DefSense: Computational Design of Customized Deformable Input Devices

Moritz Bächer[1], **Benjamin Hepp**[2†], **Fabrizio Pece**[2†]
Paul G. Kry[3], **Bernd Bickel**[4], **Bernhard Thomaszewski**[1], **Otmar Hilliges**[2]
[1]Disney Research, [2]ETH Zurich, [3]McGill University, [4]IST Austria

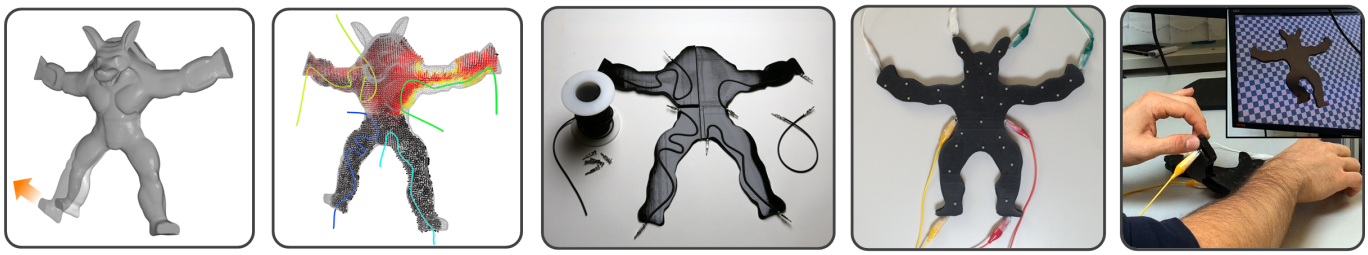

Figure 1. Design and fabrication of custom input devices (from *left* to *right*): the designer creates a set of example deformations (*1*) and roughly indicates where to place internal sensors. Our optimization algorithm then refines the sensor placement to maximize reconstruction accuracy (*2*). We fabricate our designs by inserting piezoresistive wires in-between 3D-printed body parts (*3*), then calibrate using motion capture (*4*). Our customized flexible input devices can be used in a variety of applications, e.g., to animate digital characters (*5*).

ABSTRACT

We present a novel optimization-based algorithm for the design and fabrication of customized, deformable input devices, capable of continuously sensing their deformation. We propose to embed piezoresistive sensing elements into flexible 3D printed objects. These sensing elements are then utilized to recover rich and natural user interactions at runtime. Designing such objects manually is a challenging and hard problem for all but the simplest geometries and deformations. Our method simultaneously optimizes the internal routing of the sensing elements and computes a mapping from low-level sensor readings to user-specified outputs in order to minimize reconstruction error. We demonstrate the power and flexibility of the approach by designing and fabricating a set of flexible input devices. Our results indicate that the optimization-based design greatly outperforms manual routings in terms of reconstruction accuracy and thus interaction fidelity.

Author Keywords

Input devices; fabrication; flexible; sensors; reconstruction

ACM Classification Keywords

H.5.2 User Interfaces: Input devices and strategies;

[†]These authors contributed equally to this work.

CHI 2016, May 7–12, 2016, San Jose, California, USA.

ACM ISBN 978-1-4503-3362-7/16/05 ...$15.00.
DOI: http://dx.doi.org/10.1145/2858036.2858354

INTRODUCTION

Consumer-level 3D-printers are becoming increasingly powerful and sophisticated, enabling end-users to create tangible artifacts with custom shape, appearance, and even functionality. In particular, multi-material printers can now fabricate objects with spatially-varying mechanical properties that deform in desired ways. The ambition of this work is to enable such objects to *sense* their deformation, thereby opening the door to a range of exciting applications such as personalized interactive toys, custom game controllers, and other flexible input devices. As the enabling technology, we focus on sensors made from *piezoresistive* materials that change their resistivity as a function of deformation. By embedding channels of such material into a 3D-printed object, we can infer its deformation based on the measured resistivity.

Novel input devices are of central interest to HCI and flexible input devices in particular hold the promise of making interactions more natural [3, 7, 24, 33]. However, existing devices are specialized, two-dimensional designs that were created manually by experts with extensive domain knowledge on both hardware and algorithmic levels. By contrast, we envision a design tool that allows *non-expert* users to create custom-shaped, flexible objects that can sense rich, continuous and user-specified deformations without requiring deep domain knowledge.

The goal of usable custom-designed input devices in this context critically depends on: (*a*) the richness of interactions that it supports, (*b*) how well user inputs can be recognized, and (*c*) the knowledge required to design such devices. As we show in this work, recognition accuracy strongly depends on the routing of the internal sensors. Even for moderately complex shapes, simple interactions lead to inhomogeneous de-

formation fields inside the object. The questions of how to route sensors and interpret their readings thus becomes, for any but the most trivial cases, a complex and difficult design problem. Furthermore, to afford rich and natural interactions, we do not want to constrain the allowed deformations to be mutually orthogonal. In consequence, there will generally not be a simple one-to-one mapping between sensor reading and deformation. Hence, without significant guidance, it is difficult for non-experts to design effective sensor layouts.

To address these challenges, we propose a novel optimization algorithm that computes sensor layouts based on user specified example deformations. Furthermore, we provide rich visual feedback to allow for an iterative, user-in-the-loop approach to custom controller design and fast exploration of the solution space. The key idea in our algorithm is to use reconstruction error as objective for the optimization — thus minimizing the quantity of greatest interest to the user. Relying on finite element simulations, we relate internal deformation to change in sensor response and simultaneously compute routings of conductive wires and a mapping between sensor values and 3D deformation. After fabrication, we leverage a machine-learning algorithm to robustly recover complex 3D deformations in real time, despite the presence of material imperfections and modeling errors.

We demonstrate our approach by designing and fabricating a set of flexible input devices, each adapted to a specific application. We furthermore show that a manual approach to sensor network design is not only a difficult and exhaustive exercise, but is also easily outperformed by our computational design approach. Experimental results yield significant reductions in reconstruction error between best-guess manual routings and our optimized solutions.

RELATED WORK

Fueled by innovations in sensing and additive manufacturing technologies, customized input devices have garnered a lot of attention recently. We concentrate on input devices that are self-sensing, i.e., we omit approaches relying on external sensors such as depth-cameras or motion capture systems; for a survey of externally sensed input we refer to Rendl et al. [24] and Steimle et al. [33].

2D rapid prototyping for IO devices

The Gummi system [27] simulated a hand-held, flexible display via two resistive pressure sensors. It demonstrated the rich interactive possibilities afforded by flexible input devices and inspired several related projects. With the introduction of 2D rapid prototyping techniques capable of printing basic electronic components, fully customized IO devices, such as capacitive buttons [38], transparent touch and pressure sensors [23], ink-jet printable touch pads [7], and even printed displays [18] have started to emerge. Still relying on 2D sensing elements several attempts exist to extend the input space to 2.5D. Early work used the ShapeTape sensor [6] for input into a 3D modeling application [3]. Kato et al. [14] replace the use of ShapeTape with piezoelectric thin films. More recently Rendl et al. [24] leveraged ink-jet printed piezoelectric sensors to reconstruct deformations of a transparent screen overlay with high accuracy and in real-time. Due to the manufacturing process, the above approaches are limited to 2D surfaces only. Furthermore the design of electronics and choice of sensor placement remains a manual and difficult-to-master process. Where prior work concentrated primarily on input sensing (with fixed hardware), our main contribution is an interactive method to aid novice users in designing sensor networks for custom controllers.

3D printed IO devices

3D printing has great potential for creating interactive objects. For example, printing tangibles that recognize input via acoustic sensing [25], or embedding optical elements into 3D objects to sense input [37] or to illuminate curved surfaces [5] were proposed. Moreover, flexible devices that sense topology changes via embedded contact switches have been explored [29]. Vázquez et al. [36] propose pneumatically activated 3D printed devices and Peng et al. [19] the printing of soft, interactive objects via fabric layering. Jacobson et al. [10] propose a modular, tangible input device for character animation based on Hall sensors and 3D printed parts. While providing high accuracy, the device requires an elaborate assembly process and is highly specific to 3D animation.

Recently, hybrid manufacturing systems that integrate stereolithography and direct print technologies have been proposed, enabling rapid prototyping and fabrication of 3D structures with embedded electronic circuits [11]. Leigh et al. [15] developed a conductive composite material for 3D printing. Although requiring specialized printers, these technologies demonstrate that the capability to 3D print electronically functional parts is within reach. Inspired by this vision, we explore the fully integrated design of flexible custom input devices, capable of reconstructing rich and continuous deformations.

Fabrication oriented computational design

The graphics community has recently begun to embrace the challenge of creating 3D-printable content. Among the various fabrication-oriented design tools that have been proposed so far, most relevant to our setting are methods for designing physical artifacts with controllable deformation behavior [4, 28]. Our method complements these works by augmenting the designs with embedded sensor networks, thereby enabling deformable objects to sense their state of deformation.

The problem of embedding sensing elements directly into 3D printed objects has been investigated [5, 29, 36, 37]. The design of such devices is notoriously hard and beyond the capabilities of average users — no automated solution has been proposed so far. While the related problem of automatically routing optical fibers [20] or general purpose channels [26] through 3D-printable objects has been explored, these works focus on optimizing primarily geometric properties such as channel curvature or inter-channel distance. Our setting is different in the sense that we have to simultaneously consider sensor deformations induced by multiple deformation fields, and account for the complex mapping between sensor readings and reconstructed deformation. This difference in goals requires different optimization strategies.

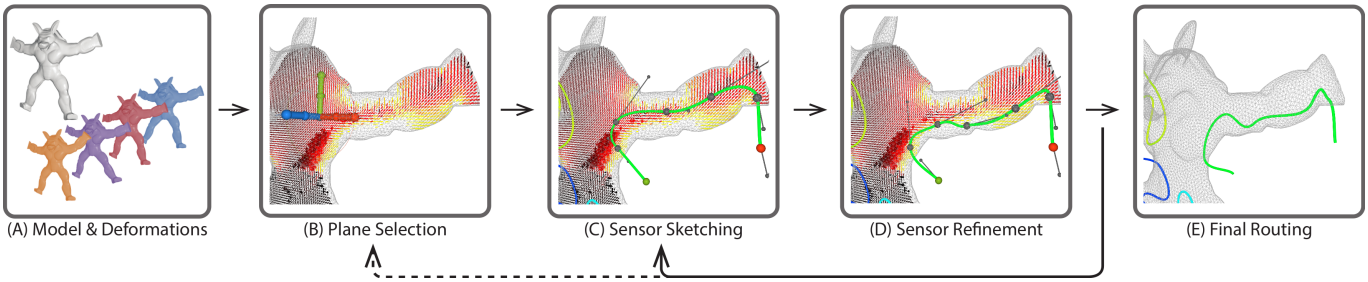

Figure 2. Design process overview (from left to right): (A) User specifies example deformations. (B) Planar cross section of internal deformation field. (C) Initial routing of sensor elements. (D) Optimization algorithm refines sensor configuration interactively. (E) Final routing ready for fabrication.

Computational UI design

UI design has traditionally been dominated by an expert driven design process. Recently, optimization techniques for UI design have begun to emerge, including methods to generate single-pointer touch keyboards [13, 30], menu layouts [1], and in-air gestures [32]. While related in spirit, these approaches have an almost orthogonal set of goals and constraints to ours. Previous work has attempted to optimize interaction techniques or UI layouts given an existing input device — the limitations of the sensor are a limiting factor in this setup. In our approach, we optimize the sensor configuration with respect to a set of desired interactions.

COMPUTATIONAL SENSOR DESIGN

The technical core of our work is an optimization-based design tool that allows non-expert users to quickly create flexible input devices with custom interaction spaces that can sense deformations on continuous scales. The underlying design problem is highly challenging, exhibiting a mix of discrete and continuous components as well as many local minima. Instead of seeking a fully-automated solution, we therefore propose a user-guided, optimization-in-the-loop approach that can leverage the intuition and experience of human designers to guide the optimization algorithm. In the following, we briefly summarize the design process, then proceed to the formal description of our method.

Overview

Using a physics-based deformation metaphor, the user starts by specifying a set of example deformations to define the interaction space (Fig. 2 A). We assume that, together with the undeformed state, these examples define a continuous space of expected user input, i.e., the space in which the device should be able to recognize its state of deformation. For each example pose, we compute an internal deformation field using finite element simulation. These deformation fields are used for automated sensor refinement and for providing visual feedback to the user during the design.

In the next step, the user incrementally adds deformation sensors, i.e., curved channels of piezoresistive material. In order to simplify fabrication and assembly, we layout individual sensors as planar curves, but allow the corresponding planes to be different for each sensor. The user can adjust the position and orientation of the planar crosscut. For guidance the user is shown a visualization of the deformation of the elements intersected by the current crosscut (Fig. 2 B). While the precise location and orientation is not critical, we obtained best results when ensuring that, for each sensor, the corresponding plane passes through at least one region that exhibits large deformation and displays variations in deformation across the different example poses. After choosing the sensor plane, the user defines an initial guess for the sensor path using a spline tool (Fig. 2 C). Except for the position of inlets and outlets, the exact paths of the sensors are not critical at this point. However, running the sensing element coarsely through regions of high deformation, and preferably high variance in deformation, helps the automatic refinement step to find a good solution. Once sensors are laid out, our optimization algorithm *refines* their paths by minimizing reconstruction error on the user-specified output (Fig. 2 D). The result of the optimization is displayed to the user by visualizing reconstruction error per example deformation. The user can then make an informed decision on whether to rearrange existing sensors, add additional sensors, or to finalize the design (Fig. 2 E).

Complemented by interactive visualization and editing tools, the pivotal element of our approach is a highly efficient sensor refinement algorithm and is described below.

Optimization-based Sensor Configuration

Our optimization algorithm builds on two basic principles. First, the sensors assume the deformation induced by their surrounding (3D-printed) material, and the sensor responses, i.e., their conductivity, changes as a function of their total deformation. With the user-defined deformations given as input, we can change the paths of the sensors in order to change their deformation and, thus, their response. Second, if we know the mapping from input (sensor response) to output (reconstructed geometry) signals, we can compute the paths of the sensors in order to minimize the discrepancy between reconstructed deformation and target deformation. While the mapping between input and output signals is not known *a priori*, we can optimize for it if we assume that the relation is affine; in our experiments, affine mappings have lead to good reconstruction accuracy. We explain the central components of this strategy in the following, starting with the computation of sensor deformations.

Computing Sensor Deformations

The deformation between the undeformed state of the model, described by positions $\mathbf{x}$, and its deformed state, described by positions $\mathbf{x}'$, is characterized at any point by the deformation

gradient $\mathbf{F} = \frac{\partial \mathbf{x}'}{\partial \mathbf{x}}$. In practice, we compute the deformation gradient in the discrete setting, using a standard finite element approach based on linear tetrahedral elements.

For simplicity, we start by considering a single sensor represented by a curve C, defined in the undeformed state of the model. At a given point $\mathbf{x}(\gamma)$ along the curve, the deformation gradient of the surrounding material $\mathbf{F}(\mathbf{x})$ transforms the curve's normalized tangent vector $\mathbf{t}(\gamma)$ to its deformed counterpart $\mathbf{t}'(\gamma) = \mathbf{F}(\mathbf{x})\mathbf{t}(\gamma)$. Normalizing the induced change in length $\sqrt{\mathbf{t}'^T\mathbf{t}'} - \sqrt{\mathbf{t}^T\mathbf{t}}$ gives rise to the Cauchy strain

$$\varepsilon(\gamma) = \sqrt{\mathbf{t}(\gamma)^T\mathbf{F}(\mathbf{x})^T\mathbf{F}(\mathbf{x})\mathbf{t}(\gamma)} - 1 \; . \tag{1}$$

In order to compute the total deformation of the sensor, we integrate the Cauchy strain along C,

$$\Delta l = \int_C \varepsilon(\gamma) \; d\gamma \; .$$

Note that a positive value for Δl is obtained if stretch outweighs compression in total, and vice versa.

Sensor Response

Piezoresistive materials generally exhibit a deformation-dependent resistivity, and often this relation is close to linear even for large deformations. As can be seen in Fig. 3, our experiments with pre-fabricated wire consistently exhibited good linearity across a wide range of stretching and compression. Hence, we model the change in resistivity as a linear function $w(\varepsilon) = k_{\text{res}} \cdot \varepsilon$, where the coefficient k_{res} is obtained by a least-squares fit to experimental data (see Sec. on Materials, Fabrication and Calibration for details). The total signal change for a deformed sensor then follows as

$$s(C) = \int_C w(\varepsilon) \; = \int_C k_{\text{res}} \cdot \left(\sqrt{\mathbf{t}^T\mathbf{F}^T\mathbf{F}\mathbf{t}} - 1 \right) \; . \tag{2}$$

To discretize the integral, we sample the curve with K samples $\mathbf{x}_k$

$$s_C = \sum_{k=1}^{K-1} k_{\text{res}} \cdot \left(\sqrt{\mathbf{d}_k^T\mathbf{F}(\mathbf{m}_k)^T\mathbf{F}(\mathbf{m}_k)\mathbf{d}_k} - 1 \right) \; , \tag{3}$$

where $\mathbf{d}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ are the difference vectors and $\mathbf{m}_k = \frac{1}{2}(\mathbf{x}_k + \mathbf{x}_{k+1})$ the midpoints.

Multiple Sensors and Spline-based Curve Representation

In practice, the user specifies multiple sensor curves C_j. To facilitate their intuitive and easy editing, we model each sensor curve C_j as a piecewise cubic Hermite spline. These splines depend on control points and tangent vectors, which we collect in a parameter vector $\mathbf{p}_j$. To allow for precise placement of inlets and outlets, we leave the last two control points and last tangent vector at either end of each sensor curve unparametrized. In addition to evaluating the position of sample points $\mathbf{x}_k$ in Eq. (3), sensor placement optimization also requires the derivatives of position with respect to the spline parameters $\mathbf{p}_j$, which are readily derived analytically.

Sensor Optimization

The goal of sensor optimization is to find parameter values for sensor placement and a mapping between input (sensor response) and output (deformed geometry) signals, minimizing reconstruction error. Our experiments showed that an affine mapping obtained through ridge regression yields sufficiently small reconstruction errors; see also [24] for similar findings. We formalize this approach below.

We represent the output geometry through O output coordinates that correspond to a set of marker points. For each output coordinate o, let $\mathbf{o}^o$ denote the vector that collects the deformed positions for the N examples. Furthermore, for each deformation-sensor pair (i, j), let s_{ij} denote the measured sensor signal. By collecting all of these M sensor signals in a $M + 1$-row vector

$$\mathbf{s}_i = \left[s_{i1} \ldots, s_{ij}, \ldots, s_{iM}, 1 \right] \; , \tag{4}$$

we can define an affine mapping between input and output signals,

$$\mathbf{s}_i\boldsymbol{\beta}^o = o_i^o. \tag{5}$$

If we further summarize all signals in an $N \times (M + 1)$ matrix $\mathbf{S}$, we obtain the linear system

$$\mathbf{S}\boldsymbol{\beta}^o = \mathbf{o}^o \; . \tag{6}$$

Our goal is now to minimize the error between the affinely mapped inputs and the desired outputs $\mathbf{o}^o$. To this end, we define a corresponding objective function

$$f(\mathbf{p}, \boldsymbol{\beta}) = \sum_o r^o \quad \text{with} \quad r^o = \|\mathbf{S}(\mathbf{p})\boldsymbol{\beta}^o - \mathbf{o}^o\|^2 + \|\lambda\hat{\mathbf{I}}\boldsymbol{\beta}^o\|^2 \; , \tag{7}$$

where $\hat{\mathbf{I}}$ is the identity matrix with a zero for entry $(M+1, M+1)$, leading to a regularizer (controlled by λ) that only penalizes linear coefficient terms.

Numerical Solution

In order to minimize Eq. (7), we employ a standard Newton-type solver, requiring that the gradient of the objective vanish with respect to all variables, i.e., both the sensor parameters $\mathbf{p}$ and the entries of the linear mapping $\boldsymbol{\beta}$. While the mapping parameters are regularized by construction, we also explicitly regularize the sensor parameters by penalizing changes in total length per sensor. The solution then yields optimized sensor routings as well as an output mapping, which provides feedback to the user about the expected reconstruction error per deformation.

RECOVERING USER INPUT

After fabrication (which we detail below) we wish to recover user input in real-time. Solving Eq. (7) for $\boldsymbol{\beta}$ will provide the desired mapping from resistance changes to outputs. However, during optimization, the signal matrix $\mathbf{S}$ is synthesized using an approximate model of the sensor response on a simulated interaction space. This clearly is an abstraction of the real-world: imprecision of our two-step manufacturing and discrepanices between simulated and physical interactions attenuate the signal. The dynamics of user interaction are also unknown during the design phase. Finally, for many cases there will be more output coordinates o than sensor readings. In order to obtain good sensing fidelity despite these challenges, we employ a simple data-driven technique to learn the input-output mapping. This step only needs to be performed once per fabricated sensor. Note that we use the same

mapping as in Eq. (7) while keeping the sensor curves fixed. Thus, both learning and sensor refinement algorithms minimize the same objective and as such are directly linked with one another.

For learning, we first acquire a training set of T signal vectors $\mathbf{s}_t$ together with O coordinates of a set of $O/3$ maker positions on the fabricated models. For acquisition, we use a standard motion capture system. Note that the sensor response is invariant to global rigid body motion and hence global translation and rotation need to be removed from motion capture data. We use Kabsch's algorithm [12] to pair marker coordinates with their correspondences in the rest pose.

By setting the gradients of the residuals $\frac{\partial r_o}{\partial \boldsymbol{\beta}^o}$ to zero, we learn an optimal mapping that minimizes the objective of Eq. (7),

$$(\mathbf{S}^T\mathbf{S} + \lambda^2\hat{\mathbf{I}})\boldsymbol{\beta}^o = \mathbf{S}^T\mathbf{o}^o \,, \tag{8}$$

where matrix $\mathbf{S}$ and the output vectors $\mathbf{o}^o$ now have T rows instead. To prevent overfitting, we use k-fold cross validation (in our case $k = 5$) to identify the optimal λ.

At runtime, 3D marker positions can then be recovered from resistances alone (i.e., not requiring the external capture system) by evaluating Eq. (5). To approximate the deforming surface we use the As-Rigid-As-Possible (ARAP) surface modeling technique by Sorkine et al. [31], with marker coordinates as position constraints.

While this linear regression scheme is simple, it has proven to be sufficiently powerful in our experiments. Similar findings were reported in [24]. The simplicity has further benefits including small model size and low computational cost for both training and evaluation. However, we would like to highlight the generality of using reconstruction error as objective during optimization. This idea can be applied to different, more powerful machine-learning models that can be formulated as regression problems of the form $\mathcal{L}\|\circ\| + \mathcal{R}\|\circ\|$, where $\mathcal{L}$ is a loss function and $\mathcal{R}$ a regularization term.

MATERIALS, FABRICATION AND CALIBRATION

In addition to the placement of the sensors and the mapping between sensor readings and deformation, a third factor that strongly influences recognition accuracy are the materials and the process used for fabrication. We performed an extensive set of experiments with different materials and fabrication processes which we detail in the Appendix.

3D-Printed Conductive Channels

We found that current multi-material 3D printers and available filaments cannot provide the level of quality and reliability required for our application. In printable piezoresistive polymer filaments, the relation between conductivity and deformation exhibits strong time-dependence and hysteresis. Furthermore, these polymers display high fluctuation between different samples and test runs (see Fig. 10). Plastic-based filaments show good conductivity behavior but they are brittle and will break when pushed beyond small deformations. Commercially-available piezoresistive polymers

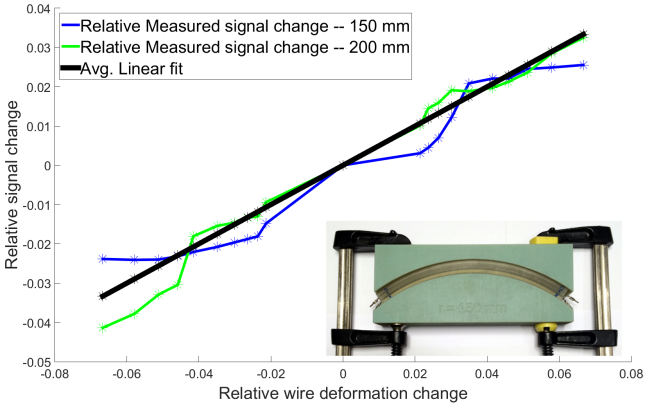

Figure 3. Fitted signal response function $w(\varepsilon)$ (black), acquired from specimens of length 150 and 200 mm. Inset: Calibration of specimen $\bar{l} = 200$ mm and $\bar{r} = 150$ mm.

with good conductivity and deformation properties are available (cf. Melnykowycz et al. [16]) but cannot be used for 3D printing yet.

Fabrication Process

In the light of these findings, we resort to a two-stage fabrication process, in which we first 3D-print deformable parts endowed with groves to host conductive material that we manually insert and secure with adhesives in a second step. Due to this fabrication process, complex spatial sensor paths become difficult to fabricate. In order to make this process tractable, we restrict our algorithm to generate planar routings. It should be noted, however, that planar sensor layouts do not not restrict interaction spaces to planar deformations as illustrated by our results. Furthermore, our formulation is general. As soon as 3D-printable piezoresistive materials of good quality are available, our method can be used to create 3D sensor routings with little modification.

Calibration

Optimizing for sensor layouts requires an accurate model for predicting the changes in resistivity as function of deformation. In order to obtain such models, we have developed a simple yet effective calibration method, applicable to our materials and others. The resulting calibration can then be used in our method without requiring any changes other than replacing the function $w(\varepsilon)$ in Eq. (2).

To calibrate the material, we 3D-printed a special-purpose specimen, which is seen inset at the bottom of Fig. 3. This test piece has three channels for embedding piezoresistive wire: one in the neutral plane, and two offset by margin $\bar{o}$ from the top and bottom boundary of the parts. In addition, we print rigid parts of various radii of curvature $\bar{r}$ (Fig. 3, inset). This part is then used to precisely bend the flexible calibration test piece, allowing us to measure and normalize the change in resistivity. The relative length change of the stretched and compressed wires can then be computed from the radius of curvature and the dimensions of the specimen, leading to two sample points for the relative signal responses. The resistance of the wire in the neutral plane serves as a sanity check as it is supposed to remain constant during bending. We performed

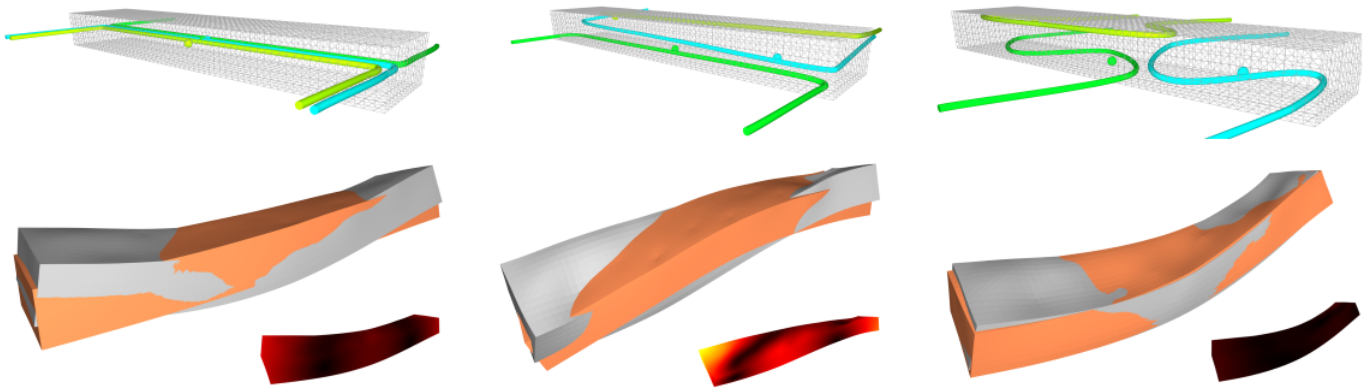

Figure 4. Comparing routing strategies. Top row (from left to right): schematic sensor routings obtained using best-guess manual placement without assistance from our tool (*manual*), our tool for feedback on deformations and predicted reconstruction error (*visual*), and our tool with automatic sensor refinement (*full*). Bottom row: Ground truth (gray) vs. reconstruction (orange) for each routing strategy. Insets show error on a heat map scale, with maximum error (white) at 22 mm (darker is better). The manual (left) routing exhibits large reconstruction error already for the simpler bend deformation. Our method with sensor refinement (right) leads to low error even for the challenging twist example, showing a clear improvement over the routings obtained using our design tool without refinement (middle).

this procedure for parts of length $\bar{l} = 150$ mm and $\bar{l} = 200$ mm, of thickness 15 mm, and offset $\bar{o} = 2$ mm. We used curvature radii 75, 87.5, 100, 112, 150, 175, 200, 225, and 250 mm. Fig. 3 shows the final averaged fitted response function, along with the relative signal responses for each specimen.

RESULTS

We have evaluated the proposed sensor design method and our overall pipeline in a number of experiments designed to illustrate our main contributions: an interactive design tool, leveraging a visualization of internal deformation fields to aid users in the sensor placement task, and an iterative refinement algorithm for optimizing the placement of the sensors.

Quantitative Evaluation

To assess the impact on reconstruction error, we fabricated three variants of a simple, yet instructive 3D object — an elongated box shape — and tested these variants with three challenging deformations: bending the object up and down, and twisting the object along its longest axis. We compare three routing strategies (cf. Fig. 4):

- ***manual***: best-guess routing created without assistance from our tool which, in this case, is to route straight wires.
- ***visual***: leverages visualization of internal deformations and predictions of reconstruction error to manually select the best planar cross section for each of the wires and corresponding example deformations.
- ***full***: using our tool with several iterations of automatic sensor refinement and sensor plane adjustments.

In order to perform a quantitative comparison between these routing strategies, we acquire test sequences of equal length using a marker-based ground-truth acquisition system and compute the difference between measured and predicted marker positions in 3D. For each design, we used a 70-30 training/test split approach, with the split performed in temporal order. Since the deformations are performed by a user, there is variation in the exact execution. However, the length of the test sequences of 2040 frames allows us to compare the recognition accuracy based on the collected data. Table 1 summarizes the reconstruction accuracy in terms of root-mean-square error (RMSE), overall maximum error (peak), and average of peak errors over the entire sequence (avg. peak).

	RMSE	peak	avg. peak
manual	4.46 ±1.76 mm	21.78 mm	8.98 mm
visual	3.66 ±1.33 mm	17.13 mm	7.12 mm
full	1.17 ±0.71 mm	10.13 mm	2.16 mm

Table 1. Reconstruction error in millimeters.

In our experiments, the ***visual*** approach consistently outperformed the ***manual*** counterpart, but our ***full*** pipeline performs best on all measures. While all three designs can reconstruct bending reasonably well, ***manual*** struggles the most in differentiating up and down directions (explained by the fact that all sensors are co-planar) and has the highest error on twisting. The ***visual*** routing tracks bending in both directions fairly well, as does ***full***. However, ***visual*** performs significantly worse than ***full*** on twisting. A one-way ANOVA revealed a statistically significant main effect for RMSE independent of deformation ($p < 0.001$) and post-hoc tests showed that ***full*** is the most accurate out of the three designs (all $p < 0.05$).

These results indicate that our design tool aids the user in selecting the most appropriate planar sensor embedding, and is indeed useful. In particular, the interplay between visualization of internal deformations and immediate display of predicted reconstruction error makes this task significantly easier. Furthermore, the iterative refinement of sensor routings using our optimization algorithm provides additional benefits over purely manual routings, illustrated by the differences in accuracy between ***full*** and ***visual***.

Finally, we conducted another experiment in order to better isolate the effect of sensor placement optimization from human contributions. To this end, we designed a 3D bar with

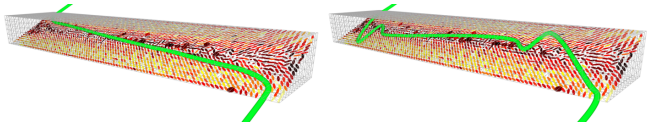

Figure 5. Single refinement step. Left: Initial condition. Right: Refinement produces symmetric routing for twist deformation.

a single sensor to reconstruct twisting about the major axis. We fabricate a best-guess version and then run a single refinement step (i.e., no user input). Fig. 5 shows the resulting routings. The predicted RMSE for the manual design is 4.2 times higher (2.542 mm) than the optimized version (0.643 mm). After fabrication, we measured an improvement of approximately 2.0 (2.397 mm versus 1.191 mm).

Sensor Count

The number of sensors is a central factor for recognition accuracy. Especially for untrained users, however, it is typically difficult to estimate how many sensors are required in order to obtain satisfying results. As illustrated with our two bar examples, a one-to-one mapping between user-provided example poses and sensors can lead to good results for simple geometries and few deformations. For geometrically more complex objects with an increasing number of example deformations, however, this simple scheme soon becomes intractable. Our design tool supports the user in this decision: sensing elements can be added and refined in an iterative fashion until the predicted reconstruction accuracy, immediately displayed by our tool, is sufficiently low. This workflow, which is best seen in the accompanying video, is illustrated in Fig. 6, which shows the example of a thin deformable sheet. To design this input device, we started with two sensors, which we refined and edited in several iterations. Although some deformation modes could be recognized accurately, the reconstruction error remained high for the remaining deformations. After adding two more sensors, we obtained a sufficiently low error for all three input deformations (the two front corners bending up individually, and together). The final specimen has

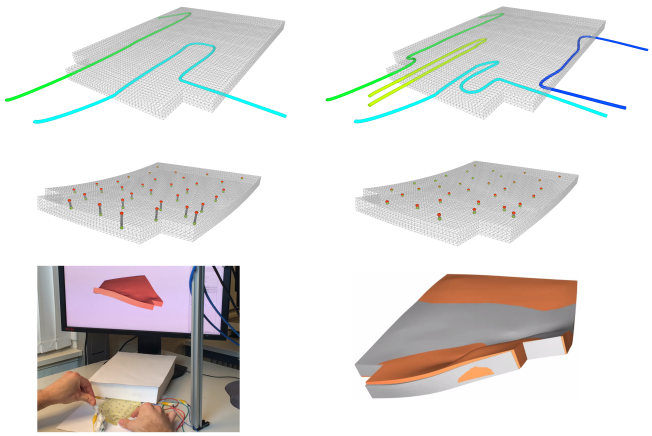

Figure 6. Snapshots of the design process. Top-left: the user placed, refined, and edited two sensors; expected reconstruction error remained high for a subset of deformations (middle-left). Final routing and (reduced) reconstruction error for a set of four sensors. Bottom row: interaction with fabricated device (left) and ground truth comparison (right).

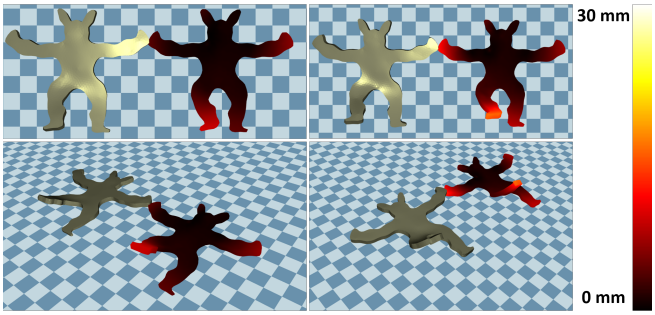

Figure 7. Continuous deformation of the Armadillo reconstructed from sensor data. Color coded error map on the right — darker is better.

an avgerage RMSE error of 1.38 mm (peak error 9.61 mm, avg. peak error 3.7 mm), again measured against ground truth data over a sequence of 2100 frames. As for our other prototypes, the discrepancy between predicted and measured accuracy can be attributed to modeling errors, sensor and measurement noise, and error introduced by the fabrication process.

Qualitative Evaluation

In addition to quantitative experiments we fabricated further 3D examples, tailored to specific application scenarios, and illustrating the utility of our method in creating fully customized deformable input devices.

3D Character Control

One interesting but also challenging application scenario is that of 3D character animation. We envision an artist or even an end-user creating a physical instance of an on-screen character by simply 3D printing it. After fabrication, the custom controller may be used to animate the on-screen character in a very direct and natural fashion. Fig. 1 depicts a custom controller based on the well-known Armadillo model. Fig. 7 illustrates the reconstruction accuracy achieved by the controller under different deformations. In general the reconstruction quality is good (avg. RMSE of 2.023 mm over 2100 frames) and the controller is responsive even under challenging out-of-plane deformations.

The armadillo created with our method also performs better than a best-guess manual routing. Fig. 8 illustrates this fact

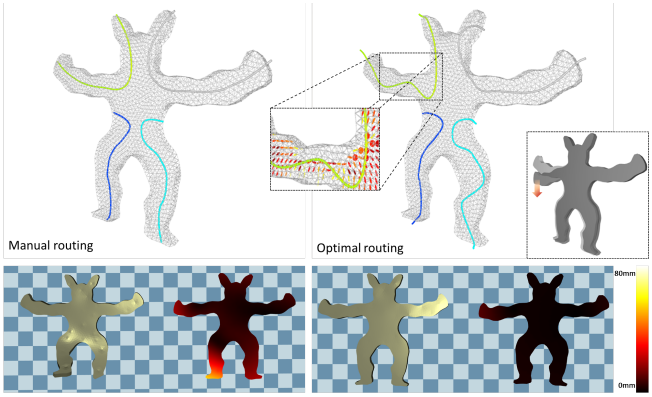

Figure 8. Comparison of best-guess manual and optimized routings.

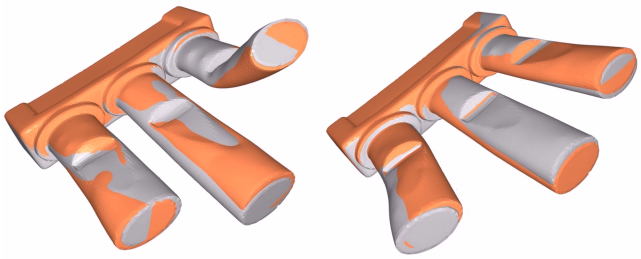

Figure 9. Two example deformations of the organ pipe model designed with our method.

for an example deformation. Notice how the optimized character reconstructs the desired deformation with very low overall error, while the manually designed controller produces larger reconstruction errors that concentrate in regions around both the left foot and left hand of the character.

Musical Instrument

For a final example, we fabricated a custom musical instrument inspired by an organ pipe. The 3D object is designed to be held in one hand and manipulated by the other. Four deformations (two bends, a twist, and a pull) of the pipes were specified by the user. We use six internal sensing elements to recover these deformations (cf. Fig. 9 and video). Beyond reconstructing deformed geometry, the output *could* be mapped to, e.g., tone, pitch, and amplitude of a simple midi-synthesizer. This example demonstrates that our method can be used to design and fabricate more complex, fully three-dimensional objects achieving good reconstruction accuracy.

LIMITATIONS

We have proposed an approach for design and fabrication of custom-shaped, flexible, and self-sensing input devices. The technical core of our method is an algorithm for computing optimal sensor routings that directly minimize reconstruction error. Experimentally, we have shown that the method produces routings that outperform best-guess manual versions by significant margins and hence provide higher interaction fidelity. We have furthermore shown that the visualization of internal deformation fields, in combination with our optimization algorithm, can help in designing better sensor configurations than those obtained with best-guess manual placement. We have demonstrated the utility and flexibility of our method by fabricating several flexible input devices.

However, our method is not without limitations. Currently, we rely on example poses as input. While our method is capable of recognizing continuous deformations in the entire space spanned by this set of poses (including in between poses), our method is currently not capable of reconstructing entirely unseen deformations. For example, if the user provides only a pose for the Armadillo with one leg raised, reconstructing a sideways motion or a bend of the leg in opposite direction would be failure cases. Alleviating this limitation is an interesting direction for future work and possible solutions may include easier methods to define interaction spaces (such as programming-by-demonstration). A different but equally promising direction could be to leverage modal analysis to identify areas of high deformation for sensor refinement.

We generally found that the predicted reconstruction error in our design tool was a good indicator of real world performance. However, there is still a considerable gap between simulation and real-world, best illustrated by the need for a machine-learning-based reconstruction algorithm. In some cases, fabricated specimens would perform poorly or not work at all. We have identified a number of factors that influence the success of the fabricated prototypes, and not all of them are modeled accurately or even included in our model. However, the limitations relating to the current state of fabrication technologies and available materials certainly have a significant impact. For example, in our model we assume a single solid object with perfectly embedded sensing elements, whereas in the real world we print objects in parts and manually embed conductive traces before assembly. Furthermore, our sensor response model is certainly not accurate and real sensor readings can deviate considerably from simulated values. These and other factors can contribute to situations where large reductions in error residual during optimization do not necessarily translate into good real-world behavior.

Future Work

We hope that future progress in digital fabrication technology will soon remove the necessity for a two-step fabrication process. There is already some success in improving printable conductive and stretchable elastomers with favorable properties for strain sensors [17]. Future printers will open up the possibility of printing with a wider selection of materials, and will thus solve the limitations of the current conductive materials and printing processes. In anticipation of these emerging technologies and materials, we have developed a computational design algorithm that, thanks to its generality, will transfer well to these new technologies. The changes introduced to accommodate the limitations of current fabrication technologies are minimal and do not restrict the method's applicability in the future.

We support the user in utilizing the method effectively via an easy-to-use graphical UI. However, we do currently not support the user in selecting which deformations should be sensed particularly well. We believe that the current division of labor between user and algorithm leverages the respective strengths of each. Nonetheless, this is an interesting area for future research, in particular the idea of combining computational design of interactions (i.e., which deformation is mapped to which task).

Finally, while we have performed and discussed extensive technical evaluations of our system, we have yet to perform extensive user tests. In particular, we would like to evaluate how successful our design tool is at enabling end-users to fabricate custom input devices.

CONCLUSION

In summary, we have proposed a computational design approach to the digital fabrication of deformable input devices. Our approach leverages internally routed piezoresistive wires for sensing of user input. In particular we have demonstrated rich natural interactions via continuous 3D deformations. We

consider the insight to leverage the reconstruction error directly as optimization objective one of our core contributions. While limitations of current fabrication technologies still impact interaction fidelity our work highlights the exciting possibilities that lie ahead in terms of fully customized, end-user designed interactive devices.

Acknowledgements

We thank Damian Karrer, Rocco Ghielmini and Jemin Hwangbo for their help in our initial explorations. We would like to thank Christian Schumacher for creating the video and Cécile Edwards-Rietmann for providing the voiceover. Maurizio Nitti helped us in designing our 3D characters. We thank Chiara Daraio for insightful discussions on material properties and 3D printing. We also thank the CHI reviewers for their feedback and guidance. Fabrizio Pece was supported by an ETH/Marie Curie fellowship (FEL-3314-1).

REFERENCES

1. Gilles Bailly and Antti Oulasvirta. 2014. Toward Optimal Menu Design. *interactions* 21, 4 (July 2014), 40–45. DOI: http://dx.doi.org/10.1145/2617814
2. Gilles Bailly, Antti Oulasvirta, Timo Kötzing, and Sabrina Hoppe. 2013. MenuOptimizer: Interactive Optimization of Menu Systems. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 331–342. DOI: http://dx.doi.org/10.1145/2501988.2502024
3. Ravin Balakrishnan, George Fitzmaurice, Gordon Kurtenbach, and Karan Singh. 1999. Exploring Interactive Curve and Surface Manipulation Using a Bend and Twist Sensitive Input Strip. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics (I3D '99)*. ACM, New York, NY, USA, 111–118. DOI: http://dx.doi.org/10.1145/300523.300536
4. Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. 2010. Design and Fabrication of Materials with Desired Deformation Behavior. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 29, 3 (2010).
5. 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 - UIST '13*. ACM Press, New York, New York, USA, 457–462. DOI: http://dx.doi.org/10.1145/2501988.2502027
6. Lee A. Danisch, Kevin Englehart, and Andrew Trivett. 1999. Spatially continuous six-degrees-of-freedom position and orientation sensor. (1999). DOI: http://dx.doi.org/10.1117/12.339112
7. Nan-Wei Gong, Jürgen Steimle, Simon Olberding, Steve Hodges, Nicholas Edward Gillian, Yoshihiro Kawahara, and Joseph A. Paradiso. 2014. PrintSense: a versatile sensing technique to support multimodal flexible surface interaction. In *Proceedings of ACM conference on Human factors in computing systems - ACM CHI*. ACM Press, New York, New York, USA, 1407–1410. DOI: http://dx.doi.org/10.1145/2556288.2557173
8. Fenner Drives Inc. Accessed: 2015-08-07. Fenner Drives Inc. NinjaFlex filament. http://www.ninjaflex3d.com. (Accessed: 2015-08-07).
9. MakerBot Industries. Accessed: 2015-08-05. MakerBot Replicator 2X. https://eu.makerbot.com/shop/de/3d-drucker/replicator-2x. (Accessed: 2015-08-05).
10. Alec Jacobson, Daniele Panozzo, Oliver Glauser, Cédric Pradalier, Otmar Hilliges, and Olga Sorkine-Hornung. 2014. Tangible and Modular Input Device for Character Articulation. *ACM Trans. Graph.* 33, 4, Article 82 (July 2014), 12 pages. DOI: http://dx.doi.org/10.1145/2601097.2601112
11. Amit Joe Lopes, Eric MacDonald, and Ryan B. Wicker. 2012. Integrating stereolithography and direct print technologies for 3D structural electronics fabrication. *Rapid Prototyping Journal* 18, 2 (March 2012), 129–143. DOI: http://dx.doi.org/10.1108/13552541211212113
12. Wolfgang Kabsch. 1976. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography* 32, 5 (1976), 922–923.
13. Andreas Karrenbauer and Antti Oulasvirta. 2014. Improvements to Keyboard Optimization with Integer Programming. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 621–626. DOI: http://dx.doi.org/10.1145/2642918.2647382
14. T. Kato, A. Yamamoto, and T. Higuchi. 2003. Shape recognition using piezoelectric thin films. In *Industrial Technology, 2003 IEEE International Conference on*, Vol. 1. 112–116 Vol.1. DOI: http://dx.doi.org/10.1109/ICIT.2003.1290252
15. S. J. Leigh, R. J. Bradley, C. P. Purssell, D. R. Billson, and D. A. Hutchins. 2012. A Simple, Low-Cost Conductive Composite Material for 3D Printing of Electronic Sensors. *PLoS ONE* 7, 11 (2012). DOI: http://dx.doi.org/doi:10.1371/journal.pone.0049365
16. M Melnykowycz, B Koll, D Scharf, and F. Clemens. 2014. Comparison of Piezoresistive Monofilament Polymer Sensors. *Sensors* 14, 1 (2014), 1278–1294. DOI: http://dx.doi.org/10.3390/s140101278
17. Joseph T Muth, Daniel M Vogt, Ryan L Truby, Yiğit Mengüç, David B Kolesky, Robert J Wood, and Jennifer A Lewis. 2014. Embedded 3D Printing of Strain Sensors within Highly Stretchable Elastomers. *Advanced Materials* 26, 36 (2014), 6307–6312.

18. 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 - UIST '14*. ACM Press, New York, New York, USA, 281–290. DOI: http://dx.doi.org/10.1145/2642918.2647413

19. Huaishu Peng, Jennifer Mankoff, Scott E. Hudson, and James McCann. 2015. A Layered Fabric 3D Printer for Soft Interactive Objects. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1789–1798. DOI: http://dx.doi.org/10.1145/2702123.2702327

20. Thiago Pereira, Szymon Rusinkiewicz, and Wojciech Matusik. 2014. Computational Light Routing: 3D Printed Optical Fibers For Sensing and Display. *ACM Transactions on Graphics* 33, 3 (June 2014), 1–13. DOI: http://dx.doi.org/10.1145/2602140

21. 3D Prima. Accessed: 2015-08-03. 3D Prima Conductive ABS. http://www.3dprima.com/en/filaments-for-3d-printers/abs-175mm/3d-prima-conductive-abs-175mm-1-kg-spool-black.html. (Accessed: 2015-08-03).

22. Recreus. Accessed: 2015-08-07. Recreus Filaflex filament. http://recreus.com/en/4filaflex-filaments. (Accessed: 2015-08-07).

23. Christian Rendl, Patrick Greindl, Michael Haller, Martin Zirkl, Barbara Stadlober, and Paul Hartmann. 2012. PyzoFlex: printed piezoelectric pressure sensing foil. In *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*. ACM Press, New York, New York, USA, 509. DOI: http://dx.doi.org/10.1145/2380116.2380180

24. Christian Rendl, David Kim, Sean Fanello, Patrick Parzer, Christoph Rhemann, Jonathan Taylor, Martin Zirkl, Gregor Scheipl, Thomas Rothländer, Michael Haller, and Shahram Izadi. 2014. FlexSense: A Transparent Self-sensing Deformable Surface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. 129–138. DOI: http://dx.doi.org/10.1145/2642918.2647405

25. Valkyrie Savage, Andrew Head, Björn Hartmann, Dan B. Goldman, Gautham Mysore, and Wilmot Li. 2015. Lamello: Passive Acoustic Sensing for Tangible Input Components. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*. ACM Press, New York, New York, USA, 1277–1280. DOI: http://dx.doi.org/10.1145/2702123.2702207

26. 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 (UIST '14)*. 3–12. DOI: http://dx.doi.org/10.1145/2642918.2647374

27. Carsten Schwesig, Ivan Poupyrev, and Eijiro Mori. 2004. Gummi: A Bendable Computer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 263–270. DOI: http://dx.doi.org/10.1145/985692.985726

28. Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. 2013. Computational Design of Actuated Deformable Characters. *ACM Trans. Graph.* 32, 4, Article 82 (July 2013), 10 pages. DOI: http://dx.doi.org/10.1145/2461912.2461979

29. Ronit Slyper, Ivan Poupyrev, and Jessica Hodgins. 2011. Sensing through structure: designing soft silicone sensors. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*. ACM, 213–220.

30. Brian A. Smith, Xiaojun Bi, and Shumin Zhai. 2015. Optimizing Touchscreen Keyboards for Gesture Typing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3365–3374. DOI: http://dx.doi.org/10.1145/2702123.2702357

31. Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, Vol. 4.

32. Srinath Sridhar, Anna Maria Feit, Christian Theobalt, and Antti Oulasvirta. 2015. Investigating the Dexterity of Multi-Finger Input for Mid-Air Text Entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3643–3652. DOI: http://dx.doi.org/10.1145/2702123.2702136

33. Jürgen Steimle, Andreas Jordt, and Pattie Maes. 2013. Flexpad: Highly Flexible Bending Interactions for Projected Handheld Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 237–246. DOI: http://dx.doi.org/10.1145/2470654.2470688

34. Stratasys. Accessed: 2015-09-25. Objet Connex 3D Printers. http://www.stratasys.com/3d-printers/design-series/connex-systems. (Accessed: 2015-09-25).

35. Creative Tools. Accessed: 2015-08-03. Palmiga conductive TPU. http://www.creativetools.se/pi-etpu-95-250-carbon-black-1kg-se. (Accessed: 2015-08-03).

36. Marynel Vázquez, Eric Brockmeyer, Ruta Desai, Chris Harrison, and Scott E. Hudson. 2015. 3D Printing

Pneumatic Device Controls with Variable Activation Force Capabilities. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1295–1304. DOI: http://dx.doi.org/10.1145/2702123.2702569

37. 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 - UIST '12*. ACM Press, New York, New York, USA, 589–598. DOI: http://dx.doi.org/10.1145/2380116.2380190

38. Martin Zirkl, Anurak Sawatdee, Uta Helbig, Markus Krause, Gregor Scheipl, Elke Kraker, Peter Andersson Ersman, David Nilsson, Duncan Platt, Peter Bodö, Siegfried Bauer, Gerhard Domann, and Barbara Stadlober. 2011. An all-printed ferroelectric active matrix sensor network based on only five functional materials forming a touchless control interface. *Advanced materials (Deerfield Beach, Fla.)* 23, 18 (May 2011), 2069–74. DOI: http://dx.doi.org/10.1002/adma.201100054

APPENDIX

DISCUSSION OF MATERIALS AND FABRICATION

There is a wide variety of different conductive materials available that one can use for 3D printing deformable objects. Rubber and ABS filament for extrusion printers can be created in conductive versions by adding carbon black powder [15] or graphene (the result is often called a composite material). In contrast to printed electronics, where high electrical conductivity is of uttermost importance, in our case we can relax this requirement and work with weakly conductive material. In our experiments, we tested two commercially available 3D printing filaments: electrically conductive ABS from 3D-Prima [21] (eABS) and electrically conductive thermoplastic polyurethane (eTPU) from Creative Tools [35]. While eABS is often used as a non-deformable print material, thin sheets and wires can be bent easily within a reasonable range without breaking and exhibit easily measurable changes in resistance. In contrast, eTPU is highly deformable and has a fracture strain of approximately 250%. Note that inks based on silver, carbon black, or other conductive materials can also be applied to models to make conductive wires. However, in our experiments, they turned out to be very brittle and easily break when used on deformable models.

In addition to piezoresistive filaments, we also investigated a commercially available flexible cylindrical cord from Images Scientific Instruments Inc. (eCord), which we manually embed in a two-step fabrication process in 3D printed objects. As the cord is stretched 50%, its resistivity will approximately double.

In a first experiment, we studied the relation of strain to resistivity change. While all materials showed a significant change of resistivity under deformation, we observed a substantial difference in time-dependence and fluctuations between test runs. In our experiments, the response of eABS and eCord under deformation was close to linear and the dependence on the rate of change small, which makes them both ideal for our purposes. In contrast to these materials, we observed various complex effects with eTPU which make the resistance highly dependent on the rate of deformation over time and the general history of the material deformation. This makes it extremely hard to reliably predict deformations solely based on observations of the resistivity change. Figure 10 illustrates this effect with a simple example in which we repeatedly elongated a 67 mm long eCord sample and eTPU filament.

Based on these findings, depending on the required accuracy, we carefully trade complexity of the deformation-to-signal correlation with fracture strain and manufacturing complexity. For the directly 3D printed examples, we choose to use dual head extrusion printing using an inexpensive, commercially available printer (MakerBot Replicator 2X [9]). We note, however, that in extrusion based prints, not all filament types stick to one another. For instance, conductive ABS does not stick well to NinjaFlex from Fenner Drives Inc. [8], but bonds well with Filaflex from Recreus [22], a thermoplastic elastomer (TPE) material. With very soft TPU filament also coming in a conductive version, eTPU, we have a natural choice for a reliable printing processes.

For examples that require a more accurate deformation reconstruction, we resort to eCord. We split the virtual object in multiple pieces such that the cords can be easily embedded in a postprocess and 3D print the individual pieces on a Stratasys Connex machine using TangoBlackPlus [34], a non-conductive rubber-like material. To prevent slipping of the cord inside the model, we glue all the pieces, including the eCord material, using a stretchable Silicone glue.

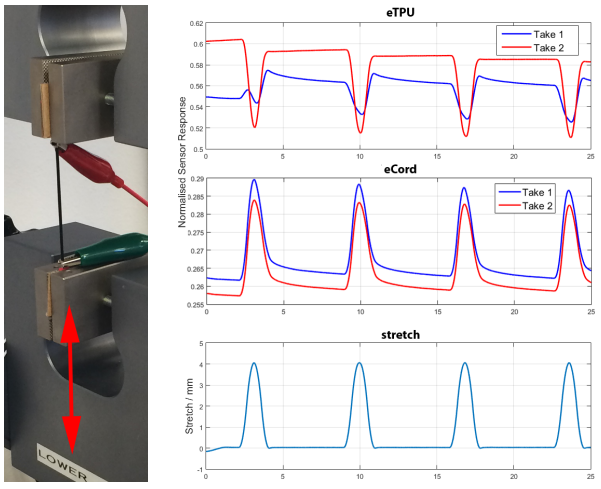

Figure 10. Characterization of electrically conductive filament. Using a tensile test machine (left), we measured the resistance change of eTPU (top, right) and eCord (center, right) twice, using the displacement pattern shown on the bottom right.