

Fabricaide: Fabrication-Aware Design for 2D Cutting Machines

Ticha Sethapakdi

ticha@mit.edu

MIT CSAIL

Cambridge, USA

Adrian Reginald Chua Sy

adrianrsy@gmail.com

MIT CSAIL

Cambridge, USA

Daniel Anderson

dlanders@cs.cmu.edu

Carnegie Mellon University

Pittsburgh, USA

Stefanie Mueller

stefanie.mueller@mit.edu

MIT CSAIL

Cambridge, USA

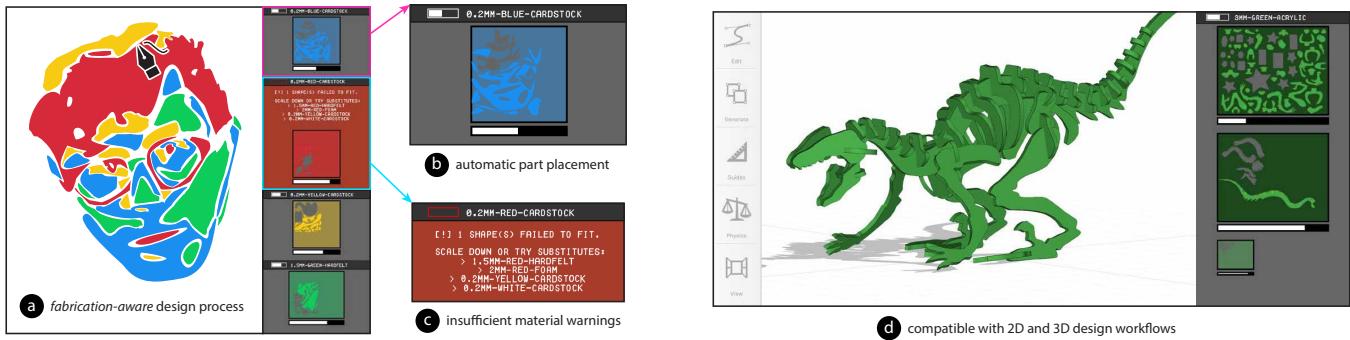


Figure 1: (a) Fabricaide enables a fabrication-aware design process for machine-cut objects by preparing them for fabrication while they are being designed. (b) As the user develops their design, Fabricaide decides good placements of parts onto the user's available materials, (c) provides warnings if the user does not have enough material for a design, and makes suggestions for how the user can resolve insufficient material cases. (d) Fabricaide extends existing design tools and can be used in both 2D and 3D workflows.

ABSTRACT

Designers of machine-cut objects must often consider whether and how their design can be fabricated with their available materials. In contrast to tools that support preparing finished designs for fabrication, we investigate shortening the feedback loop between design creation and fabrication preparation. To this end, we present Fabricaide, a *fabrication-aware* tool that interleaves the processes of creating and preparing designs for fabrication. By providing live feedback on how parts should be placed onto material sheets, analyzing how much material is consumed, and alerting users when designs are infeasible, Fabricaide enables users to proactively tailor their design to their available material. Fabricaide achieves this with a custom packing algorithm that arranges parts onto material sheets at interactive speeds. Our qualitative user study shows how Fabricaide can support different workflows, encourage material-conscious design practices, and provide insights on how to further improve similar interfaces in the future.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CHI '21, May 8–13, 2021, Yokohama, Japan

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8096-6/21/05.

<https://doi.org/10.1145/3411764.3445345>

CCS CONCEPTS

• Human-centered computing → Interactive systems and tools;

KEYWORDS

personal fabrication, laser cutting

ACM Reference Format:

Ticha Sethapakdi, Daniel Anderson, Adrian Reginald Chua Sy, and Stefanie Mueller. 2021. Fabricaide: Fabrication-Aware Design for 2D Cutting Machines. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan*. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3411764.3445345>

1 INTRODUCTION

Compared to other digital fabrication processes, laser cutters, die cutters, and similar machines have a low barrier to entry: designs can be created using simple 2D graphics software, many cutting materials can be readily obtained from local craft stores, and the fabrication process has fast turnaround times that allow for trial and error [5]. While designs are fabricated from flat sheets of material, parts can also be assembled into 3D structures, enabling laser cutters to be used for a variety of applications. Although laser cutters used to be exclusive to industrial or academic settings, recent developments such as the Glowforge and the Dremel LC40 Laser Engraver have brought laser cutters to the home.

However, laser cutters – and cutting machines as a whole – are not without their shortcomings. As a subtractive manufacturing process, they can produce significant amounts of leftover material which might never be used and go to waste. Given this, an important consideration users have is how to make the best use of the material sheets they have available. This is often tackled at the time of fabrication: for example, once users are finished with their digital design, they may try to arrange the parts so that it maximizes their space efficiency on the sheet. The task becomes particularly challenging when the sheets contain existing holes, which is a frequent occurrence in shared makerspaces and FabLabs [26]. Such sheets make it difficult for users to estimate how much material is still available and determine whether designs are feasible to fabricate. This is further complicated if the user wishes to create a multi-material design, which involves decomposing the design in their head into multiple pieces and then planning the layout of parts on multiple sheets simultaneously. Users would typically have to undergo a manual process which involves determining the locations of any holes in the sheet and adjusting the position of parts in their design accordingly, or tightly packing their design into a rectangle and trying to place it on an empty space in the sheet [25].

In recent years, several research projects have investigated how to help users organize finished designs onto their existing materials. Tools, such as VisiCut [23] and PacCam [26], for instance, provide interfaces that help users with positioning their designs onto material sheets once they are ready to fabricate them. While such interfaces aid users in planning their design at the time of fabrication, tools that do so during the design phase itself are under-explored.

In this paper, we present Fabricaide¹, a system that uses its knowledge of the user's available material sheets to interleave the design creation and fabrication preparation processes. Fabricaide acts as an interface that augments existing design tools, and is compatible with both 2D and 3D workflows (Fig. 1). As users create their design, the tool optimizes the placement of parts onto existing sheets and provides warnings if there is insufficient material with suggestions for material substitutes. By giving users feedback about the feasibility of their design as it is being created, Fabricaide allows them to design in the context of available materials. This benefits not only experienced designers, but also novices who are more prone to making mistakes that may be hard to detect in a normal design environment.

Packing parts at design time requires a 2D part placement algorithm that accounts for existing holes and works at interactive speeds. We provide such an algorithm² and, in our technical evaluation, show that it is faster than existing open-source tools for 2D packing while still producing comparable-quality solutions. We demonstrate the utility of Fabricaide in a qualitative user study in which participants used the tool for one week. In summary, with this work we contribute:

- a workflow that shortens the feedback loop between design creation and fabrication preparation for machine-cut objects by integrating material availability and fabrication preparation into the design phase,

¹code available at <https://github.com/tichaesque/Fabricaide>

²code available at <https://github.com/DanielLiamAnderson/Packaide>

- the design of a tool that facilitates this workflow, enabled by features including virtual material sheet tracking, multi-material sheet assignment, automated part placement, and material utilization analysis,
- an algorithm for packing parts onto material sheets with pre-existing holes at interactive speeds, allowing the design tool to continuously provide information on material availability and utilization,
- a qualitative user study that demonstrates how knowing about material constraints during the design phase affects the user's design process.

The aims of this work are thus twofold: our technical contributions show how to engineer interactive tools that can make use of knowledge of existing materials, and our evaluation provides insights into how these tools can support users' design processes, thereby contributing knowledge on how to improve interfaces for similar fabrication workflows. Broadly, Fabricaide contributes to the goal of making digital fabrication tools more intelligent, allowing them to act as intelligent shop assistants that can support users in their design work [31].

2 MOTIVATION

Recent exploratory work has delved into the needs and desires of professionals who regularly use cutting machines. Yildirim et al. [31] conducted workshops and interviews with professional users of digital fabrication tools and discovered that participants desired better tools for keeping track of resources, actively preventing errors, and automating mundane tasks, particularly when there is a chance for human error. To further refine our understanding of users' needs with respect to managing materials and how this may affect their process for creating machine-cut designs, we performed formative interviews with six designers who had used cutting machines for a diverse range of applications (school work, professional work, hobby projects, research). We focused on topics such as whether/how the designer considers their available material while they are creating a design, whether they plan around leftover material while designing, and in what circumstances they would choose to alter their design if they did not have sufficient material to fabricate it.

Designers frequently use leftover material for their work, particularly when they have to incur material costs themselves, although doing so comes with its own set of challenges. P2 noted how using leftover material is good to do but can be “such a pain”, as it is more prone to errors. They recounted the occasions where they had wasted material by accidentally cutting into existing holes, which made them more hesitant to use leftover material for larger scale work or when it is not obvious how the design should fit. To avoid the issue of cutting into existing holes, the designers would often measure the positions of holes on their sheets and draw them as rectangles that need to be avoided in their design tool. P4 expressed that this process “is a real pain”, and P6 said they could spend up to an hour figuring out how parts should be arranged. This demonstrates the importance of having tool assistance for such tasks that require careful planning and precision.

The option to utilize scrap material rather than always purchasing new sheets is important, since, as some designers noted, buying

new sheets is not always an option. P3 commented that leftover material is particularly useful when material is hard to find or simply inaccessible due to working outside of normal store hours. Additionally, geographic constraints may make obtaining additional material challenging or impossible. Cost and convenience was also a concern; P2 stated that they would have to think hard before having to buy expensive materials or order them through someone else, while P4 generally did not like buying material as it would require foresight and planning. An interesting consideration was having one-of-a-kind material sheets that cannot be bought. For example, P3 said they once laser cut designs out of material sheets they made themselves using thin slabs of plaster.

Current workflows cause designers to rely on arranging parts of their design onto material sheets at the very end, when they are ready to fabricate it. Some designers suggested that the reason why they do not consider how designs will be prepared for fabrication in advance is because doing so manually while designing is confusing and difficult. For example, P5's justification was that it would require them to think about the specific placement and rotation of parts: factors that often differ between the imagined design and the prepared design to be cut out by the cutting machine. This suggests that tools could support users on their imagined design, while automatically optimizing it for fabrication.

While there are tools that support preparing designs for fabrication after the design is complete, the above discussion suggests that packing earlier could be beneficial, particularly when material is scarce and there are many used sheets with existing holes. Designers had several different strategies for managing situations in which they were short on material. When purchasing additional material is not a viable solution, designers must modify their design in some way. Designers' modification strategies ranged from simple options, such as switching to a different material with more space or downsizing parts, to more complex ones which involve changing the geometry of the design. For instance, P4 would choose to make some parts thinner, while P3 and P6 might split large pieces in half and add joints to join them after fabrication. If the additional modifications were still not sufficient to make the design feasible for fabrication, designers may have to undergo a long trial-and-error process, potentially wasting significant time trying to prepare an infeasible design. Some modifications, such as downsizing a part, might require further cascading modifications to make other parts compatible, which could invalidate previous preparation efforts.

An important takeaway from these interviews is that designing for fabrication is often an *iterative*, rather than linear, process. This motivates a need to develop tools that can account for fabrication constraints *while designing* to shorten the feedback loop between design creation and fabrication preparation. Such a system requires a balance of careful engineering (i.e., fast algorithms that give users feedback quickly enough to not hinder the design iteration process) and design (i.e., an interface that clearly conveys the necessary information).

3 RELATED WORK

Research related to our work falls under three categories: assisting the user with the design of machine-cut objects, assisting the user

with preparing those designs for fabrication, and arranging parts of a 2D design onto a sheet.

3.1 Design: Creating Geometries

To support users in creating the geometry of laser-cut designs, researchers have investigated a variety of methods, including structuring freehand drawings for functional laser-cut designs (Sketch It Make It [18]) or using code as input to generate a laser-cuttable object (Codeable Objects [16]). Since laser cutters are limited to cutting 2D shapes, several research projects investigated how to support users with designing 3D objects from 2D parts. SketchChair [27] and Platener [8], for instance, allow users to create freeform 3D designs and then convert them into 2D pieces that can be assembled. Other approaches restrict the user to modeling with 2D pieces (FlatFab [21], LaserStacker [29]) or voxels (Kyub [6]), which allows for more accurate 2D conversions. Finally, tools such as CutCAD [15] and Joinery [32] generate joints based on the thickness of the material, which facilitates the design of connected parts.

The goal of our work is different as we do not focus on helping users create specific types of geometries, but rather help users ensure that those geometries can be fabricated as they are being created. Thus, our work is complementary with the work mentioned above.

3.2 Fabrication: Preparing Designs for Cutting

Several research projects have explored how to help users prepare finished designs for cutting. To aid users with placing their designs onto material sheets, many projects suggested the use of an overhead camera: VisiCut [23] and PacCam [26] display the camera image in the regular laser cutting software, users then align their drawing on the camera image. VAL [30] and ProjecTables [25] take this a step further by using an overhead projector to project the output onto the sheet before cutting. These tools allow users to better understand the scale and placement of their drawing on a sheet. Koo et al. [19] incorporate the notion of fabrication constraints to minimize wastage for furniture design. Given a design, their software suggests modifications that preserve the structural properties of the furniture but require less material wastage to produce. JigFab [20] supports users in fabricating constructions with power tools by computationally generating the physical constraints that guide the movement of those tools. It additionally allows the user to provide the size and quantity of the material they wish to use and alerts them if their design cannot be made with the available materials.

As the methods above help users with placement after they have finished their design, they come at the expense of users not knowing whether their design can be fabricated with the available materials until after they are ready to prepare it for fabrication. Without the foresight for how their design can fit onto their material sheets, designers have to go through a trial-and-error process of adjusting and checking the parts of their design until it is feasible.

3.3 Packing: Arranging Parts of a 2D Design

Producing overlap-free arrangements of 2D geometries is broadly known as the 2D *packing* problem, the *nesting* problem, the *irregular cutting stock* problem, or the *pattern marking* problem, depending

on the field, and has a range of applications in different domains that use cutting machines.

A number of algorithms for packing polygons (see e.g. [1, 7, 22]) involve a concept called the *no fit polygon* (NFP) [2], a useful tool for detecting overlap and producing candidate placement locations for sets of polygons. Packing non-convex polygons is much harder than packing convex polygons, and early solutions were only capable of handling few polygons at a time. Fischetti and Luzzi [13] improved an early approach proposed by Daniels and Milenkovic [12] based on Mixed-integer Programming. Work by Burke and Kendall [10, 11] uses metaheuristic algorithms and an approximate NFP to give one of the first highly practical solutions to the problem, on which some popular open-source software projects, such as Deepnest [24], are based.

The above algorithms are not designed for situations where the target sheets contain holes or defects that must be avoided. Early work by Heistermann and Lengaeur [14] approaches this in the context of leather manufacturing. Babu and Babu [3] give a heuristic algorithm for 2D packing that can also handle defects. Baldacci et al. [4] further explore this problem and develop a layered algorithm that uses heuristics to distribute parts onto multiple sheets, and a guided local search based on Lagrangian relaxation to optimize the placement of parts onto an assigned sheet.

These algorithms tend to be targeted towards industrial manufacturing settings which prioritize material-saving over interactivity, so none of them attempt to achieve interactive speeds. For an interactive design tool, our main goal is to achieve interactive speeds to provide the user with continuous feedback, with material-savings being a secondary goal.

4 CORE SYSTEM FEATURES

In traditional machine-cutting workflows, the machine is naïve as it does not have any knowledge about how much material is available or how much material the current design requires. Since it is missing both pieces of information, it cannot determine if a design can be fabricated with the available resources. In contrast, Fabricaide maintains information about the available space on each material sheet as well as the geometries of the user's design. Fabricaide uses this information, combined with its automated part placement, to decide whether a design is feasible before users proceed to fabricate it. The workflow enabled by Fabricaide is an iterative process that cycles between creating parts, assigning materials, and checking packing previews.

4.1 Compatible with Existing Design Tools

The Fabricaide interface (Fig. 2) is a stand-alone application that augments existing design tools and can be used with a variety of design workflows. All that is required is that the design tool exports the designs as 2D vector outlines. The export can either be done by the user manually, or, if the tool supports it, the design can be automatically exported either periodically or whenever a design part is added or modified. As part of our main walkthrough, we demonstrate Fabricaide's use with Adobe Illustrator, a 2D vector drawing tool. We later show how Fabricaide can be applied to other design tools and workflows by integrating it with a CAD tool for producing 3D laser-cuttable objects (FlatFab [21]).

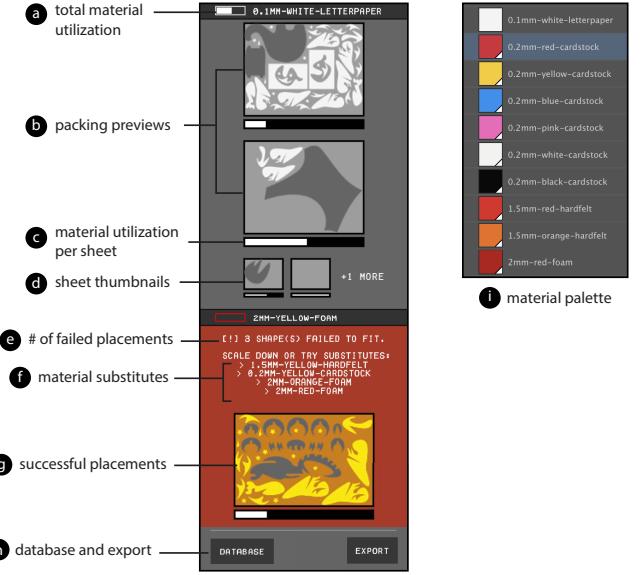


Figure 2: The components of the Fabricaide interface.

4.2 Drawing Parts and Assigning Materials

Parts can be made using any of the drawing tools in Illustrator, provided that they produce a closed path. To enable the tool to assign parts of multi-material designs to material sheets, users specify the material for a part with Fabricaide's *material palette* (Fig. 2i), which is a collection of Illustrator swatches. Each swatch has a unique name and color that identifies a material with a specific type (e.g., wood or plastic), color, and thickness. To assign materials to parts of their design, users assign a swatch to the “fill” attribute before or after the part is drawn.

4.3 Live Packing Previews

Fabricaide uses live *packing previews* to show users how the parts are placed (Fig. 2b). When the user creates their first part with a particular material assignment, Fabricaide tries to find a good placement for it on a sheet corresponding to its material, optionally considering rotations of the part. If Fabricaide places it successfully, the resulting placement is shown on its respective material sheet and displayed to the user. The packing preview renders the design parts in full opacity, the sheet in partial opacity, and the holes of the sheet in dark gray. When additional new parts for a particular material are added to the design, or an existing part is modified, Fabricaide re-optimizes a new packing layout for all parts corresponding to that material. Fabricaide only re-generates packing configurations for materials that have been affected; in other words, changing the parts for one material does not affect the packing configurations for other materials. If there are more sheets for that material that can be used, Fabricaide additionally shows the user up to two *sheet thumbnails* (Fig. 2d) as well as the number of sheets remaining.

4.4 Material Warnings and Substitutes

If Fabricaide finds that none of the sheets belonging to the assigned material have enough space for all the parts, the panel will alert the user with a *material warning*. The user is shown a preview of the parts that were successfully placed in their design (Fig. 2g) as well as the quantity of parts that could not be placed (Fig. 2e). When short on material, users may decide to resize their design or reassign materials based on either aesthetic or functional reasons, e.g., same color or same thickness. In addition to checking the database for alternative materials, the user can try the material substitutes suggested by the warning (Fig. 2f), which shows up to two same-color substitutes and up to two same-thickness substitutes.

4.5 Creating Fabrication-ready Files

When users are finished with their design, Fabricaide provides the option to export the placements as PDFs ready to be cut. Fabricaide adds the cutouts as holes to the corresponding virtual sheets stored in the material database.

5 ADVANCED FEATURES

Fabricaide also offers advanced features that further aid users with planning, giving them control over the packing configurations, and minimize the chances of error during fabrication.

5.1 Material Utilization

While packing previews provide users with information about how the space on a material sheet is being used, users may want to compare how much space is remaining across different materials and their sheets. To facilitate this, Fabricaide offers the notion of *material utilization* to approximate how much usable space there is on a material sheet. Simply computing the total area remaining on the sheet would not suffice because the arrangement of the pieces misses how usable the remaining space actually is. Material utilization therefore also accounts for the tightness of the packing, since a sheet with parts tightly packed together leaves more available space for future parts than one with parts that are spread out.

During the design process, Fabricaide continuously analyzes the material utilization of the current placements of the parts of the design. Fabricaide conveys material utilization to the user in the form of material utilization bars that approximate the percentage of remaining usable material per material type (Fig. 2a), as well as per individual sheet (Fig. 2c). A more filled bar indicates that the sheet has a greater amount of usable space.

5.2 Editing Packing Configurations

In our observations of laser cutter users, we noticed that they sometimes choose to arrange parts to help keep track of them at the time of fabrication. For instance, this approach can be useful when they are cutting many parts that looked similar with small distinctions, as is typical for mechanical parts such as gears with different numbers of teeth. To enable this, Fabricaide gives users the option to manually edit the packing for a sheet (Fig. 3a). This opens the sheet with its holes in Illustrator, allowing the user to tweak the positioning of the parts (Fig. 3b). As the user is tweaking the packing configuration, the UI updates itself to show the material utilization of the user's packing. The user's packing configuration is locked

unless the user decides to modify or add parts to that material, at which point it must be repacked to ensure that the design is still feasible.

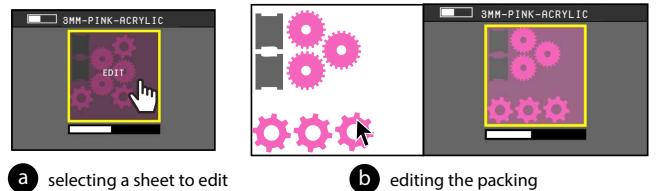


Figure 3: (a) Manually editing packing configurations for a set of gears. (b) The user modifies the packing so that similar gears are grouped together.

6 THE MATERIAL DATABASE

Several of Fabricaide's functions are enabled by having information on material sheets and their already-existing cut-outs available. To accomplish this, Fabricaide maintains a *material database* that stores the current states of all material sheets. The material database stores metadata about each material type (dimensions, thickness, color, name), and for each material type, the existing holes of each available sheet. Each material type is also associated with a unique hexadecimal color code, which is used to assign parts of a design to particular materials via the swatch library.

Fabricaide has a material database viewer which allows users to view all available material sheets in the database (Fig. 4). The database viewer enables users to inform their design based on what is available or help them decide whether they should order materials ahead of time if they see that they are low.

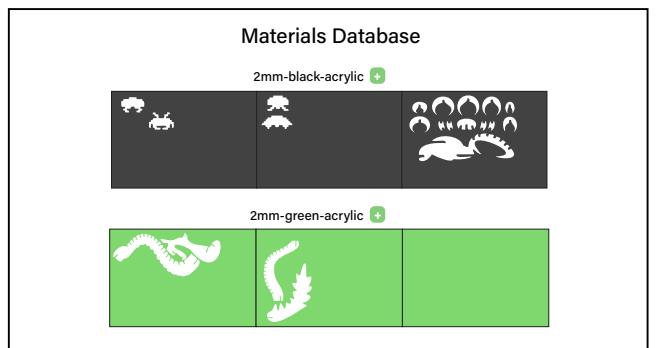


Figure 4: The material database viewer, showing the available sheets for 2mm black acrylic and 2mm green acrylic.

6.1 Adding Sheets to the Database

The first time Fabricaide is launched, the material database is empty and only contains the metadata for the supported material types. Users can add blank sheets or virtual copies of existing (used) sheets to the database through the database viewer. Once the user takes a photo of their sheet on a contrasting background (Fig. 5a), they

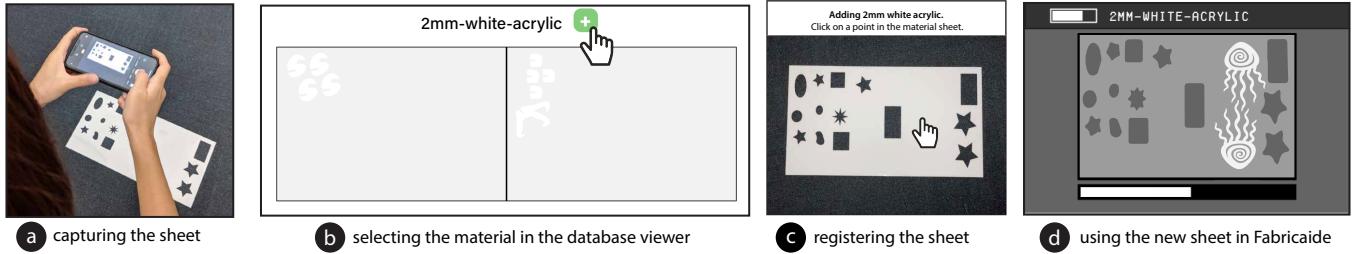


Figure 5: The workflow for adding a sheet to the Fabricaide database.

can add it to the database by uploading it to Fabricaide (Fig. 5b) then selecting a point on the photo that identifies the material sheet (Fig. 5c). Fabricaide then creates a virtual representation by extracting the outlines of any holes using image processing techniques. Once the sheet is added, users can use it for their designs (Fig. 5d).

6.2 Virtually Tracking Cutouts

Every time a design is exported for fabrication, Fabricaide updates the material database with holes corresponding to the cutouts of the newly cut parts. Tracking material usage digitally has the benefit that it does not require additional hardware, such as a camera, and does not require calibration. In addition, it is more accurate since the outlines do not have to be extracted from a photo. If a sheet is modified outside the Fabricaide environment, rendering the virtual copy out of date, users can take a photo of the updated sheet to re-add it to the database and generate a new virtual representation.

7 IMPLEMENTATION

The Fabricaide interface is implemented in Processing, and is designed to be run in parallel with existing design tools. For Adobe Illustrator compatibility, we used Illustrator scripting to periodically export the current document as an SVG file so that it can be continuously analyzed. The background data processing of the exported document is implemented in Python, and is responsible for preprocessing the design file, sending it to the packing algorithm, and then generating the resulting images that are displayed to the user.

7.1 Enabling Compatibility with Design Tools

To make Fabricaide compatible with other design tools, only minor modifications need to be made to both sides. A prerequisite is that the design tool must support exporting designs as 2D vector files that can be read by standard cutting machines (e.g., PDF and SVG formats). Fabricaide can then perform the necessary transformations to the input for packing. For software to support multimaterial sheet assignment, there must be an option to change the appearance of the parts (e.g. via a fill color), however, this is optional.

7.2 Splitting into Constituent Materials

Fabricaide first splits a design into its individual materials by analyzing the fill attribute of the parts in the exported SVG file and checking which material it maps to in the database. If a part has a fill that is not mapped to a specific material, it is ignored. The

result is a set of SVG images, each containing the contours of the parts assigned to a particular material.

7.3 Optimizing Placement on Material Sheets

Once the design is split into its constituent materials, our system's goal is to place (and optionally rotate) each part to make efficient use of material. We implemented a custom 2D packing algorithm which is based on the no fit polygon (NFP) approach [10] with custom heuristics, and additional performance optimizations. The performance-critical part of the algorithm is implemented in C++. We use the CGAL [28] library for computational geometry primitives. Pre- and postprocessing is implemented in Python. The pre-processing is responsible for discretizing SVG elements into polygons, and applying dilation to ensure that the polygons are never under-approximated and to account for laser cutter tolerance.

At a high level, our algorithm is similar to the classic first-fit decreasing heuristic for bin packing [17]. Given a set of parts to place, the algorithm greedily places each one in decreasing order of size (biggest to smallest by the area of their bounding box) onto the first sheet on which it fits, using NFPs to find feasible candidate locations, and a heuristic quality score to select the best candidate location. Given a set of placed polygons P and holes H , our main heuristic is to minimize

$$\text{AREA}(\text{BBox}(P)) + \text{AREA}(\text{BBox}(P \cup H)),$$

where BBox computes the bounding box of the given set of polygons. The contribution of $\text{AREA}(\text{BBox}(P))$ encourages new parts to be placed tightly together, and the contribution of $\text{AREA}(\text{BBox}(P \cup H))$ encourages parts to be placed tightly around existing holes. The advantage of this heuristic score is that it can be computed incrementally in constant time per vertex as more parts are placed. This makes it preferable to using a metric such as the material utilization score, which would be significantly more complex to compute incrementally. To break ties in the case of multiple equal scores, the algorithm selects the placement location (x, y) that minimizes the value of $x+y$. This has the effect of tightening the packing in the presence of non-convex parts that have large bounding boxes with lots of empty space inside them. If the user enables rotations, they can specify a parameter k such that the algorithm tries k uniformly spaced rotations between 0° and 360° and selects the best one (based on the heuristic score). This has the advantage of allowing the user to preserve the grain direction of the material by setting $k = 1$. If $k = 1$, then the only rotation considered is a 180° spin, which preserves the angle of the geometry with the grain.

We improve the performance of the algorithm by caching the computations of NFPs, not just within a run, as has been done before by Burke and Kendall [10], but across separate runs of the algorithm. Since subsequent runs of the algorithm will likely be run on designs that have only changed by a small amount, most of the same parts will be present. For instance, given a design consisting of n parts, such that only one part differs from the previous design, the number of required NFP computations drops from $\binom{n}{2}$ to just $n - 1$. Since the NFP computation is the most expensive part of the algorithm, this caching eliminates a large amount of redundant computation. Further details on the packing algorithm and its implementation are available as supplemental material.

7.4 Calculating Material Utilization

Since many 2D cutting workflows involve cutting parts from long rolls of fixed-height material, a common metric of material utilization that has been used is the width of the section of sheet that contains the placed parts, i.e. the length of sheet that must be rolled off the roll. This favors tight packings over ones with parts spread far out. This, however, does not give sensible values if we consider fixed-width material sheets with arbitrary existing holes, since the holes themselves may be spread out, and it is usually preferable to place new parts close to the existing holes, as this leaves more usable space left on the sheet.

To address the above limitation, we designed the following metric of material utilization: we consider every hole and newly placed part on the sheet, dilate them by some fixed amount, and compute the area of the union of the resulting shapes. A smaller value indicates better material utilization. This metric has the effect of penalizing parts that are placed far away from others (Fig. 6a), and rewarding tighter packings, since the dilated areas will overlap (Fig. 6b). The dilation is set to a default of 20 points or approximately 7mm to accommodate for the recommended minimum amount of spacing between parts to be laser cut (2mm) and the recommended minimum width of a part that can be cut (typically between 2mm to 5mm depending on material thickness).

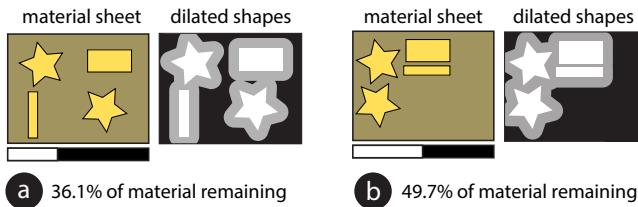


Figure 6: Using dilations to calculate material utilization. (a) shows the effect of a poor packing configuration on the utilization bar for a material sheet while (b) shows the effect of a good packing configuration. As can be seen, a smaller area after dilation indicates a better packing.

7.5 Material Tracking: Extracting Contours from Photos

To extract the existing holes in a sheet from a photo the user took, Fabricaide first asks the user to select a point in the photo containing the sheet, and uses a flood-fill (adaptive thresholding) algorithm to determine the region of the photo that corresponds to the sheet. It then crops the photo to the bounds of this region, and uses the contour finding method implemented in OpenCV [9] to find the holes. It then discards all contours that are too small and hence are likely due to noise, and smooths the extracted contours using spline interpolation. The resulting contours are then saved as SVGs to the database.

8 QUALITATIVE USER STUDY

We were interested in seeing how the workflow enabled by our system can support users' design processes. In particular, we wanted to understand: which system features users prioritize as they are designing, why they prioritize those features, and how their priorities might change over time. To this end, we ran a qualitative user study with designers, in which they used Fabricaide with Adobe Illustrator over the period of one week.

8.1 Participant Information

We recruited six designers (6 female, mean age 22.7) over email. Participants had an average of 3.8 years of experience with laser cutting and were compensated at a rate of \$20 USD per hour (for up to 14 hours) for taking part in the study. All participants had Mac computers from 2013 or newer with Adobe Illustrator 2020.

8.2 Material Database

All participants were given a material database which contained a starter set of material sheets, spread across 4 different material types (Fig. 7). These material types were chosen to provide visual variability and sufficient substitutes for each material type. Each material type contains 4-14 sheets. For realism, each material sheet in the database had some number of holes with shapes taken from laser-cut designs on Thingiverse, with varying levels of complexity. Holes were positioned on the sheets manually to simulate a situation in which sheets had previously been cut without Fabricaide.

8.3 User Study Procedure

We asked participants to install the Fabricaide software onto their computers and provided them with a reference guide for the tool. To simplify the installation, we avoided the need for users to install library dependencies by containerizing the back-end code using Docker. At the start of the study, we met with participants individually to ensure that the tool was correctly installed. Throughout the study, we also provided them with technical support as needed.

Participants were given one week to complete a set of multi-material designs using Fabricaide. We asked them to complete at least 4 designs to ensure that the tool was used regularly. Participants were asked to save any designs they made and export the packing configurations as they completed them. They also had the option to toggle the visibility of all the UI elements in Fabricaide (Fig. 2), except for the material warnings. To get a sense of when

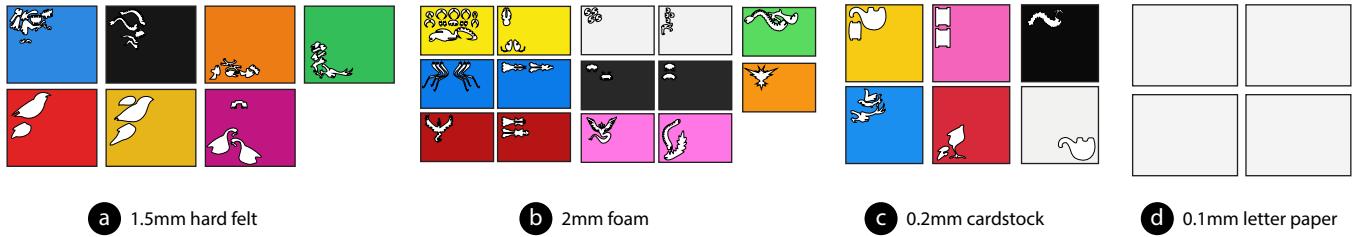


Figure 7: The material database used in the user study. The holes were taken from various designs found on Thingiverse.

participants might prioritize continuous packing, we additionally gave them the option to disable live packing and only refresh on demand.

8.4 Data Collection

Prior to the study, participants filled out an entrance survey which asked about their laser cutting experience. During the study, participants maintained a daily log of their experiences. Check-in meetings were conducted every second day to discuss what participants had reported in their daily log and review any new designs. At the end of the study, participants filled out an exit survey which asked them when and how they used various features and the ways in which those features affected their design process. We concluded with a wrap-up meeting to reflect on their experiences.

8.5 Results

Five out of six participants completed at least 4 multi-material designs and spent an average of 12 hours designing (i.e., planning and making their designs). P5 left the study after completing 2 designs and spending 6 hours designing.

8.6 User Study Findings

To recap, the goal of the study was to understand how Fabricaide could support users in their design process and when users might prioritize certain features. In this section, we discuss our findings from the feedback from our check-in meetings and responses to the exit survey.

Fabricaide supported different drawing workflows. While the drawing approaches varied between the participants, they broadly fell into two groups: those who took a largely *additive* approach to drawing and those who took a largely *subtractive* approach. Subtractive approaches involve creating initial shapes and refining them by removing portions of geometry, while additive approaches involve progressively creating shapes and extending existing ones.

Whether the designer took an additive or subtractive approach to drawing affected when they would pay attention to Fabricaide. P2, who took a largely subtractive approach, noted that Fabricaide was most relevant to her at the beginning, when she would “make an initial shape that will be the approximate size of [her] final design”, since afterwards her time was spent on “detail work that does not affect the overall size of [her] design”. P4 similarly stated that because she starts with large shapes before cutting out holes or notches, she tended to pay more attention to Fabricaide near the

beginning. This suggests that the attention given to Fabricaide is front-loaded for subtractive approaches, as subtracting material will only improve the feasibility of fitting the design onto sheets. Since their entire design depends on the initial shapes created, designers who use subtractive approaches may be better off packing their shapes closer to the beginning rather than at the end.

Conversely, designers who frequently had to add shapes in their design process checked Fabricaide more regularly than those who used mostly subtractive approaches. P1’s process involved creating an initial sketch on paper and drawing over it in Illustrator. She checked Fabricaide whenever she would finish a component in her design to confirm that it had no issues and that there was enough material.

Users became more careful as they designed. Fabricaide encouraged participants to take pauses in their design process. P6 commented that the packing previews “made [her] more careful when [she] was designing” and encouraged her to take a more deliberate approach to drawing. Seeing existing holes in the packing previews made P2 approach her designs as if the materials were being shared with other users, which made her more conscientious about the materials she was using as she did not want to use them all up herself. Since laser cutters and similar fabrication machines are often in shared spaces like workshops, users try to be conscientious of the resources they are using. Showing how the material sheets are being consumed while designing, as well as updating the database when designs are exported, further reinforces the fact that the designs the user creates has consequences on the limited materials they have.

Participants who used shapes that were large relative to their material sheet were especially keen to check the previews. P4, for instance, said that she had to check the packing previews many times with her birdhouse design (Fig. 8a) due to its large size. She additionally remarked that “It was good to know how much space I had left as I designed though, so I wouldn’t make a huge complicated thing and realize at the end it wouldn’t fit.” P5 stated that while she felt that the packing previews did not affect what she wanted to design, she kept track of them regularly to ensure that the design could be packed efficiently and confirm that she had enough material.

When participants were either making small designs relative to their material sheet or making finer adjustments to their designs, they found it less necessary to check Fabricaide frequently since those cases had little effect on packing feasibility. Similarly, participants also did not need Fabricaide for validation when they were making reference lines or shapes that were not part of the design

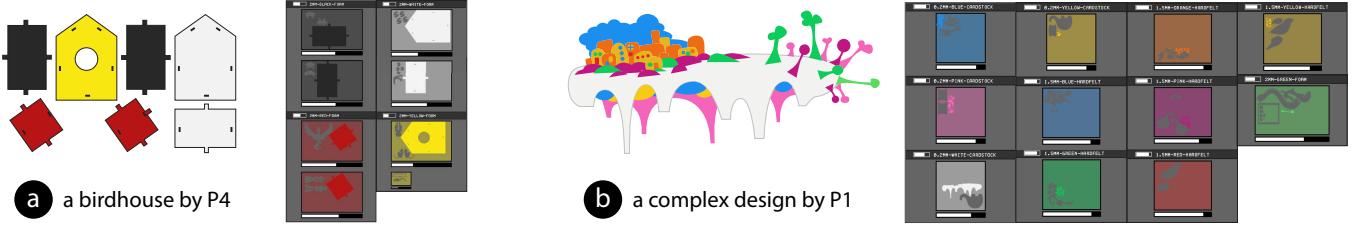


Figure 8: Selected designs and their packings from the user study.

itself. To focus more on their design task, participants sometimes disabled continuous packing and only repacked as needed. P1 liked to keep Fabricaide paused until she was done with sections of her design, since her process involved frequently refining existing shapes. P3 paused Fabricaide to prevent it from updating while she was still drafting her design. P2 commented that she liked being able to pause Fabricaide, though it came with the trade-off of needing to remember to refresh.

Material warnings prevented shortages. The insufficient material warnings helped participants avoid making designs that could not be fabricated with available materials because when the warnings occurred, participants did not leave them unresolved. Participants addressed warnings either by scaling down the design or switching out materials; the choice depended on the design itself and the amount of flexibility the designer had. Those who needed their parts to have specific measurements tended to switch materials rather than adjust the size of their parts. For example, when P4 encountered warnings as she was designing the birdhouse, which had press fit joints, she originally planned to make everything out of the same colored material but found out that she did not have enough. She resolved the warnings by opting for a multicolored design instead (while maintaining the thicknesses).

P1, P2, and P3 commented that another use case for Fabricaide's material warnings would be to quickly check how many copies of a design can fit onto different materials. P2 recalled a time in the past when she had to make a design that required many parts and had to spend a decent amount of time trying to manually calculate how many would fit on her sheet. She said that with the current Fabricaide system, this is easy to achieve by just copying and pasting the shapes on her document until a warning occurs. This could be made even easier by integrating it as a feature into the tool, so that the system would place multiple copies of each part and provide users with feedback as they are designing.

8.7 Post-study Refinements

Following the study, we incorporated several features into Fabricaide that were requested by multiple users to demonstrate that our system has the capabilities to support them.

Using a 3D editor. When we learned that some designers preferred to use CAD software to create 3D designs, we wanted to ensure that Fabricaide could indeed support that workflow. Because Fabricaide is a stand-alone application, rather than an add-on to an existing program, it can work alongside any software that can export vector files used by a cutting machine. To demonstrate this,

we added compatibility with FlatFab (Fig. 9a), a design tool for creating 3D structures from 2D planar sections.

Making multiple copies of designs. Several designers from the study felt like they could use Fabricaide's fast automatic packing to keep track of how many copies of their design could be fabricated. To aid in this process, we implemented a copies feature that allows the user to specify how many copies of their design they wish to fabricate (Fig. 9b). As this was a use case that was mentioned in the study, we additionally provide the option to pack the maximum number of copies for a design, which is calculated via a binary search and displayed to the user.

Displaying additional sheets needed. Since some designers said they would prefer to buy new sheets if they ran out of material, we implemented a feature to capture these use cases (Fig. 9c). Material warnings were modified to state how many additional blank sheets are needed to accommodate a design that is short on certain materials. If prices are known, this can be further modified to show the cost of additional material – which may incentivize designers to be more conservative with how much material they use.

8.8 Discussion

We discuss how the feedback we received can be used to improve similar design tools for fabrication, which incorporate information on material constraints.

Know when to show updates to the user. A key quality for interfaces that provide continuous feedback to the user is knowing when updates should be made and displayed. Displaying updates constantly is helpful for ensuring that users have up-to-date information, but comes at the expense of wasting computation power or becoming a distraction. As seen in the user study, whether the UI should continuously provide feedback to the user depends on where they are in their design process. In an ideal scenario, the interface would update more frequently when users are making significant changes to their design than when making finer adjustments. The simplest way to address this is to give users the option to toggle between continuous and manual updates, but it may also be beneficial to enable the system to infer when updates should be made. One option would be to automatically pause updates when the user is making many changes over a short period of time and resume once the user is idle.

Provide warnings at appropriate times. It is important that users are aware of any issues (e.g. material shortages) in their design in a timely manner so they can address them right away.

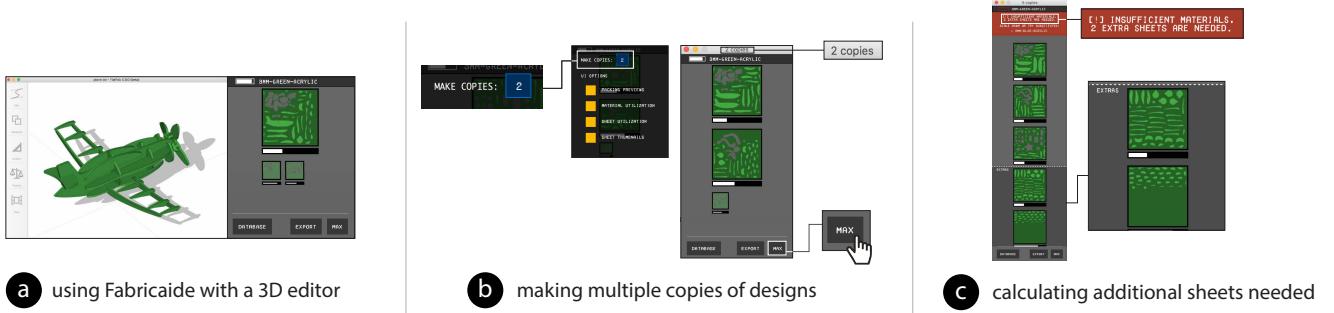


Figure 9: Refinements made to Fabricaide following the user study.

Similar to the previous point of knowing when to update, future systems can be expanded to know when warnings need to be made. Care must be taken to not show warnings earlier than needed to avoid distracting the user or providing false positives.

Encourage users to be less wasteful. Tools that incorporate information about available materials have the opportunity to encourage users to be more mindful of their making practices. As observed in the study, one designer believed that she was sharing materials with others, which affected the way she approached her designs. Encouraging more material-conscious design practices has not only financial benefits (e.g., users save money on purchasing new material by reusing existing ones), but environmental and social benefits as well (e.g., materials that can be used for essential purposes can be depleted more slowly).

9 EVALUATION OF PACKING ALGORITHM

The goal of Fabricaide’s part placement algorithm is to be fast enough to not be a hindrance to interaction without overly sacrificing the quality of the solutions. In this section, we present a quantitative evaluation of the performance of our implementation to demonstrate its suitability to this task.

9.1 Benchmark Dataset

To ensure that our benchmarks were conducted with respect to geometries typical of laser-cut designs, we created a dataset consisting of parts drawn from the designs of our user study participants, which, in total, contained 246 parts. To build our dataset, we created 10 sets of 100 parts each, which were randomly sampled from the parts of the users’ designs. The parts, when discretized with a tolerance of 2.5pts (0.035 inches) and dilated by 5pts (0.07 inches), have an average of 27 vertices. 54% of them are non-convex, and 13% contain composite holes. The target sheets are blank sheets that are sufficiently large so as to not constrain the solution (i.e. they are larger than the bounding box of any possible configuration of the parts).

9.2 Benchmark Setup

We ran our experiments on a budget personal desktop computer equipped with an AMD Ryzen 5 2600X processor and 16GB of DDR4 main memory. For each of the 10 sets of parts in our dataset, we first measured the performance of the algorithm without cross-run

caching. We measured the time taken to pack only one part (representing the first step in the design process), then pack two parts, and so on (representing an extension of the design over the entire design process). We then repeated this experiment but with cross-run caching enabled. This evaluation simulates the typical application of the algorithm during the design process, and simultaneously allows us to evaluate how much cross-run caching improves performance, and observe the scaling of the algorithm with respect to the number of parts to be placed. Rotations of parts were not considered, since they just increase the runtime linearly and hence can be factored in if desired.

9.3 Evaluation of Runtime

The results of our experiment are shown in Fig. 10. For each number of parts placed, we report the average runtime over the 10 sets. The scaling of the runtime appears approximately quadratic; packing twice as many parts takes roughly four times as long. This is congruent with the theoretical complexity of the algorithm, which, assuming parts have a constant number of vertices, is quadratic in the number of parts.

Enabling cross-run caching results in a 1.8–2x observed speedup on our benchmark data. We note, though, that this number is dependent on the complexity (number of vertices, non-convexity, etc.) of the parts. To observe this fact, we repeated the experiment but used 10 times the dilation and discretization tolerance, which resulted in the dataset having an average of just 6 vertices per polygon, 35% of which were non-convex. In this setting, caching produced a speedup of around 1.5x. This shows that caching is useful, and even more so as the designs become more complex.

We observe that our algorithm can handle 100 parts in just a few seconds with caching enabled, which is far more parts than in any individual participant’s design, which had an average of 9. The largest had 74 parts, but spread over 11 materials, which are packed separately (Fig. 8b). This demonstrates that our system is capable of packing at interactive speeds for typical laser-cut designs using modest personal computing hardware.

Note that we do not consider benchmarking on sheets with holes because the performance of the algorithm on such instances is comparable if not faster than an instance with an equivalent number of polygons on a blank sheet, since candidate placements do not have to be generated for holes. For example, the time required to pack 50 parts onto a sheet with 50 holes will be faster than the

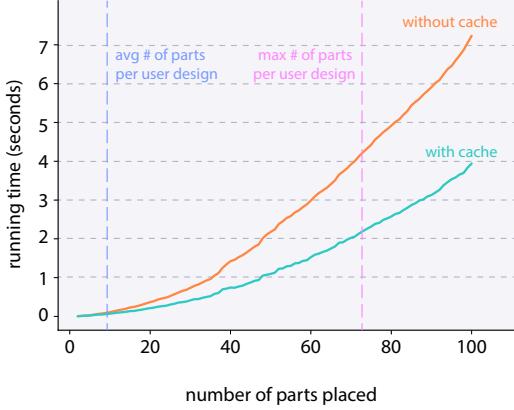


Figure 10: Average running times of the packing routine for various numbers of parts to place.

time to pack 100 parts onto a blank sheet. Our experiment therefore provides an upper bound of the running time for such instances.

9.4 Evaluation of Solution Quality

To ensure that our algorithm has not sacrificed solution quality to achieve its speeds, we compared the runtime and quality to Deepnest [24], a state-of-the-art open-source nesting tool. Deepnest uses a metaheuristic search, i.e., it can run for an arbitrarily long time searching for better quality solutions. We therefore performed two comparisons. For each set of parts in our dataset, we measured the time taken for Deepnest to produce its first feasible solution, and we recorded the best solution obtained after running it for 5 minutes. Deepnest was configured with equivalent parameter settings (curve tolerance, dilation, etc.) to ours to ensure consistency. To compare solution quality, we measure the area of the bounding box of the placed parts, which is a fair comparison since this value is used by both our and Deepnest’s objective functions. Deepnest found solutions for nine of our ten input sets, but, likely due to a bug, was unable to solve the remaining one. We therefore consider only the nine input sets that were solved in our comparison.

On average, Deepnest found its first feasible solution after 31.3 seconds, which is 4.3x longer than our algorithm without any cross-run caching at 7.3 seconds. The average quality of the first solution was also 8% worse than that of our algorithm. The best solutions found by Deepnest after 5 minutes were, on average, just 3% better than Fabricaide’s.

10 LIMITATIONS

One limitation of our prototype system is the robustness of the virtual material capture method. Since the metadata of each material type includes the size of the material sheets, sheets which have had large pieces removed from the edges (and whose bounding box therefore no longer matches the material dimensions) cannot accurately be captured by taking a photo unless the user provides the sheet’s dimensions. Additionally, using a flood-fill based method for discriminating the sheet and the background renders it unsuitable for patterned sheets. Lastly, the photo-based material capture method requires the user to accurately take a top-down shot of the

sheet. Although many of these issues can be alleviated by adding more sophisticated computer vision methods, related work on material capture has taken advantage of an overhead camera in the cutting machine to address these problems [26]. This removes the need for the user to worry about scale and perspective, and for them to assist in discriminating the sheet from the background. Using an overhead camera during fabrication can also safeguard against user errors by validating sheets before they are cut. As built-in overhead cameras are making their way into industrial and personal cutting machines (e.g. the Glowforge), a future version of our system can take this added capability into account.

Another limitation is that the system relies on the user to maintain an up-to-date library of materials. Since Fabricaide updates the database with the planned cutouts each time designs are exported, projects that are designed but not executed would render the database invalid. A more practical approach might be to maintain a list of pending jobs and only update the database for the jobs that users mark as complete. To better automate this process, a possible extension would be to connect Fabricaide to the cutting machine interface to track when a particular design was executed.

11 CONCLUSIONS AND FUTURE WORK

In this work, we investigated shortening the feedback loop between design creation and fabrication preparation for machine-cut objects. We presented Fabricaide, a tool that supports users in designing machine-cut objects by interleaving the processes of creating and preparing them for fabrication. In our qualitative user study, we saw how Fabricaide supported users and played a role in their design process, and used their feedback to provide insights on how to further improve interfaces for similar fabrication workflows. Our technical evaluation demonstrated that Fabricaide can handle typical user designs at interactive speeds on modest hardware.

Our user study findings open up several directions for future research. As touched upon, users can be more conservative with their materials if they believe that they are being shared with others. Possible extensions of Fabricaide include supporting multiple concurrent users and investigating how this can encourage conscientiousness or enable collaborative fabrication workflows. It would also be interesting to consider how Fabricaide can intelligently use material information during the design process – such as preserving the grain direction of textured materials, suggesting material substitutes that consider physical (not solely visual) properties, or warning users about making cuts that are incompatible with the material type.

While Fabricaide was built to be used at design time, an avenue for future work is how such a tool could assist users at fabrication time as well, especially if it can safeguard against fabrication errors. For example, a future version of the system can allow users to verify that they are using the correct material sheet by uploading a photo and checking that it matches the assigned sheet in their exported file. This process can be more automated with an overhead camera, as mentioned before.

Finally, while Fabricaide was built to work with 2D cutting machines, future research can explore how the principles behind our techniques could be applied to 3D subtractive manufacturing processes, such as CNC milling.

12 ACKNOWLEDGMENTS

We thank the designers who participated in our formative interviews and user study. We also thank Karan Singh and James McCrae for the FlatFab models, as well as Patrick Baudisch, Steve Hodges, and James McCann for their valuable feedback. This material is based upon work supported by the National Science Foundation under Grant No. 1716413.

REFERENCES

- [1] Michael Adamowicz and Antonio Albano. 1976. Nesting two-dimensional shapes in rectangular modules. *Computer-Aided Design* 8, 1 (1976), 27–33.
- [2] Richard Carl Art Jr. 1966. *An approach to the two dimensional irregular cutting stock problem*. Ph.D. Dissertation. Massachusetts Institute of Technology.
- [3] A Ramesh Babu and N Ramesh Babu. 2001. A generic approach for nesting of 2-D parts in 2-D sheets using genetic and heuristic algorithms. *Computer-Aided Design* 33, 12 (2001), 879–891.
- [4] Roberto Baldacci, Marco A Boschetti, Maurizio Ganovelli, and Vittorio Maniezzo. 2014. Algorithms for nesting with defects. *Discrete Applied Mathematics* 163 (2014), 17–33.
- [5] Patrick Baudisch and Stefanie Mueller. 2017. Personal Fabrication. *Foundations and Trends in Human-Computer Interaction* 10, 3-4 (2017), 165–293. <https://doi.org/10.1561/1100000055>
- [6] Patrick Baudisch, Arthur Silber, Yannis Kommana, Milan Gruner, Ludwig Wall, Kevin Reuss, Lukas Heilman, Robert Kovacs, Daniel Rechltz, and Thijis Roumen. 2019. Kyub: A 3D Editor for Modeling Sturdy Laser-Cut Objects. In *ACM CHI Conference on Human Factors in Computing Systems (CHI)*. 1–12.
- [7] Julia A Bennell, Kathryn A Dowsland, and William B Dowsland. 2001. The irregular cutting-stock problem—a new procedure for deriving the no-fit polygon. *Computers & Operations Research* 28, 3 (2001), 271–287.
- [8] Dustin Beyer, Serafima Gurevich, Stefanie Mueller, Hsiang-Ting Chen, and Patrick Baudisch. 2015. Platener: Low-Fidelity Fabrication of 3D Objects by Substituting 3D Print with Laser-Cut Plates. In *ACM CHI Conference on Human Factors in Computing Systems (CHI)*. 1799–1806.
- [9] Gary Bradski et al. 2000. The OpenCV library. *Dr. Dobb's Journal of Software Tools* (2000).
- [10] Edmund Burke and Graham Kendall. 2002. A New Approach to Packing Non-Convex Polygons Using the No Fit Polygon and Meta-Heuristic and Evolutionary Algorithms. In *Adaptive Computing in Design and Manufacture*.
- [11] Edmund K Burke, Robert SR Hellier, Graham Kendall, and Glenn Whitwell. 2007. Complete and robust no-fit polygon generation for the irregular stock cutting problem. *European Journal of Operational Research* 179, 1 (2007), 27–49.
- [12] Karen McIntosh Daniels and Victor J Milenkovic. 1995. *Containment algorithms for nonconvex polygons with applications to layout*. Ph.D. Dissertation. Harvard University.
- [13] Matteo Fischetti and Ivan Luzzi. 2009. Mixed-integer programming models for nesting problems. *Journal of Heuristics* 15, 3 (2009), 201–226.
- [14] Joerg Heistermann and Thomas Lengauer. 1995. The nesting problem in the leather manufacturing industry. *Annals of Operations Research* 57, 1 (1995), 147–173.
- [15] Florian Heller, Jan Thar, Dennis Lewandowski, Mirko Hartmann, Pierre Schoonbrood, Sophie Stönnér, Simon Voelker, and Jan Borchers. 2018. CutCAD - An Open-source Tool to Design 3D Objects in 2D. In *Designing Interactive Systems (DIS)*. 1135–1139.
- [16] Jennifer Jacobs and Leah Buechley. 2013. Codeable Objects: Computational Design and Digital Fabrication for Novice Programmers. In *ACM CHI Conference on Human Factors in Computing Systems (CHI)*. 1589–1598.
- [17] David S Johnson. 1973. *Near-optimal bin packing algorithms*. Ph.D. Dissertation. Massachusetts Institute of Technology.
- [18] Gabe Johnson, Mark Gross, Ellen Yi-Luen Do, and Jason Hong. 2012. Sketch It, Make It: Sketching Precise Drawings for Laser Cutting. In *ACM CHI Conference on Human Factors in Computing Systems (CHI)*. 1079–1082.
- [19] 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.
- [20] Danny Leen, Tom Veuskens, Kris Luyten, and Raf Ramakers. 2019. JigFab: Computational Fabrication of Constraints to Facilitate Woodworking with Power Tools. In *ACM CHI Conference on Human Factors in Computing Systems (CHI)* (Glasgow, Scotland UK) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3290605.3300386>
- [21] James McCrae, Nobuyuki Umetani, and Karan Singh. 2014. FlatFitFab: Interactive Modeling with Planar Sections. In *ACM User Interface Software and Technology Symposium (UIST)*. 13–22.
- [22] José Fernando C Oliveira and José A Soeiro Ferreira. 1993. Algorithms for nesting problems. In *Applied Simulated Annealing*. Springer, 255–273.
- [23] Thomas Oster. 2011. *Visicut: An application genre for lasercutting in personal fabrication*. Bachelor's Thesis. RWTH Aachen Univ.
- [24] Jack Qiao. 2020. Deepnest - open source nesting software. <https://deepnest.io/>.
- [25] Troels A Rasmussen and Timothy Merritt. 2018. ProjecTables: Augmented CNC tools for sustainable creative practices. *International Journal of Architectural Computing* 16, 3 (2018), 227–242.
- [26] Daniel Saakes, Thomas Cambazard, Jun Mitani, and Takeo Igarashi. 2013. Pac-CAM: Material Capture and Interactive 2D Packing for Efficient Material Usage on CNC Cutting Machines. In *ACM User Interface Software and Technology Symposium (UIST)*. 441–446.
- [27] Greg Saul, Manfred Lau, Jun Mitani, and Takeo Igarashi. 2011. SketchChair: An All-in-one Chair Design System for End Users. In *ACM International Conference on Tangible, Embedded and Embodied Interaction (TEI)*. 73–80.
- [28] The CGAL Project. 2020. *CGAL User and Reference Manual* (5.0.2 ed.). CGAL Editorial Board. <https://doc.cgal.org/5.0.2/Manual/packages.html>
- [29] Udayan Umapathi, Hsiang-Ting Chen, Stefanie Mueller, Ludwig Wall, Anna Seufert, and Patrick Baudisch. 2015. LaserStacker: Fabricating 3D Objects by Laser Cutting and Welding. In *ACM User Interface Software and Technology Symposium (UIST)*.
- [30] K. Winge, R. Hauggaard, and T. Merritt. 2014. VAL: Visually Augmented Laser cutting to enhance and support creativity. In *IEEE International Symposium on Mixed and Augmented Reality - Media, Art, Social Science, Humanities and Design (ISMAR-MASH'D)*. 31–34.
- [31] Nur Yildirim, James McCann, and John Zimmerman. 2020. Digital Fabrication Tools at Work: Probing Professionals' Current Needs and Desired Futures. In *ACM CHI Conference on Human Factors in Computing Systems (CHI)*. 1–13.
- [32] Clement Zheng, Ellen Yi-Luen Do, and Jim Budd. 2017. Joinery: Parametric Joint Generation for Laser Cut Assemblies. In *ACM SIGCHI Conference on Creativity and Cognition (C&C)*.