Kerf-Canceling Mechanisms: Making Laser-Cut Mechanisms Operate Across Different Laser Cutters

Thijs Roumen, Ingo Apel, Jotaro Shigeyama, Abdullah Muhammad, and Patrick Baudisch
Hasso Plattner Institute
Potsdam, Germany
first.lastname@hpi.de

ABSTRACT

Getting laser-cut mechanisms, such as those in microscopes, robots, vehicles, etc., to work, requires all their components to be dimensioned precisely. This precision, however, tends to be lost when fabricating on a different laser cutter, as it is likely to remove more or less material (aka "kerf"). We address this with what we call *kerf-canceling mechanisms*. Kerf-canceling mechanisms replace laser-cut bearings, sliders, gear pairs, etc. Unlike their traditional counterparts, however, they keep working when manufactured on a different laser cutter and/or with different kerf. Kerf-canceling mechanisms achieve this by adding an additional wedge element per mechanism. We have created a software tool *Kerf-Canceler* that locates traditional mechanisms in cutting plans and replaces them with their kerf-canceling counterparts. We evaluated our tool by converting 17 models found online to kerf-invariant models; we evaluated kerf-canceling bearings by testing with kerf values ranging from 0mm and 0.5mm and find that they perform reliably independent of this kerf.

Author Keywords

Personal Fabrication, Portable Fabrication, Sharing, Reuse.

CSS Concepts

- **Human-centered computing~Human computer interaction (HCI)**; *Interactive Systems and Tools*.
- **Applied Computing~Operations Research;** *Computer Aided Manufacturing.*

INTRODUCTION

While laser cutting allows fabricating a wide range of objects [4], arguably the most interesting set of models are those that perform a mechanical function, such as robotic arms (e.g. GrabCAD "braccino"), vehicles (e.g. ugearsmodels.com) and optical equipment (e.g. thingiverse id 31632). Embedding such functionality requires not only joining plates and mounting components [21], but also implementing moving parts, such as axles and bearings, sliders, sprockets and gears—also known as *mechanisms.*

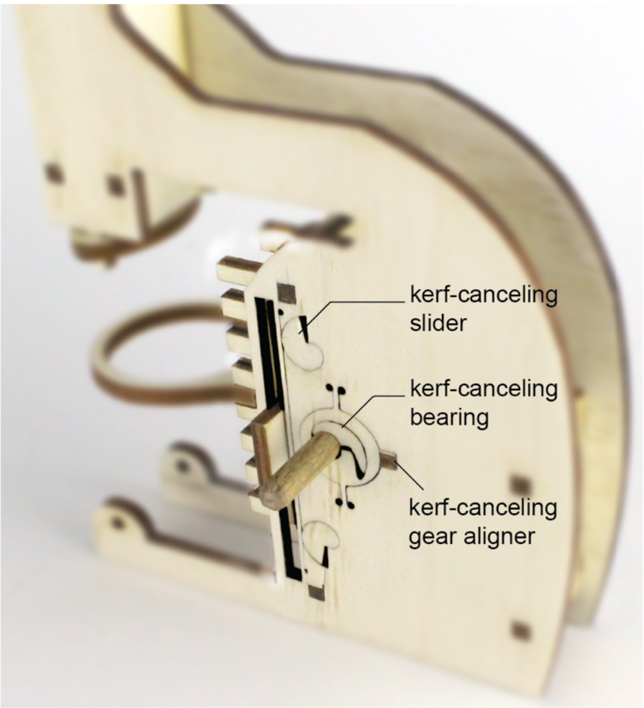

Figure 1: This laser-cut microscope (based on thingiverse id: 31632) contains three types of mechanisms that allow the microscope to adjust focus. By using *kerf-canceling* mechanisms, the focus adjustment operates reliably, independent of how much material the laser cutter that produced the microscope removes (kerf).

The main challenge in creating functional mechanisms is that all parts have to be dimensioned precisely [18], model creators achieve this by means of careful tuning. Unfortunately, this tuning tends to get lost, when the model is fabricated on a *different* laser cutter, as this other cutter is likely to remove more material or less material (aka "kerf") [10]. The resulting poorly tuned models tend to fail mechanically as axles get stuck, sliders wiggle, and gears get jammed.

While the problems that arise from kerf have been investigated in several research projects (e.g., SpringFit [21], as well as 3D editors for laser cutting: CutCAD [13], FlatFitFab [9] and Kyub [5]) these systems do not allow handling mechanisms and require input which is different from laser-cut models shared online.

UIST 2020, October 20–23, 2020, Minneapolis, MN, USA.

ACM ISBN 978-1-4503-7514-6/20/10…$15.00
https://doi.org/10.1145/3379337.3415895

In this paper, we present "kerf-canceling mechanisms". Kerf-canceling mechanisms replace laser-cut bearings, sliders, gear pairs, etc. Unlike their traditional counterparts, however, they keep working when manufactured on a different laser cutter and/or with a different kerf value. Kerf-canceling mechanisms achieve this with by adding an additional wedge element per mechanism (such as the moon-shaped inset in the bearing in the center of Figure 1).

We also present a software tool called *KerfCanceler* that locates certain types of mechanisms in SVG files and replaces them with their kerf-canceling counterparts. The resulting models function irrespective of the laser cutter or kerf values they are fabricated on—making these models particularly suitable for sharing.

CONTRIBUTION, BENEFITS & LIMITATION

In this paper, we make three contributions. First, we present an analysis of why variations in kerf cause laser-cut mechanisms to fail. Second, we address the problem by presenting *kerf-canceling* alternatives to three elementary classes of mechanisms. Third, we present a software tool that locates mechanisms in laser-cut models semi-automatically and replaces them with kerf-canceling equivalents. The tool accepts 2D cutting plans (svg) as input and produces output in the same format, allowing the resulting cutting plans to be stored and shared using existing infrastructure.

Lastly, the proposed workflow requires the designer of the model to act (using kerfCanceler) as opposed to everyone who tries to fabricate it, which is the case with the current workflow where users manually calibrate the file to their laser cutter.

Our approach is subject to three limitations; however, kerf-canceling mechanisms are less robust than traditional mechanisms, they require additional space in the cutting plan, and they may affect the aesthetics of a model.

CHALLENGE: MECHANISMS SUSCEPTIBLE TO KERF

In this section, we take a closer look at why traditional mechanisms fail as the result of variations in kerf. Figure 2 illustrates this at the example of a bearing. In its simplest form, a bearing is a round opening.

In order to work properly, a bearing should hold its axle in place without causing it to jam. In Mechanical Engineering this is referred to as "*loose fit*" [16] and achieving it in a laser-cut model requires the size of the opening to be tuned properly. Without proper tuning, a bearing that is too loose introduces slack. This slack tends to cause mechanical issues. In the focus adjustment mechanism shown in Figure 1, slack causes the microscope's rack and pinion mechanism to jam.

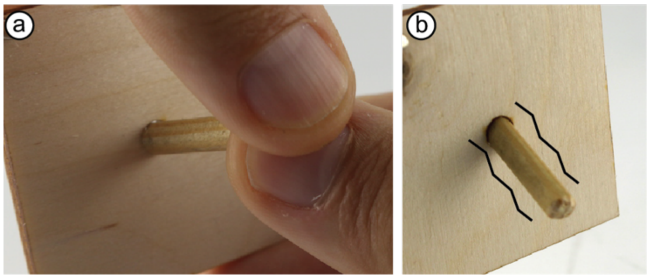

Figure 2: (a) When fabricated on a machine with smaller kerf, this bearing gets too tight. This causes friction or even prevents users from inserting the axle. (b) On machines with wider kerf, bearings are subject to slack, potentially causing adjacent mechanisms to jam.

Unfortunately, tuning tends to get lost when manufacturing a model on a different machine—as is the case when sharing a model, and the resulting models are again subject to slack and/or jamming.

This problem affects a wider range of mechanisms, including three of the four primary types of mechanisms with moving parts [1] shown in Figure 3. Red highlighting indicates areas where kerf-related problems occur.

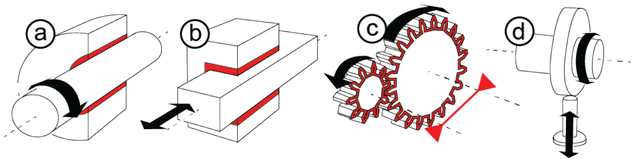

Figure 3: Three out of four elementary classes of joints [1] are subject to kerf-related issues. Susceptible surfaces are marked in red. (a) The *revolute pair* includes mechanisms that operate like the bearing shown before, (b) a *prismatic pair* allows a rod with rectangular cross section to slide forth and back, and (c) a *pair of gears*. (d) Only *cam/follower* mechanisms remain unaffected, as they are spring-loaded.

While we focus on laser cutting, kerf issues consistently affect *all* subtractive fabrication methods, irrespective of the quality of the machine and the approach presented in this paper applies to other types of machines as well, such as milling machines.

KERF-CANCELING BEARINGS

Kerf-canceling mechanisms, such as kerf-canceling bearings address this issue with the help of one additional component: the crescent-shape inset shown in Figure 4a. The figure shows how the mechanism is assembled by inserting the inset and rotating it clockwise. This jams the inset, locking it in place. At this point, the rotation of the inset has reduced the diameter of the remaining opening. The specific design of the inset, as discussed below, causes this opening to always be of the same size, irrespective of the kerf value of the machine it was fabricated on.

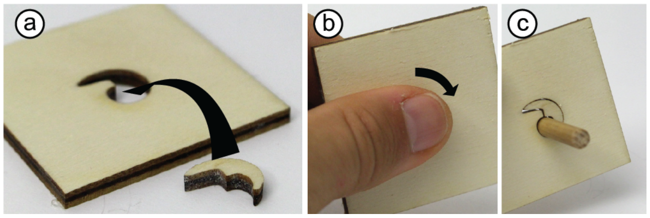

Figure 4: Assembling the kerf-canceling bearing.

As illustrated by Figure 7a, the spiral inset consists of two logical elements, which we call *jammer* and *inverter*.

The jammer: The jammer is the shape on the outside of the inset. To illustrate how it works, consider a *wedge* [12]. As illustrated by Figure 5, a wedge-shaped inset in a wedge-shaped cutout jams when slid towards the tapered side of the cutout. If we increased kerf, the inset slides further—but ultimately it will jam just the same. Note that the distance the inset slides is proportional to the kerf of the machine.

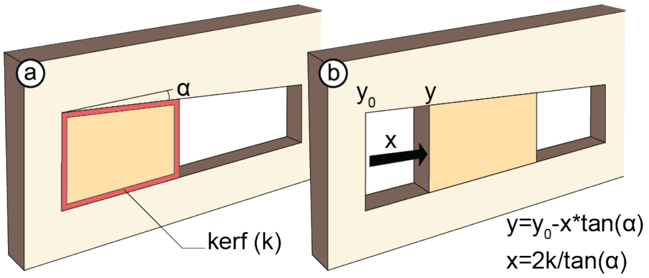

Figure 5: (a) A wedge inset jams by sliding it to the right. A larger kerf value removes the red region, (b) allowing the inset to slide further before it jams.

Applying a polar transformation to the wedge produces the spiral inset we use in kerf-canceling bearings (Figure 6). The spiral version jams when *rotated*. In analogy to the wedge, the inset's final orientation reflects the kerf value.

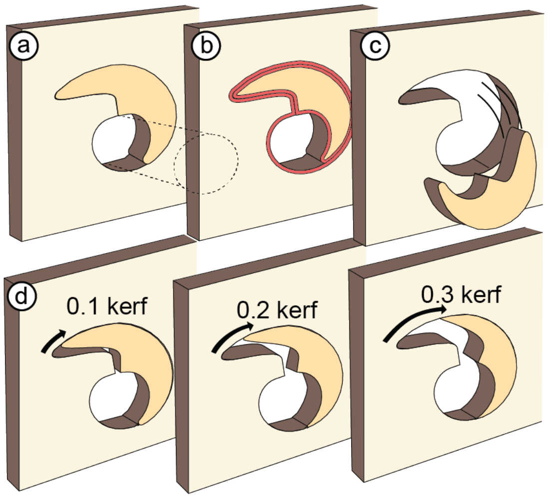

Figure 6: (a) The kerf-canceling bearing. (b) when the model is cut with more kerf, the inset gets smaller while the cutout gets wider. (c) the resulting inset falls out (d), however the self-similar shape of the inset makes that it always jams when rotated in place, even as kerf gets bigger.

The inverter: The inverter is the shape on the inside of the inset. The key idea behind the inverter is that it bears the same shape as the jammer—but mirrored. Based on the jammer translating size (= kerf) into rotation, the inverter translates rotation back into size. Since its shape is mirrored with respect to the jammer, it does so inversely though, i.e., the further it is rotated, the more it reduces the opening in its center, i.e., the bearing. This allows it to keep the size of the bearing constant. With other words, a larger kerf value makes the opening wider, but also leads to additional rotation of the jammer, which in turn causes the inverter to narrow the opening further, canceling out the effect of kerf.

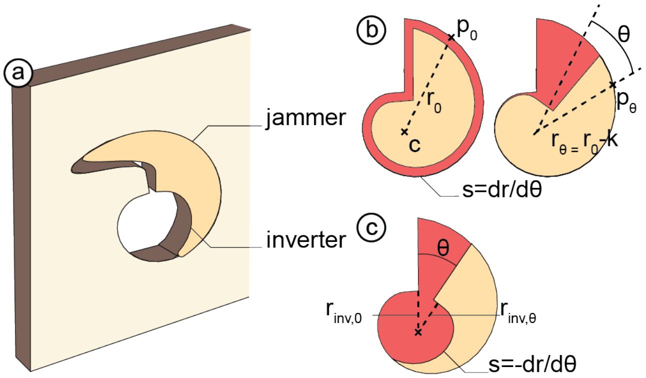

Figure 7: (a) The kerf canceling bearing consists of 2 key elements: (b) the jammer which is characterized by a self-similar nautilus shape that jams in place when rotated and (c) the inverter, which converts the rotation of the jammer back to a bearing, which ultimately holds the axle.

As illustrated by Figure 8, kerf-canceling bearings produce the same fit, irrespective of kerf and thus irrespective of the machine they were fabricated on. Even with a simulated kerf of 0.45mm the bearing continues to produce the desired fit. This exceeds the most extreme typical kerf value in a laser cutting survey by cutlasercut.com [10]. Even when executed on a milling machine with a mill bit of 1.5mm, the axle fits the resulting bearing well.

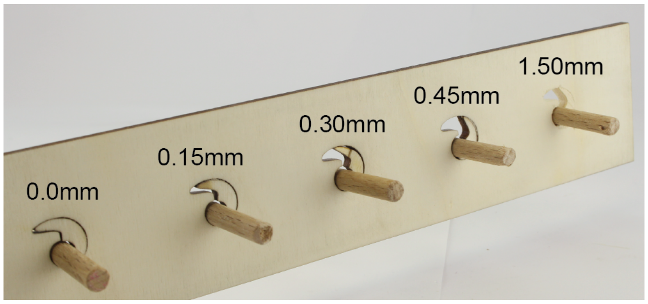

Figure 8: Kerf-canceling bearings fit their axle under variation of a wide range of kerf (by eroding the model). Even when cut on a milling machine with *much* more kerf.

Technical details

To help readers replicate our designs, we now present the necessary technical details. We begin with the jammer. The slope (s) is constant $s = d_r/d_\theta$, the radius thus decreases proportional to the angle θ from the center of the spiral. A given point p_0 on the contour of the jammer has a radius r_0 and corresponding angle θ_0. Another point on the same contour p_θ rotated by an angle of θ from r_0 is thus $r_\theta = r_0 - s*\theta$. We can rewrite this to calculate the angle θ between two points, given their radii: $\theta = (r_0 - r_\theta) / s$.

The cutout and the jammer have the same slope s. Because of kerf, the radius of the inset is *k* shorter (the red zone in Figure 7b). There is a point on the inset with r_0-k which, before jamming the inset, is aligned with p_0. This point jams in the contour where the radius cutout of the contour is r_0-k. We insert r_0-k as r_θ in the formula derived above, and find that the angle θ is $(r_0 - r_0 - k) / s = -k/s$.

The inverter has the same slope as the jammer, flipped (-s). A point on that spiral can be calculated using: $r_{inv,\theta} = r_{inv,0} + s*\theta$ (Figure 7c). If we substitute θ with -k/s, we get: $r_{inv,\theta} = r_{inv,0} + s*(-k/s)$, this simplifies to $r_{inv,\theta} = r_{inv,0} - k$. Kerf eroded the inset by k, so the radius from the center is k longer for every point, this results in: $r_{inv,\theta} = r_{inv,0} - k + k$ or $r_{inv,\theta} = r_{inv,0}$. We conclude that the kerf added, combined with the jamming of the inset results in a bearing of constant size.

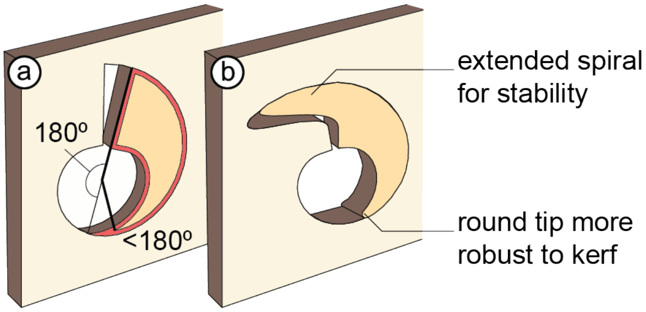

Figure 9: (a) The inset has to span 180 degrees; however, kerf makes it shorter. (b) By extending the spiral and making the tip less sharp, the inset remains stable as kerf increases.

Kerf affects the length of the spiral inset, i.e., if kerf gets wider, the inset gets shorter. To prevent it from spanning less than 180° (Figure 9a), we extend the spiral further than just 180°, by extending it on top (Figure 9b). To make the length of the inset less susceptible to changes in kerf, we round off the bottom tip.

For even better results, we manufacture the inset mirrored. As illustrated by Figure 10, kerf in laser cutting results in a non-straight edge. By mirroring the inset in the cutting plan, it gets cut from the other side, resulting in a part with the slanted edge facing the opposite direction. This allows the slanted edge of the inset to line up with the slanted edge of the rest of the mechanism (Figure 10c). An informal validation shows that flipping the inset increases the friction force by about 60%.

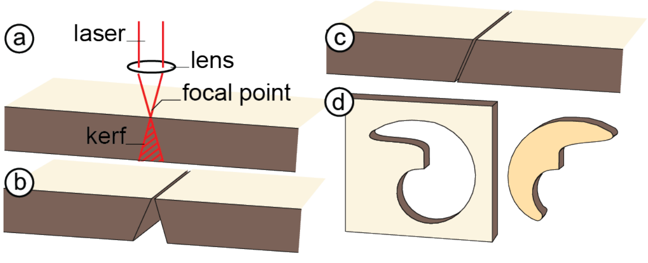

Figure 10: (a) Kerf in a laser cutter is slanted. (b) when cut from the same side, edges poorly align (c) Flipping one side of the plate results in a better fit. (d) Our software tool flips insets by default to support this.

KERF-CANCELING SLIDERS

We have applied this concept of jammer and inverter to three other types of mechanisms. Sliding mechanisms can be orthogonal or parallel to the surface of the model. In both forms, the kerf canceling variant narrows the slit to counteract kerf. Figure 11 shows kerf-canceling sliding mechanisms. We use the principle of the straight wedge (Figure 5). The V shape between the two prongs of the inset lets it slide down to narrow the slit, a spiral wegde on top locks it in place as shown in Figure 11c.

The parallel slider is narrowed down by pushing a thin spring towards the slit. The self-similar nautilus wedges responsible for this are jammed in the surface and push the spring by 1x kerf from both sides.

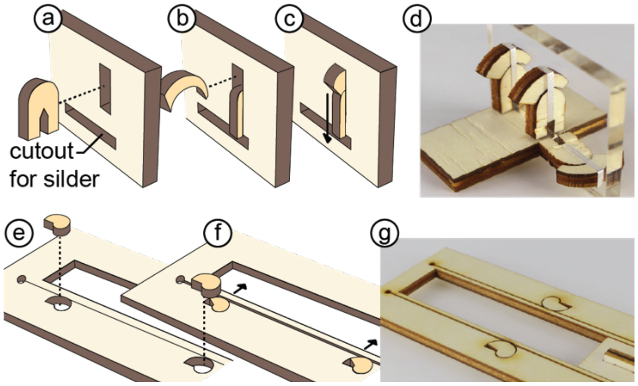

Figure 11: Kerf-canceling sliders (a-d) orthogonal, (e-g) and parallel. (a) The cutout between the prongs lets the shape slide down by 2x kerf. (b) The spiral wedge on top locks it in place. (e) For parallel sliders we insert two simple nautili next to a thin bar (f) the bar gives way as the nautili push by 1x kerf.

KERF-CANCELING GEARS

The kerf-problem with gears (and other mechanisms that interlock into each other) is that kerf makes them smaller, resulting in teeth of one gear to be further away from those of another. To cancel out kerf, we push them towards each other. As shown in Figure 12, we cut a slit around the bearing of one gear and add a wedge next to it to push it towards the other gear. The resulting translation makes the gears mesh again. To keep the bearing in the same plate as its surrounding we do not cut it out all the way but keep it connected to the plate with a thin (flexible) extension.

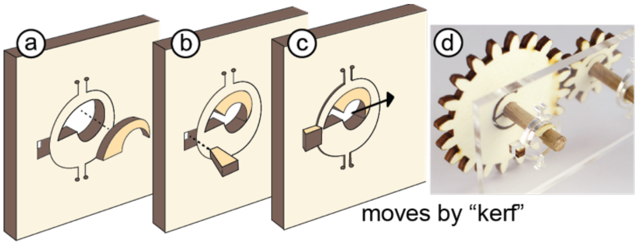

Figure 12: Assembly of the kerf-canceling gear pair. It jams the gears towards each other to compensate for the shorter teeth (a) Insert the bearing wedge, (b) then add a straight wedge next to it, which (c) jams the whole assembly to the right.

Multi-stage gearboxes by combining mechanisms

The kerf-canceling mechanisms described above can be combined to implement more exotic kerf-cancellation techniques. Figure 13 shows a combination of various wedges to form a complex mechanism: a kerf-canceling 3-stage gearbox. Both pairs of gears have to be moved towards each other. A single pair of gears is solved by moving the axles 1x kerf towards one another (Figure 11c). If we naively paired the right and the middle axles, the axle on the left would be 3x kerf away from the middle.

By *nesting* the gear pair on the left together with the middle, they are both moved 1 kerf closer to the gear on the right. Within the nested pair, the left gear is moved 2x kerf closer to the middle gear. The nester corrects kerf equivalent to the angle of the tip: the angle of the left wedge is 2x as narrow as the angle of the middle one making it correct 2x kerf as opposed to the 1x of the nested pair. When compared to the same gearbox with 0.3mm kerf, the normal one jams frequently whereas the kerf-canceled one runs fine.

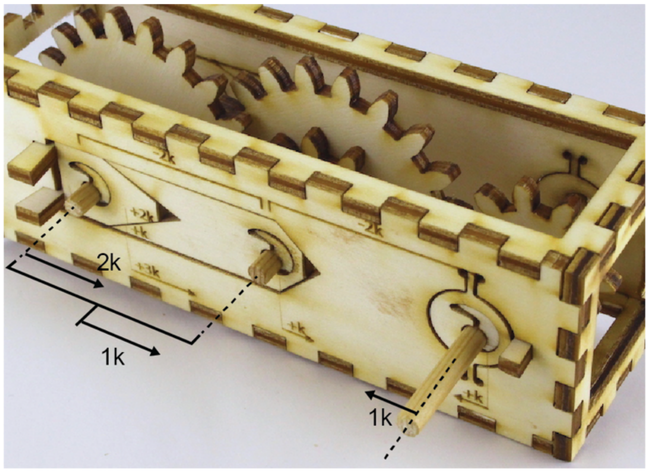

Figure 13: A kerf-canceling multi-stage gearbox.

RELATED WORK

Our work builds on research into reuse in fabrication, tolerances in fabrication, drivers/slicers for laser cutting, compliant mechanisms and strategies users use to handle kerf nowadays.

Reuse in fabrication

SpringFit [21] investigates the machine-dependency problem and presents cantilever-based springs as a replacement of the problematic press-fit joints and mounts. By extending this with our tool, a wide range of portability problems are handled.

PARTS [14] is a framework that makes functional entities in models parametric, which enables models to be portable across a range of use contexts. Similarly, *Grafter* [20] allows users to recombine elements from multiple parent models found in online repositories into a new device. It does so by keeping mechanisms together and joining them with each other. This makes the mechanical design portable across models.

Tolerances in fabrication

The problem of tolerancing in the manufacturing process is widely acknowledged. Zhang et al [22] state the importance of this problem, and highlight the variety of causes of "process variability" as: accuracy of the tools, setup errors, deformation of the machining system under external forces, thermal deformation of the system, tool wear, measurement error and impurity of the materials. These factors all contribute to variations in kerf for laser cutting. Raising the point that not just when switching to a different user/machine the kerf may be off, but even within the same machine, kerf invariant solutions would benefit the user strongly.

Geetha et al. [1] as well as the aforementioned work by Zhang et al [22] tackle the tolerance problem by optimizing the allocation of tolerances within the assembly. Hong et al [15] refer to this approach as tolerance synthesizing, which comes closest to presenting a way to handle tolerance issues as opposed to analyzing them. Most research into tolerancing however, focuses on analyzing the tolerance issues and modeling these to allow engineers or designers to handle them by tuning the models (e.g. work on 3D revolute joints [2] or linkage systems [17]). In particular within the field of mechanical design for haptics these analysis and optimization methods are widely used [1]. Carrino et al [8] took tolerance analysis and synthesis a step further by integrating it into an expert system for mechanical designers.

In mechanical engineering, the Robust Design Methodology [3] aims to reduce noise in the manufacturing process. Their research typically focuses on process design, we extend this notion towards machine invariant fabrication. Downey et al [11] propose what they call *smart features* that are more robust to tolerances using setscrews and adjustable mechanisms. We push this further by having self-adjusting mechanisms that work right away (no configuring or setting required).

“Drivers/slicers” for laser cutting

In 2D publishing, portability has largely been solved by developing drivers, a software tool that takes in a generic model (e.g. PDF) and then configures it into a low-level protocol for the specific printer at hand. In 3D printing, a similar process takes place where STL (or 3MF etc.) files are sliced with a machine-specific configuration into a g-code file that then runs directly on the printer.

In order to achieve a similar workflow for laser cutting, one would have to make a parametric model (e.g., using OpenSCAD (openscad.org) or OnShape (onshape.com). When later exported to 2D for cutting, the points where fit matters can be handled by exporting for a specific laser cutter. 3D editing systems for laser cutting that export 2D geometry are inherently parametric (e.g. Kyub [5] for finger joints, CutCAD [13] for flat joints in general, and FlatFitFab [9] for cross joints). With these systems users do not share the 2D cutting plans, but the 3D models. The conversion to 2D cutting plans then takes place in the local context of the downloading user, allowing that user to generate cutting plans for their local machine’s kerf. In practice, people do not share those high-level descriptions, rather they share SVGs as anyone can open and cut them (without having access to the, typically proprietary, 3D software).

Compliant Mechanisms

Compliant mechanisms have the benefit that they do not consist of separate parts that move with respect to one another, but instead movement happens within the part. Trease et al [19] compared large displacement compliant mechanisms in this context. Their studied mechanisms would require a redesign of the entire model or are limited in the motion range. One exception, which is not limited in motion range, would be the compliant revolute joint by Canon et al [7]. It would be a candidate for bearing mechanisms albeit at a cost of large changes to the model, including modifications to the non-laser-cut part (shaft that rotates in the bearing). We thus consider this outside the scope of mechanisms that could be integrated in off the shelf models, but we would highly recommend including such compliant mechanisms when modeling objects from scratch.

Handling kerf

The default process to handle kerf in practice is to measure the kerf of your machine and tweak the file accordingly. Unfortunately, it is not clear how much kerf and fit the original model already contained and users have to repeat the measurement for different materials or moments of cutting. Our proposed solution puts the effort on the person sharing the file once, to then let other users reliably reproduce the model on their machines. In the context of mechanisms, a valid alternative is to use spring loaded mechanisms as these are not as vulnerable to small variations. Or add little spikes in a bearing which wear/break off when assembling to match the shape of the shaft. Both spring loading and spikes add friction to the mechanisms and risk creating misalignments. While elegant and simple solutions, we think kerf-cancelation is the more reliable approach here.

THE SOFTWARE TOOL: KERFCANCELER

Our software tool, *kerfCanceler*, converts traditional mechanisms in 2D cutting plans to kerf-canceling equivalents. The tool takes the commonly shared SVG format as input and produces output in the same format, allowing users to share the result in existing pipelines/repositories. The software is designed to minimize redundant and uninspiring work for the designer of the model. It automatically guesses the locations and types of mechanisms and then allows users to fix if needed.

Walkthrough: converting the microscope of Figure 1

As shown in Figure 14, the conversion starts by loading a 2D cutting plan into the tool, here the microscope from Figure 1. The menu on the left offers 8 tools, three modify revolute pairs (bearings, mounts and gear pairs), two tools for prismatic pairs (orthogonal and parallel sliders), one utility to define the material thickness, a tool to remove suggestions and a tool that calls the algorithm of springFit [21] to make joints kerf-invariant.

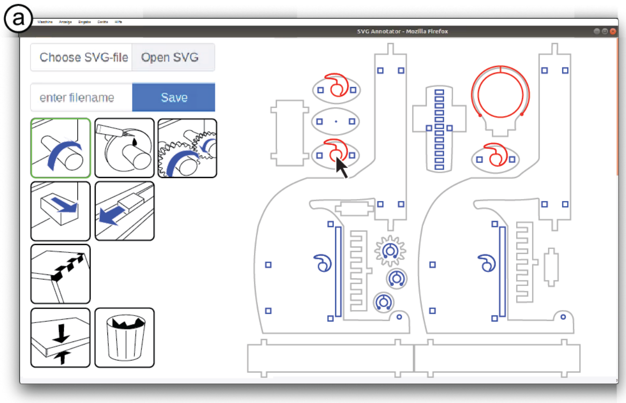

Figure 14: Converting the microscope model of Figure 1.

KerfCanceler classifies polygons when a new cutting plan is loaded (identifying rotary mechanisms with 93% accuracy, see section „Software Evaluation of kerfCanceler”). It automatically inserts kerf-canceling mechanisms. In this example, kerfCanceler added 9 mechanisms automatically.

Kerf-canceling mechanisms require more space than their traditional counterparts, they can intersect with existing geometry in the cutting plan. KerfCanceler detects such cases and highlights them in red.

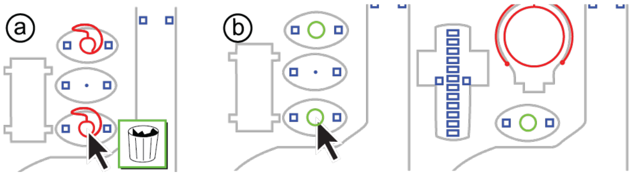

Figure 15: The user removes a kerf-canceling mechanism inserted by kerfCanceler (b) With the “remove mechanism” tool selected; the user clicks on a falsely labeled mechanism. (c) By default, all cutouts with the same diameter now have the mechanism suggestion removed (shown in green briefly to indicate the change).

The microscope has three circles which are glare-holes, but kerfCanceler guessed them to be bearings. The user removes the suggestion as shown in Figure 15b, which reverts them back to the original circular cutout. kerfCanceler recognizes that all three circles are the same size, so the user overrides them in a single click. If the user only wants to modify a single entity, it is possible to turn off "group edit".

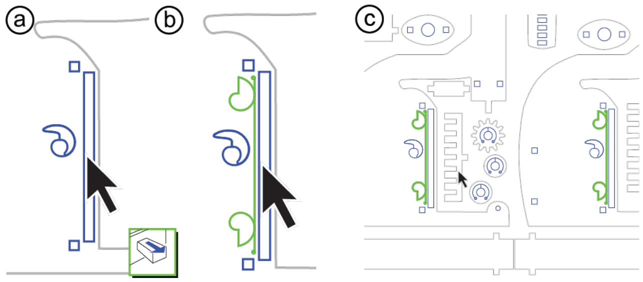

Figure 16: (a) Users add sliding mechanisms manually, using the "slider tool" (b) KerfCanceler creates a kerf-canceling version of that slider (c) both similar cutouts in the model are converted at once.

Sliding mechanisms are rare and hard to identify correctly (any polygon could be a cutout for a sliding mechanism). Based on the principle of *good guesses with little fixing*, KerfCanceler does not automatically place these. As shown in Figure 16, users apply the "slider tool" to manually turn a polygon into a sliding mechanism.

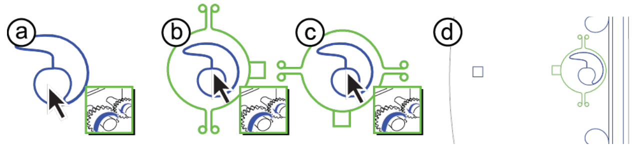

Figure 17: KerfCanceler extends a bearing with the gear tool to compensate for the increased distance between the pinion and the rack as a result of kerf.

The microscope contains a gear (aka pinion), which meshes with the rack. The "gear tool" allows users to align these. It inserts the kerf-canceling mechanism around the already existing bearing (as shown in Figure 17b). Initially, the gear is pushed from the right, by clicking repeatedly, the user rotates this to match the intended orientation. In the first four clicks it rotates by 90-degree steps. After that, granularity goes up.

In a last step, the user calls the springFit [21] algorithm to make joints and mounts kerf invariant. It extends the same data structures as kerfCanceler. We modified the algorithm to *not* place springs when they overlap with a mechanism (and nullify the fit) as the springFit springs tend to occur in abundance. In some models this requires manual fixing.

This process takes a few minutes, and results in an SVG that is fully kerf independent. The model will reproduce on any machine when the user shares it with others.

Once the model is cut, the user jams the insets in place (before assembling the model) and continues to assemble the model in a regular laser cutting workflow.

Classification and Conversion Algorithm

The algorithm to enable the workflow above proceeds in two automatic steps. First, it pre-processes the cutting plan at hand to identify mechanisms. Second, it replaces these mechanisms with kerf-canceling equivalents.

Pre-processing

KerfCanceler normalizes the SVG by breaking all SVG elements into line segments. This removes ambiguities (e.g. polylines and paths that do the same thing but are defined differently) or document properties like layers that don't influence the laser cutting.

KerfCanceler runs a parts vs cutout detection. It sorts all closed polygons by size. It checks if there is a larger polygon within which the given (smaller) polygon is enclosed and continues to do so until all are checked. It assumes that the outer cuts are outlines of parts and the inner ones are scrap.

As shown in Figure 18, the user's attention is pointed towards the content kerfCanceler assumes to be relevant. The outlines of the parts are greyed out and the cutouts are highlighted (typically the outlines of parts are not mechanisms).

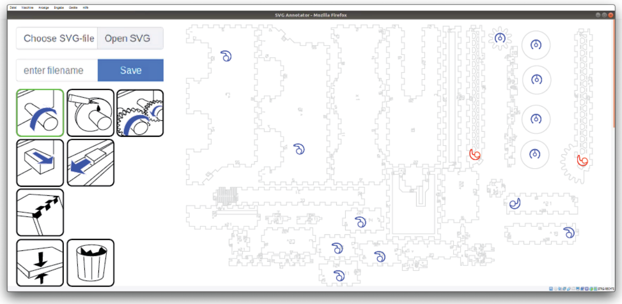

Figure 18: A model presented to the user (a firetruck). All outline geometry is greyed out to put the users' emphasis on the mechanisms guessed by KerfCanceler.

KerfCanceler iterates over the inner geometry to find mechanisms. Revolute pairs (e.g. bearings, gears, wheels, cam/followers) manifest themselves as circles in the model. KerfCanceler groups circles by diameter. As shown in Figure 19, when two similar groups occur, it assumes the group with smaller diameter is press-fit and the other group is loose fit.

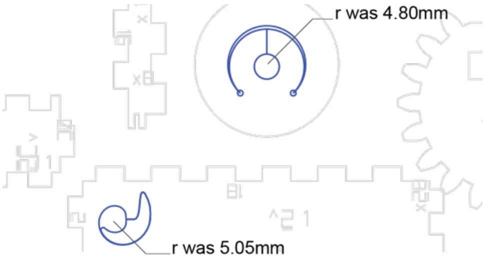

Figure 19: These circle cutouts in the firetruck are of similar size. In the entire fire-truck model, one category turned out to be around 5.05 and one around 4.80mm, KerfCanceler assumes the small opening is press-fit opening and the other one loose fit (it thus placed two different mechanisms).

Replacing mechanisms with kerf-canceling counterparts

KerfCanceler then places kerf-canceling mechanisms. At the positions where it assumed loose or press fit mechanisms, it inserts the correct version. For every circular cutout, it caches three alternatives shown in Figure 20a-c: the original circle, a press-fit mount based on cantilever springs [21] and a kerf-canceling loose-fit bearing. It displays the one it guesses to be the right version. Because these alternatives are generated *before* the user touches them, it allows for interactive response times in the web UI.

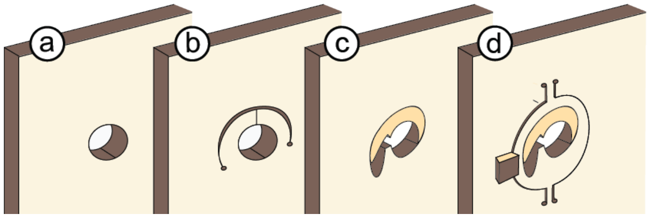

Figure 20: Possible modifications of a circle cutout. (a) The original circle (b) a circle used as a mount (press-fit) (c) the circle used as a kerf-canceling bearing and (d) the same as c but pushed to the right by "kerf" using the wedge on the left, for gears.

KerfCanceler checks for intersections with the model during pre-processing. It uses the shape of Figure 20c overlaid by b. If this intersects with the rest of the SVG model, the mechanism shows up in red, otherwise in blue. It does not use the larger kerf-canceling gear-bearing of Figure 20d as this is a rare case and would produce many false positives. When the user later on inserts a gear-pair mechanism, kerfCanceler checks the intersections locally resulting in slightly longer processing time (up to a second).

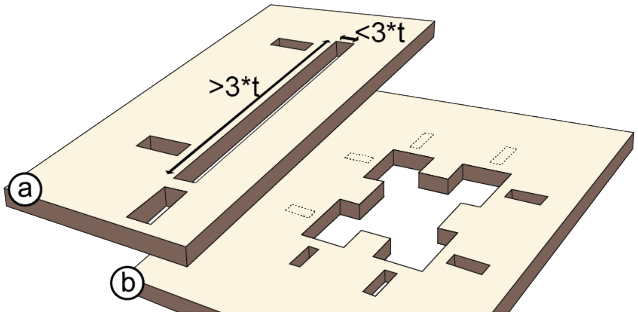

Figure 21: The placement of wedges for a sliding mechanism, (a) half of the edges of the cutout get a kerf adjusting wedge. (b) The same works for non-rectangular cutouts. There is more than 1 possible placement for each wedge (see dotted lines). KerfCanceler, excludes all that cause intersections and picks the best solution.

For guided sliding mechanisms, the wedges do not replace the original polygon, but line up on the sides. As shown in Figure 21, kerfCanceler places two wedges on each edge. It places the wedges as far apart from each other as possible to minimize the risk of jamming the slider. For short edges, it places one wedge in the middle of the edge.

TECHNICAL EVALUATION: HOW WELL DO KERF-CANCELING MECHANISMS PERFORM?

We hypothesize that kerf-canceling mechanisms are comparable in performance to the original mechanism under default kerf, and that with increased kerf, the kerf-canceling mechanisms outperform the original. We evaluate this by measuring the *friction* and the *play* of the mechanism and compare that to plain bearings, while varying kerf.

We measure friction by spinning an axle with 2 flywheels on the side held in the sampled bearing, we start at 1300rpm (=136.14 rad/s) and measure how long it takes until the shaft stops spinning as a result of angular friction.

We measure the tilt angle of the axle within each of the sampled bearings. We take a photograph with a fixed camera from the side of the bearing, tilt the axle up and down and capture both extremes. The angle between these corresponds to the maximum range of play.

Test setup

We mount the bearing with an 8mm aluminum axle. We attach a 3D printed flywheel with 4x 33g steel balls inside, to both ends of the axle. The shaft is powered using a Bosch drilling machine via a simple clutch. The total inertial moment of the flywheels is 17.4×10^{-5} kgm^2. We use the *Peaktech 2795* contactless rotation sensor to measure the rotation speed. We then calculate the frictional Torque (T) using this basic formula:

$$T = I * a$$

In which a is the angular acceleration (initial rotation (rad/s)/ time (s)) and I the moment of inertia (kgm^2).

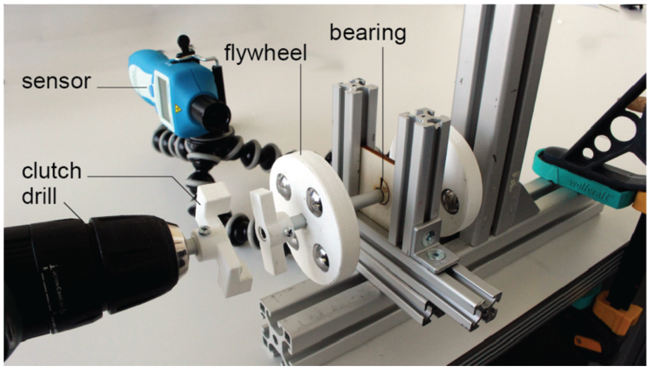

Figure 22: Experimental set-up.

Test pieces

We compare the baseline (a plain bearing) to the kerf-canceling bearing. All pieces were cut out of 4mm plywood and we simulated kerf from 0 to 0.4mm in 0.1mm increments. These kerf values we adjusted for the laser cutter used, so 0mm kerf means the bearing fully touches the axle. We repeated each experiment 3 times and report the average value to compensate for noise.

We used a *Trotec speedy 360 flexx* laser cutter with a kerf of 0.15mm. To reproduce this experiment, we have attached a test piece in the appendix of this paper.

Results

As shown in Figure 23, Kerf-canceling bearings demonstrated constant performance across variations in kerf (between 3.1 and 3.4 mN). Kerf heavily affected the plain bearing's performance. Already at a kerf variation of 0.1mm the friction went up substantially (4.7 mN). And in particular when reducing the kerf further, the bearing essentially got stuck as friction went up by a factor of more than 10. (40.6 mN).

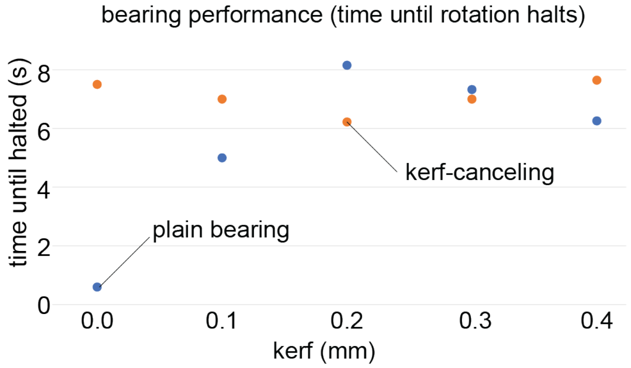

Figure 23: Results of the friction test. Kerf-canceling based bearings perform stable across kerf variations as opposed to plain bearings.

Below are the results of the play analysis. For the plain bearing, the play increases roughly linearly with the kerf. The play for the kerf-canceling bearings remained stable.

We found that strong vibrations (e.g. by accidentally misaligning the drill bit) can cause the inset to come out. For mechanisms that are expected to be exposed to such forces, we recommend adding a dot of glue before assembling the mechanism.

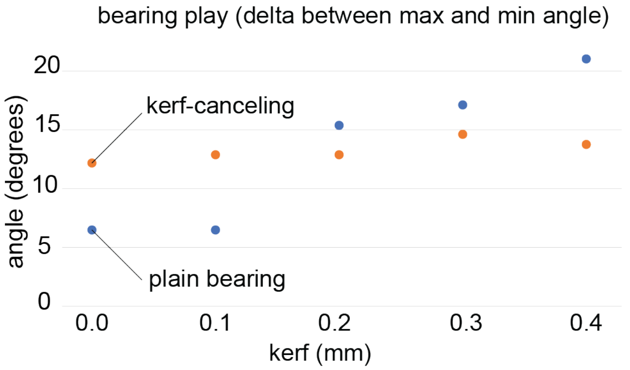

Figure 24: Results of measuring play of the bearings. The kerf-canceling bearing remains relatively stable, while play for the plain bearing almost linearly relates to increasing kerf.

Discussion

Kerf-canceling bearings demonstrate constant performance independent of the kerf, both when it comes to the play and the friction of the mechanisms. While plain bearings only perform reliably in a narrow range of kerf. We thus conclude that our bearings serve well as kerf-canceling mechanisms.

SOFTWARE EVALUATION OF KERFCANCELER

To validate the utility of our software, we ran it on 20 models found online. For each model, we measured what percentage of mechanisms were identified automatically and how many interaction steps were required to modify the mechanisms. We also measured the time it took to do this.

The models in Figure 25 are a subset of the 20 test models which we fabricated to confirm that the generated kerf-canceling mechanisms work.

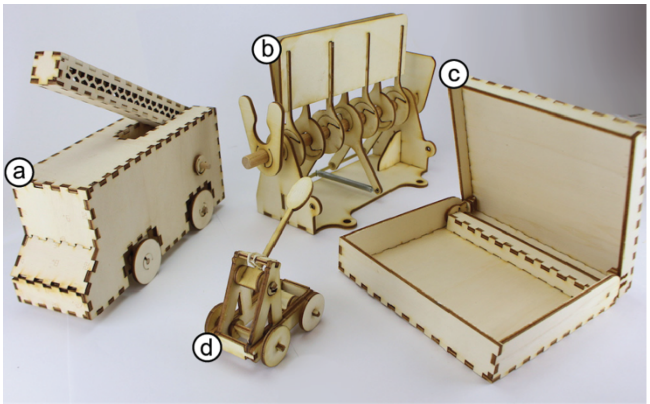

Figure 25: Models of the test set we fabricated on our laser cutter with increased kerf.

KerfCanceler achieved a 93% recognition rate for the rotary mechanisms in the models. It identified false positives in 5 models, which contained engraved (decorative) circles, these were falsely identified as mechanisms.

We used the UI to intervene with 2-21 (9 on average) overrides of the initial guessed mechanisms. Six models worked based on the guessed mechanisms alone. The "group edit" tool reduced the number of edits in most models. Pre-processing of models took on average 5.87ms of time. It took 66s of manual work per model to convert, for a user who knows the model's functionality.

Six models failed to convert. Three of them had too little physical space in the model to insert the kerf-canceling mechanisms. Four models contained lines that were intended to be engraved, which caused intersections. One model showed both problems. These intersections won't break the mechanism but may affect the aesthetics of the model depending on how meaningful the original engravings were. So, in total 17/20 models were converted using our tool with a laser cuttable result.

We conclude that many models online can be converted to become kerf-canceling with only up to three minutes (one minute on average) of user effort.

CONCLUSION AND PORTABLE LASER CUTTING

In this paper, we demonstrated how to create kerf-invariant laser cut models with the help of *kerf-canceling* mechanisms. We also presented a software tool that replaces problematic mechanisms in 2D cutting plans with kerf-canceling equivalents. The resulting cutting plans remain valid across machines and kerf values, which, for example, allows users to buy a new laser cutter without invalidating cutting plans created earlier.

Zooming out, kerf-canceling mechanisms address one facet of a larger challenge, i.e., the challenge of *portability*. Today, the majority of laser-cut models are shared as 2D cutting plans—and these are inherently machine-specific. This is problematic, as this gets in the way of collaboration and sharing, which rely on people's ability to reproduce other users' models, e.g., for the purpose of remixing them.

In the long run, we as a field should try to move away from 2D cutting plans and towards more abstract representations that are merely instantiated for the machine at hand. 3D editors for laser cutting are an important first step in this direction (such as FlatFab [9] and Kyub [5]). In the future, it would be good to see such systems not just represent a model's shape, but also the logic behind it. In 3D printing this has already started to happen—with modern file formats that contain mechanical metadata such as 3mf. For laser cutting, systems that treat laser-cut objects as 3D models [9], [5], [13] are a great first step in this direction.

Still, as of today, 2D cutting plans are the most common format for exchanging models and will most likely remain relevant legacy for decades to come. We thus have to deal with them. We think of kerf-canceling mechanisms as one facet in the bigger agenda of transitioning from 2D cutting plans to machine-independent formats.

To illustrate this idea of *kerf-invariant* (and maybe one day *portable*) 2D cutting plans, we combined our tool *KerfCanceler* with a tool that makes joints and mounts kerf-invariant (*springFit* [21]). As shown in Figure 26, we pipe SVG files through both tools, resulting in mechanisms being replaced and then mounts and joints being replaced. The resulting model now assembles and works reliably irrespective of the laser cutter it is fabricated on.

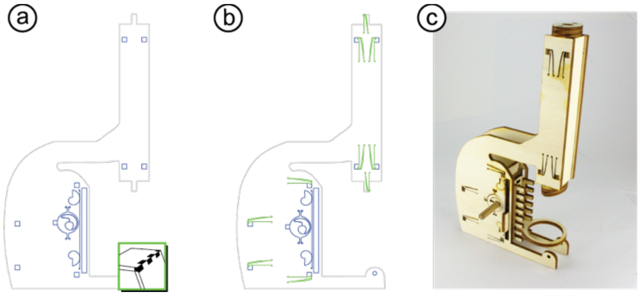

Figure 26: After making all mechanisms kerf-canceling, the "joint tool" automatically converts all joints to portable ones using the algorithm of springFit [21].

REFERENCES

[1] Aftab Ahmad, Kjell Andersson, and Ulf Sellgren. "An optimization approach toward a robust design of six degrees of freedom haptic devices." *Journal of Mechanical Design* 137, no. 4 (2015): 042301.

[2] Narendra Akhadkar, Vincent Acary, and Bernard Brogliato. "Multibody systems with 3D revolute joints with clearances: an industrial case study with an experimental validation." *Multibody System Dynamics* 42, no. 3 (2018): 249-282.

[3] Martin Arvidsson, and Ida Gremyr. "Principles of robust design methodology." *Quality and Reliability Engineering International* 24, no. 1 (2008): 23-35.

[4] Patrick Baudisch and Stefanie Mueller. 2017. Personal Fabrication, Foundations and Trends in Human–Computer Interaction: Vol. 10: No. 3–4, pp 165-293. http://dx.doi.org/10.1561/1100000055

[5] 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

[6] Leah Buechley and Hannah Perner-Wilson. 2012. Crafting technology: Reimagining the processes, materials, and cultures of electronics. ACM Trans. Comput.-Hum. Interact. 19, 3, Article 21 (October 2012), 21 pages. DOI: https://doi.org/10.1145/2362364.2362369

[7] Jesse R. Cannon, and Larry L. Howell. "A compliant contact-aided revolute joint." *Mechanism and Machine Theory* 40, no. 11 (2005): 1273-1293.

[8] Giovanni Moroni Carrino, Wilma Polini, and Quirico Semeraro. "Machining planning for tolerance synthesis." *Machining science and technology* 7, no. 3 (2003): 333-347.

[9] 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

[10] CutLaserCut kerf manual http://www.cutlasercut.com/resources/tips-and-advice/what-is-laser-kerf. Accessed January 2020.

[11] Kris Downey, Alan Parkinson, and Ken Chase. 2003. An introduction to smart assemblies for robust design. Research in Engineering Design 14, no. 4 (2003): 236-246. DOI: https://doi.org/10.1007/s00163-003-0041-5

[12] Heinrich Dubbel, and B. J. Davies. *Dubbel-Handbook of mechanical engineering*. Springer Science & Business Media, 2013.

[13] 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

[14] Megan Hofmann, Gabriella Hann, Scott E. Hudson, and Jennifer Mankoff. 2018. Greater than the Sum of its PARTs: Expressing and Reusing Design Intent in 3D Models. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18). ACM, New York, NY, USA, Paper 301, 12 pages. DOI: https://doi.org/10.1145/3173574.3173875s

[15] Hong, Y. S., and T. C. Chang. "A comprehensive review of tolerancing research." *International Journal of Production Research* 40, no. 11 (2002): 2425-2459.

[16] ISO, DIN. "286-1: ISO-System für Grenzmaße und Passungen." *Grundlagen für Toleranzen, Abmaße und Passungen* (1990).

[17] Pezzuti, E., R. Stefanelli, P. P. Valentini, and L. Vita. "Computer-aided simulation and testing of spatial linkages with joint mechanical errors." *International journal for numerical methods in engineering* 65, no. 11 (2006): 1735-1748.

[18] Ting, Kwun-Lon, and Yufeng Long. "Performance quality and tolerance sensitivity of mechanisms." *Journal of mechanical design* 118 no. 1(1996): 144-150.

[19] Brian Trease, Yong-Mo Moon, and Sridhar Kota. "Design of large-displacement compliant joints." *Journal of mechanical design* 127, no. 4 (2005): 788-798.

[20] Thijs Roumen, Willi Müller, and Patrick Baudisch. 2018. Grafter: Remixing 3D-Printed Machines. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18). ACM, New York, NY, USA, Paper 63, 12 pages. DOI: https://doi.org/10.1145/3173574.3173637

[21] Thijs Roumen, Jotaro Shigeyama, Julius Cosmo Romeo Rudolph, Felix Grzelka, and Patrick Baudisch. SpringFit Joints and Mounts that Fabricate on Any Laser Cutter. In *Proceedings of the 32th annual ACM symposium on User interface software and technology* (UIST '19). ACM, New York, NY, USA, DOI: https://doi.org/10.1145/3332165.3347930

[22] Luke Wallace, A fully lasercut kit 3D printer, on https://hackaday.io/project/164156-lp3d-a-fully-lasercut-kit-3d-printer last accessed September 2019

[23] Chun Zhang, and Wang Hsu-Pin. "Robust design of assembly and machining tolerance allocations." *IIE transactions* 30, no. 1 (1998): 17-29.

[24] Amit Zoran. 2015. Hybrid craft: showcase of physical and digital integration of design and craft skills. In ACM SIGGRAPH Art Gallery (SIGGRAPH '15). ACM, New York, NY, USA, 384-398. DOI: http://dx.doi.org/10.1145/2810185.2810187

[25] Amit Zoran and Leah Buechley. "Hybrid reassemblage: an exploration of craft, digital fabrication and artifact uniqueness." Leonardo 46, no. 1 (2013): 4-10. DOI: https://doi.org/10.1162/LEON_a_00477