



Fabricating Interactions through Optical Pathways

Himani Deshpande

Texas A&M University
College Station, TX, USA
hdshpande11@tamu.edu

Tolga Yildiz

Texas A&M University
College Station, TX, USA
tolgayildiz@tamu.edu

Madelein Villegas

Texas A&M University
College Station, TX, USA
madeleinvillegas@tamu.edu

Ergun Akleman

Texas A&M University
College Station, TX, USA
ergun.akleman@gmail.com

Jeeeon Kim

Texas A&M University
College Station, TX, USA
jeeeon.kim@tamu.edu

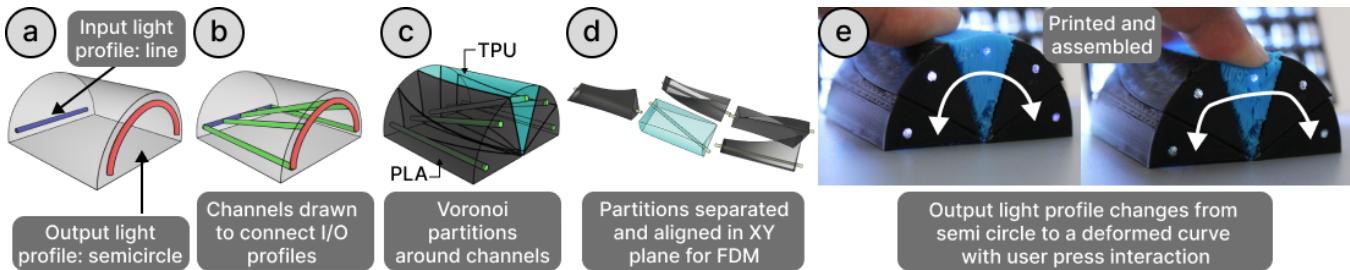


Figure 1: Two curves (a), are used for sampling 5 light channels (b), which are used for Voronoi partitioning (c), each partition can be assigned to a different material and fabrication method (d), optical channels are printed in transparent PLA in-place with the black shell in PLA and blue shell in TPU. Modularity enables material based interactivity with optical elements (e).

Abstract

Transparent materials transmit light without significant scattering or absorption due to *total internal reflection*. Transparent channels in 3D printed objects follow this theory, functioning similarly to optical fibers by transmitting input light. While prior work enabled basic interactions in custom optical sensors like push and display interactions using photopolymers, complex channeling remains challenging, particularly in desktop Fused Deposition Modeling (FDM) due to the inherent printing discontinuities. We present a framework that enables low-cost desktop fabrication of optical interactive devices using Voronoi-based segmentation of objects for in-place FDM printing of optical channels. It allows uninterrupted light transmission where the embedded channels are aligned in the XY plane for uninterrupted printing. We further explore the use of FDM optical objects in routing and channeling strategies to support applications in displays, sensing, and embodied interactions.

CCS Concepts

- Human-centered computing → Human computer interaction (HCI).

Keywords

FDM, Modeling, Optical Channels



This work is licensed under a Creative Commons Attribution 4.0 International License.
SCF '25, Cambridge, MA, USA
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2034-5/25/11
<https://doi.org/10.1145/3745778.3766660>

ACM Reference Format:

Himani Deshpande, Tolga Yildiz, Madelein Villegas, Ergun Akleman, and Jeeeon Kim. 2025. Fabricating Interactions through Optical Pathways. In *ACM Symposium on Computational Fabrication (SCF '25), November 20–21, 2025, Cambridge, MA, USA*. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3745778.3766660>

1 Introduction

Optical fibers harbor potential in HCI applications beyond their traditional usage in telecommunications and medical imaging. While previous works have explored light delivery through high-resolution techniques for 3D printed transparent channels [Pereira et al. 2014; Willis et al. 2012], the use of low cost, desktop Fused Deposition Modeling (FDM) 3D printing to investigate what transparent channels can afford in interactive devices has been underexplored. We envision FDM-printed optical channels without specialized hardware, lowering access barriers for diverse users.

We present a framework that applies Voronoi-based segmentation to 3D models, enabling modular fabrication of objects with continuous transparent channels embedded within each printed segment. These channels effectively route and guide light through FDM-printed structures while maintaining optical continuity (Figure 1(a-d)). To develop this framework, we first examine the light transmission capabilities of transparent materials available for FDM, i.e., PolyLactic Acid (PLA), and Thermoplastic PolyUrethane (TPU), focusing on how geometric parameters influence light propagation, including channel length, diameter, and curvature. Additionally, we examine how printing parameters such as layer height, line width (which affect the diameter of each printed line within the channel),

and raster angle (which determines the orientation of the printed lines) affect the light transmission.

Finally, employing the modular fabrication approach, each segmented partition is separately printed and assembled enabling complex interaction design around light, as a controlled signal source for communication. Our modular fabrication strategy not only simplifies the fabrication process on standard desktop printers but also empowers users to experiment with different materials and interaction design techniques. Parts of the target object can be printed in different materials for extent of light transmission and mechanical properties of the desired material, affording different light re-routing interactions (e.g., pushing, squeezing, or rotating). Complex routing or multiplexing within the object can enable flexibility for smart functional design such as adjusting light paths or altering the object’s interactive capabilities in real-time (e.g., Figure 1(e)). In sum, we contribute:

- Systematic evaluation of characteristics of transparent filaments for FDM printing for their potential as optical channels,
- Voronoi based computational geometry processing method to partition 3D objects with embedded optical channels for in-place printing, and
- Design space of light routing and channeling from input to output, for sensing, and dynamic information display.

2 Background

2.1 Optical Devices in HCI

Optical elements have been used extensively in Computational Fabrication for HCI research to enable sensing and information transmission. Examples include optical fiber touch sensors for interactive tables [Go et al. 2012] and touch screens [Holz and Baudisch 2013], room-scale displays [Swaminathan et al. 2020], textiles [Hashimoto et al. 2013; Olwal et al. 2018], and grasp-sensitive surfaces [Wimmer 2010]. To enhance flexibility, novel methods such as cellulose-based applications [Guridi et al. 2023] and transparent elastomers [Yao et al. 2014] have been explored. In the 3D printing domain, *Printed Optics* [Willis et al. 2012] demonstrated the creation of optical channels through PolyJet printing for sensing and simple to advanced, curved displays [Brockmeyer et al. 2013], AR mapping [Tone et al. 2020], and optimized routing [Pereira et al. 2014] of printed channels. Although PolyJet printing provides significant advantages in creating arbitrarily shaped optical channels for assembly-free interactive device fabrication, we demonstrate how FDM can be used to create optical channels for display and sensing. Furthermore, due to the fine-grained control over printing parameters and channel geometry, we show how light channeling can be varied through parameter changes.

2.2 Optical Interactivity in Passive 3D Objects

Interactivity is embedded through geometric or material encodings, only detectable by machines such as subsurface air cavities readable via structured light [Li et al. 2017], layered barcodes [Maia et al. 2019], and infrared-fluorescent inlays for markerless tracking [Dogan et al. 2023]. Lenticular structures have been used to produce view-dependent images [Zeng et al. 2021]. Surface anisotropy enables directional optical patterns (e.g., barcodes or icons) that are visible under specific lighting angles [Lu et al. 2025; Ma et al. 2023].

Embedded pipes act as light guides, or housings for post-processing elements [Savage et al. 2014]. While these prior works demonstrate approaches for embedding interactivity into 3D printed objects utilizing optics theory, most focus on isolated optical effects or tagging mechanisms.

In this work, we reframe optical channel routing as a design and interaction space, providing low cost ways of fabricating complex N:M input-output mappings. Unlike previous methods where optical I/O is arranged in a single plane for nearly parallel mapping [Willis et al. 2012], or limited to 1:1 or 1:N through auto-routing algorithms [Savage et al. 2014], our approach uses the generated channels to segment the entire object into discrete and printable modules which allows users the flexibility for post-print modifications and reconfiguration.

2.3 Fabricating Multi-Component Objects

Printing optical channels for sensing and display within FDM 3D objects requires multi-part fabrication. Modular fabrication is an approach for enabling multi-component 3D printing, decomposing a design into smaller 3D printed parts that are later assembled (e.g., [Savage et al. 2014]). In this way, integration of materials with various properties becomes relatively affordable, compared to conventional fabrication methods due to printer size limitations and complexity constraints, making it possible to build large-scale or highly detailed objects through incremental assembly [Huang et al. 2023; Moya Muñoz et al. 2024; Sun et al. 2021]. Modularity simplifies creating complex moving structures, separating functional and aesthetic sub-parts [Faruqi et al. 2023], enabling interactive behaviors [Chen et al. 2021], and accelerating prototyping by allowing parallel fabrication [Yildiz et al. 2023], significantly shortening overall build time [Ebert et al. 2025; Nisser et al. 2021], while also producing structurally robust prints that topologically interlock the pieces [Mullins et al. 2022]. Modularity also allows multiple printers to print the object at the same time—Voronoi-based schemes have been employed to divide and schedule printing work between different printers [Stone et al. 2024, 2025]. Our work adopts modular FDM fabrication for optical devices to iteratively build complex objects, precisely place materials, and embed optical components during printing.

3 Working Principles

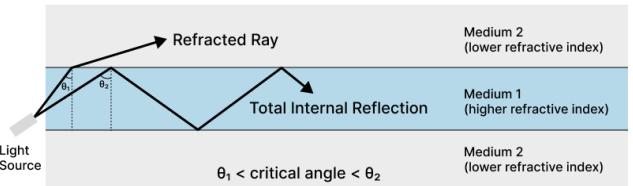


Figure 2: Light incident in a denser medium at an angle greater than the critical angle reflects within the medium.

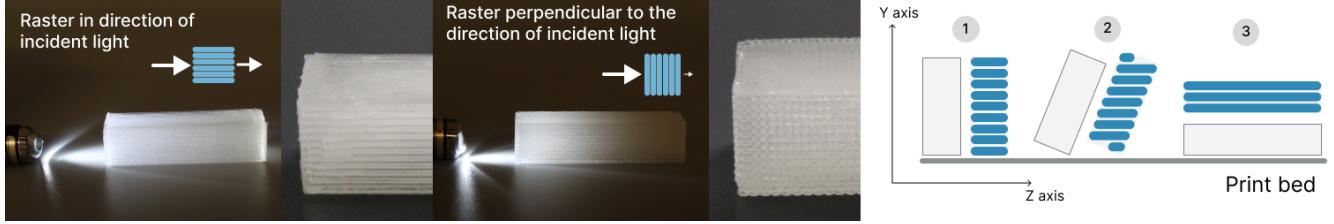


Figure 3: Light transmitted when the raster is parallel, while blocked when raster is perpendicular. To ensure printing channels such that the channel raster is horizontal and continuous, we need to print each channel in one plane.

3.1 Light Transfer in Optical Fibers & FDM Transparent Channels

Traditional optical fibers guide light via *total internal reflection* (see Figure 2) at the boundary between a higher refractive index inner core and a lower-index outer cladding.

Our empirical findings indicate that light can be visibly transmitted through a filament or transparent 3D printed channel over short distances typical in desktop 3D printing. However, the angle of light incidence to the ‘raster’ of the channel i.e. the direction of the extruder movement, affects how much light passes through the printed channel. When the raster lines are perpendicular to the incident light, reflection loses most of the light while traveling (Figure 3 (middle)). However, if the raster aligns parallel with the incident light, most of the light transmits (Figure 3 (left)).

To be able to transmit light through FDM printed channels, the channel needs to be printed in continuous strokes (see Figure 3 (right (3))), i.e. the channel needs to lie in the XY plane. If the channel is in any other orientation, the channel is disrupted while printing layer by layer (see Figure 3 right (1-2)). This planar routing motivates our Voronoi-based segmentation method, which decomposes complex objects into planar modules. While less appealing than PolyJet, that continuously cures tiny liquid droplets blurring the layer boundaries, the layers introduced in the channels through FDM present interesting parameters for light transfer which can enable controlled transmission and blocking for binary signaling (0 and 1) using one material. In that sense, a single transparent filament can print both the channel and the shell (Figure 4).

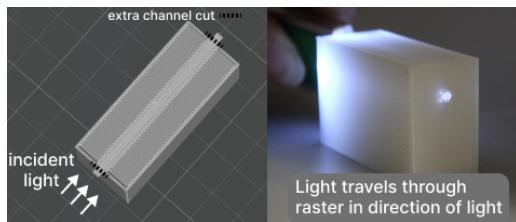


Figure 4: Object printed in same transparent material only transmits light where the raster aligns with the incident light direction and not where the raster is perpendicular.

3.2 Material Considerations and Evaluations

As one of our motivations is to lower the barrier in creating 3D printed channels to afford various interaction spaces, we consider off-the-shelf transparent materials that are commonly used in FDM printing at the time of this work: PLA, PETG, and TPU. Although

PETG showed promising clarity under the lux meter, its sensitivity to temperature and cooling speed led to unpredictable and mostly milky channels. Thus, we focus on PLA and TPU.

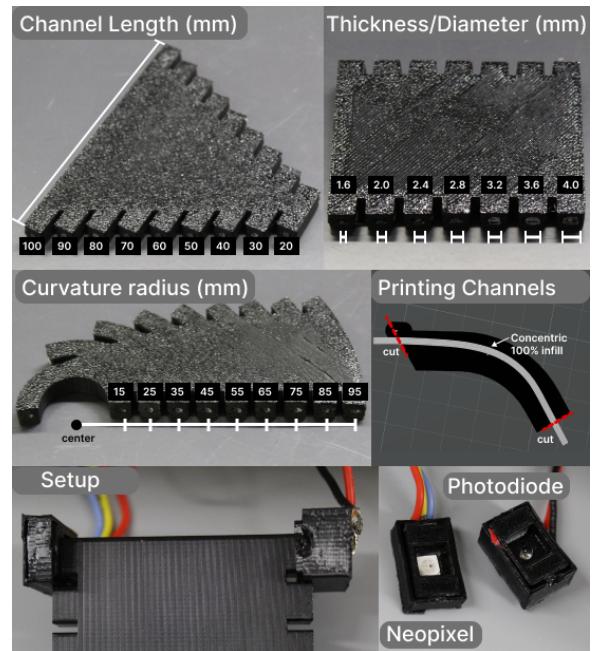


Figure 5: (Top and middle-left) Sample holders and measurement setup. For testing length (20mm to 100mm), thickness (1.6mm to 4.0mm), and curvature (15mm to 95mm). (Middle-right) Process of printing curved channels. (Bottom) Neopixel RGB LED and photodiode on each end.

3.2.1 Experiment Settings. To understand how design affects performance, we tested several geometric parameters: channel length, curvature, thickness, and layer resolution (layer height and line width). For each parameter, we printed sample holders with in-place printed channels, as seen in Figure 5. The length holder tests channel lengths from 20mm to 100mm in 10mm increments (thickness constant at 2mm). The thickness holder tests channel thickness (diameter) from 1.6mm to 4mm in 0.4mm increments (length constant at 50mm). The curvature holder tests the channel curve radii from 15mm to 95mm in 10mm increments (length and thickness constant at 50mm and 2mm, respectively). Finally the layer height/line width tests 0.2mm, 0.3mm, 0.4mm layer heights with 0.21mm, 0.31mm,

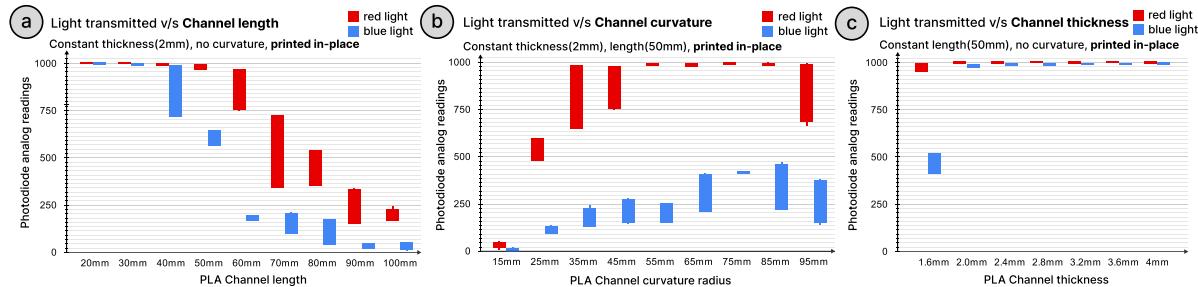


Figure 6: Light transmission through PLA in-place printed channels. Each subfigure shows the photodiode readings for red and blue light across varying parameters (a) length, (b) curvature, and (c) thickness. PLA channels show significant loss, particularly for blue light, with increasing length, tighter bend, and smaller thickness.

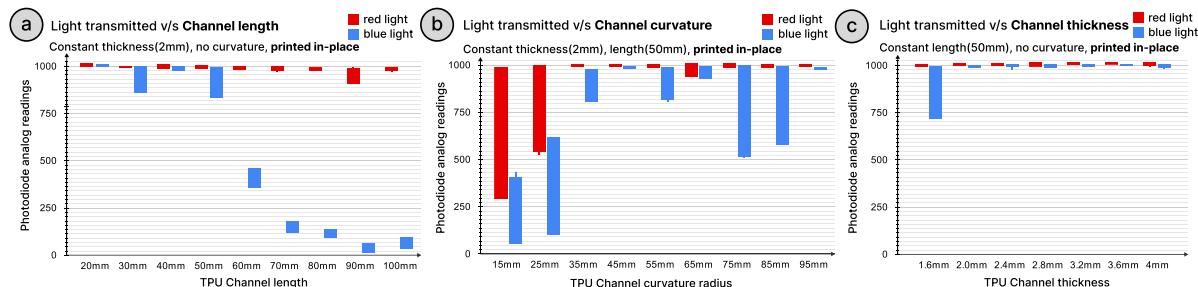


Figure 7: Light transmission through TPU in-place printed channels. Each subfigure shows the photodiode readings for red and blue light across varying parameters (a) length, (b) curvature, and (c) thickness. Like PLA, TPU channels show sharp declines in transmission, particularly for blue light, with increasing length, tighter bend, and smaller thickness.

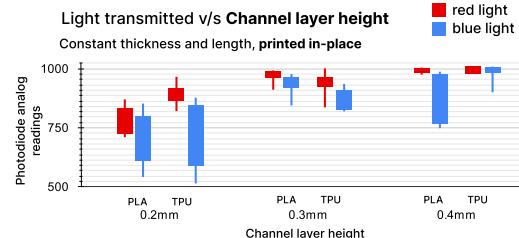


Figure 8: Light transmission through in-place printed channels with varying layer heights and line widths, various materials. Note the Y-axis starts from 500.

0.41mm line widths respectively¹, with overall length and thickness set to 50mm and 2mm respectively.

We used Bambu P1S with AMS for printing with just PLA (both shell and channel) and Flashforge Creator Pro for all samples involving TPU. We used Overture transparent TPU and Giantarm transparent PLA for printing the optical channels. For the printing of all channels, we used concentric 100% infill, an option available on most slicing softwares (we used Ultimaker Cura for Flashforge and BambuStudio). By extending the channels out of the holders, we manually cut off the edges (shown in red dashed lines in Figure 5 (middle-right)) with printed paths perpendicular to the direction of light. Detailed printing parameters can be found in Appendix B

3.2.2 Procedure and Apparatus. Each printed channel was evaluated by projecting light from a programmable RGB LED (Neopixel / WS2812B) and measuring output with a photodiode (BPW34) on the other end (Figure 5 (bottom)). For circuit diagram, see Appendix A. Light intensities for red (255,0,0) and blue (0,0,255) were tested separately at maximum brightness, and photodiode readings range from 0-1023, corresponding to 0-5V. Green was omitted because it has a broader emission spectrum that overlaps the red and blue ends of the visible light spectrum. 5 analog values for both red and blue lights were noted for each channel in the samples, measuring a total of 3 samples for each experiment. The charts are created using the final total of 15 data points for each test variable with the whisker ends pointing to the maximum and minimum values.

3.2.3 Results. Following are the findings from the evaluations:

- (1) **As the length increased, channels exhibited a noticeable decline in transmission.** As the length increased from 20mm to 100mm, the photodiode readings decreased sharply, especially for blue light which dropped to near zero at longer lengths (Figure 6(a) and Figure 7(a)). However, for the red light, TPU channels did not show much difference in transmission.
- (2) **Channels with tighter curves or smaller bend radii showed a reduction in transmitted light.** The drop was most significant between 15mm and 35mm radii, beyond 35mm the transmission stabilized (Figure 6(b) and Figure 7(b)).
- (3) **Channel thickness had less impact on light transmission.** The channel with 1.6 mm thickness transmitted much less light than the thicker ones, particularly in the blue spectrum (Figure

¹Bambu Studio gives an error if the line width is less than or equal to the layer height

- 6(c) and Figure 7(c)). Blue light was affected more than red, likely due to its shorter wavelength.
- (4) **Increasing layer height and line width resulted in increased transmission.** At 0.2mm, both PLA and TPU showed high variability and lower transmission (Figure 8). By 0.4mm, both exhibited improved transmission and tighter error bars, suggesting reduced scattering due to fewer layer transitions.

Overall, the evaluations can help us determine the geometrical and printing, as well as light wavelength parameters to control based on the desired effect.

4 Channel Design and Object Segmentation

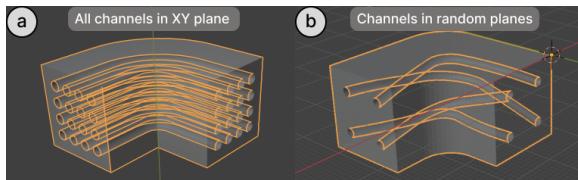


Figure 9: Planar (a) and complex (b) channel alignment

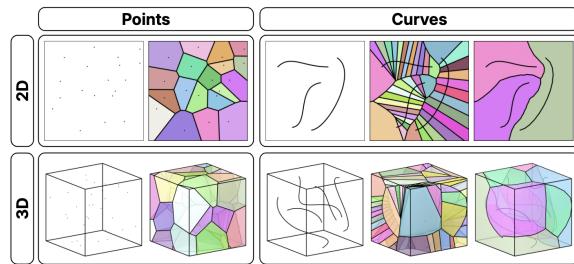


Figure 10: Voronoi algorithm on 2D/3D points and curves.

FDM printing cannot support arbitrary continuous paths along the Z-axis. Discontinuous channels, when printed, suffer from significantly reduced light transmissibility due to misaligned and discontinuous channels (see Section 3.1). To support complex FDM-compatible channel design and light routing, we propose a novel design and segmentation framework that adheres to the physical constraints of FDM. We restrict the design to planar channels, i.e. channels that lie entirely within a single plane. Such channels can be aligned with the XY plane of the print bed, allowing them to be printed seamlessly within a single layer using concentric infill. If all channels lie in the same plane, even with channels at different heights within the object, a single alignment of the object suffices (Figure 9(a)). However, in more complex scenarios where channels must lie in different planes to connect desired points within the object, a single alignment becomes infeasible (Figure 9(b)).

To allow multiple alignment operations to be applied to the object we need to segment the object. Segmentation is done in a way such that each channel belongs to only one segment and each segment covers different parts of the object. To ensure that each segment can be manufactured, we have utilized a Voronoi-based segmentation algorithm by Panchal et al. [Panchal et al. 2023] that divides the object based on the spatial distribution of the channels (Figure 10 (right column)). Voronoi-based segmentation uses channels as input

and produces a segmented model. Each channel is assigned its own segment, which can then be individually reoriented so its channel plane aligns with the XY plane. The design and segmentation workflow is implemented inside Blender using additions to the source code to enable Voronoi operation, and additional Python scripts to provide UI control elements with the built-in sketching tools. For detailed workflow, refer Appendix C.

4.1 Channel Creation

Our design framework enables users to manually define channel curves through the volume of the object, as illustrated in Figure 11. The process is directly implemented in Blender, using Blender's Python API and built-in 3D sketching tools. To begin, the user first selects two endpoints on the object surface. These points define an axis, which serves as the rotation center for a temporary 2D canvas. This plane is essential for two reasons: (1) it ensures that the light path can be sketched on a single plane, making the channel geometry printable via FDM, and (2) it allows the user to route the channel through the solid parts of the object, thus avoiding holes and empty regions. User then adjusts the rotation of the canvas about this axis, allowing fine control of the channel's orientation.

Next, the tool asks users to sketch curves using the built-in 3D sketching tools in Blender, which allow using a plane for projecting the sketches. These sketches define the desired light path. After drawing, the canvas is removed, and the drawn path is approximated using a series of Bézier curves, which offer smooth interpolation and control over curvature. This approach allows fine-tuned routing—for instance, subtly curving a channel from the bottom of the object to reach the eyes of a bunny model (Figure 11). The curve retains the orientation information from the previously defined canvas, and it is used for orienting the segments later.

To ensure robust printability, channels are extended slightly beyond the object's shell. This allows the extruded filament paths to terminate cleanly through manual cutting, minimizing scattering loss, as concentric infill at these endpoints naturally orients perpendicular to the incident light.

4.2 Partition Creation

To segment the object based on the designed channels, we first analyze the channel curves to determine whether partitioning is necessary. If all user-defined curves lie within a single plane, segmentation is not required. In this case, the object can simply be oriented with respect to that plane to facilitate fabrication. This planar scenario is identified and highlighted within the tool.

If the channels cannot be contained in a single plane, we first sample points along each Bézier curve by arc length to ensure a consistent resolution. These points serve as seeds for a 3D Voronoi decomposition. Voronoi is used for segmenting the object because it generates convex cells which are then used to intersect with the object boundary. Assuming the object is printable, boolean operation with a convex domain does not interfere with the manufacturability of the object. Traditional Voronoi diagrams are based on discrete points (Figure 10 (left column)); however, we follow the curve-based Voronoi approach introduced by Panchal et al. [Panchal et al. 2023], where sampled points from the same curve are grouped, and their

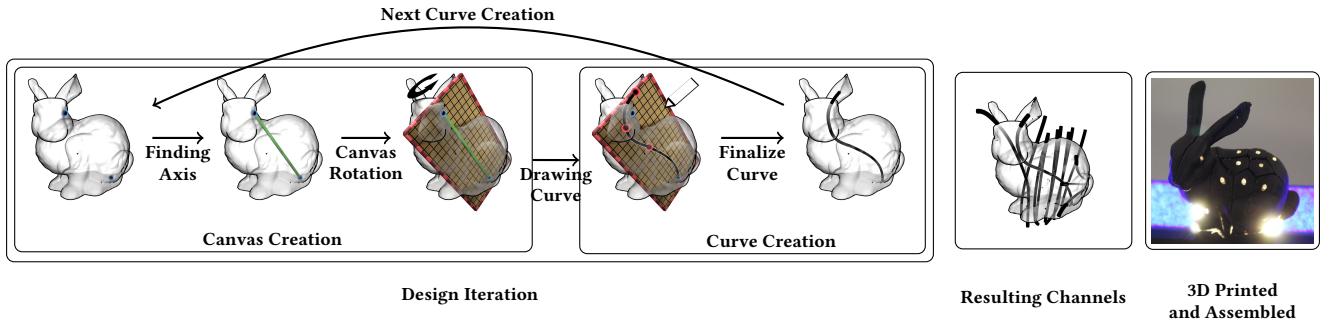


Figure 11: Overview of manual channel creation

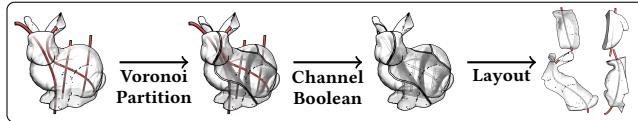


Figure 12: Channels are used for partitioning the shape. Then each partition is aligned according to its “seed” curve such that the light channels will lie in XY plane.

corresponding Voronoi cells are fused to form a region representing the entire curve (Figure 10 (right column)).

This segmentation process is implemented using Blender, where channel curves are modeled using its default curve tools. For volumetric decomposition, we integrate the Voro++ library [Rycroft 2009] into Blender’s Geometry Nodes system by implementing a C++ module that executes within the procedural workflow. Voro++ operates within a bounding box defined slightly larger than the input mesh to ensure that resulting partitions fully encapsulate the geometry and allow for robust Boolean operations.

After computing the Voronoi diagram, we finalize the segmentation using Boolean intersections between the object and each partition. This results in a set of mesh segments. To integrate the channels themselves, we perform an additional Boolean subtraction using swept volumes generated from the Bézier curves. This ensures that each channel is embedded cleanly within its corresponding mesh segment (Figure 12). To orient each segmented piece, we leverage the previously generated canvases to determine the alignment of the associated curves. We then compute the rotation that aligns each canvas with the XY plane and apply this transformation to the corresponding segment. The final design shown in Figure 11 includes eight optical channels routed across the back of the bunny, along with two additional channels that route light from the base to the eyes, demonstrating the capability of the system to handle complex, multi-directional channel placement in a manufacturable manner. Algorithm 1 shows our process pseudocode.

4.3 Parametric Design

To ease the creation of different I/O routing, we extracted parameters into a set of modifier modules, offering users control over parameters defined in the control panel in addition to manual modifications. We have implemented 3 main blocks (see Figure 13) for parametric design:

- **Transform blocks.** User can select the number of channels distributed in a square grid and change the domain size of the block.

Algorithm 1 Channel-Based Object Segmentation

Require: Set of Bézier curves $C = \{c_1, c_2, \dots, c_n\}$ on object mesh M

Ensure: Segmented object with embedded channels

```

1: if All curves lie in a single plane then
2:   Orient object such that curves lie in XY plane
3:   return  $M$  without segmentation
4: end if
5: for all  $c_i \in C$  do
6:   Sample points  $P_i = \{p_1, p_2, \dots\}$  along  $c_i$ 
7: end for
8: Initialize 3D bounding box around  $M$ 
9: Compute curve-based Voronoi diagram using grouped point sets [Panchal et al. 2023]
10: Fuse cells from same curve group  $P_i$  into region  $R_i$ 
11: for all region  $R_i$  do
12:    $S_i \leftarrow \text{BooleanIntersect}(M, R_i)$       ▷ Segment mesh with Voronoi cell
13:    $V_i \leftarrow \text{SweepVolume}(c_i)$             ▷ Generate channel volume
14:    $S_i \leftarrow \text{BooleanSubtract}(S_i, V_i)$       ▷ Embed channel into segment
15:    $S_i \leftarrow \text{OrientSegment}(S_i, c_i)$     ▷ Orient each segment with the associated curve  $c_i$ 
16: end for
17: return  $\{S_1, S_2, \dots, S_n\}$ 

```

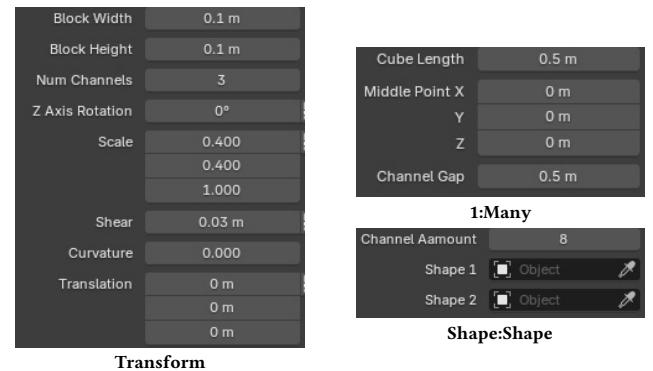


Figure 13: Custom control parameters extracted in Blender.

Transform blocks combine multiple transformations: Rotation, Scaling, Shearing, and Translation. These blocks can be rotated in

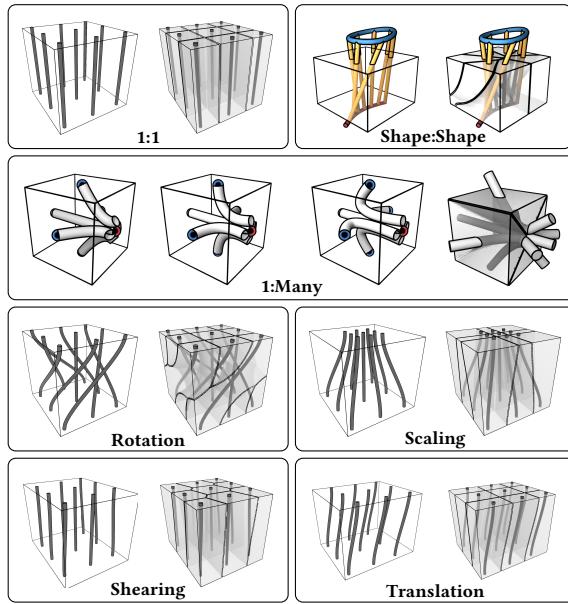


Figure 14: Parameters for design primitives and transforms.



Figure 15: Printed examples of 1 : Many and Shape : Shape routing blocks.

space for manual alignment. The rotation within the block refers to how channel entries are rotated around the Z-axis. Scaling operation scales the endpoints of the curves. Shearing operation moves the end points of the channels on X-axis depending on the Y-position of the end points. Translation operation translates the end points in a given vector direction.

- **1 : Many blocks.** This block enables a channel to be split into multiple branches where the end points are located on the center of the faces of a cube. The parameters include the size of the cube, middle control point of the curves for determining curvature of the channels, and the channel gap size between different curves.
- **Shape : Shape block.** This is a modifier to interpolate new channels between two different input shapes, using the two shapes and the number of channels to be interpolated between them.

Figure 14 highlights the parameters and effects on channel geometry. The script then generates a bounding box for the channels which can be modified using external cut and extrude options. Then the Voronoi-based partitioning modifier creates the partitions of the model. Lastly, another modifier is applied to align all the partitions on a plane such that all channels are printed in the XY plane. These models can then be exported and sliced in any slicer for printing, and assembled with an adhesive as shown in Figure 15.

5 Techniques to Afford Interaction with Optics

We introduce various interaction affordances and their proofs of concepts, starting from a simple mapping and adding on complexities to explain the interplay of control parameters. Note that objects with channels lying in the same plane are not segmented, but printed directly (case in Figure 9(a)). All other objects have been segmented and modularly printed.

5.1 1:1 Parallel Mapping

We can utilize the orientation of the channel geometry in 3D space as embedded in an object to route light from one point to another (a one-to-one mapped display routing), as shown in a toy eye reflecting changing emotions from LCD projection input [Willis et al. 2012]. Similarly, we consider each channel end to be one display point (i.e., one pixel), with multiple channel ends forming a display (see Figure 16), which can be triggered by any light source.

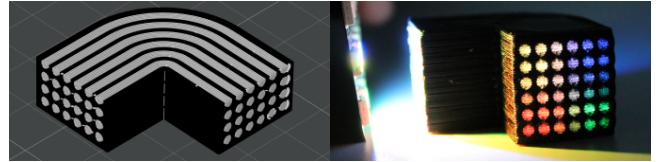


Figure 16: One-to-one light mapping without segmentation

5.2 Non-parallel Mapping

Adding a layer of complexity, we introduce offsets between the input space and output planes by routing channels along paths. This enables transmitting signals to select endpoints based on the routing such as one-to-one, one-to-many, many-to-many connections, etc (e.g., Figure 15).

Slider Mechanism. A slider mechanism presents an interaction where the channels are routed such that a linear input is optically transformed into a semicircular output. The physical translation motion of the slider is converted to a visual rotational motion. Figure 17 shows how moving the slider over static lights presents a semicircle motion of colors, with colors moving to the left, with linear sliding motion to the right.



Figure 17: Switch slides linearly over a light strip showing a circular output light movement.

Object Orientation Sensing. Using a many-to-one channel mapping, it is possible to detect which part of an object is facing a light source. By adjusting the geometric parameters—such as the length and curvature—of multiple PLA channels, we can control how much light each one transmits. When these channels converge at a single sensor, the amount of light received from each channel

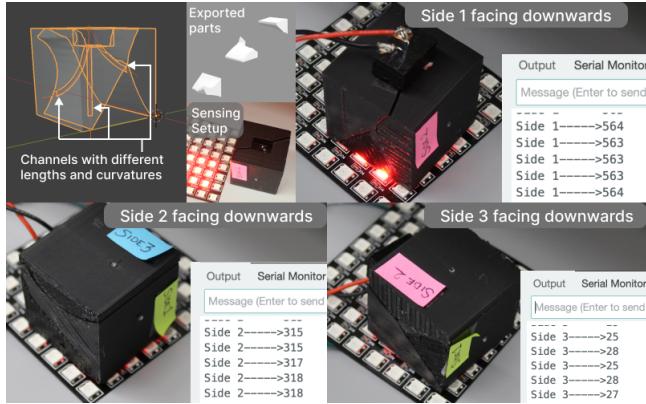


Figure 18: Many:1 channel creation with each channel having different length and curvature, affects amount of light transmitted enabling orientation sensing of the object.

provides information about the object’s orientation relative to the light source. Figure 18 shows how three sides of a cube can be detected to be facing downwards (toward the light) by thresholding the light transmitted by each channel. Each channel is designed with a unique length and curvature to change the amount of light transmitted (refer Subsection 3.2.3 (#1,2)). Similarly, blocking of light, and wavelength can also be sensed through the evaluation data and appropriate thresholding.

5.3 Motion Steering

Steering of light motion reflects a logic or adaptation that governs how output elements move in relation to input light in motion.



Figure 19: Steering light in a clockwise motion gives output light movement in anti-clockwise motion.

Due to the way channels are routed in Figure 19(b), when input light is moved clockwise, the output light moves anticlockwise. This can also be seen from the modified color mapping from input to output (Figure 19(a)).

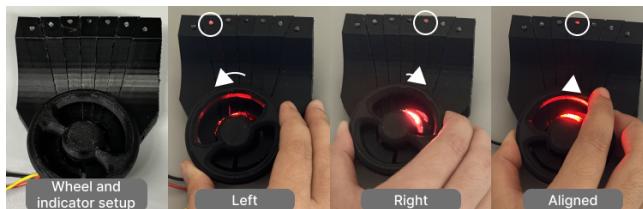


Figure 20: A semicircle to curve mapping enables detecting deviations from the center.

Driving Assistance Indicator. Using a mechanism to steer the light rotationally, we designed a lane alignment driving assistance indicator (see Figure 20) that translates a steering wheel’s left and right movements into a display on the dashboard. When the indicator remains centered, it signifies that the vehicle is properly aligned within its lane. Significant or frequent deviations from the center suggest that the driver is making excessive steering corrections, indicating a need for steadier control.

5.4 Multiplexing and Decoupling

By connecting channels printed in different objects, we can create new light paths. These include interactions afforded through the geometry of the entire object. We show rotary switches to validate this hypothesis, with an internal channel printed in place (Figure 21(a, c) that can connect different input and output channels [Willis et al. 2012]— the output channels here are printed separately and inserted for testing purposes. Plug-and-play rotary switches can be designed to afford different routing behaviors, such that one switch may have a straight channel connecting an input and output in a straight line (see Figure 21(a, b)), while another transmits input light at 90°(see Figure 21(c, d)). While not segmented, this validates the use of FDM printed channels for plug and play interactivity. With our segmentation technique, there is potential to design non-planar switches as well.

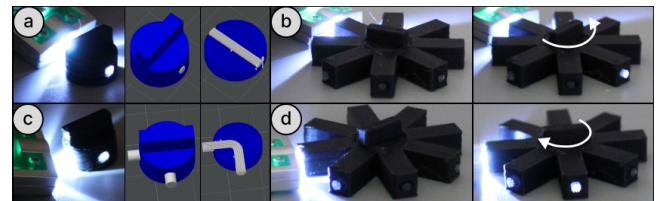


Figure 21: Rotating the replaceable central switch to connect different channels for light routing.

The properties of the material can be used to encase rigid channels in a sparse space, granting some motion space for the channel. To test this hypothesis, we inserted a printed PLA channel in a flexible TPU casing with low infill. We can push the casing to move the embedded channel to point to a different output, creating a push button’s On/Off binary signal to be captured at the receptor (Figure 22). Flexibility of the 3D printed materials affords a wide array of tangible interactions presented in prior fabrication research, such as shear rotation to lock and unlock a door latch [Ion et al. 2016], or bike handles and saddles for comfortable grip and sitting [Kim et al. 2021].

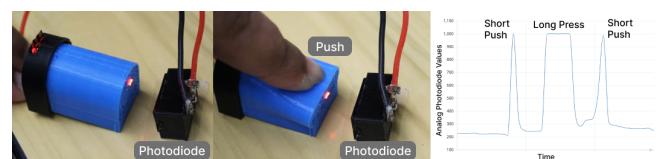


Figure 22: Channel embedded in flexible TPU can be pushed to redirect the light towards a sensor, acting as a push button



Figure 23: Squeezing TPU printed body moves the embedded channels closer together, displaying them through openings in the cap.

Squeeze to Signal Bike Handle. With tested hypothesis, we moved onto in-place printing of the bike handle, with the casing in TPU, allowing the in-place printed PLA channels to move closer together or further apart (Figure 23(a, b)). The resulting mechanical interaction can then be transmitted through internal channels, making the signal visible to others (Figure 23(c, d)). In this setup, different types of interactions can be mapped to distinct meanings. For instance, squeezing the handle could trigger a red light on the handlebar to signal caution. Alternatively, the design could incorporate directional input: a top press might shift a movable channel to connect with a static green-lit channel, signaling that it's safe for others to overtake. While the channels in this example are in the same plane making segmentation unnecessary, we can utilize the same principle for more complex routing similar to Figure 1.

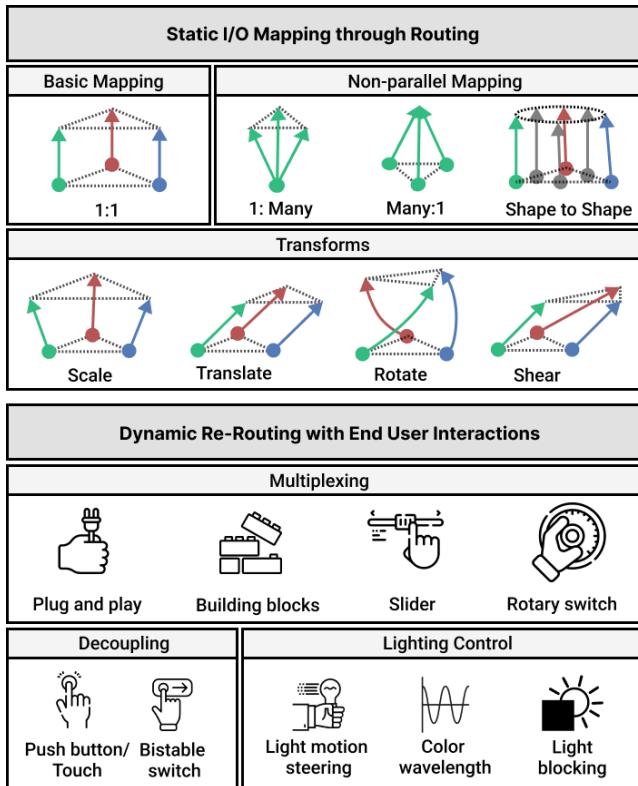


Figure 24: Design Space of Light Routing, Channeling, and User Interactions. Icons from Flaticon².

²Unplug by juicy_fish; Construction by Darius Dan; Slide, Dial, Panic button by Freepik; Slider by manshagraphics; Find by howcolour, Ocean by IconBaandar.

5.5 Design Space

Using this interplay of light, routes, and user interactions, we map out the design space for FDM optical pathways (Figure 24). We define static I/O mapping that stems only from the channel pathways such as displays. Dynamic I/O mapping is expected when dynamic user interaction with the object or input light is present such as plugging in the switch knob, or blocking the channel by grasping a target object. Here, the end user has more control over the light that can be input and the dynamic re-routing of channels. Various mechanisms can enable using light motion together with dynamic re-routing. For example, in the slider mechanism (see Figure 17), moving the slider knob causes movement of output light in relation to the input light at the bottom of the slider. A user may block light from a channel to sense which channel is being blocked or transmitting light based on channel parameters, or even decouple an existing channel for dynamic interactivity (Figure 23).

6 Discussion & Future Work

6.1 Weave Design for Assembly

Currently, we assemble the partitions using external adhesive, however, the connections are not seamless. To address the challenge posed by the lack of physical connections between the partitions, for future work we will encapsulate a weave channel design within the workflow. This approach involves restructuring the object into interlocking weave patterns that provide spaces for the optical channels. The weave design is based on volumetric Bravais weaves defined by Yildiz et al. [Yildiz et al. 2025], which are a generalization of Grunbaum and Shephard's matrix encodings of planar weaves [Grünbaum and Shephard 1988]. We utilize the simplest form of weaves which is the plain weave due to its suitability for creating a regular, interlocking structure. To construct a plain weave, we instance two line segments that are perpendicular to each other on a 2D grid. Then we connect the neighboring weave elements to form a plain weave. A 3×3 plain weave structure is used in Figure 25 to create light channel partitions.

There are other type of weaves such as satins and twill. All the periodic weaves that hang together up to 20×20 repeating domain [Grünbaum and Shephard 1986] have been catalogued. This flexibility enables the creation of custom weaves tailored to specific design requirements. Importantly, because the curves used in the weave channels lie entirely within a plane, i.e. they lack out-of-plane tangents, the design process for these weaves is streamlined, making them easier to implement compared to freeform curves. The weave design can accommodate optical channels and address the assembly related issues of the current partitioning method.

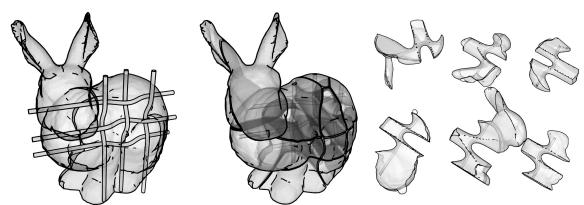


Figure 25: Light channel partitioning with woven channels.

Another approach to assemble the partitions is making use of smaller connector pieces distributed computationally across the partitions. In order to get explicit locations for placing connectors, we tried to use the channels with very few points such that Voronoi tessellation of the points creates a coarse mesh. Using this coarse mesh, we can query corners of the cells and assign connectors to each cell. This approach, however, presents challenges due to the low-resolution sampling of the curves, which leads to inconsistent neighboring information between curves. The ongoing exploration in creating parametric connectors would also make the assembly process more seamless.

6.2 Geometry, Material, & FDM Limitations

As seen from the evaluations, channels printed in FDM present some geometric and material limitations. While we test the maximum length of 100mm in the length evaluation similar to prior work [Willis et al. 2012], 3D printed channels do not transmit light over large distances as optical fibers do. Future evaluations will also look at the effect of multiple curves within the same pathway on light transmission, and the effect of printing temperature on different transparent materials. While printing, we found that PETG loses its transparency and becomes milky white at certain temperatures, printing speeds, and cooling speeds. Evaluating this effect on transparency and milkiness of material will enable more control parameters that can be utilized to transmit or block light based on material temperature and similar parameters.

Although our evaluations primarily used Giantarm PLA and Overture TPU, future work will extend to other transparent materials. Leveraging a luminance meter (lux meter) [Gandhi 2017], we aim to systematically characterize additional filaments to expand the material options for optical channel fabrication.

FDM brings in limitations due to its layer-by-layer printing approach. As discussed earlier, channels need to be printed as continuous strokes. Hence, we cannot print channels that have 3D curves such as a helix. Lastly, the resolution of printing is limited due to FDM compared to other approaches such as PolyJet printing. While, theoretically, we can print channels as thin as a printer's nozzle allows, we had trouble printing consistent channels thinner than 1.6mm due to changes in print bed alignment, and print speed causing channels to be pulled off the bed. The minimum consistent thinness we have achieved using FDM is 1.6mm. Further evaluations for print speed, temperature, and alignment consistency are warranted to print consistent thinner optical channels.

7 Conclusion

We showcase how FDM can be used to print in-place optical channels, enhancing the accessibility and customization of FDM optical interactive devices. We systematically evaluated the geometric and material characteristics of printed channels in PLA and TPU. By utilizing a Voronoi-based segmentation script, we have overcome challenges in embedding optical channels within FDM-printed structures, enabling interactive designs and functionality. We characterize the various design possibilities within a design space outlining static and dynamic interactions.

Acknowledgments

We would like to thank Abul Al Arabi for his help and suggestions with the circuit.

References

- Eric Brockmeyer, Ivan Poupyrev, and Scott Hudson. 2013. PAPILLON: designing curved display surfaces with printed optics. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 457–462.
- Yu-Wen Chen, Wei-Ju Lin, Yi Chen, and Lung-Pan Cheng. 2021. PneuSeries: 3D shape forming with modularized serial-connected inflatables. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 431–440.
- Mustafa Doga Dogan, Raul Garcia-Martin, Patrick William Haertel, Jamison John O'Keefe, Ahmad Taka, Akarsh Aurora, Raul Sanchez-Reillo, and Stefanie Mueller. 2023. BrightMarker: 3D Printed Fluorescent Markers for Object Tracking. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–13.
- Matthew Ebert, Ronnie F. P. Stone, John Koithan, Wenchao Zhou, Matt Pharr, Yuri Estrin, Ergun Akleman, Zhenghui Sha, and Vinayak Krishnamurthy. 2025. NoodlePrint: Cooperative Multi-Robot Additive Manufacturing With Helically Interlocked Tiles. *Journal of Manufacturing Science and Engineering* 147, 6 (02 2025), 061002. <https://doi.org/10.1115/1.4067617> arXiv:<https://asmedigitalcollection.asme.org/manufacturingscience/article-pdf/147/6/061002/7422467/manu-24-1396.pdf>
- Faraz Faruqi, Ahmed Katory, Tarik Hasic, Amira Abdel-Rahman, Nayeeemur Rahman, Leandra Tejedor, Mackenzie Leake, Megan Hofmann, and Stefanie Mueller. 2023. Style2Fab: Functionality-Aware Segmentation for Fabricating Personalized 3D Models with Generative AI. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–13.
- Shaunak B Gandhi. 2017. *3D Printing of Transparent Materials for Optical Applications*. Rochester Institute of Technology.
- Kentaro Go, Katsutoshi Nonaka, Koji Mitsuke, and Masayuki Morisawa. 2012. Object shape and touch sensing on interactive tables with optical fiber sensors. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*. 123–126.
- Branko Grünbaum and G.C. Shephard. 1986. An extension to the catalogue of isonemal fabrics. *Discrete Mathematics* 60 (1986), 155–192. [https://doi.org/10.1016/0012-365X\(86\)90010-5](https://doi.org/10.1016/0012-365X(86)90010-5)
- Branko Grünbaum and G. C. Shephard. 1988. Isonemal Fabrics. *The American Mathematical Monthly* 95, 1 (1988), 5–30. <https://doi.org/10.1080/00029890.1988.11971960> arXiv:<https://doi.org/10.1080/00029890.1988.11971960>
- Sofia Guridi, Emmi Pouta, Ari Hokkanen, and Aayush Jaiswal. 2023. LIGHT TISSUE: Development of cellulose-based optical textile sensors. In *Proceedings of the Seventeenth International Conference on Tangible, Embedded, and Embodied Interaction*. 1–14.
- Sunao Hashimoto, Ryoei Suzuki, Youichi Kamiyama, Masahiko Inami, and Takeo Igarashi. 2013. LightCloth: senseable illuminating optical fiber cloth for creating interactive surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 603–606.
- Christian Holz and Patrick Baudisch. 2013. Fiberio: a touchscreen that senses fingerprints. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 41–50.
- Shaoqi Huang, Jiandong Wu, Lulu Zheng, Yan Long, Junyi Chen, Jianlang Li, Bo Dai, Francis Lin, Songlin Zhuang, and Dawei Zhang. 2023. 3D free-assembly modular microfluidics inspired by movable type printing. *Microsystems & Nanoengineering* 9, 1 (2023), 111.
- Alexandra Ion, Johannes Frohnhofer, Ludwig Wall, Robert Kovacs, Mirela Alistar, Jack Lindsay, Pedro Lopes, Hsiang-Ting Chen, and Patrick Baudisch. 2016. Metamaterial mechanisms. In *Proceedings of the 29th annual symposium on user interface software and technology*. 529–539.
- Jeeeon Kim, Qingnan Zhou, Amanda Ghassaei, and Xiang 'Anthony' Chen. 2021. OmniSoft: A Design Tool for Soft Objects by Example. In *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction* (Salzburg, Austria) (TEI '21). Association for Computing Machinery, New York, NY, USA, Article 15, 13 pages. <https://doi.org/10.1145/3430524.3440634>
- Dingze Li, Avinash S Nair, Shree K Nayar, and Changxi Zheng. 2017. Aircode: Unobtrusive physical tags for digital fabrication. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*. 449–460.
- Qian Lu, Xiaoying Yang, Xue Wang, Jacob Sayono, Yang Zhang, and Jeeeon Kim. 2025. LumosX: 3D Printed Anisotropic Light-Transfer. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 820, 21 pages. <https://doi.org/10.1145/3706598.3714124>
- Zehua Ma, Hang Zhou, and Weiming Zhang. 2023. AnisoTag: 3D printed tag on 2D surface via reflection anisotropy. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–15.

- Henrique Teles Maia, Dingzeyu Li, Yuan Yang, and Changxi Zheng. 2019. LayerCode: Optical barcodes for 3D printed shapes. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.
- Gabriel G Moya Muñoz, Oliver Brix, Philipp Klocke, Paul D Harris, Jorge R Luna Piedra, Nicolas D Wendler, Eitan Lerner, Niels Zijlstra, and Thorben Cordes. 2024. Single-molecule detection and super-resolution imaging with a portable and adaptable 3D-printed microscopy platform (Brick-MIC). *Science Advances* 10, 39 (2024), eado3427.
- Cassie Mullins, Matthew Ebert, Ergun Akleman, and Vinayak Krishnamurthy. 2022. Voronoi Spaghetti & VoroNoodles: Topologically Interlocked, Space-Filling, Corrugated & Congruent Tiles. In *SIGGRAPH Asia 2022 Technical Communications* (Daegu, Republic of Korea) (SA '22). Association for Computing Machinery, New York, NY, USA, Article 14, 4 pages. <https://doi.org/10.1145/3550340.3564229>
- Martin Nisser, Christina Chen Liao, Yuchen Chai, Aradhana Adhikari, Steve Hodges, and Stefanie Mueller. 2021. LaserFactory: a laser cutter-based electromechanical assembly and fabrication platform to make functional devices & robots. In *Proceedings of the 2021 CHI Conference on human factors in computing systems*, 1–15.
- Alex Olwal, Jon Moeller, Greg Priest-Dorman, Thad Starner, and Ben Carroll. 2018. I/O Braid: Scalable touch-sensitive lighted cords using spiraling, repeating sensing textiles and fiber optics. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, 485–497.
- Haard Panchal, Ergun Akleman, Vinayak Krishnamurthy, Tolga Talha Yıldız, and Varda Grover. 2023. Curved Space-Filling Tiles Using Voronoi Decomposition with Line, and Curve Segments Closed Under Wallpaper Symmetries. arXiv:2310.15361 [cs.CG] <https://arxiv.org/abs/2310.15361>
- Thiago Pereira, Szymon Rusinkiewicz, and Wojciech Matusik. 2014. Computational Light Routing: 3D Printed Optical Fibers for Sensing and Display. *ACM Trans. Graph.* 33, 3, Article 24 (June 2014), 13 pages. <https://doi.org/10.1145/2602140>
- Chris H. Rycroft. 2009. VORO++: A three-dimensional Voronoi cell library in C++. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 19, 4 (10 2009), 041111. <https://doi.org/10.1063/1.3215722>
- Valkyrie Savage, Ryan Schmidt, Tovi Grossman, George Fitzmaurice, and Björn Hartmann. 2014. A series of tubes: adding interactivity to 3D prints using internal pipes. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) (UIST '14). Association for Computing Machinery, New York, NY, USA, 3–12. <https://doi.org/10.1145/2642918.2647374>
- Ronnie FP Stone, Matthew Ebert, Wenchao Zhou, Ergun Akleman, Vinayak Krishnamurthy, and Zhenghui Sha. 2024. SafeZone: A Topologically-Aware Voronoi-Based Framework for Fast Collision-Free Cooperative 3d Printing. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 88346. American Society of Mechanical Engineers, V02AT02A038.
- Ronnie F. P. Stone, Matthew Ebert, Wenchao Zhou, Ergun Akleman, Vinayak Krishnamurthy, and Zhenghui Sha. 2025. SafeZone*: A Graph-Based and Time-Optimal Cooperative 3D Printing Framework. *Journal of Computing and Information Science in Engineering* 25, 6 (03 2025), 061004. <https://doi.org/10.1115/1.4068117> arXiv:<https://asmedigitalcollection.asme.org/computingengineering/article-pdf/25/6/061004/7444820/jcise-24-1586.pdf>
- Lingyun Sun, Jiaji Li, Yu Chen, Yue Yang, Zhi Yu, Danli Luo, Jianzhe Gu, Lining Yao, Ye Tao, and Guanyun Wang. 2021. FlexTruss: A Computational Threading Method for Multi-material, Multi-form and Multi-use Prototyping. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–12.
- Saiganesh Swaminathan, Jonathan Fagert, Michael Rivera, Andrew Cao, Gierad Laput, Hae Young Noh, and Scott E Hudson. 2020. Optistructures: Fabrication of room-scale interactive structures with embedded fiber bragg grating optical sensors and displays. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 2 (2020), 1–21.
- Daiki Tone, Daisuke Iwai, Shinsaku Hiura, and Kosuke Sato. 2020. FibAR: Embedding Optical Fibers in 3D Printed Objects for Active Markers in Dynamic Projection Mapping. *IEEE Transactions on Visualization and Computer Graphics* 26, 5 (2020), 2030–2040. <https://doi.org/10.1109/TVCG.2020.2973444>
- Karl Willis, Eric Brockmeyer, Scott Hudson, and Ivan Poupyrev. 2012. Printed optics: 3D printing of embedded optical elements for interactive devices. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) (UIST '12). Association for Computing Machinery, New York, NY, USA, 589–598. <https://doi.org/10.1145/2380116.2380190>
- Raphael Wimmer. 2010. FlyEye: grasp-sensitive surfaces using optical fiber. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, 245–248.
- Lining Yao, Jifei Ou, Daniel Tauber, and Hiroshi Ishii. 2014. Integrating optical waveguides for display and sensing on pneumatic soft shape changing interfaces. In *Adjunct Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, 117–118.
- Tolga Yıldız, Ergun Akleman, Vinayak Krishnamurthy, and Matthew Ebert. 2023. A modular approach for creation of any bi-axial woven structure with congruent tiles. *Computers & Graphics* 114 (2023), 357–367. <https://doi.org/10.1016/j.cag.2023.06.017>
- Tolga T Yıldız, Alice C Niemeyer, Vinayak R Krishnamurthy, and Ergun Akleman. 2025. A constructive framework for discovery, design, and classification of volumetric Bravais weaves. *PNAS Nexus* 4, 8 (07 2025), pgaf219. <https://doi.org/10.1093/pnasnex/pgaf219> arXiv:<https://academic.oup.com/pnasnex/article-pdf/4/8/pgaf219/63794488/pgaf219.pdf>
- Jiani Zeng, Honghao Deng, Yunyi Zhu, Michael Wessely, Axel Kilian, and Stefanie Mueller. 2021. Lenticular objects: 3D printed objects with lenticular lens surfaces that can change their appearance depending on the viewpoint. In *The 34th annual ACM symposium on user interface software and technology*, 1184–1196.

A Circuit Diagram

In our experiments, the photodiode was connected in a reverse-bias configuration with a load resistor to form a transimpedance circuit. The cathode of the photodiode was tied to the 5 V supply, and the anode was connected to the Arduino analog input (A0). A 1 MΩ resistor was placed from the analog input to ground, so that the photocurrent generated under illumination flowed through the resistor, developing a voltage that the Arduino's ADC could measure.

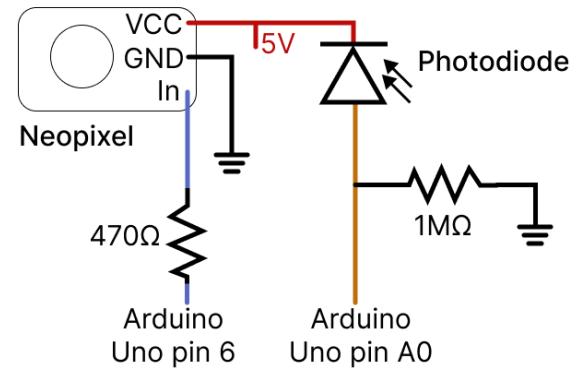


Figure 26: Diagram of circuit used for evaluations.

B Printing Parameters

Table 1 outlines the parameters utilized for printing length, thickness and curvature evaluation samples.

Table 1: 3D printing parameters used for PLA & TPU samples for length, thickness, and curvature evaluations.

Parameter	PLA	TPU
Nozzle diameter	0.4 mm	0.4 mm
Layer height	0.4 mm	0.4 mm
Line width	0.41 mm	0.41mm
First layer height	0.4 mm	0.4 mm
Nozzle temperature	220 °C	228 °C
Bed temperature	55 °C	60 °C
Infill density (channel)	100% (concentric)	100% (concentric)
Infill density (part)	15% (grid)	50% (lines)
Cooling fan	100% after 1st layer	100% after 1st layer

C User Interface for Workflow Inside Blender

As shown in Figure 27 all of the design steps are implemented in Blender. Here is the summary of the steps shown in the figure:

- (1) Select object of interest in Blender viewport

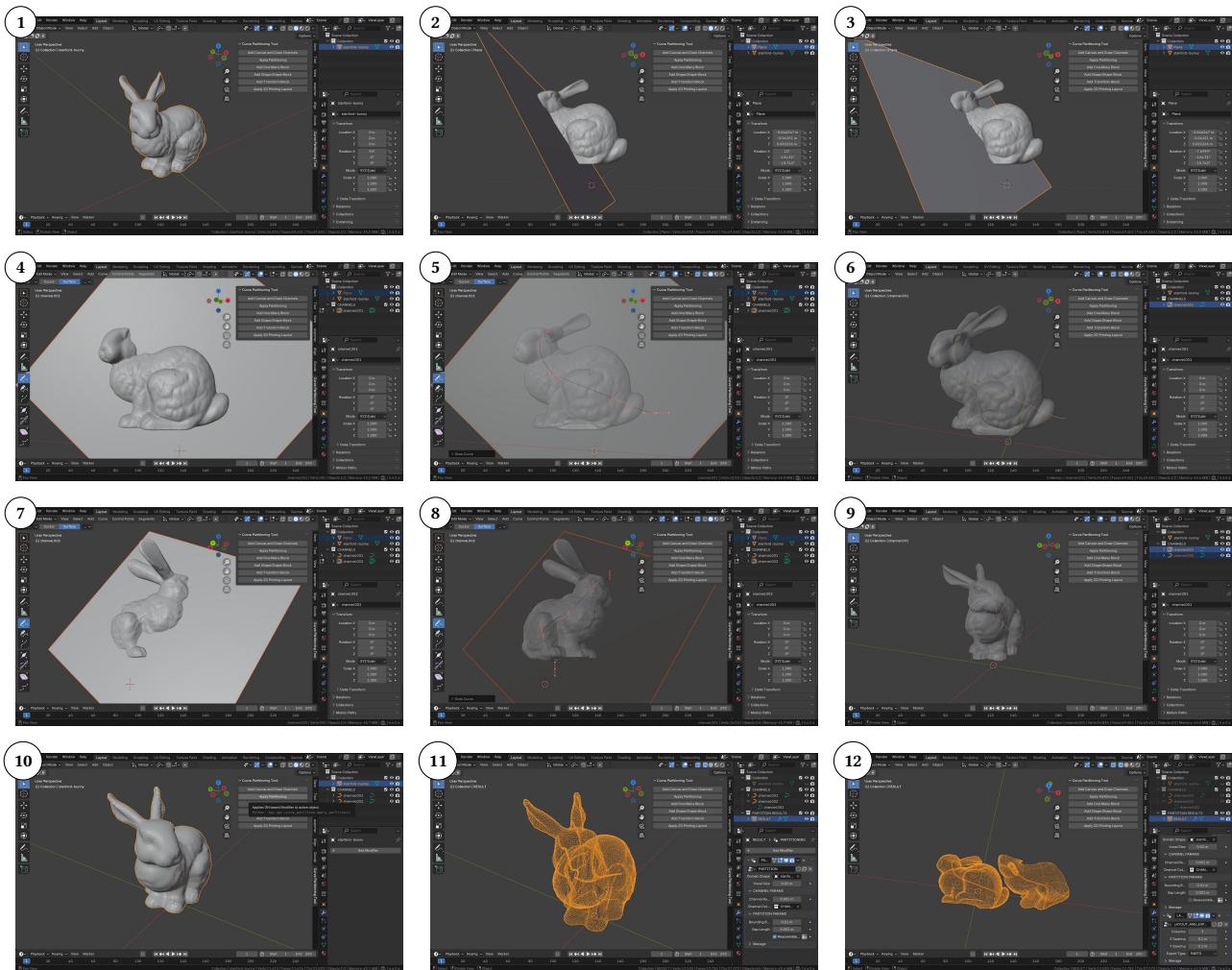


Figure 27: Screenshots from User Interface in Blender

- (2) From the Add-On Menu, go into the *Curve Partitioning Tool* and click on *Add Canvas* and *Draw Channels*. Then select two points on the object. After selecting two points, tool will generate a plane that can be rotated with the mouse movement.
- (3) Once satisfied with the plane orientation, confirm the orientation by clicking right mouse button.
- (4) Context is set to curve drawing mode after confirming plane orientation
- (5) Curves are drawn on the plane, making it easier for users to draw planar curves in 3D. At this stage all curve tools are available, whether it be deleting, drawing multiple curves, or changing vertex locations.
- (6) Once satisfied with the drawn curves, they confirm the curve by pressing escape
- (7) To draw another curve, go over the same actions with clicking on *Add Canvas* and *Draw Channels* on the Add-On menu, select points and orient the drawing plane.
- (8) Draw a new curve
- (9) Once satisfied with all the curves finalize the curves.
- (10) Click on *Apply Partitioning* on the Add-On menu to apply curve based partitioning to the object using Voronoi.
- (11) Last step is to apply a procedural modifier to the object that partitions the object. There are several parameters that can be modified such as channel size, gap between the partitions, etc.
- (12) Click on *Apply 2D Printing Layout* on the Add-On menu to align each of the partitions and adjust the spacing between the partitions. Since all the workflow is implemented in a procedural geometry creation way, user can change parameters by going into the modifier and changing the values.