that involves assembling pieces of leftover textiles [57]. Quilters put careful consideration into their designs to make visually vibrant and diverse designs. Martin writes: *"Most often, scrap quilts are made from a large number of different fabrics in a wide range of colors. This may give the illusion of randomness, but actually, most quiltmakers use the same care in planning their scrap quilts that they use in their other quilts"* [49, p.20]. Quilters face two common challenges when designing scrap quilts: either they are unsure if they have enough desirable fabrics to create an attractive quilt, or they are overwhelmed by the large number of scraps they have and

do not know where to begin to achieve an attractive design [23, 72]. The space of scrap quilt designs is very large because of the large number of potential geometric designs and placements of different fabrics. Quilters typically rely on adapting existing ***traditional quilt blocks*** (Fig. 1), which use just a few different colors of fabric, to a scrappy version with many different fabrics from their stash.

Scrap quilting is essentially a large matching problem: given a set of potential quilting designs and fabrics, create a layout that is visually attractive and feasible given available fabric. Based on our review of scrap quilting books, we learned that the two pillars of creating a good scrap quilt are ***contrast*** and ***repetition*** [23, 50, 72]. The desired visual appearance and graphic quality of a quilt block are achieved through controlling the differences between fabrics, often in terms of hue and value, for adjacent regions within individual blocks. These blocks then get arranged into groupings, called motifs, which are then arranged into the final quilt. A few common goals for making scrap quilts include maximizing value contrast between adjacent regions within a block, reproducing a traditional design as faithfully as possible given available materials, and using many different pieces of fabric [23, 49, 50].

We present ScrapMap: a design tool that automatically suggests potential fabric assignments for a scrap quilt. We formulate scrap quilting as a graph coloring problem on a planar mesh, and we show that the visual design goals of scrap quilting can be represented as objectives and constraints. Our contributions include:

- An interactive tool for designing scrap quilts in a hierarchical manner with a limited stash of fabric scraps
- A formalization of commonly shared guidance for designing scrap quilts as a constrained optimization problem
- A user-directed automatic block coloring feature using optimization solutions as suggestions

We demonstrate that leveraging the hierarchy of scrap quilt construction (e.g., blocks, motifs, and grids) allows us to provide useful automatic guidance for realistic quilt coloring problems. Current software (e.g., [16, 22, 62, 63]) for quilting allows for manual exploration of color placements within a quilt block but does not provide automatic colorization guidance, incorporate an awareness of the available fabric scraps, or explicitly support motifs. By allowing quilters to explore different realizable designs, they are able to find creative ways to use their leftover fabrics. Our user study indicates that using ScrapMap encourages quilters to explore new designs and would make them more likely to use their scraps in the future.

2 RELATED WORK

Our work builds upon prior work that uses computation to help design craft objects subject to visual and physical constraints and apply color to designs in various domains.

2.1 Textile tools

Craft-focused applications in digital design for fabrication have centered on knitting [27, 32, 34, 52, 54], embroidery [84], smocking [21, 64], and sewing cloth to make garments [5, 6, 8, 29, 37, 38, 59, 73, 77] and other soft objects [53]. Recent work on quilting has focused on two distinct parts of the process: stitching the three layers of the quilt (i.e., top, batting, and backing) and designing the quilt top. Prior work has explored techniques for producing free motion quilt designs used to stitch the quilt layers based on procedural curve generation [10, 44] and edge extraction from a photograph [47]. Prior work on designing quilt tops has focused on various quilt construction techniques, such as bargello [14], patchwork [30], and improvisational quilts [43] and has explored the underlying theory of quilt designs [13, 41]. Several recent papers [3, 40, 41, 71] have focused on foundation paper pieced quilts, which involve stitching fabrics to paper guides while adhering to constraints on the design and sewing plan. Commercial software for quilting allows users to draw, download, and color blocks [16, 22, 60, 62, 63]. While these tools support manually testing color combinations and placements, they do not explicitly focus on the distinct challenges of scrap quilting or provide automatic color placement and layout assistance. In this work we automatically suggest potential colorings for blocks.

2.2 Reducing waste through design

Recent work in computational design and fabrication has explored different ways to incorporate awareness of the material usage into design tools. Several techniques, such as Box Cutter [45], have been developed to pack 2D shapes more efficiently, which can be leveraged to reduce waste in fabrication. Fabricaide [66] allows users to design laser-cut objects while viewing efficient layouts of the parts. Koo et al. [36] support low-waste furniture design by suggesting design alternatives that utilize fewer materials. Other work focused on waste reduction explores creative ways to reuse leftover materials. For example, Larsson et al. [39] develop a human-in-the-loop system for building structures out of found tree branches, Wu and Devendorf [79] consider the design of smart textiles for later disassembly, and Scrappy [76] helps people use leftover materials as infill in 3D printed objects. InStitches [42] helps users repurpose leftover garment fabric for sewing practice. In this work we help quilters use available materials by showing different ways their scraps could be used in quilt designs and providing efficient fabric cutting layouts.

2.3 Color tools

Prior work on color recommendation and recoloring has considered both continuous and discrete coloring problems. Popular applications of recoloring involve applying color to images, videos, and vector graphics based on known priors with or without a user-in-the-loop. For example, prior work has explored palette-based recoloring of images [11, 12, 67, 83] and videos with time-varying palettes [20], image colorization [55], and supporting exploration of color palettes through direct manipulation [70]. Prior work has also focused on detecting [56] and suggesting [15] harmonious color combinations. In addition, interactive tools have been developed for mixing colors digitally [69] and comparing and sorting color palettes [35]. Several tools have been developed for recoloring other types of visual media, such as infographics [81], 2D patterns [46], line art [82], and comic books [80]. In this work we focus on the domain of scrap quilting and suggesting color palettes that meet the constraints of this domain, such as creating contrast between adjacent regions of the design and using the available materials.

Our underlying implementation is heavily influenced by ideas in graph coloring, a well-known area in combinatorial optimization (e.g., [31, 51]). The most common formulation of graph coloring

involves finding an assignment of labels to nodes such that adjacent nodes do not have the same color while minimizing the number of colors. In planar graphs this corresponds to assigning a color to each face such that no two faces that share a boundary have the same color. In this work we impose additional constraints on the graph coloring problem, inspired by the domain of scrap quilting.

3 SCRAP QUILTING BACKGROUND

Scrap quilting is a distinctive quilting style defined by both the approach to making the quilt and its resulting appearance. Quilters combine leftover fabrics, called ***scraps***, into patchwork designs. These designs appear "scrappy" because of the variations in appearance of the colors across different parts of the quilt. While this combination of different fabrics often leads to complex visual designs, the individual units, called ***quilt blocks***, that comprise these quilts often have very simple geometric structures with relatively few ***faces*** each [23, 49] (Fig. 1). Quilt blocks use a piece of fabric for each face. Quilters combine these blocks in various rotations into ***motifs***, which further add to the visual pattern of the quilt (Fig. 3). Despite the fact that these fundamental units repeat, the visual appearance is varied because each block within the motif often utilizes different fabric choices for the same geometry [72].

3.1 Scraps & stashes

Today, most quilters work with lightweight (4oz/sq.in) 100% cotton fabrics of varying colors, though sometimes clothing fabrics are utilized [49]. Quilters typically sort their fabric collections, often called ***stashes***, by hue (e.g., reds, yellows, and purples) and value (e.g., light, medium, and dark shades) [23, 49, 50, 72]. Some quilters keep pieces as small as 1-inch squares, while others only save larger pieces. Judy Martin, a well-known expert on scrap quilting, says that while quilts can use dozens of fabrics, for each block, quilters consider only a subset of the fabric choices [50].

3.2 Step 1: Selecting a block

Scrap quilters typically start with basic, traditional quilt blocks in a repeated grid [49]. Traditional quilt blocks with names, such as "log cabin," "rail fence," and "Ohio star," are geometric designs that have been used for decades, if not centuries. Traditional blocks often are associated with particular fabric hues or values. Martin says, *"Attractive (even stunning) scrap quilts can be made from the most basic patterns"* [49, p.5]. Quilters adapt these traditional designs, which only use a few colors, to scrap quilt designs, which feature many more fabrics (Fig. 1).

3.3 Step 2: Selecting a color palette

Scrap quilts can have colors appearing randomly, but often quilters choose more constrained color palettes. Even a quilt with a relatively limited color palette (e.g., "yellows and purples") can utilize several pieces of fabric with different color values (Fig. 2). Martin instructs quilters to choose their color palettes before they begin: *"If there is any special 'secret,' any key to success with a scrap quilt, this is it: choose your palette of fabrics and colors before you begin. Achieving your desired effect in a quilt is virtually automatic once your palette is selected"* [49, p.22].

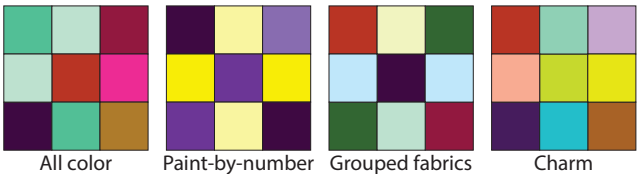


Figure 2: Common types of scrap quilts distribute the fabric colors in different ways within a block, in this case a traditional "nine patch" block. All color quilts use colors scattered throughout the block, and fabrics can be used multiple times. Paint-by-number assigns specific faces specific colors (e.g., yellow and purple). Grouped fabrics divide the fabrics into categories, such as light and dark shades. Charm blocks avoid using the same fabric more than once.

3.4 Step 3: Fabric placement within blocks

Once a quilter decides which traditional block and fabrics they will use, they have to decide where each of these fabrics should go within the block. Several common categories of block colorings for scrap quilts include:

- ***All color (no objective)***: Colors are chosen nearly randomly from available scraps [50].
- ***Paint-by-number (match a target color scheme)***: Specific faces are assigned specific types of colors (but different fabrics) [50].
- ***Grouped fabrics (differences in color/value)***: Fabrics are initially divided into two groups (e.g., light and dark) for placement in each quilt block to create contrast [50].
- ***Charm quilts***: Use no fabric more than once, making every face of every block unique [23].

The location of each color affects how much each of the individual faces and blocks will stand out or blend into their surroundings. Quilters often create multiple different block colorings, which are repeated in motifs.

3.5 Step 4: Motif building

Scrap quilts achieve their distinctive visual style through repeated blocks and motifs. Each block in a motif will have the same geometry but with different choices of rotations and block colorings (Fig. 3). Turner and Rolfe summarize the role of repetition: *"Scrap quilts can be summed up in two words: repetition and variation. The pattern in a scrap quilt is created by the repetition of a block that has a contrast within it, but the interest in the pattern is created by the variation you bring to those repetitions"* [72, p. 13]. However, it is not only the primary designs that emerge within the individual blocks that lead to effective designs; secondary patterns emerge as well. Quilter Judy Florence writes: *"Many scrap quilt patterns contain pieces that react with neighboring units or latticework to create secondary patterns. These secondary patterns come as a surprise, and usually a welcome one, at that"* [23].

3.6 Scrap quilting challenges

Scrap quilt design is hierarchical: quilters focus on selecting and placing fabric colors within blocks, which are then organized into motifs, and ultimately the finished quilt. In essence scrap quilting

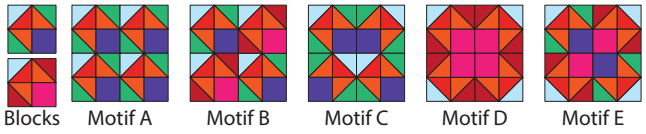


Figure 3: Blocks can be arranged into a motif, which in this case is a 2x2 grid. Even a relatively simple quilt block can create striking secondary patterns by repeating and rotating blocks with different colors.

is about maximizing contrast between certain regions in the design, which is achieved by the juxtaposition of fabrics of different colors [23, 50, 72]. Judy Turner and Margaret Rolfe, authors of a book on scrap quilting, write: *"The repetition of the dark and light values in the same place in each block usually provides the contrast that creates the overall pattern. If the dark and light contrast is not maintained, the pattern becomes lost, and the design descends into chaos"* [72, p. 9]. Turner and Rolfe rank value and then hue as the most important considerations in scrap quilting [72]. Quilt book author Lynn Roddy Brown notes: *"In most quilts, the outline of the shapes is determined by color. Scrap quilts use many different fabrics in a wide range of colors. It is the relative darkness or lightness of the fabrics, or value, which determines the pattern"* [9].

As described in popular scrap quilting books [23, 49, 50], three common goals for scrap quilters are:

- C1 Maximizing color differences between adjacent faces of the quilt block
- C2 Matching the appearance of a traditional design with a scrappy twist
- C3 Using many different fabrics and distributing them across the quilt.

Despite the practical nature of reusing fabric scraps, navigating the design of a scrap quilt is challenging. Many quilters have difficulty knowing if they have the right colors of fabrics to achieve the visual contrast they desire. Turner writes, *"It is easy to become overwhelmed with choices and not know where to begin"* [72]. Given a quilter's fabric stash, the space of possible fabric assignments is enormous. Florence adds, *"It is not unusual to be overwhelmed by your fabric collection. What most quiltmakers need is a point in the right direction and a gentle nudge"* [23, p. 27].

3.7 Design goals

In response to the known challenges of scrap quilting, we outline the following design goals for a tool to support scrap quilting:

- DG1 Help quilters adapt traditional blocks to scrappy versions.
- DG2 Support exploration of possibilities for fabric placement.
- DG3 Recommend designs given available materials.
- DG4 Allow users to personalize designs.

Based on insights from our review of scrap quilting books, we built a tool called *ScrapMap* to support the scrap quilt design process.

4 SYSTEM OVERVIEW

Fig. 4 shows the main components of the ScrapMap UI, which is implemented as a React web app.

4.1 Gallery panel

The user begins by selecting a traditional block from the gallery (Fig. 4a, DG1). Of the 70 blocks, 18 come from the popular quilting website "All People Quilt" [58], 40 come from the "Quilt Builder Card Deck" [61], and 12 come from the book "Building Blocks Sampler Quilt" [75]. These designs range from simple, two-piece geometric designs, such as the "half-square triangle" block (Fig. 7, row 3), to complex star designs, such as the 89-piece "Dutch rose" block (Fig. 1). The user can also upload their own SVG design using the upload button. In this case the user has chosen the "scrap boxes" block from the gallery.

4.2 Block panel

In the Block Panel the user views the coloring results (DG2). The available fabric swatches are shown in the "current fabrics" panel (Fig. 4c). The user can upload new fabrics by clicking the plus button and using the color picker in the dialog to assign the correct color. They can also specify the maximum usable rectangular area of the fabric using the width and height inputs. In this example, the user has chosen to work with the 25-color Kona cotton Elizabeth Hartman fabric grouping that they have in their stash [68]. They can manually apply colors in "change color" mode to personalize the design, or they can assign colors to specific faces of the design using "lock color" mode (Fig. 4d) (Sec. 3.7, DG4).

4.3 Automatic block coloring panel

The Automatic Block Coloring Panel allows users to explore different scrap quilting objectives (Sec. 3.7, DG2, DG3). The two parameterized objectives, which are discussed in more detail in Sec. 5, are based on the challenges outlined in Sec. 3.6. The user has chosen to match the hue in the original block they selected in the gallery (Fig. 4e)(Sec. 3.7, DG1). They can also use the slider to select the amount of "scrappiness," which corresponds to a notion of how many different fabrics are chosen while meeting the objectives checked above. They increase the scrappiness to add more colors. After clicking the "Generate" button, up to 5 automatically generated optimal colorings are returned (Fig. 4f). If charm mode is enabled, each generated coloring will use completely different choices of fabrics (please see Defn. 5.10 and Fig. 7 for more detail). The user can save results in the "Saved blocks" box and visualize them in the motif panel by checking "Use in motif" (Fig. 4g).

4.4 Motif panel

In the Motif Panel the user can see how their blocks will look repeated alongside other blocks in different rotations (Fig. 4h). They can control the number of blocks per row and column (Fig. 4i), and in this case, they set the values to two rows and two columns. They can also choose among four popular motif rotation configurations (Fig. 4j). Here they chose the cycle grid because they liked the radial pattern that emerges in the quilt.

4.5 Quilt panel

In the Quilt Panel (Fig. 4k) the user designs the layout of the quilt by altering the number of motif rows and columns, in this case choosing 3 rows and 4 columns (Fig. 4l). Once they have finalized the layout, they can set the real-world dimensions of each block

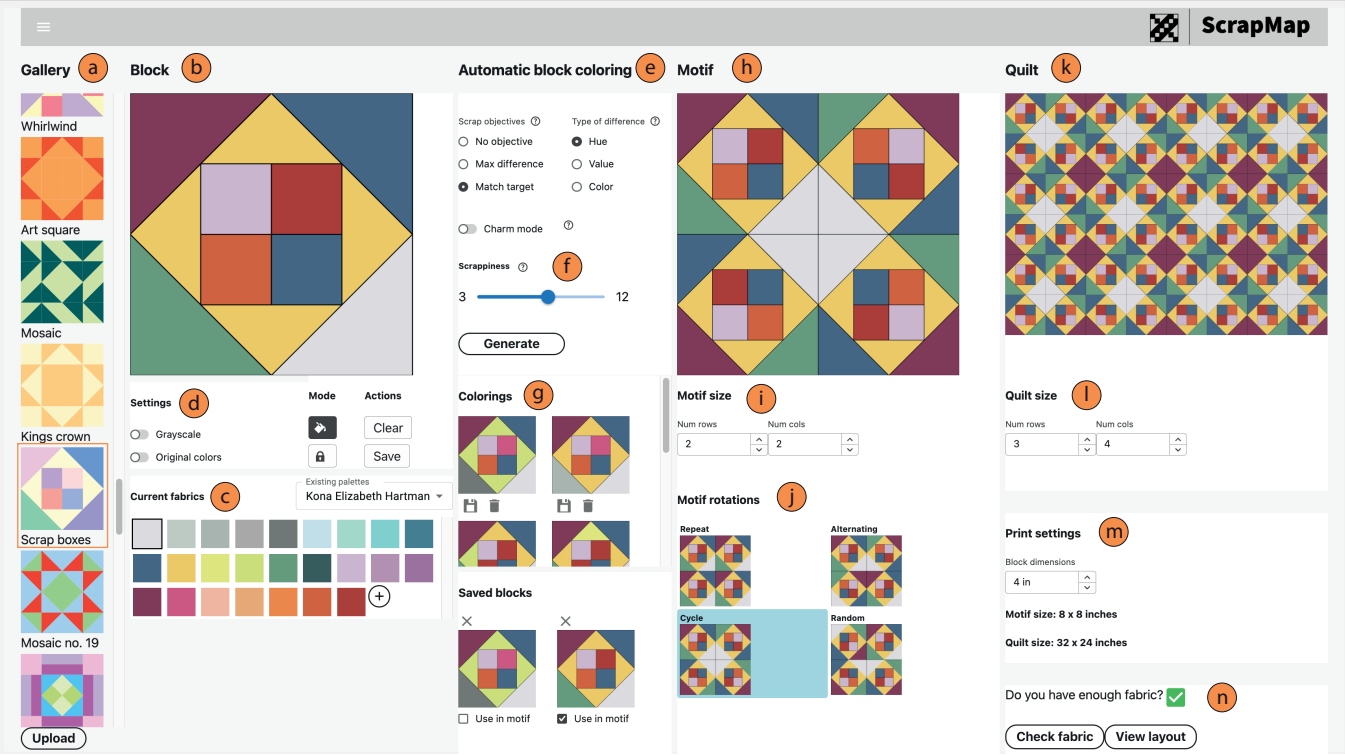


Figure 4: In the ScrapMap UI users begin by selecting a quilt block design or uploading their own in the Gallery Panel (a). In the Block Panel (b) users view the current block coloring. Users select a common fabric color palette or customize their own (c). In the block settings (d) they can view the block in grayscale, view the colors of the original block, manually recolor faces, or lock faces to be a specific color. In the Automatic Block Coloring Panel (e) they can select different coloring objectives and adjust the desired level of scrappiness (f). After clicking "Generate" they can view recommended colorings and save their favorites (g). By checking "Use in motif" they can view the block repeated in the Motif Panel (h). They can adjust the size of the motif (i) and explore different rotations of the blocks within the motif (j). Then, they can view the motif repeated in the Quilt Panel (k). They can adjust the size of the quilt (l), and the dimensions of the blocks (m). Finally, they can check if they have enough fabric for this design and view the fabric cutting layout diagram (n).

and see the corresponding size of each motif and the total quilt (Fig. 4m). They can check if this number of blocks and motifs is possible given our available fabrics by clicking the "Check fabric" button (Fig. 4n) (Sec. 3.7, DG3). If a design is infeasible given the fabric amounts, there are several ways to adjust the design, such as changing the number of blocks, the block size, or how many different fabrics are used (Fig. 6). There is no obvious best way to make this decision automatically so in ScrapMap users can quickly iterate until they reach a visually pleasing solution for which they have enough materials. In this case the user has enough fabric and can view the layout guide, which takes into account seam allowances (i.e., the region of fabric that gets sewn into the seam when fabrics are joined) and shows them how to cut the fabrics efficiently to make this quilt (Fig. 5).

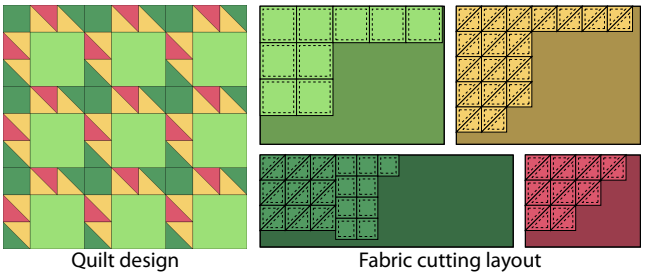


Figure 5: ScrapMap allows users to check if there is enough fabric for the chosen design. If so, it provides a cutting layout. Each polygon represents a piece, and the dashed lines account for the seam allowances, i.e., the region of the fabric that is consumed by the seam when the quilt is sewn.

5 AUTOMATIC BLOCK COLORING

The primary goal — and challenge — of designing a scrap quilt is selecting fabrics for each face of the block. ScrapMap helps users make these decisions through an interactive tool. The Automatic Block Coloring Panel (Fig. 4e) supports the generation of block coloring suggestions corresponding to user-directed parameters.

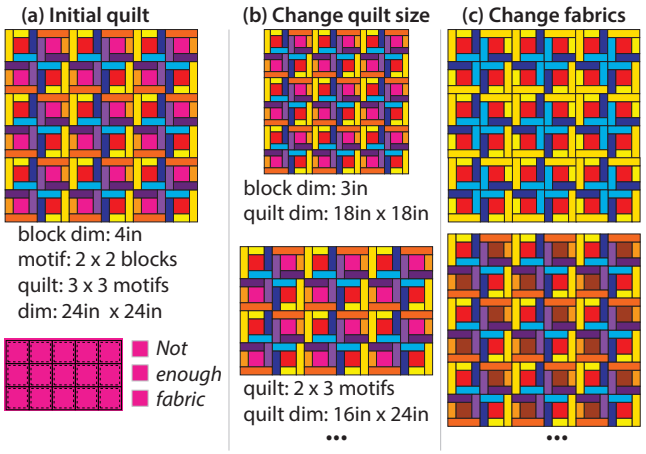


Figure 6: If there is not enough fabric to create a particular quilt, the user has many choices about what to change. We show a few possibilities here. (a) If they do not have enough of the pink fabric, (b) they could change the size of the quilt by reducing the size of each block or the number of blocks or (c) change the fabric choices, among other possibilities. ScrapMap allows users to explore these different options.

Our general approach for generating suggestions is as follows: We allow users to specify high level parameters and objectives corresponding to common requirements and goals discussed in Sec. 3.7. Our tool constructs and solves a constrained optimization problem where solutions represent high quality block coloring suggestions that adhere to the specified requirements and goals. Constraints were designed to guarantee at least one solution. Objective functions were designed so that there are often numerous distinct but equivalently-optimal solutions. The tool returns to the user up to 5 of these solutions. The rest of this section provides a formalization of the optimization problem constructions and important details of the implementation. Further implementation details are provided in Appendix B.

5.1 Problem specification

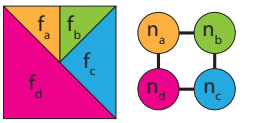
We define a *Quilt Block* as a planar mesh comprising polygonal faces, which correspond to pieces of fabric.

Definition 5.1 (Quilt Block)**.** Let $G = (V, F, x)$ be a planar mesh where V is the set of vertices, F is the set of faces (each specified as a cyclically ordered list of vertices). Every planar mesh implicitly defines a set of edges E, each of which is a pair of subsequent vertices in some face.

We assume that every planar mesh is non-degenerate (no repeated vertices in faces, no vertices outside of faces, no coincident vertices, no overlapping faces, and no holes) and is connected.

Definition 5.2 (Quilt Graph)**.** The dual graph D for planar map G with faces F, vertices V, and edges E is defined as follows: Each face $f \in F$ in G is represented as a node in D (we denote this set of nodes as N). There exists an edge $e \in E'$ between two nodes n_a and n_b in D iff. their corresponding faces f_a and f_b in G share at least one common edge in E (i.e., f_a and f_b are edge adjacent in G).

Note that a dual of a connected planar embedded graph G is always well-defined (up to isomorphism), assuming that G is connected and has a planar embedding.



Definition 5.3 (Fabrics)**.** The set of fabrics is discrete, and only a single fabric may be used for a particular face. We describe each of these fabric choices as *colors*. Let C be the set of unique fabrics available, where each $c \in C$ represents a distinct fabric color.

The automatic coloring task is to assign a color $c \in C$ to each node $n \in N$ in our Quilt Graph D. Let $\mathbf{v} = \{v_n \mid n \in N\}$ be a set of variables where each v_n takes values from the domain C. We denote an assignment to $\mathbf{v}$ as a function $f_\mathbf{v} : N \rightarrow C$, where $f_\mathbf{v}(n)$ corresponds to the assignment of variable v_n. This assignment, $f_\mathbf{v}$, is referred to as a *block coloring*. After the user specifies the objective and parameters in the UI (Sec. 4), our tool generates a corresponding constrained optimization problem using $\mathbf{v}$ as the decision variables. Solving the constrained optimization problem provides a block coloring, $f_\mathbf{v}$, that represents a coloring for the quilt block corresponding to the user's preferences.

5.1.1 Objectives. Our UI supports two objectives, as well as an option to have no objective (Fig. 4f). The first objective is to produce a coloring such that the color difference between all adjacent block faces is maximized (labeled as "Max difference" in the UI). The second objective is to best match a reference block coloring, which means minimizing the color difference between all faces and their corresponding faces in the reference block coloring (labeled as "Match target" in the UI). These objectives correspond to the three scrap quilting goals outlined in Sec. 3.6.

Definition 5.4 (Color Difference)**.** Both objectives rely on a binary function that measures the difference between two arbitrary colors. We expose a parameter $k \in (Hue, Value, Color)$ in the UI that gives the user control over what function, $M_k : (C^*, C^*) \rightarrow \mathbb{R}$ is used. We notate C^* to represent the set of all possible colors. C^* should not be confused with C, which represents the discrete set of colors available to the user. "Hue" only compares the pure color component (e.g., red, blue, or orange). "Value" compares the lightness or darkness of the colors. "Color" takes into account all components of both colors (i.e., the LAB components in CIELAB color space). Each M_k is assumed to be symmetric and computes a low value for very similar colors, and a high value for very different colors. There are many possible choices of functions. ScrapMap uses smallest angle difference for M_{hue}, relative luminance contrast ratio for M_{value} [78], and CIEDE2000 for M_{color} [48].

Definition 5.5 (Objective: Max Difference)**.** Each pair of adjacent block faces are represented by the edges E' of the dual graph. We measure the color differences based on the user's choice of function M_k. To maximize the color difference for all edges (Goal C1 in Sec. 3.6), we use the strategy of maximizing the minimum (worst case) color difference. The full objective is:

$$\underset{\mathbf{v}}{Maximize} \quad \min\{M_k(v_{n_a}, v_{n_b}) \mid n_a, n_b \in E'\}$$

Definition 5.6 (Objective: Match Target)**.** The second objective is to match a reference quilt block (Goal C2 in Sec. 3.6). We represent the target block color assignment as the function $g_{\mathbf{v}} : N- > C^*$. Note that unlike $f_{\mathbf{v}}$, where the values must be one of the fabric colors in C, values in $g_{\mathbf{v}}$ can be any color (i.e., color values that do not appear in the quilter's collection of input fabrics but appear in the original traditional block design), denoted as C^*. The goal for this objective is that for every node n, $f_{\mathbf{v}}(n)$ should match the color $g_{\mathbf{v}}(n)$. We interpret "matching" as minimizing the color difference, $M_k(f_{\mathbf{v}}(n), g_{\mathbf{v}}(n))$. Again, we give the user control over which function M_k is used. To best match all faces, we minimize the maximum (worst case) color difference across all nodes:

$$\underset{\mathbf{v}}{Minimize} \quad \max\{M_k(v_n, g_{\mathbf{v}}(n)) \mid n \in N\}$$

5.1.2 Constraints. There are two constraints that encapsulate what it means for a quilt to be "scrappy" that are applied to all optimization constructions.

Definition 5.7 (Constraint: Graph Coloring)**.** In scrappy constructs, having neighboring faces of the same color is undesirable (Goal C3 in Sec. 3.6). Thus, for all problems, we add the constraint that the coloring assignment must adhere to vertex graph coloring. Our tool requires the number of fabrics to be at least the chromatic number of the dual graph D, $\chi(D)$, to guarantee at least one solution. The constraint is shown in this equation:

$$\phi^{GC}(\mathbf{v}) := \bigwedge_{n_a, n_b \in E'} v_{n_a} \neq v_{n_b}$$

Definition 5.8 (Constraint: Unique Colors, i.e., Scrappiness)**.** "Scrappiness" refers to the variety of different fabrics used. A low level of scrappiness means using only a few fabrics, while a high level of scrappiness involves using many different fabrics.

We specify a second constraint parameterized by a user-given scrappiness level s, shown as the scrappiness slider in the UI (Fig. 4f). This specifies the number of unique colors to be used in the block, or equivalently, the cardinality of the set $\{f_{\mathbf{v}}(n) \mid n \in N\}$. This parameterized constraint is:

$$\phi_s^{scrap}(\mathbf{v}) := \quad |\mathbf{v}| = s$$

The value of s must be at least $\chi(D)$, and at most $\min\{|N|, |C|\}$ for the the constraint to be satisfied. The scrappiness slider's range in the UI is set to these bounds.

Definition 5.9 (Constraint: Fixed Colors)**.** The UI also gives the user the option to "fix" particular faces to manually chosen colors before performing automatic block coloring. Given a set of user specified (node, color) pairs $\mathbf{m} = \{(n_1, c_1), (n_2, c_2), \ldots\}$, we show the constraint below.

$$\phi_{\mathbf{m}}^{fixed}(\mathbf{v}) := \bigwedge_{n,c \in \mathbf{m}} v_n = c$$

Definition 5.10 (Charm Mode)**.** By default, the tool generates multiple different coloring solutions that are distinct, yet optimal for the given constrained optimization problem. Sometimes these solutions can look visually similar (e.g., only one face may differ in color or similar colors are chosen). In order to force more diverse suggestions, we also allow the user to choose if the set of solutions should adhere to "charm" mode. Charm mode, inspired by charm quilts [23] (Sec. 3.4), means that each coloring solution to a given problem uses a completely distinct set of colors from all other solutions. Using this mode along with maximal scrappiness allows users to construct true charm quilts in which every face in every block is different. Let C_{used} be the set of colors already used in previous solutions. We add the following additional constraint to generate the next solution.

$$\phi^{unique}(\mathbf{v}) := \bigwedge_{n \in N} v_n \notin C_{used}$$

We note that each subsequent solution could have a different optimal value in this mode.

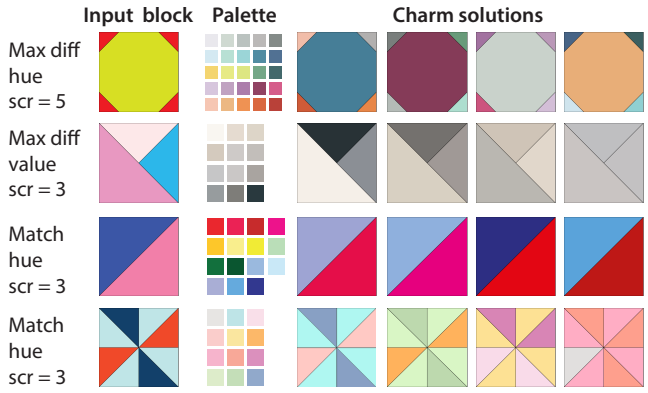


Figure 7: In charm mode ScrapMap returns solutions that each use a distinct set of fabrics from the input palette. We see that when the palette is larger (top row) or offers more colors similar to the input (third row), the top n solutions look good. However, we notice worse solutions for smaller palettes and those without several choices for good matches with the input (second and fourth rows).

5.1.3 Understanding the parameter space. There is a large space of block coloring parameters. To better understand the effect of each parameter choice, it is useful to visualize different parameter choices. Fig. 8 uses a single quilt block and color palette and shows an optimal coloring solution for each objective, color difference function, and a variety of scrappiness values. Fig. 7 shows solutions produced in charm mode for a variety of blocks, palettes, and scrappiness values.

5.2 Implementation

We use Satisfiability Modulo Theories (SMT) [4] to represent the constraints and objectives of the block coloring problem. SMT is a generalization of the boolean satisfiability (SAT) problem that allows for a more expansive set of operations and types to represent complex formulae. All constraints and objectives are encoded using the theory of bitvectors (QF_BV) [24]. More details of the SMT encoding are provided in Appendix B. Given the choice of user parameters, the tool dynamically constructs SMT formulae

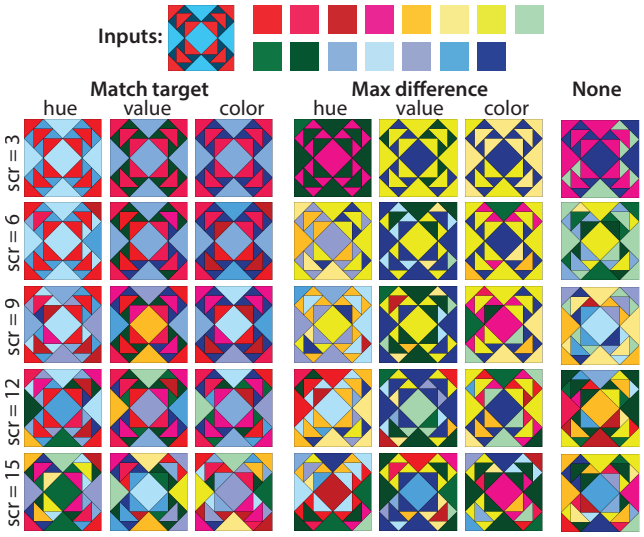


Figure 8: The choice of different objectives and parameters leads to different results, even for the same input block and color palette. Matching the target in terms of hue or overall color, leads to reds and blues often being chosen, since those are the dominant colors in the input. In this case those blue and red fabrics are also similar in value to the input so they are selected for the match value objective as well. For the max difference objective we see higher drama designs, with adjacent faces being assigned colors far apart on the color wheel. As scrappiness increases, we see the number of distinct fabrics in each block increases. Choosing no objective leads to more random looking results while adhering to graph coloring and scrappiness constraints.

for the objective function and each constraint using the hwtypes Python package [18], a QF_BV formula constructor API on top of pySMT [25]. We then use Z3 [17], a popular open source SMT solver available to pySMT via a solving API, to solve the constrained optimization problem. In particular, we leverage Z3's capability of acting not only as a constraint satisfaction solver but also as an optimizing SMT solver [7]. Because each problem is satisfiable by construction, Z3 will return an optimal model encapsulating the assignments to each decision variable $\mathbf{v}$ and the optimal value of the objective function. ScrapMap interprets these assignments as one set of color choices $f_{\mathbf{v}}(n)$, which it gives to the user as a block coloring suggestion. Subsequent solutions are enumerated by re-solving with additional constraints enforcing the known optimal objective function value and precluding previous assignments.

5.2.1 Verifying Fabric Amounts. Once a full quilt is colored to the user's satisfaction, the user can verify that the amount of fabric on hand is enough to create the quilt. The user can also download a SVG depicting how to cut the fabrics to realize the design in the physical world (Fig. 5). As is standard in quilting, we assume that each face requires a *seam allowance* (i.e., the region of fabric that is consumed in the seam when fabrics are sewn together) of 0.25 inches. We conservatively interpret this verification as a standard rectangle packing problem commonly used in other domains, such as VLSI layout [26]. First, we pack all pairs of same-size right triangles into rectangles, and then we take the rectangular bounding box of all other face shapes. We then use the open source rectpack tool [65] to pack all rectangles into the associated fabric rectangles. This tool uses greedy packing techniques [28, 33] and is fast (i.e., all examples complete in less than 1 second). ScrapMap allows and handles multiple distinct fabrics of the same color and will use all pieces for packing.

6 RESULTS

We have applied our technique to a variety of input blocks and color palettes. We explored the potential for ScrapMap to be used by quilters in our user evaluation (Sec. 6.1) and the interaction between recoloring parameters in our technical evaluation (Sec. 6.2).

6.1 User evaluation

In our user evaluation we focused on learning about how quilters would use ScrapMap to plan scrap quilts.

ID	Exp (yrs)	# Scrap quilts	Design SW	Quilt SW	Stash	Help	Fun	Diff.	Use scraps
P1	10	One	Yes	No	100+	5	5	4	5
P2	3	Many	No	No	11-100	5	5	4	4
P3	23	Many	No	No	100+	5	5	4	5
P4	1	One	Yes	No	11-100	4	5	5	5
P5	15	Many	Yes	Yes	<10	5	5	5	5

Table 1: Participants (P1-P5) varied widely in experience quilting, using design software (e.g., Adobe Illustrator and Procreate) and quilting software (e.g., ElectricQuilt), and the amount of fabrics they have on hand (stash). Overall participants found ScrapMap very helpful for designing scrap quilts (Help), fun to use (Fun), easy to use (Diff), and likely to encourage them to use scraps in the future (use scraps) (all on a 1-5 Likert scale, with 1=not at all to 5=very much).

6.1.1 Study design & format. We recruited 5 quilters, all of whom had experience making at least one scrap quilt. Participants (all women, aged 23-46) had 1-23 years of quilting experience (Table 1). Our study protocol underwent an organizational approval process, and our participants were compensated $30. All participants completed a 50-minute virtual study session. We first asked participants to discuss their quilting background and then demonstrated ScrapMap. Participants then designed at least 4 quilts each. Each participant colored ("Old windmill" and "Flock of geese") using pre-selected color palettes ("Kona Summer" and "Kona Winter" [68], respectively). For the third block ("Bear paw"), we allowed participants to use any colors they would like, either from one of the default fabric palettes or a custom one from their own fabric stash (Fig. 9). After completing the first three quilts, participants were free to choose any of the 70 available traditional blocks and any fabric colors for subsequent quilts. For every quilt participants could choose any combination of objectives and parameters, use any number of fabrics, and generate as many possibilities as they would like. They iterated for 4-12 minutes on each quilt until they found ones they would like to sew.

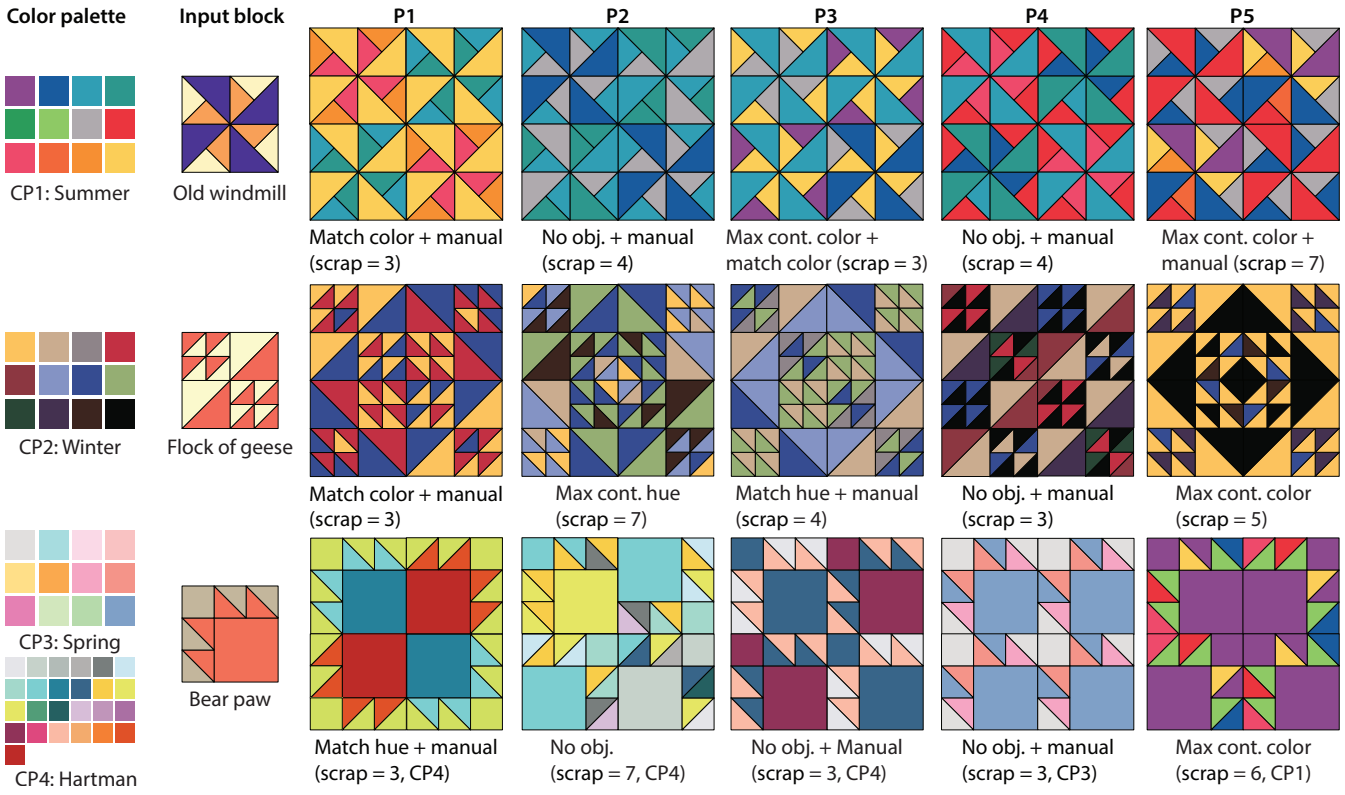


Figure 9: In the first part of our user study, 5 participants (P1-P5) created three of the same blocks before exploring other blocks on their own. For the "Old windmill" and "Flock of geese" blocks, they were required to use the specified palette (CP1 and CP2, respectively). For the "Bear paw" block, they could use any color palette. Participants each explored a wide range of different coloring objectives, scrappiness values, and motif layouts, leading to different quilts.

6.1.2 Participant workflows & outputs. Even with the same input blocks and color palettes (Fig. 9, top two rows), participants created a range of quilts with different objectives and motif grid layouts.

For the "Old windmill" block, all participants ended up with two colorings they liked using different objectives and scrappiness values (Fig. 9, top row). They then alternated these two results into a repeated grid motif. P1, P2, P4, and P5 all used some manual coloring to customize the suggested automatic coloring.

For the "Flock of geese" block, participants created very different visual effects by adjusting the scrappiness value and rotations of the blocks within the motifs (Fig. 9, middle row). P5 made a particularly dramatic block by maximizing the contrast in color.

For the "Bear paw" block, participants also ended up with very different colorings and layouts (Fig. 9, bottom row). P1 started with the suggested orange and green block, which was similar in hue to the input. They then manually created the blue and green blocks to alternate with the suggested coloring. P2 started to create a block with only 2 colors. They then decided to increase the scrappiness and try using no objective to get a scrappier look. They ended up liking this automatic coloring that was more varied. P3 and P4 also tried the no objective option to diversify the colors they were considering for the block but ultimately did some manual recoloring to return to a more predictable pattern.

P5 explored a wide range of options for the "Bear paw" block (Fig. 10). They started with drawing a simple 2-color block with yellow and orange (Fig. 10, step 1). They then created a large quilt using this single block repeated (6 x 4 block motif, 3 x 4 motif quilt) (Fig. 10, step 2). While they liked this "sunflower" appearance, they discovered that they did not have enough fabric to create it. They then replaced the large orange square with a darker orange fabric (Fig. 10, step 3). While this solved their fabric shortage problem, they also took the opportunity to explore another coloring option. They used the max contrast color option to create a teal and red block (Fig. 10, step 4). They mixed this block with the prior one they had created in a repeat and then chose the random motif grid (Fig. 10, steps 5-6). Before settling on this design, they decided to try one more option. They increased scrappiness and selected max contrast color (Fig. 10, step 7). This gave them a very different appearance, which they found much more visually dynamic. They explored all of the grid options, before finalizing their quilt with the alternating grid (Fig. 10, steps 8 & 9). They ended up with a very different design than they had originally intended, but they said that they were much happier with the dramatic final result.

For their final design P2 decided to work with the "Toad in a puddle" block and the autumn color palette [68](Fig. 11). To start their design, they began with the no objective option with low

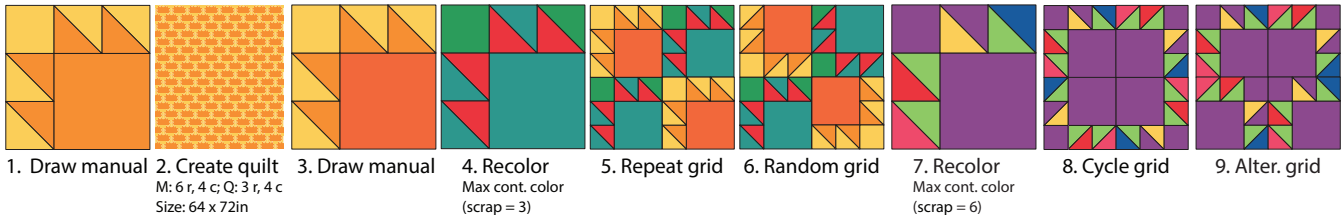


Figure 10: P5 started by filling in the colors of each face of the block manually with yellow and light orange (1). When they created a motif with 6 rows and 4 columns of blocks and then a quilt with 3 rows and 4 columns of motifs (288 total blocks, each 4in), they discovered that they did not have enough fabric to create this design (2). Then, they recolored one of the faces to a darker orange, which was feasible with the available fabrics (3). Before settling on this design, they tried recoloring the block using the max color difference constraint with minimal scrappiness (4). They then tried arranging this block in combination with the prior one in a repeated grid (5) as well as the random orientation grid (6). Before finalizing their design, they explored more choices with higher scrappiness (7) in two different grid configuration (8, 9).

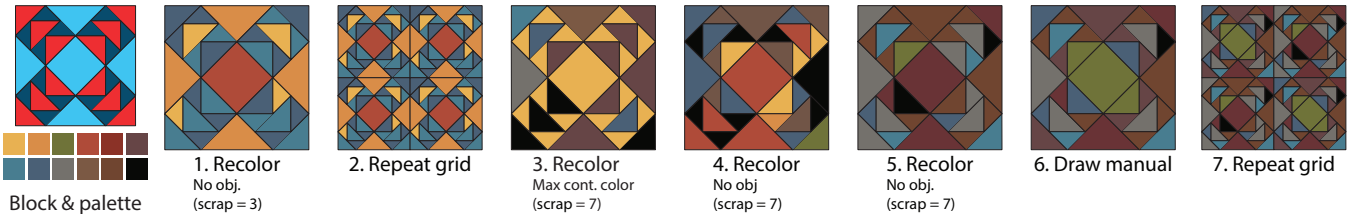


Figure 11: P2 started with a bright block with high contrast, but wanted to create a design with a more muted autumn palette. They started using no objective with scrappiness set to a small value (1). While they initially liked the design in the repeated grid (2), they decided to explore another objective that would produce more contrast and feel more scrappy (3). They then kept this higher scrappiness value but wanted something with more randomness so they chose no objective (4, 5). Then, they updated a few faces to produce a stronger secondary pattern (6) and repeated this block with the prior version in their motif (7).

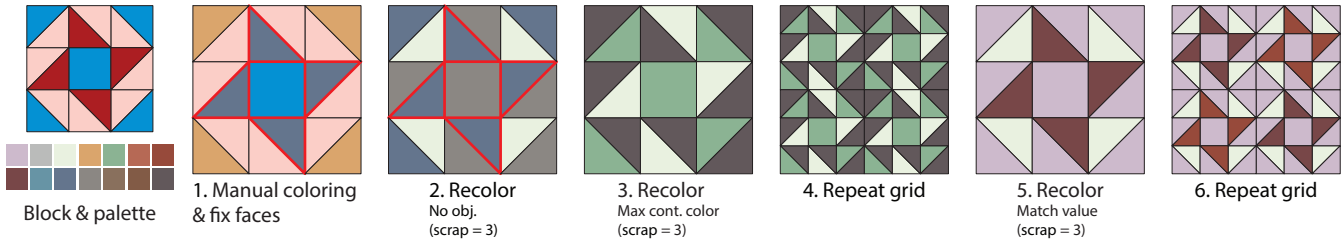


Figure 12: P4 started with a block with a distinctive red star and used fabrics from their own stash. They manually colored the corners yellow and fixed the four star points to be blue (1). They then generated an automatic coloring using no objective and a low scrappiness value (2). They then released the fixed faces and explored a very different, more varied design using the max contrast in color objective (3, 4). Returning to a simpler design, they used "match value" to get the star pattern to return (5, 6).

scrappiness to get some initial inspiration (Fig. 11, step 1). While they liked this initial design in the repeated grid as is (Fig. 11, step 2), they decided to explore another objective that would produce more contrast and feel more scrappy (Fig. 11, step 3). They liked the first two suggested colorings (Fig. 11, steps 4-5). They then manually recolored a few faces in the automatic coloring to produce a stronger secondary pattern (Fig. 11, step 6) and then repeated two similar blocks (Fig. 11, step 7).

P4 decided to work from the "Red star" block and recolor the design using the fabric colors and amounts in their own personal stash. They started by manually recoloring the outer corners gold and the points of the star blue (Fig. 12, step 1). They said that they wanted to create a blue star so they locked these particular faces in the UI (red outline). With these fixed faces, they selected the no objective option to get an initial design with the blue pointed star (Fig. 12, step 2). While they liked this initial design, they wanted to see a design with more contrast. They unlocked the faces and used the max contrast color objective (Fig. 12, step 3). They did not like that this design lost the repeated pattern in the input with the four same color points in the star (Fig. 12, steps 3 & 4). They then

tried the match value option to see if they could get back to the type of contrast present in the star points in the input block (Fig. 12, step 5). They ended up liking the first two suggested results, which featured a purple background and different shades of brown points (Fig. 12, step 6).

During and after exploring different block colorings, participants gave feedback on the overall experience of using ScrapMap to design scrap quilts, which we discuss below.

6.1.3 Taking inspiration from automatic suggestions. Participants saw value in exploring different ways they could potentially use the fabrics in their stashes. P1, who has a large stash, said, *"[Using ScrapMap] I could actually visualize what I'm doing with them [my scraps]. I think it would be handy for scraps that I like and even those I don't necessarily love but have in my stash. But if I could see them in a pattern and they were actually working, that would make me more likely to use them."* P2, a less experienced quilter, encountered a color palette (CP2) for the "Flock of geese" block in the study that included colors they do not often use (Fig. 9). However, they still found value in seeing alternatives, rather than just seeing a single set of fabrics chosen by the pattern author. ScrapMap allows users to see new and unusual combinations of fabrics, which they may not have previously considered or seen in published quilt patterns.

6.1.4 Experimenting with fabric colors and placements. Participants, particularly those with less quilt design experience, liked trying different possibilities. P3, who often sticks to a limited set of colors, said that ScrapMap could help them try new colors and blocks: *"This would definitely have me try things that I would have never tried ever in my life, just because of a fear of how it would turn out because you just don't know the end result."* P2 said that ScrapMap *"would show me different ways to do that and just increase my confidence in quilting knowing what the final result is going to look like and being confident that it's actually going to look good at the end."* Being able to experiment with fabrics digitally allows users to explore several options before cutting into their fabrics.

6.1.5 Designing quilts hierarchically. P1 and P2 were particularly fond of the intermediate step of trying out different motif sizes and rotations. After working hard on several individual blocks, P1 said, *"Motif changing was helpful to visualize and to help think of different ways that you could put it all together."* P2 said that the motif step was *"so helpful because trying to do that mental rotation in your head or even with your various fabrics on the floor is so difficult."* Participants noted this complexity of managing not only the placement of each piece of fabric but also how the individual blocks interact in these larger motif and quilt patterns.

6.1.6 Encouraging future scrap use. All participants indicated that using ScrapMap would make them far more likely to use their scraps in the future (Table 1, median=5/5). P5, an experienced quilter, shared, *"I don't want to see them [scraps] as a waste of material. It's expensive to get materials and to use all of that. And to have have ScrapMap – not only is it good for the environment, it's actually good to get more for your money and actually get that joyfulness without having to worry too much. It kind of takes away the worrying of it because you can put in what you have and it spits out the layout for you, easy peasy."* P4, a beginning quilter, who often makes smaller projects, such as coasters, with leftover fabrics, said, *"Before I felt like it would just take so much time to plan out and coordinate everything, but I could put the colors I want to buy or use or get from the thrift store in here… it'll tell me basically whether that's enough fabric. It makes it much more attainable to want to make something bigger than what I traditionally do."*

6.2 Technical evaluation

In this section, we evaluate the performance of the automatic block coloring. We explore the effect of each block coloring objective and parameter on solver performance. As discussed in Sec. 5, the four main components of the constrained optimization problem are:

- **Block**: A block is characterized by number of nodes and edges in the dual graph. We use all 70 blocks available in the UI (Sec. 4.1) [58, 61, 75].
- **Color palette**: We use 5 different color palettes constructed by randomly sampling 5, 10, 15, 20, and 25 colors from the set of 140 named CSS colors [74].
- **Objective and color difference function**: There are 6 combinations of objectives and color difference functions along with a 7th "no objective."
- **Scrappiness**: We use three scrappiness values, (low, middle, high), where the exact values are determined by the block graph and number of fabrics.

For each of the 7350 combinations of parameters (70 blocks x 5 color palettes x 7 objectives and color difference functions, x 3 scrappiness values), we run the automatic block coloring with a timeout of 60 seconds. We measure the total time including both problem construction and solve time. We note the average solve time across 2 independent trials [1] and whether any of the trials timed out. All experiments were performed on a 2021 Apple MacBook Pro M1 Max with 32GB of RAM. 7152 of the total 7350 parameterizations solved in under 60 seconds with a median time of 0.6 seconds.

6.2.1 Objectives. We first analyze the overall performance for each objective. Subsequent sections discuss performance trends for particular automatic coloring parameters. As all automatic coloring problems are satisfiable by construction, formula size analysis can potentially be used as a rough model for performance expectations. The "no objective," "max difference," and "match target" constraint sizes scale according to $|C| \cdot |N|$, $|C|^2 \cdot |E'|$ and $|C| \cdot |N|$, respectively, due to the encodings discussed in Appendix B.

In Fig. 13 we categorize all automatic coloring problem instances into the 7 objectives and sort the times from low to high. The figure shows the number of successful solves for each objective in this benchmark. "No objective" successfully solves every parameterization instance in under 7 seconds. All of the "max difference" color difference functions have similar performance, and successfully solve at least 94% instances under 60 seconds and over 50% under 2 seconds. Similarly, all of the "match target" color difference functions have similar performance and solve at least 98% in under 60 seconds with 80% under 2 seconds.

6.2.2 Block Complexity. We characterize each block's complexity by the number of nodes, $|N|$. For each block, we plot $|N|$ versus the median solve time across all palettes, all scrappiness levels, and all color difference functions. This is shown in Fig. 14 and indicates

[1] Solver execution time varies from run to run.

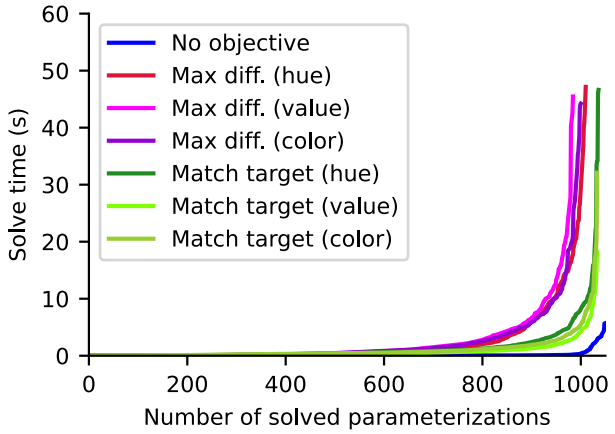


Figure 13: This graph illustrates the time, measured in seconds, required to solve each parameterization within the automatic color problem set. All 1050 "no objective" parameterizations are solved under 7 seconds. For "max difference," the solve rates are 96%, 94%, and 95% for hue, value, and color, respectively. "Match target" achieves solve rates of 99%, 98%, and 98% for hue, value, and color, respectively.

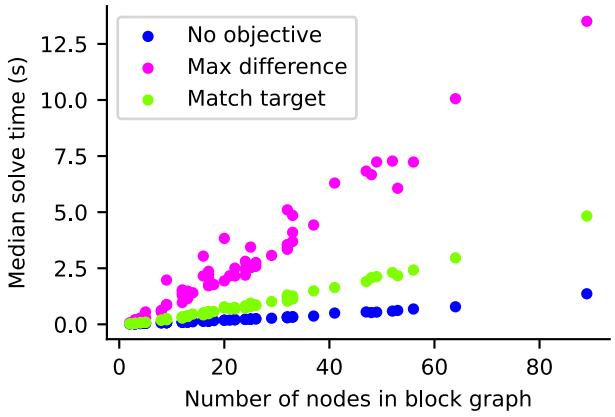


Figure 14: This graph shows the relationship between number of nodes in the block graph and median solve time for the different objectives. For all objectives, median time increases roughly linear with the number of nodes.

a linear relation between the number of nodes and median solve time.

6.2.3 Palette size. We measure the effect of palette size on performance. For each palette size (5, 10, 15, 20, 25), we calculate the median runtime and percentage of problems solved across all blocks, all scrappiness levels, and all color difference functions. This is shown in Table 2. The median time increases as the number of colors increases. Similarly, the success rate decreases as the number of colors increases. This is consistent with the formula scaling.

		Number of colors				
		5	**10**	**15**	**20**	**25**
No obj.	**Median (s)**	<0.1	0.1	0.2	0.2	0.3
	Solved (%)	100	100	100	100	100
Max diff.	**Median (s)**	0.2	0.8	2.2	4.6	7.8
	Solved (%)	100	100	97	94	85
Match target	**Median (s)**	0.2	0.4	0.6	0.9	1.2
	Solved (%)	100	100	100	98	95

Table 2: Each column represents a color palette of a different size. We show the median time measured in seconds and the solving success rate for each objective. We see that as the size of the color palette increases, the median time increases and the solve rate decreases. We see quadratic scaling of median solve time for "Max diff." and linear scaling for "Match target." This is consistent with the formula size scaling.

		Scrappiness		
		Low	**Middle**	**High**
No obj.	**Median (s)**	0.1	0.1	0.1
	Solved (%)	100	100	100
Max diff.	**Median (s)**	1.4	1.7	2.0
	Solved (%)	100	99	86
Match target	**Median (s)**	0.6	0.5	0.6
	Solved (%)	100	100	96

Table 3: Each column represents a different scrappiness level. We show the median time measured in seconds and the solving success rate for each objective. We see high scrappiness causes the vast majority of timeouts.

6.2.4 Scrappiness. We measure the effect of scrappiness on performance. For each scrappiness level, (low, middle, high), we calculate the median runtime and percentage of problems solved across all blocks, color palettes, and color difference functions shown in Table 3. We find the solves take longer the higher the scrappy value is. Cardinality constraints like ϕ_s^{scrap} are often difficult for solvers to handle efficiently. Overall, scrappiness appears to be a major factor in determining a successful solve, whereas increasing block complexity and size of palette leads to increased median times.

7 DISCUSSION

While ScrapMap focuses specifically on the domain of scrap quilting, many of the insights from the design and evaluation of our system speak to broader challenges in creating computational design tools.

7.1 Physical constraints and design exploration

Many of the barriers to the scrap quilting design process stem from the large space of possible decisions and material limitations. Allowing for digital exploration goes a long way in encouraging scrap reuse. Three of the most highly rated features of ScrapMap were being able to explore different coloring possibilities for a given block, construct different motifs digitally, and get a fabric cutting layout to ensure there is enough material. Simply seeing the different design possibilities allowed users to consider colors

and designs they may not have considered previously. P1 and P2 specifically mentioned the challenges of designing larger quilts in limited physical spaces. P2 noted the importance of doing this design and layout work digitally, saying: *"[In ScrapMap] I can really look at and find the layouts that I like without having to physically move fabrics around, which is always nice because I, like many other people who like to quilt, might not have that large of a space."*

An important consideration in any digital design tool for fabrication is how these tools can accurately reflect the realities of the physical world while encouraging design exploration within the constraints. Participants also appreciated the fabric layout guidance and knowing when a design was infeasible. When P5's initial design was infeasible (Fig. 10), reconsidering different fabric scraps led them to a very different visual outcome than they had first imagined, but they ended up liking this design better in the end. This speaks to the possibility of viewing these physical realities not as practical considerations that hinder design, but rather as ways to encourage creative exploration.

7.2 Supporting existing workflows

Early versions of ScrapMap incorporated fabric usage constraints into the color placement optimizations, but feedback from an experienced quilter indicated these should be treated separately. The size and number of blocks is part of the exploration process and are not fixed at the start of the design process. If a quilt the user generates is infeasible given the fabric amounts, there are several ways to adjust their design, such as changing the number of blocks, the block size, or how many different fabrics are used (Fig. 6), and there is no obvious best way to make this decision automatically. We therefore allow the user to explore these different ways of navigating the material limitations digitally. While in this domain it made sense to consider the colorings ahead of the material amount limitations, one could certainly imagine other fabrication workflows in which these sizing considerations should come into play earlier in the design process.

7.3 Blending automatic and manual recoloring

All participants in our user evaluation used a combination of the automatic coloring suggestions and manual block coloring at different points in their design process. We often found that participants started by manually coloring at least some faces in their early designs. However, as they started using ScrapMap more, they became more reliant on the automatic recoloring suggestions, as they gained more trust in the system and a better sense of the controls. Scrap quilting is often a deeply personal process; sometimes the fabric scraps have sentimental value. There is a careful balance in making automatic recommendations to the user and allowing them to personalize their designs. In ScrapMap we allow users to begin with manual inputs or use the manual coloring feature to alter suggested colorings. This allows them to receive guidance while adding a personal touch at any point in the design process.

7.4 Surprise and exploration in design tools

While our system is designed to suggest colorings that are optimal given the input parameters and objectives, there is a great deal of personal taste in evaluating the outcomes. The suggested automatic colorings will display the desired level of scrappiness, high contrast, or similar appearance to the input design, which are common goals for quilters (Sec. 3.6). We expose these different parameters in the UI to give users the opportunity to explore a small part of the enormous design space of possible scrap quilts. While we found that some participants gravitated toward similar parameter and objective selections for different blocks, indicating some potential for learning good combinations of settings, others appreciated exploring different possibilities and trying out different settings for different blocks and fabric color palettes. In design tools with these types of controls, there can be good and bad surprises. Good surprises present usable results to the user that maybe differ from their typical taste; bad surprises are those designs that do not match what users expected to see given the choice of input parameters. We found that participants were surprised in a good way about the suggested colorings. Many, such as P5 (Fig. 10), ended up with a very different design than where they started.

8 LIMITATIONS & FUTURE WORK

While ScrapMap enables exploration of scrap quilt designs in realistic scenarios, there are a few limitations and opportunities for future work.

8.1 Performance

We used SMT and Z3 to encode and solve the constrained optimization problem. However, other choices of encoding are possible for the problem specifications in Sec. 5.1. In particular, encoding and solving using Mixed Integer Linear Programs (MILP) are likely possible. However, using a complex color difference function like CIEDE2000 may require non-linear solvers or encoding tricks similar to those described in Appendix B. Our constraint construction uses Python libraries (i.e., [19, 25]) which account for a non-negligible portion of the total automatic coloring time. This could be optimized. Despite our chosen approach working well for a range of palette sizes and quilt block complexities, high scrappiness is a main factor for causing timeouts. Scrappiness ranges, soft constraints, or iterative techniques may be able to be used instead.

8.2 Domain-specific graph coloring

Graph coloring is a broadly applicable technique with many implications in visual domains. While many domains utilize continuous color spaces, graph coloring offers a useful solution for problems where discrete coloring is necessary. We focus on the constraints that scrap quilting imposes on the graph, but there are many other domains in which different constraints could be used to provide similar coloring guidance. For example, designing knitted colorwork or stained glass panels may present similar opportunities for suggesting material-aware colorings that preserve desired patterns.

8.3 Computational support for scrap quilting

While our characterization of scrap quilting as a constrained coloring problem factors in many important features of this domain, there are opportunities for further exploring other aspects of scrap quilt design. For example, we only consider solid color fabrics explicitly, although quilters could choose the dominant color in the printed fabric for visualization in the current tool. Future work

could explore directly supporting printed, multi-color, and directional fabrics by taking into account orientation and other attributes in the fabric selection, automatic coloring, and fabric layout panels. Similarly, future work could explore the role of symmetry in creating high contrast quilts; users were often excited to see secondary patterns emerge in the motifs. We could also imagine supporting different methods of input for digitizing and organizing scraps using off-the-shelf color sensors and scanning methods for the fabrics. Furthermore, a future version of the tool could help the user better understand the current material constraint violations (e.g., visualizing the extra fabric required for each color), and it could provide a few possible suggestions to help make the quilt realizable. Some of these suggestions could come with computational support (e.g., a button that automatically adjusts the block dimensions while keeping all other parameters fixed).

8.4 Promoting fabric reuse

ScrapMap presents an opportunity for users to envision creative uses for their leftover materials. Simply seeing the possibilities of one's available fabric can encourage quilters to make something out of materials that might otherwise be wasted. There are many opportunities for future work at the intersection of design and creative material reuse. Altering the geometry of a given design to promote more efficient layouts and material use, for example, could allow for even further reuse of scrap fabric.

9 CONCLUSION

Despite the often simple geometry of individual blocks, the mix of different fabrics leads to striking visual patterns in scrap quilts. Navigating this extremely large space of different design possibilities can be overwhelming. ScrapMap helps quilters focus on a smaller part of this design space which has the properties quilters care about most: contrast, similarity to traditional designs, and scrappiness. It is our hope that tools like ScrapMap can help quilters realize the potential of their leftover materials and create beautiful heirlooms.

REFERENCES

[1] Ignasi Abío, Robert Nieuwenhuis, Albert Oliveras, and Enric Rodríguez-Carbonell. 2013. A parametric approach for smaller and better encodings of cardinality constraints. In *International Conference on Principles and Practice of Constraint Programming*. Springer, 80–96.

[2] Craft Industry Alliance. 2022. Today's Quilting Trends. https://cialliance.wpenginepowered.com/wp-content/uploads/2022/11/2022_TodaysQuiltingTrends-1.pdf

[3] Anton Bakker and Tom Verhoeff. 2022. Algorithms to Construct Designs for Foundation Paper Piecing of Quilt Patchwork Layers. In *25th Annual Bridges Conference 2022: Mathematics, Art, Music, Architecture, Cullture*. Tessellations Publishing, 347–350.

[4] Clark Barrett and Cesare Tinelli. 2018. *Satisfiability modulo theories*. Springer.

[5] 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 (2016), 50–1.

[6] Floraine Berthouzoz, Akash Garg, Danny M Kaufman, Eitan Grinspun, and Maneesh Agrawala. 2013. Parsing sewing patterns into 3D garments. *Acm Transactions on Graphics (TOG)* 32, 4 (2013), 1–12.

[7] Nikolaj Bjørner, Anh-Dung Phan, and Lars Fleckenstein. 2015. *vz*-an optimizing SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems: 21st International Conference, TACAS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015, Proceedings 21*. Springer, 194–199.

[8] Remi Brouet, Alla Sheffer, Laurence Boissieux, and Marie-Paule Cani. 2012. Design preserving garment transfer. *ACM Transactions on Graphics* 31, 4 (2012), Article–No.

[9] Lynn Roddy Brown. 2009. *Simple strategies for block-swap quilts*. That Patchwork Place (Martingale).

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

[11] Huiwen Chang, Ohad Fried, Yiming Liu, Stephen DiVerdi, and Adam Finkelstein. 2015. Palette-based photo recoloring. *ACM Trans. Graph.* 34, 4 (2015), 139–1.

[12] Cheng-Kang Ted Chao, Jason Klein, Jianchao Tan, Jose Echevarria, and Yotam Gingold. 2023. ColorfulCurves: Palette-Aware Lightness Control and Color Editing via Sparse Optimization. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–12.

[13] Charlotte Clark and Rose Bohrer. 2023. Homotopy Type Theory for Sewn Quilts. In *Proceedings of the 11th ACM SIGPLAN International Workshop on Functional Art, Music, Modelling, and Design*. 32–43.

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

[15] Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, and Ying-Qing Xu. 2006. Color harmonization. In *ACM SIGGRAPH 2006 Papers*. 624–630.

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

[17] Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: An efficient SMT solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 337–340.

[18] Caleb Donovick, Ross Daly, Jackson Melchert, Lenny Truong, Priyanka Raina, Pat Hanrahan, and Clark Barrett. 2023. PEak: A Single Source of Truth for Hardware Design and Verification. *arXiv preprint arXiv:2308.13106* (2023).

[19] Caleb Donovick, Ross Daly, Jackson Melchert, Lenny Truong, Priyanka Raina, Pat Hanrahan, and Clark Barrett. 2023. PEak: A Single Source of Truth for Hardware Design and Verification. *arXiv preprint arXiv:2308.13106* (2023).

[20] Zheng-Jun Du, Kai-Xiang Lei, Kun Xu, Jianchao Tan, and Yotam Gingold. 2021. Video recoloring via spatial-temporal geometric palettes. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–16.

[21] Tamara Anna Efrat, Moran Mizrahi, and Amit Zoran. 2016. The hybrid bricolage: bridging parametric design with craft through algorithmic modularity. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 5984–5995.

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

[23] Judy Florence. 1995. *Scrap Quilts and How to Make Them*. Courier Corporation.

[24] Anders Franzén. 2010. *Efficient solving of the satisfiability modulo bit-vectors problem and some extensions to SMT*. Ph. D. Dissertation. University of Trento.

[25] Marco Gario and Andrea Micheli. 2015. PySMT: a solver-agnostic library for fast prototyping of SMT-based algorithms. In *SMT workshop*, Vol. 2015.

[26] Dorit S Hochbaum and Wolfgang Maass. 1985. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM (JACM)* 32, 1 (1985), 130–136.

[27] 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*. 5–16.

[28] Eric Huang and Richard E Korf. 2013. Optimal rectangle packing: An absolute placement approach. *Journal of Artificial Intelligence Research* 46 (2013), 47–87.

[29] 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.

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

[31] Tommy R Jensen and Bjarne Toft. 2011. *Graph coloring problems*. John Wiley & Sons.

[32] Benjamin Jones, Yuxuan Mei, Haisen Zhao, Taylor Gotfrid, Jennifer Mankoff, and Adriana Schulz. 2021. Computational design of knit templates. *ACM Transactions on Graphics (TOG)* 41, 2 (2021), 1–16.

[33] Jukka Jylänki. 2010. A thousand ways to pack the bin-a practical approach to two-dimensional rectangle bin packing. *retrived from http://clb. demon. fi/files/RectangleBinPack. pdf* (2010).

[34] 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.

[35] Suzi Kim and Sunghee Choi. 2021. Dynamic closest color warping to sort and compare palettes. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–15.

[36] Bongjin Koo, Jean Hergel, Sylvain Lefebvre, and Niloy J Mitra. 2016. Towards zero-waste furniture design. *IEEE transactions on visualization and computer graphics* 23, 12 (2016), 2627–2640.

[37] Maria Korosteleva and Sung-Hee Lee. 2022. Neuraltailor: Reconstructing sewing pattern structures from 3d point clouds of garments. *ACM Transactions on*

Graphics (TOG) 41, 4 (2022), 1–16.
[38] Maria Korosteleva and Olga Sorkine-Hornung. 2023. GarmentCode: Programming Parametric Sewing Patterns. *arXiv preprint arXiv:2306.03642* (2023).
[39] Maria Larsson, Hironori Yoshida, and Takeo Igarashi. 2019. Human-in-the-loop fabrication of 3D surfaces with natural tree branches. In *Proceedings of the 3rd Annual ACM Symposium on Computational Fabrication*. 1–12.
[40] Mackenzie Leake, Gilbert Bernstein, and Maneesh Agrawala. 2022. Sketch-Based Design of Foundation Paper Pieceable Quilts. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–11.
[41] Mackenzie Leake, Gilbert Bernstein, Abe Davis, and Maneesh Agrawala. 2021. A mathematical foundation for foundation paper pieceable quilts. *ACM Trans. Graph.* 40, 4 (2021), 65–1.
[42] Mackenzie Leake, Kathryn Jin, Abe Davis, and Stefanie Mueller. 2023. InStitches: Augmenting Sewing Patterns with Personalized Material-Efficient Practice. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.
[43] Mackenzie Leake, Frances Lai, Tovi Grossman, Daniel Wigdor, and Ben Lafreniere. 2021. PatchProv: Supporting Improvisational Design Practices for Modern Quilting. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–17.
[44] Yifei Li, David E Breen, James McCann, and Jessica K Hodgins. 2019. Algorithmic Quilting Pattern Generation for Pieced Quilts.. In *Graphics Interface*. 13–1.
[45] Max Limper, Nicholas Vining, and Alla Sheffer. 2018. Box cutter: atlas refinement for efficient packing via void elimination. *ACM Trans. Graph.* 37, 4 (2018), 153–1.
[46] Sharon Lin, Daniel Ritchie, Matthew Fisher, and Pat Hanrahan. 2013. Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–12.
[47] 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*. 1–8.
[48] M Ronnier Luo, Guihua Cui, and Bryan Rigg. 2001. The development of the CIE 2000 colour-difference formula: CIEDE2000. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur* 26, 5 (2001), 340–350.
[49] Judy Martin. 1985. *Scrap Quilts*. Moon Over the Mountain.
[50] Judy Martin. 2009. *Scraps*. That Patchwork Place (Martingale).
[51] David W Matula, George Marble, and Joel D Isaacson. 1972. Graph coloring algorithms. In *Graph theory and computing*. Elsevier, 109–122.
[52] 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 (TOG)* 35, 4 (2016), 1–11.
[53] Yuki Mori and Takeo Igarashi. 2007. Plushie: an interactive design system for plush toys. In *ACM SIGGRAPH 2007 papers*. 45–es.
[54] 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.
[55] MH Noaman, H Khaled, and HM Faheem. 2022. Image colorization: A survey of methodolgies and techniques. In *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2021*. Springer, 115–130.
[56] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2014. Collaborative filtering of color aesthetics. In *Proceedings of the Workshop on Computational Aesthetics*. 33–40.
[57] Patsy Orlofsky and Myron Orlofsky. 1974. *Quilts in America*. McGraw-Hill.
[58] American Patchwork and Quilting. 2022. Free Quilt Block Patterns. https://www.allpeoplequilt.com/
[59] Nico Pietroni, Corentin Dumery, Raphael Falque, Mark Liu, Teresa Vidal-Calleja, and Olga Sorkine-Hornung. 2022. Computational pattern making from 3D garment models. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–14.
[60] PreQuilt. 2020. PreQuilt. https://prequilt.com/
[61] C&T Publishing. 2021. *Quilt Builder Card Deck: 40 Block, 6 Layouts, Endless Possibilities*. C&T.
[62] Quiltography. 2016. Quiltography. https://www.quiltography.co.uk/
[63] Quiltster. 2020. Quiltster | Digital Quilt Planner. https://www.quiltster.com/
[64] Jing Ren, Aviv Segall, and Olga Sorkine-Hornung. 2023. Digital 3D Smocking Design. *ACM Transactions on Graphics* (2023).
[65] Secnot. 2024. Rectpack: A Python module for packing rectangular shapes into a larger rectangle. https://github.com/secnot/rectpack Accessed: 1/24.
[66] Ticha Sethapakdi, Daniel Anderson, Adrian Reginald Chua Sy, and Stefanie Mueller. 2021. Fabricaide: Fabrication-aware design for 2d cutting machines. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.
[67] Xinyu Shi, Mingyu Liu, Ziqi Zhou, Ali Neshati, Ryan Rossi, and Jian Zhao. 2024. Exploring interactive color palettes for abstraction-driven exploratory image colorization. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–16.
[68] Fat Quarter Shop. 2023. Pre-cut fabric bundles. https://www.fatquartershop.com/
[69] Maria Shugrina, Jingwan Lu, and Stephen Diverdi. 2017. Playful palette: an interactive parametric color mixer for artists. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–10.
[70] Maria Shugrina, Wenjia Zhang, Fanny Chevalier, Sanja Fidler, and Karan Singh. 2019. Color builder: A direct manipulation interface for versatile color theme authoring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
[71] Gillian Smith. 2017. Generative Design for Textiles: Opportunities and Challenges for Entertainment AI.. In *AIIDE*. 115–121.
[72] Judy Turner and Margaret Rolfe. 2002. *Successful Scrap Quilts from Simple Rectangles*. That Patchwork Place (Martingale).
[73] 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.
[74] World Wide Web Consortium (W3C). 2022. CSS Color Module Level 4. https://www.w3.org/TR/css-color-4/
[75] Felicity Walker. 2016. *Building Blocks Sampler Quilt: a Quilting for Beginners Quilt Pattern Tutorial*. CreateSpace.
[76] Ludwig Wilhelm Wall, Alec Jacobson, Daniel Vogel, and Oliver Schneider. 2021. Scrappy: Using Scrap Material as Infill to Make Fabrication More Sustainable. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.
[77] 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
[78] World Wide Web Consortium (W3C). 2008. Web Content Accessibility Guidelines (WCAG) 2.0. https://www.w3.org/TR/WCAG20/ Accessed: 1/24.
[79] Shanel Wu and Laura Devendorf. 2020. Unfabricate: designing smart textiles for disassembly. In *proceedings of the 2020 CHI conference on human factors in computing systems*. 1–14.
[80] Chuan Yan, John Joon Young Chung, Yoon Kiheon, Yotam Gingold, Eytan Adar, and Sungsoo Ray Hong. 2022. FlatMagic: Improving flat colorization through AI-driven design for digital comic professionals. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–17.
[81] Lin-Ping Yuan, Ziqi Zhou, Jian Zhao, Yiqiu Guo, Fan Du, and Huamin Qu. 2021. Infocolorizer: Interactive recommendation of color palettes for infographics. *IEEE Transactions on Visualization and Computer Graphics* 28, 12 (2021), 4252–4266.
[82] Lvmin Zhang, Chengze Li, Edgar Simo-Serra, Yi Ji, Tien-Tsin Wong, and Chunping Liu. 2021. User-guided line art flat filling with split filling mechanism. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9889–9898.
[83] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. 2017. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999* (2017).
[84] Liu Zhenyuan, Michal Piovarči, Christian Hafner, Raphaël Charrondière, and Bernd Bickel. 2023. Directionality-Aware Design of Embroidery Patterns. In *Computer Graphics Forum*, Vol. 42. Wiley Online Library, 397–409.

A ADDITIONAL EXAMPLES

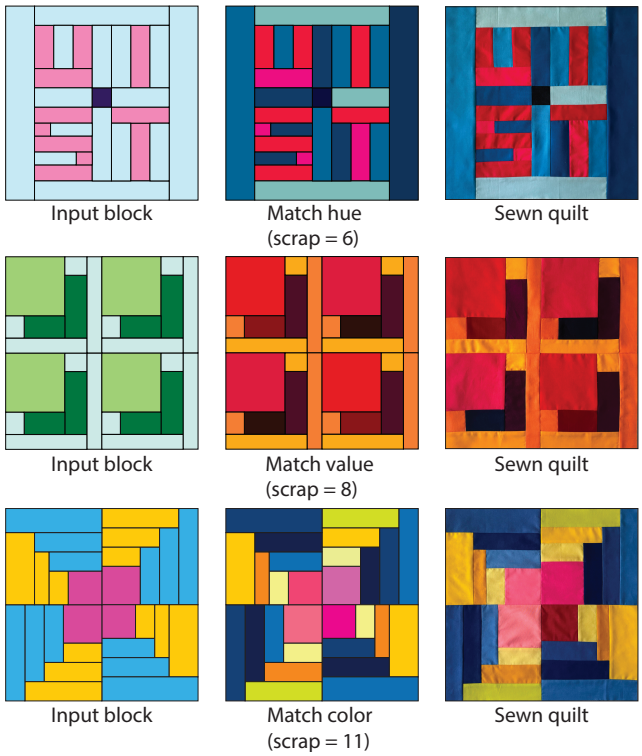


Figure 15: We used ScrapMap to color three additional scrap quilts, which we then sewed. The top and bottom rows feature blocks with custom geometry. The middle row is a traditional "floating square" block.

B SUPPLEMENTARY TECHNICAL IMPLEMENTATION

In this section we give an overview of the SMT encoding for constraints and objectives. As shorthand we use the same notational conventions as in Sec. 5 and omit SMT sorts (i.e., data types).

We use an enumerative approach for encoding the complex color difference functions. Because the coloring problem is discrete, (i.e., the number of fabrics is discrete, and the number of nodes is discrete) we pre-compute all evaluations of the metric for the given problem setup and use the resulting values as constants in the formulae. We use auxiliary variables to represent the color difference function value, along with connection constraints associating every choice of color assignments with the corresponding constant color difference evaluation value.

This enumerative approach has three major advantages: 1) It is modular, and therefore, it is easy to experiment with or add different color difference functions. 2) It folds all the computational complexities of the color difference functions into the pre-computation instead of the solver. 3) There is a level of performance consistency for all color difference functions. However, 1) For very simple functions, direct encoding results in quicker solves. 2) The scaling of the number of constraints with respect to the number of fabrics and graph nodes is better with a direct encoding.

B.0.1 Encoding max difference objective. For the first objective of maximizing edge color differences, we define an auxiliary variable for each edge, $\mathbf{x} = \{x_{n_a,n_b} \mid n_a, n_b \in E'\}$ representing each edge's color difference. We use the following connection constraint associating each possible color pair with the evaluation of the function M_k.

$$\phi_k^{conn_{MD}}(\mathbf{v},\mathbf{x}) \quad := \bigwedge_{n_a,n_b \in E'} \bigwedge_{c_a \in C} \bigwedge_{c_b \in C} (v_{n_a} = c_a \wedge v_{n_b} = c_b) \implies x_{n_a,n_b} = M_k(c_a, c_b)$$

Then, for the objective, we can use the auxiliary variables, $\mathbf{x}$.

$$\underset{\mathbf{v},\mathbf{x}}{Maximize} \quad \min \mathbf{x}$$

B.0.2 Encoding match target objective. For the objective of matching an existing colored block, $g_\mathbf{v}$, we define an auxiliary variable for each node $\mathbf{y} = \{y_n \mid n \in N\}$ representing the color difference between each node and the corresponding target color. We use a similar connection constraint associating each possible choice of color with the evaluation of M_k.

$$\phi_k^{conn_{MT}}(\mathbf{v},\mathbf{y}) \quad := \bigwedge_{n \in N} \bigwedge_{c \in C} (v_n = c) \implies y_n = M_k(c, g_\mathbf{v}(n))$$

Then, for the objective, we can use $\mathbf{y}$.

$$\underset{\mathbf{v},\mathbf{y}}{Minimize} \quad \max \mathbf{y}$$

B.0.3 Encoding constraints. The SMT encoding for ϕ^{GC}, $\phi_\mathbf{m}^{fixed}$, and ϕ^{unique} are similar to their corresponding definitions, (Defns. 5.7, 5.9, 5.10), so are omitted. For the scrappiness constraint, we use a cardinality constraint [1] counting the number of used colors[2].

$$\phi_s^{scrap}(\mathbf{v}) := \quad s = \sum_{c \in C} \left(\bigvee_{n \in N} v_n == c \right)$$

B.0.4 Example block coloring. For example, if the user wanted to maximize contrast with low scrappiness, they would choose the "Max difference" objective, choose "value" for the metric (corresponding to the M_{value} function computing relative luminance contrast ratio), and set, say, a scrappiness level of 4. Our tool would construct the following formulae.

$$\begin{aligned}
&\underset{\mathbf{v},\mathbf{x}}{Maximize} \quad \min \mathbf{x} \\
&\text{subject to} \quad \bigwedge_{n_a,n_b \in E} v_{n_a} \neq v_{n_b}\ , \\
&\qquad 4 = \sum_{c \in C} \left(\bigvee_{n \in N} v_n == c \right)\ , \\
&\qquad \bigwedge_{n_a,n_b \in E} \bigwedge_{c_a \in C} \bigwedge_{c_b \in C} (v_{n_a} = c_a \wedge v_{n_b} = c_b) \implies x_{n_a,n_b} = M_{value}(c_a, c_b)
\end{aligned}$$

[2]Each boolean term in the summation is first translated to an appropriate sort.

C GLOSSARY OF TERMS

Throughout the paper we use the following terms to describe different parts of the quilt design process in ScrapMap:

- Coloring - A particular assignment of colors to faces in a block
- Color palette – A set of possible colors to assign to a block (in this case, each color in the stash corresponds to a color in the palette)
- Face – Contiguous region of a block
- Motif - A grid of blocks repeated in the quilt
- Scrap – A single piece of fabric with certain dimensions and a single color value
- Scrappiness – Number of distinct fabrics used in a block
- Scrappy – Description of the appearance of using many fabrics in a quilt
- Stash – A set of fabrics
- Swatch – Colored square in the ScrapMap Block Panel corresponding to a scrap
- Traditional block – A named quilt block in the quilting canon (e.g., log cabin)
- Quilt – A grid of motifs
- Quilt block - The repeating geometry of the quilt