Uniquely Shaped Spaces: Object-Driven Algorithmic Shelf Design and Fabrication

Deanna Gelosi
ATLAS Institute
University of Colorado Boulder
Boulder, Colorado, USA
deanna.gelosi@colorado.edu

Michael L. Rivera
ATLAS Institute and Department of Computer Science
University of Colorado Boulder
Boulder, Colorado, USA
mrivera@colorado.edu

Laura Devendorf
ATLAS Institute & Information Science
University of Colorado Boulder
Boulder, Colorado, USA
laura.devendorf@colorado.edu

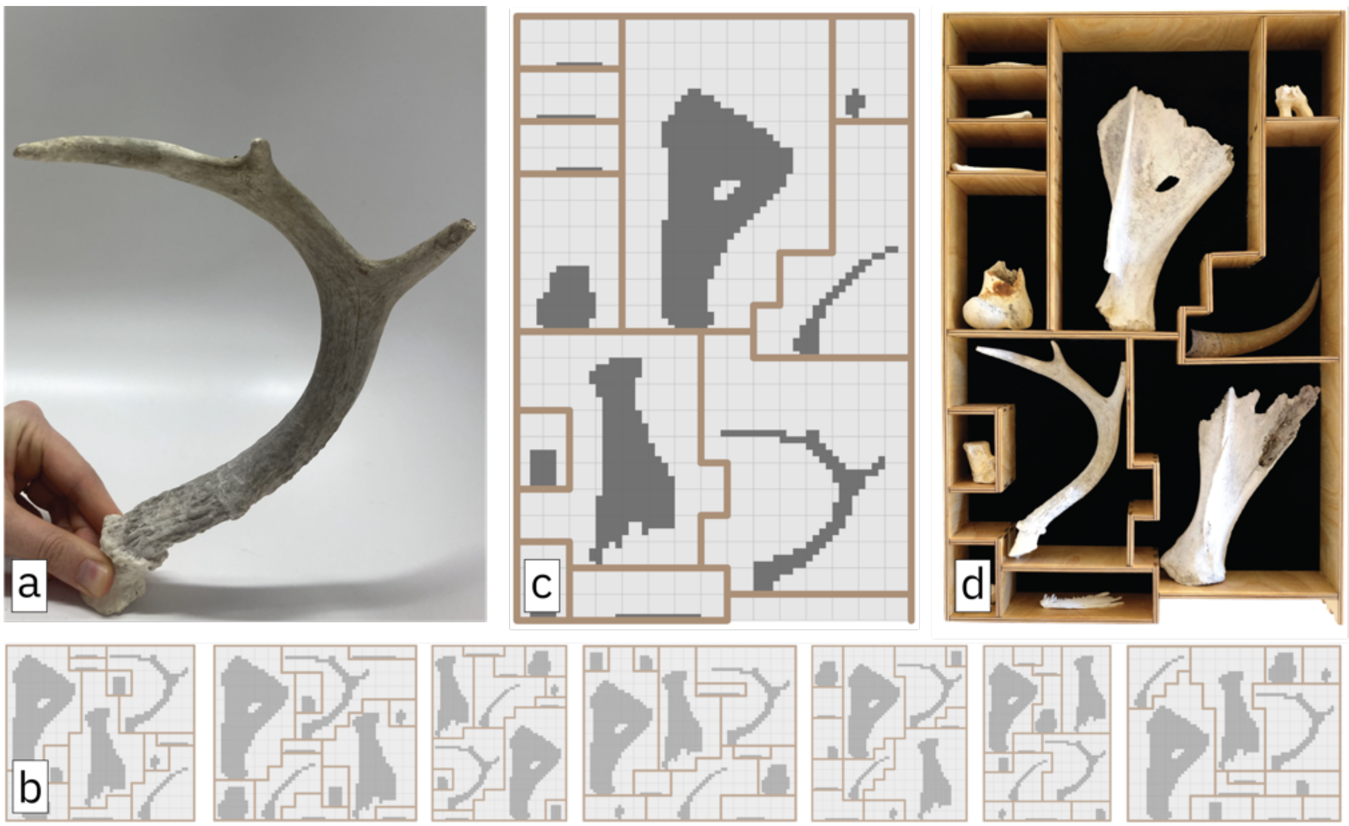

Figure 1: *Uniquely Shaped Spaces* generates shelving geometries that respond to the contours of everyday objects. For a participant's collection of bones: (a) she photographed each object; (b) generated and curated multiple layout options; (c) selected one for fabrication; (d) exported joinery-ready parts to assemble. Walls curve around bone silhouettes to capture their contours while balancing spacing between neighboring objects; this guarantees geometric fabricability, not structural engineering. Preferring a tighter visual fit, she masked her bone images slightly smaller to produce narrower voids; one bone fit less comfortably, leading her to swap two positions after assembly (see Sec. 5.1.3 for details).

CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/26/04
https://doi.org/10.1145/3772318.3791627

Abstract

Most shelving relies on rectangular compartments that ignore the contours of the objects they hold. We present *Uniquely Shaped Spaces*, an *object-driven* algorithmic tool for custom shelving generation. The workflow arranges users' object silhouettes with simulated annealing, grows walls via cellular automata to carve fitted voids, and outputs fabrication files with joinery for laser cutting. We designed the system so that objects, our algorithm, and users share authorship, and studied how this configuration played out with five participants as they designed shelves in guided workshops and then lived with the fabricated pieces. Our findings show how participants navigated object geometry, algorithmic search, and fabrication limits by curating, tweaking, and appropriating algorithmic proposals, and how the resulting shelves supported reflection and storytelling. These results point toward *object-driven* fabrication systems that foreground objects as generative constraints and explicitly support negotiation within constraint-driven workflows.

CCS Concepts

• **Human-centered computing → Interactive systems and tools**.

Keywords

Algorithmic Design, CAD, Layout Optimization, Shelving, Mixed-Initiative, Computational Fabrication

ACM Reference Format:

Deanna Gelosi, Michael L. Rivera, and Laura Devendorf. 2026. Uniquely Shaped Spaces: Object-Driven Algorithmic Shelf Design and Fabrication. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3772318.3791627

Figure 2: *Treasure* by Tung Chiang and Eddie Aya. Photographed by Derek Yarra.

1 Introduction

The practice of displaying meaningful objects through custom-built environments has deep cultural roots across civilizations. For millennia, Chinese traditions elevated object presentation through specialized display furniture, with Ming Dynasty collectors developing sophisticated practices where architectural framing became as important as the artifacts themselves [14]. Western traditions similarly celebrate object display through cabinets of curiosity (Wunderkammer) [40], while religious reliquaries position natural specimens, artistic works, and sacred items to create complex networks of meaning [20].

These historical practices continue to inspire contemporary approaches to object arrangement [10, 17, 18, 65]. In 2023, ceramic artist Tung Chiang created *Treasure* (Fig. 2) in homage to the Chinese tradition of building custom display shelves for unique artifacts. Collaborating with woodworker Eddie Aya, Chiang designed a piece in which each ceramic object determined the void that contains it, reversing the conventional logic of shelving as neutral support. *"In this case I started with the shelf. I designed it with* uniquely-shaped spaces *and created ten pieces to fit perfectly into those spaces"* [13].

Treasure offers more than aesthetic inspiration—it models a way of designing where existing artifacts dictate structure. Each form actively shapes the void around it, revealing a reciprocal relationship between object and container. We take this approach as the starting point for an *object-driven* stance in computational design: existing artifacts, rather than abstract primitives or designer-authored templates, define the constraints that a generative system must satisfy. Whereas many layout and fabrication tools for shelving begin from pre-parameterized casework, component grammars, or sketch-based outlines that are later populated with content [25, 27], we are interested in what happens when the objects themselves set the terms of the design problem. This *object-driven* stance raises a system-design question: **how can an *object-driven* generative shelving system take the geometry of users' own objects as a primary input while still producing manufacturable designs that reflect the idiosyncrasies of those artifacts?**

Motivated by this perspective and question, we developed *Uniquely Shaped Spaces*, a computational design tool that generates shelving around the silhouettes of users' own objects. We study it as a human-computer interaction (HCI) system: people specify a set of objects, watch stochastic layout search unfold, curate among algorithmically generated options, and commission fabrication. Our system uses a three-stage algorithmic workflow that takes the specific geometries of users' objects as primary inputs (Sec. 3). First, we employ simulated annealing [30] to optimize object layouts based on a multi-parameter objective function that evaluates spatial efficiency, object distribution, and structural validity (Sec. 4.2). Second, we use a cellular automata approach inspired by terrain-based pathfinding [67] to grow shelving structures around the optimized object arrangement (Sec. 4.3), producing organic yet manufacturable forms that create custom-fitted voids for each object. Third, we generate vector files with parametric joinery—including finger, mortise and tenon, and cross-lap joints—allowing direct fabrication via laser cutting (Sec. 4.4). Together, these stages instantiate a fabrication-oriented, *object-driven* pipeline in which object geometry constrains and shapes manufacturable shelving forms.

To understand the opportunities and challenges of *object-driven* design in use, we conducted a five-participant, twelve week study in which participants designed shelves in guided workshops and lived with their fabricated shelving for two weeks (Sec. 5.1). This study was motivated by a second, practice-centered question: **how do people work with such an *object-driven*, mixed-initiative system in practice, as object geometry, algorithmic search, and fabrication constraints jointly shape outcomes?** Drawing on this qualitative study, we find that as users work with the system, they often interpret its shelving designs as emerging from the objects themselves, a dynamic we describe as object agency, reflecting how people attribute agentic influence to the objects they are displaying.

For HCI and computational fabrication, we make the following contributions:

- A publicly accessible **algorithmic design tool**[1] for generating custom shelving driven by object arrangements, using a three-stage approach that builds on simulated annealing and cellular automata approaches;
- **Empirical findings** from a five-participant study that describe how people appropriate, negotiate, and adapt an *object-driven* fabrication tool in practice;
- **Design considerations** for *object-driven* generative fabrication systems that redistribute authorship between objects, algorithms, and users, and support user negotiation within constraint-driven design spaces.

2 Related Work

Uniquely Shaped Spaces draws on prior work in subtractive fabrication, flat-pack furniture design, mixed-initiative systems, and algorithmic aesthetics. Across these areas, we build on established constraint-aware workflows while extending them toward *object-driven* layout generation. We review each strand below.

2.1 Subtractive Fabrication

Across HCI and computational fabrication, subtractive pipelines converge on a representation-first norm: the design representation encodes the same constraints the fabrication process will later enforce. This stance appears in (1) mechanisms tuned to cutter behavior so assemblies tolerate kerf and machine variation [52, 54], (2) workflows that accelerate nesting and round-trip edits between sheets and assemblies [1, 53], (3) alternative laser and CNC pipelines that trade geometric generality for speed or reduced ambiguity by making machining intent explicit on material [8, 23, 45, 46], and (4) textile workflows that optimize patchwork and garment patterns for cut-based fabrication by embedding scrap availability, material behavior, and assembly constraints directly into the design representation [31, 35, 36, 38, 44]. Taken together, these norms keep constraints legible and manipulable so small edits do not compromise fabricability.

Our work draws direct inspiration from subtractive fabrication research to inform *Uniquely Shaped Spaces*, while also exploring how centering existing objects—items with personal meaning or history—reshapes the possibilities for computational design. *Carpentry Compiler* encodes joinery semantics and compiles to tool-specific operations, enforcing fabrication constraints at design time [68]. Plan-first editors *FlatFitFab* and *CutCAD* enforce thickness and joint semantics in the representation, keeping local geometric changes fabrication-valid [24, 43]. *Kerfmeter* closes the loop by making per-machine kerf an empirical input to geometry [29]. *Uniquely Shaped Spaces* extends these representation-first accounts by shifting the source of geometric constraint from designer-authored components to the objects being displayed: user-provided silhouettes become the basis for layout, subdivision, and joinery, while kerf- and thickness-aware export maintains manufacturability. In doing so, the tool advances subtractive workflows toward *object-driven* layout generation, where heterogeneous, existing objects determine the geometry that fabrication constraints govern.

[1] https://deannagelosi.github.io/shelving/

2.2 Home Furniture Design Tools

Across flat-pack furniture, a recurring pattern pairs non-expert authoring with fabrication-aware constraints across three domains to turn ideas into algorithmically-generated, cut-ready parts: (1) open design ecosystems make it easy to share ready-to-build flat-pack furniture plans for distributed making, using tab-and-slot joinery rules [19] and patterns that adapt to available sheet stock [50], (2) commercial pipelines reduce entry barriers by templating casework and producing kerf-aware, cut-ready exports [25, 27], and (3) research on material optimization reduces waste in flat-stock furniture by reshaping and nesting parts on sheets [32]. These approaches demonstrate how constraint-aware representations can guide non-experts toward buildable designs. In our work, we focus on the elements users manipulate and the interactive steps they take to arrange them, ensuring that generated designs stay feasible to build in-home shops and makerspaces.

Select furniture design and fabrication systems inform our approach. *Kyub* shows how plate-based kits with constraint checks scaffold non-expert authoring without compromising buildability [7]. Lau et al. systematize decomposition of 3D furniture into fabricable panels with explicit connection strategies [34]. Schwartzburg and Pauly formalize planar-piece constraints (thickness, orthogonality, assemblability) that keep edits valid across the design-to-fabrication pipeline [55]. Complementing feasibility, Yu et al. cast interior layout as multi-objective optimization, emphasizing functional fit in domestic settings [69]. While these systems begin from designer-authored furniture geometry or predetermined shelving forms, *Uniquely Shaped Spaces* reverses the direction of dependency: it treats object silhouettes as hard constraints at authoring time and, through user-steerable search, derives shelving geometry from the spatial relationships among the objects themselves. Rather than fully automating layout, the system surfaces multiple generative proposals that users choose between. Algorithms generate the internal subdivision, and whenever the layout changes, the system automatically regenerates walls and joinery to keep edits fabricable. Exports encode woodworking joinery semantics, explicit kerf, and stock sheet sizes for home shop fabrication, extending flat-pack workflows to *object-driven*, shelving-specific design.

2.3 Mixed-Initiative Computational Design Systems

Uniquely Shaped Spaces is designed as a mixed-initiative tool that frames shelving design as a negotiation between fabrication requirements and desires for personal object curation. Mixed-initiative systems stage negotiated control: rather than relying solely on direct manipulation or full automation, such systems choreograph turn-taking through generation, evaluation, and execution. We follow classic treatments that define mixed initiative as shared control with explicit transfers of initiative between user and system [26]. HCI researchers have created systems that keep human steering "in-the-loop" during making via bounded, real-time control of fabrication processes through live, constraint-aware milling [66], sensor-bounded freehand carving [72], haptic redirection [59], robot-assisted craft production [60], and in-situ material-aware assembly workflows that respond to irregular scrap geometries [28].

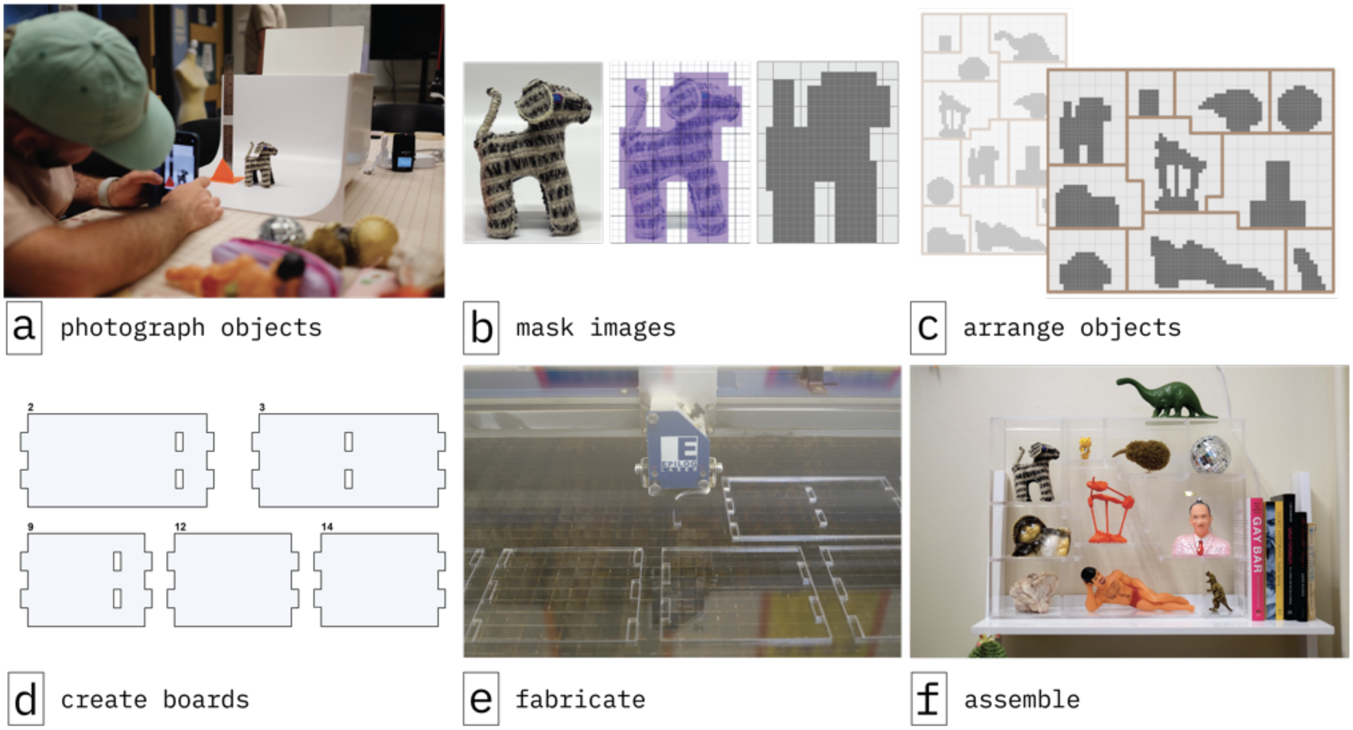

Figure 3: User Workflow. (a) Take photos of each object at the same scale. (b) Create a mask of each object on the quarter inch grid, making adjustments to the pre-generated mask by clicking individual cells. (c) Generate shelving layouts with objects and select the design to fabricate. (d) Export the selected design as a DXF file with generated joinery, finger joints on corners and mortise and tenon joints at perpendicular intersections. (e) Cut each board with a laser cutter. (f) Assemble shelving and place objects within each void.

Within design research, mixed-initiative tools have been read through lenses of craft and material speculation. Material speculation reframes artifacts as co-respondents rather than passive targets [64], while post-anthropocentric fabrication tools such as *Being the Machine* frame digital fabrication as a dialogue that "speaks back," making nonhuman actors consequential in making processes [15, 16]. Craft qualities such as care, skill, and appropriateness offer criteria for evaluating computational tools [6]. We draw on these perspectives in treating *Uniquely Shaped Spaces* as supporting situated judgment, where people respond to and refine algorithmic proposals in relation to the objects they care about.

Uniquely Shaped Spaces extends this landscape by locating negotiation upstream of fabrication and treating object geometry as a primary design input. Systems such as *Medley* [11] and *Forte* [12] couple generative exploration with fabrication-aware structures by offering a library of parameterized embeddables whose geometries encode distinct mechanical responses and staging a back-and-forth between generative layouts and user steering for laser-cut structures. Our system follows this line by tying layout generation to user judgment: each round of algorithmic layout search produces candidates that users assess and select from, and those choices directly determine which design advances to fabrication. We preserve feasibility via constraint-aware recomputation, akin to Larsson et al.'s human-in-the-loop builds with non-standardized branches [33]. In our workflow, initiative is staged across phases: users specify the set of objects (and masks) and later the fabrication parameters; the system arranges silhouettes and grows walls into fabrication-valid shelving; users then curate by regenerating, bookmarking, and selecting a candidate for fabrication. In sum, *Uniquely Shaped Spaces* frames mixed initiative as negotiated among users, algorithms, and objects: objects set constraints, algorithms derive shelving geometry, and people decide what to fabricate.

2.4 Algorithmic Aesthetics

We designed *Uniquely Shaped Spaces* in part as an exploration of algorithmic aesthetics for shelving design: instead of directly modeling a final piece of furniture, we encode procedures whose behavior becomes visible in families of related shelves. Algorithmic aesthetics in interface and media art treats rules and programs as a medium, surfacing how parameterized systems can generate series of outcomes for human judgment. Interfaces can make large parameter spaces easier to explore by grouping similar designs and showing how changes in parameters affect the resulting shapes [42, 58]. Foundational accounts situate these practices within longer traditions of formalizing aesthetic choice [9, 21, 47], and interactive artworks show how motion and response make those rules legible in use [37, 41].

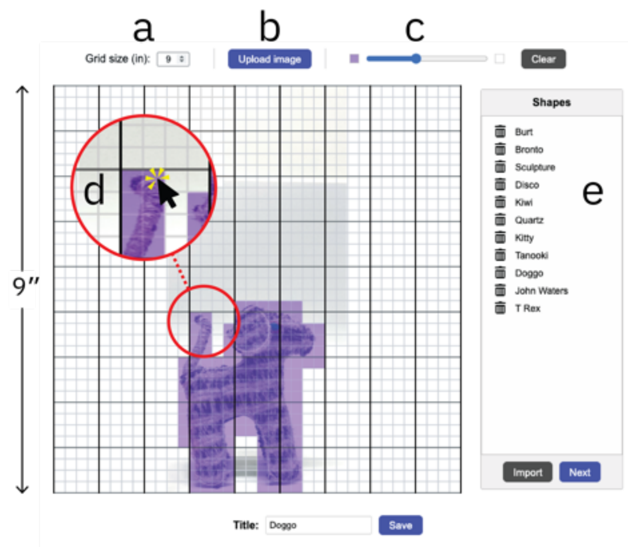

Figure 4: Object Input. (a) Users set the grid dimensions for the input screen to keep scale consistent across images. (b) They import photos into the tool, which places each object on a 2D grid for masking. (c) A coarse automatic slider provides an initial mask, and (d) users fine-tune individual grid squares for precision. The selection grid is at a quarter-inch scale. (e) Saved shapes appear in a sidebar list for later layout generation.

Our approach to algorithmic aesthetics emerges from silhouette-bounded generation and curated selection. Rather than exploring an unconstrained manifold of shelves, users bound the generative space by choosing and masking objects; the system then proposes manufacturable layouts whose silhouettes and wall patterns reflect the behavior of simulated annealing and cellular automata. The aim is not full automation, but to surface a small set of distinctive candidate designs to recognize and evaluate as shelves for their own objects. Deterministic wall growth realizes each chosen composition under explicit thickness, joinery, and kerf semantics with fabrication-ready export. In this way, our pipeline contributes an object-defined generative space in which every candidate is both visually legible and immediately fabricable, extending algorithmic aesthetics toward fabrication-constrained, artifact-driven form generation.

3 System Walkthrough

In this section, we describe how a user interacts with *Uniquely Shaped Spaces* from first photographing objects to exporting cut files for fabrication (Fig. 3). We ground the walkthrough in a usage scenario: creating custom shelving for a collection of personal objects.

Figure 5: Layout Generation Animation. A sequence of stills from one simulated annealing run shows how the shelf layout evolves over eight seconds (gray object silhouettes). Early layouts with collision penalties (red) converge to a high-scoring, feasible layout (green). The final panel satisfies the tool's constraints, making the design ready for export.

3.1 Object Input

The user begins the design process by gathering objects they wish to display in their custom shelving unit. These may include personal mementos, collected artifacts, or everyday items of interest. They photograph each object against a white background, including a ruler in frame to ensure scale proportionality between objects. Before importing images, they manually set the grid size based on the maximum height and width of the objects (Fig. 4a). This keeps all objects scaled consistently and ensures that the resulting shelving dimensions correspond to the physical collection. The user then imports each photograph into the web interface by clicking the "Upload image" button (Fig. 4b), which displays the image within the masking view.

The tool provides both automatic and manual options for creating object masks. Using the auto-masking slider (Fig. 4c), the user adjusts a threshold until the object's outline is approximately captured. For greater precision, they can click individual grid cells to refine the mask (Fig. 4d), creating an accurate discretized representation of their object on the quarter-inch grid. After saving each masked object with a name, it appears in a collection list of saved shapes (Fig. 4e). By working with these simplified digital shadows rather than full 3D models, we aimed to keep object input lightweight while preserving the essential characteristics needed for custom void generation. We discuss the limitations of this approach in Section 6.3.

3.2 Shelving Generation

The user next selects which objects to include in their shelving design. Each imported object appears in a selectable list where items can be toggled on or off by clicking their names. This selection capability allows users to experiment with different subsets of their collection, exploring how various combinations might work together in a unified shelving structure and refining the set of objects they plan to display.

Once satisfied with their object selection, the user initiates layout generation. After clicking the "Generate" button, the tool begins a layout search that the user can observe through an animated

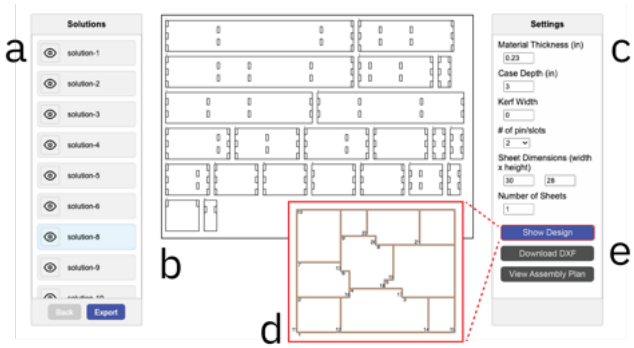

Figure 6: Shelving Export. (a) The saved layouts for selection and export. (b) Generated boards with joinery dependent on board intersection type. (c) User-defined export parameters, including board depth and material thickness. (d) Users can preview shelving design with boards numbered for assembly. (e) The DXF file export for fabrication.

visualization (Fig. 5). Objects move across the screen, swapping positions and rearranging themselves as the system explores different candidate arrangements. The animation provides transparency into what might otherwise feel like a black-box process. Users see the layout gradually transition from clearly invalid configurations—where objects overlap or span a large area—toward more compact arrangements.

When the search converges on an acceptable configuration, the animation stops and the tool presents the resulting layout and automatically grows shelving structures between and around the objects. Users can generate multiple layouts by clicking "Generate" again and save any that feel promising. Saved layouts appear in the *Results* stack on the right, where users can preview, delete, and reselect alternative solutions as they compare options. Users can save many promising layouts at once, switching between them to compare compositions and negotiate which arrangement best suited their objects.

3.3 Shelving Export and Fabrication

All user-saved layouts are available in the "Solutions" panel for export (Fig. 6a), and each selected layout generates a set of boards for fabrication (Fig. 6b). These boards automatically include joint patterns that depend on board intersection type.

Before exporting their design, users can specify fabrication parameters that affect the physical construction of their shelving unit (Fig. 6c). The primary adjustment is material thickness, which must account for both structural requirements and the capabilities of the fabrication equipment. Users select a thickness appropriate for their chosen material, and the tool uses this value when calculating joint dimensions. Users can also adjust the depth of each board, determining how far the shelving will extend from the wall or surface on which it is mounted based on the maximum depth of all objects. Additional fabrication values, such as laser kerf, the number of pins and slots per board, and material sheet dimensions, can be fine-tuned to accommodate specific machines or material sizes.

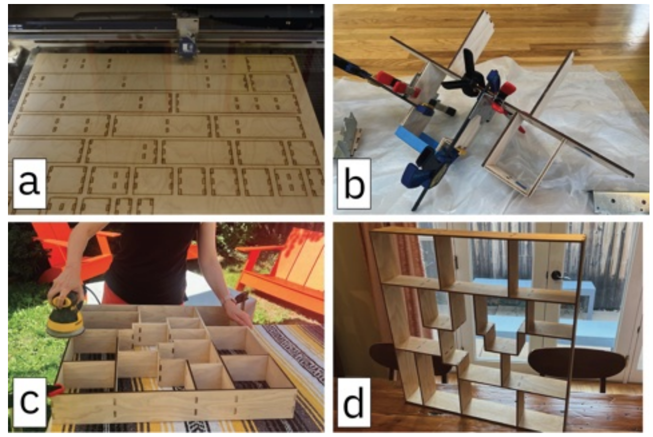

Figure 7: Fabrication and Assembly. (a) Shelves were cut on a laser cutter and (b) plywood shelves were secured with wood glue and clamped while drying in smaller subsections before gluing the entire shelving together. (c) Final finishes included sanding cut edges and finishing with an oil-based sealant before (d) the final shelving was installed.

With fabrication parameters set, the user can preview their assembled shelving by selecting "Show Design" (Fig. 6d). This visualization shows how the final shelving unit will look when constructed, with each board shown in its position and numbered to support assembly. When satisfied, the user exports a DXF file that contains every generated board with the appropriate joinery (Fig. 6e).

These files can be sent to a laser cutter to etch and cut each board (Fig. 7a). Boards can then be glued together and clamped while drying (Fig. 7b). Through fabrication trials, we found it worked best to assemble the shelving in stages rather than all pieces simultaneously. After being fully assembled, the plywood shelves are sanded to remove burnt edges left from the laser cutter (Fig. 7c) and sealed with an oil-based finish (Fig. 7d).

4 Implementation

Uniquely Shaped Spaces is implemented as a browser-based JavaScript application backed by a pipeline for layout optimization, wall generation, and fabrication export. This section outlines the computational logic of our *object-driven* shelving system, showing how 2D object silhouettes serve as primary input to generate manufacturable designs for small-to-medium wall-mounted or tabletop shelves for lightweight collections (e.g., ceramics, mementos). We focus on the conceptual architecture and design reasoning that balance generative flexibility with fabrication constraints; full pseudocode, parameter settings, and algorithmic details appear in the Appendix.

4.1 Shape Representation

Each object is modeled as a 2D shape within a quarter-inch grid using a multi-resolution approach: high-resolution data aids layout generation, while lower-resolution data supports fabrication. Specifically, each shape stores a quarter-inch silhouette mask, a one-inch rounded occupancy map, and a one-inch buffer derived

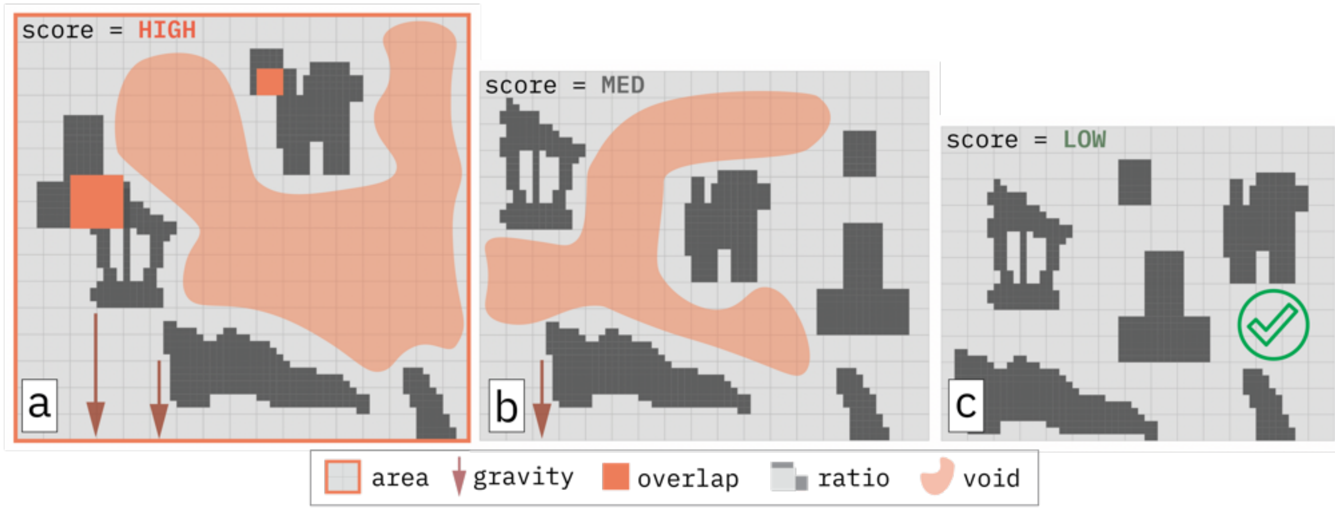

Figure 8: Simulated Annealing. The progression from HIGH (left) to LOW (right) layout scores illustrates how the algorithm iteratively reduces constraint violations. In layouts (a) and (b), the annotated areas show issues such as overlapping shapes, excessive void space, and unsupported shapes. The final layout (c) is a valid solution with no overlaps, complete bottom support, and minimal aesthetic penalties.

from the original mask. During pre-processing, the tool trims empty rows and columns from the silhouette and pads the result so that the quarter-inch grid aligns cleanly with the one-inch lattice used for layout. The one-inch occupancy map is created by aggregating each 4×4 block of quarter-inch cells and marking it occupied if any underlying cell is filled—a conservative rule that preserves irregular contours while reducing how many layout possibilities the algorithm has to consider. The one-inch buffer is produced by expanding the silhouette outward by one inch on the top and sides, ensuring sufficient clearance for wall thickness, joinery, and kerf during fabrication. Layout optimization and wall generation operate on a one-inch grid using buffered shapes, so every board segment spans at least one inch and the resulting shelving remains fabricable.

4.2 Layout Optimization

Uniquely Shaped Spaces arranges objects on the one-inch occupancy grid using simulated annealing (SA) [30] to produce multiple acceptable, manufacturable layouts with distinct spatial qualities that users can browse and curate, leveraging SA's stochastic search to yield diverse high-quality solutions across independent runs. Our implementation follows standard SA practice but adopts three mechanisms shown to support broader exploration in prior work: a temperature-dependent movement range [48], stagnation-triggered reheating [22], and a multi-start scheme [70]. These additions help the search avoid premature convergence and surface varied, fabrication-valid arrangements.

4.2.1 Objective Function. To evaluate each candidate layout ω, we use a weighted objective $\sigma(\omega)$ that combines five terms O, V, G, A, and R reflecting manufacturability, packing efficiency, and coarse aesthetic balance (Fig. 8). The overlap term O penalizes intersections between shapes, ensuring the final configuration is fabricable. The void term V penalizes large unused regions, encouraging more efficient packing and reducing unsupported spans. The grounding term G penalizes bottom-row objects that do not make contact with the shelving base, serving as a simple load-path heuristic. The area term A reflects the overall footprint of the shelving, favoring compact arrangements, and the ratio term R discourages extreme aspect ratios that yield visually awkward or structurally fragile forms.

The optimizer evaluates these terms and ranks physically valid layouts according to their combined score. Weight settings for each component were determined empirically through iterative testing. As the search proceeds, users may save any valid configuration encountered, enabling comparison among multiple workable alternatives.

4.2.2 Movement Operations and Termination Criteria. To explore the layout space, each iteration perturbs the current configuration by shifting a randomly selected object along one of eight cardinal or diagonal directions on the one-inch grid, or by swapping the positions of two objects. Move magnitudes scale with temperature, meaning that early in the search the optimizer makes broad, multi-cell adjustments, while later iterations focus on fine refinements. This stochastic neighborhood structure helps the search escape deterministic patterns and surface diverse arrangements. An annealing run ends when the temperature cools below a minimum threshold, when the iteration budget is reached, or when the user stops the process. Rather than terminating automatically when progress stalls, the optimizer reheats after prolonged non-improvement, allowing it to escape local minima and continue exploring distinct layouts. The best valid configuration encountered during a run proceeds to shelving generation, and users may generate additional candidates if desired.

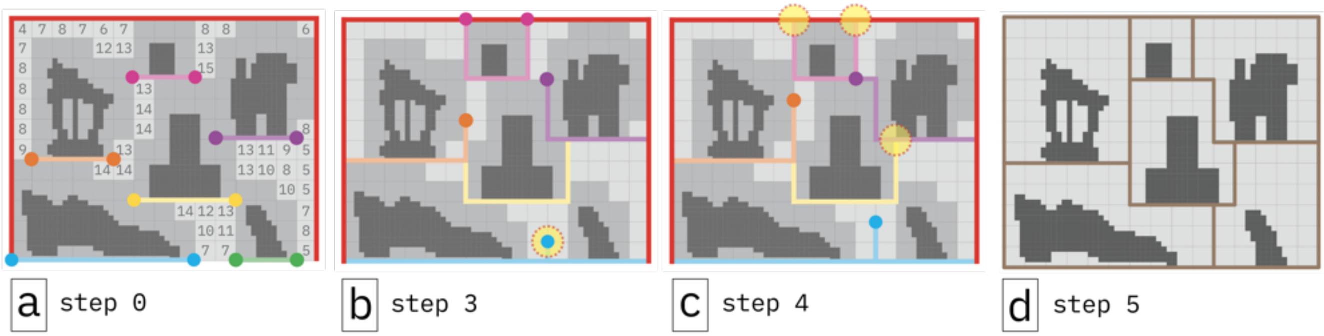

Figure 9: Cellular Automata. Selected snapshots from one run (intermediate steps omitted; left to right). (a) The tool populates inactive cells at the perimeter (red border) and bottoms of each shape (colored lines); active cells start at the ends of each bottom shelf (colored circles). (b) After three steps, two active cells meet and merge. (c) Three active cells meet inactive cells and die. (d) The process continues until all cells are inactive (five steps for this layout).

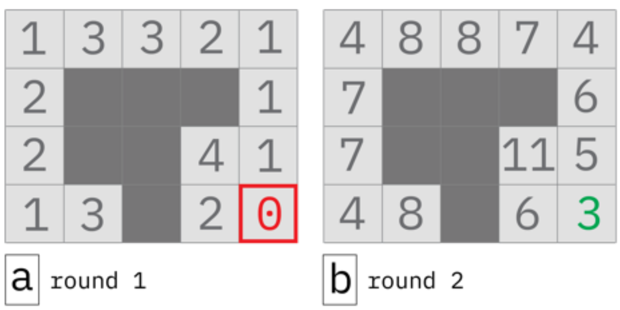

Figure 10: Terrain Scoring. (a) Cells with a score of 0 (e.g., bottom-right) represent unconnected areas. (b) The process continues until all such zero-valued cells are resolved, ensuring full connectivity across the terrain.

4.3 Shelving Generation

Once a valid layout is chosen, a cellular automaton (CA) [67] generates interior walls that divide the shelving unit into object-specific compartments. This subdivision must reflect object irregularities, maintain manufacturable orthogonal geometry, and satisfy fabrication and assembly constraints. Classical spatial subdivision techniques—such as Voronoi partitions [62] and medial axes computed with straight-skeleton methods [2]—prioritize geometric regularity, producing partitions at arbitrary angles and often missing the deep concavities and protrusions of real objects. In contrast, a grid-based growth model guarantees perpendicular intersections while allowing walls to "contour" to object silhouettes, offering both directional control and surface sensitivity.

4.3.1 Terrain Scoring and Cellular Growth. To make cellular growth responsive to object geometry, the system first assigns a terrain score to every grid square in the selected layout through an iterative neighbor-counting process. In the first round (Fig. 10a), each empty grid square counts how many of its eight neighbors (including diagonals) make contact with the shape, leaving any remaining grid squares with a score of 0. If there are any squares with a score of 0, all squares participate in another round; in subsequent passes, each grid square counts how many neighbors have non-zero terrain values (Fig. 10b). We repeat this propagation until all grid squares have non-zero scores, resulting in a connectivity gradient around objects.

From this terrain, the system then derives path values along grid edges between adjacent squares. These path values are what the CA use during wall growth to decide where to move next. Edges that lie between squares belonging to different shapes are assigned the minimum possible value, forming "valleys" that encourage walls to grow between objects rather than through them. Edges between squares associated with the same shape instead take on the average of their two terrain scores, producing higher "mountains" where objects sit and moderate slopes through open space.

After assigning terrain and path values, the system initializes the CA that will form the walls of the shelves. Inactive cells are placed along the perimeter and at the bottoms of the objects to define the boundary geometry (Fig. 9a). Active cells, each assigned a unique "strain," are positioned at the endpoints of these bottom supports and begin expanding across the terrain one step at a time. At each iteration, every strain selects its next move by comparing the path values of its neighboring edges, advancing toward locally minimal cost and merging when strains meet (Fig. 9b and 9c). Strains terminate when reaching an inactive cell or when entering buffer regions around objects; once all cells have become inactive, growth is complete and the system displays the resulting case design as the post-animation output (Fig. 9d).

4.3.2 Recursive Opportunity Scoring. While terrain and path values guide growth locally, some regions of a layout contain narrow passages or shallow concavities where a purely greedy step would trap a strain in a dead end. To address this, we incorporate a bounded spatial look-ahead that estimates the quality of several future steps before committing to the next move. Rather than choosing the single best adjacent edge, each active cell evaluates a short sequence of possible moves—up to k steps, with k=4 in our implementation—computing an opportunity score that reflects the cumulative cost of advancing along that micro-path.

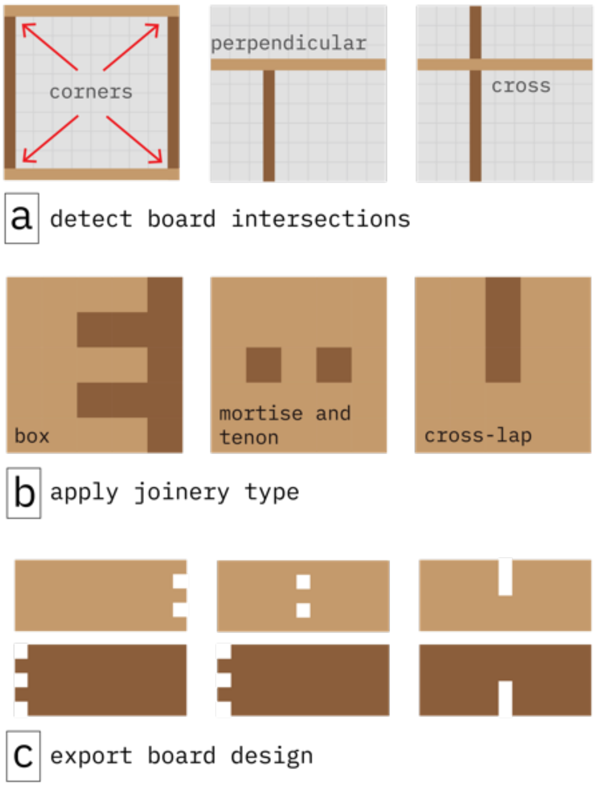

Figure 11: Parametric Joinery. (a) We detect board intersections and categorize them as corners, perpendicular, or cross. (b) A joinery type is applied to each intersection: box, mortise and tenon, and cross-lap, respectively. (c) These joinery types are used to generate exported boards.

Each active cell assigns an opportunity score $O(x, y, d)$ to stepping from cell (x, y) in direction d. This score accumulates the immediate path cost for taking the current step with the best projected cost over the remaining look-ahead, exploring forward moves (left, up, and right) from the resulting positions. Invalid moves incur a large penalty, and deterministic tie-breaking ensures repeatable growth. Because the opportunity scores are recomputed at every iteration from the current grid state, the look-ahead remains local and lightweight: each cell considers only its own potential trajectories and does not attempt to predict other strains' behavior. Instead, it detects when a seemingly attractive adjacent step would quickly lead into a dead end with no valid continuation and redirects the strain toward an alternate path, improving overall connectivity and reducing incomplete or invalid compartment formations.

4.4 Parametric Joinery

After the CA algorithm completes shelving growth, the tool converts the resulting wall layout into individual boards with parametric joinery for fabrication. As in prior computational fabrication systems that prioritize manufacturable geometry over load bearing simulation [7, 43, 68], this iteration of *Uniquely Shaped Spaces* focuses on geometric fabricability—objects do not overlap, walls form a connected network that partitions the shelving region, and board intersections map to joinery types that can be cut from planar stock. The pipeline guarantees these geometric constraints but does not estimate stiffness or mechanical safety under load.

The tool first analyzes the CA output to identify horizontal and vertical wall segments, grouping contiguous cell lines (constant y for horizontal, constant x for vertical), and merging adjacent segments on the same axis into continuous boards. Each board is assigned a unique identifier, orientation, and coordinate frame anchored at its smaller coordinate value, ensuring deterministic joint detection and simplifying downstream fabrication logic. We identify three types of intersections on boards—corner, perpendicular, and crossing (Fig. 11a)—which are mapped to joinery strategies: box joints at corners, mortise and tenon joints where one board meets the face of another, and cross-lap joints where boards cross (Fig. 11b). These relationships are recorded to track both joint type and position and parameterizes each feature using user-specified material thickness and shelving depth.

Lastly, the pipeline converts parameterized boards and joints into DXF files (Fig. 11c) suitable for laser cutters and related equipment [5]. The DXF export includes three layers: (1) sheet outlines defining material boundaries; (2) cut paths containing all board and joint geometry with kerf compensation applied; and (3) labels that number each board to support assembly. A board-packing algorithm arranges parts across one or more sheets to maximize material efficiency within user-defined sheet dimensions.

5 Evaluation

We evaluated *Uniquely Shaped Spaces* using two complementary approaches. First, we conducted a twelve week, in-situ user study with five participants to examine how object geometry, algorithmic search, and fabrication constraints jointly shaped shelving designs and how people worked with and responded to the tool from initial object selection through living with the fabricated shelves. Because *Uniquely Shaped Spaces* is mixed-initiative, we focused on user experience and the algorithm's role, adopting a small-N, interpretivist study design centered on rich accounts of use rather than summative measures of performance or efficiency.

Second, drawing on the object masks created in the study, we ran an algorithmic stress test to systematically probe how consistently the pipeline produced geometrically fabricable shelves across a broader range of object combinations. Together, these analyses show that *object-driven* design invited rich engagement, ongoing curation, and negotiated control between users, their objects, and the algorithm, and that the pipeline reliably generated geometrically valid shelving across thousands of randomized trials while still exposing rare corner-case failures that reveal the system's limits.

5.1 User Study

We conducted a twelve week, in-situ study with five participants recruited from the first author's professional network, following established practices in qualitative fabrication research where depth

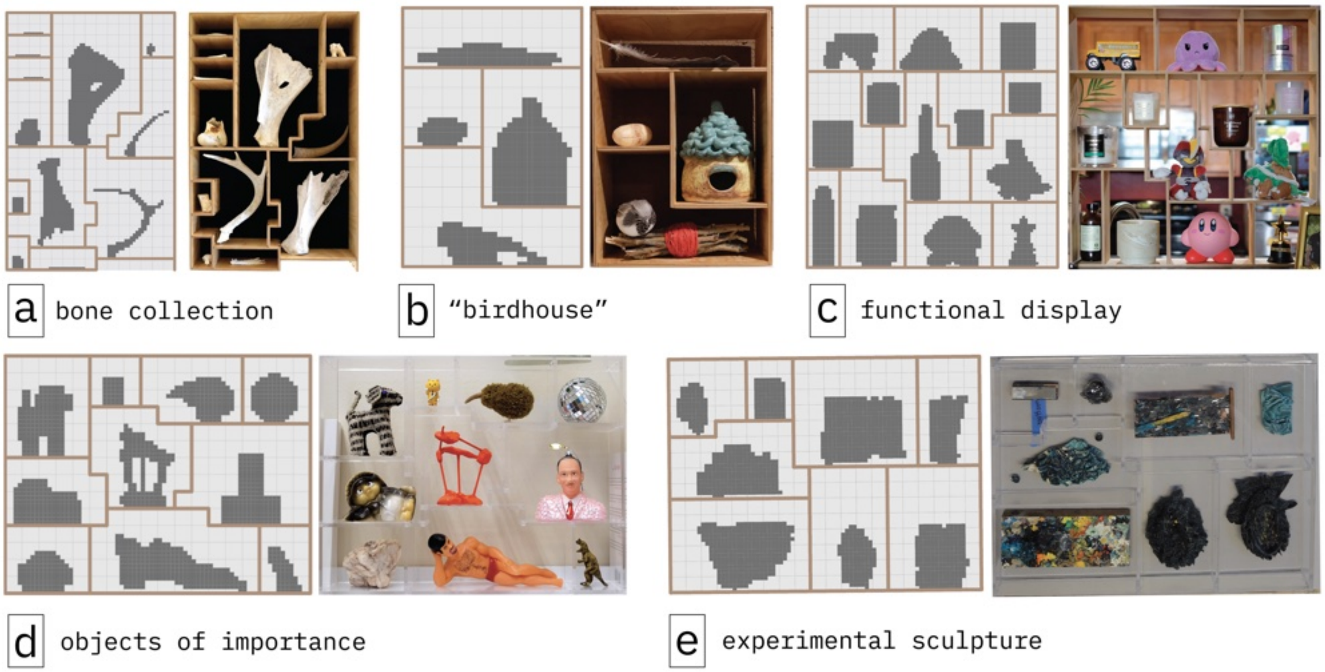

Figure 12: User Study. (a) Maria used her collection of bones that she found while hiking. (b) Sophia brought a collection of items that held special meaning around the words "birdhouse" and "home." (c) Quinn focused on functionality of housing both sentimental objects alongside everyday items. (d) Drew brought objects of importance to the design workshop, noting many were collected during travel or gifted from friends and loved ones. (e) Kai explored an experimental sculpture for engaging with ceramic shards inspired by works from artist David Altmejd [4, 39]. Photos show shelves after two weeks of use, during which all participants except Drew changed at least one object position from the original generated layout.

of engagement in small-N studies yields richer insight than broader sampling [49, 56]. Participants ranged in age from 24–56 years and represented a mix of artistic and technical expertise relevant to working and designing with physical objects. Drew (they/them) was a media studies professor and artist; Kai (he/him) was a ceramicist and MFA student; Maria (she/her) was a mechanical engineer with CAD experience; Sophia (she/her) was a ceramic artist and teacher; and Quinn (they/them) was an information science PhD student. All names are pseudonyms. Participants received $200 in compensation for their time and effort and kept the shelves they designed. All study procedures were reviewed and approved by our university's Institutional Review Board with informed consent.

5.1.1 Study Design and Data Collection. After recruitment, participants met for an initial planning session to discuss the objects they intended to use and to understand the tool's basic capabilities. Then, participants attended a two hour design workshop (two workshops with two participants, one workshop with one participant), during which they created shape masks, explored layout variations, and selected a final layout for fabrication. After our team fabricated their shelving units, participants lived with them for at least two weeks and completed a semi-structured interview at their installation site and photographed their shelving installations (Fig. 12). Our research team handled fabrication because the study was designed to focus on how people engaged with the generative workflow and lived with the resulting shelves, rather than on their ability to fabricate or find time and resources to do so. This structure allowed us to observe both real-time engagement with the generative system and longer-term interactions with the resulting artifacts.

We collected multimodal data across the study: screen recordings of all tool interactions, video of object handling during workshops, audio recordings of participants' verbal reflections, written notes completed after design sessions, and semi-structured exit interviews conducted in each participant's home, studio, or office. Fabrication and deployment were staggered, as each shelving unit required 2–3 days for cutting, assembly, and finishing; participants began their two week in-situ period as soon as their individual unit was completed. Together, these rolling deployments spanned twelve weeks in total while allowing us to maintain detailed capture of each participant's process. Analysis followed an open coding approach informed by Strauss and Corbin's grounded theory methodology [57], using iterative comparison across workshop behaviors via screen recordings and workshop videos, artifact traces, and interview accounts.

5.1.2 Object Selection and Interactions. Participants arrived at the workshops with distinct collections of physical objects that mattered to them. Some brought long-accumulated collections from

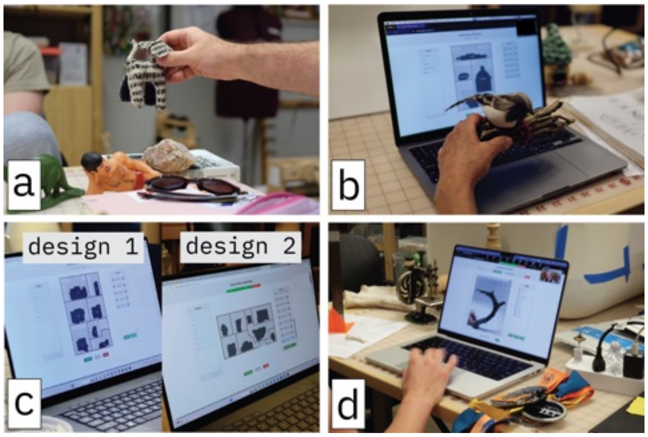

Figure 13: Object Interactions and Navigating Algorithmic Intent. (a) Drew positioned objects in space to imagine the generated shelving design. (b) Sophia held an object up to the interface while selecting shapes. (c) Kai compared two shelving designs in parallel—one with seven objects (left) and one with eight (right)—and recorded dimensions to decide how to stack his acrylic units. (d) Maria trimmed the masking on her shapes to reduce the overall size of the generated shelving.

everyday life and travel, such as small gifts, souvenirs, and keepsakes; others focused on more functional items they wanted to house alongside a few sentimental pieces. One participant explored a more speculative assemblage, using ceramic fragments and sculptural components as material for an experimental display. Taken together, these object sets mixed personal artifacts, practical objects, and curated collections, emphasizing how the tool needed to respond to the contours and meanings of specific things rather than abstract dimensions, and how these choices would later shape participants' interactions with the generative workflow.

During the workshop, several participants arranged their physical objects on the table to imagine the spatial feel of each configuration. Drew and Sophia, for instance, held objects in space relative to others to anticipate how compartments might frame them before committing to a generated layout (Fig. 13a-b). These off-screen tactics reflect known patterns in fabrication practice, where designers integrate material judgment and embodied cues when evaluating computational outputs [28, 60, 72], suggesting that *object-driven* systems may need to accommodate both on-screen and material modes of exploration. Echoing this material framing, Kai later reflected in his post-study interview that certain objects "wanted" particular positions within the shelving, interpreting the system's design proposals as the outcome of competing geometric influences rather than deliberate algorithmic choices. This anthropomorphic language highlights how participants at times experienced their objects as active agents in the generative process.

5.1.3 Navigating Algorithmic Intent. Participants approached our mixed-initiative tool through varying approaches of rapid generation to deliberate analysis. Drew and Quinn generated layouts in rapid succession to sweep across many possible configurations before committing to a final design. They treated each run as a probe into a different region of the design space, suggesting that they let the objects lead the design creation. In contrast, Sophia and Kai advanced more slowly, evaluating each configuration in detail before deciding whether to request new variations. They both had visions of what they wanted their shelving to look like, but not all of that design intent could be expressed to the system. Sophia wanted tall and narrow shelving for her birdhouse, but the objects she used for shelving generation tended toward short and wide designs. To address this, she removed one of her objects from layout generation and found a resulting design to fabricate. Kai explained that the system *"needed my interpretation"*, and set out to stack two shelving units flat on a surface to create a clear acrylic sculpture housing ceramic objects akin to David Altmejd's artwork *Flux* [18]. He imported two sets of objects into separate browser tabs, moving between the two to compare the generated designs' outer dimensions for stacking (Fig. 13c). Sophia and Kai's interaction patterns indicate that users can also take the lead in mixed-initiative systems, identifying opportunities to push back against *object-driven* design to assert agency over the resulting shelving design. They also highlight the need for stronger mechanisms to bookmark promising candidates and to express longer-term design intentions within the tool itself.

Maria also adopted a user-led stance, approaching the tool's underlying mechanism with curiosity and a desire to fully understand it. This diagnostic approach was sparked when she wanted to unpack why her thin and delicate bones often produced larger than expected voids; she explained in her final interview: *"It [Uniquely Shaped Spaces] did the job for sure, but there was the question:* 'Hmm, why'd you do that?'" To explain what was happening "under the hood," we showed Maria a diagnostic mode, not intended as part of the design workflow, that displayed the buffers around object masks and made the step-by-step cellular automata wall generation visible (Fig. 13d). With this information, she decided to reduce the size of her object masks in order to reduce the size of their resulting voids. This mostly achieved her intended effect, though one void was slightly smaller than she anticipated, which resulted in her swapping two bone positions upon receiving the shelving. This pattern shows that, for some users, *object-driven* generation becomes a way to reason about the system's internal rules, pointing to the value of exposing intermediate representations for debugging and better aligning system behavior with users' expectations.

Taken together, these contrasting approaches resonate with prior work on mixed-initiative interfaces and interactive fabrication tools, which emphasizes human steering and situated judgment in guiding algorithmic processes [12, 16, 26, 66, 72]. While participants welcomed a degree of algorithmic surprise from *Uniquely Shaped Spaces*, they also wanted the system to recognize and respect certain intentions, pointing to opportunities for future versions of the tool to surface such parameters as adjustable without undermining its generative character. These approaches to navigating generativity also show that participants treated algorithmic aesthetics less as an invitation to seek glitchy or arbitrarily strange voids and more as a way to browse families of plausible shelves that might fit their

Figure 14: Fabricated Shelves in Everyday Use. (a) When they no longer had a lint roller to fit a void in their design, Quinn placed a Pokémon plushie to fill the void. (b) Sophia moved her shelving around her home into three different rooms, but landed on putting it on her mantle alongside other art pieces. (c) At the time of our interview, Drew shared that the shelving had already sparked conversations with others who encountered it in their office. (d) Inside his studio, Kai shared his process as he continued working with the idea to make a multi-layer sculpture using the acrylic shelving.

objects and intentions—highlighting both the promise and the limits of *object-driven* design for generative fabrication.

5.1.4 Fabricated Shelves in Everyday Use. Through the workshops, all participants successfully generated shelving layouts that produced geometrically fabricable DXF exports, and each shelving unit was assembled without post-processing. The final units ranged from 12–30 inches and were fabricated in plywood or clear acrylic, depending on the preference of the participants. Because we fabricated and assembled each unit sequentially, participants received them on a rolling basis. Participants then installed their shelving in a variety of contexts—including home offices, studio spaces, and shared living areas (Fig. 14). The physical shelves matched their generated geometry and accommodated the objects selected during the workshops; later photos in this section reflect participants' own rearrangements during deployment rather than issues with fit.

Once installed, we observed the shelves becoming active sites of ongoing curation and reinterpretation. Participants described rotating objects through the compartments as their needs and interests shifted, adding new items over time, or relocating the shelving within their home to see how it altered the surrounding space. Quinn, for instance, replaced objects several times: *"Everything is special to me. But none of it is special enough or expensive enough that having that kind of shelving would have felt apropos. So I am excited for it in the sense that my belongings will continue to circulate around the house and I'll have to find stuff that fits neatly in each shelf"* (Fig. 14a). Sophia experimented both with where the shelving lived and with how her objects were distributed across its voids, moving pieces from one compartment to another before settling on a mantel installation (Fig. 14b). *"What surprised me was the desire to want to move the pieces around [physically]. I constantly wanted to rearrange it."* In Drew's case, the shelving prompted colleagues to ask about the displayed artifacts: *"I like that the objects' display allows a conversation to take place around them. And I love to think about how objects create social experiences"* (Fig. 14c). Kai used his acrylic shelving as a spark for a multi-layer sculpture, demonstrating how *object-driven* forms can extend creative inquiry beyond fabrication (Fig. 14d).

Across these examples, the full use of each shelving unit was under-specified at design time: the algorithmically derived voids acted less as permanent homes for particular objects and more as invitations for kinds of things that might later be rotated, added, or replaced. Participants' accounts suggested that the geometry of these voids mattered: less rectangular, more idiosyncratic compartments often had a more active "voice" in shaping what they chose to place where. For designers of *object-driven* systems, this raises questions about how to plan for unknown futures—whether by constraining use to relatively stable object sets (as in museum displays) or by explicitly treating shelving as evolving over time and considering how void geometry, reconfigurability, or adjustability might support ongoing reinterpretation. In this sense, *object-driven* computational design creates not only functional artifacts but also frameworks for continued creative engagement, where the constraints of algorithmically derived voids scaffold rather than fix future arrangements.

5.2 Algorithmic Stress Test

To complement the in-situ study and assess how reliably the pipeline performs beyond the five workshop sessions, we conducted an algorithmic stress test using all of the shape masks created by participants. After the study concluded, we compiled a dataset of 56 object silhouettes and ran 10,000 randomized trials, each drawing between 4 and 13 shapes per run, a range based on our smallest and largest number of selected objects by user study participants. For each trial, the system attempted a full pass through the three-stage pipeline—layout optimization, wall growth, and joinery generation—and we recorded whether it produced a valid shelving layout. In this test, we define "successful" as layouts that are geometrically fabricable under our pipeline, not structural load bearing guarantees.

Using this definition, the pipeline completed successfully in 9,913 of 10,000 trials (99.13%), generating layouts with non-overlapping objects, connected wall structures, and joinery-compatible intersections. The remaining 0.79% of trials failed when the cellular-automata wall growth became trapped in rare interior configurations that prevented the formation of a continuous shelving boundary. These edge cases did not occur in participant sessions but exposed specific geometric corner cases that we return to in Section 6 as opportunities for improved routing and recovery strategies. The stress test therefore supports the claim that the tool reliably produces valid geometry across a wide range of realistic object combinations, while also clarifying the limits of this guarantee.

6 Discussion

We examine how object geometry constrains the shelving forms the system can generate, and how people work with an *object-driven*, mixed-initiative system as they move from digital layouts toward fabricated pieces. Our study of *Uniquely Shaped Spaces* shows that taking object geometry as a primary input can still yield manufacturable shelving designs with rich variation and that this configuration makes redistributed authorship salient, opening up new forms of negotiation and improvisation as participants work with the system's layout proposals. Taken together, these results position *object-driven* generative fabrication as a way to make non-human constraints legible and negotiable in mixed-initiative tools and offer concrete patterns for future systems that orchestrate shared authorship across objects, algorithms, and users.

6.1 Objects as Generative Agents That Redistribute Design Authorship

Most computational design tools assume that form emerges from designer intent through direct manipulation, parametric constraints, or sketch-based input. *Uniquely Shaped Spaces* inverts this pattern by treating object geometry as the primary driver of the design space. This orientation toward object influence resonates with HCI scholarship on material agency and mixed-initiative design. Prior work demonstrates how materials shape design trajectories, highlighting reciprocal relationships between human intention and non-human influence [51, 61]. Other scholars argue that forms emerge through negotiation among human actors, computational processes, and material forces [63, 71]. *Uniquely Shaped Spaces* surfaces these dynamics computationally: object scale, curvature, and adjacency visibly limit and redirect the generative search process. Participants responded by reading these limitations not as algorithmic errors but as expressions of object agency.

Participants' sensemaking therefore centered on how their objects engaged with the generative system. Even when they experimented to infer system behavior—for instance, when Maria adjusted object sizes to see how void dimensions changed—she did so by manipulating material representations rather than by working from an abstract model of the algorithm. This dynamic highlights a concrete opportunity for future computational design systems: to treat objects and other non-human actors as first-class generative agents rather than background conditions to be satisfied. Instead of treating object geometry as a static input, we contend that future systems could render these constraints visible and negotiable, explicitly exposing how non-human agents steer computational trajectories and, in doing so, structurally redistribute authorship among user, algorithm, and material form. This stance not only extends but operationalizes ongoing HCI efforts to design tools that acknowledge non-human actors as participants in multi-initiative workflows [3, 11, 33]. Taken together, these systems and our findings articulate a design agenda for computational fabrication: surface non-human agents as an explicit, adjustable constraint; design interfaces to foreground how materials and other non-human actors as co-creators in generative behavior; and evaluate tools by how effectively they orchestrate shared authorship across objects, algorithms, and users.

6.2 Negotiation and Improvisation Within Generative Constraints

Generative design tools typically offer variation within a defined search space rather than producing a single predetermined outcome. *Uniquely Shaped Spaces* followed this pattern: each participant generated layouts that fell within the system's intended field of possibilities but did not always match the specific configuration they had in mind. If participants approached the system expecting an exact arrangement, the result was often not fully aligned with their vision. Rather than abandoning these outputs, participants treated them as starting points for negotiation. Kai, for instance, opened multiple browser windows in parallel to compare alternative layouts when the system could not produce his preferred stacked arrangement; Sophia, who arrived with a clear target geometry, found that non-deterministic layouts rarely converged on it and adjusted both her object choices and expectations. This improvisation extended into fabrication: Maria trimmed the input shapes of her bones to shrink their resulting shelving voids and adjusted which pieces she included so they would fit, adapting around the system's constraints across digital and material stages.

This pattern echoes broader observations in computational design and making: generative systems are most effective when designers seek inspiration or alternatives, not deterministic convergence. Prior work in improvisational craft and hybrid fabrication shows how designers work with computational outputs as suggestive structures that invite interpretation rather than precise blueprints [16, 28, 36, 72]. Small irregularities or unexpected arrangements become opportunities for situated judgment rather than barriers to completion. In this light, the frictions participants encountered with *Uniquely Shaped Spaces* were less signs of failure and more prompts for creative adjustment, mirroring material-centric practices where designers continually reconcile tool suggestions, material behavior, and evolving intent.

Extending this perspective suggests design opportunities for future constraint-driven generative systems. The *object-driven* constraints in *Uniquely Shaped Spaces*—derived from silhouettes, buffer regions, and wall growth behavior—defined the space of viable layouts while leaving room for user interpretation and adjustment. Making these constraints more legible and steerable could better support negotiation as a first-class interaction pattern. Mechanisms such as localized regeneration, partial constraint freezing, adjustable clearances, or lightweight direct manipulation of regions would allow users to refine outcomes where it matters most without collapsing exploratory variation. Designing for this style of improvisational negotiation positions generative tools not as engines for perfect convergence, but as partners that propose, users respond, and both are reshaped through ongoing interaction.

6.3 Limitations and Future Work

While *Uniquely Shaped Spaces* demonstrates how *object-driven* generativity can provoke engagement and reflection, its current formulation has several boundaries that matter for future systems. Our pipeline guarantees geometric fabricability but does not model load bearing performance. The generated shelves are manufacturable, not engineered for stiffness, which may limit adoption for larger or safety-critical installations. Our representational choices introduce related constraints: 2D silhouettes omit depth and surface detail, the global buffer cannot express per-object tolerances, and cellular-automata wall growth can still become trapped in rare interior configurations (0.79% of 10,000 stress test runs), with no user-facing recovery beyond regenerating a layout. Empirically, our study captures a single design-and-build pass with five participants and two weeks of in-home use; we do not yet know how generatively produced shelving behaves under longer term material wear, changing collections, or different fabrication contexts.

These limitations point to concrete opportunities for future *object-driven* generative shelving tools. On the system side, future work could combine the layout and wall generation pipelines with lightweight structural checks (e.g., maximum distance between supports, simple support path heuristics, or rules calibrated from empirical load tests) for constructing large, load bearing shelving; incorporate richer shape acquisition (2.5D or 3D capture, per object clearances) to allow shape rotation during layout generation; and add more robust routing with automated detection and recovery when growth becomes trapped. While our wall generation strategy can yield faceted or jagged interior partitions, we intentionally did not smooth or simplify these shapes in this iteration, treating expressive contouring as a design feature rather than optimizing for minimal part count; for larger, load bearing installations, future work could pair these generative strategies with additional structural heuristics or smoothing passes.

On the interaction side, our findings highlight how users negotiate constraints during design and continue to reinterpret them after fabrication; future systems can build on these insights by exposing constraint level controls and supporting incremental reconfiguration and add-on fabrication over time. Together, these directions treat the limitations of our prototype not as flaws to be hidden but as design challenges for more expressive, reliable, and negotiable *object-driven* generative fabrication systems.

7 Conclusion

We presented *Uniquely Shaped Spaces*, an *object-driven* algorithmic design tool for shelving layout and fabrication. The tool couples simulated annealing for layout optimization, cellular automata for shelving generation, and automated joinery and board creation to produce manufacturable shelving designs that center spatial relationships between artifacts, enabling arrangements rarely seen in conventional shelving. Our evaluative user study shows that taking object geometry as a primary constraint can still yield a wide range of feasible forms while keeping shelving designs fabricable. Participants treated their objects as generative agents and the system's proposals as prompts for negotiation and improvisation, using the resulting shelving as ongoing sites of curation, reflection, and storytelling. These findings point toward *object-driven*, constraint-aware generative systems that surface non-human constraints as first-class, support tactical flexibility within bounded search spaces, and deliberately redistribute design authorship across objects, algorithms, and users.

Acknowledgments

We thank Tung Chiang of Heath Ceramics for inspiring this project through *Treasure* (created in collaboration with Eddie Aya), and for granting permission to reproduce the *Treasure* photograph in this paper (photograph by Derek Yarra). We thank the Computational Fabrication course at CU Boulder taught by Dr. Michael Rivera for providing the space to initially explore this idea. We are grateful to Earl Wright and the ARCS Foundation for scholarship support in memory of Nancy Seacrest Wright. We also thank the Blow Things Up Lab in the ATLAS Institute and Colorado Plastics for their fabrication support. Finally, we extend our gratitude to our user study participants and our reviewers for their insightful feedback. This work was supported in part by the National Science Foundation under Grant No. 1943109.

References

[1] Muhammad Abdullah, Romeo Sommerfeld, Laurenz Seidel, Jonas Noack, Ran Zhang, Thijs Roumen, and Patrick Baudisch. 2021. Roadkill: Nesting Laser-Cut Objects for Fast Assembly. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 972–984. doi:10.1145/3472749.3474799

[2] Oswin Aichholzer, Franz Aurenhammer, David Alberts, and Bernd Gärtner. 1996. A Novel Type of Skeleton for Polygons. In *J.UCS The Journal of Universal Computer Science: Annual Print and CD-ROM Archive Edition Volume 1 • 1995*, Hermann Maurer, Cristian Calude, and Arto Salomaa (Eds.). Springer, Berlin, Heidelberg, 752–761. doi:10.1007/978-3-642-80350-5_65

[3] Lea Albaugh, Scott E. Hudson, Lining Yao, and Laura Devendorf. 2020. Investigating Underdetermination Through Interactive Computational Handweaving. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference (DIS '20)*. Association for Computing Machinery, New York, NY, USA, 1033–1046. doi:10.1145/3357236.3395538

[4] Art21. 2024. David Altmejd in "Boundaries" - Season 6 - "Art in the Twenty-First Century" | Art21. https://www.youtube.com/watch?v=Qqmvi16sqZo

[5] Autodesk. 2011. DXF Reference. https://images.autodesk.com/adsk/files/autocad_2012_pdf_dxf-reference_enu.pdf

[6] Shaowen Bardzell, Daniela K. Rosner, and Jeffrey Bardzell. 2012. Crafting quality in design: integrity, creativity, and public sensibility. In *Proceedings of the Designing Interactive Systems Conference (DIS '12)*. Association for Computing Machinery, New York, NY, USA, 11–20. doi:10.1145/2317956.2317959

[7] Patrick Baudisch, Arthur Silber, Yannis Kommana, Milan Gruner, Ludwig Wall, Kevin Reuss, Lukas Heilman, Robert Kovacs, Daniel Rechlitz, and Thijs Roumen. 2019. Kyub: A 3D Editor for Modeling Sturdy Laser-Cut Objects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3290605.3300796

[8] Dustin Beyer, Serafima Gurevich, Stefanie Mueller, Hsiang-Ting Chen, and Patrick Baudisch. 2015. Platener: Low-Fidelity Fabrication of 3D Objects by Substituting 3D Print with Laser-Cut Plates. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 1799–1806. doi:10.1145/2702123.2702225

[9] George David Birkhoff. 1933. *Aesthetic Measure.* Harvard University Press. Google-Books-ID: KPXWAAAAMAAJ.

[10] Lindsay F. Blair. 1991. *The working method of Joseph Cornell.* Open Library ID: OL47880441M.

[11] Xiang 'Anthony' Chen, Stelian Coros, and Scott E. Hudson. 2018. Medley: A Library of Embeddables to Explore Rich Material Properties for 3D Printed Objects. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3173574.3173736

[12] Xiang 'Anthony' Chen, Ye Tao, Guanyun Wang, Runchang Kang, Tovi Grossman, Stelian Coros, and Scott E. Hudson. 2018. Forte: User-Driven Generative Design. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3173574.3174070

[13] Tung Chiang. 2023. Present Ten: A Special Collection Celebrates 10 Years of Design Series. https://www.heathceramics.com/blogs/heath-journal/present-ten-a-special-collection-celebrates-10-years-of-design-series

[14] Craig Clunas. 1991. *Superfluous things : material culture and social status in early modern China.* Urbana, Ill. : University of Illinois Press. http://archive.org/details/superfluousthing0000clun

[15] Laura Devendorf, Abigail De Kosnik, Kate Mattingly, and Kimiko Ryokai. 2016. Probing the Potential of Post-Anthropocentric 3D Printing. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems.* ACM, Brisbane QLD Australia, 170–181. doi:10.1145/2901790.2901879

[16] Laura Devendorf and Kimiko Ryokai. 2015. Being the Machine: Reconfiguring Agency and Control in Hybrid Fabrication. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, Seoul Republic of Korea, 2477–2486. doi:10.1145/2702123.2702547

[17] Mark Dion and Richard Klein. 2003. *Mark Dion: drawings, journals, photographs, souvenirs, and trophies, 1990-2003.* The Aldrich Museum of Contemporary Art, Ridgefield, Conn. Open Library ID: OL18766499M.

[18] Marie-Noëlle Farcy. 2015. David Altmejd: Flux. https://www.mudam.com/exhibitions/flux

[19] Anne Filson and Gary Rohrbacher. 2008. AtFAB Open Source Furniture. https://filson-rohrbacher.com/work/atfab

[20] Paula Findlen. 1994. *Possessing nature : museums, collecting, and scientific culture in early modern Italy.* Berkeley : University of California Press. http://archive.org/details/possessingnature0000find

[21] James Gips and George Stiny. 1975. An Investigation of Algorithmic Aesthetics. *Leonardo* 8, 3 (1975), 213–220. doi:10.2307/1573240 Publisher: The MIT Press.

[22] Say Leng Goh, Graham Kendall, and Nasser R. Sabar. 2019. Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem. *Journal of the Operational Research Society* 70, 6 (June 2019), 873–888. doi:10.1080/01605682.2018.1468862 Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01605682.2018.1468862.

[23] Xinyue Gui, Ding Xia, Wang Gao, Mustafa Doga Dogan, Maria Larsson, and Takeo Igarashi. 2025. Draw2Cut: Direct On-Material Annotations for CNC Milling. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, 1–17. doi:10.1145/3706598.3714281

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

[25] Hexagon. 2025. CABINET VISION. https://hexagon.com/products/product-groups/computer-aided-manufacturing-cad-cam-software/cabinet-vision

[26] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '99)*. Association for Computing Machinery, New York, NY, USA, 159–166. doi:10.1145/302979.303030

[27] Inventables. 2023. Easel Cabinetmaker Software. https://www.inventables.com/pages/easel-cabinetmaker

[28] Ramya Iyer, Mustafa Doga Dogan, Maria Larsson, and Takeo Igarashi. 2025. XR-penter: Material-Aware and In Situ Design of Scrap Wood Assemblies. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, 1–16. doi:10.1145/3706598.3713331

[29] Shohei Katakura, Martin Taraz, Muhammad Abdullah, Paul Methfessel, Lukas Rambold, Robert Kovacs, and Patrick Baudisch. 2023. Kerfmeter: Automatic Kerf Calibration for Laser Cutting. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3544548.3580914

[30] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by Simulated Annealing. *Science* 220, 4598 (May 1983), 671–680. doi:10.1126/science.220.4598.671 Publisher: American Association for the Advancement of Science.

[31] Masahiro Kono, Maria Larsson, I-Chao Shen, and Takeo Igarashi. 2025. ScrapReCover: An Interactive Optimization System for Freeform Patchwork Layouts. In *Proceedings of the ACM Symposium on Computational Fabrication (SCF '25)*. Association for Computing Machinery, New York, NY, USA, 1–15. doi:10.1145/3745778.3766653

[32] Bongjin Koo, Jean Hergel, Sylvain Lefebvre, and Niloy J. Mitra. 2016. Towards Zero-Waste Furniture Design. doi:10.48550/arXiv.1604.00047 arXiv:1604.00047 [cs].

[33] Maria Larsson, Hironori Yoshida, and Takeo Igarashi. 2019. Human-in-the-loop fabrication of 3D surfaces with natural tree branches. In *Proceedings of the 3rd Annual ACM Symposium on Computational Fabrication (SCF '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3328939.3329000

[34] Manfred Lau, Akira Ohgawara, Jun Mitani, and Takeo Igarashi. 2011. Converting 3D furniture models to fabricatable parts and connectors. In *ACM SIGGRAPH 2011 papers (SIGGRAPH '11)*. Association for Computing Machinery, New York, NY, USA, 1–6. doi:10.1145/1964921.1964980

[35] Mackenzie Leake, Kathryn Jin, Abe Davis, and Stefanie Mueller. 2023. InStitches: Augmenting Sewing Patterns with Personalized Material-Efficient Practice. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3544548.3581499

[36] Mackenzie Leake, Frances Lai, Tovi Grossman, Daniel Wigdor, and Ben Lafreniere. 2021. PatchProv: Supporting Improvisational Design Practices for Modern Quilting. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–17. doi:10.1145/3411764.3445601

[37] Golan Levin. 2000. *Painterly Interfaces for Audiovisual Performance.* Ph. D. Dissertation. MIT.

[38] Rebecca Lin, Michal Lukáč, and Mackenzie Leake. 2025. Refashion: Reconfigurable Garments via Modular Design. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Association for Computing Machinery, New York, NY, USA, 1–18. doi:10.1145/3746059.3747632

[39] Louisiana Channel. 2016. David Altmejd Interview: The Heart is a Werewolf. https://www.youtube.com/watch?v=bc5qgYWl9lU

[40] Arthur MacGregor. 2007. *Curiosity and enlightenment : collectors and collections from the sixteenth to the nineteenth century.* New Haven [Conn.] : Yale University Press. http://archive.org/details/curiosityenlight0000macg

[41] John Maeda. 2000. *Maeda@media.* Thames & Hudson. Google-Books-ID: x6IhAQAAIAAJ.

[42] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. 1997. Design galleries: a general approach to setting parameters for computer graphics and animation. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., USA, 389–400. doi:10.1145/258734.258887

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

[44] Yuxuan Mei, Lirong Que, Mackenzie Leake, and Adriana Schulz. 2025. PatchUp: Interactive Patchwork Design for Scrap Fabric Upcycling. In *Proceedings of the ACM Symposium on Computational Fabrication (SCF '25)*. Association for Computing Machinery, New York, NY, USA, 1–17. doi:10.1145/3745778.3766667

[45] Stefanie Mueller, Bastian Kruck, and Patrick Baudisch. 2013. LaserOrigami: laser-cutting 3D objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2585–2592. doi:10.1145/2470654.2481358

[46] Martin Nisser, Christina Chen Liao, Yuchen Chai, Aradhana Adhikari, Steve Hodges, and Stefanie Mueller. 2021. LaserFactory: A Laser Cutter-based Electromechanical Assembly and Fabrication Platform to Make Functional Devices & Robots. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–15. doi:10.1145/3411764.3445692

[47] A. Michael Noll. 1967. The digital computer as a creative medium. *IEEE Spectrum* 4, 10 (Oct. 1967), 89–95. doi:10.1109/MSPEC.1967.5217127 Conference Name: IEEE Spectrum.

[48] Lars Nolle, Alec Goodyear, Adrian A. Hopgood, Phil D. Picton, and Nicholas StJ Braithwaite. 2001. On Step Width Adaptation in Simulated Annealing for Continuous Parameter Optimisation. In *Computational Intelligence. Theory and Applications*, Bernd Reusch (Ed.). Springer, Berlin, Heidelberg, 589–598. doi:10.1007/3-540-45493-4_59

[49] Behnaz Norouzi, Marianne Kinnula, and Netta Iivari. 2021. Making Sense of 3D Modelling and 3D Printing Activities of Young People: A Nexus Analytic Inquiry. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–16. doi:10.1145/3411764.3445139

[50] Opendesk. 2025. Opendesk. https://www.opendesk.cc/

[51] Daniela K. Rosner. 2012. The material practices of collaboration. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (CSCW '12)*. Association for Computing Machinery, New York, NY, USA, 1155–1164. doi:10.1145/2145204.2145375

[52] Thijs Roumen, Ingo Apel, Jotaro Shigeyama, Abdullah Muhammad, and Patrick Baudisch. 2020. 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)*. Association for Computing Machinery, New York, NY, USA, 293–303. doi:10.1145/3379337.3415895

[53] Thijs Roumen, Conrad Lempert, Ingo Apel, Erik Brendel, Markus Brand, Laurenz Seidel, Lukas Rambold, and Patrick Baudisch. 2021. autoAssembler: Automatic Reconstruction of Laser-Cut 3D Models. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 652–662. doi:10.1145/3472749.3474776

[54] Thijs Roumen, Jotaro Shigeyama, Julius Cosmo Romeo Rudolph, Felix Grzelka, and Patrick Baudisch. 2019. SpringFit: Joints and Mounts that Fabricate on Any Laser Cutter. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 727–738. doi:10.1145/3332165.3347930

[55] Yuliy Schwartzburg and Mark Pauly. 2013. Fabrication-aware Design with Intersecting Planar Pieces. *Computer Graphics Forum* 32, 2pt3 (2013), 317–326. doi:10.1111/cgf.12051 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12051.

[56] Ticha Sethapakdi, Daniel Anderson, Adrian Reginald Chua Sy, and Stefanie Mueller. 2021. Fabricaide: Fabrication-Aware Design for 2D Cutting Machines.

In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3411764.3445345
[57] Anselm Strauss and Juliet M. Corbin. 1990. *Basics of qualitative research: Grounded theory procedures and techniques*. Sage Publications, Inc, Thousand Oaks, CA, US. Pages: 270.
[58] H. Takagi. 2001. Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. *Proc. IEEE* 89, 9 (Sept. 2001), 1275–1296. doi:10.1109/5.949485 Conference Name: Proceedings of the IEEE.
[59] Rundong Tian, Vedant Saran, Mareike Kritzler, Florian Michahelles, and Eric Paulos. 2019. Turn-by-Wire: Computationally Mediated Physical Fabrication. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 713–725. doi:10.1145/3332165.3347918
[60] Iremnur Tokac, Benay Gursoy, Herman Bruyninckx, and Andrew Vande Moere. 2022. Craft-Inspired Digital Fabrication: A Study of Interactive Robotic Clay Carving. In *Proceedings of the 7th Annual ACM Symposium on Computational Fabrication (SCF '22)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3559400.3562003
[61] Anna Vallgårda and Johan Redström. 2007. Computational composites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. Association for Computing Machinery, New York, NY, USA, 513–522. doi:10.1145/1240624.1240706
[62] Tom van der Putte. 2009. Using the discrete 3D Voronoi diagram for the modelling of 3D continuous information in geosciences. (2009). https://repository.tudelft.nl/record/uuid:b35ad433-067a-4b20-8067-e18a2de76490
[63] Ron Wakkary. 2021. *Things We Could Design*. MIT Press.
[64] Ron Wakkary, William Odom, Sabrina Hauser, Garnet Hertz, and Henry Lin. 2015. Material speculation: actual artifacts for critical inquiry. In *Proceedings of The Fifth Decennial Aarhus Conference on Critical Alternatives (CA '15)*. Aarhus University Press, Aarhus N, 97–108. doi:10.7146/aahcc.v1i1.21299
[65] Whitney Museum of American Art. 2023. Rashid Johnson on New Poetry. https://www.youtube.com/watch?v=YALAi11AGT4
[66] Karl D.D. Willis, Cheng Xu, Kuan-Ju Wu, Golan Levin, and Mark D. Gross. 2010. Interactive fabrication: new interfaces for digital fabrication. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction (TEI '11)*. Association for Computing Machinery, New York, NY, USA, 69–72. doi:10.1145/1935701.1935716
[67] Stephen Wolfram. 1983. Statistical mechanics of cellular automata. *Reviews of Modern Physics* 55, 3 (July 1983), 601–644. doi:10.1103/RevModPhys.55.601 Publisher: American Physical Society.
[68] Chenming Wu, Haisen Zhao, Chandrakana Nandi, Jeffrey I. Lipton, Zachary Tatlock, and Adriana Schulz. 2019. Carpentry compiler. *ACM Trans. Graph.* 38, 6 (Nov. 2019), 195:1–195:14. doi:10.1145/3355089.3356518
[69] Lap-Fai Yu, Sai-Kit Yeung, Chi-Keung Tang, Demetri Terzopoulos, Tony F. Chan, and Stanley J. Osher. 2011. Make it home: automatic optimization of furniture arrangement. In *ACM SIGGRAPH 2011 papers (SIGGRAPH '11)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/1964921.1964981
[70] Vincent F. Yu and Shih-Wei Lin. 2014. Multi-start simulated annealing heuristic for the location routing problem with simultaneous pickup and delivery. *Applied Soft Computing* 24 (Nov. 2014), 284–290. doi:10.1016/j.asoc.2014.06.024
[71] Amit Zoran and Leah Buechley. 2013. Hybrid Reassemblage: An Exploration of Craft, Digital Fabrication and Artifact Uniqueness. *Leonardo* 46, 1 (Feb. 2013), 4–10. doi:10.1162/LEON_a_00477
[72] Amit Zoran and Joseph A. Paradiso. 2013. FreeD: a freehand digital sculpting tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Paris France, 2613–2616. doi:10.1145/2470654.2481361

A Layout Optimization (Simulated Annealing)

This appendix provides implementation details for the simulated annealing (SA) procedure. Our formulation follows the standard SA algorithm of Kirkpatrick et al. [30] and incorporates three established extensions used in prior work: temperature-dependent movement ranges [48], stagnation-triggered reheating [22], and multi-start initialization [70]. We include here only the parameter settings, movement-range function, and pseudocode necessary for reproducing our system. The full SA pseudocode appears in Algorithms 1–3.

Temperature-Dependent Movement Range

During optimization, each shape may move between 1 and 5 grid cells per step, depending on temperature. We define a movement range function:

$$R(T) = \left\lfloor \frac{T - T_{\min}}{T_0 - T_{\min}} \cdot (R_{\max} - R_{\min}) \right\rfloor + R_{\min},$$

with $R_{\min} = 1$, $R_{\max} = 5$. Algorithm 2 queries $R(T)$ each iteration, allowing coarse exploration early in the run and fine-grained adjustment as T cools.

Cooling and Reheating

We use a geometric cooling schedule $T_{i+1} = \alpha_i T_i$ with adaptive updates to α_i. When a new best solution is found, α_i is increased slightly (bounded above by 0.99) to permit extended selective search. To avoid prolonged stagnation, the algorithm reheats when no improvement has occurred for r_{reheat} iterations:

$$T \leftarrow \min(\rho\, T,\ T_0),$$

after which α resets to its base value. These mechanisms help the search escape shallow local minima while still converging reliably.

Multi-Start Structure

Before refinement, the system performs n_{starts} short, fast annealing runs from randomized initial layouts. Each run uses a distinct temperature scaling factor and a slightly accelerated cooling rate (0.75α). The best result is then refined with a slower, near-convergent schedule. This multi-start strategy yields diverse candidate layouts and improves the likelihood of identifying a compact, fabrication-valid arrangement for user curation.

Objective Function Components

For completeness, we list the exact formulations of the five terms used in Equation (1), along with their weights W_x:

$$O = (\text{numOverlap})|S|^{W_O}, \qquad W_O = 3.0$$

$$V = \sum_{i=1}^{N} (\text{sizeVoid}_i)^{W_V}, \qquad W_V = 4.0$$

$$G = (\text{numBottomSpace})|S|^{W_G}, \qquad W_G = 3.0$$

$$A = \text{totalArea} \times W_A, \qquad W_A = 0.1$$

$$R = \left(\frac{\text{aspectRatio}}{\text{targetRatio}}\right)^{W_R}, \qquad W_R = 2.0$$

Parameter Settings

For reference, we use the following default parameter values in our implementation:

- minimum temperature: $T_{\min} = 0.1$
- maximum temperature: $T_0 = 10{,}000$
- iteration budget: $\text{iter}_{\max} = 1000$
- reheating threshold: $r_{\text{reheat}} = 100$
- movement range bounds: $R_{\min} = 1$, $R_{\max} = 5$

These settings follow standard SA practice and correspond to the values used in Algorithms 1–2.

Algorithm 1: MultiStart and Refine (Solution Annealing)

Input : Shape set S; start temperature T_0; cooling rate α; maximum iterations iter_{max}; minimum temperature T_{min}; number of starts n_{starts}; re-heat counter r_{reheat}
Output : Best overall layout found $L^\star$

for $i \leftarrow 0$ **to** $n_{starts} - 1$ **do**
  $L_0 \leftarrow \text{RandomLayout}(S)$
  $T \leftarrow T_0\left(1 - \frac{i}{n_{\text{starts}}}\right)$
  $\alpha_{\text{ms}} \leftarrow 0.75\,\alpha$
  $L_i \leftarrow \text{Anneal}(L_0,\ T,\ \alpha_{\text{ms}}, \text{iter}_{\text{max}},\ T_{\text{min}},\ r_{\text{reheat}})$
end
$L^{\text{best}} \leftarrow \arg\min_{L_i} \text{Score}(L_i)$
$L \leftarrow \text{Anneal}(L^{\text{best}},\ 0.1T_0,\ 0.99,\ \text{iter}_{\text{max}},\ T_{\text{min}},\ r_{\text{reheat}})$
while $\neg\,\text{Valid}(L)$ **do**
  $L \leftarrow \text{Anneal}(L,\ T_0/n_{\text{starts}},\ 0.99,\ \text{iter}_{\text{max}},\ T_{\text{min}},\ r_{\text{reheat}})$
end
return L

B Shelving Generation (Cellular Automata)

This appendix provides the full computational specification for the shelving generation stage inspired by cellular automata [67], including path scoring, recursive opportunity scoring, and the cellular automata rules used during growth. The core growth loop and update rules are given as pseudocode in Algorithms 4–6, with rule categories summarized in Table 1.

B.1 Path Scoring Function

Path scoring assigns a cost to each edge that an active cell may traverse. Each edge $e = (s_1, s_2)$ receives a value based on the terrain of its adjacent grid squares.

Let $T(s)$ denote the terrain value assigned to square s during the iterative neighbor-propagation process. The path scoring function is

$$P(s_1, s_2) = \begin{cases} 1, & \text{if } s_1 \text{ and } s_2 \text{belong to different shapes,} \\ \frac{T(s_1)+T(s_2)}{2}, & \text{otherwise.} \end{cases} \tag{1}$$

Edges between different shapes therefore form low-cost valleys, while edges within a region inherit averaged terrain values, producing ridges under object silhouettes and moderate slopes in open areas.

Algorithm 2: Anneal

Input : Initial layout L, start temperature T_0, cooling rate α, max iterations iter_{max}, minimum temperature T_{min}, re-heat counter r_{reheat}
Output : Best layout found $L^\star$

$L^\star \leftarrow L;\quad T \leftarrow T_0;\quad c \leftarrow \alpha;\quad k \leftarrow 0$
for $t \leftarrow 0$ **to** $\text{iter}_{\text{max}} - 1$ **do**
  $m \leftarrow R(T)$
  $L' \leftarrow \text{CreateNeighbor}(L, m)$
  $\Delta E \leftarrow \text{Score}(L') - \text{Score}(L)$
  if $\Delta E < 0 \ \vee\ rand() < e^{-\Delta E/T}$ **then**
    $L \leftarrow L'$;
    if $\text{Score}(L) < \text{Score}(L^\star)$ **then**
      $L^\star \leftarrow L;\quad k \leftarrow 0;\quad c \leftarrow \min(c + 0.01,\ 0.99)$
    end
  else
    $k \leftarrow k + 1$
  end
  $T \leftarrow T\,c$
  if $k > r_{reheat}$ **then**
    $T \leftarrow \min(\rho\,T,\ T_0);\quad c \leftarrow \alpha;\quad k \leftarrow 0$
  end
  if $T < T_{\text{min}}$ **then**
    break
  end
end
return $L^\star$

Algorithm 3: CreateNeighbor (generate next anneal option)

Input : Current layout L containing shape set S; movement range m
Output : Perturbed layout L'

```
L′ ← L (deep copy)
i ← randInt(0, |S| − 1);   s ← S[i]
moveType ← randInt(1, 9)
if moveType ≤ 8 then
    (Δx, Δy) ← directionVector(moveType) × m
    s.x ← s.x + Δx;   s.y ← s.y + Δy
else
    repeat
        j ← randInt(0, |S| − 1)
    until j ≠ i
    (s.x, s.y) ⟷ (S[j].x, S[j].y)
end
if ∃ s ∈ S : s.x < 0 ∨ s.y < 0 then
    Recenter(L′)
end
MakeLayout(L′)
CalculateScore(L′)
return L′
```

Algorithm 4: GrowCells: Main CA Growth Loop

Input : shape set S; layout size (H, W); paths $\mathcal{P}$; recursion depth ρ; terrain cap ϕ
Output : final cellular grid $\mathcal{G}$

```
CreateTerrain()// construct terrain heights
CalcPathValues(𝒫)// compute path scores
MakeInitialCells(𝒢, S)// seed perimeter & shape bottoms
while AliveCount(𝒢) > 0 do
    GrowOnce(𝒢, 𝒫, p, ϕ)
    UpdateAlive(𝒢)
end
return 𝒢
```

Algorithm 5: GrowOnce: Advance One Active Cell

Input : grid $\mathcal{G}$; paths $\mathcal{P}$; active cell p; terrain cap ϕ
Output : p updated (deactivated or extended)

```
if |Alive(y, x)| > 1 then
    p ← MergeStrains(y, x) // Merge rules (M1–M2)
else if |Dead(y, x)| > 0 then
    p.alive ← false // Crowding rule (M3)
end
if ¬p.alive then
    return
end
Ω ← {left, up, right} // possible moves
foreach d ∈ Ω do
    d.valid ← IsValidOption(p, d, ϕ) // elimination rules
        (E1–E3)
    if d.valid then
        d.score ← OppScore(p, d)
    end
end
Ω ← {d | d.valid}
if Ω = ∅ then
    p.alive ← false // no valid move
end
else
    d⋆ ← ChooseDirection(p, Ω) // selection rules
        (S1–S5)
    AddCell(d⋆.y, d⋆.x, p, (x, y))
end
return
```

Table 1: Rule Categories for Algorithm 5

Category	Description of rules
Merge	M1. Standard—multiple active cells at (x, y) merge, one survives. M2. Passing—active cells that cross merge, one survives. M3. Crowded—active meets dead cell $\Rightarrow$ active dies.
Elimination	E1. No backtracking. E2. No growing through shapes. E3. Out of bounds disallowed.
Selection	S1. If only one option exists, take it. S2. Prefer moves toward different strains. S3. Prefer lower-cost paths. S4. On ties, keep direction. S5. Deterministic tie-break for repeatability.

Algorithm 6: OppScore: Recursive Look-Ahead

Input : parent p at (x_0, y_0); option $d = (\text{dir}, x, y)$; recursion depth r
Output : opportunity score s (lower = better)

if $\neg$`InBounds`(x, y) **then**
    return 1
end
$c_0 \leftarrow \begin{cases} \mathcal{P}_{y_0,x_0}[\text{dir}], & r = \rho \\ 0, & \text{otherwise} \end{cases}$
$Q \leftarrow \{\text{left}, \text{up}, \text{right}\}$
foreach $q \in Q$ **do**
    $q.\text{val} \leftarrow \mathcal{P}_{y,x}[q.\text{dir}]$
    $q.\text{valid} \leftarrow$ `IsValidPath`(p, q)
    if $\exists c \in \mathcal{G}_{q.y,q.x} : \neg c.alive \wedge c.strain \neq p.strain$ **then**
        $q.\text{val} \leftarrow 1$
    end
end
if `CellTrapped`(Q) **then**
    return ∞
end
if $r = 0$ **then**
    return $c_0 + \min_{q.valid} q.val$
end
return $c_0 + \min_{q.valid}\big(q.val + \texttt{OppScore}(p, q, r-1)\big)$