

# UC Berkeley

## UC Berkeley Electronic Theses and Dissertations

### Title

Computational Woodworking: Tools for Designing Stable Decorative Joints and Flexible Kerf Patterns

### Permalink

<https://escholarship.org/uc/item/1tk795xj>

### Author

Yao, JiaXian

### Publication Date

2021

Peer reviewed|Thesis/dissertation

Computational Woodworking: Tools for Designing  
Stable Decorative Joints and Flexible Kerf Patterns

by

JiaXian Yao

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Maneesh Agrawala, Co-chair

Professor Björn Hartmann, Co-chair

Professor Eric Paulos

Professor Sara McMains

Fall 2021

Computational Woodworking: Tools for Designing  
Stable Decorative Joints and Flexible Kerf Patterns

Copyright 2021  
by  
JiaXian Yao

## Abstract

Computational Woodworking: Tools for Designing  
Stable Decorative Joints and Flexible Kerf Patterns

by

JiaXian Yao

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Maneesh Agrawala, Co-chair

Professor Björn Hartmann, Co-chair

Traditional wood furniture often features decorative intrinsic joints and beautifully curved parts created with woodworking techniques. To craft such furniture, experienced woodworkers usually begin by drawing a *visual design* that describes not only the overall shape of the furniture but also the detailed appearances of the joints and curved parts. They then manually *verify* whether their furniture design satisfies all the constraints imposed by the intended design scenario (e.g., furniture assembled via joints must be stable when put together) and redesign if needed. Finally, they *fabricate* the furniture with conventional woodworking tools.

This traditional furniture crafting process demands experience in woodworking and has historically been inaccessible to amateur users. Encouragingly, the continuing advancement and increasing availability of CAD programs and CNC machines have lowered the entry to the visual design and physical fabrication steps of the furniture crafting process. But, neither CAD programs nor CNC machines help users verify whether their visual design of the furniture fulfills all the necessary constraints, even though such verification often requires advanced woodworking knowledge not commonly held by everyday users. As a result, the verification step of the crafting process presents a serious challenge that prevents average users from expressing their creativity in furniture design.

This thesis explores ways to assist users with verifying their furniture designs and guide users towards visual designs that satisfy all the requisite constraints. We aim to develop algorithms that automate the difficult verification step, allowing users to focus on producing the visual design of their furniture rather than analyzing the validity of their design. This thesis introduces interactive design tools that integrate algorithms informed by woodworking techniques to enable users to create (1) stable furniture models assembled via decorative intrinsic joints and (2) curved furniture parts formed through flexible kerf patterns.

To my mother, my father, and my love.

# Contents

<b>Contents</b>	<b>ii</b>
<b>List of Figures</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Woodworking Techniques . . . . .	2
1.2 Interactive Design Tools . . . . .	4
1.3 Statement of Multiple Authorship and Prior Publication . . . . .	5
<b>2 Related Work</b>	<b>6</b>
2.1 Designing Furniture Assemblies . . . . .	6
2.1.1 Joinery Design . . . . .	7
2.2 Designing Physical Properties . . . . .	7
2.2.1 Decorative Pattern Design . . . . .	8
2.2.2 Metamaterial Design . . . . .	8
<b>3 Designing Stable Decorative Joints</b>	<b>9</b>
3.1 Introduction . . . . .	9
3.2 Overview . . . . .	10
3.3 Construct Solid 3D Parts . . . . .	11
3.3.1 Find SOPA Compatible Disassembly Sequences . . . . .	12
3.3.2 Two-Pass Construction Algorithm . . . . .	13
3.3.2.1 Pass 1: Construct Maximum 3D Parts . . . . .	13
3.3.2.2 Pass 2: Re-Allocate Volume of Maximum 3D Parts . . . . .	14
3.3.3 Iterative Redesign for Infeasible Inputs . . . . .	18
3.4 Stability Analysis . . . . .	20
3.4.1 Static Analysis for Rigid Assemblies . . . . .	21
3.4.2 The Equilibrium Method . . . . .	22
3.4.3 Variational Static Analysis . . . . .	24
3.4.4 Evaluation . . . . .	26
3.4.5 Visualizing Instability . . . . .	28
3.5 Results . . . . .	29

3.6	Conclusion . . . . .	34
<b>4</b>	<b>Designing Flexible Kerf Patterns</b>	<b>35</b>
4.1	Introduction . . . . .	35
4.2	Overview . . . . .	37
4.3	Simulations of Periodic Kerf Patterns . . . . .	37
4.3.1	Simulations of Unit Kerf Patterns . . . . .	37
4.3.2	Periodic Boundary Conditions . . . . .	38
4.4	Design Principles for Flexible Kerf Patterns . . . . .	40
4.4.1	Geodesics and Curvatures . . . . .	41
4.4.2	Observations and Analyses . . . . .	42
4.4.3	Geodesic Measures and Design Principles . . . . .	43
4.5	Results . . . . .	48
4.5.1	Interactive Design Tool . . . . .	49
4.5.2	Kerf Pattern Design Walkthroughs . . . . .	50
4.6	Conclusion . . . . .	52
<b>5</b>	<b>Conclusion</b>	<b>53</b>
5.1	Summary of Contributions . . . . .	53
5.2	Limitations and Future Work . . . . .	54
	<b>Bibliography</b>	<b>56</b>
<b>A</b>	<b>Appendix for Chapter 3</b>	<b>66</b>
A.1	Gallery of 100 Joint Designs . . . . .	66
A.2	Gallery of Fabricated Joints . . . . .	75
<b>B</b>	<b>Appendix for Chapter 4</b>	<b>76</b>
B.1	Gallery of 75 Unit Kerf Patterns . . . . .	76

# List of Figures

- 1.1 There are two general categories of intrinsic woodworking joints. Box joints (e.g., dovetail joints) connect parts perpendicularly to construct drawers or cabinets (a), while frame joints (e.g., pinned double bridle joints) attach parts planarly to build window frames or backs of chairs (b). While many joints have simple yet decorative surface appearance (c), their internal geometries are often surprisingly complex and difficult to design directly (d). (Please zoom in to see the geometric details of the joints.) . . . . . 2
- 1.2 Woodworkers traditionally employ three different methods to bend wood. The steaming method softens timber pieces with heat and moisture in a steam box before bending them (a), while the lamination method bends thinly-sliced lumber strips before gluing them together (b). In comparison, the kerfing method is less labor-intensive and allows wood boards to bend flexibly by cutting deep notches into them (c). Inspired by the kerfing method, designers have created cutout patterns or kerf patterns that similarly enable rigid boards to become flexible after materials are removed from them based on the patterns (d). . . . . 3
- 3.1 Hand-crafted furniture often includes decorative joinery that is valued for its strength and beauty (a). With our interactive joinery design tool, users draw the visible 2D surface regions corresponding to parts on the surface of a solid 3D furniture model (b). Our tool then constructs solid 3D parts that conform to the user-specified regions and assemble into the solid model (c, d). (Please zoom in to see joint details in furniture models.) . . . . . 10



- 3.2 Three stage workflow of our interactive joinery design tool. In stage 1, the user creates a solid 3D model and draws a partition of the model’s exterior surface into groups of regions we call **surface 2D parts** (each color—green, orange, blue, pink, and purple—indicates a different surface 2D part). In stage 2, our tool either converts each such surface 2D part into a corresponding **solid 3D part** that together assemble into the input solid model, or reports that such conversion is infeasible. If the infeasibility is due to collisions between the surface 2D parts, our tool visualizes the collision regions and allows the user to quickly redesign the parts so that they are collision free. In stage 3, our tool uses physical simulation to check the stability of the solid 3D parts (the red arrows indicate that three legs of the Leaf table are unstable and will slide with respect to the table-top). If the parts are unstable, our tool suggest where the user might add 2D regions to a surface part in order to improve stability (highlighted yellow regions). The user can then go back to stage 1 and redesign the surface 2D parts based on the suggestions. Figure 3.1 shows the redesigned Leaf table with additional joints connecting the legs to the table-top. . . . . 11
- 3.3 Given a solid 3D model with a set of surface 2D parts (a), we compute a directional blocking graph (DBG) in a discrete set of directions (8 in this 2D example, 26 in our 3D implementation) (b). For each DBG direction we compute a blocking graph. For example in the upward direction graph at the top of the DBG, the arrow from orange to blue means that the orange part is blocked by the blue part in the upward direction. We use the DBG to compute SOPA compatible disassembly sequences one of which is shown in (c). . . . . 12
- 3.4 Given surface 2D parts and a SOPA sequence, we initialize  $Max(p_3)$  to include the complete input solid model (a). We then compute the sweep volumes of earlier surface 2D parts in their respective disassembly directions (b) and subtract them from  $Max(p_3)$  (c). We also compute the sweep volumes of subsequent surface 2D parts in the negative disassembly direction for  $p_3$  (d) and subtract them from  $Max(p_3)$  (e). Finally, we remove any extraneous volumes from  $Max(p_3)$  (f). We similarly generate all of the other maximum 3D parts (g). . . . . 13
- 3.5 Surface 2D parts  $p_1$  and  $p_2$  (a) and their corresponding maximum 3D parts (b). The maximum 3D parts are used as the initial solid 3D parts  $P_1$  and  $P_2$ . To re-allocate the intersection volume  $I_{12}$  between  $P_1$  and  $P_2$  (c), we evaluate whether it can be given to  $p_2$ . Such a gift would block a portion of  $P_1$  (labeled  $B_{12}$ ) from being disassembled upwards with the rest of  $P_1$ . Because the solid 3D part  $P_1$  still contains its surface 2D part  $p_1$  even without the intersection and blocked volumes, the gift occurs (d). . . . . 16

- 3.6 The intersection  $I_{12}$  between solid 3D parts  $P_1$  and  $P_2$  touches a portion of surface 2D part  $p_1$  (a). As a result, the intersection volume cannot be given to  $P_2$  without first removing some volume near the intersected portion of  $p_1$ . The initial thickening volume  $T_f$  for the intersected portion  $f$  is obtained by extruding  $f$  parallel to  $p_1$ 's disassembly direction until it touches a non-neighboring face of  $p_2$  (b). This initial thickening volume is iteratively reduced by half (c) until it no longer intersects  $p_{2,nnf}$ . The intersection volume is reduced by subtracting  $T_f$  (d) and can now be given to  $P_2$ , resulting in new solid 3D parts (e, f). . . . . 18
- 3.7 Disassembly sequence minimizing collisions between surface 2D parts (a bottom). Each thumbnail highlights the next part to be removed from the assembly in the disassembly direction indicated by the arrow (e.g., the green part is removed in step 1). A red border indicates that the disassembly step contains collisions. Users can inspect the collision regions within each such step; here the user examines collision regions between the green and purple parts in step 1 (a top). To resolve the collision, the user can choose to give the highlighted collision regions to either of the two regions involved (b or c). . . . . 19
- 3.8 Results of stability and tilt tests comparing analytic and experimental results with analyses from our variational static solver and the equilibrium method (EM) implementation analyzed by Shin et al. [83]. To match with prior results [83], we follow the reported material density,  $1.5 \text{ g/cm}^3$ , and friction angle,  $43^\circ$ . "S" and "U" indicate stable and unstable. For stability determination on a horizontal ground plane, our variational solver matches with the analytic and experimental solutions for all assemblies. EM differs by missing the sliding instability for the H assembly. For the tilt tests, our variational solver predicts a critical tilt angle between the angles experimentally determined and those predicted by EM. Note that mesh colors indicate block number and do not depict stress. . . . . 26

- 3.9 Results of stability analyses from our variational solver and the equilibrium method (EM) on five inadequately jointed furniture models under gravity. “S” and “U” indicate stable and unstable assembly results per method, respectively. All examples should be determined unstable in the analysis. Note that EM incorrectly determines the two leftmost assemblies stable. In addition to analyzing whether the furniture assembly is stable or not, for furniture design we require an analysis of where and how parts are sliding or hinging against each other if the system is found part-unstable. This is information that EM cannot provide. Our variational solver identifies the location and direction of sliding and hinging part-instabilities between furniture parts (bottom row). This allows our tool to suggest additional joint regions to improve part stabilities. For the bamboo table, the six small braces near the ground are modeled with density  $8.0 \text{ g/cm}^3$  and all other parts are modeled with density  $0.3 \text{ g/cm}^3$ , while the four remaining models use a density of  $0.5 \text{ g/cm}^3$ . The corresponding friction coefficients  $\mu$  used for each model are summarized above. Red and blue arrows, if present, give the local linear and angular acceleration of unstable parts (i.e., the locations and directions of the sliding and hinging instabilities, respectively). . . . . 27
- 3.10 An initial set of surface 2D parts (a) generates unstable solid 3D parts (b). Our tool reports the part instabilities and presents the relative linear accelerations between unstable parts as red arrows. Our tool highlights suggested regions in yellow on the model surface where additional 2D regions can be drawn to improve part stability (c). Re-design with additional 2D regions leads to a stable 3D part assembly (d). . . . . 29
- 3.11 A representative sample of joints designed and fabricated using our interactive tool. Some are based on traditional woodworking joints (b, e, g) and others on Japanese joinery (a, c, d, f). For all of these designs we drew the surface 2D parts using a photo of the assembled hand-crafted joint (top row) as a guide. The resulting solid 3D parts generated by our tool (3rd row) is very similar to the hand-crafted part geometry (4th row) suggesting that the internal geometry of such joints is inherently complex. Please zoom in to see the geometric details. Appendix A contains 100 decorative joints we created using our tool. . . . . 30
- 3.12 Complete furniture assemblies we designed using our interactive tool. . . . . 31
- 3.13 Our tool can generate decorative joints for free-form surfaces like the Stanford bunny. . . . . 32
- 3.14 We commissioned a woodworker to fabricate a version of the Branca chair at full size using the solid 3D parts generated by our tool. (Coffee cup was added for scale.) . . . . . 33

- 4.1 Kerf patterns allow rigid boards to bend flexibly while imparting them with unique styles (a). Starting with a unit pattern that is tileable on a grid, our design tool simulates the bending behavior of the kerf pattern (i.e., tiled unit patterns) at interactive rates (b). If the initial pattern is less flexible than desired, users can apply our design principles and effectively modify its geometry to improve its flexibility (c). (We note that the initial and final pattern designs undergo the same amount of bending force.) Users can then incorporate the final, flexible pattern into their furniture project (d). Please zoom in to see the geometric details of the kerf pattern designs. . . . . 36
- 4.2 Given two neighboring unit patterns along the  $x$ -axis, the right boundary face of the left pattern (highlighted by the orange line) and the left boundary face of the right pattern (highlighted by the green line) must match when the patterns undergo bending (a). Since unit patterns are identical and bend exactly alike, the left boundary faces of both the left and right patterns must match via a rigid-body transformation. Therefore, when a unit pattern experiences bending, its left and right boundary faces must also match via a rigid-body transformation (b). . . . . 39
- 4.3 Given a kerf pattern, its *local geodesic* is the shortest path between any pair of leftmost and rightmost vertices of the pattern (each color—green, purple, and blue—denotes a different local geodesic) (a). The *global geodesic* or *geodesic* of a pattern is its shortest local geodesic (b). To compute the *curvature* formed by a kerf pattern along the  $x$ -axis, we approximate the top and bottom surfaces of the bent pattern with least-squares paraboloids and estimate the curvature as the average of principal curvatures of the two paraboloids at their centers along the  $x$ -axis (c). . . . . 42
- 4.4 The *geodesic region* of a kerf pattern is the region of the pattern along its geodesic (highlighted by the red dotted line) (a). When a pattern with a long geodesic undergoes bending (b), portions of the geodesic region twist as opposed to bend (c). These torsions collectively enable the geodesic region, as well as the pattern, to bend beyond the elastic limit of the fabrication material (e.g., this example pattern is fabricated with rigid birch plywood). . . . . 43
- 4.5 Given a line segment of the geodesic, its width at any point is the perpendicular line segment that intersects the kerf pattern (a) and its corresponding *bar segment* is the extrusion of the area covered its widths at all points (b). We define the geodesic region of a pattern as a sequence of bar segments corresponding to the line segments of the geodesic. When a pattern experiences bending (c), a more horizontally oriented bar segment of the geodesic region can only bend minimally due to the rigidity of the material (d). On the other hand, a more vertically oriented bar segment can twist significantly in spite of the rigid material and we measure its twist angle (labeled  $\varphi$ ) as the rotation degree of its cross section (the area highlighted by the purple line) (e). Accumulatively, the twist angles of the bar segments form the majority of the curvature of the geodesic region (f). . . . 44

- 4.6 A representative sample of the 75 unit kerf patterns designed to help identify geometric features of a pattern that are directly related to its curvature. We have created these 75 patterns to have a great diversity of decorative geometric designs (a–l), a large span of geodesic lengths from short (b, d, f, j) to long (e, g, k, l), and a wide range of curvatures from inflexible (a, b, d, j) to flexible (c, e, g, k). From these patterns, we have observed that they additionally differ in three geometric features associated with geodesics (Section 4.4.3): geodesic widths (a and b, c and d), geodesic orientations (e and f, g and h), and the number of distinct local geodesics (i and j, k and l, whose distinct local geodesics, excluding the geodesic, are delineated with red dotted lines). We size all unit kerf patterns to have the same dimension ( $3.5 \times 3.5 \times 0.03$  cm) and assign them properties of birch plywood (Young’s modulus  $E = 8.0 \times 10^9$  N/m<sup>2</sup> and Poisson’s ratio  $\nu = 0.3$ ), a common (rigid) fabrication material for laser cutters and CNC machines. We apply 40 N of bending force to all these patterns and compute their curvatures (labeled under the pattern names) as outlined in Section 4.4.1. To accentuate the curvature differences between unit kerf patterns, we tile them on a  $5 \times 3$  grid, along the  $x$ - and  $y$ -axis respectively, and visualize the bending behaviors of the resulting patterns. Please zoom in to see the geometric details of the pattern designs. We provide all 75 unit kerf patterns in Appendix B. . . . . 47
- 4.7 Correlations between curvatures formed by 75 unit kerf patterns (vertical axis) and their geodesic lengths (horizontal axis), as well as three geodesic measures. We scale the geodesic lengths and measures by their respective maximum values as the actual values are irrelevant to the correlation analysis. The colors and shapes of the highlighted data points correspond to the colors and row positions of the patterns in Figure 4.6 (squares and circles refer to the top and bottom rows, respectively). The analysis reveals that the geodesic length alone is insufficiently correlated with the pattern curvature (a), which depends on three additional geometric features of the geodesic. The geodesic measure  $M_1$  takes geodesic widths into consideration so that for patterns with comparable geodesic lengths (e.g., the Star and Wave patterns highlighted by the blue data points) (a), patterns with narrower geodesics receive greater values (b). The geodesic measure  $M_2$  additionally incorporates geodesic orientations and assigns larger values to patterns whose bar segments are more vertically oriented (e.g., the Wiggle pattern as compared to the Vine pattern) (c). Finally, the geodesic measure  $M_3$  further accounts for the number of distinct local geodesics and allot patterns with fewer distinct local geodesics proportionally higher values (e.g., the Pinwheel pattern relative to the Coin pattern) (d). By considering all three additional geometric features of the geodesic, the geodesic measure  $M_3$  achieves a strong correlation with the pattern curvature and they have a high  $R^2$  of 0.902 when fitted with a least-squares linear regression. . . . . 48

- 4.8 Our interactive kerf pattern design tool contains two main windows. The design window on the left shows the 2D unit pattern (a) and the simulation window on the right displays the 3D kerf pattern, which consists of extruded 2D unit patterns tiled on a  $3 \times 5$  grid (b). As users modify the geometry of the unit pattern, the design window highlights in real-time the updated global geodesic (the red line) and distinct local geodesics (the red dotted lines) of the resulting pattern (a), while the simulation window presents the updated bending behavior of the redesigned pattern based on the user-specified bending force (d). The simulation window can additionally show the bending behavior of a known flexible kerf pattern (the purple pattern) under the same bending force to help users better gauge the flexibility level of their pattern design (d). As the pattern undergoes bending, both windows optionally visualize the stress incurred by the pattern to spotlight regions experiencing high levels of deformations (c,d). These regions often fracture easily and users can choose to redesign them first as they edit the pattern to improve its flexibility. . . . . 49
- 4.9 For this lattice pattern design, we observe that its geodesics as highlighted by our tool are primarily horizontal. We thus update the (rotated) rectangles in the pattern to be hexagons in order to introduce vertical line segments to the geodesics (Design Principle #2). While this edit does improve the flexibility of the pattern (Step 2), the improvement is minor because the geodesics remain short. Therefore, we make the geodesics longer by modifying the geometries of the octagons (Design Principle #1) and the pattern becomes more flexible as expected (Step 3). Finally, we remove a few horizontal bars from the pattern (Step 4). This removal not only further lengthens the geodesics, but also reduces the number of distinct local geodesics from four to two (Design Principle #3), resulting in a significantly more flexible pattern design. We note that we have added solid bars to the left and right sides of the fabricated pattern only to enable us to bend the pattern more easily. . . . . 50
- 4.10 To improve the flexibility of this floret pattern, we first apply decorative cuts to its petals (Step 2). These cuts increase the geodesic lengths and consequently the flexibility level of the pattern. We then redesign the geometries of the lilies so that the geodesics contain more vertical line segments and the pattern bends more flexibly as a result (Step 3). We continue to edit the geometries of the lilies along the horizontal middle of the pattern to decrease the number of distinct local geodesics by half, which allows the pattern to be even more flexible (Step 4). This editing session shows that, by applying our design principles deliberately and aesthetically, users can use our tool to create appreciably more flexible patterns while remaining faithful to the original designs. . . . . 51

- 4.11 We start redesigning this fish pattern by adding wavy cutouts that resemble markings on tropical fish to the left half of the pattern (Step 2). These cutouts force the geodesics to zigzag as opposed to traversing straight through the pattern, therefore making the geodesics longer, as well as more vertically oriented, and the pattern more flexible. We proceed to create cutouts adapted from fish scales in the original design and incorporated them into the right half of the pattern (Step 3). These cutouts additionally lengthen the geodesics and enable the pattern to bend even more. We finish the editing by thickening the cutouts we have added and the flexibility level of the pattern further increases due to the decreased geodesic widths (Step 4). As evidenced by the final design, our tool empowers users to apply the design principles creatively and produce flexible, artistic patterns that take inspirations from the source patterns. . . . . 51

## Acknowledgments

First and foremost, I would like to thank my advisor Maneesh Agrawala for his invaluable guidance and mentorship throughout my graduate degree. From shepherding me through my first research project brainstorm, to motivating me during long hours before paper deadlines, to guiding me along many iterations of conference presentations, Maneesh has always been there to help me through all stages of the research process. I still remember that, for my first project on woodworking joinery, I needed to file down some wooden table parts I had machined so that they would properly fit and assemble into a table, but I did not have the correct tools and the deadline was approaching. After hearing what had happened, Maneesh went out of his way to bring me the appropriate wood rasps I needed from the local hardware store and I was able to build the table in time thanks to these tools. This story exemplifies the constant commitment Maneesh has for his students and I am truly grateful to have been one of them. While the COVID-19 pandemic over the last two years has been challenging for everyone, Maneesh has been nothing but patient and understanding towards my challenges and struggles. I feel tremendously lucky to have received his counsel and I could not have accomplished what I did if not for his support. Thanks a lot, Maneesh!

I would also like to thank the other members of my thesis committee, Björn Hartmann, Eric Paulos, and Sara McMains for their insightful feedback and perspectives on my dissertation. I am especially thankful for Björn for his administrative and academic assistance locally at Berkeley after Maneesh relocated to Stanford. Additionally, I feel very fortunate to have collaborated with a number of people throughout my graduate degree and my research would not have been possible without their input and encouragements. In particular, I would like to thank my internship mentor, Danny Kaufman, for offering me the opportunity to work at Adobe in Seattle and for providing me with thoughtful research advice, as well as Yotam Gingold and Doug James for their tremendous support.

Throughout my many years at Berkeley, my friends in and out of lab have given me an endless well of advice and fun. I thank my labmates Amy Pavel, Alex Hall, Steve Rubin, Sean Arietta, Jonathan Harper, Colorado Reed, Armin Samii, Jessica Hullman, and many others for their wonderful and lively conversations. I also thank Andrew Head and Sarah Sterman for entrusting me to be a GSI of their excellent CS 160 course, as well as my fellow GSI Forrest Huang, Vivien Nguyen, Varsha Ramakrishnan, and many others for working with me during office hours and project support sessions. Outside the lab, my friend William Walsh has not only taken me to many culinary delights around the Bay Area, but also helped me with photographing many 3D-printed and laser-cut models. The beautifully shot figures in my papers would not have been possible without his expertise and generosity. Last but not least, I am indescribably fortunate to have met my partner Jonathon Nessralla, who has always been there for me during good and bad times. We have cooked many scrumptious dishes, baked numerous delicious pastries, watched plenty teary movies, and petted a hand-scratching amount of mischievous cats. He makes me a better person!

Finally, I am forever indebted to my mother, ShuHua Li, and my father, DanYang Yao, who have been the most selfless people I will ever meet. I thank them for everything!



# Chapter 1

## Introduction

Traditional wood furniture often features decorative intrinsic joints and beautifully curved parts created with woodworking techniques that combine form with function. To craft such furniture, experienced woodworkers usually go through three distinct steps [68]. In the first step, woodworkers draw up by hand a *visual design* that not only describes the overall shape of the furniture but also delineates the detailed surface appearances of the joints and curved parts. In the second step, they manually *verify* whether their initial furniture design satisfies all the constraints imposed by the intended design scenario (e.g., furniture assembled via joints must be stable when put together) and they iterate between modifying and verifying their design until all the constraints are met. Finally, in the last step, they *fabricate* the furniture with conventional woodworking tools, such as hand saws and carving knives.

This traditional furniture crafting process demands experience and expertise in woodworking and has historically been inaccessible to amateur users. However, the continuing advancement and increasing availability of CAD programs and CNC machines have lowered the entry to the visual design and physical fabrication of the furniture, i.e., the first and last steps of the crafting process. Specifically, CAD programs such as AutoCAD and Illustrator allow users to create furniture models with desired visual appearances relatively easily; and CNC machines such as 3D printers and CNC routers help users physically make furniture pieces by automating most of the fabrication process.

But, neither CAD programs nor CNC machines assist users with verifying whether their visual design of the furniture fulfills all the necessary constraints, even though such verification often requires advanced knowledge of woodworking and mechanics not commonly held by everyday users. For example, to design furniture assembled with intrinsic joinery, users must be familiar with the intricate geometries of woodworking joints and ensure that the furniture parts meet the *geometric constraints* and assemble into the intended configuration without collision. Similarly, to design furniture with curved parts, users need to understand the mechanics of bending and take the *physical constraints* of the fabrication material into account so that the curved parts can bend into the desired shapes without fracturing. Manually verifying these constraints can be time consuming and mentally taxing even for expert woodworkers. For average users, the verification step of the crafting process presents

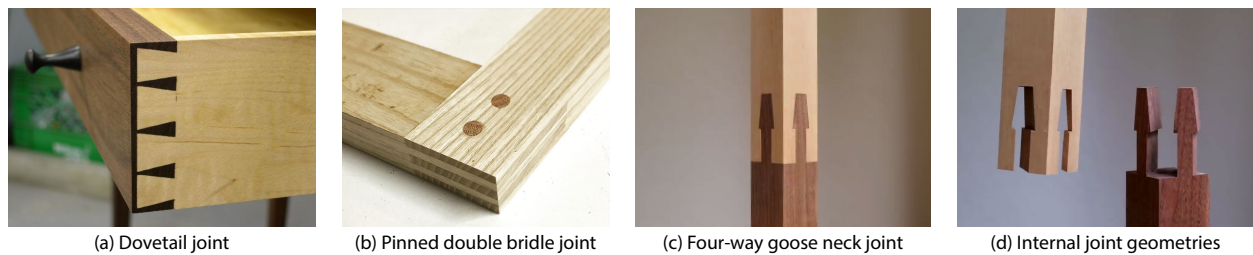


Figure 1.1: There are two general categories of intrinsic woodworking joints. Box joints (e.g., dovetail joints) connect parts perpendicularly to construct drawers or cabinets (a), while frame joints (e.g., pinned double bridle joints) attach parts planarly to build window frames or backs of chairs (b). While many joints have simple yet decorative surface appearance (c), their internal geometries are often surprisingly complex and difficult to design directly (d). (Please zoom in to see the geometric details of the joints.)

a serious challenge that prevents them from expressing their creativity in furniture design.

This thesis explores ways to help users verify their furniture designs and guide users towards visual designs that satisfy all the requisite geometric and physical constraints. We aim to develop algorithms that automate the tedious and difficult verification step, allowing users to focus on producing the visual design of their furniture rather than analyzing the validity of their design. In this thesis, we introduce interactive design tools that embody these goals, using algorithms informed by woodworking techniques and mechanical analyses to enable users to create (1) stable furniture models assembled via decorative intrinsic joints and (2) curved furniture parts formed through flexible kerf patterns.

## 1.1 Woodworking Techniques

We have studied a variety of woodworking books and websites on intrinsic joinery and kerf bending in order to guide the development of our interactive design tools [29, 34, 60, 19, 82, 70, 8, 46, 76, 4, 75, 89, 67, 48]. Here, we provide basic background on these two woodworking techniques and briefly discuss why they are challenging for amateur users to incorporate into their furniture design.

### Intrinsic Joinery

Modern ready-to-assemble furniture typically connects parts together via *external fasteners* such as nails and screws. In contrast, traditional hand-made furniture employs *intrinsic joints* to interlock parts along their mating surfaces through carefully sculpted geometries on the parts. These joints are valued for their structural integrity and aesthetic appeal. Based on the configuration of the interlocked furniture parts, woodworkers classify joints into two general categories [60, 19, 70]. They use *box joints* when parts are joined perpendicularly to build drawers and cabinets (e.g., the dovetail joint in Figure 1.1a). For window frames and

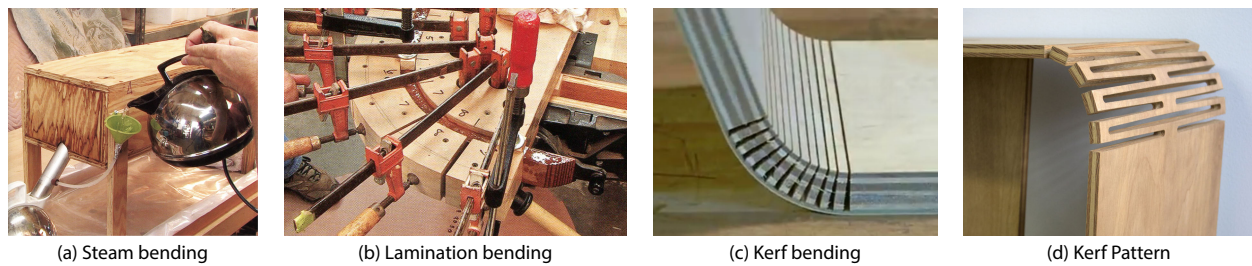


Figure 1.2: Woodworkers traditionally employ three different methods to bend wood. The steaming method softens timber pieces with heat and moisture in a steam box before bending them (a), while the lamination method bends thinly-sliced lumber strips before gluing them together (b). In comparison, the kerfing method is less labor-intensive and allows wood boards to bend flexibly by cutting deep notches into them (c). Inspired by the kerfing method, designers have created cutout patterns or kerf patterns that similarly enable rigid boards to become flexible after materials are removed from them based on the patterns (d).

backs of chairs where parts are planar, woodworkers utilize *frame joints* to attach the parts together (e.g., the pinned double bridle joint in Figure 1.1b).

While intrinsic joints often have relatively simple yet aesthetic surface appearance, figuring out their underlying geometries that match the surface appearance when assembled can often be challenging for average users. For example, the Japanese four-way goose neck joint in Figure 1.1c can only be taken apart diagonally and the necessary internal geometries that assemble together and conform to the surface appearance are surprisingly complex and difficult for users to design directly (Figure 1.1d). Moreover, when modeling joint geometries for furniture parts, users need to ensure that (1) the resulting parts do not obstruct one another during assembly and (2) the final assembled furniture remains stable under external forces such as gravity. Both of these requirements can be demanding for inexperienced users to manually verify.

## Kerf Bending

Classic aesthetic wood furniture often features elegantly curved parts that are thoughtfully crafted with wood bending. Compared to carving blocks of lumber into desired curved shapes, wood bending requires only narrow pieces and results in much less waste. Traditionally, woodworkers employ one of the three bending methods to form curved wood parts [75]. The *steaming* method softens and bends solid timber with heat and moisture, while the *lamination* method bends thinly-sliced lumber strips before gluing them together (Figure 1.2a,b) [76, 4]. Both methods are labor-intensive, allow wood to be flexible only temporarily, and require clamping the bent wood pieces overnight to set their shapes. In comparison, the *kerfing* method is considerably less time-consuming and involves only cutting deep, evenly-spaced notches into wood boards (Figure 1.2c). These notches weaken sections of wood boards, enabling them to be permanently flexible around the notch cutting direction [46, 89].

Inspired by the traditional kerfing method, professional designers have created various cutout patterns or *kerf patterns* that similarly allow rigid boards to bend flexibly around one axis after materials are removed from them according to the patterns (Figure 1.2d). But unlike the kerfing method that uses only repetitive notches, kerf patterns give users the artistic freedom to create ornamental designs that impart unique styles to their furniture. However, for everyday users, verifying whether kerf patterns are flexible can be difficult because straightforward visual inspections of their geometric designs do not obviously reveal their flexibility levels. As a result, users often create many patterns blindly in hopes that some will be flexible and resort to physically fabricating the patterns in order to evaluate their flexibility levels, which is costly in time and material and leads to an unproductive trial and error design process [6, 48].

## 1.2 Interactive Design Tools

In our research, we develop and create tools to address the aforementioned design challenges and empower users to create furniture designs that make effective use of woodworking techniques without prior knowledge. As an overview, in Chapter 2, we discuss related work in the area of design tools for fabrication. In the following two chapters, we describe our interactive tools for designing stable decorative joints and flexible kerf patterns. Finally, we conclude by summarizing the contributions and outlining future work in Chapter 5.

### Chapter 3: Designing Stable Decorative Joints

High-quality hand-made furniture often employs *intrinsic joints* that geometrically interlock along mating surfaces. Such joints increase the structural integrity of the furniture and add to its visual appeal. Chapter 3 presents an interactive tool for designing such intrinsic joints. Users draw the visual appearance of the joints on the surface of an input furniture model as groups of 2D regions that must belong to the same part. Our tool automatically partitions the furniture model into a set of solid 3D parts that conform to the user-specified 2D regions and assemble into the furniture. If the input does not merit assemblable solid 3D parts, our tool reports the failure and suggests options for redesigning the 2D surface regions so that they are assemblable. Similarly, if any parts in the resulting assembly are unstable, our tool suggests where additional 2D regions should be drawn to better interlock the parts and improve stability. To perform this stability analysis, we introduce a novel variational static analysis method that addresses shortcomings of the equilibrium method for our task. Specifically, our method correctly detects sliding instabilities and reports the locations and directions of sliding and hinging failures. We show that our tool can be used to generate over 100 joints inspired by traditional woodworking and Japanese joinery. We also design and fabricate 9 complete furniture assemblies that are stable and connected using only the intrinsic joints produced by our tool.

## Chapter 4: Designing Flexible Kerf Patterns

Inspired by traditional wood bending methods, *kerf patterns* allow rigid boards to bend flexibly by removing materials from them based on periodic cutout patterns. But, because the geometric designs of kerf patterns are not obviously related to their flexibility levels, assessing whether a pattern design is flexible is challenging without physical fabrication and if the design is inflexible, there exist no principles on how to redesign the pattern geometry to improve its flexibility. Chapter 4 introduces a design tool that enables users to interactively simulate the flexibility levels of their pattern designs and establishes a set of design principles that users can effectively apply to their pattern geometries for improved flexibilities. To perform simulations at interactive rates, we propose a fast, *periodic boundary condition*-constrained finite element method that exploits the periodicity of a kerf pattern by simulating only its constituent unit pattern. Through creating 75 unit kerf patterns and quantitatively analyzing the relationship between their flexibility levels and geometric designs, we show that the flexibility level of a pattern is well correlated with its *geodesic*, i.e., the shortest path between the leftmost and rightmost sides of the pattern. Our tool and geodesic-based design principles eliminate physical fabrication and redesign guesswork, allowing users to focus on the visual designs of their kerf patterns and giving them the creative freedom to prototype novel aesthetic flexible patterns.

### 1.3 Statement of Multiple Authorship and Prior Publication

The research presented in this thesis on the interactive tool for designing stable decorative joints is based on the paper published in ACM Transactions on Graphics 2017 [104]. I am the primary author of the publication, but this work could not have been accomplished without the help from my advisor Maneesh Agrawala and my co-authors who I am fortunate and grateful to have worked with. Danny Kaufman, my Adobe internship mentor, and Yotam Gingold have provided me with invaluable ideas and feedback for the project.

# Chapter 2

## Related Work

Developing interactive design tools for fabrication is an active area of research in the computer graphics and human–computer interaction communities. Since our work focuses on creating decorative joints and kerf patterns for fabricable furniture, we summarize the subset of techniques that are most relevant to this thesis. We additionally discuss research specifically related to the supporting algorithms of our design tools in Chapters 3 and 4.

### 2.1 Designing Furniture Assemblies

Our interactive joinery design tool allows users to create furniture assemblies that are connected using decorative joints. Researchers have developed a number of similar systems for designing fabricable furniture assemblies. The tools created by Lau et al. [40], Umetani et al. [93], and Schulz et al. [77] are particularly inspiring as they let users customize the overall shape of the furniture, while automatically adjusting the alignments and connections between the parts to ensure a fabricable result. Both Umetani et al. and Schulz et al. also analyze the physical stability of the resulting furniture. However, their methods rely on a standard set of external fasteners such as nails and screws to attach the parts together and do not account for the aesthetics of the joints.

Other works have introduced tools for designing furniture assemblies composed of flat boards of material that are connected using intrinsic slotted joints or finger joints [64, 74, 28, 81, 10, 54, 15, 3]. While these techniques can generate a wide variety of furniture shapes, the geometries of the joints between the parts are relatively rudimentary. Many recent systems have also aimed at creating furniture designs that facilitate physical fabrications. Yang et al. [103] use a material-aware database to replace furniture parts in order to simplify fabrication and Wu et al. [100] translate high-level geometric designs to low-level fabrication plans. Others enable users to easily fabricate jointed parts with portable CNC machines [90] and power tools [41]. These approaches use a library of parametrized joints and offer users limited ability to alter the joint designs.

### 2.1.1 Joinery Design

Several researchers have proposed computational systems that focus on creating joints for different furniture design scenarios.

Fu et al. [20] presented an automated tool for producing a set of intrinsic joints that interlock an input furniture assembly composed of rectangular boxes. Like earlier approaches on generating interlocking puzzle parts from input models [101, 86], the tool ensures that the resulting jointed parts are assemblable and interlocking, where every part is immobilized except for a single key that is free to move. Similar to the work by Fu et al., Song et al. [87] developed a method that generates intrinsic joints for interlocking furniture assemblies that can additionally be reconfigured into different shapes. Both techniques use a template set of woodworking joints (e.g., dovetail, mortise and tenon, etc.) and do not allow users to specify the design of the joints.

Jacobson [35] and Magrisso et al. [47] introduced systems to facilitate the design of 3D printed joints that can connect furniture parts at unconventional and visually appealing angles, enabling users to create and fabricate furniture shapes that are challenging to achieve with traditional intrinsic joints. Their systems additionally perform structural analysis on the user-designed joints and evaluate the stability of the assembled furniture. However, only tube-shaped joints are supported by both tools and users cannot meaningfully modify the geometries of the joints other than adjusting their dimensions and angles to fit the different connecting furniture parts.

Our work on joinery design is closest to that of Larsson et al. [39]. They recently proposed an interactive system for designing traditional Japanese wood joints that are directly fabricable with 3-axis CNC machines. Users manually create the geometries of the joints based on a 3D grid of voxels and the system provides users with suggestions to improve the CNC fabricability of their joint designs. While the tool empowers average users to fabricate furniture assemblies with relatively complex wood joints, its voxel-based joinery design space is fairly limiting and does not support the creation of free-form ornamental joints.

In contrast to these prior systems, our joinery design tool lets users directly delineate the 2D appearance of the joints on the furniture surface and automatically computes the necessary internal 3D joint geometries that conform to the user-specified surface appearance and ensure the stability of the assembled furniture.

## 2.2 Designing Physical Properties

A number of recent efforts have focused on designing fabricable objects with various optimized physical properties. These techniques focus on reducing stress [88], preserving balance [69], increasing aerodynamics [94], optimizing the moment of inertia [2], maintaining the stability of nailed together furniture [93] or rigid assemblies [97, 98, 83], allowing furniture to stack [43], fold [42], reconfigure [21], or interlock [96], and offsetting variations in laser cutters to ensure that fabricated objects precisely match user-specified dimensions [72].

Other methods reduce the amount of material required to fabricate objects [73, 95, 44] and split large models into small parts for fabrication in 3D printers with limited working volume [45, 13, 105] or due to other constraints [30, 27]. In comparison, our kerf pattern design tool allows users to create decorative patterns that enable rigid boards made of homogeneous material to bend significantly more than what the material elasticity normally permits.

### 2.2.1 Decorative Pattern Design

Several researchers have investigated ornamental pattern designs for objects that can achieve desired levels of structural integrity. Dumas et al. [18], Martinez et al. [53], and Schumacher et al. [78] automatically generate aesthetic cutout patterns on an object surface based on input texture or pattern exemplars and ensure that the resulting object is structurally sound under gravity or user-defined loading scenarios. Complementarily, Chen et al. [12], Ma et al. [11], and Zehnder et al. [106] presented automated design systems for producing an object surface that is formed from tightly packed ornamental filigrees, tiles, or curves, while guaranteeing that the generated patterns are well-connected and the resulting object is sufficiently strong. Additionally, Schumacher et al. and Zehnder et al. let users customize the aesthetics of their patterns by specifying the sizing or orientation fields of the patterns on the object surface. Our kerf pattern design tool similarly empower users to create decorative patterns, but with the goal of improving the flexibility levels of rigid boards.

### 2.2.2 Metamaterial Design

Many prior works have proposed techniques to design and fabricate deformable, elastic objects using heterogeneous materials. Bickel et al. [5] created heterogeneous objects consisting of parallel layers of homogeneous materials to replicate the deformations of real-world objects. Skouras et al. [85] and Xu et al. [102] both optimized a heterogeneous spatial distribution of materials within an object so that the object exhibits desired deformations.

In contrast to a heterogeneous material, a metamaterial derives its elastic property from the geometry of the material and can achieve various elasticities using a single homogeneous material. Panetta et al. [65] and Schumacher et al. [80] computed a library of micro-structures with varying elasticities and tiled them inside an object to match the target elastic behaviors. Martinez et al. [50, 51] procedurally generated structures with graded elasticities by varying the distribution of open-cell Voronoi foams. Ion et al. [32, 33] introduced a set of small-scale geometric structures that can be assembled to perform various mechanisms and exhibit different textures. Schumacher et al. [79] and Martinez et al. [52] presented methods to create periodically tiled sheets with desired elastic properties.

Our kerf pattern design tool similarly follows the metamaterial approach and alters the elastic behaviors of rigid boards using aesthetic kerf patterns. While previous techniques rely on 3D printed micro-structures that are not easily replicable with rigid boards made of wood or acrylic, users can readily fabricate kerf patterns on these boards using laser cutters and CNC machines.



# Chapter 3

## Designing Stable Decorative Joints

### 3.1 Introduction

Modern ready-to-assemble furniture made by manufacturers like IKEA is typically composed of a set of parts that are connected together via *external fasteners* such as screws, nails, bolts and pegs. In contrast, high-quality hand-made furniture often employs *intrinsic fasteners* or *joints* that are formed by carefully crafting the geometries of the parts to interlock along mating surfaces. While such joints are valued for their structural integrity, and aesthetic beauty, their 3D geometries can be difficult to design (Figure 3.1a and 3.11).

In practice, furniture designers begin by sketching the overall shape of the furniture and delineating the visible 2D surfaces of the parts that are connected at each joint [68]. These drawings directly visualize the surface aesthetics of the furniture, and commercial CAD software like SketchUp, AutoCAD and Rhino make it relatively easy to draw these 2D regions on the outer boundary of a solid furniture model. The challenge is to convert such 2D surface regions into a set of solid 3D parts that (1) conform to the specified 2D regions and (2) assemble without collision into the desired furniture shape. Often, even for seemingly simple 2D regions, the solid 3D parts that satisfy these two constraints are geometrically complex and therefore difficult for users to figure out and directly model in CAD software (e.g., in Figure 3.1c, the green leg of the table has to slide in at a diagonal angle to the tabletop). Moreover, for the resulting assembly to function as furniture it must not collapse under its own weight; the parts should remain stable and not move with respect to one another under external forces such as gravity.

We introduce an interactive tool for designing decorative joints such that the resulting parts can be assembled into structurally sound furniture. The input to our tool is a single solid 3D model of the furniture, and a partition of its outer surface into groups of 2D surface regions that must belong to the same part. Our tool automatically partitions the input solid model into a set of solid 3D parts, one per group of 2D regions, or reports that such a partitioning violates one of the structural constraints—conformance to 2D regions, assemblability, or stability. If parts violate the assemblability constraint, our tool visualizes

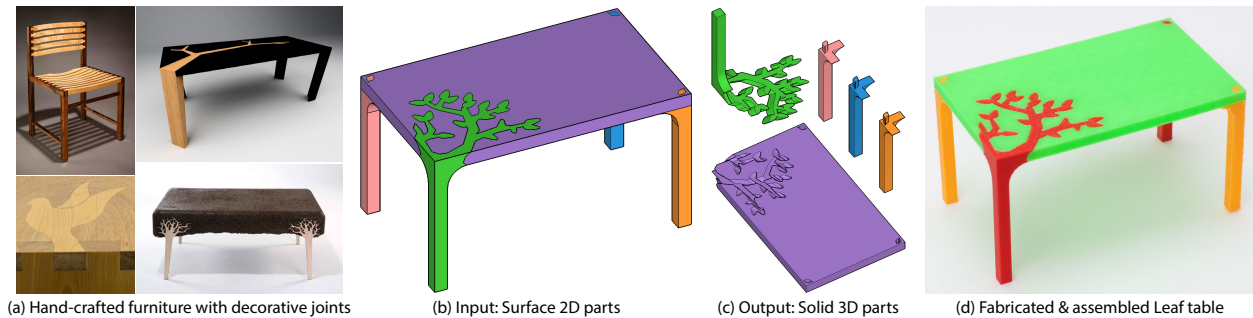


Figure 3.1: Hand-crafted furniture often includes decorative joinery that is valued for its strength and beauty (a). With our interactive joinery design tool, users draw the visible 2D surface regions corresponding to parts on the surface of a solid 3D furniture model (b). Our tool then constructs solid 3D parts that conform to the user-specified regions and assemble into the solid model (c, d). (Please zoom in to see joint details in furniture models.)

the collision regions between them and suggests how to redesign the 2D surface regions so that the resulting parts are collision-free. Similarly, if any parts violate the stability constraint, our tool suggests where additional 2D regions should be drawn to better interlock the parts and improve stability. To analyze the stability of furniture parts, we devise a novel variational static analysis solver, improving on the shortcomings of the widely employed equilibrium method [63] for our analysis tasks.

We demonstrate the generality of our tool by generating 100 joints inspired by traditional woodworking joints [70, 60], geometrically intricate Japanese joinery [82], and photographs of joints found on the Web. We also use our tool to design and fabricate 9 complete furniture assemblies that are stable and connected using only intrinsic joints. These results as well as evaluative feedback from amateur woodworkers, professional furniture designers, and novice first-time users suggest that our tool gives users the creative freedom to prototype new forms of decorative joints and furniture, while focusing on their visual appearance rather than on constructing their 3D solid geometry.

## 3.2 Overview

As shown in Figure 3.2, our interactive joinery design tool supports an iterative three stage workflow. In stage 1, users create a watertight solid 3D model and draw the *surface 2D parts*—2D regions on the model surface that must belong to the same part—using SketchUp. In stage 2, our tool constructs a set of *solid 3D parts* that each conform to one of the surface 2D parts and that together assemble into the input solid model. In stage 3, our tool checks the stability of the assembled furniture under the force of gravity and perturbations to gravity. If the parts are unstable, our tool suggests where the user might add more surface 2D part regions to improve stability. Users can then return to stage 1 and redesign the surface 2D parts to iteratively produce the final furniture model.

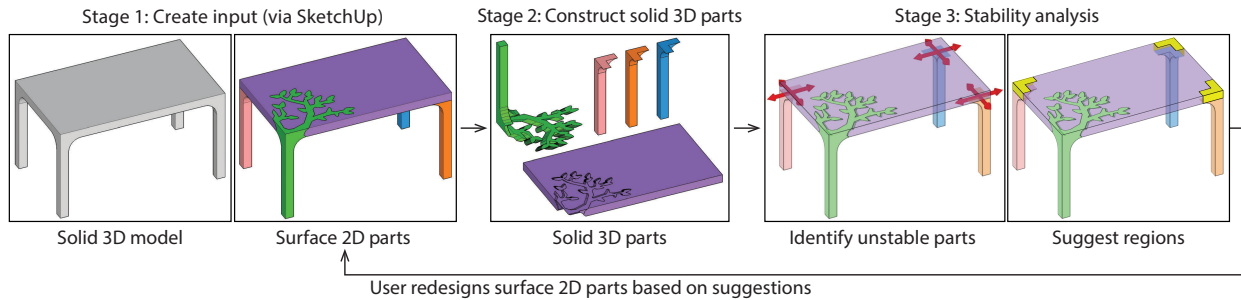


Figure 3.2: Three stage workflow of our interactive joinery design tool. In stage 1, the user creates a solid 3D model and draws a partition of the model’s exterior surface into groups of regions we call *surface 2D parts* (each color—green, orange, blue, pink, and purple—indicates a different surface 2D part). In stage 2, our tool either converts each such surface 2D part into a corresponding *solid 3D part* that together assemble into the input solid model, or reports that such conversion is infeasible. If the infeasibility is due to collisions between the surface 2D parts, our tool visualizes the collision regions and allows the user to quickly redesign the parts so that they are collision free. In stage 3, our tool uses physical simulation to check the stability of the solid 3D parts (the red arrows indicate that three legs of the Leaf table are unstable and will slide with respect to the table-top). If the parts are unstable, our tool suggest where the user might add 2D regions to a surface part in order to improve stability (highlighted yellow regions). The user can then go back to stage 1 and redesign the surface 2D parts based on the suggestions. Figure 3.1 shows the redesigned Leaf table with additional joints connecting the legs to the table-top.

### 3.3 Construct Solid 3D Parts

The first stage of our interactive joinery tool tries to construct a solid 3D part for each surface 2D part subject to the following conditions:

- (1) **Conformance to surface 2D part.** The solid 3D part must be a single solid model (it cannot be disjoint) and its visible surface in the assembled model must exactly coincide with its associated surface 2D part.
- (2) **Assemblability.** It must be possible to join the solid 3D parts to one another in a sequential order where each subsequent part is attached to the earlier parts using a single collision-free translational motion. We call this *sequential one-push assemblability (SOPA)*.

Prior work in assembly planning has shown that for an object composed of rigid parts, computing a collision-free sequence of motions that brings the parts into their assembled configuration is equivalent to starting with the object in its assembled configuration and computing a sequence of motions that separates (or disassembles) the parts [99, 91]. Therefore, our approach is to first identify all SOPA compatible disassembly sequences for the surface 2D parts (Section 3.3.1). We then iterate through each of the SOPA sequences and apply a two-pass algorithm for converting the surface 2D parts into solid 3D parts (Section 3.3.2). We stop the iteration as soon as we obtain a set of solid 3D parts that satisfy

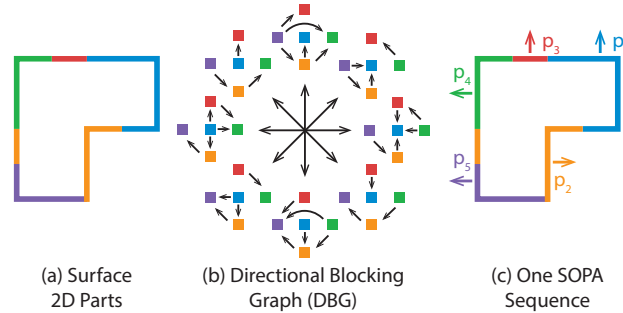


Figure 3.3: Given a solid 3D model with a set of surface 2D parts (a), we compute a directional blocking graph (DBG) in a discrete set of directions (8 in this 2D example, 26 in our 3D implementation) (b). For each DBG direction we compute a blocking graph. For example in the upward direction graph at the top of the DBG, the arrow from orange to blue means that the orange part is blocked by the blue part in the upward direction. We use the DBG to compute SOPA compatible disassembly sequences one of which is shown in (c).

the conformance and assemblability conditions. If all the SOPA sequences fail to meet these conditions, our tool reports which parts in each sequence failed so that the user can further modify those parts and try again (Section 3.3.3).

For clarity, we illustrate our construction algorithm with figures that show solid 3D models as 2D shapes and surface 2D parts as colored 1D edges (Figure 3.3–3.6).

### 3.3.1 Find SOPA Compatible Disassembly Sequences

To identify SOPA compatible disassembly sequences for the surface 2D parts, we first compute low-level blocking relationships between them. Specifically, we adapt the approach of Agrawala et al. [1] and compute a *directional blocking graph (DBG)* that encodes the set of directions in which each surface 2D part  $p$  is blocked by another surface 2D part  $q$  from translating arbitrarily far away (Figure 3.3a,b). Like Agrawala et al., we build the DBG for a discrete set of translation directions—in our case the 6 main axial directions, the 12 face diagonal directions, and the 8 corner diagonal directions. To identify the blocking relationships between pairs of parts, we first sweep each surface 2D part  $p$  in each of the translation directions far enough to escape the bounding box of the input solid model and then check if the resulting sweep volumes intersect any other surface 2D part  $q$ . Such an intersection implies that  $q$  blocks  $p$  in the corresponding translation direction.

Since our tool focuses on generating SOPA sequences, our DBG only considers blocking with respect to single translational motions. Note however, that extension of the DBG to consider multi-step translational motions and rotational motions is possible [25], but would significantly increase the size of the search space for disassembly sequences. We have found that nearly all joints that appear in woodworking books [82, 70, 60] can be assembled using the single translational motions allowed by SOPA sequences; and we leave extensions to other models of assemblability to future work.

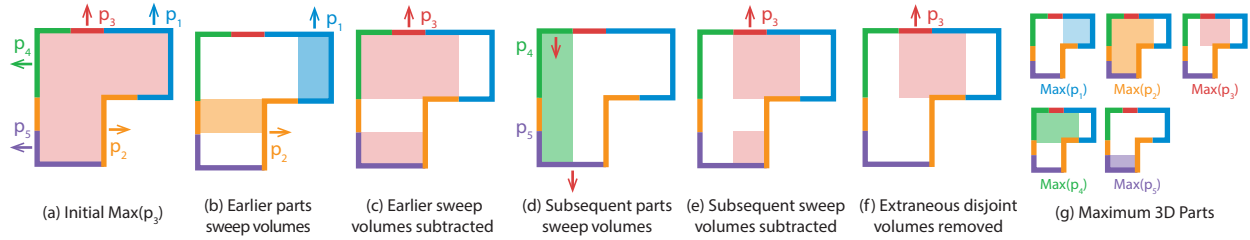


Figure 3.4: Given surface 2D parts and a SOPA sequence, we initialize  $Max(p_3)$  to include the complete input solid model (a). We then compute the sweep volumes of earlier surface 2D parts in their respective disassembly directions (b) and subtract them from  $Max(p_3)$  (c). We also compute the sweep volumes of subsequent surface 2D parts in the negative disassembly direction for  $p_3$  (d) and subtract them from  $Max(p_3)$  (e). Finally, we remove any extraneous volumes from  $Max(p_3)$  (f). We similarly generate all of the other maximum 3D parts (g).

We use the DBG to find the set of SOPA compatible disassembly sequences for the surface 2D parts as follows. For each permutation  $p_i, \dots, p_n$  of the surface 2D parts, we consider each part  $p_i$  in order and lookup the set of directions for which  $p_i$  is not blocked by the subsequent parts  $p_{i+1}, \dots, p_n$ . If every part in the permutation has at least one such unblocked direction, we add the permutation along with the unblocked directions for each part to our list of SOPA compatible disassembly sequences. Figure 3.3c shows one such SOPA compatible sequence.

### 3.3.2 Two-Pass Construction Algorithm

For each SOPA compatible disassembly sequence we obtain in the first step, we apply a two-pass algorithm that tries to construct a solid 3D part for each surface 2D part subject to the conformance and assemblability conditions. In the first pass, we compute the maximum portion of the input solid model volume that could belong to each surface 2D part  $p$ . We call this maximum volume the *maximum 3D part* and denote it  $Max(p)$ . In most cases, these maximum 3D parts intersect one another, so in the second pass we re-allocate the volume of the maximum 3D parts to remove such intersections, yielding a solid 3D part  $P$  for each surface 2D part  $p$ . All of our sweep, union, subtraction, and intersection operations are computed via constructive solid geometry (CSG) [108].

#### 3.3.2.1 Pass 1: Construct Maximum 3D Parts

Given a SOPA compatible disassembly sequence of surface 2D parts  $p_1, \dots, p_n$ , we compute a corresponding set of maximum 3D parts  $Max(p_1), \dots, Max(p_n)$ , where each  $Max(p_i)$  is the maximum portion of the solid model volume that can belong to  $p_i$  while remaining SOPA disassemblable from the other surface 2D parts (Figure 3.4). We initialize each  $Max(p_i)$  to the complete input solid model volume and then subtract volume to ensure disassemblability. We subtract volume based on two observations.

First, we observe that for each maximum 3D part  $Max(p_i)$  to be SOPA disassemblable from the other surface 2D parts, it cannot contain any of the volume swept out by the earlier parts  $p_1, \dots, p_{i-1}$  as they are translated in their respective disassembly directions (Figure 3.4b). Otherwise some earlier part would collide with  $Max(p_i)$  as it was disassembled. Therefore, we subtract all disassembly sweep volumes for earlier surface 2D parts from  $Max(p_i)$ , (Figure 3.4c).

Second, we observe that if surface 2D part  $p_i$  is SOPA disassemblable, we can translate it in its disassembly direction  $d_i$  away from the subsequent parts  $p_{i+1}, \dots, p_n$ , without collision. Equivalently, we must be able to translate the subsequent parts in the opposite direction  $-d_i$  without collision with  $p_i$ . Moreover, this translation of the subsequent parts sweeps out a volume that cannot be part of the maximum 3D part for  $p_i$ , and we therefore subtract it from  $Max(p_i)$ , (Figs. 3.4d,e).

After the subtractions, a maximum 3D part may contain disjoint volumes (Figure 3.4e). We cannot add volume to connect these disjoint volumes since the maximum 3D part is, by construction, the largest allowable volume under the SOPA condition. We can, however, remove any extraneous disjoint volume from the maximum 3D part that does not contain a portion of the associated surface 2D part on its surface (Figure 3.4f).

Recall from Section 3.3 that the conformance condition requires that the visible surface of each solid 3D part  $P_i$  must exactly coincide with its associated surface 2D part  $p_i$ . Since we will construct the solid 3D part in the second pass of the algorithm as a subset of the maximum 3D part  $Max(p_i)$ ,  $Max(p_i)$  must also contain  $p_i$  on its visible surface. Our volume subtraction approach guarantees this containment condition. By construction, the sweep volumes we subtract from  $Max(p_i)$  cannot contain  $p_i$ , because they are swept out based on SOPA disassembly directions. Therefore  $p_i$  must be contained in  $Max(p_i)$ . Note however, that  $Max(p_i)$  may include other surface 2D parts  $p_j$  on its surface. For example, in Figure 3.4f,  $Max(p_3)$  includes portions of  $p_1$  and  $p_2$  on its surface.

If each resulting maximum 3D part consists of a non-disjoint component, we proceed to the second pass of the algorithm. Otherwise, some maximum 3D part  $Max(p_i)$  remains disjoint and, since we removed extraneous volumes, a portion of the surface 2D part  $p_i$  lies on each disjoint piece. In this case, generating a non-disjoint solid 3D part  $P_i$  is impossible, so we skip the remainder of the two-pass construction and move on to the next SOPA sequence.

### 3.3.2.2 Pass 2: Re-Allocate Volume of Maximum 3D Parts

At this stage, each surface 2D part  $p_i$  has an associated maximum 3D part  $Max(p_i)$ . While the previous pass ensures that maximum 3D parts are each a single-component solid geometry whose surface includes all of the corresponding surface 2D part  $p_i$ , the maximum 3D parts often intersect each other (Figure 3.4g). In this pass, we remove volume from maximum 3D parts until they no longer intersect, resulting in our final set of solid 3D parts  $P_i$  that satisfy the conformance and assemblability conditions. Algorithm 1 provides pseudocode for our approach. For a gentler explanation, we initially describe a version of the algorithm where the intersection volume does not contain any surface 2D parts (e.g., we ignore pseudocode

---

**Algorithm 1:** Construct the final set of solid 3D parts  $P_1, \dots, P_n$  by re-allocating volume of maximum 3D parts  $Max(p_1), \dots, Max(p_n)$ .

---

**Input:** Surface 2D parts in the SOPA disassembly order  $p_1, \dots, p_n$  and the associated maximum 3D parts  $Max(p_1), \dots, Max(p_n)$ .

```

1 // Initialize solid 3D parts
2 foreach part  $Max(p_i)$  in SOPA order do
3   |  $P_i \leftarrow Max(p_i)$ ;
4 end
5 foreach part  $P_i$  in SOPA order do
6   | foreach subsequent part  $P_j, j > i$  in SOPA order do
7     |  $I_{ij} \leftarrow P_i \cap P_j$ ;
8     | if  $I_{ij} \equiv \emptyset$  then continue;
9     | // Check if we can remove  $I_{ij}$  from  $P_i$  and give it to  $P_j$ 
10    | if  $I_{ij} \cap p_i \neq \emptyset$  then  $T_i \leftarrow thicken(I_{ij} \cap p_i, P_j, d_i, I_{ij})$ ;
11    |  $B_{ij} \leftarrow block(P_i, I_{ij} - T_i)$ ;
12    | if  $(P_i - I_{ij} - B_{ij} + T_i)$  is not disjoint and contains  $p_i$  then
13    |   |  $P_i \leftarrow (P_i - I_{ij} - B_{ij} + T_i)$ ;
14    |   |  $P_j \leftarrow P_j - T_i$ ;
15    |   | continue;
16    | end
17    | // Check if we can remove  $I_{ij}$  from  $P_j$  and give it to  $P_i$ 
18    | if  $I_{ij} \cap p_j \neq \emptyset$  then  $T_j \leftarrow thicken(I_{ij} \cap p_j, P_i, d_i, I_{ij})$ ;
19    |  $B_{ij} \leftarrow block(P_j, T_j)$ ;
20    | if  $(P_j - I_{ij} + B_{ij} + T_j)$  is not disjoint and contains  $p_j$  then
21    |   |  $P_j \leftarrow (P_j - I_{ij} + B_{ij} + T_j)$ ;
22    |   |  $P_i \leftarrow P_i - B_{ij} - T_j$ ;
23    |   | continue;
24    | end
25    | // Terminate if neither removal possible
26    | terminate;
27   | end
28 end

```

---

operations shaded light gray). The gentle approach is illustrated in Figure 3.5 with a simpler model and three surface 2D parts.

We initially set the final solid 3D parts  $P_i$  to their corresponding maximum 3D parts (lines 2–4, Figure 3.5b). We then iterate over all pairs of parts to resolve their intersections.

If the pair of parts  $P_i$  and  $P_j, j > i$ , intersect, we first check whether we can give the entire intersection volume  $I_{ij}$  to  $P_j$  (lines 9–16). The gift is allowed if removing  $I_{ij}$  from

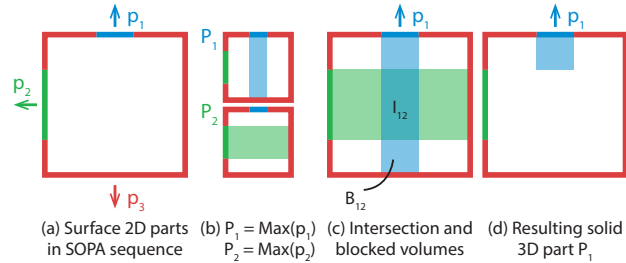


Figure 3.5: Surface 2D parts  $p_1$  and  $p_2$  (a) and their corresponding maximum 3D parts (b). The maximum 3D parts are used as the initial solid 3D parts  $P_1$  and  $P_2$ . To re-allocate the intersection volume  $I_{12}$  between  $P_1$  and  $P_2$  (c), we evaluate whether it can be given to  $p_2$ . Such a gift would block a portion of  $P_1$  (labeled  $B_{12}$ ) from being disassembled upwards with the rest of  $P_1$ . Because the solid 3D part  $P_1$  still contains its surface 2D part  $p_1$  even without the intersection and blocked volumes, the gift occurs (d).

$P_i$  still keeps  $P_i$  and its surface 2D part  $p_i$  connected. However, because  $P_i$  is disassembled before  $P_j$ ,  $P_j$  will block any volume in  $P_i$  located “in the shadow” of the intersection volume from being disassembled (Figure 3.5c,  $B_{12}$ ). We call volume *blocked* if it cannot be translated in its disassembly direction, because the volume of another part is in the way. To ensure disassemblability, we remove both the intersection volume and the volume blocked by it (lines 11–12, Figure 3.5c,  $B_{12}$ ) when testing whether the intersection volume can be given to  $P_j$ . If  $P_i$  and its surface 2D part  $p_i$  remain connected after the removal, we re-allocate the intersection volume and finish processing the pair of parts (lines 13–15).

If we cannot give the entire intersection volume to  $P_j$ , we check if we can give it all to  $P_i$  (lines 17–24). The operations are nearly symmetric, but we do not need to consider blocked volume because  $P_i$  is disassembled (removed) from the model before  $P_j$  is disassembled.

If neither re-allocation of the intersection volume keeps solid 3D parts connected to their respective surface 2D parts, we terminate the current two-pass construction and proceed to the next SOPA sequence (line 25–26).

The general case in which the intersection volume contains some or all of  $P_i$ ’s surface 2D part  $p_i$  is more complicated. Handling this case requires executing the operations of Algorithm 1 shaded in light gray. In the example of Figure 3.6a, giving all of intersection volume to  $P_2$  would disconnect  $P_1$  from  $p_1$ , guaranteeing failure. To account for this situation, when evaluating whether to give  $P_j$  the intersection volume (line 12), we *thicken* the portion of  $p_i$  that intersects the intersection volume and remove it from the gift (lines 10–14, Figure 3.6d-f). The algorithm performs a symmetric modification when considering giving the intersection volume to  $P_i$  (lines 18–22).

**Thickening.** Pseudocode for our *thicken* procedure is provided in Algorithm 2 and illustrated in Figure 3.6. The procedure takes as input a subset  $p_{sub}$  of a surface 2D part  $p$  to thicken, a solid 3D part  $Q$  for a different surface 2D part  $q$ , a disassembly direction  $d$ , and the intersection volume  $I$ . It outputs the volume  $T$  obtained by thickening every face of  $p_{sub}$



---

**Algorithm 2:** Thicken a subset of a surface 2D part  $p$  by taking volume from another intermediate solid 3D part  $Q$ .

---

**Input:** A subset  $p_{sub}$  of a surface 2D part  $p$ , an intermediate solid 3D part  $Q$  for a different surface 2D part  $q$ , a disassembly direction  $d$ , and an intersection volume  $I$ .

```

1 // Initialize thickening volume  $T$ 
2  $T \leftarrow \emptyset$ ;
3 // Try to thicken each face of  $p_{sub}$  by taking volume from  $Q$ 
4 foreach face  $f$  in  $p_{sub}$  do
5     // Choose thickening direction  $d_{thk}$ 
6     if  $f$  is inward facing w.r.t.  $d$  then  $d_{thk} \leftarrow d$ ;
7     else if  $f$  is outward facing w.r.t.  $d$  then  $d_{thk} \leftarrow -d$ ;
8     else  $d_{thk} \leftarrow$  interior facing normal of  $f$ ;
9     // Extrude  $f$  in direction  $d_{thk}$  by distance  $t_{dist}$  to build volume  $T_f$  for face  $f$ 
10     $q_{nnf} \leftarrow$  faces of  $q$  that do not neighbor  $f$ ;
11     $t_{dist} \leftarrow$  minimum distance from point on  $f$  to any face in  $q_{nnf}$ ;
12     $T_f \leftarrow extrude(f, d_{thk}, t_{dist}) \cap I$ ;
13    while  $(T_f \cap q_{nnf} \neq \emptyset)$  or  $(Q - T_f)$  is disjoint do
14        |  $t_{dist} \leftarrow \frac{1}{2}t_{dist}$ ;
15        |  $T_f \leftarrow extrude(f, d_{thk}, t_{dist}) \cap I$ ;
16    end
17    // Bisect  $T_f$  to prevent intersection with neighboring faces of  $q$ 
18     $q_{nf} \leftarrow$  faces of  $q$  that neighbor  $f$ ;
19    foreach face  $g$  in  $q_{nf}$  do
20        | if  $T_f \cap g \neq \emptyset$  then
21            | |  $G \leftarrow$  halfspace formed by plane through edge, between  $f$  and  $g$ , bisecting
22                | | the angle between them, and containing  $g$ ;
23            | |  $T_f \leftarrow T_f - G$ ;
24        | end
25    end
26    // Add  $T_f$  to total thickening volume  $T$ 
27     $T \leftarrow T \cup T_f$ ;
28 end
29 return  $T$ ;

```

---

(lines 1–4). For each such face  $f$ , we first determine the thickening direction  $d_{thk}$  by orienting the disassembly direction  $d$  towards the interior of the solid (lines 5–7). If, however,  $f$  is parallel to the disassembly direction, we thicken along the inward normal of  $f$  (line 8).

To thicken  $f$ , we seek an extrusion volume  $T_f$  that (i) does not intersect faces of  $q$ ; (ii) is

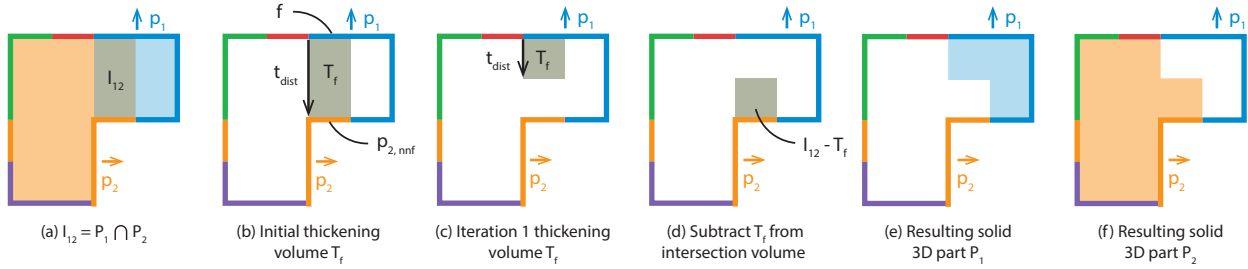
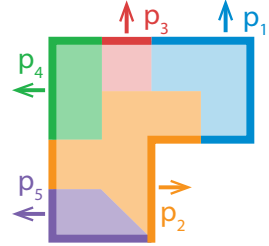


Figure 3.6: The intersection  $I_{12}$  between solid 3D parts  $P_1$  and  $P_2$  touches a portion of surface 2D part  $p_1$  (a). As a result, the intersection volume cannot be given to  $P_2$  without first removing some volume near the intersected portion of  $p_1$ . The initial thickening volume  $T_f$  for the intersected portion  $f$  is obtained by extruding  $f$  parallel to  $p_1$ 's disassembly direction until it touches a non-neighboring face of  $p_2$  (b). This initial thickening volume is iteratively reduced by half (c) until it no longer intersects  $p_{2,nnf}$ . The intersection volume is reduced by subtracting  $T_f$  (d) and can now be given to  $P_2$ , resulting in new solid 3D parts (e, f).

contained within the intersection volume  $I$ ; and (iii) keeps  $Q - T_f$  a connected solid geometry. At first, we are only concerned with intersecting faces of  $q$  that are not neighboring  $f$ , because neighboring faces of  $q$  can intersect the extrusion volume no matter the extrusion distance (lines 9–16). So, we collect the non-neighboring faces of  $q$  in a set  $q_{nnf}$  (line 10). We then find the minimum distance  $t_{dist}$  from  $f$  to any face in  $q_{nnf}$  along the thickening direction  $d_{thk}$  (line 11). If no such face exists, we set  $t_{dist}$  to “infinity”—any distance longer than the shape. To create the candidate thickened volume  $T_f$  satisfying (ii), we extrude  $f$  along direction  $d_{thk}$  and intersect it with  $I$  (line 12, Figure 3.6b). This candidate volume initially contacts a face in  $q_{nnf}$ , violating (i). To satisfy (iii) and (i) for non-neighboring faces, we repeatedly halve the extrusion distance and regenerate  $T_f$  (lines 13–16, Figure 3.6c). If no satisfactory extrusion distance can be found, we terminate and proceed to the next SOPA sequence.

To ensure that (i) is satisfied even for neighboring faces  $q_{nf}$  of  $q$  (lines 17–19), we further adjust  $T_f$  in a process we call *bisection*. For each face  $g$  in  $q_{nf}$  that intersects  $T_f$  (lines 17–20), we find the plane through the shared edge of  $f$  and  $g$  that bisects the angle between them. We form the half-space  $G$  defined by the side of the plane containing  $g$  (lines 21–22). We remove all volume from  $T_f$  that lies in  $G$ , thereby guaranteeing that  $T_f$  contains no faces of  $g$ . Finally, we join  $T_f$  with  $T$  and proceed to the next face in  $p_{sub}$  (line 27). When all faces of  $p_{sub}$  have been considered, the algorithm outputs  $T$ . The final set of solid 3D parts for our example is shown in inset.



### 3.3.3 Iterative Redesign for Infeasible Inputs

Since our tool does not impose any limitations on the surface 2D parts given as input, users can specify an infeasible set of surface 2D parts for which it is impossible to produce solid 3D parts that satisfy the conformance and assemblability conditions. One of the most common



### 3.4 Stability Analysis

Our construction algorithm generates a set of conforming, assemblable solid 3D parts. For this assembly to function as furniture, these parts must be stable under external forces. We call such an assembly *fully stable*. If an assembly is not fully stable, yet its parts nevertheless move together rigidly, we call it *part-stable*. Finally, if an unstable assembly has parts that move with respect to one another, we call it *part-unstable*. Well-designed joints form interlocking geometries that rigidly lock an assembled structure together to provide part stability under gravity. If the assembled structure is part-stable, the structure’s shape and material densities determine whether its center of mass lies over the structure’s support base. If so, it is fully stable and thus in static equilibrium; if not, it is part-stable but not fully stable. User-designed surface 2D parts do not necessarily generate solid 3D parts that, once assembled, are part-stable, as they may lack constraining joints between contacting parts (Figure 3.10a). If we know which parts are unstable and *in what directions* they can move, new joint geometries can then be designed to improve part stability.

To perform part-stability analysis for our furniture models, we require an analysis method that can (1) accurately predict whether an assembly is fully stable, part-stable, or part-unstable and (2) if the assembly is part-unstable, the method must report the locations and directions of sliding (translational) and hinging (rotational) accelerations that are generated by the part-instabilities so that these failures can be fixed with modifications to the joints.

The equilibrium method (EM) [31, 63, 97, 98, 83] is the current state of the art for assembly stability analysis in graphics (see Shin et al [83] for a thorough survey). Unfortunately, EM is not suitable for furniture part-stability analysis because it does not satisfy either of our requirements. EM can erroneously report assemblies as stable even when they are unstable.

EM is a constraint satisfaction method that formulates a set of physically-based constraints on admissible contact forces that can equilibrate a part assembly. If any feasible set of equilibrating forces are found, EM declares the assembly stable and returns a set of constraint-satisfying forces as a certificate. If not, the constraint system is declared infeasible. In this latter case, EM declares the assembly unstable. However, as a constraint-satisfaction problem, EM does not have a model of the unstable assembly forces and accelerations and therefore cannot determine whether the assembly is part-unstable, nor identify the locations and directions of sliding and hinging failures between parts if the assembly is part-unstable.

Moreover, EM does not model sliding [63] and in practice, as we will show, incorrectly reports stability in the presence of sliding failures. Although the masonry analysis literature assumes sliding failures do not occur due to high friction between blocks, this assumption is not valid for furniture assemblies where inadequately jointed furniture parts often suffer from translational sliding instabilities (Figure 3.9).

The failure to satisfy our two requirements makes EM unsuitable for our part-stability analysis. Instead, we propose a *variational static analysis* method that addresses EM’s failures. Our variational analysis correctly identifies instabilities when both hinging *and* sliding instabilities are present; it determines whether the assembly is part-unstable; and, if part-

instabilities are found, it provides the necessary per-part accelerations to report the locations and directions of sliding and hinging failures. This analysis lets us suggest additional joint regions to improve the part-stability of the assembly.

### 3.4.1 Static Analysis for Rigid Assemblies

For stability analysis, we treat our parts  $P_b$  as rigid, with each part equipped with rotational  $\mathbf{R}_b \in SO(3)$  and translational  $\mathbf{t}_b \in \mathbb{R}^3$  degrees of freedom. Per body we choose coordinates so that  $\mathbf{R}_b$  rotates from  $P_b$ 's principal-axis aligned body frame to world frame and  $\mathbf{t}_b$  gives the location of  $P_b$ 's center of mass in world frame. Corresponding angular and linear accelerations  $\dot{\boldsymbol{\omega}}_b, \ddot{\mathbf{t}}_b \in \mathbb{R}^3$  are concatenated into a single, system-wide acceleration vector

$$\ddot{\mathbf{q}} = (\ddot{\mathbf{t}}_1^T, \dot{\boldsymbol{\omega}}_1^T, \ddot{\mathbf{t}}_2^T, \dot{\boldsymbol{\omega}}_2^T, \dots)^T. \quad (3.1)$$

The corresponding, system-wide, block-diagonal mass matrix is  $\mathbf{M}$  (as in Kaufman et al. [37]). Each material point  $\mathbf{x} \in \mathbb{R}^3$  belonging to part  $P_b$  has corresponding constant body-frame coordinates  $\hat{\mathbf{x}}$  so that its world frame acceleration is  $\ddot{\mathbf{x}}(\ddot{\mathbf{q}}) = \ddot{\mathbf{t}}_b - \mathbf{R}_b \hat{\mathbf{x}} \times \dot{\boldsymbol{\omega}}_b$ .

**Contacts.** Contacts are between parts or between parts and a fixed boundary such as the ground. To simplify the following discussion, for each such contact  $k \in \mathcal{C}$ , the relative acceleration between two contacting points  $\mathbf{x}_i$  and  $\mathbf{x}_j \in \mathbb{R}^3$  (at contact  $k$ ) can be expressed via the linear map  $\Gamma_k : \ddot{\mathbf{q}} \rightarrow \ddot{\mathbf{x}}_i - \ddot{\mathbf{x}}_j$ . If  $\mathbf{f} \in \mathbb{R}^3$  is a force applied to point  $\mathbf{x}_i$  and an equal but opposite force is applied to point  $\mathbf{x}_j$ , then  $\Gamma_k^T \mathbf{f}$  is the resulting generalized force applied to the contacting system.

In turn, points in contact apply an equal and opposite force along their shared, unit-length normal  $\mathbf{n}_k$ . In global coordinates this is equivalent to applying a force of magnitude  $\alpha_k \in \mathbb{R}$  along a generalized normal

$$\mathbf{n}_k = \Gamma_k^T \mathbf{n}_k, \quad (3.2)$$

to the system of parts. The subspace of generalized normal directions

$$\mathbf{N} = (\mathbf{n}_1 \dots \mathbf{n}_{|\mathcal{C}|}) \quad (3.3)$$

then forms a basis for contact forces. Concatenating the corresponding force magnitudes in  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{|\mathcal{C}|})^T$ , the total contact force applied in the system is then  $\mathbf{N}\boldsymbol{\alpha}$ .

**Friction Forces.** A friction force, applied at a contact point, lies in the tangent plane orthogonal to the contact normal. At each contact  $k$ , we sample an orthogonal pair of unit length vectors from the tangent plane. The  $3 \times 2$  matrix composed column-wise of these samples is given by  $\mathbf{T}_k$ . A friction force,  $\mathbf{f}_k \in \mathbb{R}^3$ , applied at a contact  $k$ , lies in the span of  $\mathbf{T}_k$  so that  $\mathbf{f}_k = \mathbf{T}_k \boldsymbol{\beta}_k$ , where each  $\boldsymbol{\beta}_k = (\beta_1, \beta_2)_k \in \mathbb{R}^2$  gives the frictional response coefficients at the contact  $k$ .

The total friction force applied to the system at each contact  $k$  must be equal and opposite and is  $\mathbf{f}_k = \Gamma_k^T \mathbf{T}_k \boldsymbol{\beta}_k$ . The generalized basis for a friction force at contact  $k$  is then

$$\mathbf{D}_k = \Gamma_k^T \mathbf{T}_k. \quad (3.4)$$

We build the corresponding subspace of generalized tangent directions,

$$\mathbf{D} = (\mathbf{D}_1 \dots \mathbf{D}_{|\mathcal{C}|}). \quad (3.5)$$

and form the corresponding vector of frictional force coefficients as  $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_m)^T = ((\beta_1, \beta_2)_1, \dots, (\beta_1, \beta_2)_{|\mathcal{C}|})^T$ , so that the total friction force on the system is  $\mathbf{D}\boldsymbol{\beta}$ .

With gravity and any additional forces acting on the system summed per system DoF and stored in the system force vector  $\mathbf{g}$ , the total resultant force on the system is  $\mathbf{r} = \mathbf{N}\boldsymbol{\alpha} + \mathbf{D}\boldsymbol{\beta} + \mathbf{g}$ .

### 3.4.2 The Equilibrium Method

The equilibrium method (EM) seeks a set of feasible, equilibrating forces where the net torques and forces acting on all parts sum to zero. Concisely, it seeks a set of forces satisfying the static equilibrium condition

$$\mathbf{N}\boldsymbol{\alpha} + \mathbf{D}\boldsymbol{\beta} + \mathbf{g} = 0, \quad (3.6)$$

subject to the following two force feasibility conditions.

First, that contact forces are compressive or equivalently that their magnitude along normals are non-negative so that

$$\boldsymbol{\alpha} \geq 0. \quad (3.7)$$

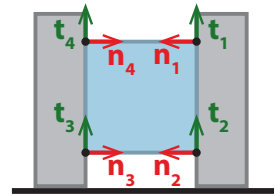
Second, that the isotropic Coulomb constraint is enforced. This restricts, per-contact, the magnitude of the friction force by the inequality

$$\|\mathbf{f}_k\| = \|\mathbf{T}_k \boldsymbol{\beta}_k\| = \|\boldsymbol{\beta}_k\| \leq \mu_k \alpha_k, \quad \forall k \in \mathcal{C}, \quad (3.8)$$

where  $\mu_k$  is the coefficient of friction and  $\alpha_k$  is the normal (contact) force magnitude at  $k$ . This restricts the friction force to lie within a disk in the tangent plane with a radius of  $\mu_k \alpha_k$ . Applications [31, 97, 98, 83] often further linearize this inequality with a polyhedral approximation and apply a QP or LP objective to seek these forces. No matter the method of finding it, if a feasible equilibrating set of forces  $\boldsymbol{\alpha}$  and  $\boldsymbol{\beta}$  satisfying (3.6), (3.7), and (3.8) exist, EM *claims* the structure is stable.

As Oschendorf [63] notes, EM cannot accurately predict when parts will slide against one another. EM is widely deployed for masonry analysis tasks under the assumption that strong friction forces between blocks prevent such sliding failures. This assumption is unfortunately not valid for furniture assemblies where inadequately jointed furniture parts often suffer from sliding instabilities (Figure 3.9).

To illustrate EM’s inability to accurately predict sliding instabilities, consider the following didactic example. A single free block is placed in contact between two *fixed* vertical walls (inset), with gravity pointing downwards along the  $y$ -axis. Their contacting surfaces are geometrically flat and there is no compression between the block and two walls. This system is symmetric so we may equivalently consider it in the plane. With a linear force distribution on contacting faces, we sample just the contact points at the four corners of the block (now a square). Contact normals point *inward* from the wall towards the block, along the  $x$ -axis, while in 2D we require just a single tangent direction per contact,  $\mathbf{t}_k$ , pointing upwards along the  $y$ -axis. The static equilibrium conditions for force and torque balance in this simple case are then respectively



$$\sum_{k \in [1,4]} (\mathbf{n}_k \alpha_k + \mathbf{t}_k \beta_k) + \mathbf{g} = 0 \quad \text{and} \quad \sum_{k \in [1,4]} (\hat{\mathbf{x}}_k \times \mathbf{n}_k \alpha_k + \hat{\mathbf{x}}_k \times \mathbf{t}_k \beta_k) = 0. \quad (3.9)$$

Notice in (3.9) that *any* set of feasible, uniform contact forces  $\alpha_k = a \in \mathbb{R}, \forall k \in [1, 4]$ ,  $a > 0$  entirely cancels out to a net torque and force equal to zero. This means that EM’s feasibility constraints allow us to apply an arbitrarily large set of uniform contact forces. When substituted into Coulomb’s constraint (3.8), arbitrarily large contact forces allow us to apply arbitrarily large friction forces. If we similarly make friction forces uniform and positive, i.e.,  $\beta_k = b \in \mathbb{R}, \forall k \in [1, 4]$  with  $b > 0$ , their net *torque* contributions cancel out, leaving us able to apply an arbitrarily large, feasible net upward force along the  $y$ -axis to balance the downward acceleration of gravity irrespective of how heavy we make the block.

No matter what coefficient of friction is used in this example, the correct solution for this configuration is that the block should always fall under gravity with no resistance. But the sliding instability is entirely missed here by EM. Instead EM incorrectly declares the single sliding block configuration to be stable. Moreover, if we turn to an example where both the vertical walls and the block are *free*, infinite forces are no longer permitted by EM. However, the EM code [84] still continues to incorrectly claim equilibrium for this system of three free parts that have no joints to hold them together; see Figure 3.8, rightmost column. Here all parts are free and the assembly should be unstable with the middle block sliding. More broadly, EM is missing adequate models of constraint and friction forces to properly predict and analyze sliding instabilities.

Independent of accuracy, when EM reports an instability, it does not and cannot report whether there are part-instabilities or not. By construction, if an assembly is determined unstable, EM is simply an oracle and does not provide any additional information on which parts of the assembly are unstable, or in which direction the parts will move relative to one another. The solid 3D parts generated by user-designed surface 2D parts often suffer from multiple stability failures (Figure 3.9 and 3.10). Without information on which parts are unstable, we cannot offer meaningful suggestions to users on how to improve the part-stabilities for the furniture assembly.

Prior work in masonry analysis has similarly needed to confront EM’s inability to analyze instability. In order to optimize structures to satisfy EM constraints, Whiting et al. [97]

remove EM's non-negativity constraint in Equation (3.7). This temporarily allows for inadmissible, tension forces along normal directions at contacts. They then minimize the squared norm of these tensile forces to find a design that best satisfies EM's material compression constraint. While this process seeks a new design satisfying EM feasibility, the defined metric does not identify the locations nor directions of sliding and hinging failures between parts.

### 3.4.3 Variational Static Analysis

To build our static analysis solver, we begin with the observation that EM makes well-founded physical assumptions, yet it misses sliding instabilities by finding physically invalid or *unrealizable* forces—something must be missing. As illustrated in the single block with two fixed walls example above, EM is underconstrained and can greedily find physically infeasible contact forces leading to incorrect predictions. To prevent such unrealizable forces, our solver invokes a pair of variational principles from classical mechanics, Gauss's Least Constraint [57] and the Principle of Maximal Dissipation [24]. These two principles restrict the static analysis model to a predictive set of feasible frictional-contact forces.

We start with Gauss's Least Constraint which, geometrically interpreted, tells us that realizable contact-constraint forces should be *minimal*. We then first require a constraint. We start with contact constraints: *points in contact should not interpenetrate*. To prevent interpenetration at a contact  $k$ , the relative acceleration between the two contacting points along their normal  $\mathbf{n}_k$ , given by  $\mathbf{n}_k^T \Gamma_k \ddot{\mathbf{q}}$ , must be non-negative. This is equivalent to enforcing the contact constraint  $\mathbf{n}_k^T \Gamma_k \ddot{\mathbf{q}} \geq 0$ . The equivalent non-penetration inequality constraint, for all points of contact simultaneously, is then

$$\mathbf{N}^T \ddot{\mathbf{q}} \geq 0. \quad (3.10)$$

Gauss's Least Constraint for contacts requires that the realizable constraint forces  $\boldsymbol{\alpha}^*$  must be minimal so that

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{N}\boldsymbol{\alpha}\|_{\mathbf{M}^{-1}}^2 \quad \text{s.t.} \quad \mathbf{N}^T \ddot{\mathbf{q}} \geq 0. \quad (3.11)$$

**Maximal Dissipation Principle.** Letting  $\mathbf{f}_k$  denote a frictional force applied at a contact point,  $\mathbf{x}_k$ , the *Maximal Dissipation Principle* requires that the friction to maximize the rate of negative work done at the contact. Adopting an instantaneous view, we maximize  $-\mathbf{f}_k^T \ddot{\mathbf{x}}$ . Combined with the Coulomb friction constraint, maximal dissipation provides an acceleration-level interpretation of the familiar Coulomb friction law.

We then find friction forces for a system in contact, with no velocity, by enforcing Maximal Dissipation simultaneously at *all* contacts to obtain an instantaneous, global minimization

$$\max_{\mathbf{f}_k} \sum_{k \in \mathcal{C}} (-\mathbf{f}_k^T \ddot{\mathbf{x}}_k) = \min_{\mathbf{f}_k} \left[ \sum_{k \in \mathcal{C}} \mathbf{f}_k^T \Gamma_k \right] \ddot{\mathbf{q}} = \min_{\mathbf{f}} \mathbf{f}^T \ddot{\mathbf{q}}. \quad (3.12)$$



**Equilibrium Testing.** Substituting,  $\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{N}\boldsymbol{\alpha} + \mathbf{D}\boldsymbol{\beta} + \mathbf{g})$ , Gauss's least constraint is equivalently the convex Quadratic Program (QP)

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{N}^T \mathbf{M}^{-1} \mathbf{N} \boldsymbol{\alpha} \quad s.t. \quad \mathbf{N}^T \mathbf{M}^{-1} (\mathbf{N}\boldsymbol{\alpha} + \mathbf{D}\boldsymbol{\beta} + \mathbf{g}) \geq 0. \quad (3.13)$$

Similarly, substituting acceleration into (3.12) we must simultaneously satisfy Maximal Dissipation as a Quadratically Constrained QP

$$\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^T \mathbf{D}^T \mathbf{M}^{-1} \mathbf{D} \boldsymbol{\beta} + \boldsymbol{\beta}^T \mathbf{D}^T \mathbf{M}^{-1} (\mathbf{N}\boldsymbol{\alpha} + \mathbf{g}) \quad s.t. \quad \|\boldsymbol{\beta}_k\| \leq \mu_k \alpha_k, \quad \forall k \in \mathcal{C}. \quad (3.14)$$

As the global system's forces must satisfy both (3.13) and (3.14) at the same time, we solve for unknown contact and friction forces with a Staggered Projections [37] sequence. We initialize (3.13) with a starting  $\boldsymbol{\beta}$  and then iterate between the solutions of the two minimizations to convergence with a relative tolerance of  $10^{-4}$ .

The solution is the total force on the system  $\mathbf{r}^* = \mathbf{N}\boldsymbol{\alpha}^* + \mathbf{D}\boldsymbol{\beta}^* + \mathbf{g}$ . We have force balance and thus equilibrium if  $\|\mathbf{r}^*\| = 0$ . Force balance guarantees that all system parts are at relative rest and that the system is not moving. If we are not at equilibrium, our analysis models the total out of balance forces  $\mathbf{r}^* > 0$  on the assembly. This allows us to compute the relative accelerations between parts in order to determine which parts are unstable and in what direction they are moving. We compute the relative accelerations between part contact pairs  $i$  and  $j$  of each contact  $k$  as

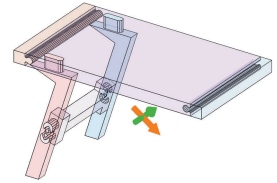
$$\mathbf{a}_k = \ddot{\mathbf{x}}_i - \ddot{\mathbf{x}}_j = \Gamma_k \mathbf{M}^{-1} \mathbf{r}^* \in \mathbb{R}^3, \quad (3.15)$$

and extract individual rigid part accelerations from the system acceleration vector  $\ddot{\mathbf{q}} = \mathbf{M}^{-1} \mathbf{r}^*$ .

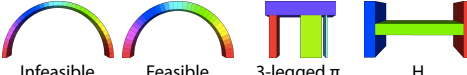
For a furniture assembly, our system reports one of three possible stability states after an execution of the stability analysis:

**Fully stable:** If all relative accelerations,  $\mathbf{a}_k$ , and individual accelerations,  $\ddot{\mathbf{x}}_i, \ddot{\mathbf{x}}_j$ , are zero for all contacts then we have a *fully stable* assembly.

**Part-stable:** (but not fully stable) If all relative accelerations,  $\mathbf{a}_k$ , are zero *but* there are individual non-zero accelerations,  $\ddot{\mathbf{x}}_i, \ddot{\mathbf{x}}_j$ , the system is part-stable but not at equilibrium. All joints are holding parts together *but* the entire furniture assembly is out of balance and needs reshaping. The parts of the inset table are interlocked with joints, yet, because it is missing two legs, the table is unbalanced and will fall as a rigid unit. Our interface shows the *global* linear and angular accelerations with orange and green arrows, respectively.



**Part-unstable:** If there exist nonzero relative accelerations,  $\mathbf{a}_k \neq 0$ , then parts in contact are moving relative to one another, and the system is part-unstable. The relative accelerations allow us to determine which parts of the assembly are unstable, as well as the locations and directions of their relative sliding and hinging instabilities. We then offer suggestions, detailed in Section 3.4.5, on where additional joint regions may be added to improve the part-stability of the assembly.



		Infeasible	Feasible	3-legged $\pi$	H
	# blocks	36	36	4	3
stability at $0^\circ$	Analytic	U	S	S	U
	Experiment	-	S	S	U
	Variational	U	S	S	U
	EM	U	S	S	S
tilt test $\theta^\circ$	Analytic	N/A	8.2	-	N/A
	Experiment	N/A	$4.7 \pm 0.2$	$14.3 \pm 0.2$	N/A
	Variational	N/A	5.3	16.4	N/A
	EM	N/A	8.2	19.1	N/A

Figure 3.8: Results of stability and tilt tests comparing analytic and experimental results with analyses from our variational static solver and the equilibrium method (EM) implementation analyzed by Shin et al. [83]. To match with prior results [83], we follow the reported material density,  $1.5 \text{ g/cm}^3$ , and friction angle,  $43^\circ$ . “S” and “U” indicate stable and unstable. For stability determination on a horizontal ground plane, our variational solver matches with the analytic and experimental solutions for all assemblies. EM differs by missing the sliding instability for the H assembly. For the tilt tests, our variational solver predicts a critical tilt angle between the angles experimentally determined and those predicted by EM. Note that mesh colors indicate block number and do not depict stress.

### 3.4.4 Evaluation

We evaluate our variational static solver with a suite of benchmark examples (Figure 3.8), a collection of inadequately jointed furniture models with a range of sliding and hinging instabilities (Figure 3.9), and all of our fabricated furniture examples (Figure 3.1 and 3.12).

In Figure 3.8, we compare the analysis of our variational static solver in determining stability to the EM implementation of Shin et al. [83]. We compute the stability of each assembly for the horizontal ground plane at  $0^\circ$  (top four rows). If an assembly is stable at horizontal, we additionally determine the critical ground-plane tilt angle where the assembly becomes unstable (bottom four rows). Under the assumption of no sliding failures, *circular masonry arches* have analytic solutions for their stability on horizontal and tilted grounds given by the thickness-to-radius ratio  $t/r$ ; arches are increasingly stable as  $t/r$  grows [63]. Here we consider an *infeasible* arch with  $t/r = 0.08$  and a *feasible* arch with  $t/r = 0.15$ . Our variational solver correctly determines the infeasible arch unstable and the feasible arch stable. As we tilt the ground plane, the variational solver predicts the critical tilt angle beyond which the feasible arch is unstable at  $5.3^\circ$ . This angle is between the experimentally determined angle of  $4.7^\circ$  and the  $8.2^\circ$  determined by EM [83]. While the analytic solution for the critical tilt angle is given as  $8.2^\circ$ , both the analytic and EM solutions neglect treatment of finite friction with sliding, opening an interesting question on the degree to which results differ due to treatment of friction and sliding.

The *three-legged  $\pi$*  assembly evaluates equilibrium analysis with static indeterminacy. Our variational solver determines the  $\pi$  assembly stable on a horizontal ground plane and computes the critical ground plane tilt angle to be  $16.4^\circ$ . As in the masonry arch examples,

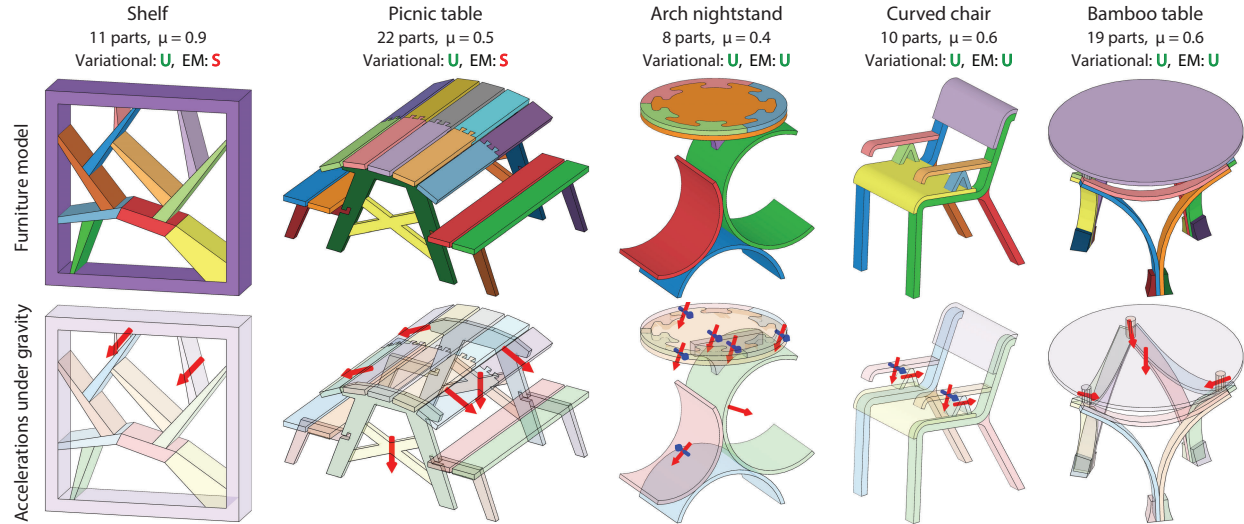


Figure 3.9: Results of stability analyses from our variational solver and the equilibrium method (EM) on five inadequately jointed furniture models under gravity. “S” and “U” indicate stable and unstable assembly results per method, respectively. All examples should be determined unstable in the analysis. Note that EM incorrectly determines the two leftmost assemblies stable. In addition to analyzing whether the furniture assembly is stable or not, for furniture design we require an analysis of where and how parts are sliding or hinging against each other if the system is found part-unstable. This is information that EM cannot provide. Our variational solver identifies the location and direction of sliding and hinging part-instabilities between furniture parts (bottom row). This allows our tool to suggest additional joint regions to improve part stabilities. For the bamboo table, the six small braces near the ground are modeled with density  $8.0\text{ g/cm}^3$  and all other parts are modeled with density  $0.3\text{ g/cm}^3$ , while the four remaining models use a density of  $0.5\text{ g/cm}^3$ . The corresponding friction coefficients  $\mu$  used for each model are summarized above. Red and blue arrows, if present, give the local linear and angular acceleration of unstable parts (i.e., the locations and directions of the sliding and hinging instabilities, respectively).

this critical angle is again between the experimentally determined angle of  $14.3^\circ$  and the angle of  $19.1^\circ$  determined by EM [83].

We also introduce the  $H$  assembly example to evaluate sliding instability. For this example, we place a horizontal block between two vertical free-standing blocks *without* any connecting joints. All assembly parts are *free* and without connections so that the middle block should fall towards the ground. Our variational solver captures the expected sliding instability with a predicted downward acceleration while the EM code incorrectly declares stability for the unsupported middle block.

In addition to these benchmark examples, we compare our variational solver with a state-of-the-art EM code [83] on five inadequately jointed furniture assembly models with complex geometries and non-trivial instability modes (Figure 3.9). Gravity points downwards along the  $y$ -axis for all examples, and the same material densities and friction coefficients are used

for both methods, as detailed in the figure. Red and blue arrows indicate translational and rotational accelerations of unstable parts respectively. Both our solver and the EM code agree and report instability for three out of five models. However, only our solver correctly captures the sliding instabilities in the *shelf* and the *picnic table* examples. The EM code incorrectly declares these examples stable. The pink and light purple parts at the top of the shelf slide due to the significant slope of the contacting blue and light green parts below them. The cross bars of the picnic table fall freely due to lack of support, while the dovetail-jointed outer slabs of the table top slide due to insufficient friction.

Unlike EM, our solver not only determines if an assembly is stable, but also finds the parts that are unstable and presents the location and direction of each sliding and hinging instability (Figure 3.9, bottom row). For the *arch nightstand*, our solver determines that the red and green arches are unstable, as well as the purple support part just below the table-top and the four lap-jointed table-top parts that rests upon it. The *curved chair*'s triangular arm-rest supports do not balance the arm rests. Our solver correctly captures the hinging and sliding failures. As the arm rests pivot and fall, the triangular supports slide along the chair seat. This is also captured by our solver. Note that triangular supports do not exhibit hinging failure due to their triangular shape. In the *bamboo table* example, the six small braces near the ground have a higher density of metal ( $8.0\text{ g/cm}^3$ ), while the nine arches, the table top, and three supporting pegs underneath the table top have the (significantly lighter) density of bamboo ( $0.3\text{ g/cm}^3$ ). Our solver reports that the nine contacting arches are stable, due to the heavy braces near the ground. However, the three pegs supporting the table top slide toward the center. As a result, the table top falls downward. Finally, all of our *fabricated furniture examples* (Figure 3.1 and 3.12) are determined stable by our variational solver and then validated as stable upon assembly of the final fabricated parts.

### 3.4.5 Visualizing Instability

In our joinery design interface, we analyze the part-stability of the solid 3D furniture parts generated by our construction algorithm, using our variational solver five times. First, we run our solver with gravity alone (pointing downwards along the  $y$ -axis). We then run our solver four additional times with perturbational forces added to gravity along the positive and negative  $x$ - and  $z$ -axes covering all forces in the downward hemisphere of directions, including horizontal forces. These perturbations allow us to suggest new regions for unattached parts such as tabletops (Figure 3.10) or table legs that should be attached but would be found part-stable under perfect conditions. While we focus on stability under gravity in this work, our solver supports analysis with arbitrary external forces; our interface can be extended to model additional specified loads and forces, such as the weight of a load on a seat or shelf.

If part-instability is reported, our tool guides the user through all pairs of contacting parts that are not part-stable, visualizing all predicted motions with arrows. Relative linear and angular accelerations  $\mathbf{a}_k$  computed by Equation (3.15) are shown with red and blue arrows, respectively (Figure 3.10b). Global linear and angular accelerations are shown with orange and green arrows, respectively (inset figure above). For every unstable pair of parts,

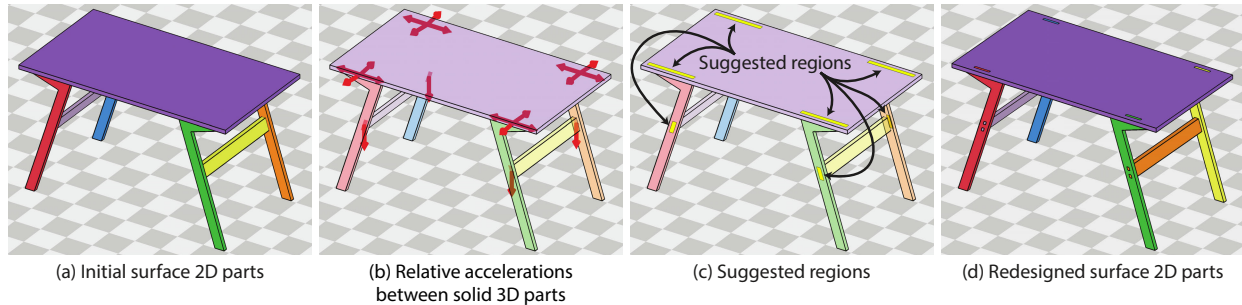


Figure 3.10: An initial set of surface 2D parts (a) generates unstable solid 3D parts (b). Our tool reports the part instabilities and presents the relative linear accelerations between unstable parts as red arrows. Our tool highlights suggested regions in yellow on the model surface where additional 2D regions can be drawn to improve part stability (c). Re-design with additional 2D regions leads to a stable 3D part assembly (d).

we highlight suggested faces on the model surface where additional 2D regions can be drawn to oppose the reported part instabilities (Fig 3.10c). To find such faces, we observe that each contact patch between a pair of unstable parts represents a site on which joints can potentially be added to prevent relative motion. We assume that added joints must lie inside the extrusion volume of the contact patch in either of its normal directions; this is standard practice in the woodworking literature for joint placement. So, for each contact patch, we find the extrusion volume that lies in either part. The suggested faces are the visible faces of this volume that lie on one of the two parts (Figure 3.10 c, d)—the part which would be less covered or “erased” by the suggestion.

## 3.5 Results

We have used our interactive joinery design tool to generate 100 different decorative joints (Figure 3.11 shows representative examples and Appendix A contains all of the joints we designed) as well as 9 complete furniture assemblies containing multiple parts and joints (Figure 3.1 and 3.12). We have also evaluated our system with amateur woodworkers, professional furniture designers and novice first-time users.

To create the joints (Figure 3.11 and Appendix A) we drew the surface 2D parts based on photographs we found in traditional woodworking guides [70, 60] (e.g., inlay dovetail, blind dovetail), Japanese joinery books [82] (e.g., four-way kanawa, puzzle joint) and examples from the web (e.g., arrow in Appendix A). We replicated 52 of the 62 joints described at two popular woodworking websites<sup>1,2</sup>, as well as 38 of the 40 Japanese joints of Seike [82]. All of the remaining joints (10 from websites, 2 from Japanese joinery book) require rotations or multi-step translational disassembly and are not SOPA compatible. The photographs

<sup>1</sup><http://wwideas.com/2015/11/the-most-impressive-wood-joints/>

<sup>2</sup><https://winterdienst.info/50-digital-wood-joints-by-jochen-gros/>

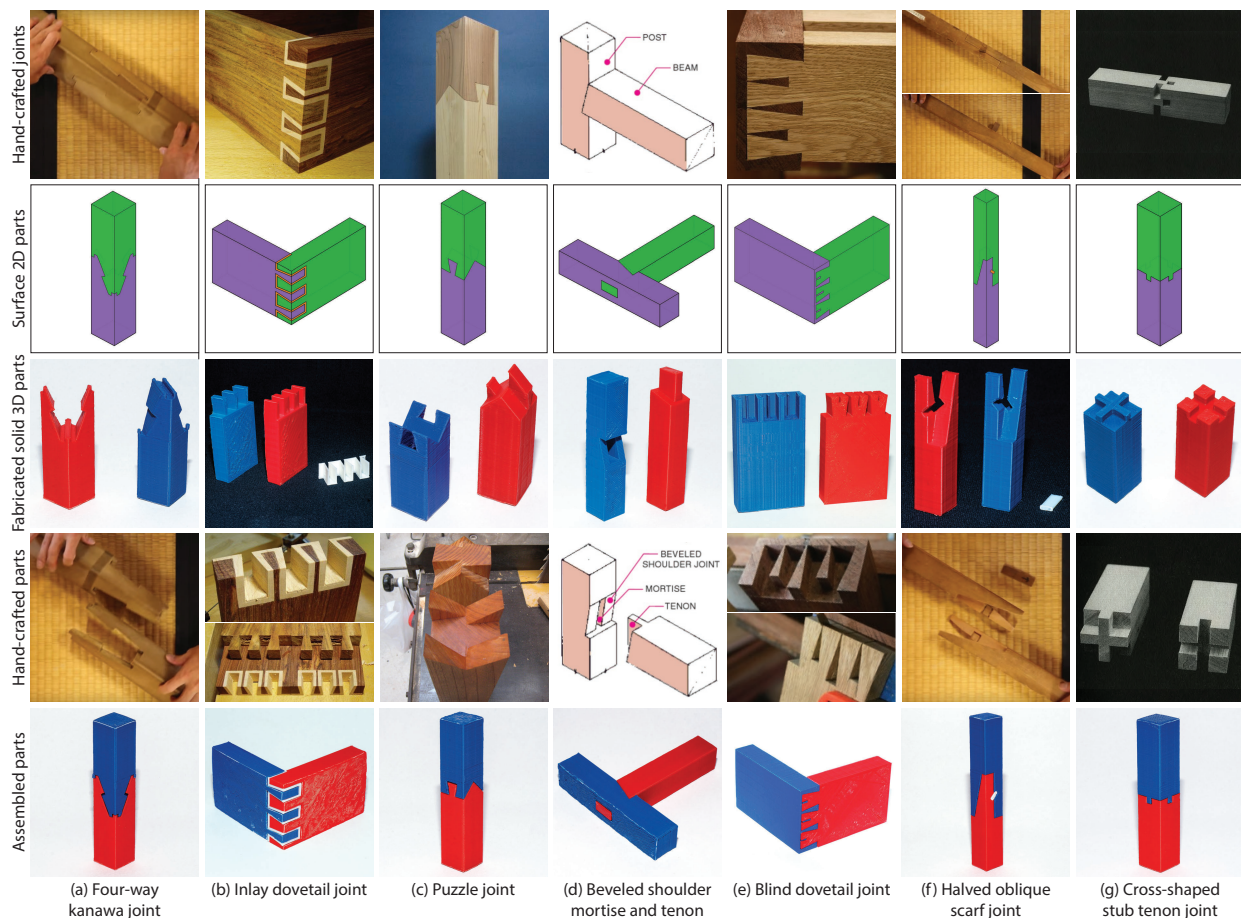


Figure 3.11: A representative sample of joints designed and fabricated using our interactive tool. Some are based on traditional woodworking joints (b, e, g) and others on Japanese joinery (a, c, d, f). For all of these designs we drew the surface 2D parts using a photo of the assembled hand-crafted joint (top row) as a guide. The resulting solid 3D parts generated by our tool (3rd row) is very similar to the hand-crafted part geometry (4th row) suggesting that the internal geometry of such joints is inherently complex. Please zoom in to see the geometric details. Appendix A contains 100 decorative joints we created using our tool.

often served as inspiration and we adapted them to test different appearances. For example, we varied the thickness of the inlay in the inlay dovetail, or modified the shapes of the arrows. The 3D geometry generated by our tool was often surprisingly complex despite relatively simple surface 2D parts. Yet, for the joints inspired by traditional woodworking and Japanese joinery, our tool produced similar, if not identical, 3D geometry suggesting that the complexity is inherent in certain joint designs.

We designed the complete furniture models (Figure 3.1 and 3.12) to contain a variety of different joint types, such as the following:

**Mortise-and-tenon joints:** joining purple Duffy table-top to legs; Duffy table legs to



Figure 3.12: Complete furniture assemblies we designed using our interactive tool.

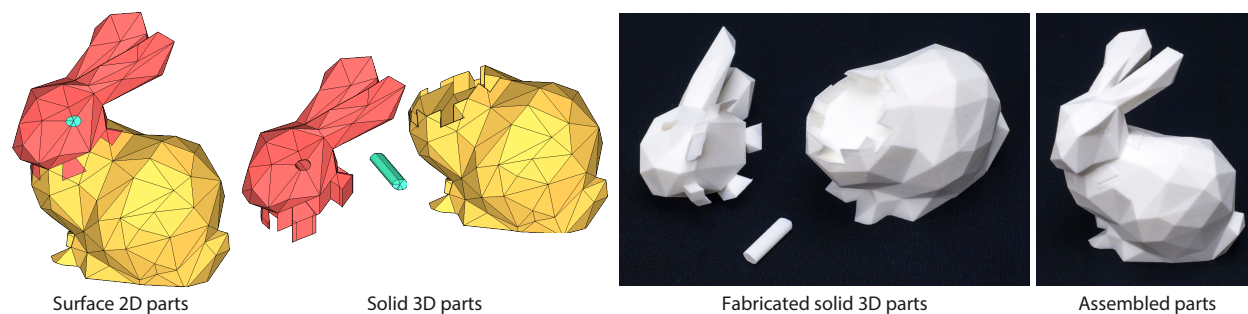


Figure 3.13: Our tool can generate decorative joints for free-form surfaces like the Stanford bunny.

yellow and light purple leg braces, sides of Nightstand to green top.

**Straight and curved finger joints:** joining sides of Nightstand to back; green side of Bench to top, red and green legs of Arch chair.

**Lap and half-lap joints:** joining light purple top seat-back to side frames of Branca chair; dark purple and orange seat parts of Branca chair to sides.

**Tongue-and-groove joints:** joining left/right ends of Duffy table to purple table-top.

**Other custom joint designs:** tree-shaped legs on Leaf table and Tree-side table; multi-circular joints at corners of Bookshelf; curved variants of puzzle joint between red side and top of Bench and between yellow top of seat-back and sides frames of Arch chair; hooked joints on Coffee table-top.

We adapted several of these designs from Web photographs of finished hand-crafted furniture.

As we iterated the joint designs, we regularly generated SOPA incompatible surface 2D parts and used the collision feedback from our interface (Section 3.3.3) to modify the parts — often directly using one of the solutions provided by our tool. The furniture examples also contain enough parts that our stability analysis often identified places where we forgot to add joints to prevent sliding between parts. In such cases, we added joints in the regions suggested by our tool to eliminate such part instabilities. Figure 3.10 shows an early design for the Duffy table in which we identified instabilities between the table-top and legs as well as the legs and the leg braces. Despite the complexity of some of the resulting part geometries, all of the resulting furniture models are assemblable and stable.

Our tool can also generate decorative joints for more free-form geometric surfaces like the Stanford bunny (Figure 3.13). Together these examples demonstrate that our tool is flexible enough to explore a wide space of decorative joint designs.

Our implementation takes between 3–15 minutes to construct the solid 3D parts for the joint models. Most of the furniture assemblies take between 30 minutes (e.g., the 8-part Branca chair) and 60 minutes (e.g., the 7-part Nightstand). Two assemblies take much longer (about 24 hours for the 13-part Bookshelf and the 3-part Bench). In all cases, computing a SOPA compatible assembly sequence for the surface 2D parts takes 1–4 minutes. If there are collisions, they are presented to the user at this point. The remainder of the time is spent





Figure 3.14: We commissioned a woodworker to fabricate a version of the Branca chair at full size using the solid 3D parts generated by our tool. (Coffee cup was added for scale.)

constructing the solid 3D parts. Our running time is dominated by the performance of mesh-based CSG. We use `libigl` [108], a performant and freely available CSG implementation. The running time depends primarily on the tessellation of the surface 2D parts—when the edges of these parts are curved, they require finer tessellation, which in turn requires more work when applying the sweep, union, and subtraction operations of our algorithm. The Bench and the Bookshelf assemblies contain intricately curved parts which require very high tessellation. Since our algorithm considers pairwise interactions between parts, its running time depends quadratically on the number of parts.

**3D Fabrication.** We used a PrintrBot and MakerBot with PLA or ABS plastic material to fabricate physical parts for many of our joints and furniture assemblies. Due to the resolution of these printers and expansion of 3D printed parts, we have observed that the fabricated parts often do not fit each other when their geometries are directly supplied to the 3D printer. To ensure they fit and can be assembled, we assume uniform material expansion and offset the 3D part geometries before supplying them to the 3D printer. We translate each vertex of a part by  $\epsilon$  in the average of the normal directions of its incident faces. For both the PrintrBot and MakerBot, we have found empirically that  $\epsilon = 0.25$  mm ensures that the fabricated parts can be assembled together and that their fits are tight. As a proof of concept, we commissioned a woodworker to fabricate the Branca chair at full size using the solid 3D parts generated by our tool (Figure 3.14).

**User Feedback.** We have shown our tool and results to seven hobbyist (amateur) woodworkers and two professional furniture designers from local studios. All of them thought that our tools made it easier to design and generate novel, decorative furniture joints. They especially appreciated the creative freedom with which they could focus on sketching the surface aesthetics of joints while letting our tool determine their complex internal geometries. A hobbyist woodworker said *“I avoided complex 3D geometries in my joint designs”* in the past, and explained that with our tool he could focus on designing more decorative joints. They liked that our tool gave suggestions for fixing SOPA incompatibilities and for correcting unstable part configurations. One of the professional furniture designers noted *“This is a super cool system! I could see people creating some amazing pieces using this technique. It would be really powerful in the hands of an artist.”*

We also asked three novice designers to generate joints for the Bench furniture model using our tool. Some of them drew surface 2D parts based on inspirational images of joints found online while others incorporated free-form shapes, such as letter-forms and icons. They generally did not think about constraints on assemblability of parts and were focused on surface appearance. In a few cases our system could not find a SOPA sequence for the surface 2D parts, and instead suggested how the parts might be redesigned to produce a SOPA sequence. The users went back and iteratively redesigned the joints based on the suggestions; most often using one of the automatically suggested redesigns. In a few cases the redesign involved going back into SketchUp to re-sketch the surface 2D parts. While using our tool, the novice designers did raise concerns over the rate at which our tool constructed the solid 3D parts and believed improving the construction time would further facilitate the joinery design process. However, they still believed our tool was much easier to use than directly creating the 3D geometry in SketchUp and were surprised at the complexity of 3D geometry that was necessary to produce conforming, assemblable joints.

## 3.6 Conclusion

We have described an interactive joinery design tool. Users draw the visual appearance of joints on the surface of an input model, and our tool automatically generates an assemblable set of solid 3D parts that conform to the drawings. Our tool reduces by a dimension the amount of user effort needed to create assemblable solid 3D parts. In addition, we provide a stability analysis technique for analyzing the resulting assembly of solid 3D parts to identify part instabilities. When our tool finds unstable parts that can move with respect to one another, it visualizes the instability and suggests where the user might draw additional surface regions to fix the problem. Thus, our approach can save the user from unnecessary costly and lengthy fabrication. As personalized fabrication devices become more and more ubiquitous, we believe design tools such as ours are needed to make fabrication more accessible.

# Chapter 4

## Designing Flexible Kerf Patterns

### 4.1 Introduction

Classic aesthetic wood furniture often features beautifully curved parts that are handcrafted using one of the three traditional wood bending methods [75]. The *steaming* method softens and bends solid timber with heat and moisture, while the *lamination* method bends thinly-sliced lumber strips before gluing them together [76, 4]. Both methods are labor-intensive and require clamping the bent wood pieces overnight to set their shapes. In comparison, the *kerfing* method is considerably less time-consuming and involves only cutting deep, evenly-spaced *kerfs* (i.e., notches) into wood boards. These kerfs weaken sections of the wood boards, enabling them to bend significantly around the kerf cutting direction [46, 89].

Inspired by the traditional kerfing method, Porterfield [67] and other designers [6, 9, 48, 26, 22] have created various cutout patterns that similarly allow rigid boards to bend flexibly around one axis after materials are removed from them based on the patterns (Figure 4.1a). But unlike the kerfing method, which requires partial cuts into the boards and must often be carefully executed with sawing machines, the cutout patterns only need through cuts and can be easily fabricated with laser cutters. Moreover, while the kerfing method creates repetitive notches usually covered up by veneers, the cutout patterns are often ornamental and provide an appealing way to impart styles to woodworking and craft projects. These advantages have made cutout patterns a popular technique to bend rigid boards (e.g., patterns designed by Porterfield [67] have been viewed over 400,000 times online). Using kerfs to mean cutouts as designers have widely adopted, we will refer to cutout patterns as kerf patterns.

Most commonly, kerf patterns are periodic as they are easier to create and adapt to boards of different sizes than aperiodic patterns. Various sophisticated tilings, such as Escher-inspired tiles [36], provide bases for periodicity. But, through an in-person interview with Porterfield [66], we have learned that designers generally prefer creating periodic kerf patterns on a simple grid. Far from limiting creativity, the grid in fact supports a diverse collection of decorative pattern designs (Figure 4.1 and 4.9-4.11). However, even with the simple grid as the periodicity basis, designing kerf patterns that are flexible is still challenging because

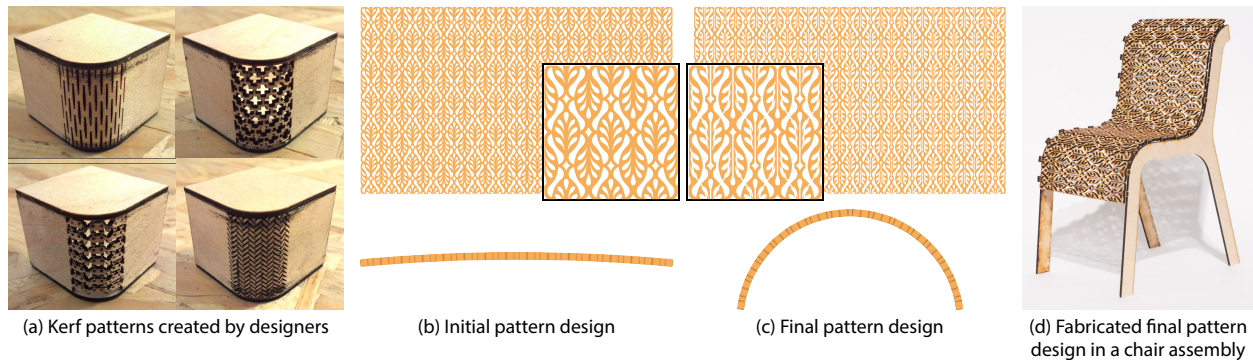


Figure 4.1: Kerf patterns allow rigid boards to bend flexibly while imparting them with unique styles (a). Starting with a unit pattern that is tileable on a grid, our design tool simulates the bending behavior of the kerf pattern (i.e., tiled unit patterns) at interactive rates (b). If the initial pattern is less flexible than desired, users can apply our design principles and effectively modify its geometry to improve its flexibility (c). (We note that the initial and final pattern designs undergo the same amount of bending force.) Users can then incorporate the final, flexible pattern into their furniture project (d). Please zoom in to see the geometric details of the kerf pattern designs.

their geometric designs are not obviously related to their flexibility levels. As a result, designers typically create many patterns blindly in hopes that some will be flexible and rely on physically fabricating the patterns to assess their flexibility levels, which can be costly in time and material [6, 48]. Furthermore, when pattern designs are less flexible than desired, modifying their geometries for improved flexibilities is a tedious trial and error process since to our knowledge there exists no design principles on what geometric features contribute to flexible kerf patterns. Designers often iterate only a few times between fabricating the patterns to check if they are flexible and redesigning their geometries based on guesswork before giving up and opting to use known flexible patterns instead [66]. Consequently, most flexible kerf patterns available online are simply variations of a few known designs [49, 16].

We present a design tool that allows users to interactively preview the flexibility levels of their grid-based kerf pattern designs without physical fabrication; and we establish a set of design principles for flexible kerf patterns that users can effectively apply to their pattern geometries for improved flexibilities. To simulate the bending behaviors of kerf patterns at interactive rates, we exploit their periodicities and propose a fast, *periodic boundary condition*-constrained simulation based on the finite element method. Specifically, given a kerf pattern, we simulate only its constituent unit pattern and enforce the periodic boundary conditions that the left and top boundary faces of the bent unit pattern must respectively match its right and bottom boundary faces via rigid-body transformations. In order to formulate the design principles, we create 75 unit kerf patterns with a great diversity of geometric designs and flexibility levels. By quantitatively analyzing the relationship between the two, we show that the flexibility level of a kerf pattern is well correlated with its *geodesic*, i.e., the shortest path between the leftmost and rightmost sides of the pattern.

Our geodesic-based design principles allow users to make informed editing decisions through visually reasoning about how their modifications to the pattern geometries affect the flexibility levels. By eliminating physical fabrication and redesign guesswork from the kerf pattern design process, our tool and design principles let users focus on the visual designs of their patterns and give them the creative freedom to prototype new decorative flexible patterns.

## 4.2 Overview

Creating a flexible grid-based kerf pattern requires iterating between designing the geometry, checking the flexibility level of the design, and modifying the geometry if the design does not meet the desired flexibility level [66, 26]. Users first design in a vector graphics editor their initial 2D unit pattern such that it forms a single, connected kerf pattern when tiled on a grid. Our design tool then extrudes the unit pattern along the  $z$ -axis and simulates the bending behavior of the 3D kerf pattern (i.e., tiled 3D unit patterns) using a periodic boundary condition-constrained finite element method (Section 4.3). When the kerf pattern is less flexible than desired, users can follow our geodesic-based design principles to effectively modify the geometry of their unit pattern (Section 4.4) and use our tool to verify that the flexibility level of the redesigned pattern has improved. Since kerf patterns can be designed to bend flexibly around any single direction, our design principles support creating patterns that are flexible around the  $y$ -axis.

## 4.3 Simulations of Periodic Kerf Patterns

Linear elastostatics [14] and Finite Element Method (FEM) with linear tetrahedron elements [17] are commonly used in computer graphics to simulate solid objects with small deformations at steady states [62, 61, 59]. For objects with large deformations, however, linear elastostatics and FEM distort them unrealistically and researchers have developed a number of advanced simulation methods to address this problem [58, 55]. While a flexible grid-based kerf pattern can bend significantly, its large global deformation results from an accumulation of small local deformations of its periodically tiled unit patterns. Therefore, given a kerf pattern, we can simulate the small bending behavior of its constituent unit pattern using linear elastostatics and FEM (Section 4.3.1) and exploit the periodicity of the kerf pattern to fully capture the large bending behavior of the entire pattern (Section 4.3.2).

### 4.3.1 Simulations of Unit Kerf Patterns

In general, given a tetrahedral mesh with  $N$  vertices, we can simulate its deformation due to external forces,  $\mathbf{f}$ , by solving the linear elastostatic equation,

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \tag{4.1}$$

for vertex displacements,  $\mathbf{u} \in \mathbb{R}^{3N}$ , where  $\mathbf{K}$  is the stiffness matrix. However, we cannot directly apply (4.1) to simulate the bending behaviors of unit kerf patterns for two reasons.

First, (4.1) does not have a unique solution because  $\mathbf{K}$  is known to be rank deficient. Its rank is always  $3N - 6$  due to a rank-6 null space spanned by translation and linearized rotation of the unconstrained rigid body. To obtain a unique displacement, one common method anchors a number of vertices of the tetrahedral mesh [59, 55], but these anchors introduce extraneous constraints and decrease the simulation accuracy.

Second, because a unit pattern is periodically tiled on a grid to form a kerf pattern, a unit pattern does not bend freely on its own and its boundary faces are constrained by its neighboring unit patterns to the left and right (along the  $x$ -axis) and top and bottom (along the  $y$ -axis). Therefore, we must impose additional constraints on the boundary faces of the unit pattern to respect the periodicity of the kerf pattern.

The sparse and direct least-squares solver proposed by Zheng et al. [107] computes a unique solution to (4.1) without enforcing extraneous constraints by projecting out the null space of  $\mathbf{K}$ . We closely follow their approach and incorporate additional constraints into their solver to properly simulate the bending behaviors of unit kerf patterns. We briefly describe their method here and detail our modification to their solver in Section 4.3.2.

**Sparse and Direct Least-Squares Solver.** Zheng et al. first use an orthogonal projection  $\mathcal{P}$  to remove the linearized rigid-body motion from  $\mathbf{f}$  to ensure the system is compatible, i.e.,  $\mathcal{P}\mathbf{f} \in \text{range}(\mathbf{K})$ . They then devise a  $3N \times (3N - 6)$  sparse orthogonal matrix  $\mathbf{V}$ , which serves as a vertex displacement basis to eliminate translation and linearized rotation degrees of freedom from  $\mathbf{u}$ . Finally, they substitute  $\mathbf{u} = \mathbf{V}\mathbf{r}$  and premultiply (4.1) by  $\mathbf{V}^T$  to arrive at a sparse, symmetric, and *full-rank*  $(3N - 6) \times (3N - 6)$  system:

$$(\mathbf{V}^T\mathbf{K}\mathbf{V})\mathbf{r} = \mathbf{V}^T\mathcal{P}\mathbf{f}. \quad (4.2)$$

This system can be solved directly using Cholesky factorization for the unique displacement,  $\mathbf{u} = \mathcal{P}\mathbf{V}\mathbf{r}$ , where  $\mathcal{P}$  is used to project out the particular linearized rigid-body motion chosen by  $\mathbf{V}$ . We refer readers to their paper for details on  $\mathcal{P}$  and  $\mathbf{V}$ .

### 4.3.2 Periodic Boundary Conditions

To correctly simulate the constituent unit pattern of a kerf pattern, we modify the vertex displacement basis  $\mathbf{V}$  in (4.2) to additionally impose constraints on the boundary faces of the unit pattern in order to enforce the periodicity of the kerf pattern. We refer to these additional constraints as *periodic boundary conditions* and we formulate them based on the following observation.

When two neighboring unit patterns along the  $x$ -axis experience bending, the right boundary face of the left pattern must match the left boundary face of the right pattern (Figure 4.2a), which in turn must match the left boundary face of the left pattern via a rigid-body transformation because unit patterns are identical and bend exactly alike. This

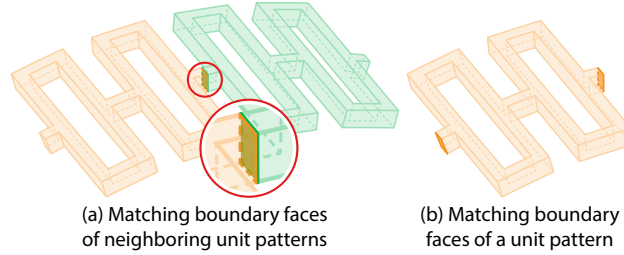


Figure 4.2: Given two neighboring unit patterns along the  $x$ -axis, the right boundary face of the left pattern (highlighted by the orange line) and the left boundary face of the right pattern (highlighted by the green line) must match when the patterns undergo bending (a). Since unit patterns are identical and bend exactly alike, the left boundary faces of both the left and right patterns must match via a rigid-body transformation. Therefore, when a unit pattern experiences bending, its left and right boundary faces must also match via a rigid-body transformation (b).

observation reveals that, given a unit pattern under bending force, all corresponding vertices on its left and right boundary faces,  $\mathbf{p}^l$  and  $\mathbf{p}^r$ , must match via a *single* rigid-body transformation (Figure 4.2b). Since the deformation of a unit pattern is generally small, we approximate this transformation with a translation  $\mathbf{t}$  and a linearized rotation  $\mathbf{R}$ .

We can therefore impose periodic boundary conditions on the left and right boundary faces of a unit pattern by requiring that their corresponding vertices satisfy  $\mathbf{R}(\mathbf{p}^l + \mathbf{u}^l) + \mathbf{t} = \mathbf{p}^r + \mathbf{u}^r$ , where  $\mathbf{u}^l$  and  $\mathbf{u}^r$  are the respective vertex displacements. This equation takes the following matrix form:

$$\begin{bmatrix} 1 & -r_z & r_y \\ r_z & 1 & -r_x \\ -r_y & r_x & 1 \end{bmatrix} \begin{bmatrix} p_x^l \\ p_y^l \\ p_z^l \end{bmatrix} + \begin{bmatrix} 1 & -r_z & r_y \\ r_z & 1 & -r_x \\ -r_y & r_x & 1 \end{bmatrix} \begin{bmatrix} u_x^l \\ u_y^l \\ u_z^l \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} p_x^r \\ p_y^r \\ p_z^r \end{bmatrix} + \begin{bmatrix} u_x^r \\ u_y^r \\ u_z^r \end{bmatrix}.$$

With some simplifications, we can equivalently express the equation as between the displacements of corresponding vertices:

$$\begin{bmatrix} u_x^l \\ u_y^l \\ u_z^l \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} 0 & p_z^l & -p_y^l \\ -p_z^l & 0 & p_x^l \\ p_y^l & -p_x^l & 0 \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} u_x^r \\ u_y^r \\ u_z^r \end{bmatrix}. \quad (4.3)$$

We devise our modified vertex displacement basis  $\mathbf{V}_{\text{PBC}}$  based on (4.3) by replacing the deformation degrees of freedom of all right vertices in  $\mathbf{V}$  (i.e., vertices on the right boundary face) with a single linearized rigid-body transformation. We similarly impose periodic boundary conditions on the top and bottom boundary faces by replacing the degrees of freedom of all bottom vertices in  $\mathbf{V}$  with a different rigid-body transformation.

Our modified vertex displacement basis  $\mathbf{V}_{\text{PBC}}$  has  $3N$  rows and  $3(N - N^r - N^b) + 6$  columns corresponding to the periodic boundary conditions-constrained deformation degrees





relationship between the geometric design and flexibility level of a kerf pattern is usually not obvious and, to our knowledge, there exist no design principles that directly relate the two.

Without such principles, users are unable to visually determine whether their modifications to the geometry allow the pattern to be more flexible; and as a result, they cannot make informed editing decisions to increase the flexibility level of their pattern based on visual inspections of its geometry. Instead, users have to edit their pattern by trial and error and rely on running the simulation after every geometry modification to check if the resulting pattern becomes more flexible—an ineffective and time-consuming process.

In order to facilitate a productive editing process, we first identify geometric features of a kerf pattern that are directly related to its flexibility level. We then formulate a set of design principles, based on the identified geometric features, that users can effectively apply to their geometric pattern design to improve the flexibility without resorting to simulations.

Through bending a variety of physically fabricated kerf patterns, we have empirically found that the shortest path across a pattern is related to its flexibility level. We define the shortest path and flexibility level of a kerf pattern in Section 4.4.1 and we use these definitions to detail our findings in Section 4.4.2. In order to verify our findings, we create 75 unit kerf patterns and quantitatively analyze the relationship between their shortest paths and simulated flexibility levels. The analysis shows that while the two attributes are related, the shortest path alone is not sufficiently correlated with the flexibility level. However, by further examining the unit kerf patterns, we have discovered three geometric features associated with the shortest path that together correlate well with the flexibility level of a pattern. In Section 4.4.3, we describe these geometric features and establish a set of principles on designing flexible kerf patterns.

#### 4.4.1 Geodesics and Curvatures

To help present our empirical findings from physical kerf patterns, we provide definitions for the shortest path across a kerf pattern and its flexibility level.

Given two points on a surface, the shortest path between them that lies on the surface is called a *geodesic*. Adapting this term to kerf patterns, we refer to the shortest path between any pair of leftmost and rightmost vertices of a pattern as a *local geodesic* of the pattern (Figure 4.3a); and we refer to the shortest local geodesic as the *global geodesic* or simply the *geodesic* of the pattern (Figure 4.3b). Based on the new terms, we define the shortest path across a kerf pattern as the geodesic of the pattern. Since we represent a pattern as a triangle mesh, its geodesic is a sequence of line segments and can be computed exactly using the algorithm proposed by Mitchell et al. [56].

Because we design kerf patterns to bend around the  $y$ -axis, we define the flexibility level of a kerf pattern as the *curvature* formed by the pattern along the  $x$ -axis when bent with a set force, following standard engineering practices [38]. To compute this curvature, we apply a fixed amount of bending force to the left and right sides of the pattern and simulate its bending behavior with our periodic boundary condition-constrained FEM. We then approximate the top surface of the bent pattern with a least-squares paraboloid, expressed

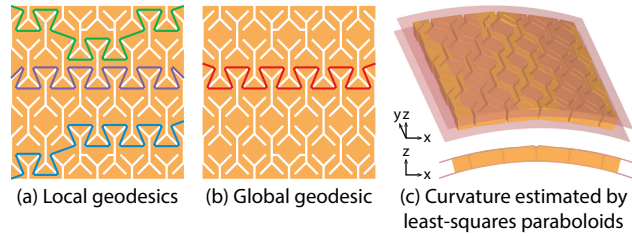


Figure 4.3: Given a kerf pattern, its *local geodesic* is the shortest path between any pair of leftmost and rightmost vertices of the pattern (each color—green, purple, and blue—denotes a different local geodesic) (a). The *global geodesic* or *geodesic* of a pattern is its shortest local geodesic (b). To compute the *curvature* formed by a kerf pattern along the  $x$ -axis, we approximate the top and bottom surfaces of the bent pattern with least-squares paraboloids and estimate the curvature as the average of principal curvatures of the two paraboloids at their centers along the  $x$ -axis (c).

as  $z = ax^2 + by^2$ , and estimate the curvature of the top surface as  $2a$ , i.e., the principal curvature of the paraboloid at its center along the  $x$ -axis. We similarly evaluate the curvature of the bottom surface and take the average of the two curvatures to represent the curvature formed by the pattern (Figure 4.3c).

#### 4.4.2 Observations and Analyses

Based on the bending behaviors of physical kerf patterns, we have made two observations on the relationship between the geodesic of a pattern and the curvature formed by the pattern.

First, we have observed that the *geodesic region*, i.e., the region of a kerf pattern along its geodesic (Figure 4.4a), often bends the least and forms the lowest curvature compared to other regions across the pattern. Since a pattern can only bend as much as its least bendable region, the geodesic region determines the curvature formed by the pattern.

Second, we have observed that when a kerf pattern has a longer geodesic, its geodesic region and consequently the pattern itself generally form a higher curvature while under the same bending force. Intuitively, when such a pattern bends, some portions of its geodesic region twist as opposed to bend (e.g., the vertical bars in Figure 4.4c). These torsions (i.e., twistings) cumulatively allow the geodesic region and thus the pattern to form a higher curvature than what the elasticity of the fabrication material normally permits (Figure 4.4b). On the other hand, when a pattern has a shorter geodesic (e.g., an H-shaped pattern), most portions of its geodesic region lie straight across the pattern and cannot twist; and the pattern forms a lower curvature as a result.

Together, these two observations reveal that we can determine the curvature formed by a kerf pattern using the curvature of the geodesic region, which we can in turn estimate by the length of the geodesic. To verify our observations and quantitatively analyze the direct relationship between the pattern curvature and the geodesic length, we have created 75 unit kerf patterns with a wide range of aesthetic geometric designs and sized them to have the same dimension so that variations in pattern curvatures and geodesic lengths arise only due

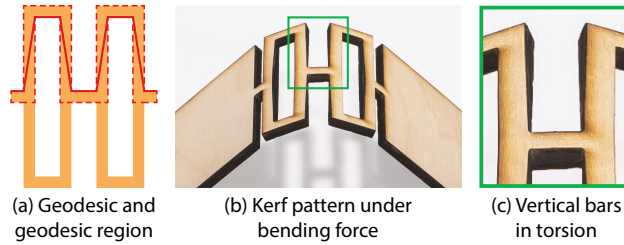


Figure 4.4: The *geodesic region* of a kerf pattern is the region of the pattern along its geodesic (highlighted by the red dotted line) (a). When a pattern with a long geodesic undergoes bending (b), portions of the geodesic region twist as opposed to bend (c). These torsions collectively enable the geodesic region, as well as the pattern, to bend beyond the elastic limit of the fabrication material (e.g., this example pattern is fabricated with rigid birch plywood).

to differences in pattern designs. Figure 4.6 showcases a representative sample of the unit kerf patterns, along with their curvatures and geodesics, and we provide all 75 unit kerf patterns in Appendix B.

To conduct the quantitative analysis, we plot the curvatures formed by the unit kerf patterns against their geodesic lengths. The scatter plot (Figure 4.7a) shows that, very broadly, patterns with longer geodesics have higher curvatures than those with shorter geodesics. However, as the plot illustrates, pattern curvatures and geodesic lengths are inadequately correlated and they have a low coefficient of determination  $R^2$  of 0.514 when fitted with a least-squares linear regression. In particular, the plot contains numerous data points whose patterns share similar geodesic lengths but form measurably different curvatures (e.g., the highlighted pairs of pink and blue data points). Upon close examinations of these patterns, we have discovered that the curvature of the geodesic region, which determines the pattern curvature based on our observations, is not fully captured by the geodesic length alone and is, in fact, additionally dependent on (1) the geodesic widths, (2) the geodesic orientations, and (3) the number of distinct local geodesics of the pattern. Therefore, we devise a geodesic measure that incorporates these additional geometric features in order to better estimate the curvature of the geodesic region and consequently improve the correlation with the pattern curvature.

### 4.4.3 Geodesic Measures and Design Principles

To facilitate devising the geodesic measure, we give a geometrically precise definition for the geodesic region of a kerf pattern and we use this definition to examine how the curvature of the geodesic region is formed.

Recall from Section 4.4.1 that the geodesic of a kerf pattern is a sequence of line segments. We analogously define the geodesic region as a sequence of *bar segments*, where each bar segment is the extrusion of the pattern area covered by the widths of the corresponding geodesic line segment. More precisely, we define the width at any point along a geodesic line segment as the perpendicular line segment that intersects the pattern (Figure 4.5a).

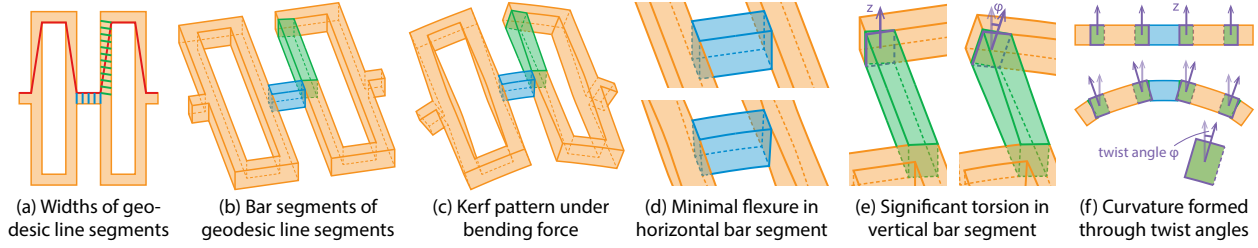


Figure 4.5: Given a line segment of the geodesic, its width at any point is the perpendicular line segment that intersects the kerf pattern (a) and its corresponding *bar segment* is the extrusion of the area covered its widths at all points (b). We define the geodesic region of a pattern as a sequence of bar segments corresponding to the line segments of the geodesic. When a pattern experiences bending (c), a more horizontally oriented bar segment of the geodesic region can only bend minimally due to the rigidity of the material (d). On the other hand, a more vertically oriented bar segment can twist significantly in spite of the rigid material and we measure its twist angle (labeled  $\varphi$ ) as the rotation degree of its cross section (the area highlighted by the purple line) (e). Accumulatively, the twist angles of the bar segments form the majority of the curvature of the geodesic region (f).

For each geodesic line segment, its widths at all points cover an area of the pattern and its corresponding bar segment is then the extrusion of this area along the  $z$ -axis to the thickness of the pattern (Figure 4.5b).

When a kerf pattern undergoes bending, bar segments of the geodesic region that are more horizontally oriented will experience more flexures (i.e., bendings), while bar segments that are more vertically oriented will experience more torsions. Since the pattern is made of rigid material, bar segments under flexures can only bend negligibly and therefore contribute little to the curvature of the geodesic region (Figure 4.5d). In contrast, bar segments under torsions can twist significantly despite the rigidity of the material (Figure 4.5e). The accumulated twist angles of these bar segments, measured as rotation degrees of their cross sections, contribute to form the majority of the curvature of the geodesic region (Figure 4.5f).

We thus devise a geodesic measure that estimates the curvature of the geodesic region via the total twist angle of its constituent bar segments. We build the geodesic measure incrementally by incorporating, one at a time, the group of geometric features identified in the quantitative analysis (Section 4.4.2) and we show that this geodesic measure is well correlated with the pattern curvature.

**(1) Geodesic Widths.** Given a kerf pattern, we first consider the widths of the bar segments in computing their twist angles and estimating the curvature of the geodesic region. We define that a bar segment and its corresponding geodesic line segment share the same length and widths along the length. If such widths are constant and the bar segment has a uniform rectangular cross section, we can analytically compute its twist angle  $\varphi$  as

$$\varphi = \frac{TL}{Gab^3}, \quad (4.5)$$

where  $T$  is the applied torque,  $G$  is the modulus of rigidity of the material,  $L$  is the length of the bar segment,  $a$  is the larger of its width or thickness, and  $b$  is the smaller of the two values [92, 23, 7]. But since the widths along most bar segments vary, we approximate the twist angle of such a bar segment as

$$\varphi \approx \frac{TL}{Gn} \sum_{\text{sample } i} \frac{1}{\max(t, w_i)\min(t, w_i)^3}, \quad (4.6)$$

where  $n$  is the number of points evenly sampled along the length of the bar segment,  $t$  is the thickness, and  $w_i$  is the width at each sample point. To simplify the formulation of the geodesic measure, we assume all kerf patterns are made of the same material. We additionally assume that for any pattern, the geodesic region is subject to the entire bending force, its bar segments undergo identical torques, and their twist angles contribute fully to the curvature of the geodesic region. (We will address these additional assumptions when considering the next two geometric features.) Based on these assumptions, we devise the geodesic measure  $M_1$  to estimate the curvature of the geodesic region by approximating the total twist angle of the bar segments:

$$M_1 = \sum_{\text{bar } j} \frac{L_j}{n_j} \sum_{\text{sample } i} \frac{1}{\max(t, w_i)\min(t, w_i)^3}. \quad (4.7)$$

**(2) Geodesic Orientations.** We now incorporate the orientations of the bar segments into the geodesic measure to improve its curvature estimation of the geodesic region. We define the orientation of a bar segment as the angle between the  $x$ -axis and its corresponding geodesic line segment and we have observed that this orientation affects the curvature of the geodesic region in two different ways.

First, the orientation of a bar segment affects the torque applied to the bar segment, which then affects its twist angle that contributes to the curvature of the geodesic region. Specifically, because we bend a kerf pattern around the  $y$ -axis, the bending force applies the largest torque to a vertical bar segment and such a bar segment undergoes the greatest torsion (Figure 4.5e). But as the bar segment becomes more horizontally oriented, the applied torque from the same bending force gets smaller and eventually reduces to zero when the bar segment is horizontal and does not twist at all (Figure 4.5d). Therefore, given a bar segment with the orientation  $\theta$ , we approximate its applied torque as  $T\sin\theta$  and its twist angle as  $\varphi\sin\theta$ .

Second, the orientation of a bar segment affects the contribution of its twist angle to the curvature of the geodesic region, which we have defined to be along the  $x$ -axis (Section 4.4.1). When a bar segment is vertical, its cross section and twist angle lie entirely on the  $xz$ -plane, so the twist angle contributes fully to the curvature along the  $x$ -axis (Figure 4.5e,f). (Recall from earlier in Section 4.4.3 that we measure the twist angle of a bar segment as the rotation degree of its cross section.) However, as the bar segment

becomes more horizontally oriented and its cross section rotates further away from the  $xz$ -plane, less of its twist angle contributes to the curvature along the  $x$ -axis until eventually none of the twist angle does when the bar segment is horizontal and its cross section lies entirely on the  $yz$ -plane. Therefore, given a bar segment with the orientation  $\theta$ , we approximate the proportion of its twist angle that contributes to the curvature of the geodesic region as  $\sin \theta$ .

Taking into account both effects of the orientations of the bar segments, we devise the geodesic measure  $M_2$  to better estimate the curvature of the geodesic region:

$$M_2 = \sum_{\text{bar } j} \frac{L_j}{n_j} \sum_{\text{sample } i} \frac{\sin^2 \theta}{\max(t, w_i) \min(t, w_i)^3}. \quad (4.8)$$

**(3) Number of Distinct Local Geodesics.** For simplicity, the geodesic measure  $M_1$  and  $M_2$  assume that the bending force applied to a kerf pattern is exerted entirely on its geodesic region. But the bending force is in fact distributed over the whole pattern, particularly regions along its local geodesics that also experience flexures and torsions when the pattern undergoes bending. (Recall from Section 4.4.1 that a local geodesic is the shortest path between any pair of leftmost and rightmost vertices of a pattern.) We thus approximate that the bending force and the induced torque exerted on the geodesic region are inversely proportional to the number of distinct (minimally overlapping) local geodesics of the pattern. We define a local geodesic as distinct if it is the shortest among all local geodesics whose regions overlap its region by over 50%. Consider the Pinwheel and Coin pattern designs in Figure 4.6. Both patterns are subject to the same bending force and their geodesic regions share very similar geometries. But because Coin has three times as many distinct local geodesics as Pinwheel, Coin has only a third as much bending force and induced torque exerted on its geodesic region and Coin consequently forms a curvature a third as large as Pinwheel. Incorporating the number of distinct local geodesics  $N$ , we devise the final geodesic measure  $M_3$  to give the best estimate of the curvature of the geodesic region:

$$M_3 = \frac{1}{N} \sum_{\text{bar } j} \frac{L_j}{n_j} \sum_{\text{sample } i} \frac{\sin^2 \theta}{\max(t, w_i) \min(t, w_i)^3}. \quad (4.9)$$

To evaluate how well the geodesic measures correlate with the pattern curvature, we follow the quantitative analysis in Section 4.4.2 and similarly plot the curvatures formed by the 75 unit kerf patterns against each of the three geodesic measures. The scatter plots (Figure 4.7b,c) show that, compared to the geodesic length, the geodesic measure  $M_1$  and  $M_2$  have only modestly better correlations with the pattern curvature. However, by incorporating all the identified geometric features, the geodesic measure  $M_3$  achieves a marked improvement over all other measures in correlating with the pattern curvature and reaches a high  $R^2$  of 0.902 when fitted with a least-squares linear regression (Figure 4.7d). Such a strong correlation demonstrates that we have successfully discovered a meaningful group of

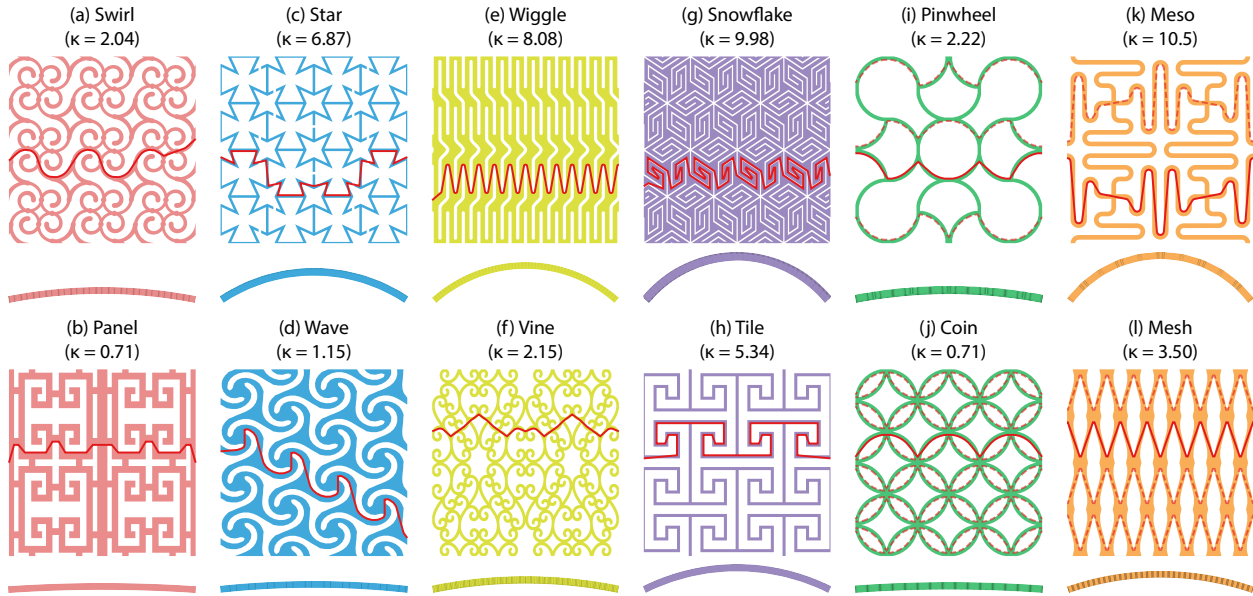


Figure 4.6: A representative sample of the 75 unit kerf patterns designed to help identify geometric features of a pattern that are directly related to its curvature. We have created these 75 patterns to have a great diversity of decorative geometric designs (a–l), a large span of geodesic lengths from short (b, d, f, j) to long (e, g, k, l), and a wide range of curvatures from inflexible (a, b, d, j) to flexible (c, e, g, k). From these patterns, we have observed that they additionally differ in three geometric features associated with geodesics (Section 4.4.3): geodesic widths (a and b, c and d), geodesic orientations (e and f, g and h), and the number of distinct local geodesics (i and j, k and l, whose distinct local geodesics, excluding the geodesic, are delineated with red dotted lines). We size all unit kerf patterns to have the same dimension ( $3.5 \times 3.5 \times 0.03$  cm) and assign them properties of birch plywood (Young’s modulus  $E = 8.0 \times 10^9$  N/m<sup>2</sup> and Poisson’s ratio  $\nu = 0.3$ ), a common (rigid) fabrication material for laser cutters and CNC machines. We apply 40 N of bending force to all these patterns and compute their curvatures (labeled under the pattern names) as outlined in Section 4.4.1. To accentuate the curvature differences between unit kerf patterns, we tile them on a  $5 \times 3$  grid, along the  $x$ - and  $y$ -axis respectively, and visualize the bending behaviors of the resulting patterns. Please zoom in to see the geometric details of the pattern designs. We provide all 75 unit kerf patterns in Appendix B.

geometric features that is directly related to the flexibility level of a kerf pattern. We therefore formulate a set of design principles for flexible kerf patterns based on these geometric features:

- (1) **Long Geodesics.** A kerf pattern with a longer (global) geodesic is more flexible. Note that this requires all local geodesics of the pattern to be longer as well.
- (2) **Narrow and Vertical Geodesics.** A kerf pattern also becomes more flexible when all local geodesics, including the (global) geodesic, are narrower and more vertically oriented.

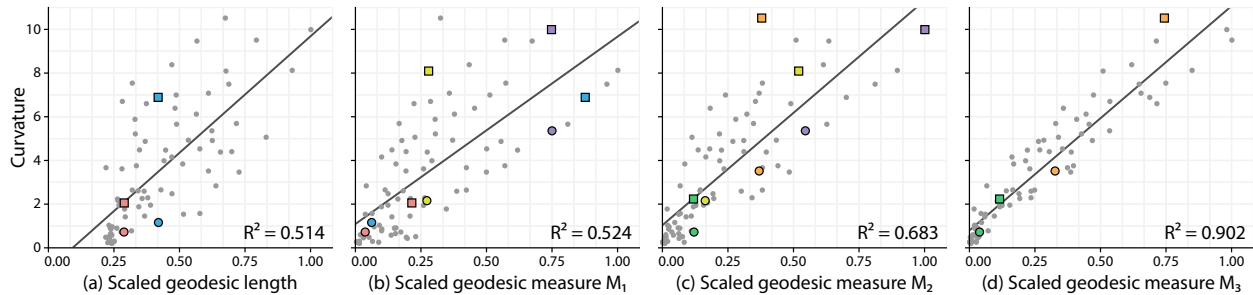


Figure 4.7: Correlations between curvatures formed by 75 unit kerf patterns (vertical axis) and their geodesic lengths (horizontal axis), as well as three geodesic measures. We scale the geodesic lengths and measures by their respective maximum values as the actual values are irrelevant to the correlation analysis. The colors and shapes of the highlighted data points correspond to the colors and row positions of the patterns in Figure 4.6 (squares and circles refer to the top and bottom rows, respectively). The analysis reveals that the geodesic length alone is insufficiently correlated with the pattern curvature (a), which depends on three additional geometric features of the geodesic. The geodesic measure  $M_1$  takes geodesic widths into consideration so that for patterns with comparable geodesic lengths (e.g., the Star and Wave patterns highlighted by the blue data points) (a), patterns with narrower geodesics receive greater values (b). The geodesic measure  $M_2$  additionally incorporates geodesic orientations and assigns larger values to patterns whose bar segments are more vertically oriented (e.g., the Wiggle pattern as compared to the Vine pattern) (c). Finally, the geodesic measure  $M_3$  further accounts for the number of distinct local geodesics and allot patterns with fewer distinct local geodesics proportionally higher values (e.g., the Pinwheel pattern relative to the Coin pattern) (d). By considering all three additional geometric features of the geodesic, the geodesic measure  $M_3$  achieves a strong correlation with the pattern curvature and they have a high  $R^2$  of 0.902 when fitted with a least-squares linear regression.

**(3) Small Number of Geodesics.** Finally, a kerf pattern that has a smaller number of distinct local geodesics is more flexible.

Using these design principles, users can make informed and effective editing decisions to improve the flexibility of their kerf pattern design based on visual inspections of its geometry alone, without having to tediously rely on the simulation to check whether their geometry modifications yield a more flexible pattern.

## 4.5 Results

We have incorporated our periodic boundary condition-constrained simulation and geodesic-based design principles into an interactive tool to help users create flexible kerf pattern designs (Section 4.5.1). We have used our tool to substantially improve the flexibilities of 16 inflexible kerf patterns (Figure 4.1 and 4.9-4.11 and Appendix B) and we walk through the design processes of three patterns (Figure 4.9-4.11) to illustrate the effectiveness of our tool (Section 4.5.2).



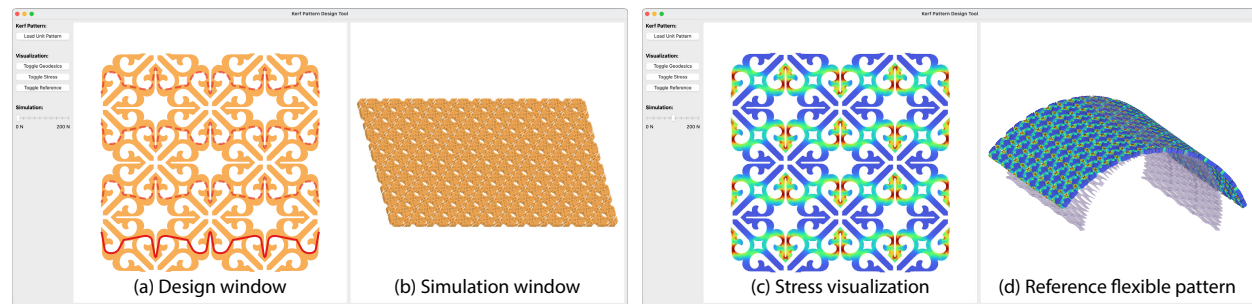


Figure 4.8: Our interactive kerf pattern design tool contains two main windows. The design window on the left shows the 2D unit pattern (a) and the simulation window on the right displays the 3D kerf pattern, which consists of extruded 2D unit patterns tiled on a  $3 \times 5$  grid (b). As users modify the geometry of the unit pattern, the design window highlights in real-time the updated global geodesic (the red line) and distinct local geodesics (the red dotted lines) of the resulting pattern (a), while the simulation window presents the updated bending behavior of the redesigned pattern based on the user-specified bending force (d). The simulation window can additionally show the bending behavior of a known flexible kerf pattern (the purple pattern) under the same bending force to help users better gauge the flexibility level of their pattern design (d). As the pattern undergoes bending, both windows optionally visualize the stress incurred by the pattern to spotlight regions experiencing high levels of deformations (c,d). These regions often fracture easily and users can choose to redesign them first as they edit the pattern to improve its flexibility.

### 4.5.1 Interactive Design Tool

Our interactive kerf pattern design tool takes a 2D vector graphics of a unit kerf pattern as input. The unit pattern must be non-disjoint and form a single, connected kerf pattern when tiled on a grid. Given an input unit pattern, our interactive tool facilitates the design process by providing users with immediate feedbacks on how their modifications to the pattern geometry affect its geodesics and flexibility level in the design and simulation windows (Figure 4.8).

The design window on the left shows the 2D unit kerf pattern and updates in real-time the distinct local geodesics (including the global geodesic) of the pattern as users adjust its geometric design (Figure 4.8a). Based on the updated geodesics, users can follow our design principles and iteratively modify the pattern geometry to reduce the number of distinct local geodesics and make them longer, narrower, and more vertically oriented.

In addition to geodesics, the design window optionally visualizes the stress incurred by the pattern as it undergoes bending to help users identify regions of the pattern experiencing high levels of deformations (Figure 4.8c). These regions often fracture easily, experience more flexures than torsions, and contribute little to the flexibility level of the pattern. Consequently, these regions present users with opportunities to improve the geodesics and flexibility of the pattern and users can choose to redesign these regions first as they edit the pattern. After every modification to the pattern geometry, users can promptly view the flexibility level of the resulting pattern in the simulation window.

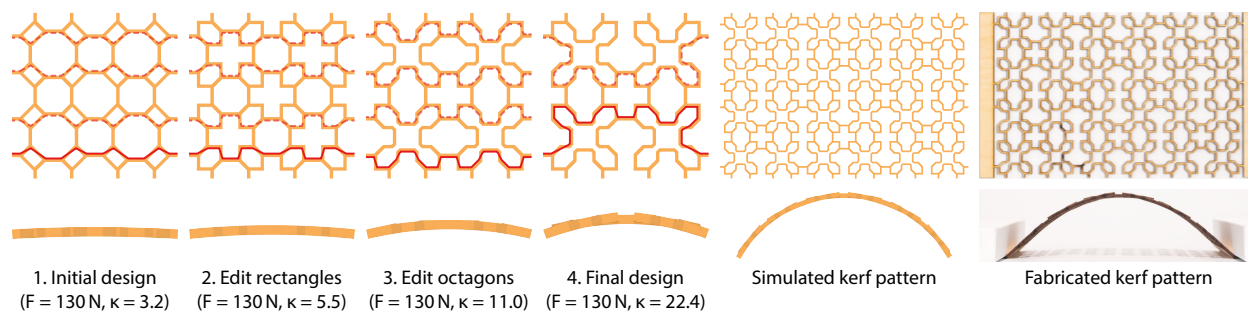


Figure 4.9: For this lattice pattern design, we observe that its geodesics as highlighted by our tool are primarily horizontal. We thus update the (rotated) rectangles in the pattern to be hexagons in order to introduce vertical line segments to the geodesics (Design Principle #2). While this edit does improve the flexibility of the pattern (Step 2), the improvement is minor because the geodesics remain short. Therefore, we make the geodesics longer by modifying the geometries of the octagons (Design Principle #1) and the pattern becomes more flexible as expected (Step 3). Finally, we remove a few horizontal bars from the pattern (Step 4). This removal not only further lengthens the geodesics, but also reduces the number of distinct local geodesics from four to two (Design Principle #3), resulting in a significantly more flexible pattern design. We note that we have added solid bars to the left and right sides of the fabricated pattern only to enable us to bend the pattern more easily.

The simulation window on the right displays the 3D kerf pattern, which consists of extruded 2D unit patterns tiled on a  $3 \times 5$  grid (Figure 4.8b). Users can apply a varying amount of virtual bending force to the left and right sides of the pattern and our tool computes the bending behavior of the pattern at interactive rates based on our periodic boundary condition-constrained simulation (Figure 4.8d). Since virtual bending forces are abstract and difficult to understand tangibly in real-world settings, users can additionally view the bending behavior of a known flexible kerf pattern under the same bending force to better gauge the flexibility level of their pattern design.

By offering users real-time updates on the geodesics and flexibility level of their unit kerf pattern as they edit its geometry, our tool allows users to focus on the visual design of their pattern and lets them efficiently experiment with different geometry modifications to produce a pattern with their desired aesthetic and flexibility level.

### 4.5.2 Kerf Pattern Design Walkthroughs

We have used our interactive tool to create 16 flexible kerf pattern designs (Figure 4.1 and 4.9-4.11 and Appendix B). In Figure 4.1, we significantly increased the flexibility level of a leaf pattern by modifying its geometry based on our design principles and verified that the redesigned leaf pattern is flexible through physical fabrication. We walk through in Figure 4.9-4.11 how we applied our design principles to considerably improve the flexibility of a lattice pattern, a floret pattern, and a fish pattern using our tool. We have fabricated all

three updated patterns to confirm their increased flexibility levels. We show the remaining 12 flexible kerf patterns in Appendix B.

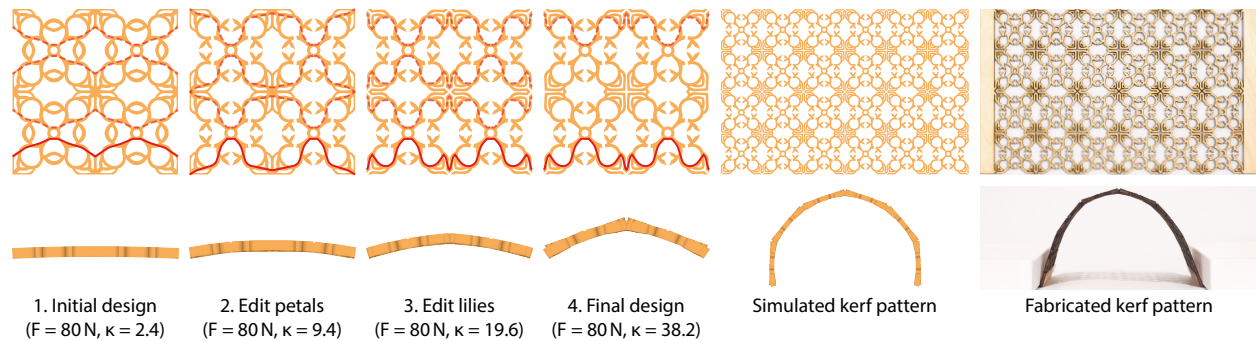


Figure 4.10: To improve the flexibility of this floret pattern, we first apply decorative cuts to its petals (Step 2). These cuts increase the geodesic lengths and consequently the flexibility level of the pattern. We then redesign the geometries of the lilies so that the geodesics contain more vertical line segments and the pattern bends more flexibly as a result (Step 3). We continue to edit the geometries of the lilies along the horizontal middle of the pattern to decrease the number of distinct local geodesics by half, which allows the pattern to be even more flexible (Step 4). This editing session shows that, by applying our design principles deliberately and aesthetically, users can use our tool to create appreciably more flexible patterns while remaining faithful to the original designs.

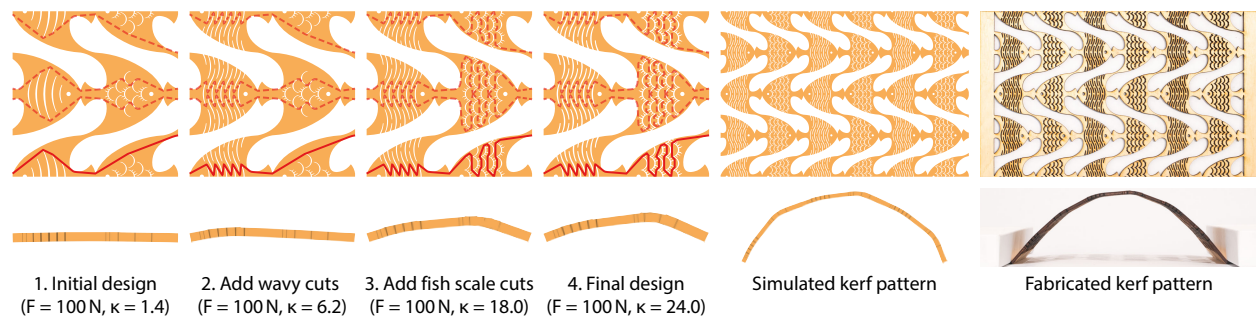


Figure 4.11: We start redesigning this fish pattern by adding wavy cutouts that resemble markings on tropical fish to the left half of the pattern (Step 2). These cutouts force the geodesics to zigzag as opposed to traversing straight through the pattern, therefore making the geodesics longer, as well as more vertically oriented, and the pattern more flexible. We proceed to create cutouts adapted from fish scales in the original design and incorporated them into the right half of the pattern (Step 3). These cutouts additionally lengthen the geodesics and enable the pattern to bend even more. We finish the editing by thickening the cutouts we have added and the flexibility level of the pattern further increases due to the decreased geodesic widths (Step 4). As evidenced by the final design, our tool empowers users to apply the design principles creatively and produce flexible, artistic patterns that take inspirations from the source patterns.

## 4.6 Conclusion

Kerf patterns enable rigid boards to bend flexibly and have become popular among designers thanks to the aesthetic appearances of the patterns and their relative ease of fabrication. But, because their geometric designs are not obviously related to their flexibility levels, designing flexible kerf patterns is challenging and most patterns available online are merely variations of a few known flexible designs. We have presented a kerf pattern design tool that allows users to interactively simulate the bending behavior of their grid-based kerf pattern design and we have established a set of design principles for creating flexible kerf patterns that users can directly apply to their design to improve its flexibility. We believe our interactive tool and design principles will encourage more designers to create and share their own flexible kerf pattern designs.

# Chapter 5

## Conclusion

### 5.1 Summary of Contributions

In this thesis, we have explored ways to empower users to create furniture designs that make effective use of woodworking techniques without prior knowledge. Towards these goals, we have developed interactive tools that help users verify their furniture designs and guide users towards visual designs that satisfy all the requisite geometric and physical constraints. The contributions of our interactive tools can be summarized as follows:

- An interactive tool for enabling users to design and analyze the stability of decorative intrinsic joints for furniture.
  - An algorithm that takes surface 2D parts (i.e., groups of 2D regions on the input solid model surface that must belong to the same part) and automatically generates a set of solid 3D parts that each conform to one of the surface 2D parts and that together assemble without collisions into the input solid model.
  - A novel variational static analysis solver that improves on the shortcomings of the popular equilibrium method to perform stability analysis on the assembly of solid 3D parts and report the locations of sliding and hinging accelerations that are experienced by the unstable parts, if any.
  - A gallery of 100 joint designs and 9 complete furniture assemblies created using our joinery design tool.
- An interactive tool for assisting users in designing ornamental kerf patterns that enable boards made of rigid material to bend flexibly.
  - A periodic boundary condition-constrained finite element method that simulates the bending behaviors of grid-based kerf pattern designs at interactive rates.
  - A set of easily applicable geodesic-based design principles for flexible kerf patterns.
  - A gallery of 75 unit kerf patterns created to help formulate our design principles.

## 5.2 Limitations and Future Work

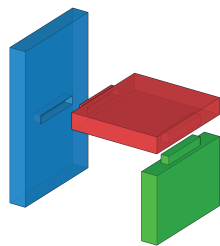
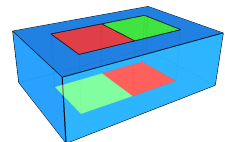
Our research on computational woodworking has shown promising results, but there are many opportunities for improvements and extensions. We discuss here limitations and a few areas of future work related to our interactive design tools for stable decorative joints and flexible kerf patterns.

### Design Tool for Stable Decorative Joints

While our joinery design tool can successfully generate solid 3D parts for a wide variety of user-specified joint designs, our algorithm does have some limitations. First, since we only search over a fixed set of 26 disassembly directions for parts, our tool may deem some input surface 2D parts as not assemblable when in fact they are. One way to mitigate this problem is to increase the search space over additional disassembly directions, but each additional direction adds a dimension to the search space. Combining our approach with an analytical model of the directional blocking graph [71] is a direction for future work.

Second, since our algorithm focuses on generating SOPA compatible parts, it can only generate joints that assemble via single translational motions. Our approach cannot generate joints that assemble via rotational motions, multi-step translations, or deformation-based snap-fit motions. Handling such joints would likely require a much higher dimensional search for identifying collision-free assembly sequences and more sophisticated algorithms for distributing volume between the parts.

Third, we know of one algorithmic limitation. In the inset figure, the red and green 2D parts have identical maximum 3D parts. Our algorithm cannot generate 3D geometries for the red and green squares because neither reallocation attempted by our intersection resolution leaves both parts connected. However, 3D joint geometry similar to a cross lap joint is a valid solution. Note that this is the only algorithmic limitation we have discovered and it never arose in practice after creating over one hundred joints covering a wide range of design space.



Fourth, as our tool is focused on interactively designing decorative joinery, users must manually design joints to correct all part instabilities. As an alternative, we have experimented with automatically adding hidden joints. These are joints on the mating surfaces between two unstable parts which interlock them yet cannot be seen on any surface of the assembled furniture. In our experiments, we automatically add mortise-and-tenon joints aligned with the disassembly direction to the mating surfaces, as shown in the inset figure. This appears to work well in many cases, but more work is needed to generate hidden joints when the disassembly direction is parallel to the mating surface.

Fifth, our tool requires users to draw surface 2D parts as input, i.e., a complete partition of the solid 3D model surface. For more complex models or joinery designs, creating a total partition can sometimes be time-consuming. We believe exploring methods to facilitate or partly automate the partitioning process provides a fruitful direction for future work.

## Design Tool for Flexible Kerf Patterns

Our kerf pattern design tool allows users to apply our geodesic-based design principles and manually modify the geometry of their pattern design to improve its flexibility. While we have successfully created a great diversity of decorative, flexible kerf patterns using this approach, figuring out how to modify the pattern to lengthen its geodesics without significantly altering the appearance of the design can sometimes be challenging. We believe an automated system, built on our design principles, that offers users real-time feedbacks on where and how to redesign their pattern would further facilitate and expedite the kerf pattern design process.

In our own design experiences, we have found that strategically placed cuts on a kerf pattern (i.e., erasures of small areas on the pattern that disconnect some of its regions from one another) affect its geometric design minimally, while forcing its geodesics to reroute and become longer as a result. Additionally, these cuts can reduce the number of distinct local geodesics of the pattern while making them more vertically oriented. Suggesting to users where to apply these cuts on a pattern to increase its flexibility level serves as a meaningful first step in developing an automated kerf pattern design system.

Our interactive tool focuses on facilitating the design of grid-based, periodic kerf patterns that bend flexibly around one direction. While these patterns are the most common among those created by professional designers, our tool can be extended to support more types of kerf pattern designs. First, the rectangular tiling utilized by our tool is a special case of isohedral polygonal tiling, a particular class of periodic bases in which all tiles are congruent to a single polygon. Incorporating this general tiling, as well as other periodic bases, into our tool would enable users to produce an even greater variety of geometric pattern designs.

Second, in addition to periodic patterns, aperiodic patterns can be visually fascinating but often challenging to design. Exploring both the simulation techniques and the design principles to assist users with the design of flexible, aperiodic kerf patterns provides an exciting direction for future research.

Third, the kerf patterns created with our tool can only form simple, cylindrical surfaces when they undergo bending. While these curved surfaces are sufficient in many furniture design scenarios, other disciplines such as architecture and engineering may desire more complex surfaces that resemble paraboloids or hyperboloids. Extending our tool to allow users to create kerf patterns that bend in multiple directions and form geometrically interesting curved surfaces would offer users more opportunities to express their creativities.

# Bibliography

- [1] Maneesh Agrawala et al. “Designing Effective Step-by-step Assembly Instructions”. In: *ACM Trans. Graph.* 22.3 (July 2003), pp. 828–837. ISSN: 0730-0301. DOI: 10.1145/882262.882352. URL: <http://doi.acm.org/10.1145/882262.882352>.
- [2] Moritz Bächer et al. “Spin-it: Optimizing Moment of Inertia for Spinnable Objects”. In: *ACM Trans. Graph.* 33.4 (July 2014), 96:1–96:10. ISSN: 0730-0301. DOI: 10.1145/2601097.2601157. URL: <http://doi.acm.org/10.1145/2601097.2601157>.
- [3] Patrick Baudisch et al. “Kyub: A 3D Editor for Modeling Sturdy Laser-Cut Objects”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–12. ISBN: 9781450359702. URL: <https://doi.org/10.1145/3290605.3300796>.
- [4] Jonathan Benson. *Woodworker’s Guide to Bending Wood*. Fox Chapel Pub., 2009. ISBN: 9781565233607. URL: <https://books.google.com/books?id=XplAvgAACAAJ>.
- [5] Bernd Bickel et al. “Design and Fabrication of Materials with Desired Deformation Behavior”. In: *ACM Trans. Graph.* 29.4 (July 2010), 63:1–63:10. ISSN: 0730-0301. DOI: 10.1145/1778765.1778800. URL: <http://doi.acm.org/10.1145/1778765.1778800>.
- [6] Martin Breuer. *Kerf Bending Patterns*. 2012. URL: <https://www.martin-breuer.com/kerf-bending-patterns>.
- [7] R.G. Budynas and A.M. Sadegh. *Roark’s Formulas for Stress and Strain, 9E*. McGraw-Hill Education, 2020. ISBN: 9781260453751. URL: <https://books.google.com/books?id=JnfyygEACAAJ>.
- [8] Bernhard E. Burdek and Goos Jochen. *Digital Wood Joints*. 1998.
- [9] Glenn Catchpole. *Kerf Table Lamp*. 2013. URL: <https://www.instructables.com/Kerf-Table-Lamp>.
- [10] Desai Chen et al. “Computing and Fabricating Multiplanar Models”. In: *Computer Graphics Forum* 32.2pt3 (2013), pp. 305–315. ISSN: 1467-8659. DOI: 10.1111/cgf.12050. URL: <http://dx.doi.org/10.1111/cgf.12050>.
- [11] Weikai Chen et al. “Fabricable Tile Decors”. In: *ACM Trans. Graph.* 36.6 (Nov. 2017). ISSN: 0730-0301. DOI: 10.1145/3130800.3130817. URL: <https://doi.org/10.1145/3130800.3130817>.



- [12] Weikai Chen et al. “Synthesis of Filigrees for Digital Fabrication”. In: *ACM Trans. Graph.* 35.4 (July 2016), 98:1–98:13. ISSN: 0730-0301. DOI: 10.1145/2897824.2925911. URL: <http://doi.acm.org/10.1145/2897824.2925911>.
- [13] Xuelin Chen et al. “Dapper: Decompose-and-pack for 3D Printing”. In: *ACM Trans. Graph.* 34.6 (Oct. 2015), 213:1–213:12. ISSN: 0730-0301. DOI: 10.1145/2816795.2818087. URL: <http://doi.acm.org/10.1145/2816795.2818087>.
- [14] T.J. Chung. *Applied Continuum Mechanics*. Cambridge University Press, 1996. ISBN: 9780521482974. URL: <https://books.google.com/books?id=GzzXn5F23MUC>.
- [15] Paolo Cignoni et al. “Field-aligned Mesh Joinery”. In: *ACM Trans. Graph.* 33.1 (Feb. 2014), 11:1–11:12. ISSN: 0730-0301. DOI: 10.1145/2537852. URL: <http://doi.acm.org/10.1145/2537852>.
- [16] CncFactory. *Patterns Collection - For Bending Rigid Materials*. 2017. URL: <https://www.instructables.com/Patterns-Collection-for-Bending-Rigid-Materials>.
- [17] R.D. Cook. *Finite Element Modeling for Stress Analysis*. Wiley, 1995. ISBN: 9780471107743. URL: <https://books.google.com/books?id=UyNEAQAAIAAJ>.
- [18] Jérémie Dumas et al. “By-example Synthesis of Structurally Sound Patterns”. In: *ACM Trans. Graph.* 34.4 (July 2015), 137:1–137:12. ISSN: 0730-0301. DOI: 10.1145/2766984. URL: <http://doi.acm.org/10.1145/2766984>.
- [19] N. Engler. *Joining Wood: Techniques for Better Woodworking*. Workshop Companion. Penguin Group USA, 1999. ISBN: 9780762102013. URL: <https://books.google.com/books?id=KjeZNAACAAJ>.
- [20] Chi-Wing Fu et al. “Computational Interlocking Furniture Assembly”. In: *ACM Trans. Graph.* 34.4 (July 2015), 91:1–91:11. ISSN: 0730-0301. DOI: 10.1145/2766892. URL: <http://doi.acm.org/10.1145/2766892>.
- [21] Akash Garg, Alec Jacobson, and Eitan Grinspun. “Computational Design of Reconfigurables”. In: *ACM Trans. Graph.* 35.4 (July 2016). ISSN: 0730-0301. DOI: 10.1145/2897824.2925900. URL: <https://doi.org/10.1145/2897824.2925900>.
- [22] Michael Gilligan. *Live Hinge Table No.1*. 2017. URL: <https://www.mfgilligan.com/#/livehingetableno1>.
- [23] B.J. Goodno and J.M. Gere. *Mechanics of Materials, Enhanced Edition*. Cengage Learning, 2020. ISBN: 9780357377918. URL: <https://books.google.com/books?id=WQfFDwAAQBAJ>.
- [24] Suresh Goyal, Andy Ruina, and Jim Papadopoulos. “Planar sliding with dry friction, Part 1. Limit surface and moment function”. In: *Wear* 143 (1991), pp. 307–330.
- [25] Leonidas J Guibas et al. “A simple and efficient procedure for polyhedral assembly partitioning under infinitesimal motions”. In: *IEEE Robotics and Automation*. Vol. 3. IEEE, 1995, pp. 2553–2560.

- [26] Sigga Helga. *Computer-Controlled Machining*. 2015. URL: [https://fabacademy.org/archives/2015/eu/students/hauksdottir.sigridur\\_helga/W8\\_computer\\_controlled\\_machining.html](https://fabacademy.org/archives/2015/eu/students/hauksdottir.sigridur_helga/W8_computer_controlled_machining.html).
- [27] Philipp Herholz, Wojciech Matusik, and Marc Alexa. “Approximating Free-form Geometry with Height Fields for Manufacturing”. In: *Computer Graphics Forum (Eurographics)* 34.2 (2015), pp. 239–251. DOI: 10.1111/cgf.12556.
- [28] Kristian Hildebrand, Bernd Bickel, and Marc Alexa. “crdbrd: Shape Fabrication by Sliding Planar Slices”. In: *Comp. Graph. Forum* 31.2pt3 (May 2012), pp. 583–592. ISSN: 0167-7055. DOI: 10.1111/j.1467-8659.2012.03037.x. URL: <http://dx.doi.org/10.1111/j.1467-8659.2012.03037.x>.
- [29] R.B. Hoadley. *Understanding Wood: A Craftsman’s Guide to Wood Technology*. Woodworking (Taunton). Taunton Press, 2000. ISBN: 9781561583584. URL: <https://books.google.com/books?id=5HBH2ibu-ZwC>.
- [30] Ruizhen Hu et al. “Approximate Pyramidal Shape Decomposition”. In: *ACM Trans. Graph.* 33.6 (Nov. 2014), 213:1–213:12. ISSN: 0730-0301. DOI: 10.1145/2661229.2661244. URL: <http://doi.acm.org/10.1145/2661229.2661244>.
- [31] S. Huerta. *Mechanics of masonry vaults: the equilibrium approach*. 2001.
- [32] Alexandra Ion et al. “Metamaterial Mechanisms”. In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. UIST ’16. Tokyo, Japan: ACM, 2016, pp. 529–539. ISBN: 978-1-4503-4189-9. DOI: 10.1145/2984511.2984540. URL: <http://doi.acm.org/10.1145/2984511.2984540>.
- [33] Alexandra Ion et al. “Metamaterial Textures”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–12. ISBN: 9781450356206. URL: <https://doi.org/10.1145/3173574.3173910>.
- [34] A. Jackson et al. *Good Wood Joints*. Collins Good Wood Series. Collins, 2002. ISBN: 9780007139767. URL: <https://books.google.com/books?id=8ejpAAAACAAJ>.
- [35] Alec Jacobson. “RodSteward: A Design-to-Assembly System for Fabrication using 3D-Printed Joints and Precision-Cut Rods”. In: *Computer Graphics Forum* (2019).
- [36] Craig S. Kaplan and David H. Salesin. “Escherization”. In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’00. USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 499–510. ISBN: 1581132085. DOI: 10.1145/344779.345022. URL: <https://doi.org/10.1145/344779.345022>.
- [37] Danny M. Kaufman et al. “Staggered Projections for Frictional Contact in Multibody Systems”. In: *ACM TOG (SIGGRAPH Asia)* 27.5 (2008), pp. 1–11.

- [38] S. Krenk. *Mechanics and Analysis of Beams, Columns and Cables: A Modern Introduction to the Classic Theories*. Springer Berlin Heidelberg, 2014. ISBN: 9783642625916. URL: <https://books.google.com/books?id=p3HooAEACAAJ>.
- [39] Maria Larsson et al. “Tsugite: Interactive Design and Fabrication of Wood Joints”. In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. UIST ’20. Virtual Event, USA: Association for Computing Machinery, 2020, pp. 317–327. ISBN: 9781450375146. DOI: 10.1145/3379337.3415899. URL: <https://doi.org/10.1145/3379337.3415899>.
- [40] Manfred Lau et al. “Converting 3D Furniture Models to Fabricatable Parts and Connectors”. In: *ACM Trans. Graph.* 30.4 (July 2011), 85:1–85:6. ISSN: 0730-0301. DOI: 10.1145/2010324.1964980. URL: <http://doi.acm.org/10.1145/2010324.1964980>.
- [41] Danny Leen et al. “JigFab: Computational Fabrication of Constraints to Facilitate Woodworking with Power Tools”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–12. ISBN: 9781450359702. URL: <https://doi.org/10.1145/3290605.3300386>.
- [42] Honghua Li et al. “Foldabilizing Furniture”. In: *ACM Trans. Graph.* 34.4 (July 2015), 90:1–90:12. ISSN: 0730-0301. DOI: 10.1145/2766912. URL: <http://doi.acm.org/10.1145/2766912>.
- [43] Honghua Li et al. “Stackabilization”. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 158:1–158:9. ISSN: 0730-0301. DOI: 10.1145/2366145.2366177. URL: <http://doi.acm.org/10.1145/2366145.2366177>.
- [44] Lin Lu et al. “Build-to-last: Strength to Weight 3D Printed Objects”. In: *ACM Trans. Graph.* 33.4 (July 2014), 97:1–97:10. ISSN: 0730-0301. DOI: 10.1145/2601097.2601168. URL: <http://doi.acm.org/10.1145/2601097.2601168>.
- [45] Linjie Luo et al. “Chopper: Partitioning Models into 3D-printable Parts”. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 129:1–129:9. ISSN: 0730-0301. DOI: 10.1145/2366145.2366148. URL: <http://doi.acm.org/10.1145/2366145.2366148>.
- [46] Nancy Macdonald. *Woodworking*. Cengage Learning, 2013. ISBN: 9781285700502. URL: <https://books.google.com/books?id=zTUXAAAAQBAJ>.
- [47] Shiran Magrisso, Moran Mizrahi, and Amit Zoran. “Digital Joinery For Hybrid Carpentry”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–11. ISBN: 9781450356206. URL: <https://doi.org/10.1145/3173574.3173741>.
- [48] Denzil Makes. *Living Hinge Patterns*. 2017. URL: <https://denzilmakes.com/collections/plans/products/living-hinge-pattern-dxf-file>.
- [49] Denzil Makes. *Living Hinge Patterns II*. 2019. URL: <https://denzilmakes.com/collections/plans/products/living-hinge-patterns-ii-dxf-file>.

- [50] Jonàs Martinez, Jérémie Dumas, and Sylvain Lefebvre. “Procedural Voronoi Foams for Additive Manufacturing”. In: *ACM Trans. Graph.* 35.4 (July 2016), 44:1–44:12. ISSN: 0730-0301. DOI: 10.1145/2897824.2925922. URL: <http://doi.acm.org/10.1145/2897824.2925922>.
- [51] Jonàs Martinez et al. “Orthotropic k-Nearest Foams for Additive Manufacturing”. In: *ACM Trans. Graph.* 36.4 (July 2017). ISSN: 0730-0301. DOI: 10.1145/3072959.3073638. URL: <https://doi.org/10.1145/3072959.3073638>.
- [52] Jonàs Martinez et al. “Star-Shaped Metrics for Mechanical Metamaterial Design”. In: *ACM Trans. Graph.* 38.4 (July 2019). ISSN: 0730-0301. DOI: 10.1145/3306346.3322989. URL: <https://doi.org/10.1145/3306346.3322989>.
- [53] Jonàs Martinez et al. “Structure and Appearance Optimization for Controllable Shape Design”. In: *ACM Trans. Graph.* 34.6 (Oct. 2015), 229:1–229:11. ISSN: 0730-0301. DOI: 10.1145/2816795.2818101. URL: <http://doi.acm.org/10.1145/2816795.2818101>.
- [54] James McCrae, Nobuyuki Umetani, and Karan Singh. “FlatFitFab: Interactive Modeling with Planar Sections”. In: *User Interface Software and Technology (UIST)*. Honolulu, Hawaii, USA, 2014, pp. 13–22. ISBN: 978-1-4503-3069-5. DOI: 10.1145/2642918.2647388. URL: <http://doi.acm.org/10.1145/2642918.2647388>.
- [55] Johannes Mezger et al. “Interactive Physically-based Shape Editing”. In: *Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling*. SPM ’08. Stony Brook, New York: ACM, 2008, pp. 79–89. ISBN: 978-1-60558-106-4. DOI: 10.1145/1364901.1364915. URL: <http://doi.acm.org/10.1145/1364901.1364915>.
- [56] Joseph S. B. Mitchell, David M. Mount, and Christos H. Papadimitriou. “The Discrete Geodesic Problem”. In: *SIAM J. Comput.* 16.4 (Aug. 1987), pp. 647–668. ISSN: 0097-5397. DOI: 10.1137/0216045. URL: <https://doi.org/10.1137/0216045>.
- [57] Jean Jaques Moreau. “Quadratic Programming in Mechanics: One-Sided Constraints”. In: *Journal SIAM Control* 4.1 (1966), pp. 153–158.
- [58] Matthias Müller and Markus Gross. “Interactive Virtual Materials”. In: *Proceedings of Graphics Interface 2004*. GI ’04. London, Ontario, Canada: Canadian Human-Computer Communications Society, 2004, pp. 239–246. ISBN: 1-56881-227-2. URL: <http://dl.acm.org/citation.cfm?id=1006058.1006087>.
- [59] Matthias Müller et al. “Real-time Simulation of Deformation and Fracture of Stiff Materials”. In: *Proceedings of the Eurographic Workshop on Computer Animation and Simulation*. Manchester, UK: Springer-Verlag New York, Inc., 2001, pp. 113–124. ISBN: 3-211-83711-6. URL: <http://dl.acm.org/citation.cfm?id=776350.776361>.
- [60] T. Noll. *Joint Book: The Complete Guide to Wood Joinery*. A Quarto book. Chartwell Books, 2009. ISBN: 9780785822271. URL: <https://books.google.com/books?id=pvyaPQAACAAJ>.

- [61] James F. O'Brien and Jessica K. Hodgins. "Graphical Modeling and Animation of Brittle Fracture". In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 137–146. ISBN: 0-201-48560-5. DOI: 10.1145/311535.311550. URL: <http://dx.doi.org/10.1145/311535.311550>.
- [62] James F. O'Brien, Chen Shen, and Christine M. Gatchalian. "Synthesizing Sounds from Rigid-body Simulations". In: *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '02. San Antonio, Texas: ACM, 2002, pp. 175–181. ISBN: 1-58113-573-4. DOI: 10.1145/545261.545290. URL: <http://doi.acm.org/10.1145/545261.545290>.
- [63] J A Ochsendorf. "Collapse of Masonry Structures. University of Cambridge". PhD thesis. University of Cambridge, 2002.
- [64] Yeonjoo Oh et al. "The Designosaur and the Furniture Factory". English. In: *Design Computing and Cognition*. Ed. by Johns. Gero. Springer Netherlands, 2006, pp. 123–140. ISBN: 978-1-4020-5130-2. DOI: 10.1007/978-1-4020-5131-9\_7. URL: [http://dx.doi.org/10.1007/978-1-4020-5131-9\\_7](http://dx.doi.org/10.1007/978-1-4020-5131-9_7).
- [65] Julian Panetta et al. "Elastic Textures for Additive Fabrication". In: *ACM Trans. Graph.* 34.4 (July 2015), 135:1–135:12. ISSN: 0730-0301. DOI: 10.1145/2766937. URL: <http://doi.acm.org/10.1145/2766937>.
- [66] Aaron Porterfield. In-person interview. 2018.
- [67] Aaron Porterfield. *Curved Laser Bent Wood*. 2014. URL: <https://www.instructables.com/Curved-laser-bent-wood>.
- [68] J. Postell. *Furniture Design*. Wiley, 2012.
- [69] Romain Prévost et al. "Make It Stand: Balancing Shapes for 3D Fabrication". In: *ACM Trans. Graph.* 32.4 (July 2013), 81:1–81:10. ISSN: 0730-0301. DOI: 10.1145/2461912.2461957. URL: <http://doi.acm.org/10.1145/2461912.2461957>.
- [70] Gary Rogowski. *The Complete Illustrated Guide To Joinery. Complete Illustrated Guide*. Taunton Press, 2002.
- [71] Bruce Romney et al. "An efficient system for geometric assembly sequence generation and evaluation". In: *COMPUTERS IN ENGINEERING* (1995), pp. 699–712.
- [72] Thijs Roumen et al. "Kerf-Canceling Mechanisms: Making Laser-Cut Mechanisms Operate across Different Laser Cutters". In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. UIST '20. Virtual Event, USA: Association for Computing Machinery, 2020, pp. 293–303. ISBN: 9781450375146. DOI: 10.1145/3379337.3415895. URL: <https://doi.org/10.1145/3379337.3415895>.

- [73] Daniel Saakes et al. “PacCAM: Material Capture and Interactive 2D Packing for Efficient Material Usage on CNC Cutting Machines”. In: *User Interface Software and Technology (UIST)*. St. Andrews, Scotland, United Kingdom, 2013, pp. 441–446. ISBN: 978-1-4503-2268-3. DOI: 10.1145/2501988.2501990. URL: <http://doi.acm.org/10.1145/2501988.2501990>.
- [74] Greg Saul et al. “SketchChair: An All-in-one Chair Design System for End Users”. In: *Tangible, Embedded, and Embodied Interaction (TEI)*. Funchal, Portugal, 2011, pp. 73–80. ISBN: 978-1-4503-0478-8. DOI: 10.1145/1935701.1935717. URL: <http://doi.acm.org/10.1145/1935701.1935717>.
- [75] Lon Schleining. *The Complete Manual of Wood Bending: Milled, Laminated, and Steam-bent Work*. Linden, 2002. ISBN: 9780941936545. URL: <https://books.google.com/books?id=hQdUAAAAMAAJ>.
- [76] Lon Schleining. *Wood Bending Made Simple*. Made Simple (Taunton Press) Series. Taunton Press, 2010. ISBN: 9781600852497. URL: <https://books.google.com/books?id=BmvB2TQ1Y2AC>.
- [77] Adriana Schulz et al. “Design and Fabrication by Example”. In: *ACM Trans. Graph.* 33.4 (July 2014), 62:1–62:11. ISSN: 0730-0301. DOI: 10.1145/2601097.2601127. URL: <http://doi.acm.org/10.1145/2601097.2601127>.
- [78] Christian Schumacher, Bernhard Thomaszewski, and Markus Gross. “Stenciling: Designing Structurally-sound Surfaces with Decorative Patterns”. In: *Proceedings of the Symposium on Geometry Processing. SGP ’16*. Berlin, Germany: Eurographics Association, 2016, pp. 101–110. DOI: 10.1111/cgf.12967. URL: <https://doi.org/10.1111/cgf.12967>.
- [79] Christian Schumacher et al. “Mechanical Characterization of Structured Sheet Materials”. In: *ACM Trans. Graph.* 37.4 (July 2018). ISSN: 0730-0301. DOI: 10.1145/3197517.3201278. URL: <https://doi.org/10.1145/3197517.3201278>.
- [80] Christian Schumacher et al. “Microstructures to Control Elasticity in 3D Printing”. In: *ACM Trans. Graph.* 34.4 (July 2015), 136:1–136:13. ISSN: 0730-0301. DOI: 10.1145/2766926. URL: <http://doi.acm.org/10.1145/2766926>.
- [81] Yuliy Schwartzburg and Mark Pauly. “Fabrication-aware Design with Intersecting Planar Pieces”. In: *Computer Graphics Forum* 32.2 (2013), pp. 317–326.
- [82] K. Seike, Y. Yobuko, and R.M. Davis. *The Art of Japanese Joinery*. Weatherhill, 1986. URL: <https://books.google.com/books?id=DgHvgAEACAAJ>.
- [83] Hijung V Shin et al. “Reconciling Elastic and Equilibrium Methods for Static Analysis”. In: *ACM Trans. Graph.* 35.2 (2016), pp. 13–16.
- [84] Hijung V. Shin. personal communication. 2016.

- [85] Mélina Skouras et al. “Computational Design of Actuated Deformable Characters”. In: *ACM Trans. Graph.* 32.4 (July 2013), 82:1–82:10. ISSN: 0730-0301. DOI: 10.1145/2461912.2461979. URL: <http://doi.acm.org/10.1145/2461912.2461979>.
- [86] Peng Song, Chi-Wing Fu, and Daniel Cohen-Or. “Recursive Interlocking Puzzles”. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 128:1–128:10. ISSN: 0730-0301. DOI: 10.1145/2366145.2366147. URL: <http://doi.acm.org/10.1145/2366145.2366147>.
- [87] Peng Song et al. “Reconfigurable Interlocking Furniture”. In: *ACM Trans. Graph.* 36.6 (Nov. 2017). ISSN: 0730-0301. DOI: 10.1145/3130800.3130803. URL: <https://doi.org/10.1145/3130800.3130803>.
- [88] Ondrej Stava et al. “Stress Relief: Improving Structural Strength of 3D Printable Objects”. In: *ACM Trans. Graph.* 31.4 (July 2012), 48:1–48:11. ISSN: 0730-0301. DOI: 10.1145/2185520.2185544. URL: <http://doi.acm.org/10.1145/2185520.2185544>.
- [89] Zachary Taylor. *Wood Bender’s Handbook*. Sterling, 2008. ISBN: 9781402756146. URL: <https://books.google.com/books?id=jgsmzNcIe0gC>.
- [90] Rundong Tian et al. “MatchSticks: Woodworking through Improvisational Digital Fabrication”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–12. ISBN: 9781450356206. URL: <https://doi.org/10.1145/3173574.3173723>.
- [91] Godfried Toussaint. “Movable separability of sets”. In: *Computational Geometry*. Citeseer, 1985.
- [92] A.C. Ugural and S.K. Fenster. *Advanced Strength and Applied Elasticity*. Prentice Hall PTR, 2003. ISBN: 9780130473929. URL: <https://books.google.com/books?id=BqZRAAAAMAAJ>.
- [93] Nobuyuki Umetani, Takeo Igarashi, and Niloy J. Mitra. “Guided Exploration of Physically Valid Shapes for Furniture Design”. In: *ACM Trans. Graph.* 31.4 (July 2012), 86:1–86:11. ISSN: 0730-0301. DOI: 10.1145/2185520.2185582. URL: <http://doi.acm.org/10.1145/2185520.2185582>.
- [94] Nobuyuki Umetani et al. “Pteromys: Interactive Design and Optimization of Free-formed Free-flight Model Airplanes”. In: *ACM Trans. Graph.* 33.4 (July 2014), 65:1–65:10. ISSN: 0730-0301. DOI: 10.1145/2601097.2601129. URL: <http://doi.acm.org/10.1145/2601097.2601129>.
- [95] Weiming Wang et al. “Cost-effective Printing of 3D Objects with Skin-frame Structures”. In: *ACM Trans. Graph.* 32.6 (Nov. 2013), 177:1–177:10. ISSN: 0730-0301. DOI: 10.1145/2508363.2508382. URL: <http://doi.acm.org/10.1145/2508363.2508382>.

- [96] Ziqi Wang, Peng Song, and Mark Pauly. “DESIA: A General Framework for Designing Interlocking Assemblies”. In: *ACM Trans. Graph.* 37.6 (Dec. 2018). ISSN: 0730-0301. DOI: 10.1145/3272127.3275034. URL: <https://doi.org/10.1145/3272127.3275034>.
- [97] Emily Whiting, John Ochsendorf, and Frédo Durand. “Procedural Modeling of Structurally-sound Masonry Buildings”. In: *ACM Trans. Graph.* 28.5 (Dec. 2009), 112:1–112:9. ISSN: 0730-0301. DOI: 10.1145/1618452.1618458. URL: <http://doi.acm.org/10.1145/1618452.1618458>.
- [98] Emily Whiting et al. “Structural Optimization of 3D Masonry Buildings”. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 159:1–159:11. ISSN: 0730-0301. DOI: 10.1145/2366145.2366178. URL: <http://doi.acm.org/10.1145/2366145.2366178>.
- [99] Randall H Wilson. “On geometric assembly planning”. PhD thesis. Stanford University, 1992.
- [100] Chenming Wu et al. “Carpentry Compiler”. In: *ACM Trans. Graph.* 38.6 (Nov. 2019). ISSN: 0730-0301. DOI: 10.1145/3355089.3356518. URL: <https://doi.org/10.1145/3355089.3356518>.
- [101] Shiqing Xin et al. “Making Burr Puzzles from 3D Models”. In: *ACM Trans. Graph.* 30.4 (July 2011), 97:1–97:8. ISSN: 0730-0301. DOI: 10.1145/2010324.1964992. URL: <http://doi.acm.org/10.1145/2010324.1964992>.
- [102] Hongyi Xu et al. “Interactive Material Design Using Model Reduction”. In: *ACM Trans. Graph.* 34.2 (Mar. 2015), 18:1–18:14. ISSN: 0730-0301. DOI: 10.1145/2699648. URL: <http://doi.acm.org/10.1145/2699648>.
- [103] Yong-Liang Yang, Jun Wang, and Niloy J. Mitra. “Reforming Shapes for Material-aware Fabrication”. In: *Comput. Graph. Forum* 34.5 (Aug. 2015), pp. 53–64. ISSN: 0167-7055. DOI: 10.1111/cgf.12696. URL: <http://dx.doi.org/10.1111/cgf.12696>.
- [104] Jiaxian Yao et al. “Interactive Design and Stability Analysis of Decorative Joinery for Furniture”. In: *ACM Trans. Graph.* 36.2 (Mar. 2017), 20:1–20:16. ISSN: 0730-0301. DOI: 10.1145/3054740. URL: <http://doi.acm.org/10.1145/3054740>.
- [105] Miaojun Yao et al. “Level-set-based Partitioning and Packing Optimization of a Printable Model”. In: *ACM Trans. Graph.* 34.6 (Oct. 2015), 214:1–214:11. ISSN: 0730-0301. DOI: 10.1145/2816795.2818064. URL: <http://doi.acm.org/10.1145/2816795.2818064>.
- [106] Jonas Zehnder, Stelian Coros, and Bernhard Thomaszewski. “Designing Structurally-sound Ornamental Curve Networks”. In: *ACM Trans. Graph.* 35.4 (July 2016), 99:1–99:10. ISSN: 0730-0301. DOI: 10.1145/2897824.2925888. URL: <http://doi.acm.org/10.1145/2897824.2925888>.



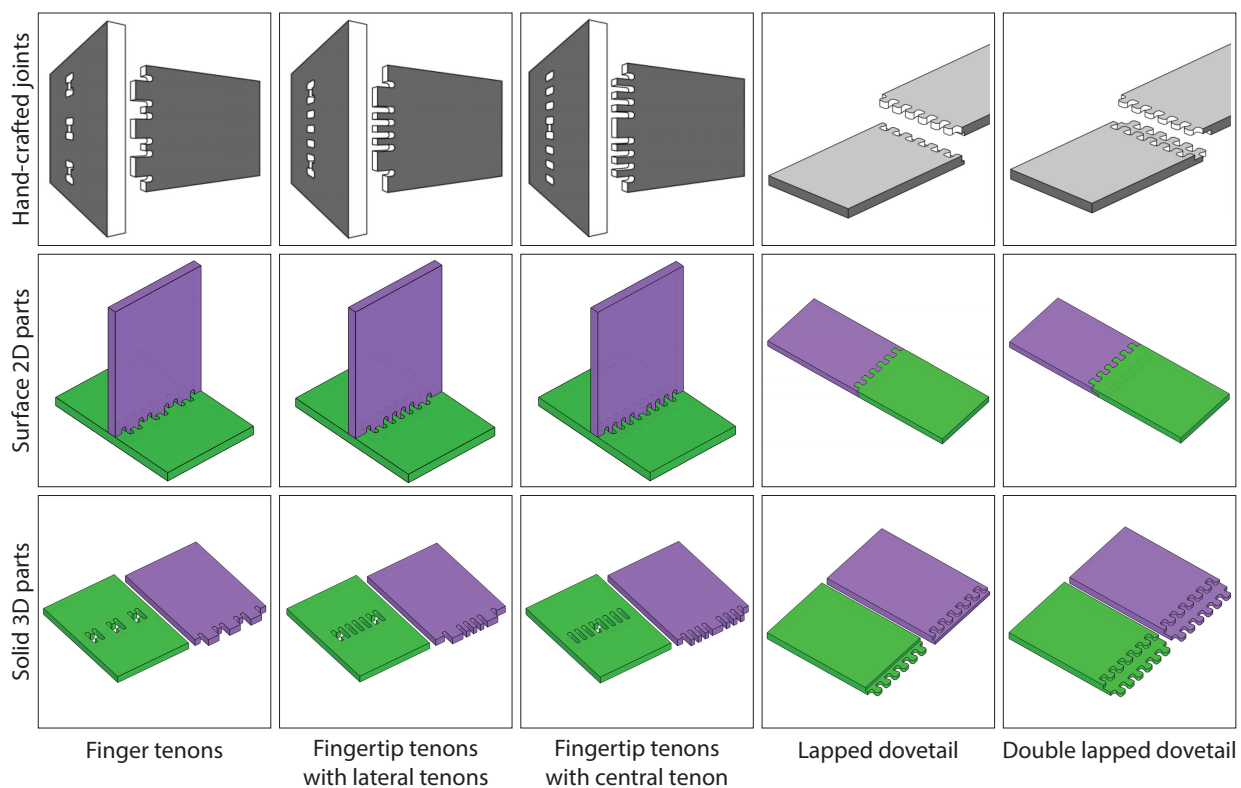
- [107] Changxi Zheng and Doug L. James. “Rigid-Body Fracture Sound with Precomputed Soundbanks”. In: *ACM SIGGRAPH 2010 Papers*. SIGGRAPH '10. Los Angeles, California: Association for Computing Machinery, 2010. ISBN: 9781450302104. DOI: 10.1145/1833349.1778806. URL: <https://doi.org/10.1145/1833349.1778806>.
- [108] Qingnan Zhou et al. “Mesh Arrangements for Solid Geometry”. In: *ACM Trans. Graph.* 35.4 (July 2016), 39:1–39:15. ISSN: 0730-0301. DOI: 10.1145/2897824.2925901. URL: <http://doi.acm.org/10.1145/2897824.2925901>.

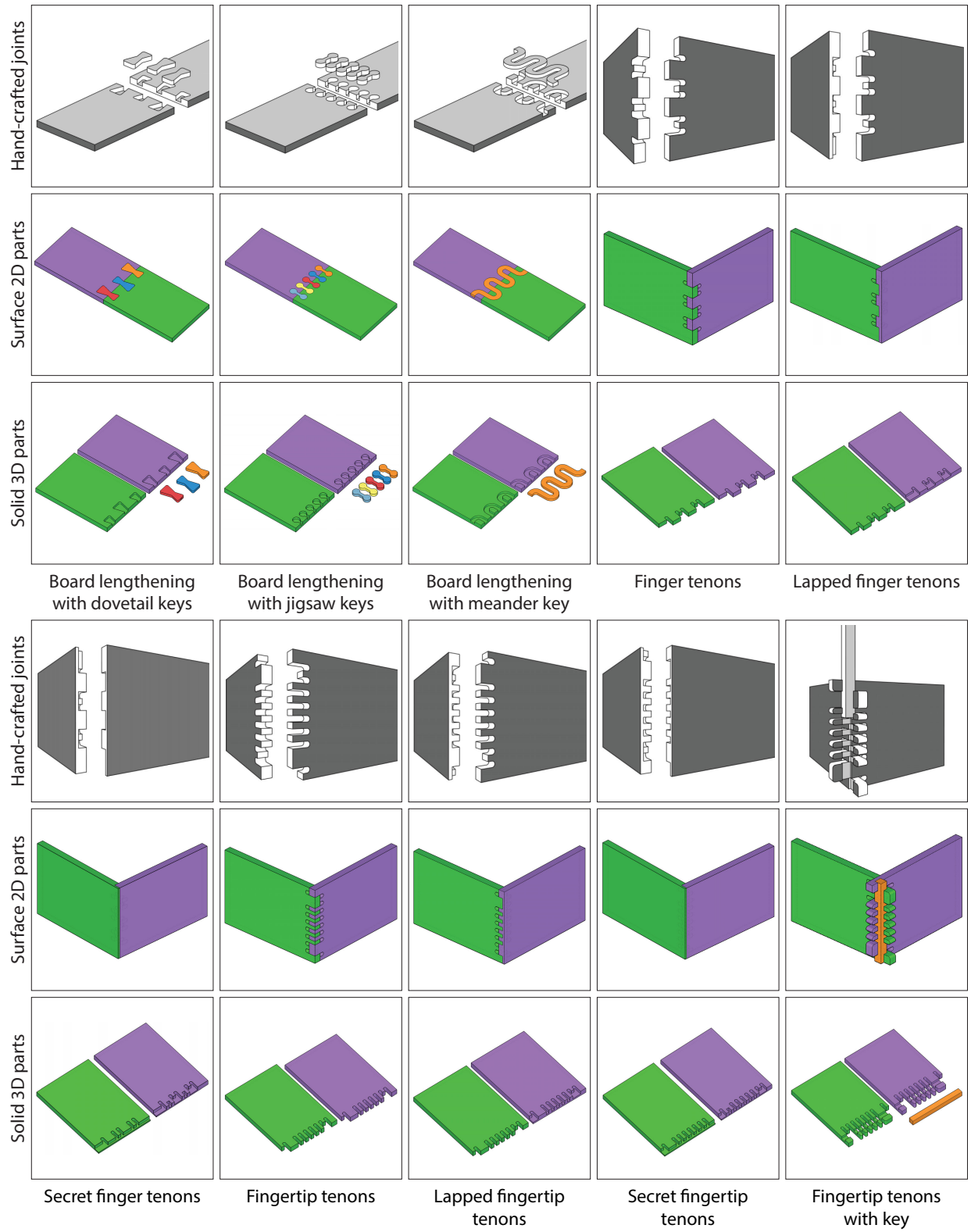
# Appendix A

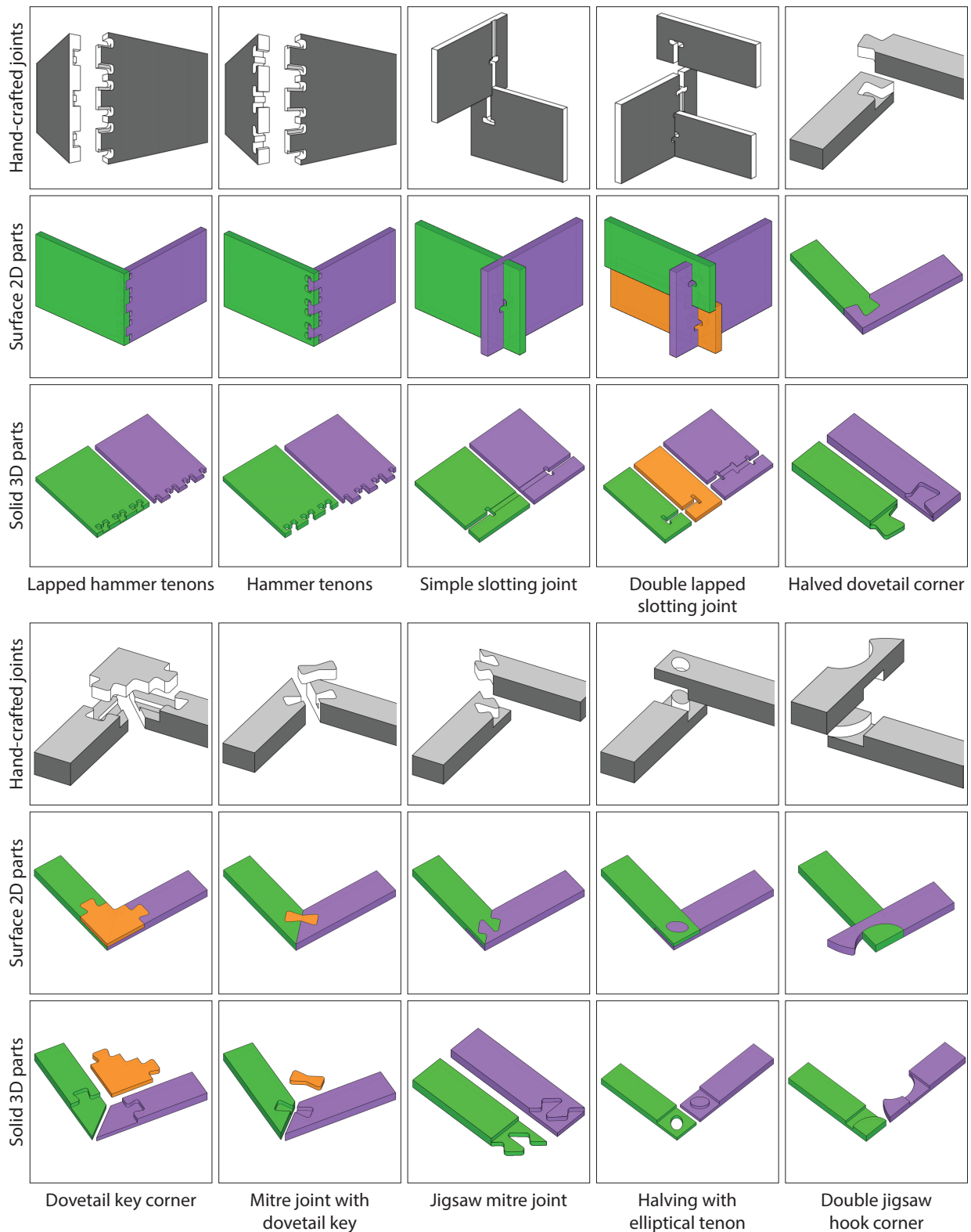
## Appendix for Chapter 3

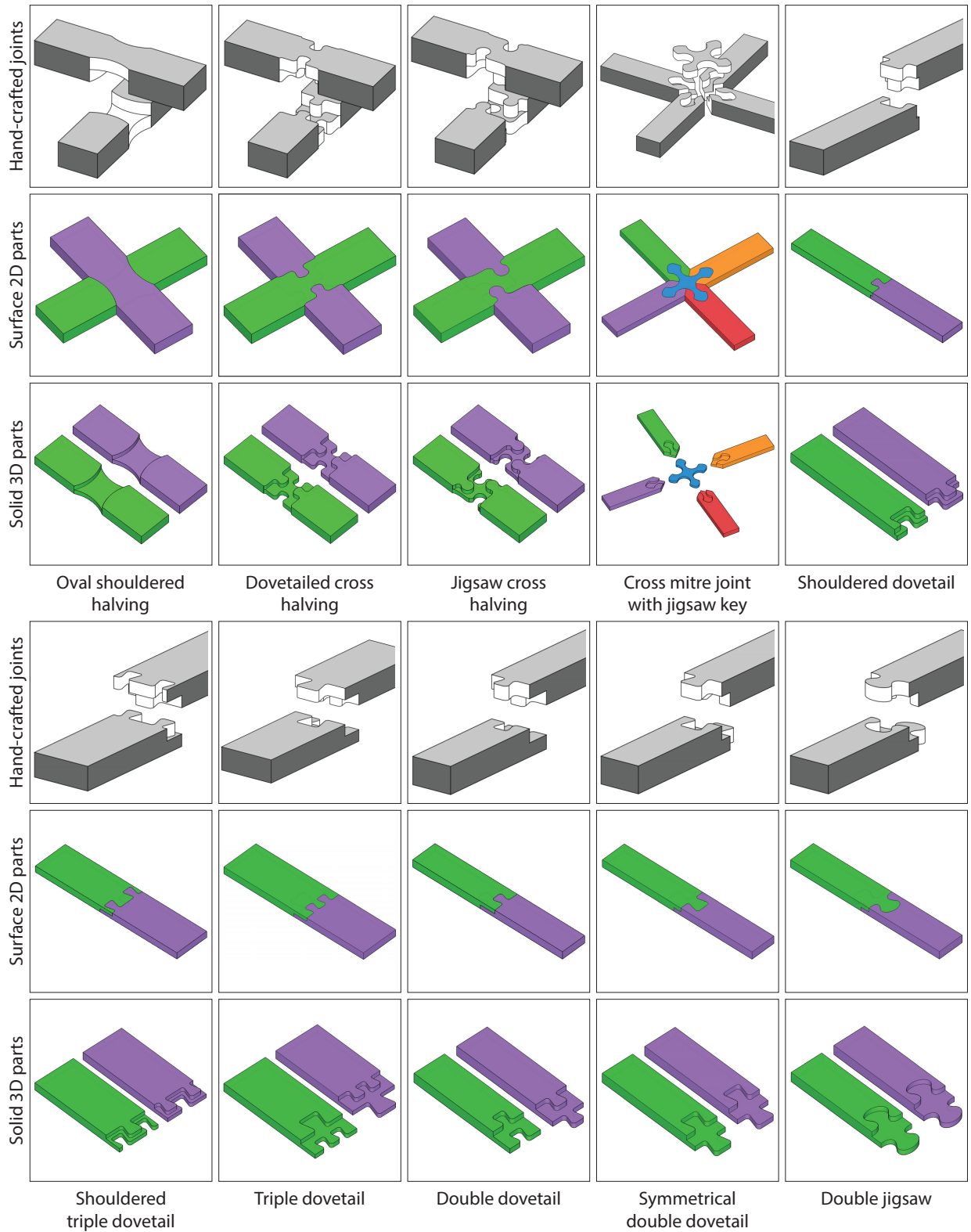
In this appendix, we provide 100 decorative joints we have created using our interactive joinery design tool, as well as additional joints we have fabricated.

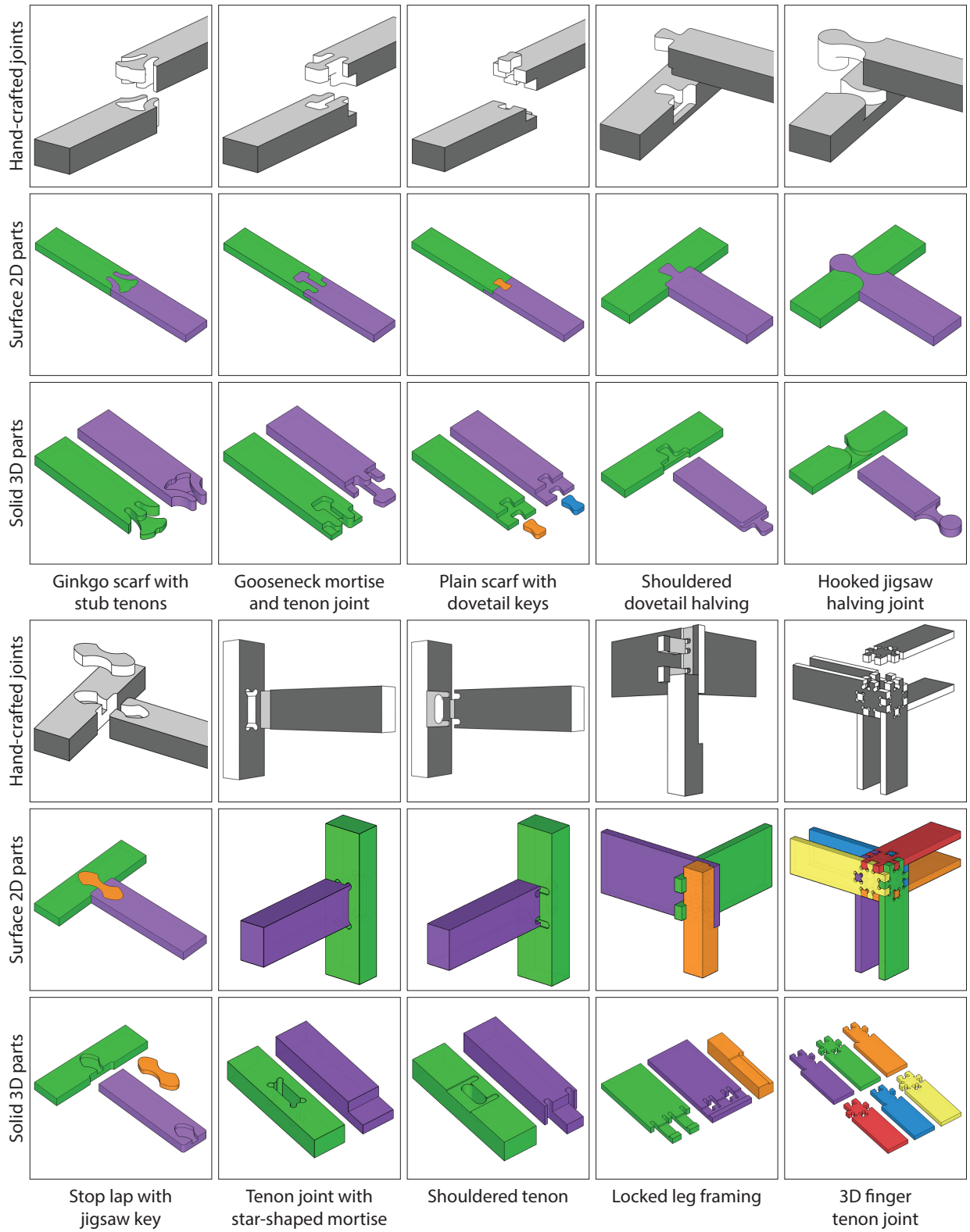
### A.1 Gallery of 100 Joint Designs

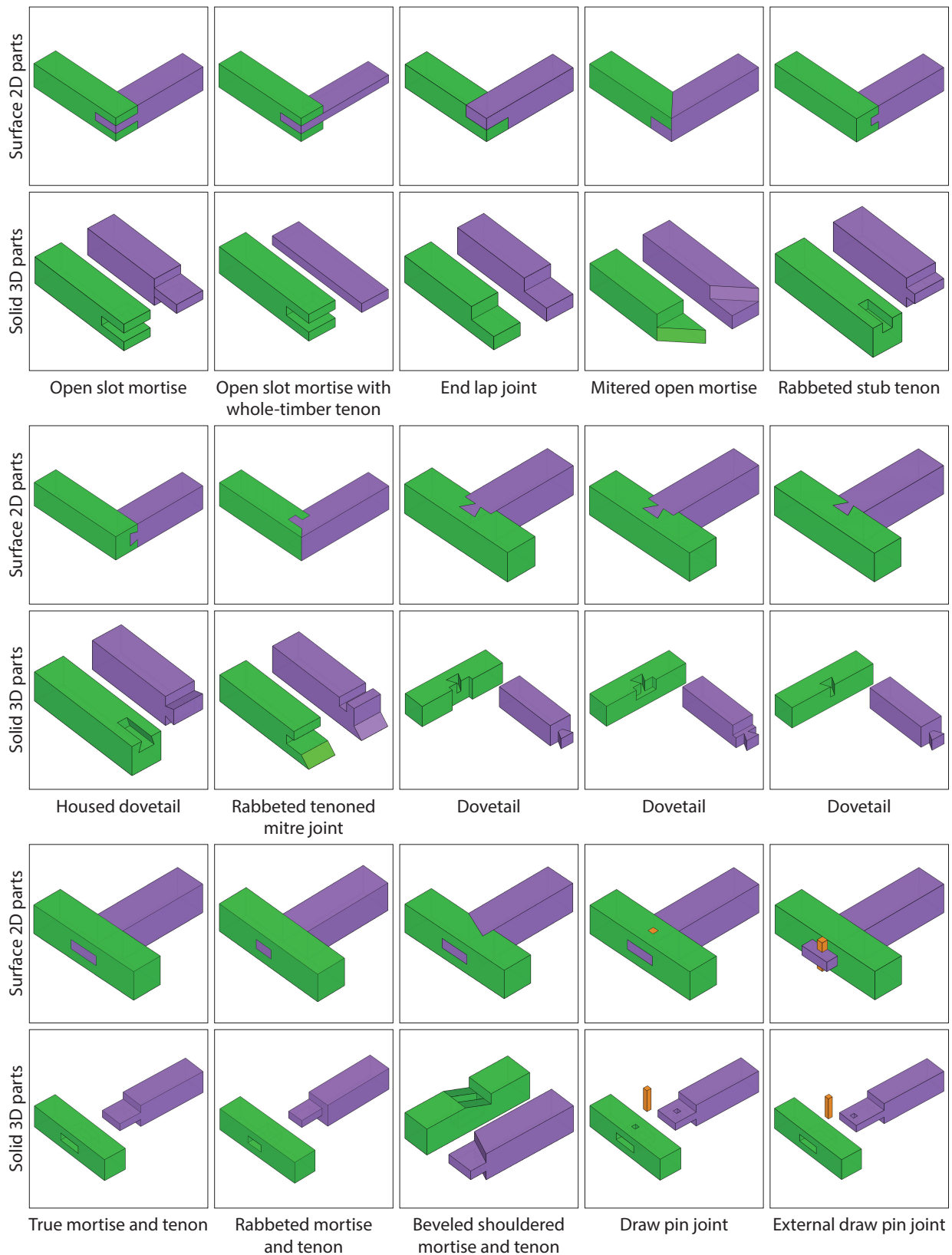


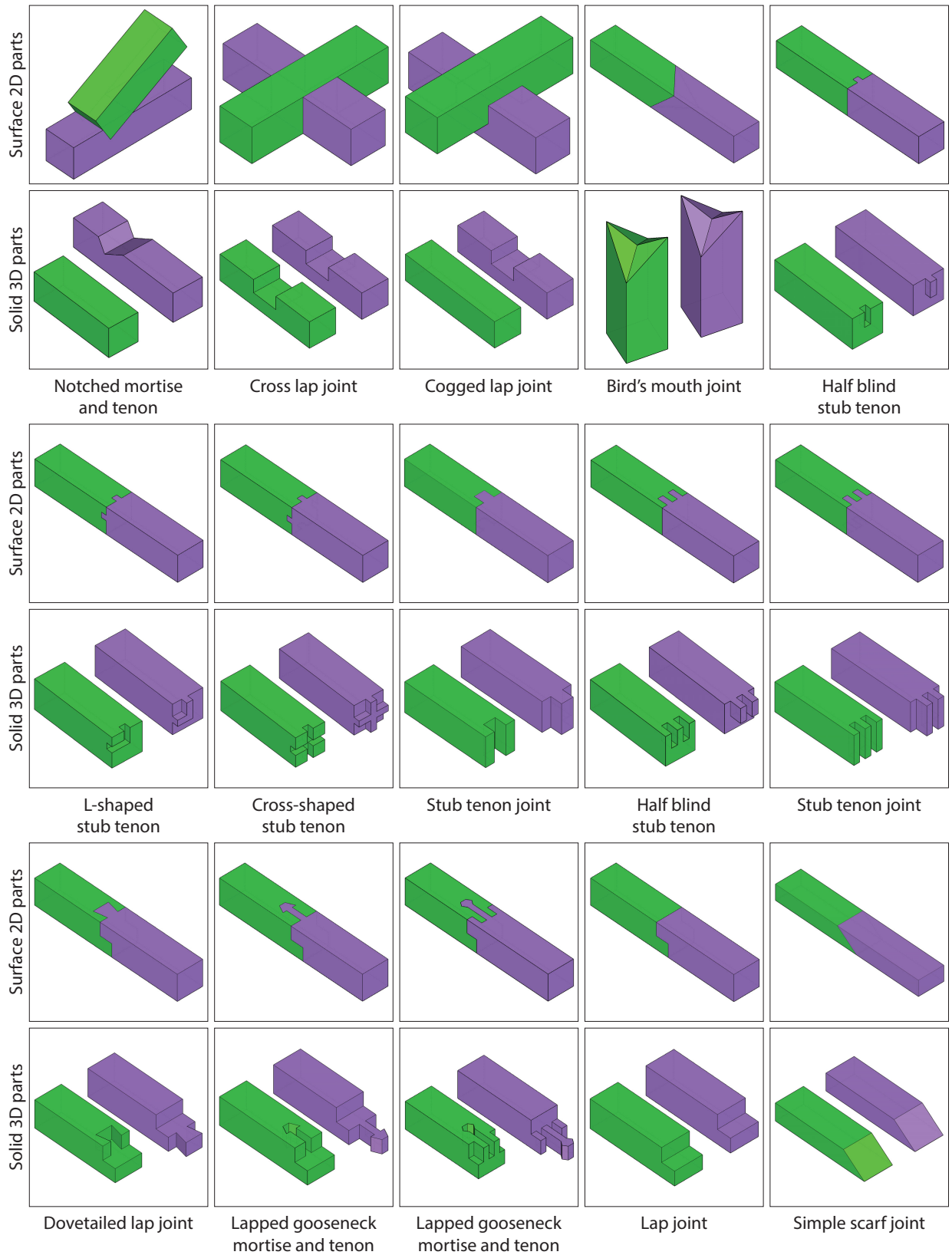




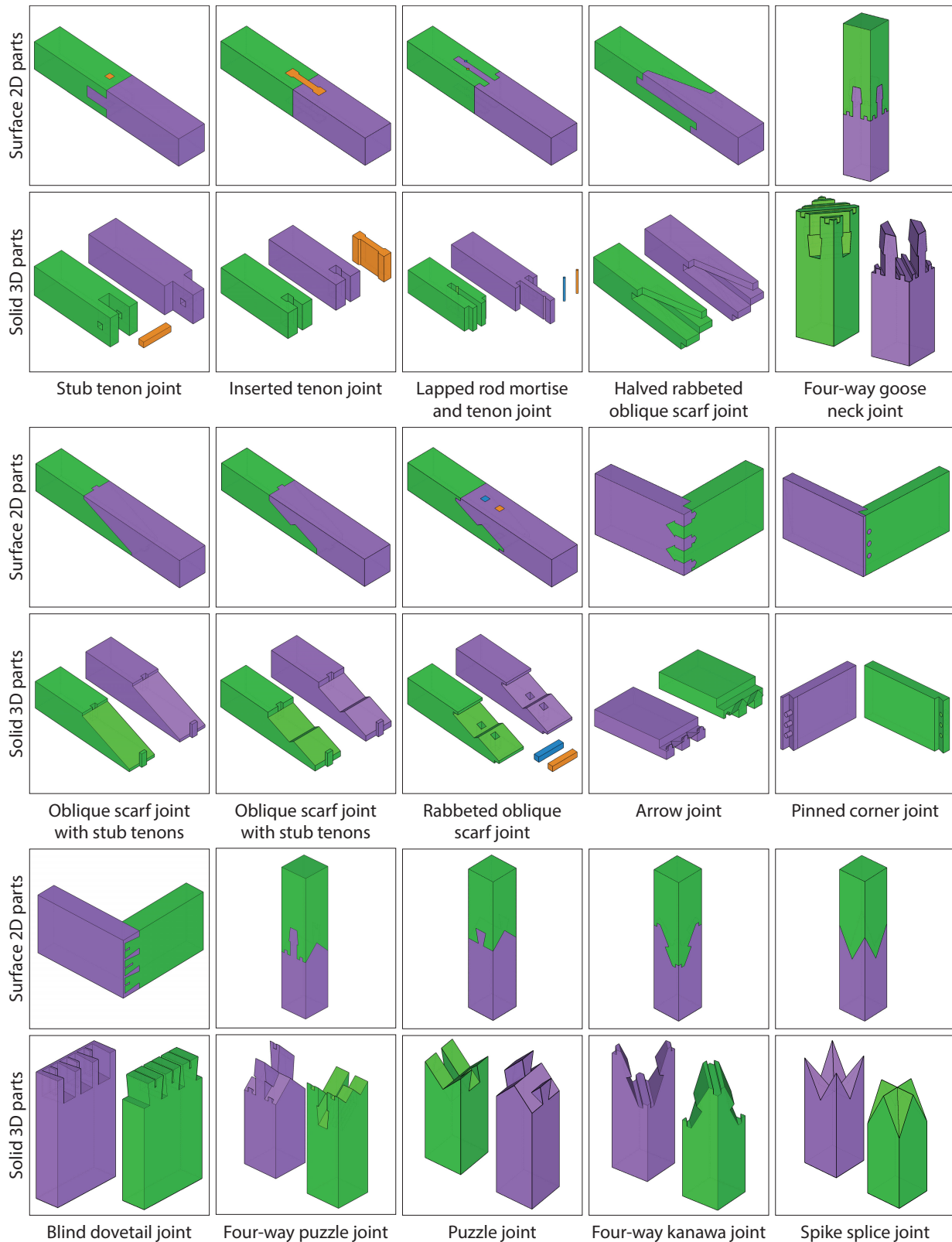


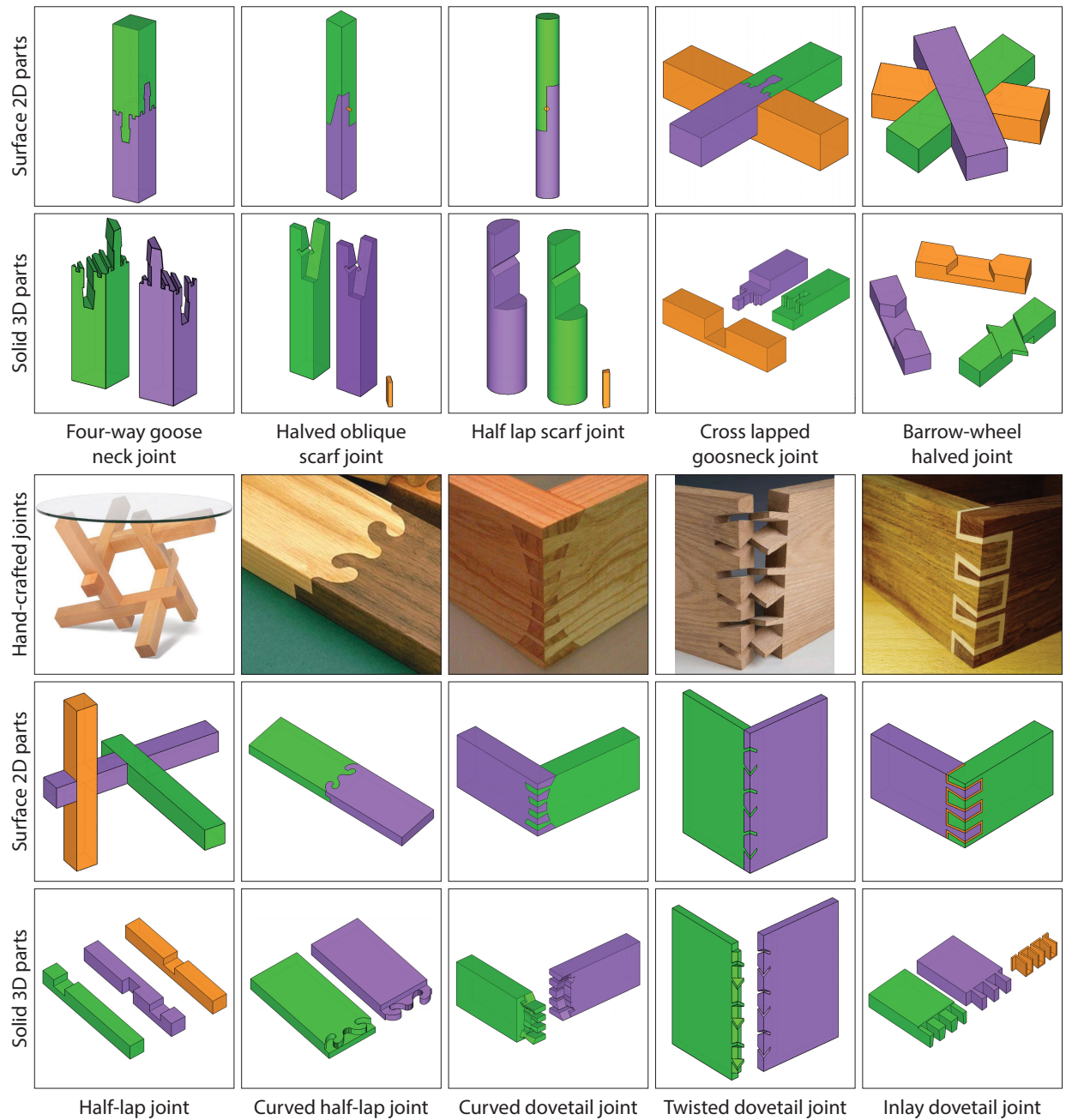










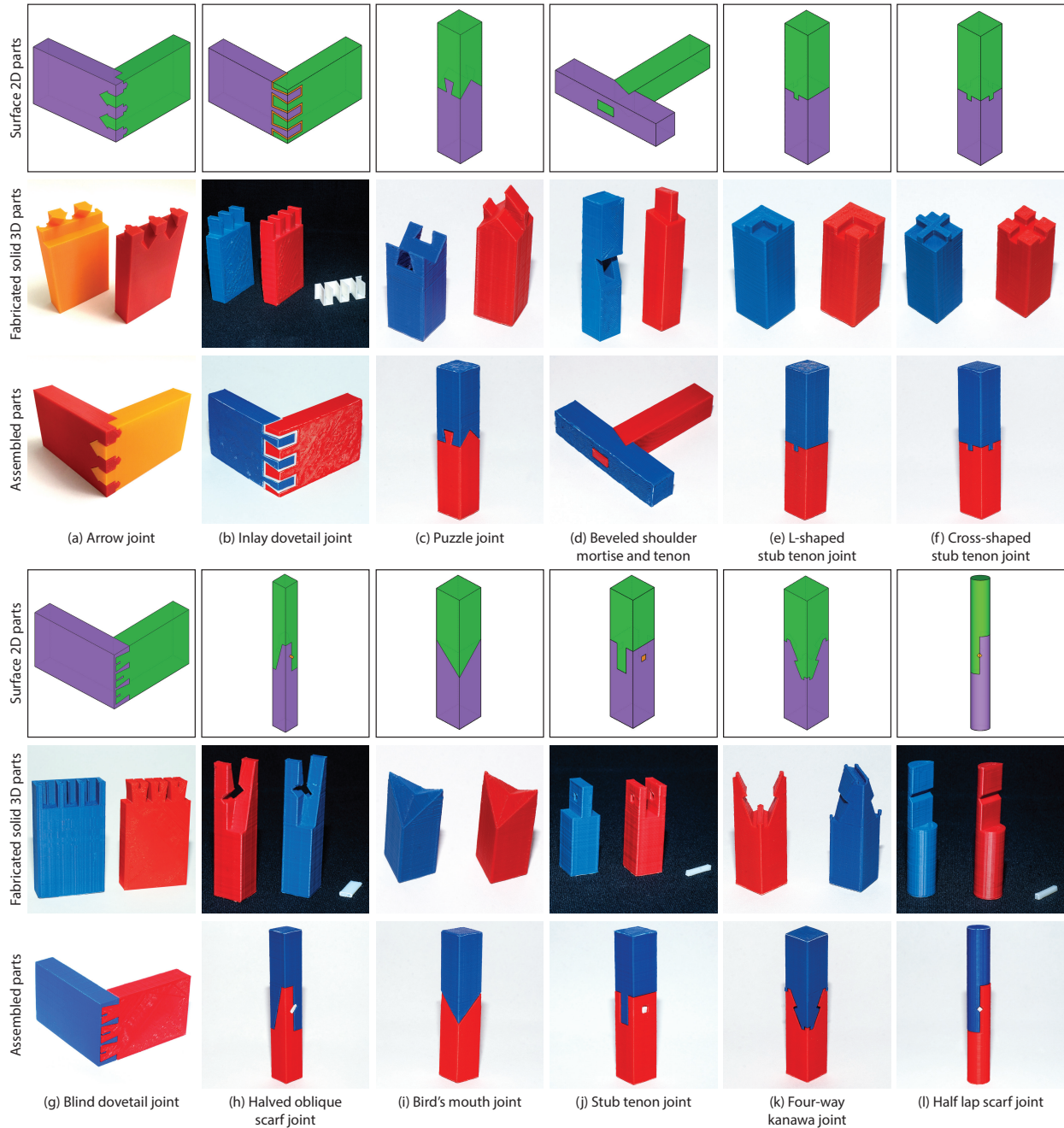


We have created these 100 joints based on photographs we have found in traditional woodworking guides [70, 60], Japanese joinery books [82], and joinery examples from the popular woodworking websites<sup>1,2</sup>.

<sup>1</sup><http://wwideas.com/2015/11/the-most-impressive-wood-joints/>

<sup>2</sup><https://winterdienst.info/50-digital-wood-joints-by-jochen-gros/>

## A.2 Gallery of Fabricated Joints



# Appendix B

## Appendix for Chapter 4

In this appendix, we provide the 75 unit patterns we have created to help formulate our design principles for flexible kerf patterns (Section 4.4). As described in Figure 4.6, we have sized all unit patterns to have the same dimension ( $3.5 \times 3.5 \times 0.03$  cm) and assigned them properties of birch plywood (Young's modulus  $E = 8.0 \times 10^9$  N/m<sup>2</sup> and Poisson's ratio  $\nu = 0.3$ ), a common (rigid) fabrication material for laser cutters and CNC machines.

For each unit kerf pattern, we highlight its geodesic in red and label its curvature due to 40 N of bending force under its name. To accentuate the curvature differences between unit patterns, we tile each pattern on a  $5 \times 3$  grid, along the  $x$ - and  $y$ -axis respectively, and visualize the bending behavior of the resulting kerf pattern.

As discussed in Section 4.5.2, we have significantly increased the flexibility levels of 16 initially inflexible unit kerf patterns by redesigning their geometries. We first show these 16 pairs of inflexible and flexible unit patterns (32 in total) before providing the remaining 43 unit patterns. Please zoom in to see the geometric details of the pattern designs.

### B.1 Gallery of 75 Unit Kerf Patterns

