PacCAM: Material Capture and Interactive 2D Packing for Efficient Material Usage on CNC Cutting Machines

Daniel Saakes [1]
daniel@saak.es

Thomas Cambazard [1]
thomas.cambazard@epfl.ch

Jun Mitani [1,2]
mitani@cs.tsukuba.ac.jp

Takeo Igarashi [1,3]
takeo@acm.org

[1]JST ERATO Igarashi Design Interface Project [2]University of Tsukuba [3]The University of Tokyo

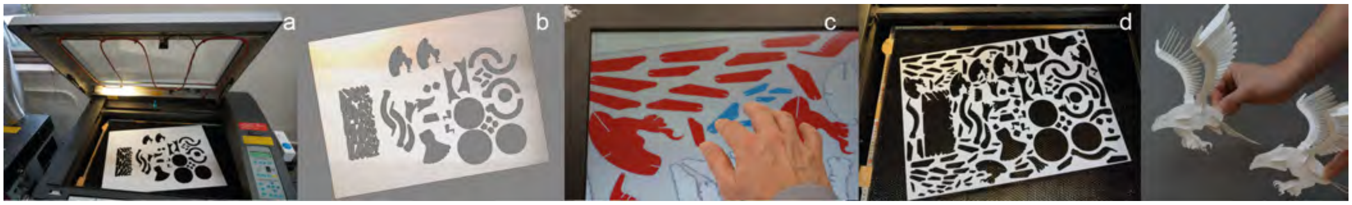

Figure 1. PacCAM supports users in packing 2D parts by capturing materials in a CNC machine (a) and using the captured materials (b) in a dedicated user interface based on 2D rigid body simulation and snapping (c), resulting in optimized utilization of material for efficient fabrication (d).

ABSTRACT

The availability of low-cost digital fabrication devices enables new groups of users to participate in the design and fabrication of things. However, software to assist in the transition from design to actual fabrication is currently overlooked. In this paper, we introduce PacCAM, a system for packing 2D parts within a given source material for fabrication using 2D cutting machines. Our solution combines computer vision to capture the source material shape with a user interface that incorporates 2D rigid body simulation and snapping. A user study demonstrated that participants could make layouts faster with our system compared with using traditional drafting tools. PacCAM caters to a variety of 2D fabrication applications and can contribute to the reduction of material waste.

INTRODUCTION

Many objects are composed out of raw materials that enter the fabrication process flat; as sheets or in rolls. Computer numerical control (CNC) fabrication devices such as plotters, routers, and cutters transform these materials into multiple parts that make up 3D objects. Specialists optimize the process of fabricating physical parts with many goals, such as efficient material use, avoid complex tooling, and minimization of machine time. Although there are fast and advanced algorithms [4, 7] (NP-hard) for packing 2D shapes, there are certain cases that automatic methods can not handle. For instance, source materials that contain graphics or have spatially varying quality require complicated trade-offs between function, aesthetics and efficient material use. In cases with such soft constraints, layout is performed by hand.

UIST'13, October 6–9, 2013, St. Andrews, United Kingdom.

http://dx.doi.org/10.1145/2501988.2501990

With the proliferation of decentralized [22] and personal [11] digital fabrication, new groups of occasional designers/makers are emerging. However, material usage is far from optimized in this emerging user community. We visited numerous fablabs, hackerspaces, and university labs and found piles of partly used materials as shown in Figure 1b. In this paper, we present a new system that is specifically aimed at manual packing of shapes to reduce material waste.

When designing physical objects that are composed of flat materials, designers encounter and switch between multiple representations: a) the intended 3D object, b) the decomposition of the object in 2D parts, and c) the layout and fitting of the 2D parts for actual fabrication on the sheets/rolls. A multitude of research and commercial software exists to support the user or automate the transition from step a to b [2, 14, 17, 18, 19, 23, 24]. However, the transition from step b to c is is currently overlooked. This final step from design to actual manufacturing is a one way process, only output. We ultimately aim to make this final step bi-directional: both output and input by using material features as a creative constraint in the design of an object.

Our contribution is a dedicated user interface for making overlap-free 2D layouts. Our solution consists of capturing materials inside the CNC machine that serve as a guide and constraint in a packing and layout task. On the basis of observed natural behavior, we adapt existing ideas of 2D rigid body simulation in tabletop computing and extend research on snapping with collision-snapping. We evaluate the performance of collision-snapping in a small user study and compare performance improvement in simple tasks with a traditional interface.

RELATED WORK

In general, the notion of a material is lost when using computer aided design software [25]. Only a few applications [10, 29, 30] consider the imperfections of materials as an artistic expression. In industries that work with natural materials such as leather, features or defects influence the price [4], and therefore, craftsmen make trade-offs to minimize the visibility of a defect in the final product while ensuring efficient material use.

A discrepancy exists between a virtual representation during design and spatial impact in the real world [3]. Interactive fabrication [26] aims to automate all the steps of design and fabrication to make a direct connection between human input and machine output. CopyCad [9] is a camera projector system on a CNC device that aims to integrate physical objects into CAD. The Visicut project [21] employs a camera in a laser cutter for a WYSIWYG preview and positioning of graphics on objects. However, these methods do not address layout or packing problems directly.

In computer graphics, various tools are offered to the user for mapping a 2D texture on a 3D model. For instance, in UV texture mapping, the artist switches between a texturemap view and the 3D object. Other tools that assist occasional users in the transition from 3D concept to 2D parts. For instance, in the sketchchair [23] application, the user sketches a chair and the software constructs the 3D shape of the chair and calculates the 2D parts for fabrication. Autodesk 123D software [2] assists the user in fabricating 3D objects and provides a number of ways to decompose a 3D shape into 2D parts. However, these applications do not provide intelligent support for interactive manual packing or provide bidirectional feedback to inform design choices.

Bridging paper and electronic documents has been addressed by various prior art. Early systems aimed to simplify the process of digitizing paper documents [15]. Later, systems such as PADD [13] focused on interexchange, i.e., combining the affordances of both paper and digital documents. Our effort to capture and layout parts on materials is similar to the interexchange of physical and digital documents.

Within the domain of fabrication, a large body of work exists for machine optimization of layouts, bin-packing of rectangular shapes, or irregular shape nesting in the domain of fabrication [4, 7]. Machine packing is an NP-hard problem and algorithms are typically intended for mass production of shapes in clothing or metal industries. These algorithms cannot handle nontrivial or conflicting constraints present in many real-world design scenarios. Manual layout allows the user to leave material in a certain shape intentionally when using only part of the material. Users may layout parts considering their structure, e.g., grouping related parts together, which is helpful in the subsequent manual assembly process. Finally, when a design does not fit the material, interactive packing helps the user to decide which parts to remove, how to modify its shape, or how much additional material is necessary. Therefore, we focus on manual and interactive packing.

CAD drafting tools or 2D vector illustration tools (e.g. Adobe Illustrator, Rhino and Autocad) are regularly used for making layouts. Although direct manipulation of translation is supported, rotation generally requires activation via a button. Therefore, a packing task with combined manipulation of orientation and positioning [16] is less supported. Research into multitouch and tabletop computing has introduced natural interaction using physics simulation. For instance, the Bumptop system [1] allows informal organization of desktop icons using simulation of dragging and pushing motions for creating piles. Other systems have introduced 2.5D interfaces, such as the simulation of cards sliding over each other [27], or use whole-hand gestures to sweep objects [28]. Although all these systems contain collision detection and physical simulation, none are optimized specifically for packing tasks.

Other related work has focused on snapping and smart guides to assist users in layout tasks. The common implementation in 2D vector illustration software is magnetic snapping [6]; when an object is near a snapping opportunity, the object snaps into place. The snap-and-go system [5] argues that this behavior leads to inaccessible areas and proposes to change the mapping from motor space to screen space temporarily. However, these mapping systems only work via indirect manipulation (mouse + pointer). In addition, current snapping methods only support either rotation, position or scaling, and only function for specific angles (orthogonal/diagonal). Fitting irregular shapes requires manipulation of both position and orientation and snapping to arbitrary objects.

MATERIAL CAPTURE

Capturing material shape and fitting a design is an essential part of the packing task. For accurate capturing, we implemented two methods: an overlooking camera, similar to Visicut [21], and using the toolhead of a laser cutter to scan materials on the table. Both methods are shown in Figure 2.

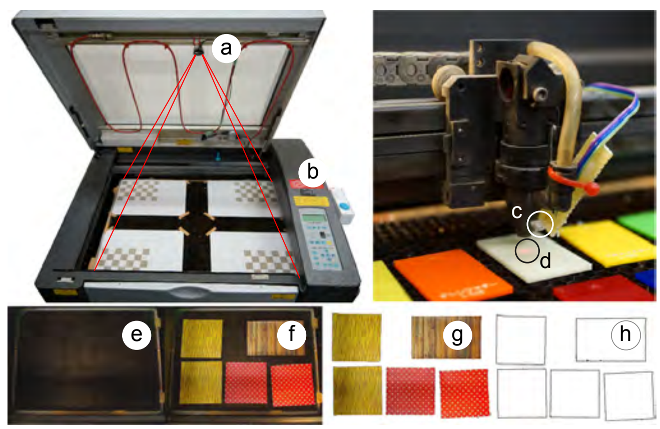

Figure 2. Top Left: an overlooking camera (a), mounted in the lid of a laser cutter for capturing materials, activated by a remote control attached to the side (b). Top right: a method for capturing that uses a photodiode (c) mounted on the toolhead of the laser cutter. The presence of material is detected by the reflection (d) of the sensing laser. Bottom row: a captured empty (e) and full table (f), the transformed and masked material (g), and the extracted edges (h).

Overlooking Camera

An overlooking camera is a commonly implemented method (e.g. [9, 10, 20, 21, 29]) when merging physical objects and virtual representations. Using projective transformation a pixel x,y in the camera image space is mapped to a point x,y on the CNC table. Our prototype is implemented on a laser cutter (7050-60w Commax Co. Ltd.) with a table size of 695mm $\times$ 495mm. We mounted a calibrated Logitech C905 webcam (960 × 720 pixels) to the lid of the laser cutter, as shown in Figure 2a. The camera captures the entire table when the lid is open, resulting in a resolution of 1 pixel to 1.17 mm.

Extrinsic calibration is performed once, when the camera is installed. White cardboard is placed in the corners of the table. The laser cutter prints calibration patterns using rastering and the homography matrix is calculated using OpenCV as shown in Figure 2. This eliminates the need for manual or visual alignment. The accuracy of this method is established in a reverse process by first cutting a grid of test patterns on cardboard that covers the entire table. A picture is taken and the perspective transformation is calculated for a number of points (N = 48). The x,y positions are then compared to the coordinates on the table. This results in a mean error of 0.83 mm and a standard deviation of 0.50 mm.

In everyday use, users capture two pictures, one of the empty table and one of the material on the table. To enable this, we attached a remote control (Figure 2b) to the side of the laser cutter. Simple background subtraction results in a mask that indicates the presence of the material (Figure 2efg). The OpenCV findContours function is used to detect the edges of the material, as shown in Figure 2h. The resulting polylines are simplified, cleaned and saved in an SVG file that contains both bitmap and vector representations. The SVG file can be used with 2D layout software to include the material position and orientation as a constraint in the design process.

Scanning Laser

An exact but slow method to acquire material on the CNC table is to use the toolhead both to cut parts and scan material. Thus, the presence of a material is directly correlated to the coordinate system of the device without prior calibration. In addition to the burning laser, a laser cutter has a low-power laser for previewing a programmed toolpath. We assume that the low-power laser is in perfect alignment with the high-power laser. As shown in Figure 2cd, an inexpensive photodiode (Taos TSL250R) is mounted on the toolhead and is directed toward the point where the laser hits the source material.

For each x,y position on the table, the material is sensed (non-contact) by detecting the reflection of the laser, which is measured by a 10bit analog-to-digital converter. The intensity of the reflection is dependent on the wavelength of the laser and the color of the material. This technique is accurate, but because the toolhead must be moved to probe a point, it is slow, depending on the resolution. We currently employ a serpentine scanning algorithm prototyped on a pen-plotter. Intelligent and adaptive scanning, possibly guided by the overlooking camera will speed up this process.

INTERACTIVE 2D PACKING

The interactive packing user interface makes use of a physics simulation. Parts have friction and can collide and are translated and orientated by means of a virtual spring/damper joint. In addition, basic layout functionality has been implemented such as copy, cut, paste, duplicate, grouping, parts margins, flipping parts, locking position and fixing orientation. The captured material is loaded as a background image. Around the material and inside its cavities, invisible and static boundary parts are created that collide with the parts through 2D rigid body simulation. Parts are initially placed and orientated as in the original drawing. In case of overlap with other parts or boundary parts, parts begin in a *ghost* state, i.e. no collision detection is performed, and they are drawn translucent. Ghost parts are constantly and automatically checked for overlap, and if possible, are fit in the physics simulation.

Observation Study

To obtain inspiration when designing the packing interface, we asked several people (N=8, all male, computer graphics and architecture students) to solve two packing tasks and observed the process. First, the packing of a sketchchair [23] and second, a standard problem [7] from the clothing industry often used to benchmark packing algorithms. For the observation study, we laser cut the parts from acrylic. From our observations, we extracted a number of patterns that people use when manipulating parts. 1) grouping similar parts together in subsolutions 2) compacting a group of parts by pushing parts, 3) fitting groups or parts together by subtle testing of angles and positions, and 4) using both hands. On the basis of these observations, we designed a new interface that has a foundation in both earlier physical simulation work and snapping.

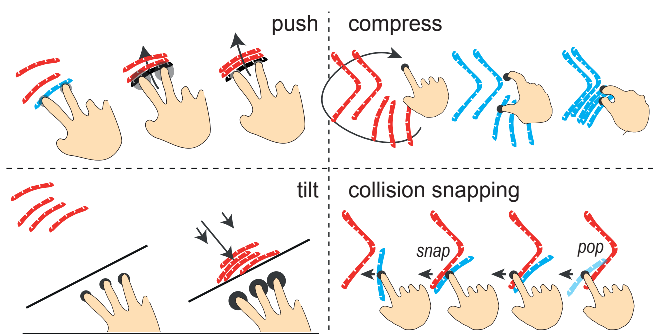

Figure 3. User Interface for making layouts. Shown are single handed multitouch ideas: (top left) a press with two fingers to push parts with the selection; (top right) a pinch gesture that compresses the selection; (bottom left) pressing on the side with three fingers initiates a tilt and all parts move towards the fingers; (bottom right) collision snapping is a combination of parts colliding and sliding over each-other.

Interaction Techniques

It was observed that users mostly gently drag parts without pushing, and that their actions consist of subtle sliding and rotating motions, and testing. Therefore, in the interface, the default behavior is that parts moved by the user collide and slide with other parts but do not alter the position or orientation of other parts. In the simulation, we temporarily lower the weight of selected parts.

Pushing. Pushing (Figure 3) to move parts was also observed. This is a default behavior in many user interfaces with physical simulation. For example, in the Roomplanner application, Wu and Balakrishnan [28] used a vertical hand motion to sweep and push objects. Likewise, Wilson et al. [27] used multiple fingers to push objects. We implemented pushing or sweeping objects as a *push* state of the selection. Activated with a modifier key or by applying pressure, the selected objects change color and push and move other objects.

Compress. Another often observed behavior was the grouping of parts into subsolutions and packing these parts tightly together in a sequence of small pushes of the outer parts. This roughly compares to building piles [1], A lasso selection selects a number of parts and is followed by a command that piles the objects. We implemented a similar compress function to pack parts together. The current prototype implementation determines the centroid of mass of the selection and applies a force toward that center for each part.

Tilt. The tilt operation lets gravity slide all the parts towards the same direction. When activated, tilt applies a force (gravity) to all parts in the layout parallel towards the position of the mouse. This is useful for moving and compressing all parts in a corner or side of the material.

Collision snapping. Dragging and pushing shapes alone is insufficient for making a 2D layout. Shapes often have to cross obstacles or move inside cavities. Therefore, the ability to *pick up*, move around, and *put back* a shape is required. As shown in Figure 4, this is achieved through the *ghost* state, which excludes a part from collision detection. In this manner, shapes can be temporarily stored in the vicinity of a target location before actually being used in the solution.

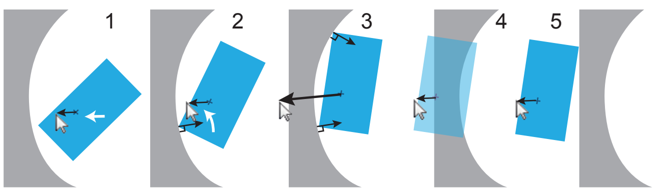

Figure 4. Collision snapping in action. From left to right: 2) The part snaps to the boundary. 3) the force in the spring between the cursor and registration point increases; 4) the part pops into ghost state; and 5) when the part has no overlap, it pops back into collision detection.

The popping back and forth from and into collision detection is implemented for mouse input interfaces. When the user drags a selection, a spring/damper joint is attached between the cursor and the selected parts. When the selection collides with other parts, the force in the spring is measured. If it exceeds a threshold value, e.g., the distance between the cursor position and part is too large, the selection pops into the ghost state and snaps to the mouse pointer. While in the ghost state, parts still have friction with the ground but do not collide. In addition, when ghost parts become overlap free, they are automatically popped back into collision detection.

Implementation

The prototype system is implemented using the Qt toolkit and supports basic functionality such as the import and export of SVG and DXF drawings and printing layouts directly to the lasercutter using the Qt printer driver. Box2D [8] is used as the physics engine. For performance, parts are simplified using polyline optimization and approximate convex decomposition [12]. We support a variety of input methods, traditional mouse, pen, and single-hand multitouch. The system is currently running in our lab and is used regularly.

EVALUATION OF THE USER INTERFACE

A full-fledged evaluation of the proposed system is beyond the scope of this paper. However, we evaluated the performance of collision snapping in a small user study. We asked participants to complete three small packing tasks for three conditions in a 3 × 3 experiment design. The first condition is a **traditional** drafting interface with rotation handles, as shown in Figure 5 left. The second condition involved a **fluid dragging** interface with fluid integration of rotation and translation [16]. The third condition added **collision** snapping to fluid dragging. We expected that subjects would be faster using the proposed interface when manipulating parts in the fluid condition compared with that in the traditional interface. We also expected that collision would render the fastest completion times because it reduces that need for precision manipulations.

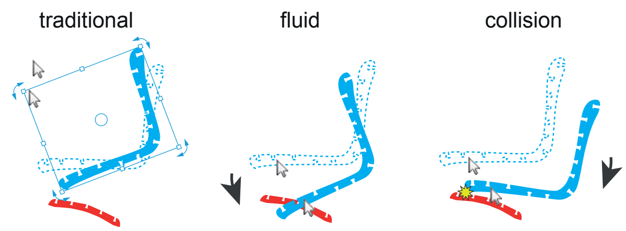

Figure 5. The user study consisted of a 3 ×3 design. Three conditions are shown: traditional modal drafting interface (traditional), fluid integration of rotation and translation (fluid), and collision snapping (collision).

Within each condition, we asked the participants to complete three tasks, as shown in Figure 6. In each task, a participant drags the given parts from a start position into a target area. The participants were instructed to avoid overlap in their solution. The task order was fixed: 1) dragging with simple target area, 2) dragging with irregular target area, and 3) compacting parts.

We chose a within-subjects design to reduce the error variance that we expected to find in a packing task. The three conditions were ordered semirandomly, and all permutations of the conditions were distributed among the participants. The observations during a pilot study using packing tasks revealed that we had to address carry-over effects. Therefore, we specifically chose simple tasks with minimal packing challenge and measured completion time and not utilization. The exactly same tasks were chosen for all three conditions. To counteract task knowledge and suppress exploration time, we showed the task and solution beforehand and provided time to practice the tasks at the start of each condition.

Setup

Twelve participants (2 female, 10 male), ranging in age from 19 to 33, were recruited from local universities. All participants had prior experience with vector drawing software, and some participants had experience with digital fabrication. One participant (participant 13) did not understand the assignment and was excluded from the results.

Each evaluation took approximately 30 minutes. The software consisted of a subset of the functionality because it only allowed single-part manipulation. The software ran on a 13-inch MacBook Air. The packing task was presented to the users in full-screen mode on a 20-inch display (SyncMaster 204B). The participants used a standard wireless mouse (Logitech M310) for input. Time was measured from the mouse-down of the first drag operation to the mouse-up of the last drag operation when the participants indicated they had completed the task. The study was performed in a meeting room.

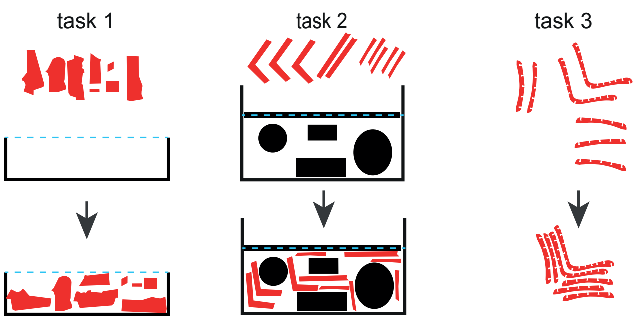

Figure 6. Within each condition of the user study, participants executed three packing tasks that involved moving parts to a target area. Top: task start and bottom: suggested solution.

Result

In the first task we tested the performance of simple packing. As depicted in Figure 6, eight almost convex shapes of various sizes had to be fit into a rectangular target area without overlapping. The required utilization of the solution was 42%. Three parts had to be rotated to fit into the target area. The repeated measures analysis of variance showed significant effects on completion time, as observed in Figure 7, ($F(2, 22) = 14.43, P<.0005$) as shown in Figure 7. Post-hoc tests using the Bonferroni corrections showed a significant effect between the collision and traditional conditions (collision<traditional, P<.0005) and between the collision and fluid conditions (collision<fluid, P<.05). However, no significant difference was found between the fluid and traditional conditions.

The second task consisted of fitting concave shapes into an irregular shaped target area as depicted in Figure 6. Although utilization was only 30%, the target area required the user to reorient the majority of ten parts to make them fit. This task was specifically designed to test collision snapping because all parts had to be popped into the target area, with static boundary parts (drawn in black) blocking the entrance. The repeated measures analysis of variance showed significant effects on completion time ($F(2, 22) = 4.08, P<.05$) .

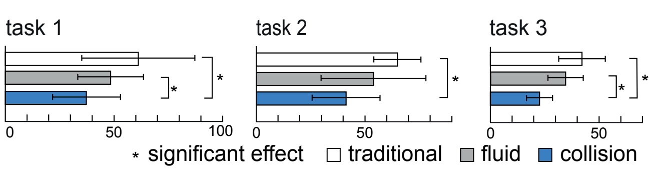

Figure 7. Results of the user study. Completion time was measured in seconds (faster is better).

Post-hoc tests using the Bonferroni correction showed a significant effect between only the collision and traditional conditions (collision<traditional, P<0.05).

The final task consisted of compacting seven shapes. This task was straightforward and did not leave much room for creative packing behavior. The task was specifically designed to test snapping and orientation. All users completed this task in less than 1 minute. A repeated measures analysis of variance determined that completion time differed significantly between the conditions ($F(2, 22) = 17.96, P<.0005$) . Post-hoc tests using the Bonferroni correction revealed a significant difference between the collision and the traditional conditions (collision<traditional, P<.005), and collision and fluid dragging (collision<fluid, P<.05). However, no significant difference was found between the fluid and traditional conditions.

Discussion

Collision snapping resulted in significantly faster completion time over the traditional interface because it does not require the precise manipulations to position and orient parts. The first two tasks, in which the participants started puzzling and optimizing their solution between the conditions, allowed for multiple solutions, and therefore, they yielded less clear results compared with the third task. Nevertheless, the results indicate that although having little prior experience with the proposed improvements, our subjects performed the tasks faster than the status quo. The results also indicate that collision snapping plays a significant role in completing time for such tasks.

A number of participants noted that the fluid dragging implementation had too little friction; i.e. the parts rotated too quickly and center of mass had to be estimated to translate parts without reorientation. In future, it will be better to allow the user to adjust the magnitude of friction and the pop-force threshold in collision snapping. A few of participants reported that they preferred the traditional interface for precision manipulation.

Task 2 (irregular target area) was reported to be confusing. Because of the noncolliding ghost parts and their tight fit with the boundary parts in the target area, ghosts often remained floating and did not snap or collide with other parts. We think an improved version of the collision-snap should involve a local search to test overlap on parts in ghost state, i.e. testing for small rotation and translation variants. Then, when a fit is found, ghost parts should automatically reorient and be placed back into the simulation. We see this as a first step towards a hybrid interface.

CONCLUSION AND FUTURE WORK

We presented a system that integrates current technologies found in sensing, computer vision, physics simulations, and desktop user interfaces to assist users in packing tasks for manufacturing. A small user study indicated an increase in performance using the proposed user interface over the traditional interface.

In this paper we focused on manual layouts. The results of the second task suggest combining the collision-snapping technique with automatic packing into a hybrid interface. In the future we plan to implement hybrid solutions that offer suggestions throughout the process of making manual layouts, as well as start, finish or suggest solutions. A new avenue of research is integrating the proposed system into design software and using the features of a material and packing constraints as input for the creative process, which will allow visualization of the relationship between materials and design decisions.

Packing is a generic problem that has many applications outside the field of fabrication with 2D materials, such as in transportation and storage. We hope that our efforts will contribute to human-in-the-loop solving of open-ended spatial layout problems across various domains.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their suggestions and ideas. We also thank Masahiro Furukawa for arranging participants for the user study. The eagle model used in Figure 1 was designed by user jtronics and is available at http://thingiverse.com/thing:30990.

REFERENCES

1. Agarawala, A., and Balakrishnan, R. Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In *Proc. CHI'06* (2006), 1283–1292.
2. Autodesk. Autodesk 123d. `http://www.123dapp.com`.
3. Azuma, R. A survey of augmented reality. *Presence 6*, 4 (1997), 355–385.
4. Baldacci, R., Boschetti, M. A., Ganovelli, M., and Maniezzo, V. Algorithms for nesting with defects. *Discrete Applied Mathematics* (2012). In Press.
5. Baudisch, P., Cutrell, E., Hinckley, K., and Eversole, A. Snap-and-go: helping users align objects without the modality of traditional snapping. In *Proc. CHI '05* (2005), 301–310.
6. Bier, E. A., and Stone, M. C. Snap-dragging. In *Proc. SIGGRAPH '86* (1986), 233–240.
7. Bounsaythip, C., and Maouche, S. Irregular shape nesting and placing with evolutionary approach. In *Proc. SMC '97*, vol. 4 (1997), 3425–3430.
8. Catto, E. Box2d: A 2d physics engine for games. `http://box2d.org`. Accessed: 25/01/2013.
9. Follmer, S., Carr, D., Lovell, E., and Ishii, H. Copycad: remixing physical objects with copy and paste from the real world. In *Adj. proc. UIST'10* (2010), 381–382.
10. Fukuchi, K., Jo, K., Tomiyama, A., and Takao, S. Laser cooking: a novel culinary technique for dry heating using a laser cutter and vision technology. In *Proc. CEA '12* (2012), 55–58.
11. Gershenfeld, N. *Fab: The Coming Revolution on Your Desktop–from Personal Computers to Personal Fabrication*. Basic Books, Inc., 2007.
12. Ghosh, M., Amato, N. M., Lu, Y., and Lien, J.-M. Fast approximate convex decomposition using relative concavity. *CAD 45*, 2 (Feb. 2013), 494–504.
13. Guimbretière, F. Paper augmented digital documents. In *Proc. UIST'03* (2003), 51–60.
14. Hildebrand, K., Bickel, B., and Alexa, M. crdbrd: Shape fabrication by sliding planar slices. *Comp. Graph. Forum 31*, 2 (May 2012), 583–592.
15. Johnson, W., Jellinek, H., Klotz, Jr., L., Rao, R., and Card, S. K. Bridging the paper and electronic worlds: the paper user interface. In *Proc. INTERACT '93 and CHI '93* (1993), 507–512.
16. Kruger, R., Carpendale, S., Scott, S. D., and Tang, A. Fluid integration of rotation and translation. In *Proc. CHI '05* (2005), 601–610.
17. McCrae, J., Singh, K., and Mitra, N. J. Slices: a shape-proxy based on planar sections. *ACM Trans. Graph. 30*, 6 (Dec. 2011), 168:1–168:12.
18. Mitani, J., and Suzuki, H. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 259–263.
19. Mori, Y., and Igarashi, T. Plushie: an interactive design system for plush toys. In *Proc. SIGGRAPH '07* (2007).
20. Mueller, S., Lopes, P., and Baudisch, P. Interactive construction: interactive fabrication of functional mechanical devices. In *Proc. UIST'12* (2012), 599–606.
21. Oster, T. Visicut: An application genre for lasercutting in personal fabrication. RWTH Aachen Univ., 2011.
22. Rifkin, J. *The third industrial revolution: how lateral power is transforming energy, the economy, and the world*. Palgrave Macmillan, New York, NY, USA, 2011.
23. Saul, G., Lau, M., Mitani, J., and Igarashi, T. Sketchchair: an all-in-one chair design system for end users. In *Proc. TEI '11* (2011), 73–80.
24. Schwartzburg, Y., and Pauly, M. Fabrication-aware design with intersecting planar pieces. *Comp. Graph. Forum 32*, 2pt3 (2013), 317–326.
25. Vallgårda, A., and Sokoler, T. A material strategy: Exploring material properties of computers. *IJDesign. 4*, 3 (2010).
26. Willis, K. D., Xu, C., Wu, K.-J., Levin, G., and Gross, M. D. Interactive fabrication: new interfaces for digital fabrication. In *Proc. TEI'11* (2011), 69–72.
27. Wilson, A. D., Izadi, S., Hilliges, O., Garcia-Mendoza, A., and Kirk, D. Bringing physics to the surface. In *Proc. UIST '08* (2008), 67–76.
28. Wu, M., and Balakrishnan, R. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *Proc. UIST '03* (2003), 193–202.
29. Yoshida, H. Bridging synthetic and organic materiality: graded transitions in material connections. In *ACM SIGGRAPH 2011 Studio Talks* (2011), 10:1–10:1.
30. Zoran, A., and Paradiso, J. The freed: a handheld digital milling device for craft and fabrication. In *Adj. proc. UIST'12* (2012), 3–4.