

Retrofitting Existing 3D Objects with Surface-Conforming Capacitive Sensing

ANDELA ILIC, JUNPENG GAO, ZHIPENG LI, YIJING JIANG, RACHEL SCHUCHERT, MANUEL MEIER, PHILIPP HERHOLZ, AND CHRISTIAN HOLZ, Department of Computer Science, ETH Zürich, Switzerland



Fig. 1. We propose a computational fabrication pipeline that equips existing 3D objects with mutual-capacitance sensing for touch input. Starting from a 3D scan of a physical object (left), our method generates two layers of electrode curves that conform to the object’s surface geometry, tailoring their layout with linear optimization for sensor coverage and distribution (center). We fabricate these curves from copper foil using a vinyl cutter, attach them to the surface, and connect them to a mutual-capacitance controller for live sensing and visualization (right).

Augmenting the surface of 3D objects with capacitive sensing is challenging when their volumes cannot be modified. In this paper, we present a generative computational fabrication pipeline that retrofits surface-only sensor layouts to 3D geometries for multi-touch interaction. Our method scans a real-world object to obtain its 3D mesh, generates and optimizes a 3D sensor design of drive and sense lines for mutual-capacitance sensing under physical and hardware constraints, and unfolds the design into individual 2D stencils that can be cut from conductive material. Our fabrication pipeline cuts these stencils from thin copper foil with a vinyl cutter and then assists manual sensor attachment by projecting the sensor design onto the dynamically registered real-world object. We connect the resulting electrode mesh to a mutual-capacitance scanning controller and resolve touch interaction in real time. We demonstrate our approach with four 3D geometries and evaluate our method and fabrication pipeline on them.

CCS Concepts: • **Computing methodologies** → **Shape analysis**; • **Human-centered computing** → **Interaction devices**.

Additional Key Words and Phrases: Mutual-capacitance sensing, fabrication.

ACM Reference Format:

Andela Ilic, Junpeng Gao, Zhipeng Li, Yijing Jiang, Rachel Schuchert, Manuel Meier, Philipp Herholz, and Christian Holz. 2026. Retrofitting Existing 3D Objects with Surface-Conforming Capacitive Sensing. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26)*, July 19–23, 2026, Los Angeles, CA, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3799902.3811213>

Author’s Contact Information: Andela Ilic, Junpeng Gao, Zhipeng Li, Yijing Jiang, Rachel Schuchert, Manuel Meier, Philipp Herholz, and Christian Holz Department of Computer Science, ETH Zürich, Switzerland.



This work is licensed under a Creative Commons Attribution 4.0 International License. *SIGGRAPH Conference Papers '26, Los Angeles, CA, USA*
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2554-8/2026/07
<https://doi.org/10.1145/3799902.3811213>

1 Introduction

Touch-sensitive surfaces enable direct interaction with physical objects and expand the ways we engage with digital systems. A wide range of domains benefit from this functionality, including toys, tangible interfaces, and, not least, interactive devices. On them, touch sensing registers both contact events and their spatial locations [Grosse-Puppenthal et al. 2017], which becomes increasingly powerful when entire surfaces act as continuous sensors for flat [Parilusyan et al. 2022; Rus et al. 2014; Wessely et al. 2020; Zhang et al. 2018] and non-planar [Nittala et al. 2018; Palma et al. 2024; Savage et al. 2012; Schmitz et al. 2015; Villar et al. 2009] geometries.

However, covering the surface of a given 3D object with a continuous touch sensor is technically difficult. This has often required fabricating the object with embedded sensing components, either through 3D printing conductive materials [Palma et al. 2024; Schmitz et al. 2015] or embedding discrete elements, such as pressure sensing [Parzer et al. 2017] or fiber-based light sensing [Willis et al. 2012]. These methods typically trade off spatial resolution and practicality. Interesting exceptions to this are commercial devices such as the Apple Magic Mouse or Microsoft’s TouchMouse [Villar et al. 2009]; they integrate mutual-capacitance for curved surfaces, albeit for moderate curvature and simple geometry that could accommodate a handcrafted sensor layout.

To augment *existing objects* whose volume cannot be modified, reproduced, or redesigned, previous approaches have instrumented select sub-regions with sensors [Kawahara et al. 2013; Pourjafarian et al. 2019; Savage et al. 2012]. Alternatives have relied on external sensing through cameras to monitor hand-object interaction from one [Cao et al. 2021; Izadi et al. 2011] or more vantage points [Brahmbhatt et al. 2020]. Common in such setups is occlusion during direct manipulation, such that estimating touch contact from vision alone requires resolving both the occlusion of fingers [Streli et al. 2024] and ambiguous object surface geometry [Fan et al. 2024].

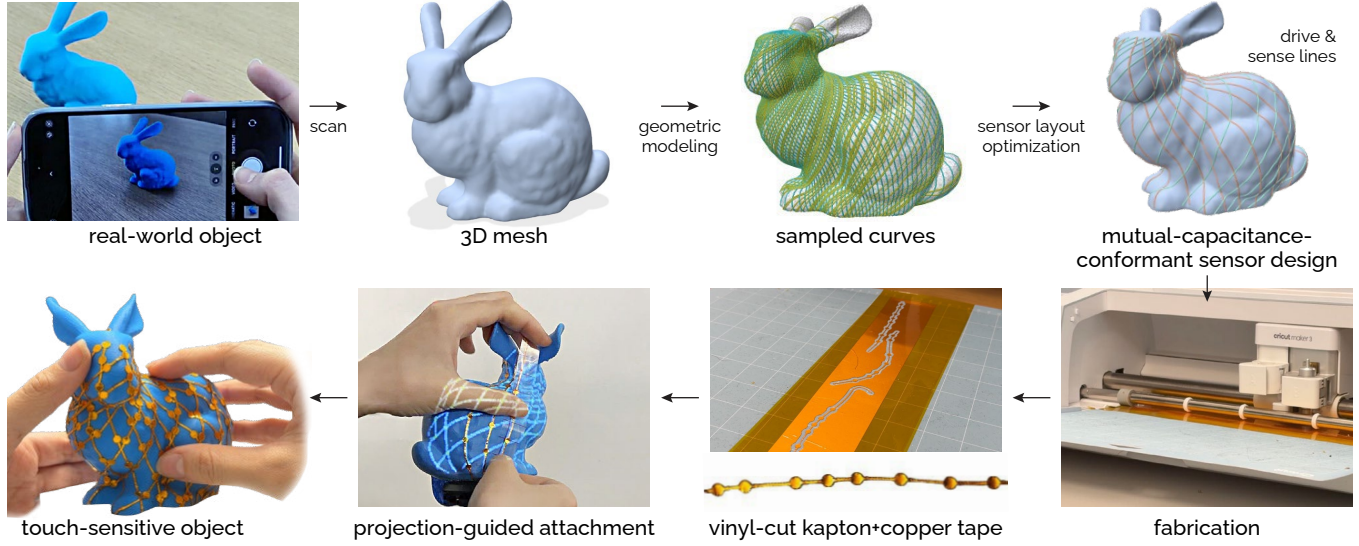


Fig. 2. Our computational fabrication pipeline comprises real-world object scanning, 3D sensor layout generation, optimization for hardware constraints and sensor surface coverage, conductor fabrication, projection-guided sensor attachment, and interactive touch sensing and visualization.

In this paper, we present a generative computational fabrication pipeline for augmenting 3D geometries with capacitive sensing for multi-touch interaction. Our approach generates surface-only sensor layouts for an object’s 3D geometry and requires no modification of its volume to establish real-time multi-touch input across its surface.

1.1 Mutual-capacitance sensing across a 3D surface

Fig. 2 shows our fabrication pipeline for generating a mutual-capacitance sensor layout that conforms to an existing real-world object.

We first scan the object to obtain its 3D mesh for geometry modeling. Our method then generates two dense layers of smooth curves along the surface, one for drive lines and one for sense lines in a mutual-capacitance sensing configuration, such that only curves from different layers intersect. Intersections between layers define the spatial distribution of input locations on the object surface.

Next, we sample curves from both layers to uniformly distribute sensing intersections across the surface under geometric, hardware, and fabrication constraints. We cast this as an optimization problem that maximizes surface coverage and intersection uniformity.

Finally, our fabrication pipeline unfolds the generated 3D curves onto the 2D plane and exports the resulting vector outlines as individual stencils, cut from conductive material using a vinyl cutter. To assist manual attachment, our system projects the optimized sensor design directly onto the 3D object using dynamic registration via an RGB-D camera. After attaching the electrodes to the target object’s surface and connecting them to a digitizer, our signal processing pipeline resolves touches and contact areas across the 3D surface in real time and maps them onto the 3D geometry for visualization.

Our target use cases include objects that cannot be refabricated or internally modified (e.g., a *functional* mouse, ceramic figurine, or wooden toy), as well as thin objects (e.g., plates) without space for interior routing. Our system supports standard touch interactions—tap, swipe, pinch, and multi-touch—on 3D surfaces, such as sliding along curved handles or tapping objects for region-specific input.

1.2 Contributions

We contribute a computational method that generates and fabricates surface-only mutual-capacitance sensor layouts for existing 3D objects while preserving their interior and structure, including:

- (1) a geometric method for generating surface-intrinsic multi-touch sensor layouts that retrofit existing 3D objects,
- (2) a constrained optimization method for determining fabrication-ready sensor layouts with broad and uniform coverage under mutual-capacitance, hardware, and fabrication constraints, and
- (3) a fabrication and assembly process that unfolds the generated 3D routings into planar conductive traces and guides their attachment to the physical object using dynamic projection.

2 Related work

2.1 Mutual-capacitance touch sensing

Widely adopted, mutual-capacitance sensing detects touch by measuring the change in capacitive coupling between two sets of electrodes [Zimmerman et al. 1995]. Drive (Tx) and sense (Rx) lines form orthogonal grids in which electrode pads at each intersection establish capacitors (Fig. 3). A scanning controller sequentially drives Tx lines while measuring responses on Rx lines to create a 2D map of coupling changes caused by finger proximity. This localizes simultaneous touches and has made capacitive sensing the default in phones, tablets, etc. (see the survey by Grosse-Puppenthal et al. [2017] for capacitive sensing designs and applications).

The sensor layout for mutual-capacitance measurements imposes strict geometric constraints: conductors must not intersect within the same layer, and each drive line must intersect each sense line at most once to avoid spatial ambiguity in localizing measured changes in mutual capacitance. Designing sensors for non-planar 3D surfaces thus introduces challenges in routing, intersection control, and fabrication. The Multitouch-Kit proposed by Pourjafarian et al. [2019] showed a promising direction and achieved finger-scale resolution

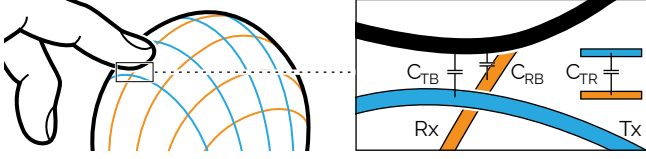


Fig. 3. Mutual-capacitance sensing model with drive line (Tx), sense line (Rx), finger (B), and resulting capacitors. Capacitors to GND and the finger’s RC circuit to GND are omitted for simplicity.

using dense Tx/Rx grids that can be conformed to mildly curved surfaces. Related to our fabrication pipeline, Multitouch-Kit assumes that the sensing layout remains a developable 2D pattern: as surface curvature increases or topology becomes complex, maintaining electrode alignment and spatial correspondence becomes a nontrivial developability problem that is not addressed by Multitouch-Kit.

2.2 Touch sensing on 3D objects

Previous work has explored a variety of techniques to augment 3D objects with touch sensing. Examples of capacitive sensing include embedding conductive traces during 3D printing and manufacturing [Burstyn et al. 2015; Götzelmann and Althaus 2016], post-processing 3D surfaces using screen printing [Olberding et al. 2014] or inkjet printing [Kawahara et al. 2013; Khan et al. 2019], and augmenting objects with dedicated electrodes [Zhang et al. 2019]. Spray coating with conductive materials [Wessely et al. 2020; Zhang and Harrison 2018], hydrographic transfers [Groeger and Steimle 2018], and functional adhesive stickers or patches [Klamka et al. 2020; Strohmeier et al. 2018] can also augment existing objects with capacitive sensing. Therefore, while prior approaches implement sparse or single-electrode sensing and require per-object calibration to localize touches, our grid-based approach inherently implements *spatial* capacitive sensing for multi-touch localization.

Closely related is electric field sensing (EFS). ElecTrick applies conductive paint to an object for large-area touch sensitivity [Zhang et al. 2017]. Touch is localized via tomographic reconstruction between *boundary-attached electrodes* and per-object *calibration*. This means that position estimates are constrained to regions spanned between electrodes and depend on a learned inverse mapping. It is unclear, however, how well EFS can scale spatial sensing to general 3D surfaces or closed geometries where no clear electrode boundary exists and geometric correspondence to the surface is required.

Beyond capacitive sensing, resistive sensing has been applied to curved and soft surfaces to detect touch and pressure input. Example materials to coat existing curved objects include stitched conductive threads [Villar et al. 2018], zebra tiles [Parzer et al. 2017], or deformable substrates [Sundholm et al. 2014]. Alternatively, optical approaches have enabled touch sensing on curved surfaces by custom-shaping acrylic into optical waveguides [Roudaut et al. 2011].

Recent fabrication-oriented approaches have explored routing with low complexity for input detection. Sakura and Kakehi [2025] demonstrated a single-stroke conductive-path touch sensor for 3D-printed objects that minimizes path complexity. Bae et al. [2025] proposed a single-wire interior approach for touch sensing on 3D objects, which reduces instrumentation overhead but constrains spatial resolution to the wire’s trajectory.

2.3 Geodesic layouts

Vector field design has been widely used to guide the sampling of curves on surfaces. Designing tangent vector fields with user control is well-studied [de Goes et al. 2016; Vaxman et al. 2017] and has been applied to generate field-aligned, fabrication-aware planar polylines on surfaces [Cignoni et al. 2014]. Given a set of user-defined constraints, previous methods have used edge-based representations to construct tangent vector fields [Crane et al. 2010; Fisher et al. 2007]. Others used vertex representations [Liu et al. 2016] where users provide example fields or draw strokes directly on the surface. Beyond vector fields, several recent projects directly sample curves on object surfaces [Noma et al. 2024; Yu et al. 2021].

From a geometry-processing perspective, our problem is related to tubular decomposition and curve-network simplification. Shape Representation by Schüller et al. [2018] addresses decomposition into topological cylinders via tubular parameterization. Similarly, Stability-Aware Simplification of Curve Networks [Neveu et al. 2022] is relevant to our pruning stage, which balances structural constraints and coverage objectives under optimization. We refer the reader to the survey by Xiao-Ming et al. [2021] for further information on mapping-based alternatives and their robustness trade-offs.

For the *spatial* mutual-capacitance sensing, curves sampled on the surface of 3D objects must satisfy specific constraints as mentioned above. Palma et al. [2024] presented a fabrication method that embeds electrodes on the surfaces of 3D-printed objects, requiring electrode wiring to be routed through hollow tubes inside the 3D print. Their approach partitions surfaces into quadrilateral patches for the traditional grid-based input localization, but it requires access to and modification of the object’s interior. Even before, Götzelmann and Althaus [2016] fabricated Tx and Rx wires directly within 3D-printed objects during fabrication and similarly relied on internal routing and fabrication-time layout decisions. In contrast, we address the setting of deploying mutual-capacitance touch sensing on existing objects through *surface-only* instrumentation.

3 Method overview

Our approach models surface-based mutual-capacitance sensing on a 3D object as a geometric curve-routing problem on its surface. Our formulation treats each conductor as an independent surface-intrinsic curve, with sensing locations defined by pairwise intersections between curves from two layers. This abstraction enables spatial touch sensing on 3D geometries.

3.1 Problem definition

We define two sets of surface curves, A for drive lines and B for sense lines, subject to the following constraints:

- (1) Curves within the same layer must not intersect to ensure proper and unambiguous drive and sense behavior.
- (2) A curve in one layer may intersect a curve in the other layer at most once, so that each measurement corresponds to a unique sensing location.
- (3) All curves must connect to the object’s base, which accommodates electronic connections to the scanning controller.

We formalize these constraints in Section 6 and extend them with fabrication feasibility constraints.

4 Scanning real-world objects

Our pipeline begins by acquiring the 3D mesh of a given real-world object. Our system integrates SAM3D [Yao et al. 2025] to continuously track a 3D object moved and rotated in front of an Azure Kinect camera to extract its initial surface geometry. We load the scan into MeshLab for mesh repair, downsampling, and isotropic remeshing to standardize the input into our geometry pipeline.

5 Geometric modeling to sample intrinsic curves

Given a 3D mesh, regions of the surface are designated for touch sensing, along with a non-sensing region that accommodates wiring to the scanning controller. Fig. 4 illustrates this process. In practice, this non-sensing region corresponds to the object base where it rests on a supporting surface. Starting from a seed face, we identify the contiguous base region by expanding across neighboring faces whose normals deviate by less than an angular threshold, thereby detecting the perimeter of the base. For objects such as cups or the bunny, this region typically spans the full footprint of the object.

On the remaining sensing surface, two layers of smooth, surface-intrinsic curves are generated to represent drive (Tx) and sense (Rx) conductors. Curve generation is guided by discrete trivial connections [Crane et al. 2010], which define consistent transport of tangent directions across mesh edges. This intrinsic formulation enables coherent curve families on arbitrary meshes without requiring a global surface parameterization, while ensuring that all conductor paths remain confined to the surface. Explicit intrinsic tracing is preferred over parameterization because it avoids mesh cutting and surgery required by parameterization, and it naturally enforces the single-intersection constraint without post-processing. The overall layout of each curve family is controlled by a small set of singularities that determine the topology of the induced direction fields.

5.1 Computing trivial connections

We follow Crane et al. [2010] and represent a discrete trivial connection by assigning each mesh edge a rotation angle that specifies how a tangent vector is transported between adjacent triangles. The vector field assigns one oriented vector per face, with the trivial connection represented as an angle per dual edge. Given a connection, we trace curves by integrating within a triangle until reaching an edge, transporting the tangent direction using the stored rotation, and continuing in the neighboring triangle.

All examples in this paper use genus 0 meshes and singularities of index 1. Singularity placement is defined relative to the non-sensing base region. We grow the first exclusion region by face expansion until either a geodesic-distance threshold or a normal-deviation threshold is reached, and we define the second exclusion region using geodesic distance alone. The Tx and Rx layers use different singularity placements, as described next. (See Crane et al. [2010] for discretization details and mathematical properties.)

5.2 Intrinsic curve sampling

Given a trivial connection and its associated singularities, curves are sampled by tracing integral curves on the surface from prescribed seed locations and initial directions. Tracing proceeds until a termination condition is met.

The Tx layer places a singularity at the center of the non-sensing base region, and curves are traced from a seed point defined as the point on the sensing surface with maximum geodesic distance from the base region. Initial directions are sampled in the tangent plane, yielding curve families that distribute evenly across the sensing surface. The Rx layer places two adjacent singularities at the center of the non-sensing base region, and curves are traced from a set of starting locations along the base boundary. Initial directions are sampled relative to the boundary normal, producing curves that extend from the base into the sensing region, promote intersections with the Tx layer, and remain connected to the base.

Curve tracing terminates as trajectories approach singularities. Tracing can also halt earlier when vectors on adjacent faces both point toward their shared edge. We detect this case explicitly; it occurs rarely in our pipeline, in about 3% of traces, due to isotropic remeshing preprocessing. Intra-layer intersections are additionally prevented by constraint C1 in the integer linear programming formulation. In addition to reflecting the topology induced by the connection, this criterion improves fabrication robustness, as curves tend to exhibit increased curvature near singularities.

5.3 Implementation

We sample the two layers of intrinsic curves using two different strategies to ensure complementary coverage of the surface (Fig. 4). For the first layer, we generate curves by sampling initial directions at fixed angular intervals in the tangent plane. All curves originate from one singularity and extend toward the other singularity. These curves are sampled at uniform angular intervals, resulting in a regularly spaced and approximately uniform layout. For the second layer, we vary the curve orientation to complement the first layer. We first select starting faces whose geodesic distance to one singularity is identical. Starting faces are selected by increasing the geodesic-distance threshold around the singularity until we obtain the required number of candidate faces for tracing. From each starting face, curves are traced toward the opposite singularity with an initial angular offset. We sample this initial angle in the range $[-50^\circ, 50^\circ]$ at 10° intervals, generating one candidate curve set per angle. A wider angular range produces undesired multiple intersections between curve pairs, while finer angles increase runtime without improved quality. We sample 100 curves for each layer.

6 Design optimization for capacitive sensing

Having sampled two layers of intrinsic curves on the object surface, we must now ensure that they meet the requirements for mutual-capacitance sensor designs. Our method can sample large numbers of smooth curves, but fabricating them for real-world objects dictates additional feasibility constraints. Beyond the constraints listed above, these constraints include augmenting Tx and Rx line intersections with small electrode pads to increase the signal-to-noise ratio for sensing touches, which may intersect with other curves, and accounting for the digitizer’s limited number of Tx and Rx pins.

Therefore, we adapt the layout of Tx and Rx lines by selecting the optimal subset of curves from each layer such that the intersections between the two layers are distributed as evenly as possible across the surface. A second optimization stage then selects a subset of

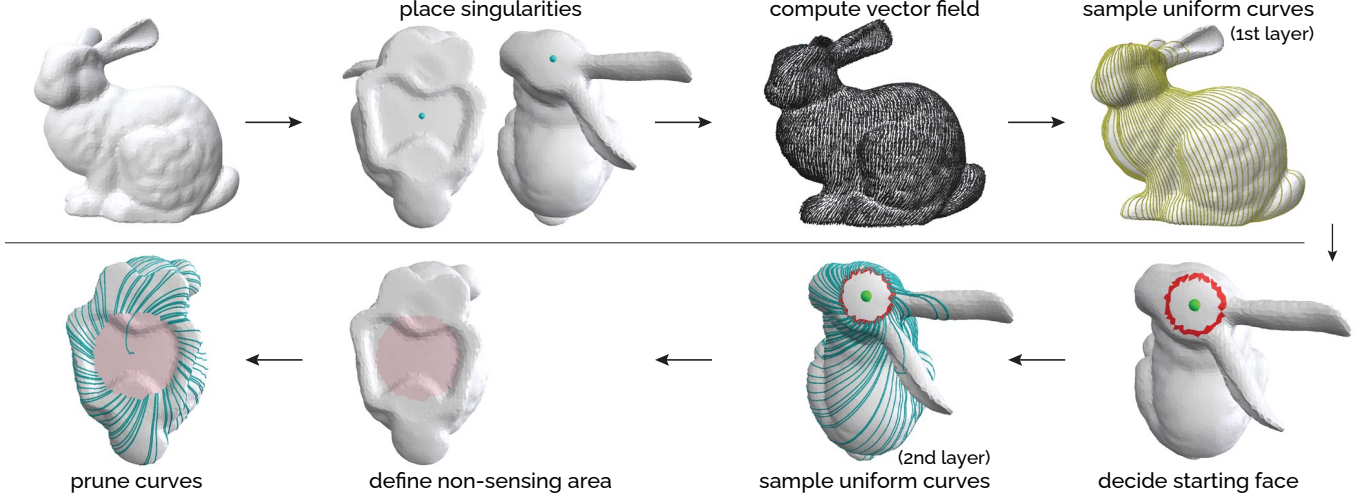


Fig. 4. The curve layout is sampled as follows. For a 3D mesh, we first place singularities in non-sensing regions of the surface (e.g., the bunny’s base and head). Next, we compute a vector field and uniformly sample the first layer of curves. We then select starting faces for the second layer and sample curves at different angles. Finally, we refine the non-sensing regions and trim the curves at their boundaries.

intersections at which pads are placed while maintaining sufficient spatial separation between them. We formulate this layout selection as an integer linear programming problem.

6.1 Input & Output

The sensor design optimization proceeds in two phases. The input consists of two layers of sampled intrinsic curves. Let the first and second layers be denoted as

$$A = \{a_1, a_2, \dots, a_m\}, \quad B = \{b_1, b_2, \dots, b_n\}, \quad (1)$$

where each curve $a_i \in A$ and $b_j \in B$ is an intrinsic curve sampled using the method described in subsection 5.2.

Phase 1: Curve selection. We select subsets $A' \subseteq A$ and $B' \subseteq B$ such that the intersections between curves in A' and B' are distributed evenly over the object surface.

Phase 2: Electrode pad placement. We maximize the number of electrode pads placed at intersections while enforcing a minimum distance constraint between any pair of selected intersections to avoid signal interference. From the intersections between the curve subsets A' and B' , we select a subset at which electrodes are placed.

Optimization output. is (1) the selected subsets of curves A' and B' , and (2) the subset of intersections at which electrode pads are placed for each selected curve.

6.2 Phase 1: Curve selection details

$$\text{Let } I = \{i_{(a_i, b_j)} \mid a_i \in A, b_j \in B\}, \quad (2)$$

denote the set of all candidate intersections, where $i_{(a_i, b_j)}$ represents the intersection point between curves a_i and b_j , if it exists. To quantify surface coverage, we uniformly sample the object surface to obtain a reference set

$$S = \{s_1, s_2, \dots, s_K\}, \quad (3)$$

which serves as a discrete approximation of the surface geometry during optimization. For each intersection $i \in I$, we compute its geodesic distance to the closest sampled surface point:

$$d_i = \min_{s_k \in S} \text{dist}_{\mathcal{M}}(i, s_k). \quad (4)$$

$\text{dist}_{\mathcal{M}}(\cdot, \cdot)$ denotes the geodesic distance along the surface manifold. We select subsets of curves from the two layers such that the subset of active intersections $I' \subseteq I$ minimizes the maximum distance between any active intersection and its closest sampled point:

$$\min d_{\max}, \quad \text{where } d_{\max} = \max_{i \in I'} d_i. \quad (5)$$

This objective ensures that every region of the surface lies within a bounded distance of at least one sensing location.

As a secondary objective, we minimize the average distance

$$\min \frac{1}{|I'|} \sum_{i \in I'} d_i, \quad (6)$$

to discourage clusters and promote even spatial distribution.

We specify several constraints required by grid-based mutual-capacitance and the hardware limitations.

C1: No intersections within the same layer. Curves on one layer are either Tx or Rx lines and must not intersect lest electrical cross-talk is introduced and ambiguity in sensing location arises. Thus, curves selected from the same layer must not intersect:

$$a_i \cap a_j = \emptyset \quad \forall i \neq j, \quad b_i \cap b_j = \emptyset \quad \forall i \neq j. \quad (7)$$

C2: At most one intersection between curves from different layers. Since multiple intersections between the same pair of curves lead to ambiguity in touch location, we enforce:

$$|a_i \cap b_j| \leq 1 \quad \forall a_i \in A, b_j \in B. \quad (8)$$

C3: Bounded number of selected curves. The scanning controller provides a limited number of Tx and Rx channels and the number of selected curves in each layer must not exceed these hardware-imposed limits. Let $x_i \in \{0, 1\}$ and $y_j \in \{0, 1\}$ denote binary decision variables indicating whether curve $a_i \in A$ or $b_j \in B$ is selected,

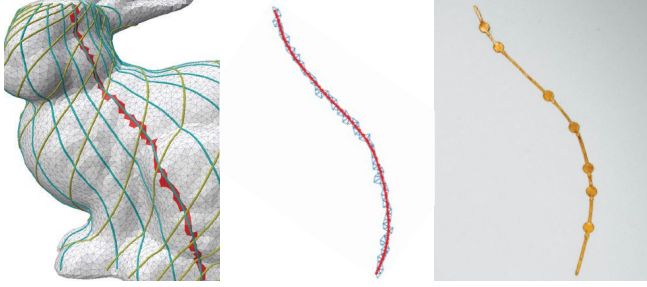


Fig. 5. Trace fabrication. Our method unfolds each selected curve (left), outlines its 2D shape and combines it with the placed electrode pads (center), and sends it to a vinyl cutter for fabrication (right). We use Kapton tape-covered adhesive copper tape to obtain insulated conductors in one go.

respectively. We impose:

$$\sum_{i=1}^m x_i \leq N_{\text{drive}}, \quad \sum_{j=1}^n y_j \leq N_{\text{sense}}. \quad (9)$$

Together, these constraints ensure that the resulting sensor layout is suitable for capacitance sensing on existing physical objects.

6.3 Phase 2: Selecting intersections

We now optimize where electrode pads are placed, taking the set of all intersections I' from the selected curve subsets A' and B' as input. Each intersection in I' is a candidate location for placing electrode pads. The objective is to maximize the number of selected intersections while ensuring reliable capacitive sensing. This requires that electrode pads be sufficiently separated to avoid interference. We thus impose a minimum distance constraint between any pair of selected intersections. Let $z_k \in \{0, 1\}$ be a binary decision variable indicating whether intersection $i_k \in I'$ is selected for electrode placement. We formulate the objective as:

$$\max \sum_{i_k \in I'} z_k, \quad (10)$$

subject to the constraint that any two selected intersections must be separated by at least a minimum distance $d_{\min}$:

$$\text{dist}_M(i_k, i_l) \geq d_{\min} \quad \forall i_k, i_l \in I', \quad k \neq l \text{ with } z_k = z_l = 1. \quad (11)$$

This constraint ensures that electrode pads are separated on the surface to reduce interference and improve sensing robustness.

Phase 2 outputs a subset of intersections $I'' \subseteq I'$ where electrode pads are rendered. Combined with the selected curves (Phase 1), this produces the final sensor layout for wire routing and pad placement, as shown in Figure 7.

7 Fabrication and attachment to the 3D object

Fig. 5 shows our fabrication pipeline. Starting with the selected 3D curves on the mesh, it produces cut traces of thin Kapton-copper-Kapton laminate for attachment to the real-world object.

7.1 Curve unfolding and outlining to planar traces

Our method converts the 3D geodesic curves to 2D representations as described by Mitani and Suzuki [2004] by unfolding triangle strips. Starting from the initial triangle that contains the trace, we isometrically construct each adjacent triangle onto a 2D plane. Since our geodesic segments are intrinsic, the unfolded traces can be reconstructed by connecting the corresponding curve nodes on

the edges of each triangle. Once unfolded, we derive an outline of each 2D representation with 1 mm width. At locations where curves intersect, we union a tiny circle as a visual landmark to aid later attachment. For the intersections selected in our optimization, our method additionally unions an electrode pad with a radius of 2 mm, offset by 3 mm into the routing direction. We also union a small pad to the end of each trace for soldering wires to a digitizer.

7.2 Electrode trace fabrication

We fabricate all sensor traces using a three-layer Kapton-copper-Kapton sandwich structure, cut with a commodity vinyl cutter (Cricut Maker 3). The laminate is flexible and thin (0.05 mm Kapton layer, 0.06 mm copper foil). Unrolled, outlined, and pad-augmented traces are densely placed on a sheet for collective cutting. The vinyl cutter has millimeter precision and supports rapid batch fabrication of all unfolded traces, cutting trace outlines in less than a minute. We transfer each cut trace onto the object using adhesive tape.

7.3 Attaching 2D trace outlines to the 3D object

Our approach guides precise manual attachment of the fabricated traces onto the object by projecting a textured 3D mesh onto the real-world object. For this, our system integrates an Azure Kinect RGB-D camera to detect and track the object and an XGIMI Elfin Flip projector. Our system segments the image using OVE6D [Cai et al. 2022] and estimates the object’s 3D rotation and translation relative to the camera based on the depth map with temporal smoothing to stabilize estimates and reduce jitter. We project the routed curves via 3D polylines offset along surface normals and mapped to UV coordinates using barycentric interpolation for rasterization. This step includes back-face culling to discard surfaces invisible to the projector. We preserve spatial accuracy during projection via perspective-correct interpolation of texture coordinates. The tracking updates in real-time, such that the 3D object can be moved and rotated during trace assembly. Before use, we calibrate both with a checkerboard and gray-code structured-light patterns to recover the intrinsic parameters of both devices and the rigid transformation between both coordinate frames.

The projection facilitates manual attachment of the cut traces simply by following the projected lines for the bottom layer. For attaching the traces of the second layer, we have found projection to be less helpful and instead rely on the small intersection landmarks we incorporated during unrolling and outlining (Fig. 5, right) that visually guide, point-by-point, how traces are to be attached.

Finally, we solder flexible wires to trace ends and route them to a mutual-capacitance scanning controller. Cable bundles are shielded to minimize stray capacitance and reduce sensing interference.

8 Sensing, signal processing, and visualization

Via a soldered cable, each trace connects to a pin of an embedded mutual-capacitance digitizer IC. Our sensing pipeline integrates a Muca board [Muca 2024], which breaks out an FT5316DME mutual-capacitance digitizer (Micros, PL), which has been used in several prior projects for resolving touch on fabricated surfaces (e.g. [Parilusyan et al. 2022; Teyssier et al. 2019]). The chip provides 21 Tx and 12 Rx lines and reports mutual-capacitance measurements M_{ij} at 10 Hz through an SPI interface to our embedded controller. This

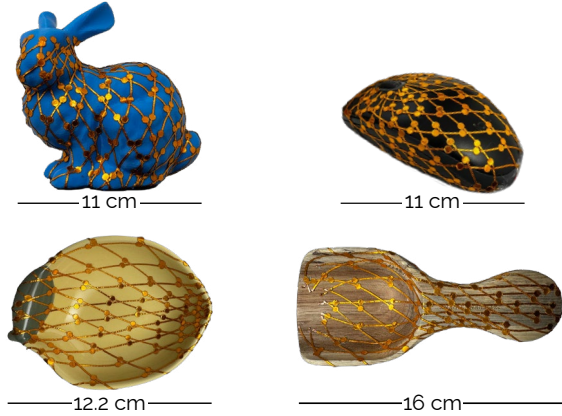


Figure 6. Four touch-sensitive demonstrators created by our method via object scanning, curve generation, intersection optimization, and fabrication.

pin count defined the constraint values in our optimization for the objects we instrumented. We stream these measurements to a PC via USB, where a simple processing app removes DC biases.

Mapping capacitive measurements to the 3D object

The digitizer provides a matrix of capacitive measurements, commonly visualized as a 2D image. Since our routing does not follow the traditional orthogonal grid approach, we reproject each measurement M_{ij} from the mutual-capacitance matrix back to the surface location where drive line $a_i \in A$ intersects with sense line $b_j \in B$ using the known intrinsic curve layout.

3D visualization and contact detection. Mapping each intersection $i_{(a_i, b_j)} \in I$ to the corresponding face on the input triangle mesh allows us to display the registered intensity change in mutual-capacitance as a heatmap over the object surface. This directly shows touch contacts on the 3D object with spatial correspondence to the electrode layout (Fig. 1, right).

Given the spatial resolution of electrodes, we perform the common interpolation of individual mutual-capacitance intensities to estimate contact areas [Streli and Holz 2021] when fingers or hands couple to more than one intersection. Interpolating the intensities yields the centroid of connected components on the mesh as a single input coordinate for the purpose of detecting input events [Holz and Baudisch 2010; Streli and Holz 2021]. This enables a rich interaction vocabulary similar to conventional touch panels, including multi-touch, swipes, pinches, and taps—all mapped onto 3D surfaces with real-time spatial correspondence. For example, users can slide along

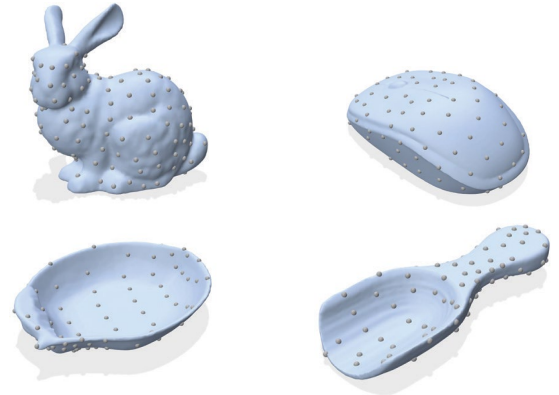


Figure 7. Curve-intersection locations, which later serve as sensing locations, generated by our two-phase optimization.

a curved surface for intuitive 3D editing, or tap specific regions of a figurine to trigger animations.

9 Evaluation

9.1 Touch sensing prototypes

We fabricated and tested our sensing pipeline on four physical objects with varied geometry and scale (Fig. 6).

9.2 Evaluating sensor layout optimization

To evaluate spatial uniformity, we measure the coefficient of variation (CV) of Voronoi cell areas induced by the optimized intersections, where lower CV indicates more uniform coverage. Table 1 shows that our method produces highly uniform sensor layouts across all objects. Fig. 8 visualizes our optimization results for ten physical objects. Fig. 9 shows our optimization results under different constraints on the maximum number of curves allowed per layer, which is a limit determined by the number of pins available on the hardware sensing board.

9.3 Capacitive sensing signal-to-noise analysis

We measure SNR from repeated mutual-capacitance recordings across 15 touch locations per object with 10 repetitions each under varying finger orientations. Table 2 summarizes the measured SNR values across all objects. For touch sensing, $\text{SNR} \geq 15$ is recommended and $\text{SNR} \geq 7$ is the minimum threshold for reliable performance [Davison 2010]. All our SNR measurements surpassed both thresholds by a wide margin.

Table 1. Uniformity metrics of the optimized intersections. Lower Voronoi cell coefficients of variation (CV) indicate more uniform coverage.

Object	Voronoi coefficient of variation (CV)
Stanford bunny	0.374
Mouse	0.303
Plate	0.594
Spoon	0.451

Table 2. Signal-to-noise ratio for touch contact.

Object	mean SNR	SNR stdev
Stanford bunny	64.68	32.68
Mouse	95.55	81.42
Plate	64.29	42.72
Spoon	52.51	51.04

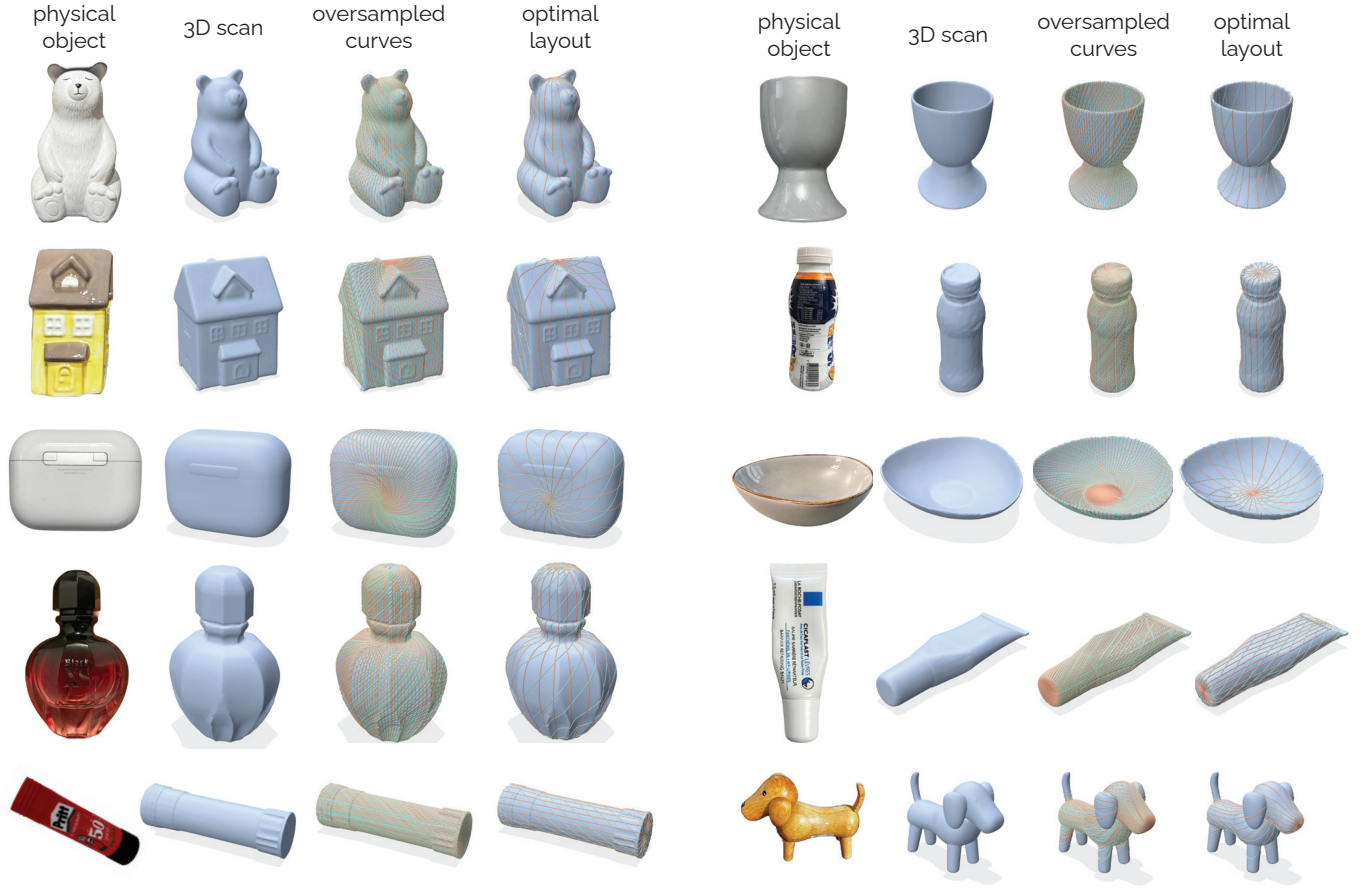


Fig. 8. We apply our pipeline to ten additional existing physical objects without fabrication. For each object, we present photographs of the physical object, its 3D scan, oversampled intrinsic curves, and the corresponding optimization results.

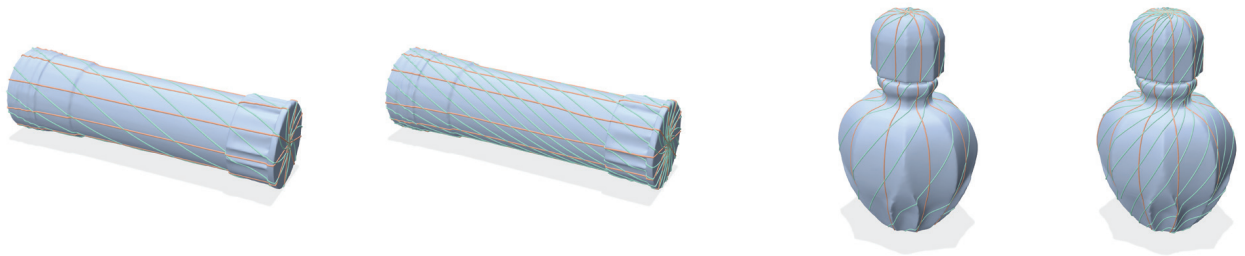


Fig. 9. Optimization results with different numbers of curves, demonstrating support for varying resolution requirements.

9.4 Localization accuracy

We evaluate localization accuracy using 15 surface locations with 10 repeated touches per location. Touch events are detected from raw sensor signals and reconstructed to 3D positions, with no observed false positives or negatives. Table 3 lists the localization errors.

9.5 Evaluating trace projection accuracy

We evaluate projection accuracy by comparing reprojected curve intersections with manually annotated ground truth on Azure Kinect captures. Pixel distances are converted to millimeters using known object dimensions (Table 4).

Table 3. 3D localization accuracy (mm) across fabricated objects.

Object	mean	min	max	mad	std	std ²	RMSE
Mouse	0.76	0.21	1.24	0.28	0.32	0.101	0.82
Plate	0.65	0.20	1.12	0.25	0.28	0.078	0.71
Stanford bunny	1.05	0.20	1.80	0.38	0.46	0.214	1.14
Spoon	1.60	0.14	3.39	1.03	1.10	1.221	1.95

10 Discussion

Across our fabricated prototypes, our pipeline generates spatially consistent sensing layouts on a range of geometries without requiring access to the object interior. Voronoi analysis shows that optimized intersections achieve uniform surface coverage across objects of varying scale and curvature. Higher variation for the plate is due to limited sensing area and boundary effects rather than routing failure. Overall, intrinsic curve generation and constrained optimization produce stable sensor layouts with predictable spatial resolution. The measured SNR values show that surface-only mutual-capacitance layouts enable robust touch sensing across diverse geometries. Mean SNR exceeds the recommended threshold for reliable detection and remains above the minimum usable level across all objects. Higher standard deviations are due to finger orientation, contact area, and multi-intersection touches rather than sensing failures. Overall, surface-intrinsic routing and projection-guided assembly preserve capacitive signal quality and support reliable multi-touch interaction. Our evaluation reveals that the system achieves sub-millimeter to ~ 2 mm localization accuracy across a range of object geometries, demonstrating that the projected electrode layouts support reliable spatial touch sensing on curved surfaces. Projection accuracy is sufficient for practical fabrication and supports our approach. Mean errors of a few millimeters are small relative to finger contact size and sensor spacing. This preserves the spatial correspondence assumed by the sensing model. Our setting is not directly comparable to Palma et al. [2024], which uses internal routing in 3D-printed objects, whereas we attach traces to existing surfaces. Mean errors (3–4 mm) are small relative to finger contact diameter (10 mm) and object scale (bunny: 1.77%, plate: 1.94%, spoon: 2.51%, mouse: 2.55% of bounding-box diagonal).

Design and fabrication tradeoffs. Our surface-only approach retrofits existing objects without modifying their volume, but the use of copper traces still occludes their appearance. Future work could explore hydrographic printing or conductive paints and assess their sensing characteristics. Regarding instrumentation overhead, soldering is a one-time step per object rather than per electrode. Our Muca-based approach requires only attaching trace ends to through-holes, simplifying assembly compared to prior work such as Palma et al. [2024], which relies on internal tube routing and global surface parameterization. Future work could further reduce overhead using alternative connectors such as snap-fit or pogo-pin solutions.

Sensor layout generality. Genus-0 topology covers the vast majority of practical retrofit targets, including household items, toys, figurines, and consumer products such as computer mice, bottles, and tools—making our current assumptions of genus-0 topology and a clear non-sensing base reasonable for the envisioned use cases.

Table 4. Projection accuracy onto 3D objects.

Object	mean distance [mm]	distance stdev
Stanford bunny	3.05	1.56
Mouse	3.29	1.98
Plate	2.99	1.61
Spoon	4.44	2.45

While we can accommodate many practical shapes, more complex topologies will require more singularities for the computation of the discrete connections and additional evaluation for consistency. Narrow protrusions, such as the ears of the bunny, may receive fewer traces because only a small range of initial directions at the seed point routes curves into these regions. Coverage may thus be limited and reseeding or added singularities could improve it.

Sensing performance. Beyond trace layout, capacitive sensing quality depends on aspects such as trace length, spacing, and shielding. Our approach optimizes spatial distribution of intersections, but we do not currently account for electrical impedance or parasitic capacitance introduced by long or tightly curved traces. Future work could integrate electrical models into the optimization to select configurations that balance spatial coverage with signal integrity.

11 Conclusion

We have introduced a generative computational fabrication approach for retrofitting existing 3D objects with surface-only capacitive touch sensing. Our method formulates sensor design as a constrained geometric routing problem: it generates surface-intrinsic drive and sense lines, optimizes their intersections for spatial coverage under mutual-capacitance, hardware, and fabrication constraints, and converts the resulting layouts into planar traces for projection-guided assembly. Our process thus preserves the object’s interior and structure while adapting the multi-touch sensor design to the object’s 3D geometry. Across the four real-world objects we augmented with our method, the generated sensors achieved reliable spatial sensing, with signal-to-noise ratios well above recommended thresholds and localization errors of ~ 2 mm or less. Our design studies further show that our method can produce layouts across a broader range of object shapes and sensing resolutions.

More broadly, our work shows that spatial touch sensing can be computationally designed as a retrofit for a given physical object, rather than requiring the object to be redesigned, internally instrumented, or reproduced around a predefined sensor layout. This is particularly relevant for functional, unique, or otherwise irreplaceable objects whose geometry and interior must remain intact. By adapting the sensing layout to the existing surface, our approach extends computational fabrication from creating interactive objects to augmenting objects that already exist.

Acknowledgments

We sincerely thank Marc Teyssier for providing us with several Muca Boards on short notice. Zhipeng Li was partially supported by the Swiss National Science Foundation (Grant No. 10004941).

Appendix

A Implementation details of the optimizer

Optimization parameters. We determine the minimum distance threshold $d_{\min}$ for electrode pad placement based on the physical size of the pad. We set $d_{\min}$ to twice the pad diameter to avoid electrical interference between neighboring electrodes. To approximate surface coverage in Phase 1, we uniformly sample 300 points across the surface. We evaluated an alternative Voronoi-based sampling strategy that produced identical optimization outcomes across all 14 evaluated objects, but it was substantially slower. Therefore, we use uniform sampling with distance-threshold rejection, which takes about 10 s. This provides a good trade-off between approximation accuracy and computational efficiency.

The maximum numbers of selectable curves in the two layers, N_{drive} and N_{sense} , are set according to the channel limitations of the scanning controller (Section 8). For the second layer, we run the full two-phase optimization independently for each sampled initial angle and select the angle that yields the best objective value across all runs. In all fabricated prototypes, the integer linear programs returned feasible solutions. Should infeasibility occur, we can relax constraints by reducing electrode pad size (thereby lowering $d_{\min}$) or by reducing the number of selected curves.

Optimization solver. We implement both optimization phases as integer linear programs and solve them using Gurobi [Gurobi 2026]. Runtime scales primarily with mesh resolution. Using our setup with Gurobi's default settings, optimization for meshes of approximately 20k vertices with 21 drive and 12 sense candidate curves takes ~20 min, while meshes around 50k vertices require more than 1 hour. In our tested range, varying pin count changed runtime only marginally compared to changing mesh size.

B Evaluation ablation: Effect of secondary objective

An ablation of Eq. 6 shows that removing this term increases the Voronoi cell area coefficient of variation from 0.3–0.5 to 0.4–0.7 across all four objects, indicating that Eq. 6 helps promote a more uniform distribution of intersections.

References

- S. Sandra Bae, Takanori Fujiwara, Danielle Szafir, Ellen Yi-Luen Do, and Michael L. Rivera. 2025. Computational Design and Single-Wire Sensing of 3D Printed Objects with Integrated Capacitive Touchpoints. In *Proceedings of the ACM Symposium on Computational Fabrication (SCF '25)*. ACM, 1–19. doi:10.1145/3745778.3766650
- Samarth Brahmabhatt, Chengcheng Tang, Christopher D. Twigg, Charles C. Kemp, and James Hays. 2020. ContactPose: A Dataset of Grasps with Object Contact and Hand Pose. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 361–378. doi:10.1007/978-3-030-58601-0_22
- Jesse Burstyn, Nicholas Fellion, Paul Strohmeier, and Roel Vertegaal. 2015. Printput: Resistive and capacitive input widgets for interactive 3D prints. In *Human-Computer Interaction – INTERACT 2015: 15th IFIP TC 13 International Conference, Bamberg, Germany, September 14–18, 2015, Proceedings, Part I* 15. Springer, 332–339.
- Dingding Cai, Janne Heikkilä, and Esa Rahtu. 2022. OVE6D: Object Viewpoint Encoding for Depth-based 6D Object Pose Estimation. arXiv:2203.01072 [cs.CV] <https://arxiv.org/abs/2203.01072>
- Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. 2021. Reconstructing Hand-Object Interactions in the Wild. 12417–12426.
- Paolo Cignoni, Nico Pietroni, Luigi Malomo, and Roberto Scopigno. 2014. Field-aligned mesh joinery. *ACM Trans. Graph.* 33, 1, Article 11 (Feb. 2014), 12 pages. doi:10.1145/2537852
- Keenan Crane, Mathieu Desbrun, and Peter Schröder. 2010. Trivial Connections on Discrete Surfaces. *Computer Graphics Forum (SGP)* 29, 5 (2010), 1525–1533.
- Burke Davison. 2010. Techniques for robust touch sensing design. *AN1334 Microchip Technology Inc* 53 (2010).
- Fernando de Goes, Mathieu Desbrun, and Yiyong Tong. 2016. Vector field processing on triangle meshes. In *ACM SIGGRAPH 2016 Courses (Anaheim, California) (SIGGRAPH '16)*. Association for Computing Machinery, New York, NY, USA, Article 27, 49 pages. doi:10.1145/2897826.2927303
- Zicong Fan, Maria Parelli, Maria Eleni Kadoglou, Xu Chen, Muhammed Kocabas, Michael J. Black, and Otmar Hilliges. 2024. HOLD: Category-Agnostic 3D Reconstruction of Interacting Hands and Objects from Video. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 494–504. doi:10.1109/cvpr52733.2024.00054
- Matthew Fisher, Peter Schröder, Mathieu Desbrun, and Hugues Hoppe. 2007. Design of tangent vector fields. *ACM Trans. Graph.* 26, 3 (July 2007), 56–es. doi:10.1145/1276377.1276447
- Timo Götzelmann and Christopher Althaus. 2016. TouchSurfaceModels: Capacitive Sensing Objects through 3D Printers. In *Proceedings of the 9th ACM International Conference on Pervasive Technologies Related to Assistive Environments (Corfu, Island, Greece) (PETRA '16)*. Association for Computing Machinery, New York, NY, USA, Article 22, 8 pages. doi:10.1145/2910674.2910690
- Daniel Groeger and Jürgen Steimle. 2018. ObjectSkin: augmenting everyday objects with hydroprinted touch sensors and displays. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 1–23.
- Tobias Grosse-Puppenthal, Christian Holz, Gabe Cohn, Raphael Wimmer, Oskar Bechtold, Steve Hodges, Matthew S. Reynolds, and Joshua R. Smith. 2017. Finding Common Ground: A Survey of Capacitive Sensing in Human-Computer Interaction. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3293–3315. doi:10.1145/3025453.3025808
- Gurobi. 2026. Gurobi Optimizer Reference Manual. <https://www.gurobi.com> Accessed: 2026-04-20.
- Christian Holz and Patrick Baudisch. 2010. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 581–590.
- Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. 2011. KinectFusion: Real-Time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. Association for Computing Machinery, New York, NY, USA, 559–568. doi:10.1145/2047196.2047270 event-place: Santa Barbara, California, USA.
- Yoshihiro Kawahara, Steve Hodges, Benjamin S Cook, Cheng Zhang, and Gregory D Abowd. 2013. Instant inkjet circuits: lab-based inkjet printing to support rapid prototyping of UbiComp devices. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. 363–372.
- Arshad Khan, Joan Sol Roo, Tobias Kraus, and Jürgen Steimle. 2019. Soft inkjet circuits: rapid multi-material fabrication of soft circuits using a commodity inkjet printer. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 341–354.
- Konstantin Klamka, Raimund Dachselt, and Jürgen Steimle. 2020. Rapid iron-on user interfaces: Hands-on fabrication of interactive textile prototypes. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.
- Beibei Liu, Yiyong Tong, Fernando De Goes, and Mathieu Desbrun. 2016. Discrete Connection and Covariant Derivative for Vector Field Analysis and Design. *ACM Trans. Graph.* 35, 3, Article 23 (March 2016), 17 pages. doi:10.1145/2870629
- Jun Mitani and Hiromasa Suzuki. 2004. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Trans. Graph.* 23 (08 2004), 259–263. doi:10.1145/1015706.1015711
- Muca. 2024. Muca Breakout: Hardware Interface for Multi-Touch Sensing. <https://muca.cc/shop/muca-breakout-6>. Accessed: 2026-04-20.
- William Neveu, Ivan Puhachov, Bernhard Thomaszewski, and Mikhail Bessmeltsev. 2022. Stability-Aware Simplification of Curve Networks. In *ACM SIGGRAPH 2022 Conference Proceedings (Vancouver, BC, Canada) (SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 20, 9 pages. doi:10.1145/3528233.3530711
- Aditya Shekhar Nittala, Anusha Withana, Narjes Pourjafarian, and Jürgen Steimle. 2018. Multi-Touch Skin: A Thin and Flexible Multi-Touch Sensor for On-Skin Input. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3173574.3173607
- Yuta Noma, Silvia Sellán, Nicholas Sharp, Karan Singh, and Alec Jacobson. 2024. Surface-Filling Curve Flows via Implicit Medial Axes. *ACM Trans. Graph.* 43, 4, Article 147 (July 2024), 12 pages. doi:10.1145/3658158
- Simon Olberding, Michael Wessely, and Jürgen Steimle. 2014. PrintScreen: fabricating highly customizable thin-film touch-displays. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 281–290.

- Gianpaolo Palma, Narges Pourjafarian, Jürgen Steimle, and Paolo Cignoni. 2024. Capacitive Touch Sensing on General 3D Surfaces. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–20.
- Brice Parilussyan, Marc Teyssier, Valentin Martinez-Missir, Clément Duhart, and Marcos Serrano. 2022. Sensurfaces: A Novel Approach for Embedded Touch Sensing on Everyday Surfaces. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 2 (July 2022), 67:1–67:19. doi:10.1145/3534616
- Patrick Parzer, Adwait Sharma, Anita Vogl, Jürgen Steimle, Alex Olwal, and Michael Haller. 2017. SmartSleeve: real-time sensing of surface and deformation gestures on flexible, interactive textiles, using a hybrid gesture detection pipeline. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. 565–577.
- Narjes Pourjafarian, Anusha Withana, Joseph A. Paradiso, and Jürgen Steimle. 2019. Multi-Touch Kit: A Do-It-Yourself Technique for Capacitive Multi-Touch Sensing Using a Commodity Microcontroller. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New Orleans, LA, USA, 1071–1083. doi:10.1145/3332165.3347895
- Anne Roudaut, Henning Pohl, and Patrick Baudisch. 2011. Touch input on curved surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Vancouver, BC, Canada) (CHI '11)*. Association for Computing Machinery, New York, NY, USA, 1011–1020. doi:10.1145/1978942.1979094
- Silvia Rus, Tobias Grosse-Puppenthal, and Arjan Kuijper. 2014. Recognition of Bed Postures Using Mutual Capacitance Sensing. In *Ambient Intelligence*, Emile Aarts, Boris de Ruyter, Panos Markopoulos, Evert van Loenen, Reiner Wichert, Ben Schouten, Jacques Terken, Rob Van Kranenburg, Elke Den Ouden, and Gregory O'Hare (Eds.). Springer International Publishing, Cham, 51–66. doi:10.1007/978-3-319-14112-1_5
- Rei Sakura and Yasuaki Kakehi. 2025. A 3D-Printed Touch Sensor with a Single-Stroke Conductive Path. 1–6. doi:10.1145/3706599.3720023
- Valkyrie Savage, Xiaohan Zhang, and Björn Hartmann. 2012. Midas: fabricating custom capacitive touch sensors to prototype interactive objects. In *Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12)*. Association for Computing Machinery, New York, NY, USA, 579–588. doi:10.1145/2380116.2380189
- Martin Schmitz, Mohammadreza Khalilbeigi, Matthias Balwierz, Roman Lissermann, Max Mühlhäuser, and Jürgen Steimle. 2015. Capricate: A Fabrication Pipeline to Design and 3D Print Capacitive Touch Sensors for Interactive Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. Association for Computing Machinery, New York, NY, USA, 253–258. doi:10.1145/2807442.2807503
- Christian Schüller, Roi Poranne, and Olga Sorkine-Hornung. 2018. Shape Representation by Zippables. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)* 37, 4 (2018).
- Paul Strelt and Christian Holz. 2021. CapContact: Super-resolution Contact Areas from Capacitive Touchscreens. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (Yokohama, Japan) (CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 289, 14 pages. doi:10.1145/3411764.3445621
- Paul Strelt, Mark Richardson, Fadi Botros, Shugao Ma, Robert Wang, and Christian Holz. 2024. TouchInsight: Uncertainty-aware Rapid Touch and Text Input for Mixed Reality from Egocentric Vision. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24)*. Association for Computing Machinery, New York, NY, USA, 1–16. doi:10.1145/3654777.3676330
- Paul Strohmeier, Jarrod Knibbe, Sebastian Boring, and Kasper Hornbæk. 2018. zPatch: Hybrid resistive/capacitive textile input. In *Proceedings of the twelfth international conference on tangible, embedded, and embodied interaction*. 188–198.
- Mathias Sundholm, Jingyuan Cheng, Bo Zhou, Akash Sethi, and Paul Lukowicz. 2014. Smart-mat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*. 373–382.
- Marc Teyssier, Gilles Bailly, Catherine Pelachaud, Eric Lecolinet, Andrew Conn, and Anne Roudaut. 2019. Skin-on interfaces: A bio-driven approach for artificial skin design to cover interactive devices. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 307–322.
- Amir Vaxman, Marcel Campen, Olga Diamanti, David Bommes, Klaus Hildebrandt, Mirela Ben-Chen Technion, and Daniele Panizzo. 2017. Directional field synthesis, design, and processing. In *ACM SIGGRAPH 2017 Courses (Los Angeles, California) (SIGGRAPH '17)*. Association for Computing Machinery, New York, NY, USA, Article 12, 30 pages. doi:10.1145/3084873.3084921
- Nicolas Villar, Daniel Cletheroe, Greg Saul, Christian Holz, Tim Regan, Oscar Salandin, Misha Sra, Hui-Shyong Yeo, William Field, and Haiyan Zhang. 2018. Project Zanzibar: A portable and flexible tangible interaction platform. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.
- Nicolas Villar, Shahram Izadi, Dan Rosenfeld, Hrvoje Benko, John Helmes, Jonathan Westhues, Steve Hodges, Eyal Ofek, Alex Butler, Xiang Cao, and Billy Chen. 2009. Mouse 2.0: multi-touch meets the mouse. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (Victoria, BC, Canada) (UIST '09)*. Association for Computing Machinery, New York, NY, USA, 33–42. doi:10.1145/1622176.1622184
- Michael Wessely, Ticha Sethapakdi, Carlos Castillo, Jackson C Snowden, Ollie Hanton, Isabel PS Qamar, Mike Fraser, Anne Roudaut, and Stefanie Mueller. 2020. Sprayable user interfaces: Prototyping large-scale interactive surfaces with sensors and displays. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–12.
- 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*. 589–598.
- Fu Xiao-Ming, Jian-Ping Su, Zheng-Yu Zhao, Qing Fang, Chunyang Ye, and Ligang Liu. 2021. Inversion-free geometric mapping construction: A survey. *Computational Visual Media* 7 (09 2021), 289–318. doi:10.1007/s41095-021-0233-9
- Junsheng Yao, Lichao Mou, and Qingyu Li. 2025. SAM 3D for 3D Object Reconstruction from Remote Sensing Images. arXiv:2512.22452 [cs.CV] <https://arxiv.org/abs/2512.22452>
- Chris Yu, Henrik Schumacher, and Keenan Crane. 2021. Repulsive Curves. *ACM Trans. Graph.* 40, 2, Article 10 (May 2021), 21 pages. doi:10.1145/3439429
- Yang Zhang and Chris Harrison. 2018. Pulp nonfiction: Low-cost touch tracking for paper. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–11.
- Yang Zhang, Gierad Laput, and Chris Harrison. 2017. Electrick: Low-cost touch sensing using electric field tomography. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. 1–14.
- Yang Zhang, Michel Pahud, Christian Holz, Haijun Xia, Gierad Laput, Michael McGuffin, Xiao Tu, Andrew Mittereder, Fei Su, William Buxton, and Ken Hinckley. 2019. Sensing Posture-Aware Pen+Touch Interaction on Tablets. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (Glasgow, Scotland Uk) (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3290605.3300285
- Yang Zhang, Chouchang (Jack) Yang, Scott E. Hudson, Chris Harrison, and Alanson Sample. 2018. Wall+: Room-Scale Interactive and Context-Aware Sensing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–15. doi:10.1145/3173574.3173847
- Thomas G. Zimmerman, Joshua R. Smith, Joseph A. Paradiso, David Allport, and Neil Gershenfeld. 1995. Applying electric field sensing to human-computer interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*. ACM Press/Addison-Wesley Publishing Co., USA, 280–287. doi:10.1145/223904.223940