Cecil Piya[1]
C-Design Lab,
School of Mechanical Engineering,
Purdue University,
West Lafayette, IN 47907
e-mail: cpiya@purdue.edu

Vinayak
C-Design Lab,
School of Mechanical Engineering,
Purdue University,
West Lafayette, IN 47907
e-mail: fvinayak@purdue.edu

Karthik Ramani
Donald W. Feddersen Professor
School of Mechanical Engineering,
Purdue University,
West Lafayette, IN 47907
e-mail: ramani@purdue.edu

Proto-TAI++: Exploring Perceptually Consistent Creation of Planar Shape Assemblies Through Multimodal Tangible Interactions

We explore tangible 3D interactions that allow for geometric and perceptual correspondence between a midair modality and the 3D elements it controls. To demonstrate our approach, we use a concrete application scenario through Proto-TAI++, a multimodal system using pen-based drawing of planar shapes and their subsequent midair assembly via a hand-held planar proxy. The planarity of the proxy is a key element that physically embodies virtual planar shapes during 3D manipulation. We also investigate advanced operations such as patterning and blending using asymmetric bimanual interactions that augment midair motion with multitouch gestures. We describe a three-stage user evaluation with our system wherein our goal is to (a) study the effectiveness of the planar proxy as a tangible 3D modality, (b) contrast key features of our approach with a GUI-based planar shape assembly system, and (c) evaluate user experience and performance in creative tasks using Proto-TAI++. [DOI: 10.1115/1.4034266]

1 Introduction

Holding and manipulating physical objects is a natural task that humans learn to perform at an early age. Drawing inspiration from this, several works, most notably *graspable user interfaces* [1], have explored tangible interactions where hand-held manipulation of physical objects are used to expressively control virtual elements. Ishii and Ullmer [2] stated that within such interfaces, “the physical forms serve as both representations and controls for their digital counterparts,” and Fitzmaurice et al. [1] pointed out that “the affordances of the physical handles are inherently richer than what virtual handles afford through conventional (desktop) manipulation techniques.” The notion of using physical objects as interactive media has been found particularly relevant to digital 3D contexts [3].

While several works have utilized generic devices as such media, we find that there is little perceptual correspondence between the device and the virtual shapes being controlled. Recently, Arisandi et al. [4] demonstrated *virtual handcrafting*, an interesting way to use real-world tool metaphors for virtual modeling. Inspired by such works, we explore ordinary objects as physical proxies of virtual 3D elements. In addition to their tangibility, such proxies can also bear physical and semantic similarities to virtual counterparts, allowing users to provide perceptually consistent 3D inputs and shape manipulations.

To investigate such tangible interactions, we take an application-based approach and choose the context of planar shape assembly. Given the planarity of the unit assembly shapes, this context naturally lends itself toward the use of a planar proxy for holding, manipulating, and assembling the shapes. We posit that such interactions can allow users to vicariously hold and configure planar shapes using suggestive midair actions. The simplicity of this context also allows us to study the proposed interactions in-depth, and explore their utility within creative design scenarios.

Planar shape assemblies are used in art and design to represent 3D ideas in a minimal way, and can be fabricated with simple laser cutters. Their utility has been shown in digital modeling contexts like early-stage ideation and customization of artifacts. While prior works [5,6] have used digital sketching to define planar shapes, their lack of a 3D assembly medium results in extraneous design iterations and waste of resources. Thus, our earlier work [7] explored an interactive system called *Proto-TAI*, which enables both 2D drawing and 3D assembly in a common interface (Fig. 1). This allows users to quickly construct, explore, iterate planar assemblies in the virtual space, and close the gap between shape creation and final model realization. Here, we used the planar proxy for directly manipulating the planar shapes in 3D space.

Recently, we have observed additional interest toward interactive planar shape assembly. Most notably, works like *FlatFitFab* [8] have demonstrated how advanced modeling capabilities can enhance the esthetic quality and level of detail in planar shape assemblies. Motivated with these developments, we present our extended system, *Proto-TAI++*, and explore how our interactions can support advanced operations. Given the robustness and accessibility of *FlatFitFab*, it also provides a basis for evaluating and benchmarking our work.

1.1 Contributions.

Proto-TAI++ extends our earlier work [7] to include capabilities like interactive shape modification (Sec. 5.1.2), automatic assembly adjustment (Sec. 5.2.3), and procedural operations (Sec. 5.2.2). Here, we aim to enhance the complexity of planar shape assemblies, while retaining the simplicity of the interactions. For this, we explore asymmetric bimanual interactions that combine proxy-based 3D inputs with multitouch discrete events (Figs. 14(*a*)–14(*f*)). We also explore how freeform inputs can be contextualized to infer precise and structured modeling intent (Figs. 14(*g*)–14(*i*)). We then conduct in-depth user studies, to first assess the planar proxy's efficacy toward 3D shape manipulation (Sec. 4). Next, we perform a comparative study

[1]Corresponding author.

Contributed by the Computers and Information Division of ASME for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received March 14, 2016; final manuscript received July 12, 2016; published online August 11, 2016. Editor: Bahram Ravani.

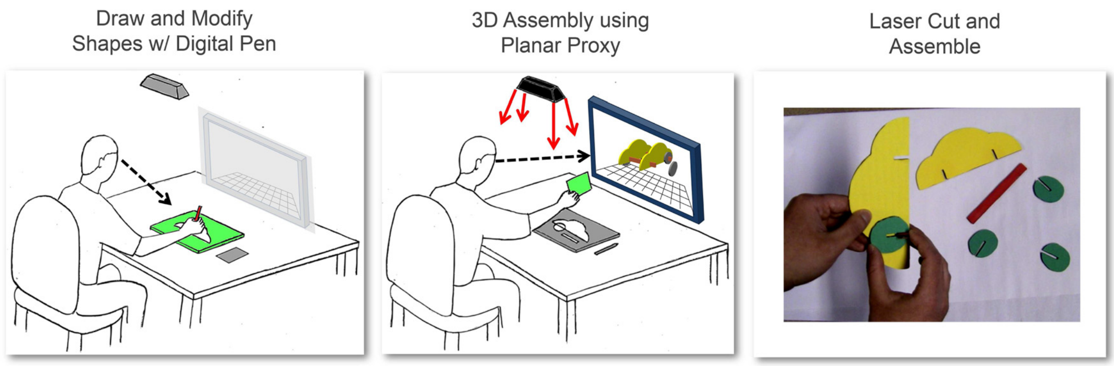

Fig. 1 General workflow: (*a*) planar shapes are drawn and modified on a digital surface, (*b*) concurrent midair inputs from a planar proxy and multitouch inputs on the tablet (with nondominant hand) are used to assemble shapes in 3D, and (*c*) the shapes can be laser cut and physically assembled[1]

between our approach and *FlatFitFab* (Sec. 6.1). Finally, we study the creative outcomes achievable using *Proto-TAI++* (Sec. 6.2).

2 Related Works

2.1 Three-Dimensional Design With Planar Shapes. Planar shapes have been commonly used as base geometries to define extrusions in CAD, or as structural scaffolds in organic 3D shape modeling [9]. While these methods infer 3D models from planar shapes, other works present algorithms to do the inverse, i.e., extract planar shape abstractions form existing 3D models [10,11].

In contrast, our work involves interactive construction of meaningful planar shape assemblies from a blank slate. Sketch-it-make-it [6] and designosaur [5] explored a similar mode of design, by drawing planar shapes on a digital sketch medium for subsequent fabrication and assembly. Similarly, interactive construction [12] allows users to provide laser-based 2D inputs to guide real-time fabrication of planar shapes. But since these methods lack a modeling space for 3D assembly, users cannot explore different configurations and intershape compatibility before fabrication. This can increase the number of *ideate–construct–iterate* cycles, leading to loss of time and material. In contrast, FlatFitFab [8] enabled a collocated virtual drawing and assembly medium, along with procedural operations to add complexity and details. SketchChair [13] also utilized procedural operations on planar shapes to generate complex designs from simple interactions.

Our approach also enables both construction and assembly of planar shapes within a common interface. However, in our multimodal system drawing and assembly are performed separately in 2D and 3D modeling spaces. We also use a planar proxy as a physical embodiment of planar shapes for direct manipulation during assembly.

2.2 Combined 2D and Midair Interfaces. With advancements in both touch and 3D motion sensors, several works have explored "continuous interaction spaces" [14,15] which integrate multitouch surface inputs with above-the-surface gestures. Such interfaces have been shown to allow seamless transition between 2D and 3D input spaces based on the needs of the current task. In mockup builder [16], the authors demonstrated the utility of such interfaces within a 3D design scenario.

Given that our workflow comprises both 2D sketching and 3D assembly, we find the value in leveraging such dual modality for fluid interactions. However, in contrast to existing works, which utilize midair hand gestures, we use tangible midair interactions via a physical proxy. Since the planar proxy is noninstrumented, we also utilize concurrent asymmetrical bimanual inputs [17], where the dominant hand communicates 3D inputs in midair and the nondominant hand indicates discrete events through multitouch gestures.

2.3 Midair Interactions for 3D Design. Several works have explored midair interactions in 3D design. In contrast to GUI-based 3D widgets, they enable a more direct and efficient means for providing spatial inputs, but at a coarser level of precision. We find that given the quick-and-dirty nature of planar shape assembly, this is a reasonable tradeoff. Midair interactions are broadly classifiable as follows.

Free-hand gestures have been increasingly used in 3D tasks like mechanical and scene assembly [18], and conceptual design [19]. While several works have explored continuous hand pose tracking [20], their utility toward robust unimanual shape manipulation has remained a challenge [21]. They also cannot provide tactile feedback for kinesthetic control of virtual shapes.

Digital controllers use in-built motion sensors and click buttons for greater 3D control. Commercial hand-held [22] or hand-worn [23] controllers have been used in design contexts that require more precision. Here, they mainly use generic hardware and metaphors for interacting with a wide range of 3D objects. Given their growing ubiquity, smartphones have also been repurposed [24] as digital controllers.

Customized controllers typically embed electronic sensors or AR markers on ordinary physical objects. They have been used in task-specific contexts like virtual sculpting [25] and handcrafting [4]. Our planar proxy is also a customized controller that is constructed from simple cardboard and tracked using a noninstrumented vision-based approach.

Planar shaped controllers were first explored by Hinckley et al. [3] for 3D data visualization. Kato et al. [26] extended them as spatial manipulators for 3D scene arrangement. The key difference in our work is that we use the planar proxy not just as a 3D controller, but as a physical embodiment of planar shapes for perceptually consistent interactions. Additionally, we also focus on how the proxy can be used for constructing new 3D designs and supporting expression of creative ideas.

3 System Implementation

Our setup (Fig. 2) in *Proto-TAI++* consists of (a) a tablet PC for computational power, 2D drawing, and multitouch inputs; (b) an overhead depth sensor for 3D tracking; and (c) a display to render the virtual 3D scene. As our goal is to provide a proof-of-concept for perceptually consistent tangible interactions, we found it important to minimize complexity within the current system by using off-the-shelf components. However, the system capabilities can be extended by integrating input/output media recommended in 3DUI literature [27].

The interaction space is defined as a $600 \times 440 \times 400$ mm volume, lying over the desk surface (with respect to sensor's frame

[1] Online link providing a demo of our system: https://youtu.be/fbScM4Ds3Dg

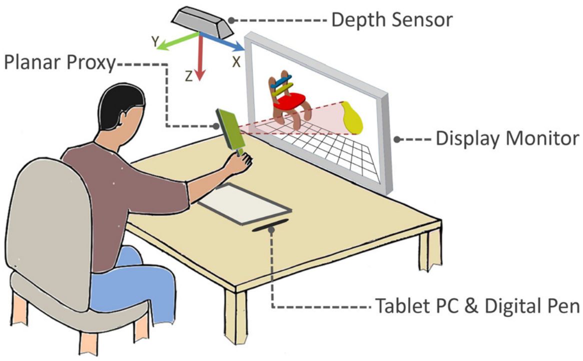

Fig. 2 Generalized setup of proposed system

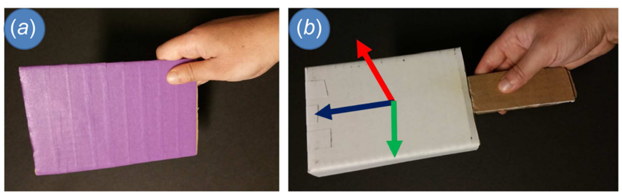

Fig. 3 Planar proxy: (*a*) preliminary version and (*b*) refined version with increased thickness and a flat handle. Arrows define local reference.

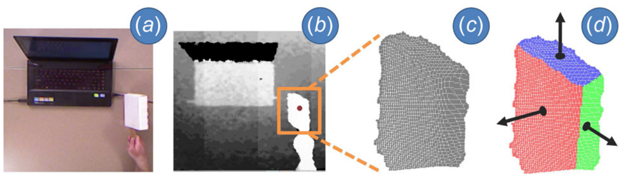

Fig. 4 Tracking proxy's motion: (*a*) interaction space viewed from depth sensor, (*b*) depth map with 3D position and bounding box, (*c*) 3D data inside bounding box, and (*d*) segmented faces with normals

of reference). The 3D scene (rendered in perspective view) spatially corresponds to this space, and has a $14 \times 7 \times 8$ unit volume. These dimensions can be adjusted based on different requirements or constraints. Initially, we had used a rectangular cardboard cutout as the planar proxy (Fig. 3(*a*)). However, we included a flat handle for ergonomic grasping and subtle 3D inputs via wrist and finger movements [28] (Fig. 3(*b*)). We also increased its thickness to prevent self-occlusions during tracking.

3.1 Tracking Proxy's Midair Motion. While we use a Kinect depth sensor (Fig. 4) to track the proxy's motion, other approaches with AR markers and webcams are also viable [26]. For position tracking, we utilize OpenNI's hand tracking functions. This position roughly lies at the proxy's center (Fig. 4(*b*)), and is measured with respect to the sensor's frame of reference. The proxy's orientation is represented by three orthogonal directions, and is tracked using the following steps.

Proxy data acquisition: An axis-aligned 3D bounding box is drawn with its center at the proxy's 3D position. The point cloud data within this box is stored as the proxy data. We empirically found a $200 \times 200 \times 170$ mm volume bounding box to be suitable for a proxy with a $100 \times 150 \times 25$ mm dimensions.

Point cloud segmentation: Laplacian smoothing is applied on the proxy data to reduce noise, and normals computed at each point. Using spherical *k*-means [29] on the normals, the data is then segmented into three (or less) planar regions. Adjoining planar regions within 15 deg are merged together. Each planar region represents a proxy face, and its normal is computed using principal component analysis (PCA).

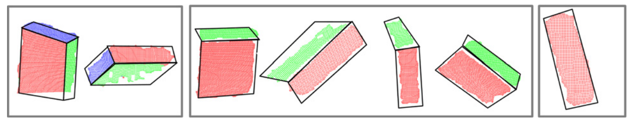

Fig. 5 Segmenting planar regions on 3D data: (left) three faces visible to sensor; (middle) two faces visible; and (right) only one face visible

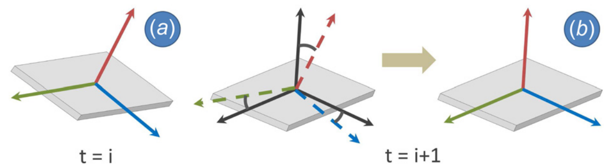

Fig. 6 Temporal coherence of orientation: (*a*) face normals at frame *i* and (*b*) assigning measured directions in frame $i+1$ to appropriate faces, based on angular proximity with normals in frame *i*

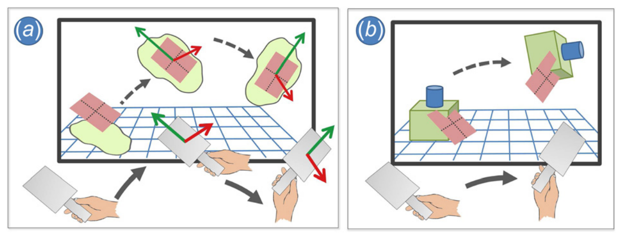

Fig. 7 Using planar cursor to clutch and manipulate: (*a*) a planar shape and (*b*) a nonplanar shape

Orientation estimation: When a principle direction is unmeasurable due to occlusions, it is inferred from the other directions. For example, when one direction is missing (Fig. 5 (middle)), it is set as the cross product of the other two. Similarly, two missing directions can be estimated using PCA on the single visible face (Fig. 5 (right)).

Temporal coherence of orientation: Tracking is initialized by holding the proxy, such that it is roughly aligned with the global frame of reference (top face horizontal, front face away from user). This allows the system to assign principle directions to correct faces on the proxy. At ensuing frames, a measured direction is matched to a face if its normal in the prior frame is closest to the direction (Fig. 6).

We apply single exponential smoothing ($\alpha = 0.3$) to remove jitter during position tracking. For the orientation, its principal directions are first represented as quaternions (with respect to the global *x*-axis) and smoothened using the double exponential function [30].

3.2 Mapping Proxy's Motion Onto Planar Cursor. Users can express *intent to interact* with the 3D scene by raising the proxy toward the sensor [31]. This activates an interactive planar cursor in the scene, which can be controlled by manipulating the proxy in midair (Fig. 7(*a*)). To stop interacting, the proxy is placed back on the desk.

We impose spatial correspondence between the interaction and modeling spaces by having the sensor face vertically downward and its *X*–*Y* axes roughly along the desk edges. This allows us to linearly map the proxy's motion onto the cursor. Here, a 5 mm linear motion of the proxy gets mapped as a 0.12 unit cursor

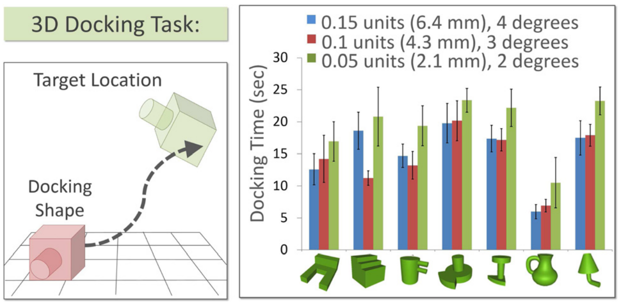

Fig. 8 Task 1: (left) docking task and (right) task completion times for seven asymmetric shapes at three levels of docking tolerances

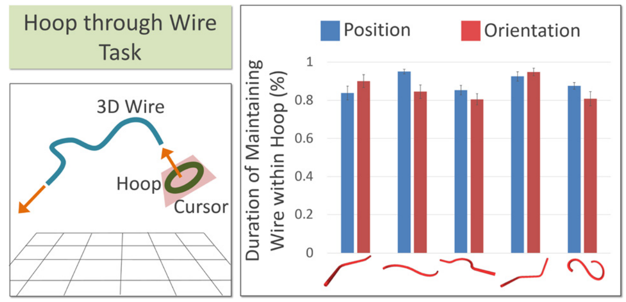

Fig. 9 Task 2: (left) guiding a circular hoop across a 3D wire and (right) duration of maintaining the wire within the hoop

displacement. The proxy's orientation can be directly applied to the cursor. For a different sensor orientation, the angular offset between its frame of reference and the desk surface must be accounted for.

A planar shape is clutched by hovering the cursor over it and applying a single tap gesture on the tablet (with nondominant hand). A clutched shape gets overlaid on the cursor and moves with the proxy (Fig. 7(*a*)). This interaction can also be generalized for nonplanar shapes (Fig. 7(*b*)), except here the cursor is firmly fixed into the shape.

4 Evaluation of Planar Proxy

We evaluated the proxy's utility toward both planar and general 3D shape manipulations. Here, we recruited 15 participants (11 male, 4 female) from engineering, natural sciences, and liberal arts. Only five participants were familiar with 3D modeling, and none had exposure to midair interactions. They were first familiarized with the interactions (15 min) and assigned the following tasks.

4.1 Task 1: Docking 3D Objects. This task evaluated spatial accuracy attainable with the proxy when configuring shapes at 3D locations. In a given trial, an asymmetrical shape was picked up and aligned with a fixed target (Fig. 8 (left)). A trial was successful if the shape's position and orientation was brought within a predefine tolerance. We used seven different shapes in each run, and tightened the tolerance between successive runs. By predefining a shape's starting and docking locations, we ensured adequate challenge, and also minimized variability during analysis. However, we randomized the sequence of the shapes to avoid learning effects.

We found that users were unable to dock shapes if the positional and orientation tolerances were less than 0.05 units (equivalent of 2.1 mm in interaction space) and 2 deg, respectively. Above these values, we observed 97.8% of the trials to be completed in under 30 s (Fig. 8 (right)), and the remaining under 45 s.

4.2 Task 2: Hoop Through 3D Wire. This task determined if the proxy can provide reasonable control of shapes along a constrained path. The objective was to guide a circular hoop along a 3D wire without touch the wire (Fig. 9 (left)). Here, the hoop was directly overlaid on the cursor, giving the impression of holding its rim. We used five trials in the study with differently shaped wires. We empirically found that a hoop with 0.5 unit inner diameter provided the right balance between task feasibility and challenge adequacy. The hoop was considered to touch the wire if (a) the distance between its center and the wire exceeded the inner hoop radius, and/or (b) the angle between its normal and the wire exceeded 35 deg. Color changes in the wire were used to notify users of such deviations, allowing them to readjust the hoop.

For each trial, we recorded the frequency of both positional and orientation deviations. Figure 9 (right) shows that on average users were able to keep the hoop within both thresholds during at least 80% of the path. By observing the two deviations separately, we intended to find if one was more dominant than the other. However, the results show that the frequency of the two kinds of deviations were similar.

4.2.1 Takeaways. Here, we demonstrated the proxy's utility in supporting both 6DOF and constrained manipulation of 3D shapes. These properties are essential during planar shape assembly, as 6DOF manipulability enables direct configuration of shapes, while controlled manipulations allow for subtle changes in the assembly for exploring different forms.

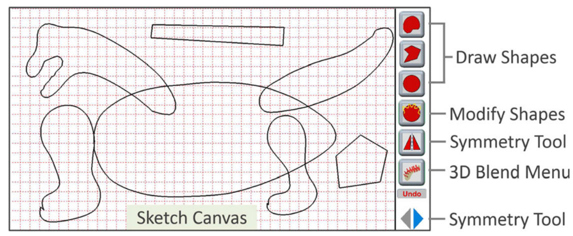

Fig. 10 Two-dimensional drawing interface displayed on the tablet

5 Proto-TAI++

In our setup (Fig. 2), planar shapes are drawn directly on the tablet using a digital pen (Fig. 10), and subsequently assembled via midair interactions with the planar proxy. Sections 5.1 and 5.2 describe the interactions used during these two modes.

5.1 Drawing Planar Shapes. Figure 11 shows the drawing interface. It contains a 2D canvas with gridlines and a set of buttons for indicating different sketch operations. All of these operations are carried out with the digital pen and are as follows.

5.1.1 Draw. Freeform curves and circles are drawn using a single stroke input, while polygons defined through their vertices (Fig. 11 (top)). Each shape is represented as a closed sequence of 2D points, which are evenly spaced using uniform resampling. Freeform curves are neatened using method described in Ref. [32]. Open curves are assumed to be closed by a straight line joining their end points.

5.1.2 Modify. The overdrawing technique [33] allows users to modify a shape's geometry, add details, or improve its appearance (Fig. 11 (middle)). It provides an intuitive mode for editing shapes and also reflects how most people modify drawings using pen-and-paper.

5.1.3 Mirror. To create a symmetric shape, a line-of-symmetry is first defined. An open curve or polygon drawn on one

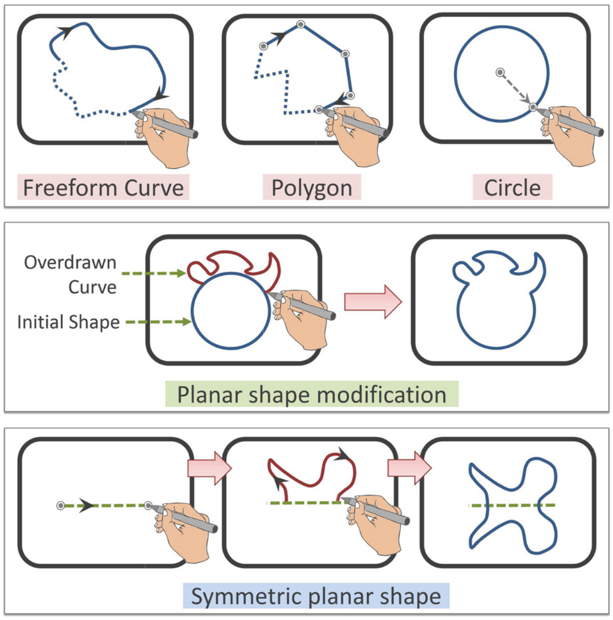

Fig. 11 Drawing operations: (top) drawing shapes; (middle) editing shapes; and (bottom) creating a symmetric shapes

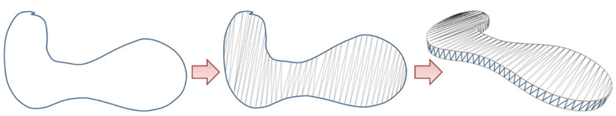

Fig. 12 Converting 2D drawing to planar mesh for use in assembly

side of this line gets mirrored across to create a complete symmetric shape (Fig. 11 (bottom)).

5.1.4 Shape Manipulation. Shapes drawn on the canvas can be moved around or resized for directly observing and adjusting relative proportions between shapes. Similarly, by using copy, modify, and delete operations, users can create variations of one shape and explore different forms.

5.1.5 Generate Planar Section. Each drawn shape can be saved as a planar mesh for use in the 3D assembly (Fig. 12). Here, constrained Delaunay triangulation first creates a 2D mesh over the shape. The mesh is then extruded by offsetting its own copy by a predefined width, and stitching the two copies with a series of triangles.

5.2 Three-Dimensional Assembly of Planar Shapes. The 3D modeling scene (Fig. 13) comprises a horizontal desk over which the saved planar shapes are laid out. To avoid clutter, only six shapes are displayed at a time, but the scroll buttons allow access to other shapes. Shadows are rendered on the desk surface to provide depth perception. The trash bin is used for discarding unwanted assembly shapes.

As shown in Fig. 14, 3D interactions are performed by coordinating the cursor's hover state (dominant hand) with multitouch inputs (nondominant hand). Visual feedback as color changes indicates proximity between the cursor and the scene elements. Here, each cell shows the proxy's motion in the interaction space, the cursor's hover state in the 3D scene, the touch input on the tablet, and their collective effect.

Touch gestures can be applied anywhere on the tablet, without directly looking at it. These include one-, two-, and three-finger taps for selection, long presses for global clutching, and two-finger pinch and twist motions for constrained manipulations.

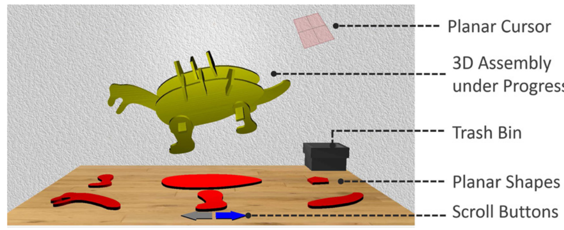

Fig. 13 Three-dimensional scene for assembling planar shapes

We used the freeglut library for multitouch sensing, and implemented touch gesture recognition based on established guidelines [34].

5.2.1 Three-Dimensional Manipulation and Configuration of Planar Shapes. Multiple copies of the shapes laid out on the desk can be individually picked up and configured into a 3D model. A clutched shape is overlaid on the cursor, such that it follows the proxy's motion (Fig. 14(*a*)). It can then be placed at a static 3D location by simply releasing it there.

Shapes that intersect with pre-existing assembly shapes are automatically adjusted to be orthogonal to their adjoining neighbors. This ensures physical connectivity during fabrication [8] and provides a structured appearance of the model. The entire assembly can also be rotated or translated at any point for adjusting the viewing direction or gaining access to occluded regions (Figs. 14(*c*) and 14(*f*)). Assembly shapes can be discarded by releasing them over the trash bin.

A clutched shape can also be manipulated along constrained DOF fine-level control. For example, if a one-finger double tap gesture (instead of single tap) is used to clutch a shape, its motion is restricted to translation only (Fig. 14(*d*)). Similarly, by applying a two-finger twist gesture, a clutched (or assembly) shape can be rotated about its planar face normal (Fig. 14(*e*)). This allows users to achieve difficult shape orientations without straining the wrist.

5.2.2 Procedural Operations. We enable procedural operations for geometric regularity, esthetic design, and structural fidelity. Here, users can create a parallel pattern of identical shapes along a linear path (Fig. 14(*g*)) or the contour of another shape (Fig. 14(*h*)). For this, a two- or three-finger single tap gesture indicates the type of patterning, and copies the shape hovered by the cursor. The proxy's motion then defines the placement of the copied shape along a constrained path. Users can also create a blended pattern between two nonidentical shapes (Fig. 14(*i*)) by first placing two end-point shapes over a base shape, and indicating the number of intermediate shapes in the blend menu (displayed on the tablet). The intermediary shapes are obtained by interpolating corresponding vertices between the two end-point shape profiles, similar to Ref. [8].

5.2.3 Shape Modification. By applying a two-finger pinch gesture, a clutched or assembly shape can be uniformly scaled (Fig. 14(*e*)). Likewise, when a 2D profile in the drawing interface is modified, the changes are reflected in the corresponding shapes in the assembly. This is analogous to *brushing-and-linking* in data visualization, where identical data within different spaces maintain a consistent representation [35].

When an assembly shape is modified, its neighbors are reconfigured such that it does not cut through or disconnect from the assembly. In Figs. 15(*a*) and 15(*b*), when shape 1 (large oval shaped base) is modified at a specific region, adjoining shape 2 (circular shape) is displaced by vector **v**, such that it maintains the same intersection point and tangential angle (θ) with shape 1. This transformation is then propagated across all shapes connecting to shape 2. To identify the connecting shapes, we apply *depth-first-search* on an undirected graph of the assembly (Fig. 15(*c*)). Here, the nodes and edges represent the shapes and connectivity between shapes.

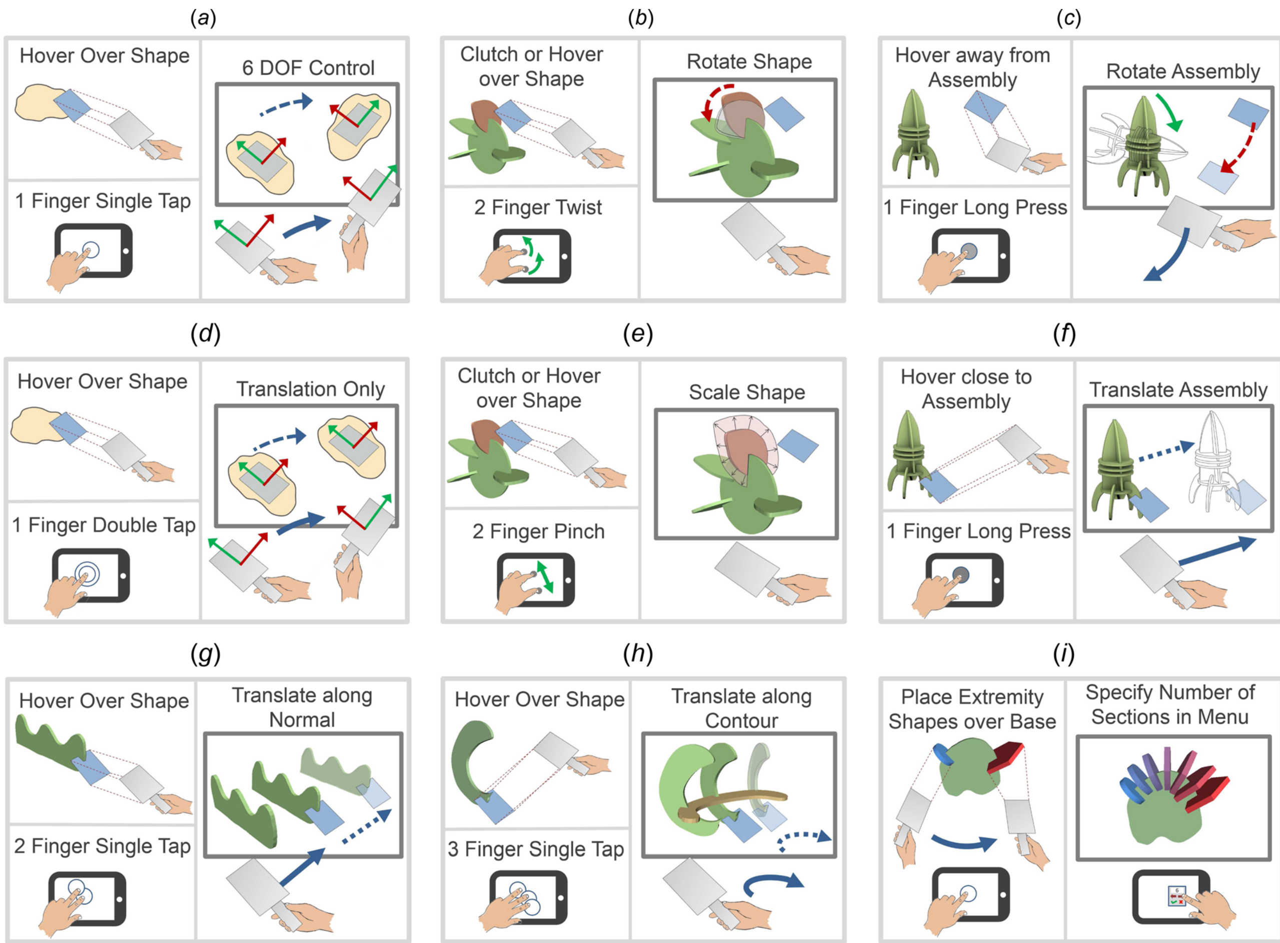

Fig. 14 Three-dimensional assembly interactions using asymmetric bimanual coordination of the proxy's motion with multi-touch gestures on the tablet. Each cell shows the hover state of the planar cursor, the multitouch input, the proxy's motion, and the outcome in the 3D scene: (*a*) clutch shape for 6DOF control, (*b*) rotate shape about normal, (*c*) rotate full assembly, (*d*) clutch shape for translation only, (*e*) uniform scale shape, (*f*) translate full assembly, (*g*) linear pattern, (*h*) contour pattern, and (*i*) sectional blending.

Shape 2 is not reconfigured if it is significantly larger (>100%) than shape 1. We impose this since larger shapes are typically used as base geometries, and displacing them could disrupt the assembly. Adjoining shapes forming a cyclic loop around shape 1 are also exempt from reconfiguration. This is to ensure that the loop structure, which could be intentional, does not get detached from shape 1.

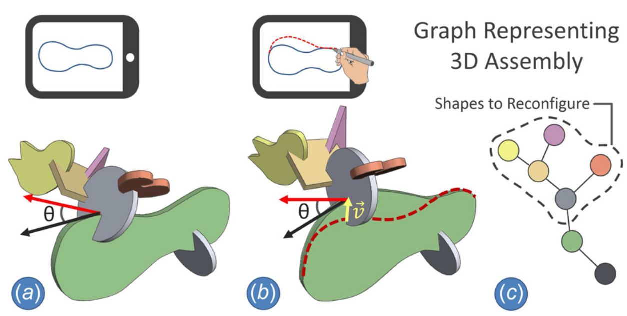

Fig. 15 (*a*) Initial assembly, (*b*) adjusted neighbors after base shape modification, and (*c*) graph structure representing the assembly

5.3 Fabrication. As shown in Figs. 16(*b*) and 16(*c*), a 3D model constructed in *Proto-TAI++* can be physically constructed using a laser cutter and cheap material like cardboard, plywood, or acrylic sheets. To support assembly of the laser cut shapes, the system automatically inserts slit joints between adjoining shapes (Fig. 16(*a*)). The joint size and location are determined by first computing the intersecting region between adjoining shapes, and then removing the volume from that region. A slit joint is generally placed on the shape with larger area, unless the larger shape already has two slits in it.

6 User Evaluation and Results

6.1 User Study 1: Comparison With *FlatFitFab*. Given that *FlatFitFab* is a proven system intended for planar shape assembly, it serves as an appropriate basis for evaluating the unit operations in *Proto-TAI++*. This comparison mainly helps us assess the proxy's efficacy in configuring planar shapes in comparison to GUI-based tools. It also allows us to compare the advantages of a collocated drawing and assembly space used in *FlatFitFab* with a separated multiview system used in *Proto-TAI++*. We frame our observations in terms of drawing and assembly precision, task completion time, and user experience. Here, we recruited another group of 12 participants (nine males, three females) from a pool similar to Sec. 4. We used a within-subjects design where a

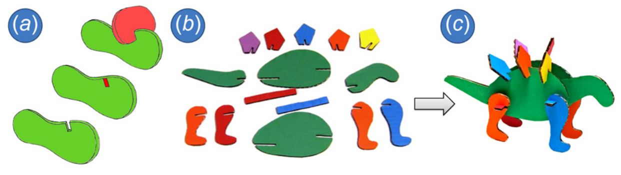

Fig. 16 (*a*) Automatic insertion of a slit joints, (*b*) laser cut shapes, and (*c*) physical assembly built using *Proto-TAI++*

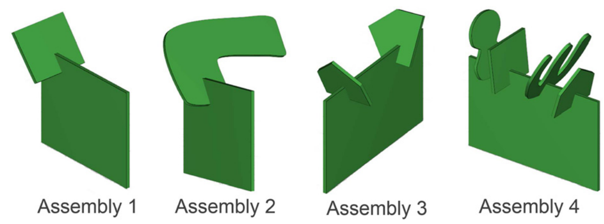

Fig. 17 Ground truth assemblies that are replicated by participants during user study 1

training session and four tasks were performed using one of the interfaces, and subsequently repeated with the other interface. To reduce learning effects, we alternated the order in which the interfaces and tasks were presented.

Given the planar shape assembly context, the tasks involved constructing four assemblies shown in Fig. 17. For each assembly, we provided a common base shape (square or rectangle) to maintain consistent physical references for measuring task accuracy. Participants were asked to draw and configure shapes mounted on this base. To cover multiple possibilities, we used eight shapes ranging from polygons to freeform curves, and also included varying levels of assembly clutter. Participants could refer to multiview images of the ground truth assemblies during the task. We evaluated the results based on the following factors.

6.1.1 Task Completion Time. Figure 18 shows statistical data pertaining to completion times of the four assemblies using both *Proto-TAI++* and *FlatFitFab*. We initially hypothesized that task completion for *Proto-TAI++* would take longer due to the separation of sketching and assembly modes.

Ho: The mean completion times of each assembly using the two interfaces are equal. ($\mu_{t_F} = \mu_{t_P}$)

Ha: The mean completion time using *FlatFitFab* is lower ($\mu_{t_F} < \mu_{t_P}$).

We first verified the normality of the completion time data using the Shapiro–Wilk test, and compared them for each assembly using a paired *t*-test. The *p*-values for each comparison were below the significance level ($\alpha = 0.05$), indicating that we cannot safely reject **Ho**. This suggests that, despite the separation of modalities in *Proto-TAI++*, users could still draw and configure planar shapes at a rate competitive with a collocated drawing-assembly medium.

6.1.2 Task Accuracy. To quantify task accuracy, we utilized four error metrics (Fig. 19) measured with respect to the ground truth shapes (Fig. 17). The first three are positional, angular, and twist errors, and they collectively represent the spatial disparity between a user-defined shape and the corresponding ground truth shape. Fixed references on the base (corner point, profile tangent) and ground truth shapes (profile normal and major axis) were used for measuring these errors. Similarly, the fourth error defines geometric dissimilarity between the two shapes, and is measured using Procrustes analysis [36]. It indicates how accurately users could reproduce a shape geometry using the two interfaces.

Figure 20 shows statistical distribution of the errors observed when drawing and assembling the eight shapes using both interfaces. Here, each shape is labeled based on which assembly they lie

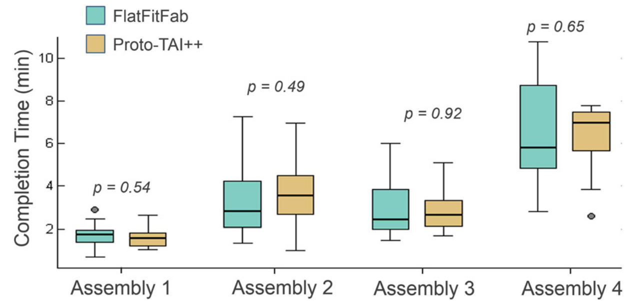

Fig. 18 Statistical distribution of task completion times using the two interfaces

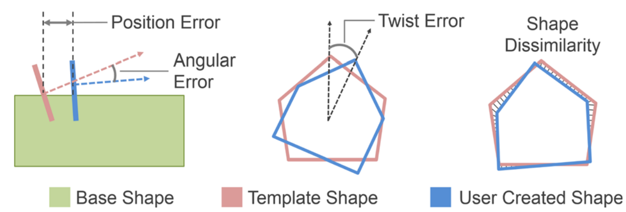

Fig. 19 Error metrics with respect to ground truth shapes: (left) positional and angular errors, (middle) twist error, and (right) shape similarity

in and their identity in that assembly (e.g., A4S3 implies third shape in assembly 4). We initially hypothesized that given the controllability of GUI interactions in *FlatFitFab*, it would result in lower error values.

H0: The mean errors using the two interfaces are equal. ($\mu_{e_F} = \mu_{e_P}$)

H1: The mean errors using *FlatFitFab* is lower ($\mu_{e_F} < \mu_{e_P}$).

Using the Shapiro–Wilk test, we checked the normality of each pairwise difference distributions in the error data. A paired *t*-test was used for normal distributions, and a Wilcoxon signed-ranked test for those without. The resulting *p*-values are shown above their respective box plots in Fig. 20. Given that most *p*-values for shape dissimilarity are greater than $\alpha = 0.05$, we cannot conclude that a collocated modeling space in *FlatFitFab* enabled more accurate reproduction of planar shapes.

For spatial configurations, we found that positional errors were comparable between the two interfaces. However, angular and twist errors for several shapes were larger in *Proto-TAI++*. While participants could position the shapes at 3D targets with reasonable accuracy, it was more difficult for them to define shape orientations. This could be attributed to the inability of midair interactions to support fine-level manipulations required for orientating shapes. Mechanical constraints of the hand could also play a role in limiting the rotational movement.

6.1.3 Subjective User Feedback. Figure 21 compares the ease-of-use of *Proto-TAI++* against *FlatFitFab*, based on a five-point Likert scale questionnaire. While participants were able to easily learn both systems, we found notable differences in how they perceived specific components. For example, they expressed difficulty with in situ shape creation in *FlatFitFab* due to visual clutter: "previously drawn shapes would be in the way of seeing the current shape." Some also pointed out that the collocated 2D–3D modeling spaces "required me to carefully plan ahead in terms of (concurrent) placement and shape creation." In contrast, most users found it intuitive to first draw the planar shapes on the 2D canvas and assemble them later. As one indicated, "it was natural to follow the procedure." Additionally, adjusting relative proportions between shapes was found to be easier on the 2D canvas than the 3D assembly ("it allows me to see the size differences between the things I drew and the ones I want to draw next").

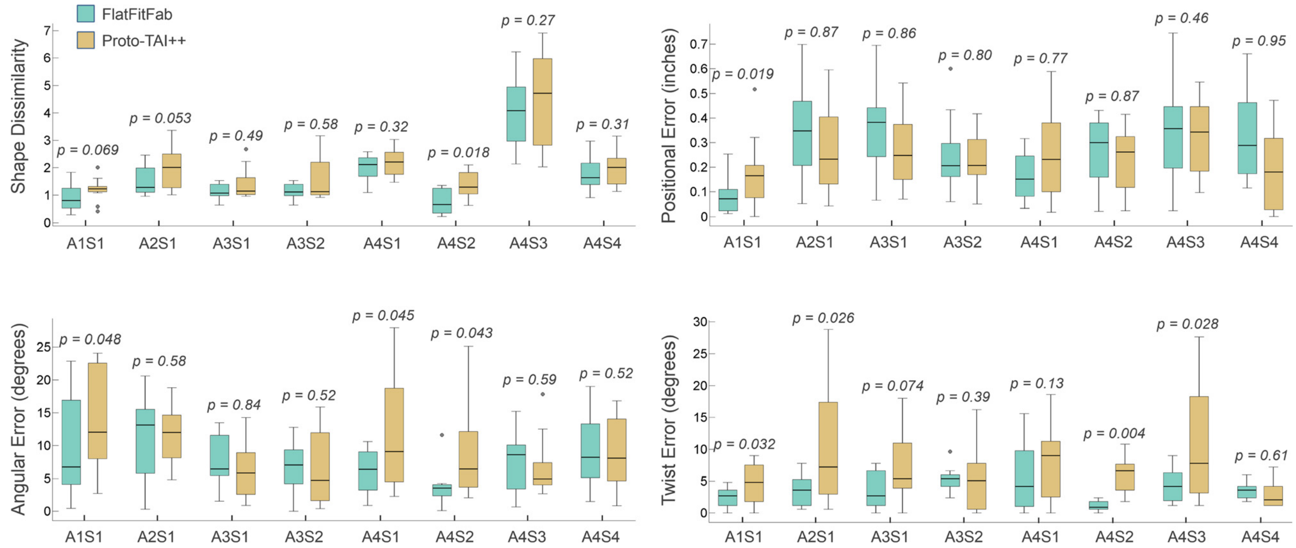

Fig. 20 Statistical distribution for errors on user-defined shapes measured with respect to corresponding ground truth shapes. Each pair of box plots shows results for a given shape using the two interfaces. Pairwise *t*-test results (*p*-values) for comparing errors are shown above each pair. The shapes are labeled according to the assembly they lie on and their id within the assembly (e.g., A4S3).

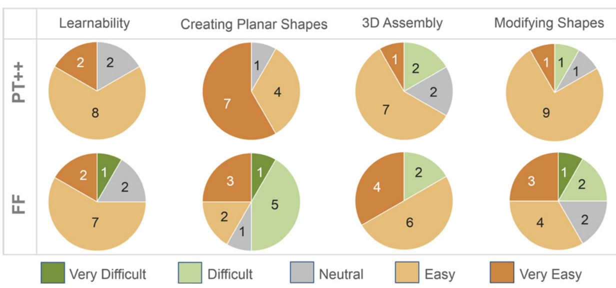

Fig. 21 Five-point Likert scale results on usability of two interfaces. The rows correspond to interfaces and each column indicates a given usability metric. Each number represents how many participants selected a given option in the scale.

However, several participants appreciated not having to switch between sketching and assembly modes in *FlatFitFab*.

Most participants could easily assemble planar shapes in Proto-TAI++, and appreciated use of suggestive actions for 3D inputs: "(by) holding the cardboard (it) was easy to control the shapes"; "it felt better moving the shapes in space with my hands"; "I could get a one-one mapping using the cardboard." However, some also commented on the lack of precise control with the proxy: "I would have also liked an alternate more precise control mechanism in a few instances." In contrast, many liked that the 3D widgets in FlatFitFab enabled "precise adjustments or fine tuning of the (shape's) position."

6.2 User Study 2: Creative Expression. This study evaluated *Proto-TAI++*'s support toward creative expression of ideas. We recruited 12 (nine male, three female) designers from engineering and industrial design. None of them had prior experience with midair interactions, but most were familiar with 3D modeling. In each session, participants were given a 15 min tutorial and familiarized with 3D planar shape assembly. They were then assigned two tasks (15–20 min each). The first task involved gaining familiarity with the system by constructing a familiar object like a chair, while the second task was open ended requiring them to both conceive and construct a design.

Figure 22 shows models created in an average time of 10 min and 36 s (drawing: 2 min and 30 s; assembly: 8 min and 6 s). Each

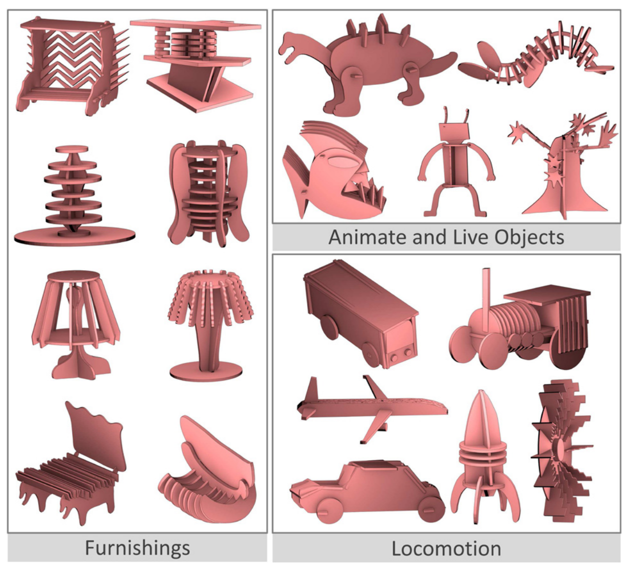

Fig. 22 Planar shape assembly models constructed by participants using *Proto-TAI++*. The models are grouped by broad level categories observed in the study.

model is identifiable and shows more complexity and details than our prior system [7]. This was achieved mainly through advanced operations added in *Proto-TAI++*. Figure 23(*a*) indicates that users found *Proto-TAI++* to support creative expressiveness. Most were able to reasonably externalize their mental ideas, and appreciated the ability to freely switch between drawing and assembly modes to explore different forms. As two users expressed: "I enjoyed creating a concept with no formal or constraining rules," and "I found myself alternating between sketching and assembly in different frequencies."

As shown in Fig. 23(*b*), users did not find issues coordinating concurrent midair and multitouch inputs. During training, they were only told which gestures to use and not given time to memorize them. But despite that, most users could recall the gestures, but few needed reminders during early stages ("I could not always remember touch controls").

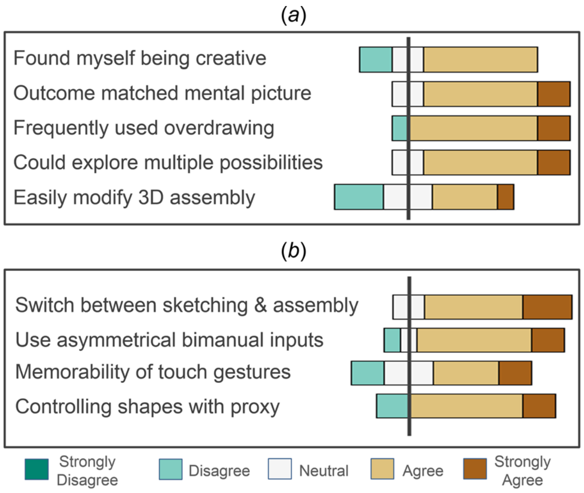

Fig. 23 Five-point Likert scale-based feedback: (*a*) factors influencing participants' self-evaluated creative expression when using *Proto-TAI*++ and (*b*) ease of use of specific system features and interactions

7 Discussion, Implications, and Future Directions

In *Proto-TAI*++, we demonstrated how a tangible proxy can be used as a physical embodiment of virtual modeling elements in a 3D design scenario. Given our results, we find a scope for extending this concept toward proxies with other kinds of shapes (e.g., spherical, cubical, conical, etc.), material properties (e.g., density, surface texture, etc.), functionalities, and semantic identities. The broader goal is to investigate different affordances and perceptually consistent metaphors such proxies can provide within diverse 3D contexts like design, architecture, data exploration, and education.

Given that users could easily coordinate asymmetric bimanual gestures on separate modalities, augmenting midair inputs with multitouch gestures can be particularly useful when using noninstrumented controllers for 3D interactions. However, a more elaborate study is necessary to understand the general applicability and user perceptions on such input mechanisms. This study can include unique combinations of multitouch media and physical proxies, with the aim to establish design guidelines and explore interesting 3D contexts. Just as with midair inputs, it would also make sense to explore touch gestures that are perceptually consistent with their respective operations. This could improve touch gesture memorability.

While the proxy provided reasonable 3D control, its precision cannot be compared to GUI tools. However, using context-specific constraints enabled us to interpret freeform inputs as structured 3D operations. For example, orthogonality between adjoining shapes compensated for limited orientation accuracy, leading to improved appearance and fabricability. Similarly, constrained procedural operations enabled geometric regularity in the models. These constraints in *Proto-TAI*++ provided improved the complexity and level-of-detail in the resulting models (Fig. 22) compared to the earlier version [7], and was achieved without the same interactive scheme. Thus, we find value in exploring context-specific constraints within other midair interaction scenarios.

In 3D modeling, datum planes frequently serve as scaffolds for 2D sections that define complex 3D geometries. Given the proxy's ability to arrange planar shapes in 3D, it could serve as a tangible datum plane for efficiently arranging 2D sections for 3D surface and volume construction. This metaphor extends *Proto-TAI*++ beyond planar shape abstractions and enables users to construct extrusions, swept surfaces, and inflation models.

Given its planarity, it would be interesting to use the tablet (or a smaller smartphone) itself as the proxy, enabling both shape creation and manipulation via the proxy. This can lead to a stronger correspondence between the proxy and virtual shapes. Additionally, the touch sensitive surface precludes the need for asymmetric bimanual inputs, and focuses all interactions on one modality. We can also leverage the GUI display for precise inputs such as menu navigation, alphanumerical entry, sliders, buttons, etc.

As our system mainly provides a proof-of-concept for perceptually consistent tangible interactions, we used off-the-shelf components to reduce complexity. However, future extensions could integrate more advanced features such as haptic rendering, auditory feedback, and stereoscopic displays to support more 3D immersive contexts [27]. Additionally, these can also help reduce some of our current limitations such as arm fatigue due to lack of collision feedback, and limited depth perception due to use of a flat screen display.

8 Conclusions

We explored physical proxies as a means for achieving tangible 3D interactions where there is perceptual correspondence between users' input actions on the proxy and the intended outcome in the virtual 3D space. Here, we took an application-based approach by demonstrating *Proto-TAI*++, a multimodal pen-based drawing and planar proxy-based midair assembly system for constructing planar shape assemblies in virtual space (extended version of our prior work [7]). Through user studies, we showed the proxy's efficacy toward 3D manipulation of virtual shapes, and demonstrated *Proto-TAI*++ as a tool for novice designers to express creative ideas. We find this work to have broad implications in both 3D design and human computer interactions. While this paper provides a basis for extending planar shape abstractions into more complex 3D modeling contexts, the concept of perceptually consistent tangible interactions opens up interesting possibilities for novel interaction and system designs. We hope our work will lead toward a larger framework, aimed at integrating physical reality with intuitive and seamless interactions with the virtual world.

Acknowledgment

This work was supported by NSF CMMI-ESD (1538868) and NSF CPS:Synergy (1329979).

References

[1] Fitzmaurice, G. W., Ishii, H., and Buxton, W. A., 1995, "Bricks: Laying the Foundations for Graspable User Interfaces," SIGCHI Conference on Human Factors in Computing Systems, Denver, CO, May 7–11, pp. 442–449.
[2] Ishii, H., and Ullmer, B., 1997, "Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms," SIGCHI Conference on Human Factors in Computing Systems, Atlanta, GA, Mar. 22–27, ACM, New York, pp. 234–241.
[3] Hinckley, K., Pausch, R., Goble, J. C., and Kassell, N. F., 1994, "Passive Real-World Interface Props for Neurosurgical Visualization," SIGCHI Conference on Human Factors in Computing Systems, Boston, MA, Apr. 24–28, ACM, New York, pp. 452–458.
[4] Arisandi, R., Otsuki, M., Kimura, A., Shibata, F., and Tamura, H., 2014, "Virtual Handcrafting: Building Virtual Wood Models Using ToolDevice," Proc. IEEE, **102**(2), pp. 185–195.
[5] Oh, Y., Johnson, G., Gross, M., and Do, E. Y.-L., 2006, "The Designosaur and the Furniture Factory," Design Computing and Cognition '06, Springer, Dordrecht, The Netherlands, pp. 123–140.
[6] Johnson, G., Gross, M., Do, E. Y.-L., and Hong, J., 2012, "Sketch It, Make It: Sketching Precise Drawings for Laser Cutting," SIGCHI Extended Abstracts on Human Factors in Computing Systems, Austin, TX, May 5–10, ACM, New York, pp. 1079–1082.
[7] Piya, C., and Ramani, K., 2014, "Proto-TAI: Quick Design Prototyping Using Tangible Assisted Interfaces," ASME Paper No. DETC2014-35442.
[8] McCrae, J., Umetani, N., and Singh, K., 2014, "FlatFitFab: Interactive Modeling With Planar Sections," 27th Annual ACM Symposium on User Interface Software and Technology, Honolulu, HI, Oct. 5–8, ACM, New York, pp. 13–22.
[9] Rivers, A., Durand, F., and Igarashi, T., 2010, "3D Modeling With Silhouettes," SIGGRAPH '10 Special Interest Group on Computer Graphics and Interactive Techniques Conference, Los Angeles, CA, July 26–30, ACM, New York, Paper No. 109.
[10] McCrae, J., Singh, K., and Mitra, N. J., 2011, "Slices: A Shape-Proxy Based on Planar Sections," ACM Trans. Graphics, **30**(6), p. 168.

[11] Hildebrand, K., Bickel, B., and Alexa, M., 2012, "crdbrd: Shape Fabrication by Sliding Planar Slices," Comput. Graphics Forum, **31**(2 Pt. 3), pp. 583–592.
[12] Mueller, S., Lopes, P., and Baudisch, P., 2012, "Interactive Construction: Interactive Fabrication of Functional Mechanical Devices," 25th Annual ACM Symposium on User Interface Software and Technology, Cambridge, MA, Oct. 7–10, ACM, New York, pp. 599–606.
[13] Saul, G., Lau, M., Mitani, J., and Igarashi, T., 2011, "SketchChair: An All-in-One Chair Design System for End Users," Fifth International Conference on Tangible, Embedded, and Embodied Interaction, Funchal, Portugal, Jan. 22–26, ACM, New York, pp. 73–80.
[14] Marquardt, N., Jota, R., Greenberg, S., and Jorge, J. A., 2011, "The Continuous Interaction Space: Interaction Techniques Unifying Touch and Gesture on and Above a Digital Surface," Human-Computer Interaction—INTERACT 2011, Springer, Lisbon, Portugal, pp. 461–476.
[15] Huppmann, D., Luderschmidt, J., Haubner, N., Lehmann, S., Dörner, R., and Schwanecke, U., 2012, "Exploring and Evaluating the Combined Multi-Touch and In-the-Air Tabletop Interaction Space," Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR, C. Geiger, J. Herder, and T. Vierjahn, ed., Shaker Verlag, Aachen, Germany, Vol. 9, pp. 37–48.
[16] De Araújo, B. R., Casiez, G., Jorge, J. A., and Hachet, M., 2013, "Mockup Builder: 3D Modeling on and Above the Surface," Comput. Graphics, **37**(3), pp. 165–178.
[17] Guiard, Y., 1987, "Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model," J. Mot. Behave., **19**(4), pp. 486–517.
[18] Wang, R., Paris, S., and Popović, J., 2011, "6D Hands: Markerless Hand-Tracking for Computer Aided Design," 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, Oct. 16–19, ACM, New York, pp. 549–558.
[19] Vinayak, Murugappan, S., Liu, H., and Ramani, K., 2013, "Shape-It-Up: Hand Gesture Based Creative Expression of 3D Shapes Using Intelligent Generalized Cylinders," Comput.-Aided Des., **45**(2), pp. 277–287.
[20] Oikonomidis, I., Kyriazis, N., and Argyros, A. A., 2011, "Efficient Model-Based 3D Tracking of Hand Articulations Using Kinect," 22nd British Machine Vision Conference, BMVC 2011, Dundee, Scotland, UK, Aug. 29–Sept. 2, Vol. 1, pp. 101.1–101.11.
[21] Song, P., Goh, W., Hutama, W., Fu, C., and Liu, X., 2012, "A Handle Bar Metaphor for Virtual Object Manipulation With Mid-Air Interaction," CHI Conference on Human Factors in Computing Systems, Austin, TX, May 5–10, ACM, New York, pp. 1297–1306.
[22] Jerald, J., Mlyniec, P., Yoganandan, A., Rubin, A., Paullus, D., and Solotko, S., 2013, "MakeVR: A 3D World-Building Interface," IEEE Symposium on 3D User Interfaces (3DUI), Orlando, FL, Mar. 16–17, pp. 197–198.
[23] Schkolne, S., Pruett, M., and Schröder, P., 2001, "Surface Drawing: Creating Organic 3D Shapes With the Hand and Tangible Tools," Human Factors in Computing Systems, Seattle, WA, Mar. 31–Apr. 5, ACM, New York, pp. 261–268.
[24] Mine, M., Yoganandan, A., and Coffey, D., 2014, "Making VR Work: Building a Real-World Immersive Modeling Application in the Virtual World," Symposium on Spatial User Interaction, Honolulu, HI, Oct. 4–5, ACM, New York, pp. 80–89.
[25] Sheng, J., Balakrishnan, R., and Singh, K., 2006, "An Interface for Virtual 3D Sculpting Via Physical Proxy," International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, GRAPHITE '06, Kuala Lumpur, Malaysia, Nov. 28–Dec. 2, Vol. 6, pp. 213–220.
[26] Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., and Tachibana, K., 2000, "Virtual Object Manipulation on a Table-Top AR Environment," IEEE and ACM International Symposium on Augmented Reality (ISAR), Munich, Germany, Oct. 5–6, pp. 111–119.
[27] Bowman, D., Kruijff, E., LaViola, J. J., Jr., and Poupyrev, I., 2004, *3D User Interfaces: Theory and Practice, CourseSmart eTextbook*, Addison-Wesley, Boston, MA.
[28] Zhai, S., Milgram, P., and Buxton, W., 1996, "The Influence of Muscle Groups on Performance of Multiple Degree-of-Freedom Input," SIGCHI Conference on Human Factors in Computing Systems, Vancouver, BC, Canada, Apr. 13–18, ACM, New York, pp. 308–315.
[29] Buchta, C., Kober, M., Feinerer, I., and Hornik, K., 2012, "Spherical *k*-Means Clustering," J. Stat. Software, **50**(10), pp. 1–22.
[30] LaViola, J. J., 2003, "Double Exponential Smoothing: An Alternative to Kalman Filter-Based Predictive Tracking," Workshop on Virtual Environments, Zurich, Switzerland, May 22–23, ACM, New York, pp. 199–206.
[31] Schwarz, J., Marais, C. C., Leyvand, T., Hudson, S. E., and Mankoff, J., 2014, "Combining Body Pose, Gaze, and Gesture to Determine Intention to Interact in Vision-Based Interfaces," CHI Conference on Human Factors in Computing Systems, Toronto, ON, Canada, Apr. 26–May 1, ACM, New York, pp. 3443–3452.
[32] Thiel, Y., Singh, K., and Balakrishnan, R., 2011, "Elasticurves: Exploiting Stroke Dynamics and Inertia for the Real-Time Neatening of Sketched 2D Curves," 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, Oct. 16–19, ACM, New York, pp. 383–392.
[33] Baudel, T., 1994, "A Mark-Based Interaction Paradigm for Free-Hand Drawing," 7th ACM Symposium on User Interface and Software Technology, Marina del Rey, CA, Nov. 2–4, ACM, New York, pp. 185–192.
[34] Olafsdottir, H., and Appert, C., 2014, "Multi-Touch Gestures for Discrete and Continuous Control," International Working Conference on Advanced Visual Interfaces, Como, Italy, May 27–29, ACM, New York, pp. 177–184.
[35] Eick, S. G., and Wills, G. J., 1995, "High Interaction Graphics," Eur. J. Oper. Res., **81**(3), pp. 445–459.
[36] Gower, J. C., 1975, "Generalized Procrustes Analysis," Psychometrika, **40**(1), pp. 33–51.