

Easy Modeling of Man-Made Shapes in Virtual Reality

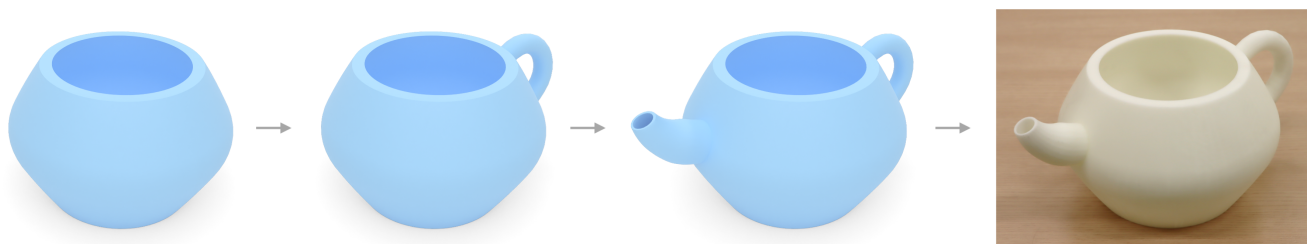
Haoyu Tang¹Fancheng Gao¹Kennedy Tsu Wei Choo¹Bernd Bickel²Peng Song¹¹Singapore University of Technology and Design, Singapore²ETH Zürich, Switzerland

Figure 1: We present a VR tool that enables general users to interactively model man-made shapes for personalized fabrication. Our tool models each shape part (i.e., body, handle, and spout of the teapot) as a generalized cylinder and iteratively assembles them to form the target shape. The 3D printed teapot is usable since each part is modeled with a correct shape type, e.g., shell-like shape for the spout.

Abstract

Virtual Reality (VR) offers a promising platform for modeling man-made shapes by enabling immersive, hands-on interaction with these 3D shapes. Existing VR tools require either a complex user interface or a post-processing to model fabricable man-made shapes. In this paper, we present a VR tool that enables general users to interactively model man-made shapes for personalized fabrication, simply by using four common hand gestures as the interaction input. This is achieved by proposing an approach that models complex man-made shapes using a small set of geometric operations, and then designing a user interface that intuitively maps four common hand gestures to these operations. In our shape modeling approach, each shape part is modeled as a generalized cylinder with a specific shape type and iteratively assembled in a structure-aware manner to form a fabricable and usable man-made shape. In our user interface, each hand gesture is associated with a specific kind of interaction tasks and is intelligently utilized for performing the small set of operations to create, edit, and assemble generalized cylinders. A user study was conducted to demonstrate that our VR tool allows general users to effectively and creatively model a variety of man-made shapes, some of which have been 3D printed to validate their fabricability and usability.

CCS Concepts

• **Computing methodologies** → **Modeling and simulation**; • **Human-centered computing** → **Interaction design**;

1. Introduction

3D modeling of man-made shapes remains a core task in computer graphics, personalized design, and digital fabrication. Virtual Reality (VR) offers a promising platform for general users to interactively model man-made shapes due to two advantages. First, VR allows users to view and interact with man-made shapes at actual scale in a fully 3D environment, improving depth perception and spatial awareness. Second, VR enables direct 3D interaction with man-made shapes using hand gestures or controllers, mimicking real-world actions like grabbing, bending, and twisting.

Many existing VR tools [Til21, LLB24] focus on modeling man-made shapes for conceptualization, and thus require extensive post-processing on the geometry [RRS19] before they can be fabri-

cated (e.g., by 3D printing) and used in the real world. Other VR tools [Gra25, Goo24] allow users to model fabricable man-made shapes by interactively modeling and combining different kinds of shape primitives. However, they require users to perform a variety of modeling operations via a complex user interface (usually with 3D menus), including selecting shape primitives to create, specifying shape parameters, applying geometric constraints such as symmetry, and transforming the created primitives etc. The tremendous modeling operations together with the complex user interface make these VR tools hard to learn and use for general users.

In this paper, our goal is to develop an *easy-to-use* VR tool that enables general users to interactively model man-made shapes for *personalized fabrication*, simply by using a few common hand gestures as the interaction input. In particular, we focus on modeling

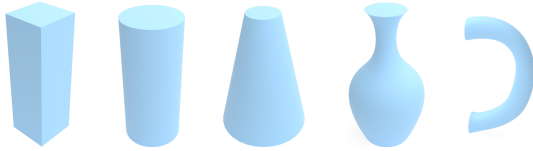


Figure 2: A variety of 3D shapes that can be modeled by using a single generalized cylinder.

daily man-made shapes that can be used for holding, pouring, storing, supporting, transferring, and entertaining etc. To ensure fabricability of the man-made shape, each shape part should be modeled as a closed manifold surface and all the shape parts should be connected without gaps. To ensure usability of the man-made shape, each shape part should have a correct shape type; e.g., a teapot consists of three shape parts, a body with a container-like shape, a spout with a shell-like shape, and a handle with a solid shape; see Figure 1. In addition, some geometric constraints have to be satisfied among the shape parts for usability; e.g., coplanarity of a teapot's handle and spout for pouring.

To achieve the goal, our idea is to first devise an approach that allows modeling complex man-made shapes using a small set of geometric operations, and then design a user interface that intuitively maps a few common hand gestures to the modeling operations. In the shape modeling, we propose to model each shape part as a *generalized cylinder* and assemble multiple generalized cylinders in a *structure-aware* manner [MWZ*13] to form a fabricable and usable man-made object. We choose generalized cylinder as the shape primitive since it has a compact geometric representation, is able to model a wide variety of shapes (see Figure 2), and can be easily processed to model different shape types (see Figure 1). In the user interface, our interaction design allows general users to use only four common hand gestures for performing the small set of operations for creating, editing, and assembling generalized cylinders. This is achieved by associating each hand gesture with a specific kind of interaction tasks and intelligently performing the geometric operations based on the interaction context.

Overall, this paper makes the following contributions:

- We propose an approach suitable for modeling man-made shapes in VR by modeling each shape part as a generalized cylinder and assembling them in a structure-aware manner; see Section 3.
- We design an easy-to-use user interface that enables general users to model fabricate and usable man-made shapes in VR by simply using four common hand gestures; see Section 4.

We show that our VR tool allows modeling man-made shapes with a variety of functionalities and forms, some of which have been 3D printed to validate their fabricability and usability. We conducted a user study to compare our VR tool with an existing one [Gra25], demonstrating that our VR tool enables general users to effectively and creatively model man-made shapes.

2. Related Work

Interactive 3D shape modeling in VR. Interactive 3D shape modeling in VR can be broadly classified into three approaches: sketch-based, volume-based, and surface-based approaches. In the sketch-based approach, one creates freeform 3D shapes in

VR by drawing 3D sketches [SPS01, DGK*20, JZF*21, Til21] or curves [YAS*21] with a moving hand/controller and then computing a manifold surface that fits the sketches/curves [RRS19]. In the volume-based approach, a virtual object is modeled as a solid and sculpting operations are applied on the solid to deform it until reaching a desired shape. A number of existing VR applications use this approach for 3D shape modeling, including SculptVR [Scu25], Adobe Medium [Ado25], Shapelab [Sha25], and Kodon [Kod25]. Besides sculpting, this approach also creates 3D objects with complex shapes via interactively combining simple primitive shapes, e.g., by using Constructive Solid Geometry (CSG) boolean operations [CXX*24, ZTZ*22] or surface blending [NUK98, ZSY*17]. In the surface-based approach, a 3D shape is typically represented as a parametric surface and one creates a 3D shape in VR by interactively transforming and/or deforming control curves of the parametric surface, including NURBS surface [LM04, CRV20, Gra25], swept surface [ACJ12, VMLR13, MGS17], and surface represented by a curve network [LLB24]. Besides that, a 3D shape also can be represented as a low-resolution polygon mesh and one interactively creates a 3D shape via moving vertices of the mesh [KAHF05, Goo24]. Please refer to [Cha24] for a comprehensive survey on interactive 3D shape modeling in VR.

The above existing VR tools enable easy modeling of 3D shapes for virtual environments, which however are usually not fabricable due to disconnected shape features and/or non-manifold geometry. To model fabricable man-made shapes, they typically rely on a complex user interface [Gra25, Goo24] and/or post-processing of the modeled shapes [RRS19]. In contrast, our VR tool aims to model daily man-made shapes for personalized fabrication and its geometric modeling approach and easy-to-use user interface enable general users to effectively create these shapes.

Shape modeling with generalized cylinders. A generalized cylinder is a geometric shape formed by sweeping a 2D curve (called the cross section) along a 3D space curve (called the spine). In some definitions, the cross-section can also scale, rotate, or deform as it moves along the spine. Due to the ubiquity of generalized cylinders in man-made objects (e.g., potteries, toys, and furniture), they have been widely used in a variety of shape modeling tasks, including surface reconstruction [LLZM10, CZS*13, YHZ*14], shape deformation [YK06, HYC*05], shape decomposition [MPS*04, GMP*12, ZYH*15], and shape editing [ZFCO*11, FCSF16, ZCHH24]. In particular, generalized cylinders have been used for modeling different kinds of man-made objects such as paper-craft models [MGE08], personalized positive airway pressure (PAP) masks [LWS*24], and pottery-like objects [VMLR13].

Among the above works, our work is closely related to [VMLR13] that interactively models pottery-like objects using freehand gestures tracked by a depth camera. In [VMLR13], each object is modeled as a single generalized cylinder, which greatly limits the complexity of shapes that can be modeled. Moreover, their modeled shapes are used for conceptualization, many of which are not fabricable or usable in the real world. In contrast, our tool allows users to interactively model each shape part as a generalized cylinder, specify its shape type (i.e., solid, shell-like, or container-like shapes), and iteratively assemble them in a structure-aware manner to form a fabricable and usable man-made shape.

3. Modeling Man-Made Shapes

To model a man-made shape, we first model each shape part as a generalized cylinder (Section 3.1). Next, we allow a family of editing operations on a generalized cylinder in order to achieve a desired shape (Section 3.2). Lastly, we assemble multiple generalized cylinders iteratively to model a complex man-made shape and allow editing the shape in a structure-aware manner (Section 3.3).

3.1. Modeling a Generalized Cylinder

A generalized cylinder can be modeled by sweeping a cross-sectional curve with varying shape along a spine curve [VMLR13]. However, a surface modeled in this way may not be easy to edit. Instead, we let the user input as a set of key cross-sectional curves specified in the global coordinate frame, and model a generalized cylinder as a *parametric surface* that *smoothly interpolates* these key cross-sectional curves.

Key cross-sectional curves. We represent the input key cross-sectional curves as planar closed NURBS curves in the global coordinate frame using:

$$\mathbf{C}_k(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_{i,k}, \quad k = 1, 2, \dots, K \quad (1)$$

where k denotes the index of the key cross-sectional curve, K is the total number of key cross-sections, $N_{i,p}$ is the basis functions of degree p , $\mathbf{P}_{i,k}$ is a control point of curve $\mathbf{C}_k(u)$, and n is the number of control points of curve $\mathbf{C}_k(u)$. Note that the control points $\{\mathbf{P}_{i,k}\}$ are expressed using homogeneous coordinates, where the fourth component is the weight $w_{i,k}$. To simplify computation, we assume that the key cross-sectional curves in Equation 1 are with the same degree and knot vector.

Lofting. We model a generalized cylinder by applying a lofting algorithm [PT97] to smoothly interpolate the key cross-sectional curves; see Figure 3(a) for an example. The resulting generalized cylinder is a NURBS surface represented using:

$$\begin{aligned} \mathbf{S}(u, v) &= \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) M_{j,q}(v) \mathbf{P}_{i,j} \\ \text{s.t. } \mathbf{S}(u, v_k) &= \mathbf{C}_k(u), \quad k = 1, 2, \dots, K \end{aligned} \quad (2)$$

where $N_{i,p}$ and $M_{j,q}$ are the basis functions of degree p and q in the u - and v -directions, respectively, n and m are the number of control points in the u - and v -directions, respectively, and $\mathbf{P}_{i,j}$ is a control point of the surface expressed using homogeneous coordinates.

Spine curve. The spine curve is defined by the center points of the key cross sections as well as the lofting procedure in Equation 3. The spine curve is represented using:

$$\begin{aligned} \mathbf{A}(v) &= \sum_{j=0}^m M_{j,q}(v) \mathbf{P}_j \\ \text{s.t. } \mathbf{A}(v_k) &= \mathbf{O}_k, \quad k = 1, 2, \dots, K \end{aligned} \quad (3)$$

where \mathbf{P}_j is a control point of the spine curve, \mathbf{O}_k is the center point of the key cross-sectional curve $\mathbf{C}_k(u)$, and $\mathbf{A}(v_k)$ is called a key point on the spine curve. In this paper, we assume that the spine

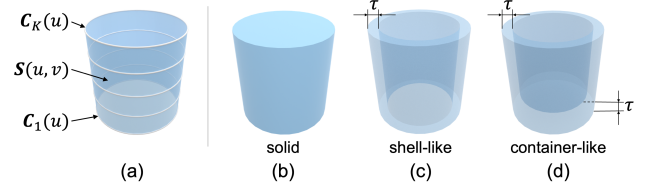


Figure 3: Modeling three types of shapes from (a) a generalized cylinder: (b) solid, (c) shell-like, and (d) container-like shape.

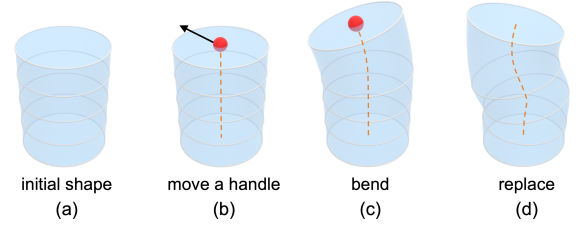


Figure 4: Given (a) a generalized cylinder, our approach allows editing its shape by (b&c) bending the spine curve via selecting and moving a key point (in red color) on the curve and (d) replacing the whole spine curve.

curve is a planar curve for ease of interaction with the fabricated generalized cylinder such as grasping.

Shape type. We specify a shape type for a generalized cylinder $\mathbf{S}(u, v)$ to transform it into a usable shape or shape part. We support three shape types in this paper.

1. **Solid shape.** By default, a generalized cylinder has a solid shape; see Figure 3(a&b). The shape surface is composed of three faces: a side face $\mathbf{S}(u, v)$, a top face enclosed by $\mathbf{C}_K(u)$, and a bottom face enclosed by $\mathbf{C}_1(u)$. Typical examples are a teapot's handle and a chair leg.
2. **Shell-like shape.** A shell-like shape is obtained by thickening the surface $\mathbf{S}(u, v)$ along the inward direction with thickness τ ; see Figure 3(c). Typical examples are a teapot's spout and a pipe.
3. **Container-like shape.** A container-like shape is a shell-like shape plus a solid top (bottom) obtained by thickening the face enclosed by $\mathbf{C}_K(u)$ ($\mathbf{C}_1(u)$) with thickness τ ; see Figure 3(d). Typical examples are a teapot's body and a vase.

3.2. Editing a Generalized Cylinder

To achieve a desired shape, we allow editing a generalized cylinder by taking its spine curve and key cross-sectional curves as handles.

Editing via the spine curve. We provide two editing operations via the spine curve.

1. **Bending.** To bend a generalized cylinder, we first select and translate a key point $\mathbf{A}(v_h)$, $h \in [1, K]$ on the spline curve in the plane that the spine curve $\mathbf{A}(v)$ lies on. After the translation, the spine curve $\mathbf{A}(v)$ is deformed such that it interpolates the updated key points $\{\mathbf{A}(v_k)\}$. We update the surface $\mathbf{S}(u, v)$ by recomputing the local frames for each key cross section using the rotation minimizing frame [WJZL08] along the deformed

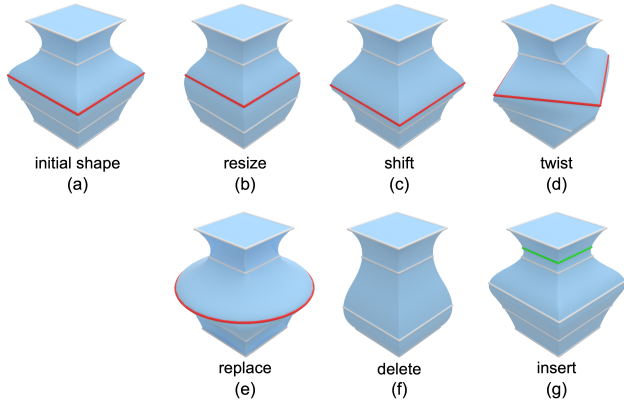


Figure 5: Given (a) a generalized cylinder, our approach allows editing its shape by (b) resizing, (c) shifting, (d) twisting, (e) replacing, and (f) deleting a selected key cross section (in red color). It also allows (g) inserting a new key cross section (in green color) without modifying the generalized cylinder's shape.

spine curve $\mathbf{A}(v)$, updating each key cross section based on the transformation of its local frame, and rerunning the lofting algorithm. Figure 4(a-c) shows a generalized cylinder before and after the bending.

2. *Replacing.* The replacing operation substitutes the original spine curve $\mathbf{A}(v)$ with a specified curve $\mathbf{A}'(v)$, after which the surface $\mathbf{S}(u, v)$ is updated accordingly; see Figure 4(a&d) for an example.

Editing via key cross-sectional curves. We provide six editing operations via key cross-sectional curves. After each editing operation, we update the generalized cylinder surface $\mathbf{S}(u, v)$ by rerunning the lofting algorithm.

1. *Resizing.* Resizing is achieved via selecting and uniformly scaling a key cross-sectional curve $\mathbf{C}_h(u)$, $h \in [1, K]$ in its local frame; see Figure 5(a&b).
2. *Shifting.* Shifting is achieved via translating a selected key cross-sectional curve $\mathbf{C}_h(u)$ along the spine curve from $\mathbf{A}(v_h)$ to $\mathbf{A}(v'_h)$ and reorienting the cross section using the local frame at $\mathbf{A}(v'_h)$; see Figure 5(a&c). To avoid changing the order of the key cross sections, we require $v'_h \in (v_{h-1}, v_{h+1})$.
3. *Twisting.* Twisting is achieved via rotating a selected key cross-sectional curve $\mathbf{C}_h(u)$ in its local frame. To make the deformation natural, we propagate the rotation linearly to all the other key cross sections except fixing $\mathbf{C}_1(u)$ and $\mathbf{C}_K(u)$; see Figure 5(a&d).
4. *Replacing.* Replacing is achieved via substituting a selected key cross-sectional curve $\mathbf{C}_k(u)$ with a new curve $\mathbf{C}'_k(u)$, which should have the same degree and knot vector as $\mathbf{C}_k(u)$; see Figure 5(a&e).
5. *Deleting.* Deleting is achieved via removing a selected key cross section $\mathbf{C}_k(u)$ from the set of input key cross sections; see Figure 5(a&f).
6. *Insertion.* Insertion is achieved via selecting a point on the spine curve say $\mathbf{A}(v')$ and creating a new key cross-sectional curve

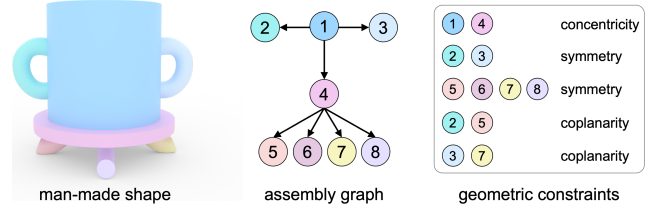


Figure 6: (Left) Modeling a man-made shape by assembling multiple generalized cylinder. The assembly procedure is represented using (Middle) an assembly graph, during which (Right) several geometric constraints are automatically identified.

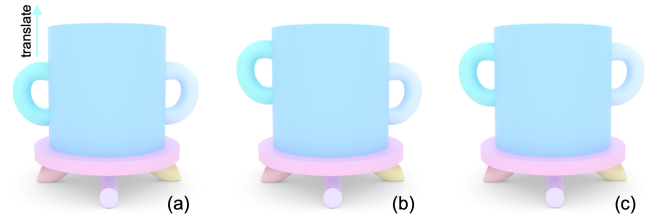


Figure 7: Structure-aware shape editing. The editing effects caused by a user's (a&b) translation of a handle is (c) propagated to a shape part that has a symmetry relation with the handle being edited.

by extracting an iso-curve at v' from the generalized cylinder surface $\mathbf{S}(u, v)$. Insertion does not modify the generalized cylinder's shape; see Figure 5(a&g). Yet, the first four operations can be subsequently applied on the inserted key cross section for the shape editing.

3.3. Assembling Multiple Generalized Cylinders

To model man-made objects with complex shape, we propose to iteratively assemble multiple generalized cylinders by a user. We represent the assembly procedure using a directed graph called the *assembly graph*, where a node represents a generalized cylinder and a directed edge $a \rightarrow b$ represents that a newly created generalized cylinder a is assembled to an existing generalized cylinder b . In particular, we call cylinder a a child of cylinder b since cylinder a is created after cylinder b and has to be attached with cylinder b . Figure 6 shows an example man-made shape, its assembly graph, and associated geometric constraints.

We provide three operations to assemble generalized cylinders.

1. *Creating.* One can create the geometry of a generalized cylinder from scratch. In case the cylinder to be created is not the first one, it has to be always connected with an existing generalized cylinder, ensuring fabricability of the resulting shape.
2. *Copying.* We observe many shape parts in man-made objects have exactly the same shape but different transformations such as handles of a pot and legs of a table. Hence, one can copy the shape of an existing generalized cylinder to create an instance.
3. *Transforming.* One can transform a created or copied generalized cylinder, including translation and rotation.

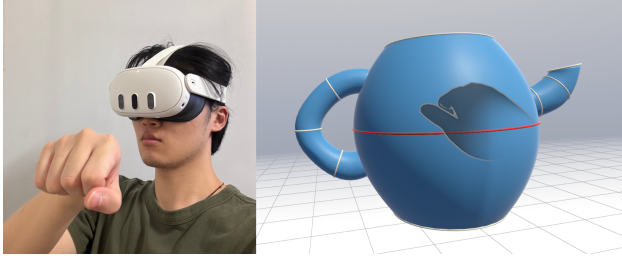


Figure 8: Left: a user is modeling a man-made shape with mid-air hand gestures using our VR tool. Right: our user interface for modeling a man-made shape, where the white/red curves are key cross sections.

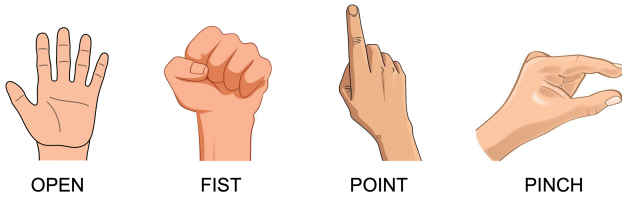


Figure 9: Four common hand gestures used in our user interface for modeling man-made shapes.

Identifying geometric constraints. We automatically identify geometric constraints among different generalized cylinders by analyzing the assembly procedure of these cylinders. These geometric constraints will be used for the structure-aware editing of the shape.

1. **Connectivity.** Two generalized cylinders are connected if there are neighbors in the assembly graph.
2. **Coplanarity.** Two generalized cylinders are coplanar if their spine curves are coplanar.
3. **Concentricity.** Two generalized cylinders are concentric if 1) they are neighbors in the assembly graph; and 2) their spine curves meet at one end.
4. **Symmetry.** Two generalized cylinders are symmetric if 1) they share the same parent in the assembly graph and; 2) their spine curves are symmetric with respect to the parent's spine curve.

Structure-aware shape editing. Structure-aware shape editing aims to preserve structural characteristics like connectivity and symmetry during the shape editing process [MWZ*13]. We perform structure-aware shape editing on an assembly of generalized cylinders following the approach in [ZFCO*11]. During the shape editing, we restrict that when transforming a shape part, it should always attach to its parent part (if any), aiming to satisfy the shape connectivity constraint; see Figure 6 and 7(a&b). We propagate the modification of a shape part's geometry and/or transformation to other parts by symmetry and proximity. In detail, for a shape part P , we first identify all the shape parts that are in the same symmetry group as P , and then apply the same transformation and shape editing operations to them as P ; see Figure 7(b&c). Moreover, we propagate the modification on a shape part P to all its descendant nodes in the assembly graph in a breadth-first manner, aiming to preserve the shape structure by satisfying the identified geometric constraints. Once satisfied with the modeled shape, we convert the parametric representation of each generalized cylinder into a wa-

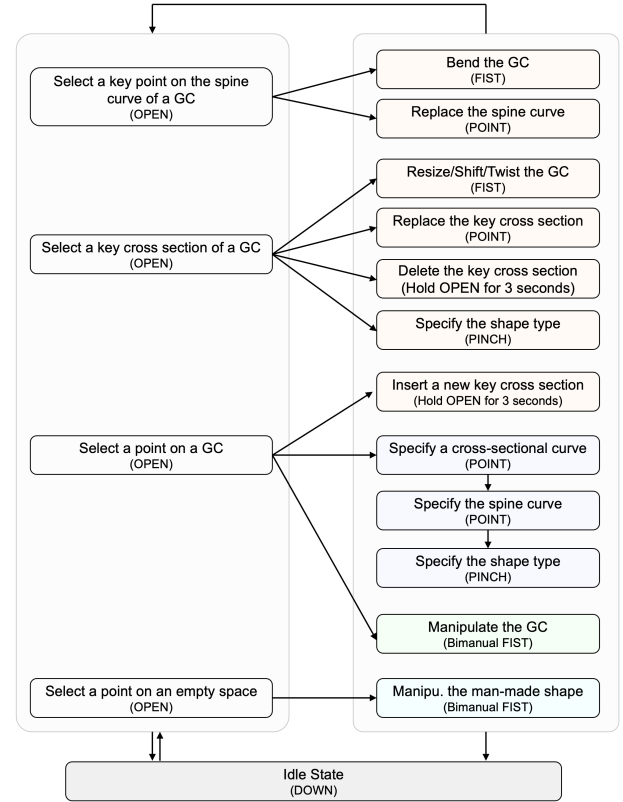


Figure 10: The state transition diagram and associated hand gestures for modeling a man-made shape. User interactions for editing, creating, and manipulating a generalized cylinder are colored in orange, blue, and green, respectively. User interaction for manipulating the man-made shape is colored in cyan.

tertight mesh. Next, we perform CSG boolean operations for these part meshes to obtain a single mesh representing the shape.

4. Gesture-based User Interface

Taking the shape modeling approach in Section 3 as a foundation, we propose a gesture-based user interface for modeling man-made shapes in VR. Our interaction hardware is a VR headset (i.e., Meta Quest 3), which tracks and recognizes the user's mid-air hand gestures using its built-in cameras for 3D user interaction; see Figure 8. We first describe the hand gestures used in our user interface, as well as principles to assign modeling operations to these gestures. Next, we explain our user interaction design that makes use of the gestures for creating, editing, and manipulating a generalized cylinder, as well as for manipulating the man-made shape. Figure 10 describes the state transitions and the expected hand gestures at each state when a user models a man-made shape. Figure 11 shows a running example of creating a teapot model using our interface. Please watch the accompanying video of a live demo of various user interactions in our VR tool.

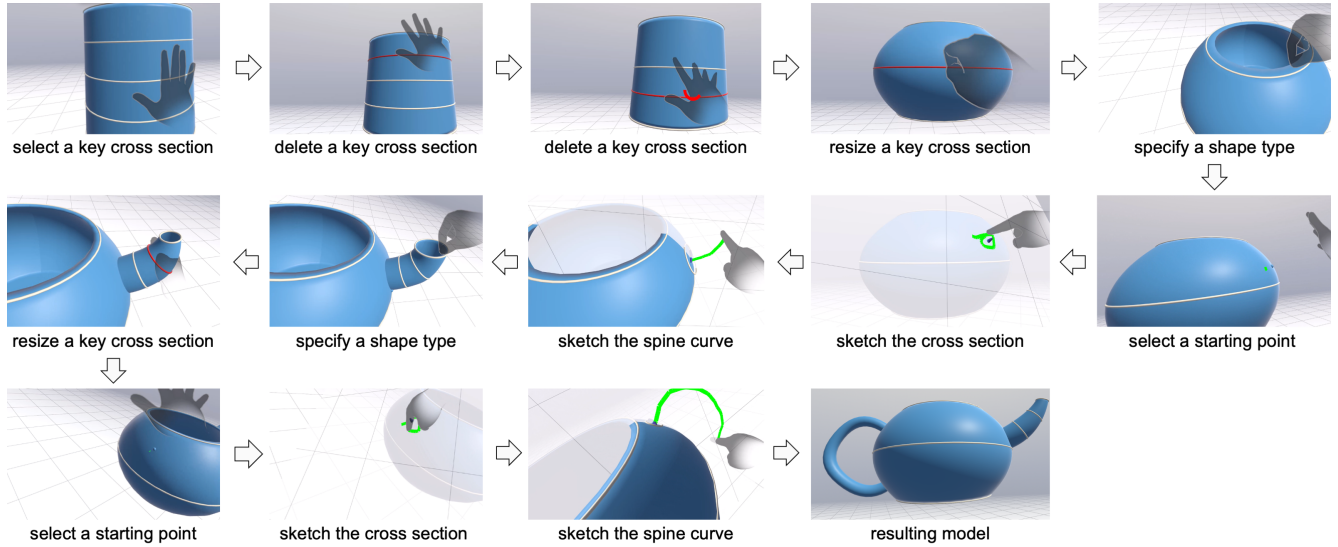


Figure 11: A running example of using our VR tool to interactively create a teapot model with three shape parts (i.e., body, spout, and handle), each of which is modeled as a generalized cylinder and assembled sequentially.

Hand gestures. We choose four hand gestures for the 3D user interaction according to established HCI principles (i.e., simple gestures, natural mapping, and consistency); see Figure 9.

- **OPEN.** An OPEN palm gesture is used for selecting a shape primitive with ray casting, including point, curve, and generalized cylinder. When the ray intersects with a shape primitive, the user performs a FIST/POINT/PINCH gesture to select the primitive.
- **FIST.** A FIST gesture is used for manipulating a selected shape primitive, mimicking the experience of manipulating objects in the real world.
- **POINT.** A POINT gesture is used for sketching a planar curve on a 3D plane to specify/replace the spine curve or a key cross-sectional curve.
- **PINCH.** A PINCH gesture is used for specifying the shape type and thickness parameter (if any) of a generalized cylinder.

All the four gestures are unimanual in our interface, except for the FIST gesture. In particular, we use a unimanual FIST gesture for manipulating a point, a curve, or a generalized cylinder and a bimanual FIST gesture for manipulating the man-made shape since the bimanual FIST gesture allows scaling the man-made shape. To distinguish between one and two-handed interactions, a DOWN gesture is assigned to a hand if the hand is located below the center of the user's body.

Creating and assembling a generalized cylinder. To create a generalized cylinder from scratch and assemble it, we propose the following procedure and interaction design; see Figure 12.

- **Select a starting point for the spine curve.** A starting point O_s is selected on an existing generalized cylinder using an OPEN \rightarrow POINT gesture; see Figure 12(b&c). After that, a 3D plane centered at the starting point O_s and tangent to the existing generalized cylinder appears.
- **Specify a cross-sectional curve.** Next, the user specifies a cross-sectional curve via sketching a closed curve with a POINT

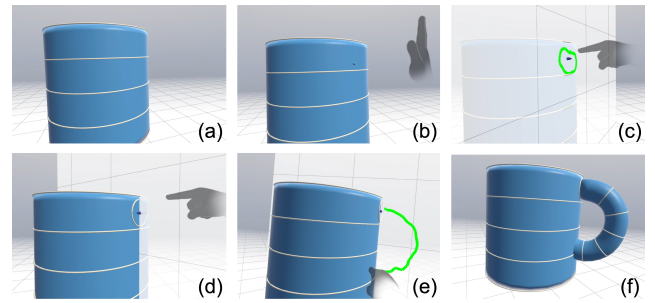


Figure 12: Creating and assembling a generalized cylinder as the handle of a mug. (a) The body of the mug has been modeled as a generalized cylinder. (b) Select a point on the cylinder using an OPEN gesture. (c) Sketch a cross-sectional curve using a POINT gesture. (d) The sketch is approximated with a smooth curve. (e) Sketch the spine curve using a POINT gesture. (f) The sketch is approximated with a smooth curve, and a new generalized cylinder is created accordingly.

gesture on the 3D plane; see Figure 12(c). The sketching is done when the user changes to an OPEN gesture. After that, the sketched curve is automatically simplified using the Ramer-Douglas-Peucker (RDP) algorithm [DP73] and matched with a set of predefined template curves, including circles and regular polygons. We choose the template curve with the highest similarity to the sketch as the cross-sectional curve; see Figure 12(d).

- **Specify the spine curve.** The user further specifies the spine curve via sketching an open curve with a POINT gesture on a 3D plane that passes through the starting point O_s . The 3D plane's normal is $\mathbf{n}_s \times \mathbf{t}_s$, where \mathbf{n}_s is the surface normal at the point O_s and \mathbf{t}_s is the tangent vector of the spine curve of the existing generalized cylinder corresponding to the point O_s ; see Figure 12(e). The sketching is done when the user changes to an OPEN gesture. After that, the sketched curve is automatically



Figure 13: Our VR tool allows modeling a variety of man-made shapes that can be used for different purposes in our daily life, each of which was modeled within a few minutes using our tool.

simplified using the RDP algorithm and then approximated using a NURBS curve. Note that the spine curve's endpoint may (e.g., teapot handle) or may not (e.g., teapot spout) be on the parent part (e.g., teapot body), which is identified by computing the distance between the endpoint and the parent part.

- **Construct a generalized cylinder.** A solid generalized cylinder is created by sweeping the cross-sectional curve along the spine curve, on which a few key cross-sectional curves are automatically created at sampled positions of the spine curve; see Figure 12(f). The created generalized cylinder is guaranteed to be connected with its parent part to ensure fabricability since its spine curve's starting point O_s is on the parent part's surface.

Editing a generalized cylinder. After creating a generalized cylinder, our interface allows editing the generalized cylinder using the operations described in Section 3.2. We introduce our user interaction design for executing these operations, some of which are visually demonstrated in Figure 11.

To edit a generalized cylinder via its spine curve, the user first selects a key point on the spine curve using an OPEN \rightarrow FIST/POINT gesture mentioned above.

1. **Bending.** To bend the generalized cylinder, the user moves the FIST gesture to translate the selected key point within the plane of the spine curve.
2. **Replacing.** To replace the spine curve, the user uses a POINT gesture to sketch a new curve within the plane of the original spine curve for replacing it.

To edit a generalized cylinder via key cross sections, a user first selects a point P on a key cross section using an OPEN \rightarrow FIST/POINT/PINCH gesture.

1. **Resizing.** To scale the key cross section, the user moves the

FIST gesture along the radial direction at the point P within the key cross section's plane.

2. **Shifting.** To shift the key cross section, the user moves the FIST gesture along the normal direction of the key cross section's plane.
3. **Twisting.** To rotate the key cross section, the user moves the FIST gesture along the tangent direction at the point P within the key cross section's plane.
4. **Replacing.** To replace the key cross section, the user uses a POINT gesture to sketch a new curve within the plane of the original key cross section for replacing it.
5. **Deleting.** To delete the key cross section, the user holds the OPEN gesture for 3 seconds after the ray intersects with the key cross section.
6. **Insertion.** Different from the above five operations, the user selects a point on the generalized cylinder but not on any existing key cross section using an OPEN gesture. To insert a new key cross section at the selected point, the user holds the OPEN gesture for 3 seconds.
7. **Specifying the shape type.** When a top/bottom key cross section is selected, the user uses a PINCH gesture to specify the shape type and shell thickness. In detail, an open (a close) PINCH gesture indicates that the generalized cylinder's top/bottom is a shell (a solid). Moreover, the distance between the two fingers in an open PINCH gesture specifies the shell thickness.

Manipulating a generalized cylinder. To manipulate a generalized cylinder, the user first selects it using an OPEN \rightarrow FIST gesture, and then manipulates it using a FIST gesture. Since we restrict that a generalized cylinder has to be always attached to its parent part during the manipulation in Section 3.3, we only allow translating the generalized cylinder and update its orientation automatically according to its position relative to the parent part. After the

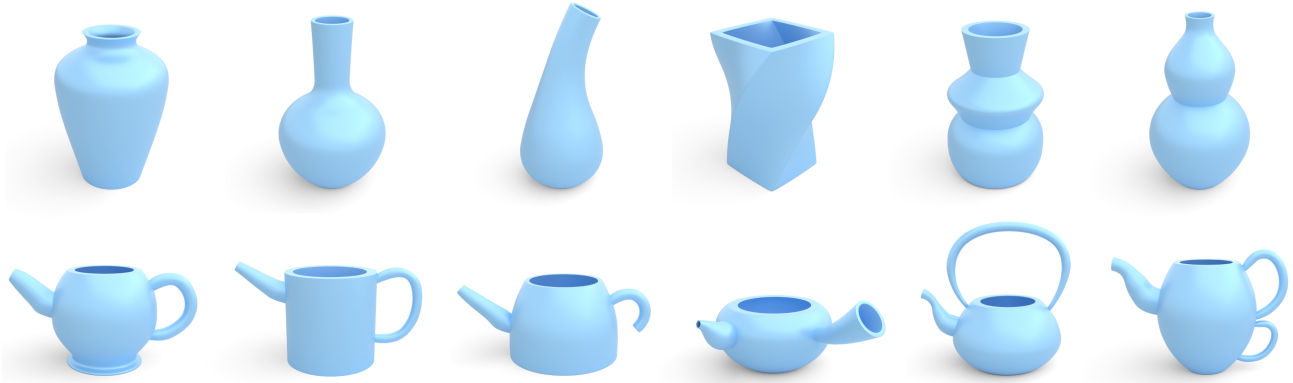


Figure 14: Our VR tool allows modeling man-made shapes with the same functionality but different forms: (top) vase and (bottom) teapot models of different forms.



Figure 15: A gallery of man-made shapes modeled by using our VR tool and fabricated with 3D printing. From left to right: vase, kettle, funnel, smoking pipe, and toy car. Note that the toy car is an assembly of three parts, each of which is modeled using our tool.

manipulation, the man-made shape will be updated in a structure-aware manner using the approach in Section 3.3.

When switching from an OPEN gesture to a FIST gesture, if we hold the FIST gesture static for more than 3 seconds, our interface will create an instance of the selected generalized cylinder, and then the manipulation operation will be performed on the instance instead. This strategy allows easy modeling of symmetric parts, which are common in man-made shapes.

Manipulating the man-made shape. To select the man-made shape, the user selects a point on an empty space using an OPEN → FIST gesture. After that, the selected man-made shape is manipulated with a bimanual FIST gesture via the handle bar metaphor [SGH*12], which offers seamless 7-DOF manipulation (3 translations, 3 rotations, and 1 scaling). This manipulation not only allows inspecting the man-made shape from a desired perspective but also finding a suitable view for creating/editing/manipulating a specific generalized cylinder.

5. Results

Implementation. We implemented our VR modeling tool using Unity. We used Meta XR SDKs to track and recognize the four hand gestures in Figure 9, which are captured by using the cameras built into the Meta Quest 3 headset. We used LibCSG-Runtime library [LR23] to perform CSG operations on the modeled generalized cylinders for creating and exporting a watertight mesh representing the man-made shape, which can be used for further rendering, fabrication, and processing.

Virtual results. Figure 13 shows that our VR tool allows modeling a wide variety of fabricable man-made shapes with different functionalities, including a table and a chair for supporting, a cup and a vase for containing, a teapot for purifying, a funnel for transferring, a hammer for striking, a toy car of playing, a clothes rack for hanging, and a hat for protection. Each of the man-made shapes with 1 or 2 generalized cylinders (e.g., vase, cup, hammer, and hat) only took less than 2 minutes to model by an expert user using our tool. Man-made shapes with a larger number of generalized cylinders took a longer time to model. For example, the chair with 12 generalized cylinders took 15 minutes to model by the expert user.

Figure 14 shows that our VR tool allows modeling man-made objects with the same functionality but different geometric forms. For example, some vase models are bended or twisted while others have cavities at the bodies since they are represented by multiple generalized cylinders; see Figure 14 (top). The teapot models also have a variety of forms; see Figure 14 (bottom). First, the teapot models' bodies, handles, and spouts have different shapes. Second, the handles are connected to the bodies in different ways. For example, the handle can connect to either the side or the top of the body, with one or two contact regions.

Fabrication. We 3D printed several man-made objects modeled using our tool, to validate their fabricability and usability; see Figure 15. For example, we show that we can pour water from the kettle to the vase through the funnel. Among all the prints, the toy car is special; see Figure 15 (right). This is because each of its component parts (i.e., body and two wheels) was individually modeled us-

ing our tool. Then, these three parts were 3D printed and physically assembled to form a toy car that can roll on the ground, demonstrating that our tool can be used to model articulated objects that allow relative motions among their component parts. Please watch the accompanying video for a live demo of using these prints.

6. User Study

We conducted a within-subjects user study to evaluate the effectiveness of our VR tool and how well it supports creativity. 8 participants (7 males and 1 female) aged between 22 to 30 years (mean 25.8 years) were enrolled in the study. None of the participants have performed 3D shape modeling in VR before but two have played VR headset-based games.

The study comprises two distinct sessions. The first session involves users recreating two target man-made shapes to assess the effectiveness of our tool. In the second session, users create a specified man-made shape, allowing us to evaluate the level of creativity supported by our tool. Participants used our tool and Gravity Sketch [Gra25] for both sessions of the study in a counterbalanced design to mitigate learning effects. We chose Gravity Sketch as a baseline since it is a widely used VR tool for modeling man-made shapes. Before the start of the study, participants were taught how to use both tools by a domain expert. Training lasted about 10 minutes for our tool and 15 minutes for Gravity Sketch. Note that the participants were only taught the basic features of Gravity Sketch needed for the tasks, including modeling a surface, assigning thickness, and assembling parts.

Task #1: Recreation of target shapes. The task of the first session is to recreate two target man-made objects (i.e., a mug and a cauldron) using both tools; see the inset. Before the start of the session, we demonstrated the two target shapes to the participants, and explained that the task is to recreate a mug and a cauldron that are similar to the two targets, respectively, by using both tools. At the end of the session, we recorded the time taken to model each shape by the participants, and saved the modeled shape as a mesh file. The participants were asked to complete the System Usability Scale (SUS) questionnaire [Lew18].

Figure 16 shows shapes modeled by four of eight participants using both tools, along with the time taken for the modeling. Participants were able to complete the task with both tools using a similar amount of time. In detail, the average time to model the mug is 9.2 minutes for both tools; the average time to model the cauldron is 13.5 minutes and 11.5 minutes for the baseline and our tool, respectively. The key difference lays in the quality of the modeled shapes. We found that all the shapes modeled by the baseline tool are not fabricable due to disconnected parts and/or parts with zero thickness. In contrast, the shapes modeled using our tool are all fabricable, thanks to our structure-aware modeling approach based on assembling generalized cylinders. Besides the fabricability, we also found that shapes modeled by our tool are more aesthetically pleasing, since they satisfy geometric constraints (e.g., symmetry and coplanarity) that are common in man-made objects.

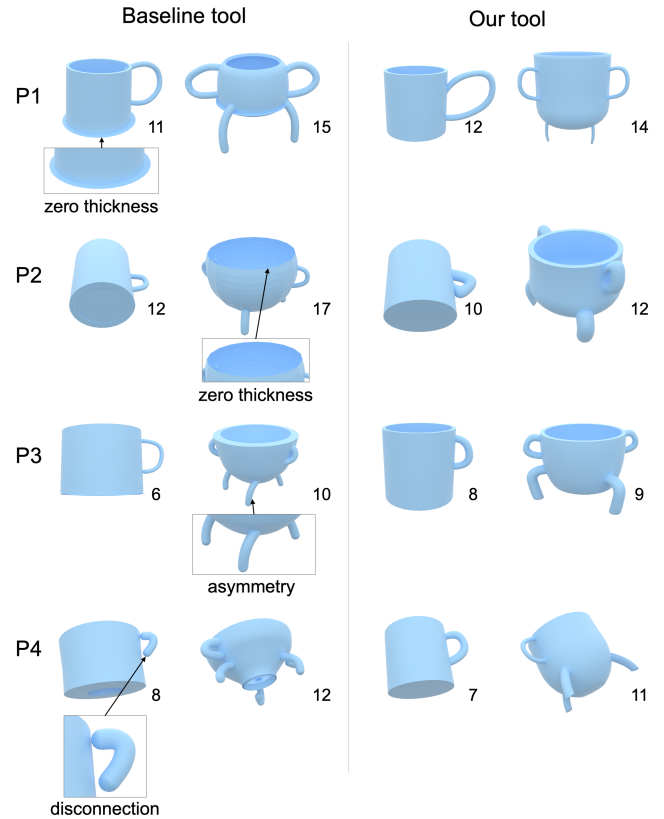


Figure 16: We present mug and cauldron models created by four participants in the first session of the user study using both tools, where the number beside each model is the time taken to create it in minutes. We found that all the models created by the baseline tool cannot be fabricated, due to zero-thickness parts and/or disconnected parts. Some models created by the baseline tool are not aesthetically pleasing, e.g., due to asymmetry of the parts.

Figure 17 shows 7-point Likert scales of the SUS questionnaire rated by the participants for both tools. We can see that the participants generally prefer using our tool for modeling man-made shapes. In particular, the average rating of our tool (6.4) is higher than that of the baseline (5.6) for the 7th question “I would imagine that most people would learn to use this system very quickly.” The ease of learning of our tool is likely because of our natural user interaction design simply based on four common hand gestures. In addition, the average rating of our tool (6.1) is higher than that of the baseline (3.7) for the 3rd question “I thought the system was easy to use.” One possible reason for the ease of use of our tool is that our structure-aware modeling approach automatically satisfies some geometric constraints of the shape parts such as connectivity and symmetry during the interactive modeling process, reducing the workload of the user.

Task #2: Creation of new shapes. In this session, the participants were asked to create a table model using both tools. We briefed the participants to create a table model that they like in 30 minutes, and that they can create as many models as they want. After this, participants evaluated the level of creativity supported by each tool via the

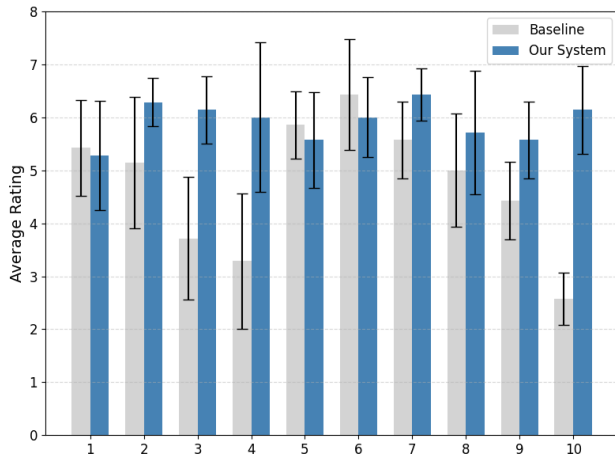


Figure 17: User ratings on a Likert scale from 1 (strongly disagree) to 7 (strongly agree) for the SUS questionnaire done after the first session of the user study. The questionnaire consists of 10 questions; please refer to [Lew18] for details. For the negative questions in the questionnaire (i.e., questions 2, 4, 6, 8, and 10), the reported scale is $8 - s$, where s is the scale rated by each participant for the original question.



Figure 18: Table models created by the participants using our tool in the second session of the user study.

Creativity Support Index (CSI) [CL14]. We skipped the collaboration part of the questionnaire, as we did not build any collaborative functionality in our VR tool.

Figure 18 shows the table models created by some participants using our tool. These table models have a variety of geometric forms and some of them have a creative shape such as the table with three integrated seats; see the bottom right model in Figure 18. Table 1 shows a detailed summary of the CSI scores. Overall, participants rated our tool (CSI = 13.87) higher than Gravity Sketch (CSI = 12.3). This result shows that the participants found our VR tool is enjoyable and supports exploration, enabling to fully express themselves in the creative modeling of man-made shapes.

7. Conclusion

We proposed a new VR tool for modeling man-made shapes using natural hand gestures in virtual reality. Our method consists of two key components: a geometric modeling approach based on assembling generalized cylinders in a structure-aware manner, and a nat-

Table 1: The CSI scores (out of 20, higher is better) for the baseline tool and our tool (mean and standard deviation).

	Baseline Tool	Our Tool
Enjoyment	11.83 (2.11)	14.50 (2.50)
Exploration	12.33 (2.56)	14.17 (1.95)
Expressiveness	12.50 (2.36)	14.33 (2.49)
Immersion	14.17 (2.79)	12.67 (3.25)
Results Worth Effort	10.67 (2.21)	13.67 (2.43)
CSI	12.30	13.87

ural user interface that intuitively maps four common hand gestures to all the necessary shape modeling operations. Various man-made shapes created using our tool are shown in the paper, some of which are 3D printed to demonstrate their fabricability and usability. The user study validates the effectiveness of and creativity supported by our tool for modeling fabricable and usable man-made shapes.

Limitations and future work. First, our current gesture-based user interface does not allow precise specification of parameters for modeling man-made shapes, which can be resolved by augmenting the interface with a virtual keyboard for parameter input. Second, our template cross-sectional curves are limited to circles and regular polygons. In the future, we plan to expand the templates by including non-regular polygons such as rectangles, aiming to extend the diversity of shapes that can be modeled by a single generalized cylinder. Third, there exist man-made shapes that cannot be represented as an assembly of generalized cylinders. To address this limitation, one possible approach is to generalize our geometric representation, e.g., to swept surfaces or even tensor product surfaces. Lastly, we plan to enhance our interactive modeling tool by incorporating advanced fabrication requirements, such as minimizing support material, ensuring structural soundness, and minimizing fabrication cost/time.

8. Acknowledgement

We thank the reviewers for their valuable comments and the users for their participation in our user study. This work was supported by the Singapore MOE AcRF Tier 2 Grants (MOE-T2EP20222-0008, MOE-T2EP20123-0016).

References

- [ACJ12] ARAÚJO B. R. D., CASIEZ G., JORGE J. A.: Mockup Builder: Direct 3D modeling on and above the surface in a continuous interaction space. In *Proc. of Graphics Interface* (2012), pp. 173–180. 2
- [Ado25] ADOBEMEDIUM: Adobe Medium, 2025. <https://www.adobe.com/products/medium.html>. 2
- [Cha24] CHATTERJEE S.: Free-form shape modeling in XR: A systematic review, 2024. arXiv:2401.00924. 2
- [CL14] CHERRY E., LATULIPE C.: Quantifying the creativity support of digital tools through the creativity support index. *ACM Transactions on Computer-Human Interaction* 21, 4 (2014), 21:1–21:25. 10

- [CRV20] COHEN M. W., REGAZZONI D., VRUBEL C.: A 3D virtual sketching system using NURBS surfaces and leap motion controller. *Computer-Aided Design and Applications* 17, 1 (2020), 167–177. 2
- [CXX*24] CHEN S., XU R., XU J., XIN S., TU C., YANG C., LU L.: QuickCSGModeling: Quick CSG operations based on fusing signed distance fields for VR modeling. *ACM Trans. on Multimedia Computing, Communications and Applications* 20, 7 (2024), 189:1–189:18. 2
- [CZS*13] CHEN T., ZHU Z., SHAMIR A., HU S.-M., COHEN-OR D.: 3-Sweep: Extracting editable objects from a single photo. *ACM Trans. on Graph. (SIGGRAPH Asia)* 32, 6 (2013), 195:1–195:10. 2
- [DGK*20] DREY T., GUGENHEIMER J., KARLBAUER J., MILO M., RUKZIO E.: VRSketchIn: Exploring the design space of pen and tablet interaction for 3D sketching in virtual reality. In *Proc. of ACM CHI* (2020), pp. 1–14. 2
- [DP73] DOUGLAS D. H., PEUCKER T. K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* 10, 2 (1973), 112–122. 6
- [FCSF16] FU Q., CHEN X., SU X., FU H.: Natural lines inspired 3D shape re-design. *Graphical Models* 85 (2016), 1–10. 2
- [GMP*12] GOYAL M., MURUGAPPAN S., PIYA C., BENJAMIN W., FANG Y., LIU M., RAMANI K.: Towards locally and globally shape-aware reverse 3D modeling. *Computer-Aided Design* 44, 6 (2012), 537–553. 2
- [Goo24] GOOGLEBLOCKS: Google Blocks, 2024. <https://github.com/googlevr/blocks>. 1, 2
- [Gra25] GRAVITYSKETCH: Gravity Sketch, 2025. <https://www.gravitysketch.com/>. 1, 2, 9
- [HYC*05] HYUN D.-E., YOON S.-H., CHANG J.-W., SEONG J.-K., KIM M.-S., JÜTTLER B.: Sweep-based human deformation. *The Visual Computer* 21 (2005), 542–550. 2
- [JZF*21] JIANG Y., ZHANG C., FU H., CANNAVÒ A., LAMBERTI F., LAU H. Y. K., WANG W.: HandPainter - 3D sketching in VR with hand-based physical proxy. In *Proc. of ACM CHI* (2021), pp. 412:1–412:13. 2
- [KAHF05] KIM H., ALBUQUERQUE G., HAVEMANN S., FELLNER D. W.: Tangible 3D: Hand gesture interaction for immersive 3D modeling. In *Proc. of the 11th Eurographics Workshop on Virtual Environments* (2005), pp. 191–199. 2
- [Kod25] KODON: Tenklabs kodon, 2025. <https://www.kodon.xyz/>. 2
- [Lew18] LEWIS J. R.: The system usability scale: Past, present, and future. *International Journal of Human-Computer Interaction* 34, 7 (2018), 577–590. 9, 10
- [LLB24] LEE S.-H., LEE J. H., BAE S.-H.: Bimanual interactions for surfacing curve networks in VR. In *Extended Abstracts of ACM CHI* (2024), pp. 60:1–60:7. 1, 2
- [LLZM10] LI G., LIU L., ZHENG H., MITRA N. J.: Analysis, reconstruction and manipulation using arterial snakes. *ACM Trans. on Graph. (SIGGRAPH Asia)* 29, 6 (2010), 152:1–152:10. 2
- [LM04] LIVERANI A., MORIGI S.: Efficient 6DOF tools for free-form surface modelling. *The Visual Computer* 20 (2004), 554–564. 2
- [LR23] LIBCSG-RUNTIME: LibCSG-Runtime, 2023. <https://github.com/LokiResearch/LibCSG-Runtime>. 8
- [LWS*24] LU Y., WANG Y., SONG P., WONG H. S., MOK Y., LIU L.: Computational design of custom-fit PAP masks. *Comp. & Graph.* 122 (2024), 103998:1–103998:11. 2
- [MGE08] MASSARWI F., GOTSMAN C., ELBER G.: Paper-craft from 3D polygonal models using generalized cylinders. *Comp. Aided Geom. Des.* 25, 8 (2008), 576–591. 2
- [MGS17] MCGRAW T., GARCIA E., SUMNER D.: Interactive swept surface modeling in virtual reality with motion-tracked controllers. In *Proc. of the Symposium on Sketch-Based Interfaces and Modeling* (2017), pp. 4:1–4:9. 2
- [MPS*04] MORTARA M., PATANÉ G., SPAGNUOLO M., FALCIDIENO B., ROSSIGNAC J.: Plumber: A method for a multi-scale decomposition of 3D shapes into tubular primitives and bodies. In *Proc. of ACM Symposium on Solid Modeling and Applications* (2004), pp. 339–344. 2
- [MWZ*13] MITRA N. J., WAND M., ZHANG H., COHEN-OR D., BOKELOH M.: Structure-aware shape processing. In *Eurographics State-of-the-Art Reports* (2013), pp. 175–197. 2, 5
- [NUK98] NISHINO H., UTSUMIYA K., KORIDA K.: 3D object modeling using spatial and pictographic gestures. In *Proc. of ACM Symposium on Virtual Reality Software and Technology* (1998), pp. 51–58. 2
- [PT97] PIEGL L., TILLER W.: *The NURBS Book (2nd edition)*. Springer-Verlag Berlin Heidelberg, 1997. 3
- [RRS19] ROSALES E., RODRIGUEZ J., SHEFFER A.: SurfaceBrush: From virtual reality drawings to manifold surfaces. *ACM Trans. on Graph. (SIGGRAPH)* 38, 4 (2019), 96:1–96:15. 1, 2
- [Scu25] SCULPTRVR: Sculptvr, 2025. <https://www.sculptrvr.com/>. 2
- [SGH*12] SONG P., GOH W. B., HUTAMA W., FU C.-W., LIU X.: A handle bar metaphor for virtual object manipulation with mid-air interaction. In *Proc. of ACM CHI* (2012), pp. 1297–1306. 8
- [Sha25] SHAPELAB: Shapelab, 2025. <https://shapelabvr.com/>. 2
- [SPS01] SCHKOLNE S., PRUETT M., SCHRÖDER P.: Surface Drawing: Creating organic 3D shapes with the hand and tangible tools. In *Proc. of ACM CHI* (2001), pp. 261–268. 2
- [Til21] TILTBRUSH: Google Tilt Brush, 2021. <https://github.com/googlevr/tilt-brush>. 1, 2
- [VMLR13] VINAYAK, MURUGAPPAN S., LIU H., RAMANI K.: Shape-It-Up: Hand gesture based creative expression of 3D shapes using intelligent generalized cylinders. *Computer-Aided Design* 45, 2 (2013), 277–287. 2, 3
- [WJZL08] WANG W., JÜTTLER B., ZHENG D., LIU Y.: Computation of rotation minimizing frames. *ACM Trans. on Graph.* 27, 1 (2008), 1–18. 3
- [YAS*21] YU E., ARORA R., STANKO T., BÆRENTZEN J. A., SINGH K., BOUSSEAU A.: CASSIE: Curve and surface sketching in immersive environments. In *Proc. of ACM CHI* (2021), pp. 190:1–190:14. 2
- [YHZ*14] YIN K., HUANG H., ZHANG H., GONG M., COHEN-OR D., CHEN B.: Morfit: Interactive surface reconstruction from incomplete point clouds with curve-driven topology and geometry control. *ACM Trans. on Graph. (SIGGRAPH Asia)* 33, 6 (2014), 202:1–202:12. 2
- [YK06] YOON S.-H., KIM M.-S.: Sweep-based freeform deformations. *Comp. Graph. Forum (Eurographics)* 25, 3 (2006), 487–496. 2
- [ZCHH24] ZHU X., CHEN Z., HU R., HAN X.: Controllable shape modeling with neural generalized cylinder. In *Proc. of SIGGRAPH Asia* (2024), pp. 80:1–80:11. 2
- [ZFCO*11] ZHENG Y., FU H., COHEN-OR D., AU O. K.-C., TAI C.-L.: Component-wise controllers for structure-preserving shape manipulation. *Comp. Graph. Forum (Eurographics)* 30, 2 (2011), 563–572. 2, 5
- [ZSY*17] ZHU X., SONG L., YOU L., ZHU M., WANG X., JIN X.: Brush2Model: Convolution surface-based brushes for 3D modelling in head-mounted display-based virtual environments. *Comp. Anim. Virtual Worlds* 28, 3–4 (2017), e1764:1–e1764:10. 2
- [ZTZ*22] ZHU Y., TANG X., ZHANG J., PAN Y., SHEN J., JIN X.: 3DBrushVR: From virtual reality primitives to complex manifold objects. In *Proc. of IEEE International Symposium on Mixed and Augmented Reality Adjunct* (2022), pp. 423–428. 2
- [ZYH*15] ZHOU Y., YIN K., HUANG H., ZHANG H., GONG M., COHEN-OR D.: Generalized cylinder decomposition. *ACM Trans. on Graph. (SIGGRAPH Asia)* 34, 6 (2015), 171:1–171:14. 2