Kyub: a 3D Editor for Modeling Sturdy Laser-Cut Objects

Patrick Baudisch, Arthur Silber, Yannis Kommana, Milan Gruner, Ludwig Wall, Kevin Reuss, Lukas Heilman, Robert Kovacs, Daniel Rechlitz, and Thijs Roumen
Hasso Plattner Institute at the University of Potsdam
Potsdam, Germany
firstname.lastname@hpi.de

Figure 1: A selection of objects created using *kyub,* a software system that allows users to design 3D objects for laser cutting. By affording closed box structures, objects made using kyub are very strong. This allows users to make tables, shelves, and chairs that can hold a person. (All shown objects are assembled from 4mm plywood sheets—pressure fit, not glued).

ABSTRACT

We present an interactive editing system for laser cutting called kyub. Kyub allows users to create models efficiently in 3D, which it then unfolds into the 2D plates laser cutters expect. Unlike earlier systems, such as FlatFitFab, kyub affords construction based on closed box structures, which allows users to turn very thin material, such as 4mm plywood, into objects capable of withstanding large forces, such as chairs users can actually sit on. To afford such sturdy construction, every kyub project begins with a simple finger-joint "boxel"—a structure we found to be capable of withstanding over 500kg of load. Users then extend their model by attaching additional boxels. Boxels merge automatically, resulting in larger, yet equally strong structures. While the concept of stacking boxels allows kyub to offer the strong affordance and ease of use of a voxel-based editor, boxels are not confined to a grid and readily combine with kuyb's various geometry deformation tools. In our technical evaluation, objects built with kyub withstood hundreds of kilograms of loads. In our user study, non-engineers rated the learnability of kyub 6.1/7.

ACM ISBN 978-1-4503-5970-2/19/05…$15.00

https://doi.org/10.1145/3290605.3300796

CCS CONCEPTS

• Human-centered computing~Human computer interaction

KEYWORDS

Personal fabrication; laser cutting; interactive editing.

ACM Reference format:

Patrick Baudisch, Arthur Silber, Yannis Kommana, Milan Gruner, Ludwig Wall, Kevin Reuss, Lukas Heilman, Robert Kovacs, Daniel Rechlitz, and Thijs Roumen. 2019. Kyub: a 3D Editor for Modeling Sturdy Laser-Cut Objects. In *2019 CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland, UK.* ACM, New York, NY, USA. https://doi.org/10.1145/3290605.3300796

1 INTRODUCTION

A key strength of laser cutting is that it tends to be orders of magnitude faster than other "rapid prototyping" technologies, such as 3D printing [[2]]. While laser cutting used to require designers to be tech-savvy enough to break down their 3D designs into 2D plates, *FlatFitFab* [[7]] allowed users to model in 3D and convert to 2D plates automatically. This allowed users to design and fabricate objects quickly, such as the glass holder shown in Figure 21a (see section "Related Work").

Unfortunately, however, objects produced using *FlatFitFab* are of limited stability, as the system is limited to models consisting of cross sections held together by *notch* joints (aka *cross* joints, *slotted* joints or *edge lap* joints). Even under ideal conditions, objects created using *FlatFitFab* were shown to withstand only forces in the hundreds-of-grams range (Figure 21b). This limits this approach to objects that users may want to look at, but probably will not physically interact with. Similar constraints apply to *Slices* [[6]], which automatically subdivides 3D models into grids of such notch joints (also in the *Slicer of Fusion360* [[33]]).

In this paper, we propose a different approach to designing laser cut 3D objects. Our system, called *kyub*, allows users to create 3D objects based on closed box structures. All objects shown in Figure 1 are made using kyub and all are capable of withstanding very large forces. The chairs, for example, are strong enough to comfortably hold the weight of a human—despite being made from 4mm plywood, i.e., material 25-50x less stiff than the 12-15mm plates used by other systems, such as *SketchChair* [[19]]. As we show in our "technical evaluation" section, basic structures created using kyub are capable of withstanding forces in the hundreds of kilograms, which makes them several orders of magnitudes stronger than the aforementioned constructions based on notch joints.

At the same time, creating closed box structures tends to be more elaborate than creating structures based on cross sections and notch joints. To allow users to create such structures efficiently, kyub offers a novel interaction principle that allows users to create such structures by clicking together what we call *boxels*. As illustrated by Figure 2a, boxels are small finger-joint boxes that (b, c) automatically merge with each other when stacked. While this provides kyub with the affordance of a voxel-based editor [[11]], the resulting objects are not bound to a grid, so that users can (d) reshape kyub objects using a range of deformation tools. This way, the boxel interaction technique is designed to afford shapes that remain box-like whenever possible, because such structures tend to be sturdiest (and easiest to assemble)

In our user study, non-expert participants rated the learnability of this approach to 3D modeling as 6.1/7 (see section "User Study").

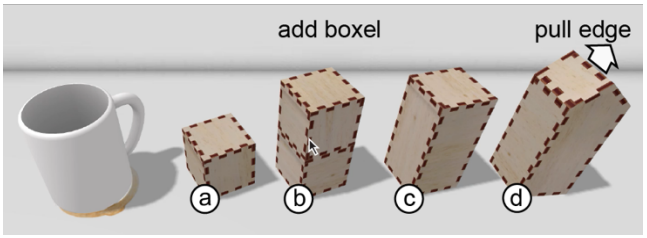


Figure 2: Kyub allows users to create sturdy objects by stacking volumetric elements, which we call boxels. (a) A single boxel can withstand >500kg of load, (b) Added boxels merge automatically, resulting in a larger, yet equally strong structure. (c) While kyub offers the affordance of a voxel-based editor, its objects are not bound to a grid; users can reshape them using a wide range of deformation tools.

2 3D EDITING BASED ON BOXELS

Figure 3 shows a "hello world" example in which we create a simple picture frame, essentially only using kyub's *add boxel tool*. (a) Any editing experience starts with a *boxel* dropping from above. This boxel, like any object in kyub, is represented in a realistic way, i.e., how it will look once laser cut and assembled [[1], [12]]. The boxel bounces off the ground and comes to a rest, demonstrating that this is a physics-based environment. A cup serves as size reference.

(b) The user picks the *add boxel tool* from the menu and clicks into the scene. (c) This produces another boxel. (d) The user selects the *add boxel tool* again, but this time clicks onto a boxel already on stage. The new boxel automatically aligns itself with the clicked one and both merge automatically. This results in a single box the size of two boxels. This merging is an important step in that the resulting geometry not only features a minimal number of plates, but also makes interlocking plates extend across the entire objects; this produces very *sturdy* structures.

By double clicking the *add boxel tool*, the user makes the tool "sticky". (e) Another six clicks cause the *add boxel* tool to create (f) a simple, but stylish picture frame. (g) The user engraves images on each of the six sides of the lone boxel and is now ready to showcase it in the picture frame.

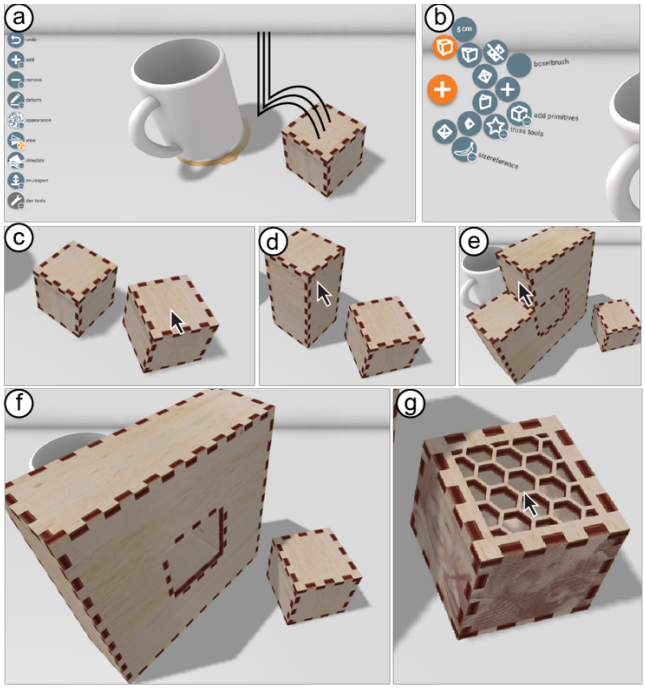


Figure 3: Sole use of the add boxel tool already allows making simple objects, here a picture frame. (a) A boxel falls into the scene. (b) The user selects add boxel and (c) clicks the stage, which produces a second boxel. (d) Holding the add boxel tool, the user clicks a boxel already on stage. This stacks a boxel on top and both merge automatically. (e) Adding another six boxels (f) completes the frame. (g) Engraving six images into the lone boxel prepares it for being displayed in the frame.

The user now exports the box to the laser cutter using kyub's *export* menu. Kyub responds by breaking the 3D model down into individual plates and by correcting for the specifics of the target machine (*kerf* correction [[30]]). As shown in Figure 4, it adds engraved marks that tell the user which plates to connect when assembling the model. Kyub then lays out plates onto sheets (also known as *nesting*, see also [[10], [23]]) and writes each sheet into a separate .svg file.

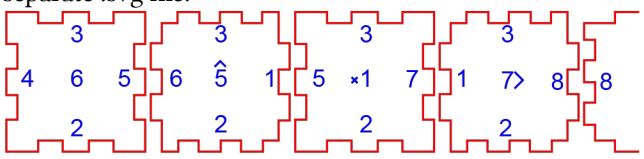


Figure 4: An exported model from Kyub, the red lines are what the laser cutter cuts. Numbers along the edges tell users which plates (numbers in centers) to connect to, when assembling the model. A '^' indicates "this side up"; an 'x' indicates placement at the bottom.

The user sends the .svg file(s) to the laser cutter and assembles the fabricated parts. Figure 5 shows the final result, here with manually sanded edges.

Figure 5: A fabricated, assembled, and sanded picture frame.

Boxels appear to be on a grid—but they are not

The voxel-like behavior of *add boxel* suggests boxels be limited to a grid. This would be a dramatic limitation, as it would eliminate many of the benefits of personal fabrication—which is to create one-off objects that fit a specific use case. Fortunately, kyub boxels *are not* confined to a grid.

As shown in Figure 6, kyub allows users to manipulate boxel-based geometry with a range of deformation tools. For example, (a-b) the user may compress the picture frame using the *push-pull tool*. Or (c) users may pull out just one of the edges to give the picture frame a slanted top or base.

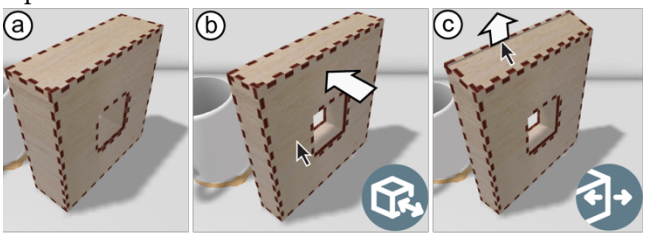


Figure 6: (a) Kyub allows the boxel-based picture frame to be manipulated using (b) *push/pull* and (b) *push/pull edge tools.*

Interestingly, *add boxel* continues to be applicable after geometry has been deformed. In Figure 7, we set boxels to half their usual size and then (a-b) apply *add boxel* to produce two prongs that (c) form a stand for our picture frame. *Add boxel* not only remains applicable, but also snaps into an invisible grid, making it easy to align the prongs with the frame.

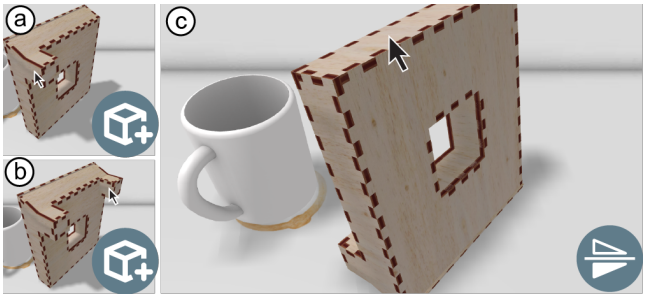


Figure 7: *Add boxel* remains applicable *after* the use of deformation tools.

Kyub's secret to achieving this combination of grid and free deformation is to *not memorize* any grid, but to instead *re-infer* the grid with every tool interaction, i.e., kyub analyzes the current geometry and determines what grid *the user* might be trying to refer to.

Figure 8 further illustrates this. (b) Removing half a boxel on one side of a part and (c) adding it back at the other end looks like it might lead to a grid aligned half-way. (d) However, boxels added now snap into position based on the current shape of the part, not based on its history. We describe the underlying algorithm in detail in the "Implementation" section.

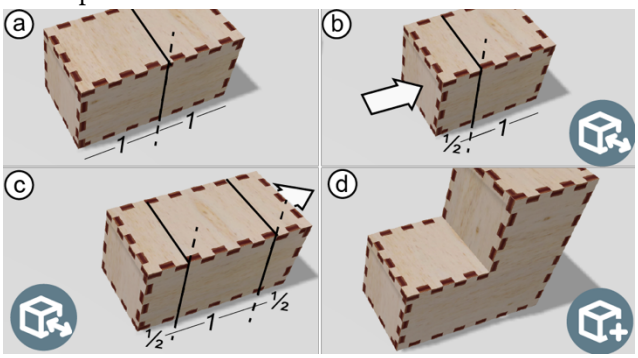


Figure 8: Kyub infers the grid, rather than maintaining it (a) Laying down two boxels, (b) pushing in half a boxel and (c) pulling out half a boxel on the other side results in 2x1 boxel arrangement. (d) Adding a boxel snaps into position based on the current shape of the part, not its history.

Kyub's ability to maintain grid-like behavior allows it to offer good default behavior when placing new geometry. This is key, for example, when running kyub on devices with very coarse input capabilities, such as touch screens (Figure 9).

The fact that kyub determines the currently active "grid" only based on the object's current and thus visible geometry generally relieves kyub of the necessity to display the grid. This allows kyub to adopt a particularly uncluttered and natural look.

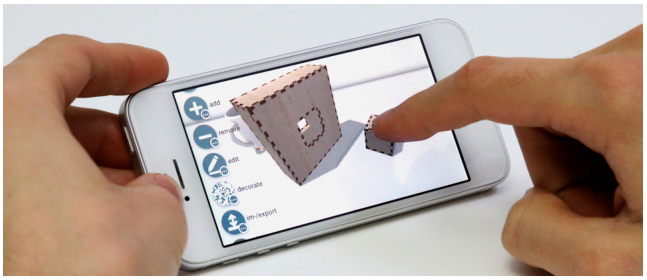


Figure 9: The grid-like behavior of boxels is crucial in allowing kyub users to create precise geometry on low-precision input devices [[26]] (here an *iPhone SE*).

Making things that fit together

One of the desirable side effects of the boxel-based approach is that everything naturally fits together. Figure 10 illustrates this at the example of a decorative robot figurine, created by one of the participants in our user study (see section "User Study"). Here the user (a) has modeled the foot of the robot and then cuts two holes into it using the *subtract boxel tool*. (b) The fact that boxel-sized extrusions naturally fit into boxel-sized holes allows the two parts to form a dowel-like connection. Here the user uses the *insert tool* to try this out in the editor. By default, kyub designs all parts with a "fixed fit" (H7/n6) [[5]], allowing parts to be mounted and removed with a light pressing force. (c) The dowel connections allow the resulting figurine to be posed (d-e) in a variety of ways.

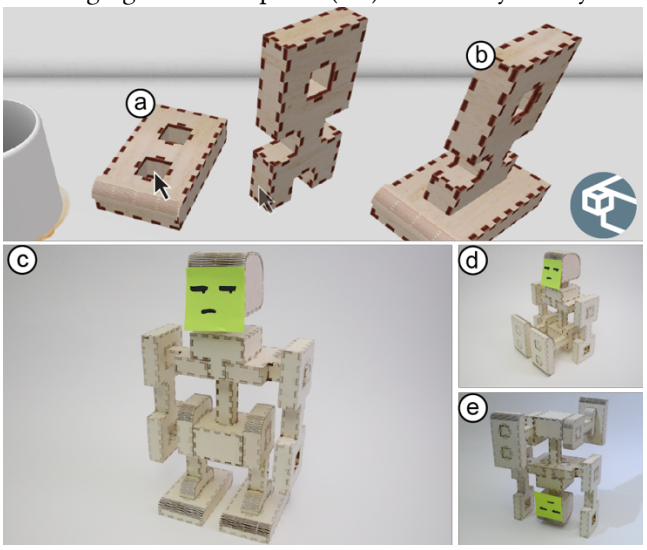


Figure 10: Parts created using kyub naturally fit together. (a) Parts with extruding boxels and parts with boxel-shaped hole naturally match and (b) users can combine them using the insert tool. (c) The resulting dowel joints allow this decorative robot figurine (d, e) to be posed in various ways.

As illustrated by Figure 11, boxel-based dowel connections allow us to create collections of parts that can be combined interchangeably. The result is a custom construction kit that allows making a range of different models, a rudimentary version of something like a LEGO construction kit.

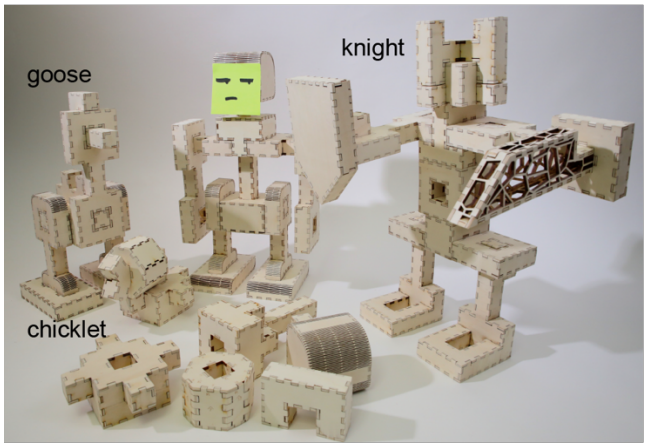


Figure 11: Combining the parts of the robot figurine with a few compatible elements results in a simple construction kit.

While the shown models are clearly designed around the notion of rectilinearity, they also include some non-rectilinear parts. As shown in Figure 12, these were made quickly and efficiently by applying *non-rectilinear* boxels, here specifically *90-degree prisms* and *equilateral prisms.*

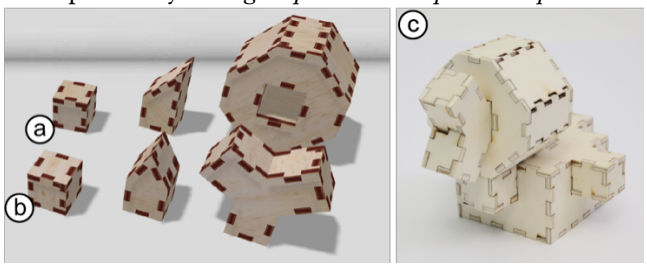


Figure 12: Non-rectilinear boxels add expressiveness to boxel-based construction. (a) Here we use four 90-degree prisms to create a duckling's head and (b) one equilateral prism to make its beak. (c) Resulting duckling.

The concept of non-rectilinear boxels extends past prisms. Figure 13 shows how we recreated the well-known *tetrahedron puzzle* by complementing (a) a *pyramid boxel* with (b) two *tetrahedron boxels.* Cloning the resulting shape completes the puzzle and (c) and with a little bit of trying out it can be assembled into… a tetrahedron.

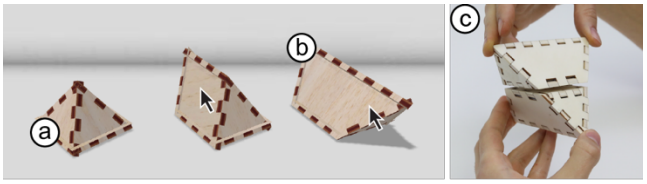


Figure 13: The boxel concept goes beyond rectilinear boxes. The pieces of this tetrahedron puzzle were made by combining a pyramid boxel with two tetrahedron boxels.

3 DESIGNING STURDY STRUCTURES

As discussed earlier, a key objective behind boxel-based construction is to create *sturdy* structures. The chair shown in Figure 14 is such a structure. This particular model was created using the boxel-based workflow described above, sped up by using a *clone* and an *attach tool.*

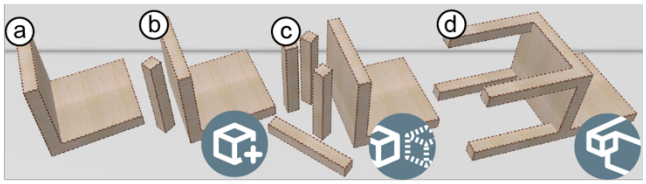


Figure 14: (a, b) Modeling a chair in kyub using *add boxel* (c) the *clone tool* and (d) the *attach tool.*

The common approach to designing furniture is to use thicker material, such as 12mm plywood [[19], [24]] and as shown in Figure 15, kyub supports arbitrary material thicknesses. We could make the chair from 6mm plywood, for example, and even though plates of half thickness should carry only an eighth of the weight, the result works reliably thanks to the box-based construction.

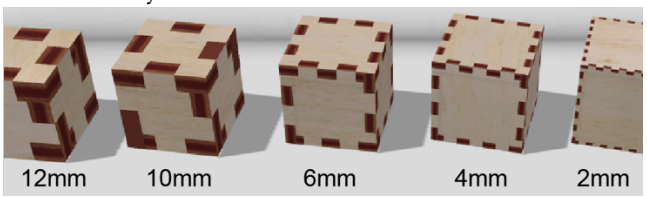


Figure 15: Kyub supports material thicknesses.

Figure 16a shows how an experienced user can push the design of our chair design one step further in order to allow manufacturing it from the same 4mm plywood we used in all previous examples. By adding two *internal* plates that extend through the seat and the backrest using the *reinforcement tool*, this design prevents the seat from buckling and the backrest from breaking, despite the thin material. The design shown in Figure 16b is even resilient against rocking, as the insides of pairs of legs are combined into contiguous U-shaped plates.

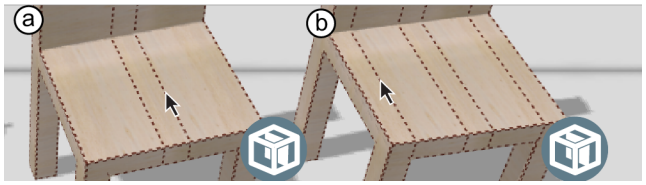


Figure 16: (a) Chair with front-to-back reinforcement, (b) additional reinforcement supporting the legs.

In order to get the most strength out of internal plates, kyub merges internal plates with other plates whenever possible. As shown in Figure 17a, internal plates are usually centered, but that would prevent the internal plates in Figure 16b from merging with the legs. However, as shown in Figure 17b kyub's internal plates are given some slack, allowing them to snap into the plane of the leg. Kyub heavily relies on its grid inferrer for this functionality (see "Implementation" section for algorithmic details).

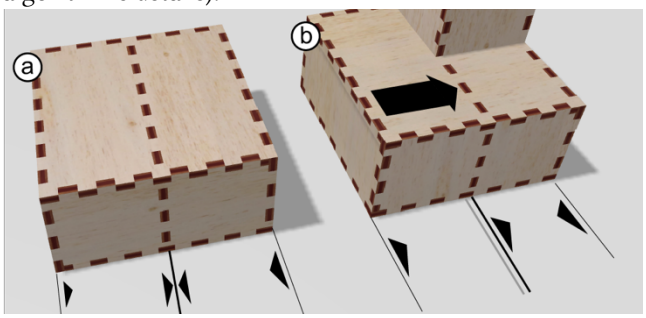


Figure 17: (a) When applying the reinforcement tool to this part, the reinforcement centers itself, (b) However, when adding a boxel, reinforcement automatically shifts by half a plate thickness so as to line up with the left plate of the added boxel, producing a sturdier result.

Finally, Figure 18 shows an expert design that solves the challenge without any internal plates. Instead, a slot cut into seat and backrest using the *subtract boxel* tool reinforces seat and backrest—and creates what we think of as an appealing design element.

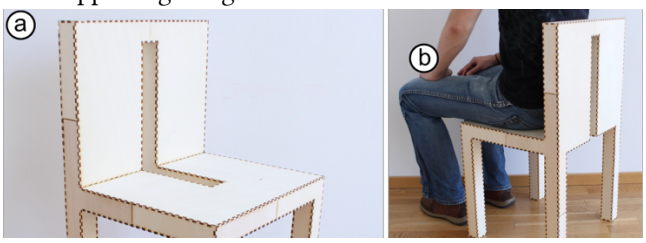


Figure 18: Here a slot cut into the chair reinforces the chair's seating surface and backrest.

Making large objects using tessellation

The closed-box structures afforded by kyub are generally strong enough to produce sturdy objects even at large scale. While some laser cutters allow cutting very large objects in one go, kyub allows owners of smaller devices to fabricate large objects as well.

Kyub achieves this by composing large objects from smaller plates. Users configure the maximum plate size available to them and when enlarging an object now, kyub breaks down the oversized part into two or more *cells as shown in* Figure 19a. Figure 19b shows the specific wood joint kyub creates to hold cells together—a *supported lengthening joint*) Adjacent cells share a single membrane of finger joints; then both cells connect into this membrane using butterfly joint-like tabs.

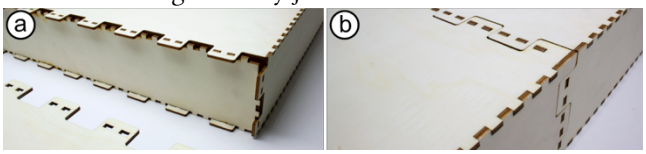


Figure 19: The cell structure created by tessellation. The big finger joints lock the two coplanar plates on the top while supported by a vertical plate.

The joined cells are strong enough to carry human weight even on very large designs, such as the 1.80m dining room table shown in Figure 20, assembled from 40 "A2"-size plates (60 x 40cm; 24" x 16").

Figure 20: The table from Figure 1 is assembled of separate cells which are capable of holding a human sitting on it.

4 CONTRIBUTION, BENEFITS, & LIMITATIONS

Our first contribution is the actual system that allows users to model sturdy laser-cut objects. Kyub achieves sturdy construction by affording closed box structures, which allows users to make objects capable of withstanding large forces, such as chairs users can actually sit on, with regular laser cutters and from materials substantially thinner than those used in the related work.

Our second contribution is the concept of 3D editing by stacking 3D primitives that merge automatically. While this allows kyub to offer the strong affordance and ease of use of a voxel-based editor, boxels are not confined to a grid and readily combine with kuyb's geometry deformation tools. By re-inferring a grid, rather than memorizing it, boxel tools remain applicable even after deformation.

We provide both a technical evaluation and a user study. Our technical evaluation demonstrates that objects built with kyub withstand hundreds of kilograms of loads. In our user study, non-engineer participants reported very high satisfaction with the models they had made using kyub (6.4/7 on a Likert scale). They also rated the learnability of our system 6.1/7.

Our current prototype is limited in that its functionality almost exclusively targets boxel-based construction; in contrast, more traditional construction elements, such as freestanding plates have only rudimentary support. Furthermore, the presented tools tend to impose a certain "boxy" aesthetics onto people's designs.

5 RELATED WORK

Kyub builds on related work in structural engineering, modeling in 3D, and 3D editors for laser cutting.

Structural engineering based on boxes

The concept of building from perpendicular plates is prominent in US American home construction, where box-like constructions are held together by wooden frames joined using metal connectors in the typical "platform framing" construction method [[14]]. Sass et al. demonstrate how this method can be used in combination with digital fabrication for rural areas [[18]].

Closed box structures can also be found prominently in the furniture industry, such as IKEA. Such furniture tends to be based on particleboard plates [[16]] reinforced with a cardboard honeycomb filling.

Structural constructions and analysis in HCI

TrussFab [[13]] supports users in building sturdy large structures by allowing them to combine 3D print with PET bottles. The resulting structures are sturdy, because they contain closed triangle structures, also known as *trusses*.

Yao et al. [[29]] analyze the structural integrity of the decorative joints in their furniture designs. Furthermore *MatchSticks* allows users to build sound constructions directly using guided woodworking [[22]].

Modeling directly in 3D

Williams originally introduced the idea of modeling based on volumetric primitives (aka voxels) with 3D Paint [[28]]. *Foundry* [[25]] allows users to specify shape using voxels that are complemented by code that defines material behavior. The concept of composing scenes from voxel-like elements is popular in game design (e.g., *Minecraft* [[34]]). *Minecraft* being close to Kyub in expressiveness, shows the tremendous potential of what can be built with simple boxes. Kyub allows users to turn such creations into real fabricated objects, experienced users benefit from the *grid inferrer* to modify geometry past the grid.

Teddy [[10]] is a sketch-based 3D editor that generates 3D shape automatically from 2D input gestures, similar to *SKETCH* which focuses on general purpose 3D sketching [[31]]. *PARTS* [[9]] and *Grafter* [[17]] use mechanisms as primitives to build functional machines.

3D editors for laser cutting

Traditionally, users create contents for laser cutters by describing the involved plates in 2D. This concept is still popular, as it offers the highest level of control. *Constructable* [[15]], for example, allows users to construct the parts and joints required for 3D geometry directly inside the laser cutter. *Joinery* [[32]] helps users create joints connecting laser-cut plates with a variety of joining mechanisms.

The downside of the 2D approach is that it requires users to perform the 3D-to-2D conversion in their heads. To evade this, researchers created various systems that allow users to design models for laser cutting directly in 3D.

Platener [[3]] allows users to create models for laser cutting by converting 3D models initially designed for 3D printing. *Platener* works best with models containing plate-like geometry. Similarly, *CofiFab* [[20]] fills in 3D models with geometry that can be lasercut, with a special focus on objects' decorative qualities. *CutCAD* [[8]] lets users interact in 2D, while it allows users to preview their compositions in 3D.

Other systems tackle the challenge of 3D construction for laser cutting by considering specific domains, such as enclosures (*enclosed* [[27]] and to a certain extent *make-a-box* [[35]]) or mechanically actuated characters [[4]].

FlatFitFab [[7]] allows users to create laser cut 3D objects from linked flat dissections, which allows creating very good approximations of shape. In their evaluation, the glass holder shown in Figure 21 held up to 275 grams.

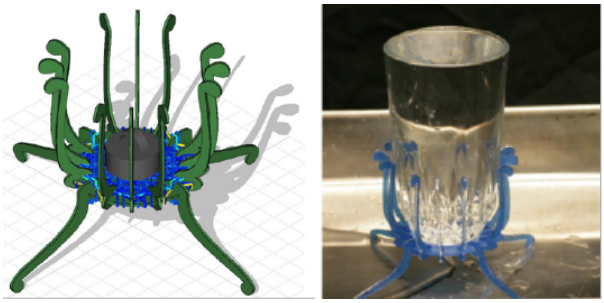


Figure 21: A glass holder made using *FlatFitFab* [[7]] holding up a glass of water weighing 275 grams.

SketchChair [[19]] allows users to create chairs users can sit on. However, SketchChair requires 12mm-15mm plates in order to achieve stability. Schwartzburg and Pauly [[21]] use a similar construction method (notch joints) as FlatFitFab, their algorithm ensures interlocking of the joints which allows them to scale to furniture sized objects given sufficiently thick materials (10mm+ MDF).

6 TECHNICAL EVALUATION

To validate the strength of closed box structures underlying kyub, we conducted a technical evaluation during which we fractured seven types of structures by applying appropriate subsets of up to six different types of forces and measured the force required.

To obtain a conservative lower bound of the sturdiness of the tested objects, we used the weakest and cheapest material we could find, i.e., 4mm 3-layer plywood at a price of roughly 7 Euros/m^2. Also, all objects were held together by only press fitting them, i.e., without glue.

Objects tested

Figure 22 shows the objects we tested. The first four objects allowed us to test basic boxel geometry: (a) a 5cm boxel, (b) a 35cm stick made from 7 boxels, (c) an L-shape made from 3 boxels, (d) a 3D L-shape made from 4 boxels.

The next two models allowed us to test reinforcement. (e) The star consisted of 7 boxels, but was internally reinforced using three pairs of parallel plates. (f) The dumbbell consisted of two 3x3x3 boxels at each end, connected by two boxels in the middle. However, we used reinforcement to extend the center portion into both 3x3x3 boxels.

(g) The final model allowed us to test tessellation. This model was another 7-boxel stick. However, each plate was subdivided into two interlocking plates.

Procedure

To administer the test, we mounted test objects into the custom testing apparatus as shown in Figure 23. The apparatus was essentially a custom vise made from aluminum profiles (from *item Inc.*). We actuated the device by tightening a nut on a threaded rod until the test object would break. A properly placed force sensor (*forceX 2.30*) measured the applied forces.

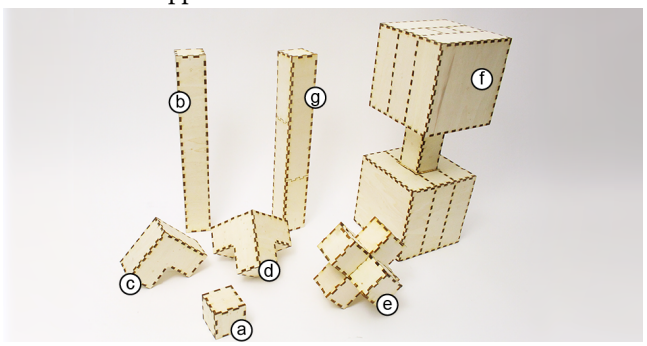


Figure 22: Objects tested

We used the apparatus to apply (a) compression along the object's main Cartesian axis, (b) compression against the object tilted by 45 degrees, and (c) compression against the object tilted along two axes. (d) We measured tension by pulling the test object with pairs of straps, and (e) torsion, by holding the test object in place using clamps while twisting the opposite end using a lever. (f) Finally, we measured buckling, by applying a force to test objects at three points.

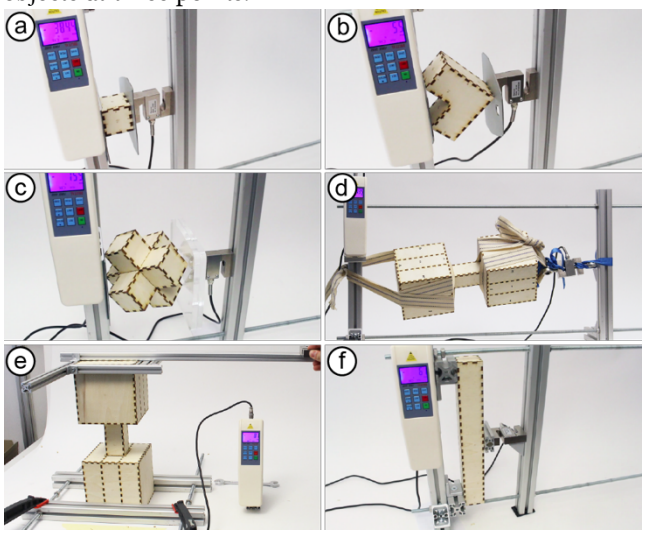
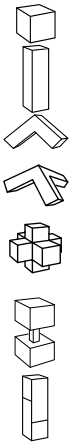


Figure 23: The test apparatus.

Results

Figure 24 shows the main results, i.e., amount of force or torque required to fracture the respective objects. “>500kg” means that the test object was still intact when we exceeded the 500kg value range of our measuring device. “>140kg” indicates that the test object was still intact when our test apparatus started to collapse when applying tension to the dumbbell model.



compression			tension	torsion	buckle
from top	tilted	...2D			
>500 kg					
>500 kg				49Nm	135kg
	293 kg				
		91kg			
>500kg	>500kg	398kg			
			>140kg	40Nm	230kg
>500 kg				38Nm	188kg

Figure 24: Forces required to break the respective object.

7 USER STUDY

We evaluate the usability of our system with a user study in which non-engineer participants designed 3D models using kyub, then cut and assembled them. Participants filled in a questionnaire about their experience. We hypothesize that participants find kyub easy to learn and use.

Task

We asked all teams to create a roughly foot-high “persona” figurine for future “design thinking” sessions.

Format

While users spent only 90 minutes with kyub, we conducted our evaluation as part of a two half-day’s workshop on laser cutting, a format that gave us time to learn about participants’ experience. Participants were in the same space at the same time, as well as several team members. They worked in self-selected teams of two.

During the first day, we gave participants a 2h introduction to the traditional process of laser cutting. During this period participants created their first laser cut designs i.e., figurines to serve as personas for future design thinking activities, which they drew directly in *Adobe Illustrator* or *Inkscape*. Participants added notch joints manually by overlaying rectangles of appropriate width taken from a template onto their designs.

We then showed a demo of kyub and gave participants 90 minutes to design their own models in kyub.

We laser cut the objects offline and participants reconvened a week later for the second half-day session during which they assembled their models and learned more about personal fabrication. Finally, participants filled in a questionnaire.

Participants

We recruited 18 participants (8 female) with an average age of 35 years. They were part of the design curriculum at an affiliated institution. Few participants (four) had used a laser cutter before. The workshops took place in two rounds, one with 10 participants and one with 8.

Results

All teams succeeded at modeling their personas. Figure 25 shows the resulting designs. Two teams who finished early made additional models, i.e., an advent calendar and the name of their institution in 3D characters.

All participants reported high satisfaction with the models they had made using kyub (6.4/7 on a Likert scale).

Participants rated their enjoyment of using kyub high (5.8/7). In particular, participants report being pleased with the sturdiness of the models (6.6/7). Participants strongly agreed that kyub had helped them create models they could not make before (6.5/7) and strongly indicated that it would be time-consuming to make these models without kyub (6.9/7).

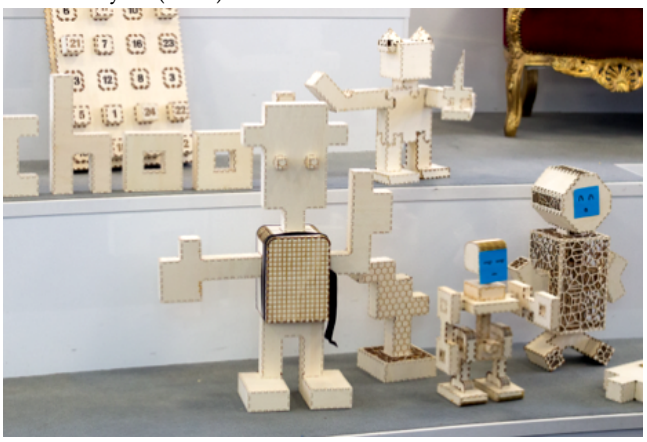


Figure 25: Participant teams created figurines to function as storytelling personas. One team used the remaining time to make an advent calendar one created the name of their institution in 3D characters.

Participants liked the physics simulation in the editor (6.1/7) but said they felt it would be useful to temporarily disable it. P7 explained "gravity helped me model, but once it tipped over it was hard to get it back up".

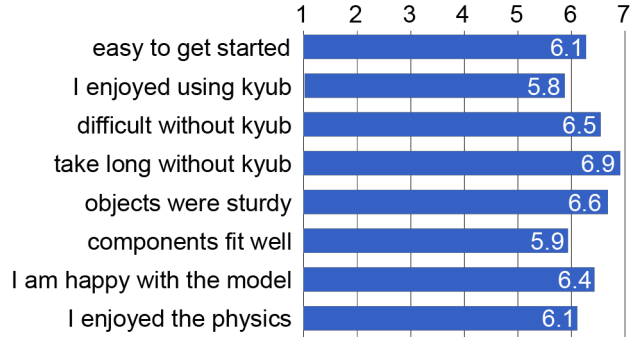


Figure 26: Questionnaire results.

In general, participants found it easy to get started using kyub (6.1/7). Interestingly, some participants credited the physics engine for this. P3: "because everything looks and behaves like the final result, including gravity, it was very easy to understand what was going on". P1: "I just loved how that cube fell in the scene at first, it encouraged me to try things out and model in a playful way with this editor as opposed to Blender which I used before!").

Three participants mentioned appreciating the modularity of kyub. Five participants reported that their favorite feature was the realistic look and feel of the editor. Three mentioned that they liked it best that kyub afforded the creation of sturdy objects.

The overall excitement during the workshop was high and two of the teams approached us later on asking for permission to use the software for making prototypes of their regular projects.

8 IMPLEMENTATION

Kyub is implemented as a web-based application using JavaScript and WebGL. It consists of about 70k lines of code. The server runs in Node.js but almost all computation is done on the client, making it easily scalable. The architecture of the client is similar to that of a game engine, because of the different subsystems at play, such as the physics engine (from cannonjs.org) that is continuously running in the background.

Core to the implementation is the *grid inferrer*. As introduced before, it provides alignment when adding a boxel to an existing assembly. Figure 27 shows an example, i.e., a user adding a boxel to a somewhat irregular "base" geometry.

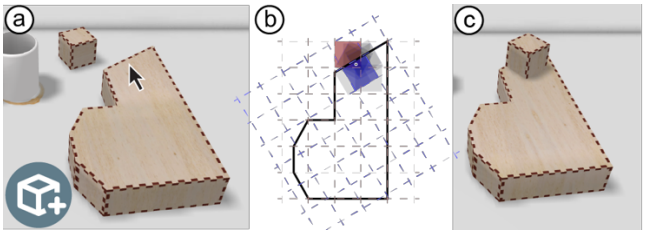


Figure 27: The grid inferrer. (a) a user applies the *add boxel tool* at the shown location. (b) Kyub takes the projection of the clicked surface and infers all possible grids to which the boxel could be aligned as shown in Algorithm 1. (c) After weighing the different grids, it places the boxel and merges the geometry.

When the user releases the mouse button (Figure 27a), kyub passes the mouse-up location to the grid inferrer. Implementing the algorithm shown below, the grid inferrer traverses all edges in the base geometry and extends each edge into a grid. The blue grid shown in Figure 27b, for example, resulted from the base geometry's top edge. The algorithm now reduces each grid to the one cell that contains the mouse click location. Finally, the algorithm picks the cell that maximizes the number of edges supporting this grid, minimizes the distance between these edges and the mouse click location, and minimizes the resulting number of plates.

ALGORITHM 1: GRID INFERRER

Input: clicked point P
2D projection of the outline of the base B
2D projections of the outline of the connectors C
Output: connector c^*, position p^* and rotation r^* of object aligned to (and placed on) B
Parameters: CENTER, IGNORE_PROTRUDING, WEIGHTS
candidates ← new Array()
for each *connector* ∈ C ***do***
  for each pair of edges (e_B, e_C) ∈ *edges(B)* × *edges(C)*
    alignments ← position and rotation to align start or end of e_C with either start or end of e_B respectively
    if CENTER **then**
      add to alignments the position and rotation to align center of e_C with the center of e_B
    end
    for each *alignment* ∈ *alignments* ***do***
      apply *alignment* to *connector*
      G ←create grid by repeating the bounding box (AABB) of *connector*
      candidate ← grid cell of G that contains P

```
        if IGNORE_PROTRUDING then
            add candidate to candidates if it is fully
            overlapping with B
              else
                    add candidate to candidates
              end
        end
          candidate.connector ← connector
          candidate.feature_distance ← d(P, e_B)
          candidate.mergeable_plates ← the number of
          coplanar plates after placement (see Figure 28)
        end
end
deduplicate candidates, count in candidate.duplicates
normalize candidates' criteria, so each is within [0,1]
best ← select the best candidate considering the weights
of the criteria defined by WEIGHTS
c* ← best.connector
r* ← best.rotation
p* ← best.position
```

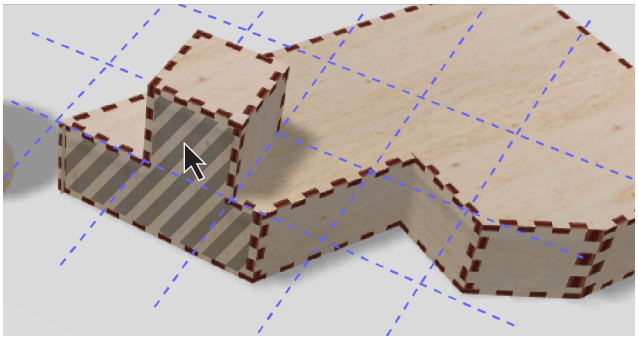


Figure 28: The resulting boxel is merged with the assembly. The dashed surface is an example of a coplanar plate that is unified with the side of the added boxel.

Whenever possible, the grid inferrer tries to place added geometry such that it stays within the circumference of the base geometry. Kyub accomplishes this by calling the grid inferrer with *ignore_protruding* set to *true*. This will produce the desired result in most cases. If the *grid inferrer* returns no result, however, kyub calls the *grid inferrer* again, this time with *ignore_protruding* set to false. This allows the grid inferrer to also explore configurations, where the added geometry protrudes past the edge of the base geometry. This two-pass approach allows kyub to make sure that the *insert tool* works reliably, including such cases as the insertion of the arm of the robot in Figure 10.

9 CONCLUSIONS

We presented kyub, an interactive editing system for laser cutting. Kyub affords construction based on closed box structures, which allows users to make objects capable of withstanding large forces, such as chairs users can actually sit on. Users construct such models by stacking boxels. We have validated our results using a technical evaluation, a user study. As future work, we plan to integrate kyub with plate-based functionality.

REFERENCES

[1] Anand Agarawala and Ravin Balakrishnan. 2006. Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06), Rebecca Grinter, Thomas Rodden, Paul Aoki, Ed Cutrell, Robin Jeffries, and Gary Olson (Eds.). ACM, New York, NY, USA, 1283-1292. DOI: http://dx.doi.org/10.1145/1124772.1124965

[2] Patrick Baudisch & Stefanie Mueller (2017). Personal fabrication. *Foundations and Trends® in Human–Computer Interaction*, *10*(3–4), 165-293.

[3] 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 Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). ACM, New York, NY, USA, 1799-1806. DOI: https://doi.org/10.1145/2702123.2702225

[4] Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W. Sumner, Wojciech Matusik, and Bernd Bickel. 2013. Computational design of mechanical characters. ACM Trans. Graph. 32, 4, Article 83 (July 2013), 12 pages. DOI: https://doi.org/10.1145/2461912.2461953

[5] ISO 286-1:2010 - Geometrical product specifications (GPS) - ISO code system for tolerances on linear sizes - Part 1: Basis of tolerances, deviations and fits. 2010: International Organization for Standardization (ISO).

[6] James McCrae, Karan Singh, and Niloy J. Mitra. 2011. Slices: a shape-proxy based on planar sections. ACM Trans. Graph. 30, 6, Article 168 (December 2011), 12 pages. DOI: https://doi.org/10.1145/2070781.2024202

[7] James McCrae, Nobuyuki Umetani, and Karan Singh. 2014. FlatFitFab: interactive modeling with planar sections. In Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST '14). ACM, New York, NY, USA, 13-22. DOI: https://doi.org/10.1145/2642918.2647388

[8] Florian Heller, Jan Thar, Dennis Lewandowski, Mirko Hartmann, Pierre Schoonbrood, Sophy Stönner, Simon Voelker, and Jan Borchers. 2018. CutCAD - An Open-source Tool to Design 3D Objects in 2D. In Proceedings of the 2018 Designing Interactive Systems Conference (DIS '18). ACM, New York, NY, USA, 1135-1139. DOI: https://doi.org/10.1145/3196709.3196800

[9] Megan K. Hofmann, Gabriella Han, Scott E. Hudson, Jennifer Mankoff. 2018. Greater than the Sum of its PARTs: Expressing and Reusing Design Intent in 3D Models. To be published in 36th Annual ACM Conference on Human Factors in Computing Systems (CHI '18). ACM, New York, NY, USA

[10] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. 1999. Teddy: a sketching interface for 3D freeform design. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques (SIGGRAPH '99). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 409-416. DOI: http://dx.doi.org/10.1145/311535.311602

[11] Tinsley A. Galyean and John F. Hughes. 1991. Sculpting: an interactive volumetric modeling technique. In Proceedings of the 18th annual conference on Computer graphics and interactive techniques (SIGGRAPH '91). ACM, New York, NY, USA, 267-274. DOI: http://dx.doi.org/10.1145/122718.122747

[12] Kimball, Ralph, and B. Verplank E. Harslem. "Designing the Star user interface." *Byte* 7 (1982): 242-282.

[13] Robert Kovacs, Anna Seufert, Ludwig Wall, Hsiang-Ting Chen, Florian Meinel, Willi Müller, Sijing You, Maximilian Brehm, Jonathan Striebel, Yannis Kommana, Alexander Popiak, Thomas Bläsius, and Patrick Baudisch. 2017. TrussFab: Fabricating Sturdy Large-Scale Structures on Desktop 3D Printers. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17). ACM, New York, NY, USA, 2606-2616. DOI: https://doi.org/10.1145/3025453.3026016

[14] Li-Chu Lin 2006 The adaptability of two-by-four wood framing construction. In *Adaptables2006: International Conference On Adaptable Building Structures*. Eindhoven, the Netherlands 2006.

[15] Stefanie Mueller, Pedro Lopes, and Patrick Baudisch. 2012. Interactive construction: interactive fabrication of functional mechanical devices. In Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12). ACM, New York, NY, USA, 599-606. DOI: https://doi.org/10.1145/2380116.2380191

[16] Bo Nilsson 2013. Particle Board *U.S. Patent No. 8,398,905*. Washington, DC: U.S. Patent and Trademark Office.

[17] Thijs Roumen, Willi Mueller and Patrick Baudisch. 2018. Grafter: Remixing 3D Printed Machines. In Proceedings of the 36th Annual ACM Conference on Human Factors in Computing Systems (CHI '18). ACM, New York, NY, USA. DOI: https://doi.org/10.1145/3173574.3173637

[18] Lawrence Sass, and Marcel Botha. "The instant house: a model of design production with digital fabrication." *International Journal of Architectural Computing* 4, no. 4 (2006): 109-123.

[19] Greg Saul, Manfred Lau, Jun Mitani, and Takeo Igarashi. 2010. SketchChair: an all-in-one chair design system for end users. In Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction (TEI '11). ACM, New York, NY, USA, 73-80. DOI: http://dx.doi.org/10.1145/1935701.1935717

[20] Peng Song, Bailin Deng, Ziqi Wang, Zhichao Dong, Wei Li, Chi-Wing Fu, and Ligang Liu. 2016. CofiFab: coarse-to-fine fabrication of large 3D objects. ACM Trans. Graph. 35, 4, Article 45 (July 2016), 11 pages. DOI: https://doi.org/10.1145/2897824.2925876

[21] Yuliy Schwartzburg and Mark Pauly. "Fabrication-aware design with intersecting planar pieces." In *Computer Graphics Forum*, vol. 32, no. 2pt3, pp. 317-326. Oxford, UK: Blackwell Publishing Ltd, 2013. DOI: http://doi.org/10.1111/cgf.12051

[22] Rundong Tian, Sarah Sterman, Ethan Chiou, Jeremy Warner, and Eric Paulos. 2018. MatchSticks: Woodworking through Improvisational Digital Fabrication. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18). ACM, New York, NY, USA, Paper 149, 12 pages. DOI: https://doi.org/10.1145/3173574.3173723

[23] Oster, Thomas. Visicut: An application genre for lasercutting in personal fabrication. *RWTH Aachen Univ* (2011).

[24] Nobuyuki Umetani, Takeo Igarashi, and Niloy J. Mitra. 2012. Guided exploration of physically valid shapes for furniture design. ACM Trans. Graph.31, 4, Article 86 (July 2012), 11 pages. DOI: https://doi.org/10.1145/2185520.2185582

[25] Kiril Vidimce, Alexandre Kaspar, Ye Wang, and Wojciech Matusik. 2016. Foundry: Hierarchical Material Design for Multi-Material Fabrication. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16). ACM, New York, NY, USA, 563-574. DOI: https://doi.org/10.1145/2984511.2984516

[26] Daniel Vogel and Patrick Baudisch. 2007. Shift: a technique for operating pen-based interfaces using touch. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07). ACM, New York, NY, USA, 657-666. DOI: https://doi.org/10.1145/1240624.1240727

[27] Christian Weichel, Manfred Lau, and Hans Gellersen. 2013. Enclosed: a component-centric interface for designing prototype enclosures. In Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction (TEI '13). ACM, New York, NY, USA, 215-218. DOI: http://dx.doi.org/10.1145/2460625.2460659

[28] Lance Williams. 1990. 3D paint. SIGGRAPH Comput. Graph. 24, 2 (February 1990), 225-233. DOI: http://dx.doi.org/10.1145/91394.91450

[29] Jiaxian Yao, Danny M. Kaufman, Yotam Gingold, and Maneesh Agrawala. 2017. Interactive Design and Stability Analysis of Decorative Joinery for Furniture. ACM Trans. Graph. 36, 2, Article 20 (March 2017), 16 pages. DOI: https://doi.org/10.1145/3054740

[30] Yilbas, B. S. "Effect of process parameters on the kerf width during the laser cutting process." *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 215, no. 10 (2001): 1357-1365.

[31] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. 2007. SKETCH: an interface for sketching 3D scenes. *In ACM SIGGRAPH 2007 courses(SIGGRAPH '07)*. ACM, New York, NY, USA, Article 19 . DOI: https://doi.org/10.1145/1281500.1281530

[32] Clement Zheng, Ellen Yi-Luen Do, and Jim Budd. 2017. Joinery: Parametric Joint Generation for Laser Cut Assemblies. In Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition (C&C '17). ACM, New York, NY, USA, 63-74. DOI: https://doi.org/10.1145/3059454.3059459

[33] AutoDesk. 2017. Slicer for Fusion360 https://apps.autodesk.com/FUSION/en/Detail/Index?id=8699194120463301363&os=Win64&appLang=en

[34] Microsoft. 2011. Minecraft https://minecraft.net

[35] Reinvent-One. 2014. Make a Box https://makeabox.io