Tabby: Explorable Design for 3D Printing Textures

R. Suzuki[1], K. Yatani[2], M. D. Gross[1], and T. Yeh[2]

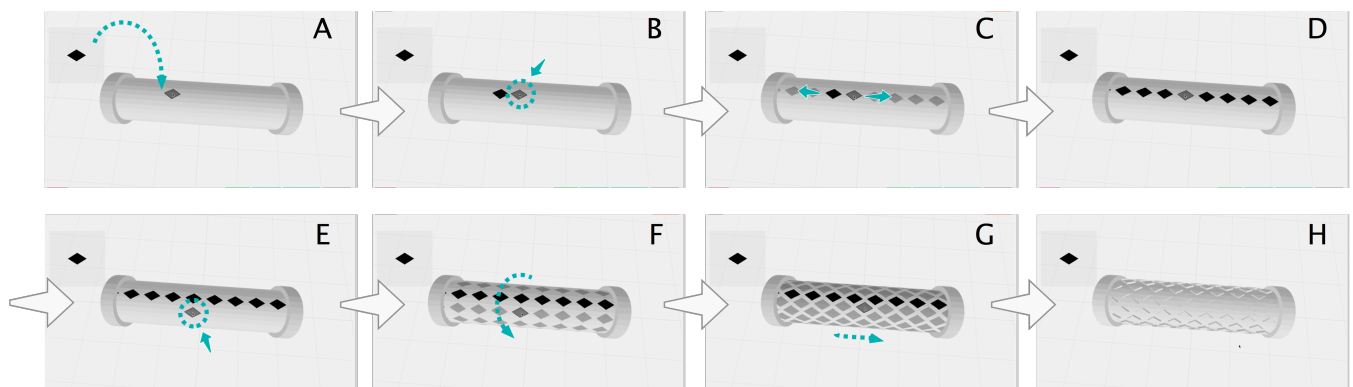[1]University of Colorado Boulder
[2]The University of Tokyo

Figure 1: *An overview of the workflow for creating gripping surface texture with Tabby. (A) Users sketch a texture element for grips and drag-and-drop it onto the 3D model. The system infers a possible target surface based on the user interaction. (B) Users copy and paste the texture element a couple of times. (C) When the system detects repetitive operations, Tabby suggests texture patterns. (D) Users can accept the suggested pattern. (E – G) Another example of texture creation with Tabby. (H) Users extrude the texture, creating a 3D-printable pattern.*

Abstract

This paper presents Tabby, an interactive and explorable design tool for 3D printing textures. Tabby allows texture design with direct manipulation in the following workflow: 1) select a target surface, 2) sketch and manipulate a texture with 2D drawings, and then 3) generate 3D printing textures onto an arbitrary curved surface. To enable efficient texture creation, Tabby leverages an auto-completion approach which automates the tedious, repetitive process of applying texture, while allowing flexible customization. Our user evaluation study with seven participants confirms that Tabby can effectively support the design exploration of different patterns for both novice and experienced users.

CCS Concepts

•*Human-centered computing* → *User centered design;* •*Computing methodologies* → *Mesh geometry models;*

1. Introduction

Texture is an essential property of physical objects that affects aesthetics, usability, and functionality. For example, patterns in a lampshade can enhance the aesthetics of the light design, brick patterns in architectural models can improve visual details, and a rough texture can affect an object's usability by providing a gripping surface. Designing and applying textures, however, remains difficult and time-consuming. It requires proficient 3D modeling skills and manual repetitive operations. Existing CAD tools usually support texture design through parametric modeling which allows a designer to explore different patterns through changing parameters or to synthesize textures from pre-defined examples. Such parametric design is powerful, but it suffers from two key limitations. First, the design process involves indirect manipulation of repetitive parameter tuning, which creates a large gulf of execution. Second, there is a steep learning curve to master commands. Prior work reveals that typical CAD users do not utilize most of the available commands in such tools due to high learning cost [MLGF09]. Thus, our formative study reveals many designers, even experienced users, often simply repeat copy-and-paste to create desired patterns, and it severely limits flexibility of design exploration. This observation leads us to explore an alternative interaction model where users can naturally adopt high-level commands and explore alternative designs through direct manipulation.

To investigate this new approach, we developed Tabby, an interactive design tool for creating and exploring 3D printing textures. With Tabby, a designer would only need to demonstrate the first few units of a texture pattern. Then, the system can automatically

infer a complete pattern the designer may have in mind. The designer can simply accept it or adjust to achieve the desired texture (Figure 1). Informed by our formative study, we introduce three techniques for efficient texture creation: 1) *auto-completion*: given the user's copy and paste operation, our system automates the tedious, repetitive process of applying texture, 2) *2D element manipulation*: in contrast to existing 3D modeling tools, our system supports 2D drawing operations to allow the user to define and arrange a texture element which is later automatically converted to a 3D printable texture , and 3) *semantic region selection*: our system infers which surface region to fill the repeated pattern.

To evaluate the efficiency and flexibility of texture design with Tabby, we conducted a controlled experiment with seven designers. Our study shows that Tabby speeds up texture creation by 80% over conventional tools. This performance gain becomes even larger with more complex target surfaces. Our qualitative result confirms that designing and applying textures with our system is simpler and more effective.

In summary, this paper contributes:

1. Tabby, an interactive system that instantiates the auto-completion method in the context of 3D texture design;
2. A set of workflow and techniques that support 2D operations for designing 3D-printable texture patterns;
3. A controlled experiment with seven designers that shows users can efficiently and flexibly design textures compared to conventional tools.
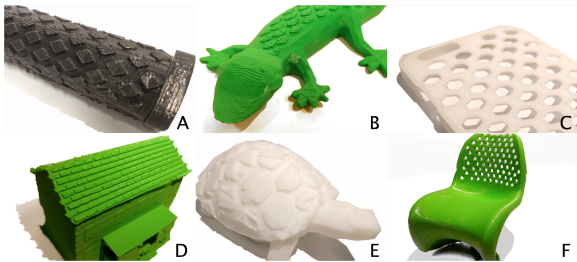

Figure 2: *Examples of 3D printed texture with Tabby.*

2. Related Work

Prior research has developed interactive systems for various mesh editing tasks. Systems like MeshMixer [SS10] and GeoBrush [TSS*11] provide an interactive texture cloning tool, and users can copy a geometric feature from example models by a simple brushing operation. However, these systems are not designed to support the creation of repeated patterns like tactile textures. Exemplars of 3D texture are typically difficult to obtain, inducing another limitation in existing texture cloning tools. Also, a user must manually choose a region of both the source texture and the target surface. Such a task can be time-consuming if the target region is large.

Other approaches have demonstrated texture creation through an automatic geometric synthesis method [DLL*15, ZHW*06]. However, these systems mainly aim to fully automate texture synthesis, and do not support interactive design explorations. One of our main objectives is to support interactive tactile texture creation for 3D models. Similar to our work, prior systems have introduced

an interactive tool for texture pattern synthesis [LS10, TCKP15]. While users can adjust the position and orientation of the pattern, the design exploration process in these tools is mostly limited in parameter tuning. In contrast, Tabby enables users to naturally and intuitively explore different texture patterns through direct manipulation.

Creating repetitive patterns is a tedious manual process. To alleviate user workload, prior work has developed auto-completion techniques in 2D drawing applications. Kazi et al. [KIZD12] demonstrated Vignette, an interactive drawing application that can facilitate user-defined 2D textures. The user draws a part of a texture and gestures to automatically fill a 2D region with the texture. Later research adapted this concept to enable data-driven decorative patterns [LBW*14], hand-drawn patterns [XCW14, XWSY15], and 3D sculpting [PXW18]. These systems inspired us to investigate how we can integrate these auto-complete techniques into 3D modeling to support interactive texture design.

3. Design Goals

To discover the needs and challenges, we conducted two formative studies where we observed the texture design process of 25 novice users and 3 professional CAD users. These observations led us to the following high-level design goals.
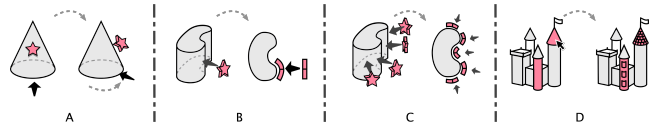

Figure 3: *Four difficulties in texture design revealed by the formative study.*

1. **Exploiting 2D Operations:** We found that novice users struggle to align, locate, and manipulate objects in 3D space. They needed to continuously change the camera angle to ensure that all texture elements were attached correctly to the target surface (Figure 3A). This task becomes even harder on curved surfaces because the user must deform a texture to fit the target curvature (Figure 3B). In contrast, 3D modeling becomes manageable when it is presented as a series of 2D operations [HAC16]. Thus, we decided to exploit 2D operations in our system.
2. **Liberating from Repetitive Operations:** Textures often consist of repeated patterns. We observed that users typically perform copy-and-paste operations to create such repetitive patterns. Thus, a repetition of such operations can signal an attempt to create textures, particularly for novice users. While it is a natural workflow, this simple copy-and-paste approach can be tedious since it requires many repetitive operations (Figure 3C). To simplify this manual operation, we should exploit this behavior to infer the user's intentions and, if possible, automatically complete the intended texture.
3. **Allowing Intuitive Exploration:** Texture design process is not one time. Users usually need to explore different patterns by changing the design properties (e.g., shape, size, rotation, orientation, alignment and region to fill). While users often change these properties through parameter tuning, we found that it becomes difficult when designers have to deal with multiple parameters and constraints simultaneously. Thus, our tool should

support designers to easily map their actions to intended results through direct manipulation (Figure 3D).

4. Tabby: An Interactive and Explorable Texture Design Tool

Our system, Tabby, supports the design process of rich, user-defined tactile texture creation. The typical workflow in Tabby is as follows:

Step 1: Import a 3D object into Tabby's working space.

Step 2: Sketch or import an SVG image for a desired texture element, and place it on the object (Figure 1A).

Step 3: The system infers the target region based on the placement of the texture element. Users can accept the suggested pattern (Figure 1D) or adjust the region selection if needed.

Step 4: After users copy and paste the texture elements a couple of times, the system suggests auto-completion of the pattern (Figure 1B, C, E and F).

Step 5: Users adjust the properties of the pattern (Figure 1G).

Step 6: Once users confirm, the system extrudes the 2D texture, converting it into a 3D geometry while maintaining water-tightness (Figure 1H).

Step 7: Users can download the modified model for 3D printing.

4.1. Semantic Region Selection

Users start with defining an element for texture patterns. In Tabby, they draw the element in a 2D sketching canvas or import the element as an SVG file. After deciding the element, users drag it into the main working space. The system displays a shadow of the element as visual feedback. As users move the element, the system automatically infers the surface region where they intend to create textures, and highlights it in light blue (Figure 4).
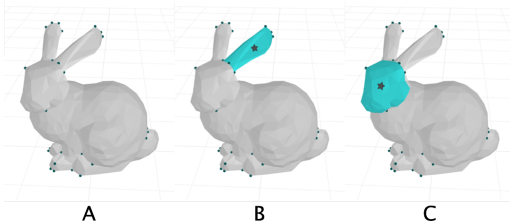
Figure 4: *Semantic region selection based on the cursor position.*

To enable the interactive region selection, we leverage and extend the existing mesh segmentation algorithms. Our method is based on cross-boundary mesh decomposition [ZT10], which computes a harmonic field with a Laplacian matrix to obtain segmentation boundaries by cutting along an isoline of the harmonic field. To obtain boundary positions, we calculate highly distorted vertices as boundary points because the distortion occurs in high surface curvature. The distortion of a vertex i is defined as: $D(i) = \max_{0 \leq r \leq R} \frac{2\pi - \sum_j \tau_j(r)}{2\pi}$, where R is a region radius and $\tau_j(r)$ are the angles at i of face j inside of region radius r. After calculating the distortion of each vertex, we extract high distorted points using a terminal vertex selection algorithm [SH02] with weights defined by the distance between the current mouse position and the target vertices.

4.2. Texture Auto-completion

After users place the first element, they can perform copy-and-paste operations to start forming a texture by auto-completing repetitive patterns. Tabby's auto-completion process is as follows. First the system tries to detect the user's copy-and-paste operations. When such operations are detected, it tracks the placements of the first two texture units and calculates the relative positions between the two. Then, Tabby makes a suggestion for auto-completion by presenting an example where each individual element is visualized as a shadow cast on the surface (Figure 1 and 5). Tabby's extrapolation can support both patterns in x and y, and curves lines with the user's additional demonstration.
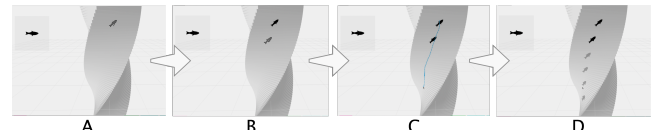
Figure 5: *Auto-completion of repeated patterns by drawing a line.*

After completing the design, the system automatically converts the 2D drawing element into a series of triangle meshes to create 3D textures. Users can also interactively change texture types (e.g., bumps or cutting holes). As we will show in the following section, the system also ensures that these added triangle meshes are properly fused into the target to obtain a water-tight result. We first triangulate the texture unit, and obtain the boundary points in 2D coordinates and corresponding triangle faces (Figure 6A-B). Then, we replace the original mesh surface with triangulated new meshes (Figure 6 C).
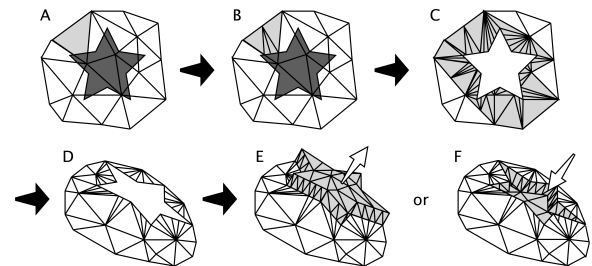
Figure 6: *Texture mesh generation. Triangulate the texture and replace the surfaces with the new mesh (A-D). The user can perform extrusion or intrusion of the texture element based on the obtained vertex normals (E-F).*

Once the boundary positions and surrounding surfaces are determined, the system creates a corresponding enclosure to maintain the water-tightness. We compute the vertex normal vector for each boundary point, and get a set of outer vertices by extending with the normal vector. We then perform the same constrained Delaunay triangulation on the outer vertices to obtain outer polygonal surfaces (Figure 6D). Note that the resulting textures are smoothly deformed to fit the target curvature. With the same interactive operation, users can also obtain an embossed texture by intruding the boundary points or a hollow texture by creating internal cavities inside the mesh with a consistent wall thickness (Figure 6E-F).

Figure 7 shows example results of texture patterns using Tabby:

A) an architectural model, B) a grip on a bike handle, C) a lampshades, D) smartphone cases, E) blocks, F) scale of geckos, G) chairs, H) tactile picture books, I) shell of turtles, and J) anti-slip finishing for cup holders.
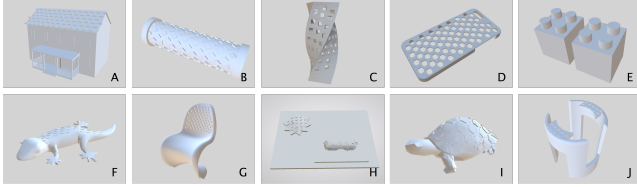

Figure 7: *Examples of possible real-world use scenarios.*

5. User Evaluation

To evaluate the usability of Tabby, we conducted a controlled experiment with 7 participants (5: male, 2: female) with some experience in existing CAD tools (Min: 10 months, Max: more than 6 years, Average: 3.2 years). They self-reported the expertise of 3D modeling tools as 5.3 out of 7 in average.

Tasks: We asked participants to create and modify textures on six different geometries: 1) a flat rectangle; 2) the side of a cylinder; 3) a sphere; 4) the side of a cone; 5) the head of a knight; and 6) the body of the Stanford bunny. In the creation tasks, the participants were asked to place a small cylinder texture element in a 3x3 grid. We measured the time by comparing Tabby with existing CAD tools (we allowed them to choose a tool they felt most comfortable.)

Evaluation Results With Tabby, participants completed their tasks with 29 seconds on average (Flat: 24 sec, Cylinder: 26 sec, Sphere: 41 sec, Cone: 24 sec). In contrast, in the reference condition, they needed 248 seconds on average (Flat: 117 sec, Cylinder: 270 sec, Sphere: 275 sec, Cone: 332 sec). Two participants gave up completing the tasks for sphere and cone surfaces. Our repeated-measure ANOVA test found significant results for both tools ($F_{1,6} = 48.6$, $p < .01$) and surfaces ($F_{3,18} = 6.19$, $p < .01$) as well as their interaction ($F_{3,18} = 6.03$, $p < .01$), confirming performance advantages of Tabby. Participants rated our tool as useful (mean: 6.3 out of 7), easy to use (5.7), and effective to perform the task (6.4). The participants liked each feature of Tabby (texture generation: 6.7, auto-completion: 6.1, drag-and-drop manipulation: 5.5, and sketching: 5.7).
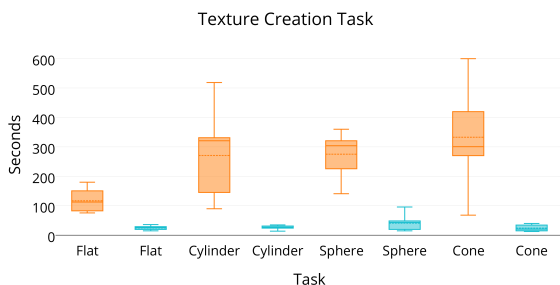

Figure 8: *Completion time of the texture creation tasks with the reference tool (orange) and Tabby (light blue).*

6. Conclusion

We present Tabby, an interactive tool to support designing and applying 3D printable textures on an arbitrary complex surface of the existing object. To minimize repetitive manual efforts, we adopt the *auto-completion* metaphor, which automatically infers the user's demonstration and suggests the possible desired patterns. To enable this, we develop a series of techniques which infer the user's intention, select the semantic region, and convert 2D shapes into 3D textures. Our experiment shows that Tabby enables the participants to create and modify 3D texture much faster than conventional tools, especially for complex target surfaces.

References

[DLL*15] DUMAS J., LU A., LEFEBVRE S., WU J., MÜNCHEN T., DICK C., MÜNCHEN T.: By-example synthesis of structurally sound patterns. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 137. 2

[HAC16] HUDSON N., ALCOCK C., CHILANA P. K.: Understanding newcomers to 3d printing: Motivations, workflows, and barriers of casual makers. In *Proceedings of CHI* (2016), ACM. 2

[KIZD12] KAZI R. H., IGARASHI T., ZHAO S., DAVIS R.: Vignette: interactive texture design and manipulation with freeform gestures for pen-and-ink illustration. In *Proceedings of CHI* (2012), ACM, pp. 1727–1736. 2

[LBW*14] LU J., BARNES C., WAN C., ASENTE P., MECH R., FINKELSTEIN A.: Decobrush: Drawing structured decorative patterns by example. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 90. 2

[LS10] LANDRENEAU E., SCHAEFER S.: Scales and scale-like structures. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 1653–1660. 2

[MLGF09] MATEJKA J., LI W., GROSSMAN T., FITZMAURICE G.: Communitycommands: command recommendations for software applications. In *Proceedings of UIST* (2009), ACM, pp. 193–202. 1

[PXW18] PENG M., XING J., WEI L.-Y.: Autocomplete 3d sculpting. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 132. 2

[SH02] SHEFFER A., HART J. C.: Seamster: inconspicuous low-distortion texture seam layout. In *Proceedings of the conference on Visualization'02* (2002), IEEE Computer Society, pp. 291–298. 3

[SS10] SCHMIDT R., SINGH K.: Meshmixer: an interface for rapid mesh composition. In *ACM SIGGRAPH 2010 Talks* (2010), ACM, p. 6. 2

[TCKP15] TORRES C., CAMPBELL T., KUMAR N., PAULOS E.: Hapticprint: Designing feel aesthetics for digital fabrication. In *Proceedings of UIST* (2015), ACM, pp. 583–591. 2

[TSS*11] TAKAYAMA K., SCHMIDT R., SINGH K., IGARASHI T., BOUBEKEUR T., SORKINE O.: Geobrush: Interactive mesh geometry cloning. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 613–622. 2

[XCW14] XING J., CHEN H.-T., WEI L.-Y.: Autocomplete painting repetitions. *ACM Transactions on Graphics (TOG) 33*, 6 (2014), 172. 2

[XWSY15] XING J., WEI L.-Y., SHIRATORI T., YATANI K.: Autocomplete hand-drawn animations. *ACM Transactions on Graphics (TOG) 34*, 6 (2015), 169. 2

[ZHW*06] ZHOU K., HUANG X., WANG X., TONG Y., DESBRUN M., GUO B., SHUM H.-Y.: Mesh quilting for geometric texture synthesis. *ACM Transactions on Graphics (TOG) 25*, 3 (2006), 690–697. 2

[ZT10] ZHENG Y., TAI C.-L.: Mesh decomposition with cross-boundary brushes. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 527–535. 3