FlexSense: A Transparent Self-Sensing Deformable Surface

Christian Rendl[1], David Kim[2], Sean Fanello[2], Patrick Parzer[1], Christoph Rhemann[2], Jonathan Taylor[2], Martin Zirkl[3], Gregor Scheipl[3], Thomas Rothländer[3], Michael Haller[1], Shahram Izadi[2]

[1]Media Interaction Lab, University of Applied Sciences Upper Austria
[2]Microsoft Research
[3]Institute of Surface Technologies and Photonics, Joanneum Research

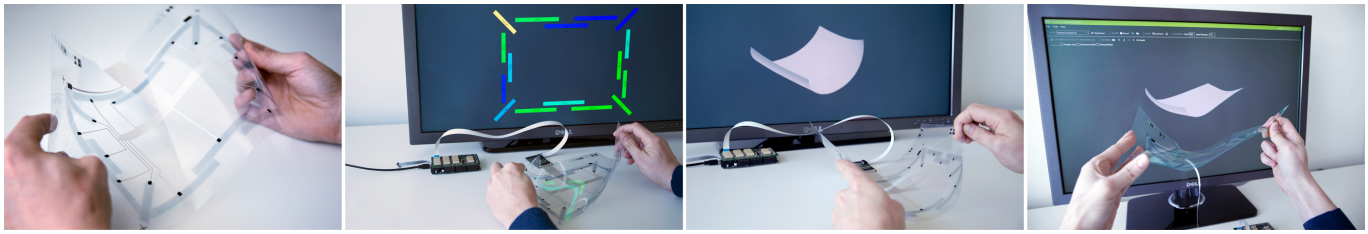

Figure 1: FlexSense is a fully flexible, transparent, thin-film surface, comprising of sparse printed piezoelectric sensors (left). Our main contribution is a new set of algorithms that takes these sparse sensor measurements (center), and reconstruct the dense 3D shape of the device and complex deformations in real-time (right).

ABSTRACT

We present FlexSense, a new thin-film, transparent sensing surface based on printed piezoelectric sensors, which can reconstruct complex deformations without the need for any external sensing, such as cameras. FlexSense provides a fully self-contained setup which improves mobility and is not affected from occlusions. Using only a sparse set of sensors, printed on the periphery of the surface substrate, we devise two new algorithms to fully reconstruct the complex deformations of the sheet, using only these sparse sensor measurements. An evaluation shows that both proposed algorithms are capable of reconstructing complex deformations accurately. We demonstrate how FlexSense can be used for a variety of 2.5D interactions, including as a transparent cover for tablets where bending can be performed alongside touch to enable magic lens style effects, layered input, and mode switching, as well as the ability to use our device as a high degree-of-freedom input controller for gaming and beyond.

Author Keywords

Flexible, transparent, sensor, deformation, reconstruction.

INTRODUCTION

There has been considerable interest in the area of flexible or deformable input/output (IO) digital surfaces, especially with recent advances in nano-technology, such as flexible transistors, eInk & OLED displays, as well as printed sensors. The promise of such devices is making digital interaction as simple as interacting with a sheet of paper. By bending, rolling or flexing areas of the device, a variety of interactions can be enabled, in a very physical and tangible manner.

Whilst the vision of flexible IO devices has existed for some time, there have been few self-contained devices that enable rich *continuous* user input. Researchers have either created devices with limited *discrete* bending gestures, or prototyped interaction techniques using *external* sensors, typically camera-based vision systems. Whilst demonstrating compelling results and applications for bending-based interactions, the systems suffer from practical and interactive limitations. For example, the bend sensors used in [7, 17] are limited to simple bending of device edges, rather than the complex deformations one expects when interacting naturally with a sheet of paper. In contrast, vision-based systems e.g. [31] are not self-contained, are more costly and bulky, and can suffer from occlusions, particularly when the hand is interacting with the surface.

In this paper, we present FlexSense, a transparent thin input surface that is capable of precisely reconstructing complex and continuous deformations, without any external sensing infrastructure (see Figure 1). We build on prior work with printed piezoelectric sensors (previously used for touch, gesture and pressure sensing, [23, 41]). Our new design uses only a sparse set of piezoelectric sensors printed on the periphery of the surface substrate. A novel set of algorithms fully reconstruct the surface geometry and detailed deformations being performed, purely by interpreting these sparse sensor measurements. This allows an entirely self-contained setup, free of external vision-based sensors and their inherent limitations. Such a device can be used for a variety of applications, including a transparent cover for tablets supporting complex 2.5D deformations for enhanced visualization, mode switching and input, alongside touch; or as a high degree-of-freedom (DoF) input controller.

In summary our contributions are as follows:

- We present a new sensor layout based on the prior PyzoFlex system [23], specifically for sensing precise and continuous surface deformations. This previous work demonstrated the use of printed piezoelectric sensors for touch and pressure sensing. In contrast, we present the idea of using these sensors to enable rich, bidirectional bending interactions. This includes the design of a new layout, associated sensor and driver electronics specifically for this purpose.
- Our main contribution are two algorithms that can take measurements from the sparse piezoelectric sensors and

accurately reconstruct the 3D shape of the surface. This reconstructed shape can be used to detect a wide-range of flex gestures. The complexity of the deformations afforded by our reconstruction algorithms have yet to be seen with 'self-sensing' (i.e. self-contained) devices, and would typically require external cameras and infrastructure.

- We train and evaluate the different algorithms using a ground-truth multi-camera rig, and discuss the trade-offs between implementation complexity and reconstruction accuracy. We also compare to a single camera baseline.
- Finally, we demonstrate new interaction techniques and applications afforded by such a novel sensor design and reconstruction algorithms. In particular as a cover for tablets where IO is coupled, and as a high DoF input controller.

RELATED WORK

There is broad range of work on flexible sensing and displays. [31] distinguishes this work into two categories. The first are *external input devices* where the sensor is used to control a remote display and UI, and output is typically non-flexible. The second are *deformable handheld devices*, where input and output is coupled, and the display can also be deformable. We extend this categorization further, by identifying differences in sensing approaches. *Self-sensing* systems contain onboard sensors on the device, which can be used to directly estimate the deformation (e.g. devices based on embedded bend sensors). *External sensing* systems use sensors embedded in the environment rather than the device to estimate deformations (e.g. camera-based deformation systems). In this section we explore prior work on bendable and deformable surfaces, based on this broad taxonomy.

Perhaps the first example of a deformable *external input device* that supported *self-sensing* is the work by Balakrishnan et. al. [1] which used the ShapeTape sensor [3], and a NURBS (non-uniform rational basis spline) representation for 3D modeling. ShapeTape is a thin long rubber tape subdivided into a series of fiber optic bend sensors, each detecting bend and twist (inspired by the sensors used in data gloves e.g. [40]). This showed the potential for exploiting bimanual input for modeling 3D curves. [12] demonstrated the use of piezoelectric thin films to produce a device similar to ShapeTape. [2] use *external sensing* in the form of camera and vision techniques to reconstruct complex curves from long passive pieces of wire. The form-factor of these devices make them ideal for 3D curve drawing, but more complex 2D and 2.5D flexing and bending interactions are less natural.

The first conceptual *deformable handheld device* was the Gummi system [28]. The project was motivated by innovations in flexible displays and transistors. The prototype device could be considered *self-sensing* but used bend sensors placed behind a TFT display and a 2D trackpad at the rear. Only small discrete bending gestures were supported, but the work demonstrated some of the interactive possibilities that bending interfaces afford. [32] use a similar prototype setup to Gummi, for flicking through pages in an e-reader. Whilst IO is coupled in these systems, the display remains rigid.

PaperPhone was one of the first devices that coupled flexible IO into a single self-contained device [17]. A flexible PCB housed five bend sensors, and a kNN-based classifier was used to detect discrete bend gestures. A user study proposed a classification scheme that categorized bend gestures by location (top corner, side, or bottom corner) and direction (up or down). [36] also use a flexible PCB but with different bend sensor layout, to extend this classification scheme to include bend size and angle. The Kinectic Phone by Nokia [15] demonstrates a full color bendable phone, a manifestation of the early Gummi vision. Using this device, Kildal et al. [3] explore bending and twisting, and propose design guidelines for deformable devices. PaperTab [33] is a self-contained electronic-reader with two bidirectional FlexPoint sensors for input. The system demonstrates a progression of the digital desk concept [9, 38], which is a clear motivation for flexible, paper-like devices, but for the first time uses self-contained IO, as opposed to projectors and cameras.

Other *self-sensing* systems look purely at input sensing. Bookisheet used two bend sensors on the back of acrylic sheets to create both a dual and single sheet flexible input device [37]. Bend gestures were coarsely categorized into four discrete classes, and mapped to turning pages in eBooks. FlexRemote [21] consists of 16 flex sensors on the periphery of a thin acrylic sheet which can recognize eight deformation gestures for remote input. Twend [8] uses eight fiber optic bend sensors embedded into a thicker substrate to detect 18 unique bends.

All these systems can be thought as *self-sensing* allowing for a great deal of mobility and compactness, and avoiding occlusions. These systems typically are not focused on accurate 3D reconstructions of surface shape, and most detect discrete bend gestures. This is partly because of the complexity of mapping from raw sensor readings to precise continuous deformations. An exploratory study by Lee et al. [20] captures data from users deforming a variety of non-digital substrates including paper, plastic as well as stretchable fabrics, demonstrating the richness and complexity of deformations afforded. These types of interactions can be difficult to capture purely using discrete gestures.

To enable such types of reconstructions, systems have employed *external sensors*, generally in the form of cameras. [6] use retro-reflective tags embedded into a deformable substrate to support flexing and folding gestures. [18] use the Vicon motion tracking system to reconstruct sparse 3D points and fit a mesh for simple 3D modeling. [16] use a magnetic tracker and spherical projection onto a deforming sheet of diffuse acrylic. [39] use a combination of external WiiMote sensor and projector, coupled with onboard pressure and bend sensors to create a hybrid deformable projected display. [31] use a Kinect and projector for advanced manipulations of a sheet of paper with coupled output. A 25×25 vertex plane is deformed to fit the observed Kinect data. Eight distinct poses can be linearly combined using a weighted interpolation scheme to form more complex shapes. Further, the system analyses the Kinect dot pattern to disambiguate the user's hand from the deformation surface.

Related to this area, is work on projector-vision based foldable [19, 13], rollable [14] and shape-changing displays [25]. Another related field is the shape sensing of malleable surfaces and materials. In Jamming User Interfaces [5], for example, the authors use two different techniques, namely structured light and electric field sensing for deriving shape information. DeforMe [22] projects images realistically onto deformed surfaces by tracking the deformation with an invisible infrared-based dot pattern on the material. In contrast,

PhotoelasticTouch [26] detects changing optical properties of a transparent deformable material, and Digital Foam [29] introduces a deformable device with embedded pressure sensors for 3D modeling.

Whilst interesting areas of research, our work is firmly focused on reconstructing the 3D shape and continuous deformations of a thin transparent surface. PrintSense demonstrates capacitive touch and proximity sensing on a flexible substrate, and allows for bend sensing using transmit and receive electrodes. Flexible touch and pressure sensors have been demonstrated, either using transparent piezoelectric sensors [23] or opaque IFSR sensors [24]. Murata have developed a high-transparency organic piezoelectric film for bend sensing on mobile devices[1] although details are currently limited.

Our work is inspired by and builds on these previous systems. We support rich, continuous and accurate 3D reconstructions of deformations of the surface, allowing high DoF input. Our system is fully self-sensing, enabling a compact, mobile form-factor, but without limiting the range of deformations. Our sensor is semi-transparent, allowing for unique capabilities, such as allowing placement at the front of a display rather than embedded in the back. Our sensor therefore supports both, the use as an external input device, or coupled closely with a display to support novel application scenarios.

DESIGNING FLEXSENSE

Our work builds on the flexible, transparent PyzoFlex sensor [23, 41]. In this section we provide a brief introduction to this work, which specifically relates to using these sensors for bending. We refer the reader to [23] for further technical details regarding the underlying piezoelectric sensor. Whereas in this prior work, the sensors were used for pressure sensing, our main goal in this paper is to exploit these sensors for precise 3D reconstruction of surface shape and bending to facilitate deformation-based interactions. This requires a rethinking of the existing sensor layout, as detailed in this section.

Piezoelectric bend sensors

Piezoelectric sensors work as follows: deformations of a piezo element causes a change in the surface charge density of the material resulting in a charge appearing between the electrodes. The amplitude and frequency of the signal is directly proportional to the applied mechanical stress. Since piezoelectricity reacts to mechanical stress, continuous bending is a well-suited application for such sensors. We propose an entirely printed, semi-transparent, and bidirectional bend sensor based on different functional inks (see Figure 2). The active sensor material is formed by a poled copolymer P(VDF-TrFE) which shows a large piezoelectric coefficient (32 pC/N) and can be printed as a 5μm thick transparent layer [41]. The screen printing process in general ensures low-cost and simple fabrication without cleanroom requirements or evaporation.

All components of the sensor have relatively good transparency values with low distracting absorbency ($\sim 85\%$). Note that the current sensors are optimized to provide a good trade-off between signal strength and transparency. If even higher transparency ($> 90\%$) is desired, metal nanowire inks can be used as electrode material instead of PEDOT:PSS. For reading sensor measurements printed, non-transparent conductive silver wires are used. In order to limit occlusions caused by silver

[1] `http://murata.com/new/news_release/2011/0921`

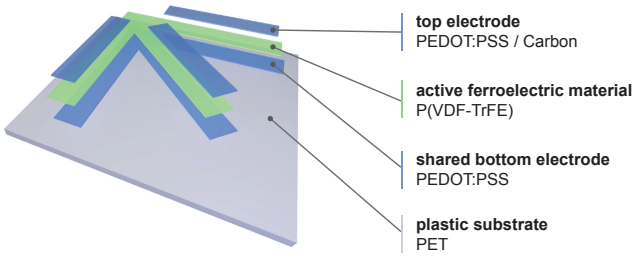

Figure 2: Our piezoelectric bend sensors consist of different, entirely printable functional inks. To have less (non-transparent) conductive silver lines all bend sensors printed on one substrate share one bottom electrode.

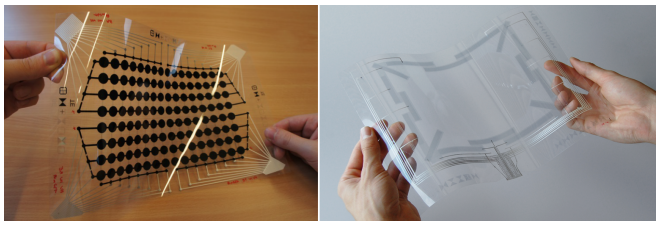

Figure 3: Layout of original PyzoFlex sensor (left) and our deformable sensor (right).

lines, all sensors on one substrate share one bottom electrode, which is separately wired to the driver electronics. The conductive traces can be placed freely across the surface, depending on application scenarios e.g. on the periphery or more central.

Sensor Layout

Multiple sensors must be combined together in order to be able to reconstruct the full deformations of the surface. We chose an A4 like form-factor because this matches commercially available flexible displays as well as common tablet screens, but is also big enough to be used as standalone input sensor.

Figure 3 (left) shows the original PyzoFlex sensor layout. As this layout was specifically designed for touch and pressure sensing, a dense grid of 16×8 sensors was employed in an active matrix arrangement. However, for bend sensing and shape reconstruction, this design is non-optimal. First, there is a large amount of sensor redundancy across the film. Second, the active matrix requires a large part of the outer film to be used for the printed silver wires. Finally, the active matrix scheme can suffer from ambiguities when many sensor measurements must be read at once, making such an arrangement problematic when bending (for more details see [23]).

Instead for FlexSense, we sought to use a sparse set of sensors, but placed in a more meaningful arrangement to optimize for a wide variety of continuous and smooth deformations, including extreme bends. The final design of our layout is shown in Figure 3 (right). In the remainder of this section, we provide more rationale for this sensor layout.

Existing Layouts in Related Work From related work, we identified the following, most essential requirements for our sensor layout: In order to track complex deformations an optimal bend sensor should consider properties such as location (where does the bend action happens), direction (upwards or downwards), size (involved surface area), angle (sharp or round angle), speed, and duration of bend actions [36].

To infer an optimal layout, we investigated arrangements used in prior work [7, 17, 36] in terms of the presented requirements and their sensor alignment. We recreated those layouts using

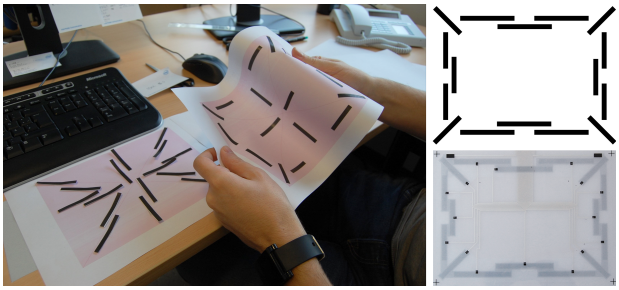

Figure 4: Paper prototyping to evaluate designs further (left). Our final layout (top right) and final printed clear surface (bottom right) is an optimal trade-off between limited amount of sensors and strong deformation sensing capabilities.

off-the-shelf bend sensor strips (as used in the papers) to be able to understand their strengths and weaknesses. Hence, we combined different concepts of existing sensor configurations into one solution, which is capable of tracking the identified requirements and therefore complex deformations.

Our Layout To create and evaluate several different sensor patterns, we used paper and Scotch tape for fast prototyping (see Figure 4, left). This enabled us to quickly evaluate different designs "hands-on". The flexibility of paper and tape allowed us to bend the prototypes into all directions revealing the pros and cons for each pattern, and a ruler was used to follow bend directions/orientations and to identify spots which are not covered by any sensors.

The final layout arranges 16 sensors at an outer ring of the sheet (see Figure 4, right). Since we use a plastic foil as substrate, it is not possible to deform the sheet only in the center. Therefore, every deformation changes the shape of the edges, which means that by tracking the edges accurately nearly every shape deformation can be reconstructed. Another strength of this layout is that each sensor overlaps with one or multiple other sensors resulting in additional information as to where a bend actually happens (as proposed in [36]). For example, the corner areas which are particularly interesting for bend interaction are covered by three sensors, one at a $45°$ angle and two at right angles. From this arrangement it is possible to accurately detect how the corner is bent (bend angle) as well as where the bend happens (bend position). Due to the permanent overlap of sensors nearly every deformation actuates multiple sensors, which enables the detection of a wide variety of possible deformations. Those complex patterns of sensor actuations allows for the reconstruction of complex and continuous deformations.

Driver Electronics

As mentioned earlier, every piezoelectric sensor creates a surface charge which correlates to the applied deformation. Since the total number of sensors in our layout is small, we can connect each individually using conductive silver ink to the driver board. This removes the issues associated with the active matrix described earlier. Each sensor is connected to a LMC6482 CMOS rail-to-rail amplifier. These are placed on a small PCB board that is connected to the foil. Before the amplification the signals run through a low pass filter, which protects it against electrostatic discharge. After the amplification a second low pass filter protects the signal from anti-aliasing issues. Each signal gets measured through a MAX127 12-bit data acquisition system which sends the data via a two-wire serial interface to a microcontroller board (Atmel SAM3X8E).

Signal Processing

The electric charges generated in a piezoelectric sensor decay with a time constant determined by the dielectric constant, the internal resistance of the active material and the input impedance of the readout electronics. Due to this fact, the raw signal of a piezoelectric sensor are not usable for *absolute* measurements. However, once the parameters of the exponential signal discharge of the piezoelectric sensors are known it is possible to predict the signal progression over time. Every deformation applied to the sensor will cause a deviation of this predicted signal, which is directly proportional to the applied mechanical stress. Integration over these deviations leads to absolute sensor measurements, which directly correlate with the strength of the applied deformations [23]. Note there is a trade-off with this approach, as integrated errors can persist over time and lead to sensor drift. This can be eliminated with a heuristic which resets the signal e.g. when there is no active interaction on the sensor. In the next section, we describe two reconstruction algorithms where one relies on the integrated signal and one is able to bypass the described issues working directly from the raw signal.

RECONSTRUCTING FLEXSENSE

Now that we have described the FlexSense hardware and sensor layout, we turn our attention to how these sparse (raw or integrated) sensor measurements can be used to accurately recover the 3D shape of the surface during deformation.

Main Pipeline

Reconstructing the full 3D surface shape from a sparse set of sensor measurements is clearly a challenging task. Each sensor reading from our foil is an amplified voltage measurement, and somehow we need to map these combined values to a real-world reconstruction of the surface. In this section we present two data-driven algorithms that tackle this problem. Both of our methods are first trained using pairs of sensor measurements and ground truth 3D shape measurements of the foil. This pre-processing training phase is what enables our algorithms to infer the shape of the foil from the sparse sensor data at runtime.

To collect ground truth measurements of the shape of the foil together with corresponding sensor measurements, we follow the approach illustrated in Figure 5. We print an array of markers on a sheet of paper covering our sensor foil. Then we use a custom-built multi-camera rig (described later) to track the 3D position of the markers with high accuracy. We leverage multiple cameras in order to track as many markers as possible despite occlusions due to the deforming foil and interacting hands. To estimate the positions of the remaining occluded markers (shown in red in Figure 5 and 7) we exploit the prior knowledge that the foil is locally rigid to extrapolate the reconstructed surface (described later).

We use this sequence of resulting ground truth shapes along with the corresponding sparse sensor measurements to train our two algorithms. In the upper left part of Figure 6 we show the training for our *linear interpolation* based approach. It clusters the training data into $K = 30$ common shapes. The mean shape of each cluster is called a blendshape and is stored together with its averaged corresponding sensor measurements. At runtime (see Figure 6, bottom left) the sensor data is used to retrieve the $K = 7$ blendshapes which match the input sensor data (based on a distance metric described later). The

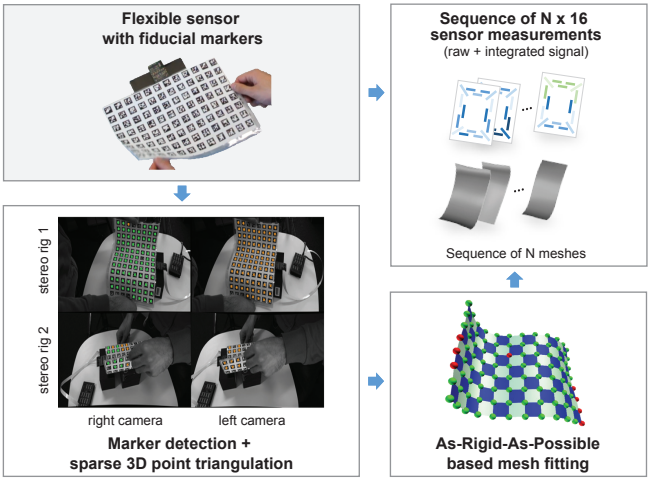

Figure 5: Processing pipeline for ground truth capture. We place fiducial markers on the front and backside of the foil and record their 3D position using a two stereo camera rig, together with the recorded sensor measurements (see text for more details).

final estimated shape is a weighted average of the retrieved blendshapes. This scheme is similar to the approach of [31] and its main benefit is that it is simple to implement, and can yield compelling results. However, a shortcoming of this approach is that it does not generalize well to large sensor signal variations. In practice, the measured sensor signal depends on many variables such as bending speed, temperature and whether the sensor is being grasped. In certain scenarios, this can result in inaccuracies and accumulation of error over time (see detailed discussion later).

To tackle these challenges we propose to use a more sophisticated *machine learning* algorithm. We train this method using the raw sensor data (see Figure 6, top right) to learn a (non-linear) mapping function from the raw sensor measurements to the mesh model. At runtime (see Figure 6, bottom right) it can efficiently infer the geometry given the learned mapping function and the sparse sensor data. This method generalizes well to unseen variations in the sensor signal and hence performs superior to the *linear interpolation* approach. However, it is more complex to implement. As shown later, there is value in each approach and we describe both in detail in the next section.

A tale of two algorithms

In this section we provide a deeper formulation of our two algorithms to aid replication.

Preliminaries Raw sensor measurements are represented as a vector $\mathbf{x} \in \mathbb{R}^F$ and integrated measurements as $\mathbf{z} \in \mathbb{R}^F$ ($F = 16$ for our sensor foil). Our goal is then to estimate the 3D "vertex" positions $\mathcal{V} = \{\mathbf{v}_1, ..., \mathbf{v}_N\} \subseteq \mathbb{R}^3$ of $N = 96$ canonical locations arranged in a 12×8 grid on the sheet. When the sheet is at rest, these vertices take on their known positions $\bar{\mathcal{V}} = \{\bar{\mathbf{v}}_1, ..., \bar{\mathbf{v}}_n\} \subseteq \mathbb{R}^3$.

In the following, it will sometimes be helpful to vectorize the vertex positions $\mathcal{V} = \{\mathbf{v}_n\}_{n=1}^N$ into a column vector $\mathbf{V} \in \mathbb{R}^{3N}$ that we wish to estimate given the raw or $\mathbf{x} \in \mathbb{R}^F$ or integrated signal $\mathbf{z} \in \mathbb{R}^F$. Further, when we are required to deal with a set of J instances of these variables, we will often label these variables as $\mathcal{V}_j = \{\mathbf{v}_n^j\}_{n=1}^N$, $\mathbf{V}_j$ and $\mathbf{x}_j$. In the case of a temporal sequence of length T we will instead use t to index these variable instances. We will, however, often leave off these indices when speaking of a single instance of these variables or when doing so makes equations more clear.

Ground Truth Capture As mentioned we take a data-driven approach for reconstructing the deformations of FlexSense and therefore require training data in order to correlate the sensor measurements of the flexible sensor with the 3D vertex positions. In this section, we therefore explain how to obtain ground truth for a variety of deformations.

In order to generate data of this form, we print an array of $N = 12 \times 8$ fiducial markers onto the front and backside of the flexible sensor foil as to coincide with the N vertex positions we wish to estimate (see Figure 5, top left). Although it is possible to directly estimate the position (and even orientation) of each marker with a single camera [11], this estimation is unreliable when the markers are considerably deformed as in our scenario. We therefore create a calibrated stereo rig (i.e., two cameras with known relative position and orientation) that we label A. In rig A, we are able to detect some subset $\mathcal{C}_A \subseteq \{1, ..., N\}$ of the N markers in both of the rig's cameras. For such a marker $n \in \mathcal{C}_A$, we use triangulation to estimate $\hat{\mathbf{v}}_n^A \in \mathbb{R}^3$ of vertex n.

Note that due to occlusions and strong deformations often only a small number of the marker positions can be estimated with stereo rig A. To increase the number of vertex positions that we can estimate, we use a second stereo rig that we label B to obtain estimates for a second set of markers $\mathcal{C}_B \subseteq \{1, ..., N\}$. The set of indices with estimations from both rigs $\mathcal{C}_A \cap \mathcal{C}_B$ defines two input point clouds in one-to-one correspondence. We obtain the optimal rigid transformation by using the Kabsch algorithm [10] to minimize

$$E_{\text{rigid}}(R, \tau) = \sum_{n \in \mathcal{C}_A \cap \mathcal{C}_B} \|\hat{\mathbf{v}}_n^A - (R\hat{\mathbf{v}}_n^B + \tau)\|^2 \qquad (1)$$

where $R \in SO^3$ is a rotation matrix and $\tau \in \mathbb{R}^3$ is a translation. Using this transformation, we can obtain a set $\mathcal{C} = \mathcal{C}_A \cup \mathcal{C}_B$ in a common coordinate frame. To do this for $n \in \mathcal{C}$, we estimate the corresponding vertex as

$$\hat{\mathbf{v}}_n = \begin{cases} \frac{1}{2}(\hat{\mathbf{v}}_n^A + R\hat{\mathbf{v}}_n^B + \tau) & n \in \mathcal{C}_A \cap \mathcal{C}_B \\ \hat{\mathbf{v}}_n^A, & n \in \mathcal{C} - \mathcal{C}_B \\ R\hat{\mathbf{v}}_n^B + \tau & n \in \mathcal{C} - \mathcal{C}_A \end{cases} \qquad (2)$$

As-Rigid-As-Possible Refinement Unfortunately, we are generally unable to estimate the locations of all N points even with two stereo rigs. Therefore, we exploit our prior knowledge that the sheet is locally rigid to extrapolate the reconstructed surface and provide estimates for the remaining points. To do this, we utilize the as-rigid-as-possible (ARAP) regularizer [30] which measures how locally non-rigid a point cloud with a neighborhood structure is. Here the local-rigidity for point n is measured in relation to the other points in a neighborhood $\mathcal{N}(n) \subseteq \{1, ..., N\}$, which in our case is simply the neighbors of n in the 12×8 grid. The ARAP measure of deformation with respect to the rest configuration $\bar{\mathcal{V}}$ is

$$E_{\text{ARAP}}(\mathcal{V}) = \sum_{n=1}^{N} \min_R \sum_{n' \in \mathcal{N}(n)} \|(\bar{\mathbf{v}}_{n'} - \bar{\mathbf{v}}_n) - R(\mathbf{v}_{n'} - \mathbf{v}_n)\|^2$$

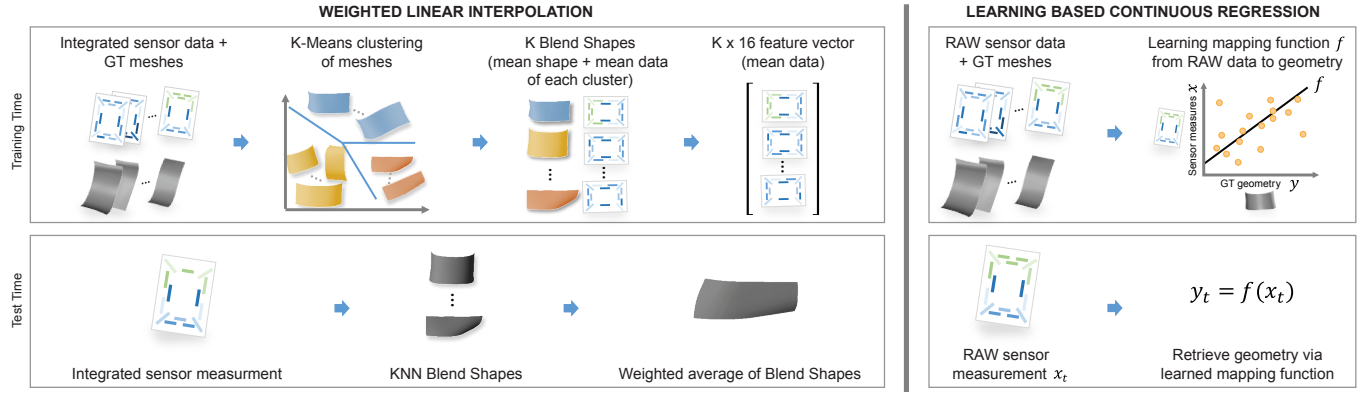

Figure 6: We present two reconstruction algorithms, both with different trade-offs. On the left: A weighted linear interpolation scheme using kNN. On the right: A machine learning based approach (see text for more details).

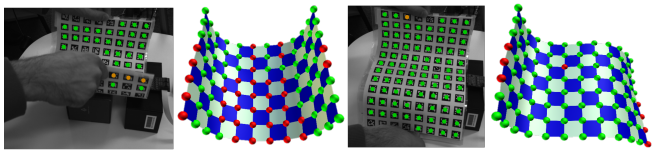

Figure 7: Ground truth capture. Markers are detected in each camera image (green and orange dots), and triangulated as a 3D point across each stereo pair (green dots in images and mesh). An ARAP refinement step regularizes for unseen or occluded vertices (red dots on mesh).

where $R \in SO^3$. This energy will be low for the smooth deformations that we expect (see Figure 4) as a rigid rotation can be used to approximate the deformation of the local deformation of each neighborhood. When this is impossible (e.g. sharp creases), however, the energy will be high. We therefore formulate the following energy

$$E(\mathcal{V}) = \sum_{n \in \mathcal{C}} \|\hat{\mathbf{v}}_n - \mathbf{v}_n\|^2 + \lambda_{\text{ARAP}} E_{\text{ARAP}}(\mathcal{V}) \qquad (3)$$

defined over the vertex positions $\mathcal{V}$. By minimizing this energy, we are seeking to find a set of vertex positions $\mathcal{V}$ that approximately match the estimates $\{\hat{\mathbf{v}}_n : n \in \mathcal{C}\}$ while not substantially deforming (in the ARAP sense) from the rest configuration.

To minimize (3), we follow the lead of [34] and label each rotation matrix in (3) with latent variables $\mathcal{R} = \{R_n\}_{n=1}^N \subseteq SO^3$. This allows one to define

$$E'_{\text{ARAP}}(\mathcal{V}, \mathcal{R}) = \sum_{n=1}^{N} \sum_{n' \in \mathcal{N}(n)} \|(\bar{\mathbf{v}}_{n'} - \bar{\mathbf{v}}_n) - R_n(\mathbf{v}_{n'} - \mathbf{v}_n)\|^2$$

with the property that $E_{\text{ARAP}}(\mathcal{V}) = \min_{\mathcal{R}} E'_{\text{ARAP}}(\mathcal{V}, \mathcal{R})$. Plugging this into (3), we get that $E(\mathcal{V}) = \min_{\mathcal{R}} E'(\mathcal{V}, \mathcal{R})$ where

$$E'(\mathcal{V}, \mathcal{R}) = \sum_{n \in \mathcal{C}} \|\hat{\mathbf{v}}_n - \mathbf{v}_n\|^2 + \lambda_{\text{ARAP}} E'_{\text{ARAP}}(\mathcal{V}, \mathcal{R}) \,. \qquad (4)$$

We thus minimize $E'(\mathcal{V}, \mathcal{R})$ using Levenberg-Marquardt[2]. We initialize this procedure by setting $\mathbf{v}_n = \bar{\mathbf{v}}_n$ and $R_n = I_3$ for each $n \in \{1, ..., N\}$. Finally, we parameterize the rotations using an axis-angle representation to enforce the constraint that each R_n remains a rotation matrix. In practice, this leads to a robust and real-time method for recovering the full mesh from a sparse set of vertices (see Figure 7).

[2] **https://code.google.com/p/ceres-solver/**

Dataset Construction We have shown how to capture ground truth vertex positions $\mathcal{V}$ to associate with a signal $\mathbf{x} \in F$. As we are capturing temporal sequences, in addition to the raw signal $\mathbf{x}_t$ and vertex positions $\mathcal{V}_t$ at time t, we will also record the integrated signal $\mathbf{z}_t \in \mathbb{R}^F$ up to time t and the instantaneous vertex displacements $\mathcal{D}_t = \{\mathbf{d}_1^t, ..., \mathbf{d}_N^t\} \subseteq \mathbb{R}^3$. The vertex displacement for vertex n is simply $\mathbf{v}_n^t - \mathbf{v}_n^{t-1}$. By capturing this information for a wide variety of sequences, we obtain a large fixed dataset $\{(\mathbf{x}_j, \mathbf{z}_j, \mathcal{V}_j, \mathcal{D}_j)\}_{j=1}^J$ that we will use for training.

Learning to Infer Shape and Deformation

In this section, we describe how to leverage this training set to predict the positions $\mathcal{V}$ of the vertices. In particular, as the rest pose $\bar{\mathcal{V}}$ is easily detected when $\mathbf{x} \approx \mathbf{0}$, we assume that we have seen a subsequent t time steps and attempt to predict $\mathcal{V}_t$. In order to estimate the 3D points $\mathcal{V}_t$ at time t, we propose two different approaches, both data-driven. The first uses a distance metric in the sensor input space (i.e. voltage measurements) to interpolate a set of key deformation modes. The second one uses machine learning techniques to directly regress the 3D vertex displacements.

Nearest Neighbor Lookup and Linear Interpolation Our first method assumes that the vertex position vector $\mathbf{V} \in \mathbb{R}^{3N}$ is a linear combination of a set of K blendshapes or deformation modes which we denote as $\{\mathbf{B}_1, ..., \mathbf{B}_K\} \subseteq \mathbb{R}^{3N}$. That is,

$$\mathbf{V} = \sum_{k}^{K} \alpha_k \mathbf{B}_k \qquad (5)$$

where α_k defines the weight of the k'th deformation mode $\{v_{nk}\}_{n=1}^N \subseteq \mathbb{R}^3$. To find these K deformations we run K-means to extract K modes $\{\bar{\mathbf{v}}_k\}_{k=1}^K \subseteq$ from our set and which appear to cover all the common configurations we expect to use. For each k, we average the integrated signal

$$\bar{\mathbf{z}}_k = \frac{1}{|\zeta_k|} \sum_{j \in \zeta_k} \mathbf{z}_j \qquad (6)$$

where $\zeta_k \subseteq \mathbb{R}^F$ is the set of integrated signals in the training set whose corresponding vertex positions were assigned to mode k in K-means. At runtime we see a new integrated signal $\mathbf{z}$ and compute α_k as

$$\alpha_k = \left(1 - \left(\frac{|\bar{\mathbf{z}}_k - \mathbf{z}_t| \frac{F}{f_k}}{\sum_j |\bar{\mathbf{z}}_j - \mathbf{z}_t| \frac{F}{f_j}}\right)\right)^{\beta}, \qquad (7)$$

where $\frac{F}{f_k}$ weights the distances based on the number of sensors involved during the bending gesture. In particular F is the total number of sensors attached to the sheet and f_k are the activated sensors of the model k, those absolute values are greater than $\theta = 200$; β is a regularization term to ensure smoothness during the transition between different configurations. We then plug these weights into (5) to reconstruct the positions $\mathcal{V}$.

Learning-based Continuous Regression The basic method proposed in the previous section assumes that all the possible poses can be generated through a linear combination of K modes. Also, recall that the signal $\mathbf{z}$ comes from an integration process that could lead to drift effects over the time. Unfortunately, these properties can lead to inaccuracies in the final reconstruction.

To address these issues, we consider regression-based methods that directly estimate the vertex displacements $\mathcal{D}$ using the raw signal $\mathbf{x} \in \mathbb{R}^F$. In particular, we seek to leverage our large training set to directly learn a mapping

$$f_{in} : \mathbf{x} \rightarrow \mathbf{d}_{in} \quad \forall n = 1, \ldots, N \quad \forall i = 1, 2, 3 \tag{8}$$

from the raw signal $\mathbf{x}$ to the displacement $\mathbf{d}_{in}$ in coordinate i of vertex n. For coordinate i of vertex n, we extract from our training set the following set of tuples $\{(\mathbf{x}_j, \mathbf{d}^j_{in})\}^J_{j=1}$ of size J. We use this to learn a function f_{in} that minimizes the empirical risk

$$\frac{1}{J} \sum_{j=1}^{J} \mathcal{L}\big(f_{in}(\mathbf{x}_j, \mathbf{d}^j_{in})\big) + \mathcal{R}(f), \tag{9}$$

where $\mathcal{L}(\circ)$ is the loss function and $\mathcal{R}(\circ)$ is a regularization term that gives a tradeoff between the accuracy and the complexity of the model.

Linear Model. Since we want to predict the position of all the N vertices with real-time performance we use a linear regression model: $f_{in}(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}_{in}$, with $\mathbf{w}_{in} \in \mathbb{R}^F$. Indeed, with a high sampling rate of the signal, it is reasonable to assume that the relation between the input and output can be approximated by such a linear function. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_J]^\top \in \mathbb{R}^{J \times F}$ be the matrix of all the examples J and given the groundtruth vertex displacements $\mathbf{Y} = [\mathbf{D}_1^\top, \ldots, \mathbf{D}_J^\top] \in \mathbb{R}^{J \times 3N}$ where $\mathbf{D}_j$ is just the set $\{\mathbf{d}^j_{in} : n \in \{1, ..., N\}, i \in \{1, 2, 3\}\}$ vectorized into a column vector. This allows us to rewrite (9) for all N vertices simultaneously as

$$\mathbf{W}^\star = \arg\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_2^2 + \lambda \|\mathbf{W}\|^2 \tag{10}$$

where $\mathbf{W} \in \mathbb{R}^{F \times 3N}$ is the matrix of all the linear regressors, and $\| \circ \|^2$ is a regularizer that favors smaller norms (i.e. lower complexity). The above optimization problem is known as *Tikhonov regularization* or *Regularized Least Square (RLS)* [35, 4] and it has the close form solution:

$$\mathbf{W}^\star = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{Y} \tag{11}$$

with $\mathbf{I} \in \mathbb{R}^{F \times F}$ being the identity matrix.

We also extended this approach to handle non-linear functions by using the Representer Theorem [27]. Both the linear and non-linear model are able to describe the relation between the sensor signal and vertices. However, due to the higher complexity of the non-linear model (linear growth with the number of examples T), our machine learning based approach relies on the linear model.

Run-Time. At run time, given the current signal $\mathbf{x}_t$ we compute the N vertex displacements as $\mathbf{D}_t = \mathbf{W}^\top \mathbf{x}_t$. The current position of the sheet is then computed as $\mathbf{V}_t = \mathbf{V}_{t-1} + \mathbf{D}_t$. The initial position $\mathbf{V}_0$ is assumed to be known (i.e. the resting pose). The learning method we propose ensures enough robustness against drift effects, which has little impact in the final reconstruction. However, it is always possible to accurately detect the resting pose when there is no activity on the sensor: by simply computing the standard deviation σ of the signal $\mathbf{x}_t$ and classify the current pose as initial position if $\sigma < \theta$.

RESULTS AND EXPERIMENTS

We now evaluate the proposed surface reconstruction methods, whereas the core question of the experiment is to measure our system accuracy with regard to the two proposed algorithms. For this we again use our ground truth rig. We acquire 10 sequences of bending gestures covering the most common deformations for interaction scenarios. Each sequence contains approximately $4,000$ frames, in total, so our data-set is composed of $40,000$ frames. We randomly split the data-set into training and validation set, where $30,000$ examples are used for training and the remaining $10,000$ for testing.

The error (in meters) is computed using the average Euclidean distances between the ground truth vertices and the predicted configuration.

Reconstruction Error

We first compare the reconstruction performances of the linear interpolation (blendshape model) and regularized least squares (RLS). In Figure 9 (left) the average error over $10,000$ bending poses is reported, as a function of number of training examples. The learning methods achieve the best results with an average error of 0.015 ± 0.007m. Thanks to the strong correlation between the signal and the actual vertex displacement, the linear regressions are able to describe the relation between the sensor signal and the vertices. The blendshape approach, despite its simplicity also performs very well, with an average error of 0.018 ± 0.009m. Notice that around $10,000$ training frames are enough for achieving the best results: this correspond roughly to a video sequence of 5 mins. A comparison of the running times shows, that both linear interpolation and RLS have a complexity equal to $\mathcal{O}(MN)$ where MN is the grid size. Qualitative examples of the shape reconstructions are shown in Figure 8 (left).

As a second experiment we evaluated which part of the flexible sheet obtains the highest error. We show the qualitative results in Figure 9 (right). As expected most of the errors are around the corners, where the interaction is mostly occurring. RLS has a maximum and minimum error of 0.021 ± 0.013m and 0.006 ± 0.001m respectively. Linear interpolation instead has a maximum error equal to 0.032 ± 0.024m and minimum of 0.007 ± 0.015m.

Comparisons with Marker-Based Vision Systems

Although camera-based systems have certain different constraints compared to our self-contained setup (i.e. spatial resolution of the camera, depth dependency, occlusions), we feel that existing camera-based systems (e.g. FlexPad [31]) offer the greatest degree of reconstruction quality currently. Therefore, being able to reconstruct our deformations without this

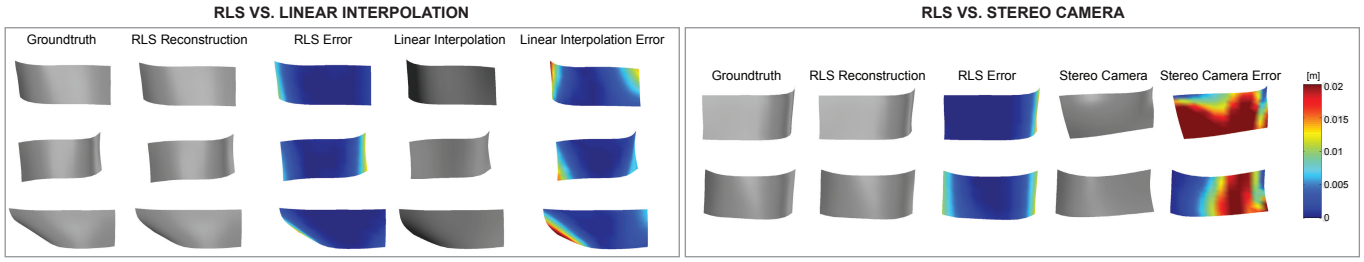

Figure 8: Left: Qualitative results showing the reconstruction performances of the linear interpolation (blendshapes) model versus the regularized linear regression of the vertex positions. Right: Reconstruction comparisons between RLS and a single stereo camera using markers. Most of the errors in the vision based system are due to occlusions.

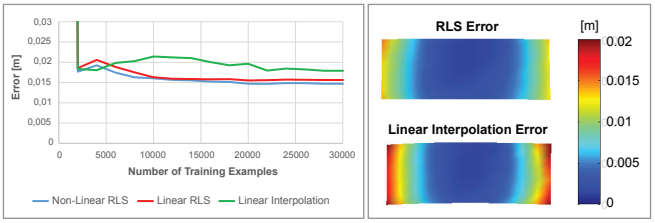

Figure 9: Left: Error [m] for the three reconstruction algorithms as a function of training samples. Right: Error maps of the linear interpolation and the regularized linear regressions with respect to the vertex positions. As expected, most of the error is located where the interactions often occurr.

type of sensing infrastructure felt like an important question to answer. For that reason, we compared our reconstructions with a single-view marker-based system. In Figure 8 (right) we show some examples where a single stereo camera is not able to reconstruct the surface of the sheet. This mainly occurs when the hands are occluding the markers and surface. On average, the single stereo marker-based system has an error of 0.023 ± 0.014m, whereas the machine learning approach achieves 0.011 ± 0.06m. This not only motivates our need for two stereo cameras for the ground truth capture, but also highlights that occlusions are a big challenge for systems that make use of single depth or regular cameras for surface tracking.

APPLICATIONS AND INTERACTION TECHNIQUES

In this section we cover some of the interactive affordances of FlexSense in two different application scenarios. These strongly highlight the benefit of the continuous, smooth and detailed level of reconstruction capabilities of our system, its transparency, and its compact and self-contained form-factor.

Transparent smart cover for tablets

In the first scenario, we used FlexSense as a replacement for existing tablet covers. By just adding a thin, transparent cover our sensor layout allows for paper-like interaction with rigid displays. Figure 10 depicts novel and natural usage examples, like switching between different layers in Photoshop or in online maps, performing rapid application switching to copy&paste content, comparing rendering enhancements for image or video filters, similar to tangible magic lens, or revealing on-demand solutions for games or education. Initial user feedback suggests that the *paper-like* feeling provides a completely novel, highly intuitive and natural interaction experience. The one-to-one mapping and directness of the input, as the user peels the foil back and forth, greatly utilizes the accuracy of the detailed reconstruction and lets the users accurately choose the area to reveal, providing direct immediate feedback.

One exciting use case for the transparent smart cover is in expanding traditional animation, where each frame is painstakingly drawn by hand. This work is mainly done with so-called 'animation desks', where animators sketch sequences on sheets of semi-transparent drafting film. These sheets are usually put on top of a light table and the animator is able to switch between the different frames by flipping the paper. Although, these light tables are being increasingly replaced by graphics tablets, many designers still use this "old style" of animation[3]. Our transparent sensor is thin enough to sense the stylus-input on the tablet. This allows the user to sketch directly on top of the transparent FlexSense and flip between different frames by bending the sensor, ushering this manual approach to animation back into digital domain.

With the use as cover of course certain practical implications arise, in particular general "wear and tear" issues of the foil, which are a general challenge of flexible devices. For later prototypes it will be necessary to think about other issues like durability, which we have not considered in this paper. However, we feel that new exciting possibilities are enabled through this configuration and inspire HCI researchers (including us) to work further in this space.

External high-DoF input device

As highlighted in related work, flexible sensors have also been used as external input devices [1]. What makes these types of sensors appealing is that they afford both 2.5D interactions, resting on a surface, as well as true in-air 3D interactions, while maintaining a level of tangible feedback (a problem for touchless interfaces in general). In Figure 10 (right) we demonstrate how the accurate reconstructions of the flexible sheet can be used in a 3D game. Continuous and discrete gestures are easily mapped to actions such as flying, steering, and firing a gun at varying rates. These gestures are simple to implement as we provide a fully tracked, temporally consistent mesh. A further example shown in Figure 10 (far right), is a physically realistic game where the reconstructions of shape and deformations, enable interactions with virtual objects (e.g. to catapult an object). The diversity of control in the examples would be difficult to achieve without the accuracy of our system and shows that continuous and precise reconstructions are crucial for these scenarios (as shown by prior work [1]). Whilst these are fun "toy" examples and exemplary show our system's capabilities, there are certain other scenarios, which could greatly benefit of the accurate reconstruction, such as 3D modeling and manipulations.

[3] `http://www.youtube.com/watch?v=fR3IDisAQwE` (from 4:50 mins)

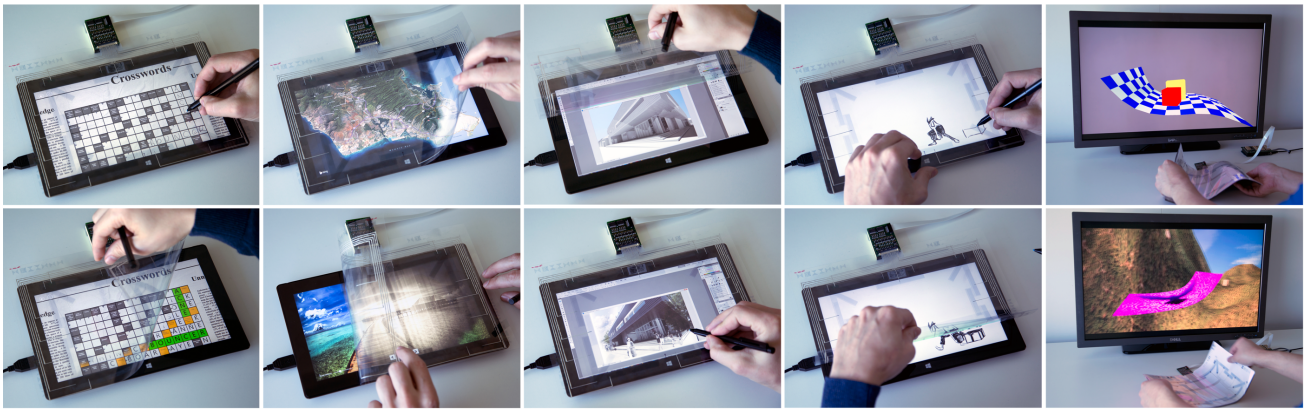

Figure 10: Example applications created using our sensor. A transparent tablet cover example, which acts as a magic lens revealing hidden metadata, applying rendering effects in photoshop, supporting window management, and allowing paper-like digital animation. Far right: using the foil as a high DoF 3D controller.

DISCUSSION & LIMITATIONS

We have discussed our sensor layout, reconstruction algorithms, and potential application scenarios. Our methods of reconstructing real-world shape and deformation from a sparse set of piezoelectric measurements are novel, and affords new interactive capabilities in self-contained lightweight form-factor. This combined with the transparent nature of the sensors creates many new application possibilities.

In describing our reconstruction algorithms, we have purposefully presented two methods, which have worked incredibly well in practice. Our linear interpolation model is relatively simple to implement, and works well even for complex models. Indeed some of the clear cover application scenarios were implemented with this model. The model can be trained and extended with new examples, in a relatively straightforward manner, and with limited samples can generate compelling results. The main need for an extended algorithm, however, comes in dealing with larger variations of shapes and deformations, and generalizing to unseen training data. For example, if we wanted to develop a 3D modeling tool using the precise surface deformation as input. However, given that the latter requires machine learning knowledge, we feel that both methods will have great value for practitioners. It is also worth noting, that our method should in theory generalize to other self sensing or external sensing setups, including other bend sensors. However, this remains future work. In terms of our algorithm, we have experimented with using ARAP during the prediction phase, at runtime. This type of regularization leads to over-smoothing, but another area of future investigation is to think about such a run-time regularizer particularly if more complex materials or geometries are to be modeled.

In terms of hardware, developing a new sensor is challenging and the FlexSense sensor does have limitations. The sensors deteriorate over prolonged periods when performing extreme gestures. Gestures such as folding are also problematic. The sensors are also not fully transparent, although indium tin oxide (ITO) used in traditional touch screens, has similar transmissive capabilities. Another limitation is, that the integrated sensor signal used in the linear interpolation approach can suffer from drift issues when performing rapid interactions.

Generally, however, the sensor is highly promising and has a lot of potentials for further projects. It will be interesting to conduct more profound user studies, particularly as new interaction techniques are developed further. Moreover, detailed comparisons of different existing sensor configurations with the FlexSense setup would be highly interesting. From a more general point of view, we would like to combine our input sensor with a flexible display (e.g. eInk / OLED). Smart watches are becoming more and more popular; especially, if they are combined with deformable shapes, we can imagine devices which have not been possible before. Another interesting area is that of touch/pressure sensing in combination with bend sensors. In particular the signal processing challenges of differentiating one from the other.

CONCLUSIONS

In this paper, we presented FlexSense, a new thin-film, transparent self-sensing surface, which can reconstruct complex deformations without the need for any external sensing, such as cameras. We have built on prior work to demonstrate a new piezoelectric bendable input device, with sensors printed on the periphery of the surface substrate. Our main contribution has been to devise a novel set of algorithms to fully reconstruct the complex deformations of the sheet, using only these sparse sensor measurements. We have demonstrated a number of new types of applications for such a device that exploit the accurate shape and deformations afforded.

ACKNOWLEDGEMENTS

We acknowledge Florian Perteneder, Cem Keskin and Barbara Stadlober for their invaluable input. The research leading to these results has received funding from the European Union, Seventh Framework Programme FP7/2007-2013 under grant agreement N^{o} 611104.

REFERENCES

1. Balakrishnan, R., Fitzmaurice, G., Kurtenbach, G., and Singh, K. Exploring Interactive Curve and Surface Manipulation Using a Bend and Twist Sensitive Input Strip. In *I3D'99*, ACM, 1999, 111–118.
2. Caglioti, V., Giusti, A., Mureddu, L., and Taddei, P. A Manipulable Vision-Based 3D Input Device for Space Curves. In *Articulated Motion and Deformable Objects*. Springer, 2008, 309–318.
3. Danisch, L. A., Englehart, K., and Trivett, A. Spatially continuous six-degrees-of-freedom position and

orientation sensor. In *Photonics East*, International Society for Optics and Photonics, 1999, 48–56.
4. Evgeniou, T., Pontil, M., and Poggio, T. Regularization Networks and Support Vector Machines. In *Advances in Computational Mathematics*, 2000.
5. Follmer, S., Leithinger, D., Olwal, A., Cheng, N., and Ishii, H. Jamming User Interfaces: Programmable Particle Stiffness and Sensing for Malleable and Shape-changing Devices. In *UIST'12*, ACM, 2012, 519–528.
6. Gallant, D. T., Seniuk, A. G., and Vertegaal, R. Towards More Paper-like Input: Flexible Input Devices for Foldable Interaction Styles. In *UIST'08*, ACM, Oct. 2008, 283.
7. Gomes, A., Nesbitt, A., and Vertegaal, R. MorePhone: A Study of Actuated Shape Deformations for Flexible Thin-Film Smartphone Notifications. In *CHI'13*, ACM, Apr. 2013, 583.
8. Herkenrath, G., Karrer, T., and Borchers, J. TWEND: Twisting and Bending as new Interaction Gesture in Mobile Devices. In *CHI'08 EA*, ACM, Apr. 2008, 3819.
9. Holman, D., Vertegaal, R., Altosaar, M., Troje, N., and Johns, D. PaperWindows: Interaction Techniques for Digital Paper. In *CHI'05*, ACM, 2005, 591–599.
10. Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica* (1976).
11. Kato, H., and Billinghurst, M. Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In *IWAR'99*, IEEE Computer Society, 1999.
12. Kato, T., Yamamoto, A., and Higuchi, T. Shape recognition using piezoelectric thin films. In *IEEE Industrial Technology*, vol. 1, IEEE, 2003, 112–116.
13. Khalilbeigi, M., Lissermann, R., Kleine, W., and Steimle, J. Foldme: Interacting with double-sided foldable displays. In *TEI'12*, ACM, 2012, 33–40.
14. Khalilbeigi, M., Lissermann, R., Mühlhäuser, M., and Steimle, J. Xpaaand: Interaction Techniques for Rollable Displays. In *CHI'11*, ACM, 2011, 2729–2732.
15. Kildal, J., Paasovaara, S., and Aaltonen, V. Kinetic Device: Designing Interactions with a Deformable Mobile Interface. In *CHI EA'12*, May 2012.
16. Konieczny, J., Shimizu, C., Meyer, G., and Colucci, D. A Handheld Flexible Display System, 2005.
17. Lahey, B., Girouard, A., Burleson, W., and Vertegaal, R. PaperPhone: Understanding the Use of Bend Gestures in Mobile Devices with Flexible Electronic Paper Displays. In *CHI'11*, ACM, May 2011, 1303.
18. Leal, A., Bowman, D., Schaefer, L., Quek, F., and Stiles, C. K. 3D Sketching Using Interactive Fabric for Tangible and Bimanual Input. In *GI'11*, Canadian Human-Computer Communications Society, 2011, 49–56.
19. Lee, J. C., Hudson, S. E., and Tse, E. Foldable interactive displays. In *UIST'08*, ACM, 2008, 287–290.
20. Lee, S.-S. et al. How Users Manipulate Deformable Displays as Input Devices. In *CHI'10*, ACM, Apr. 2010, 1647.
21. Lee, S.-S. et al. FlexRemote: Exploring the effectiveness of deformable user interface as an input device for TV. In *HCI International 2011–Posters' Extended Abstracts*. Springer, 2011, 62–65.
22. Punpongsanon, P., Iwai, D., and Sato, K. DeforMe: Projection-based Visualization of Deformable Surfaces Using Invisible Textures. In *ETech SA'13*, ACM, 2013.
23. Rendl, C. et al. PyzoFlex: Printed Piezoelectric Pressure Sensing Foil. In *UIST'12*, ACM, 2012.
24. Rosenberg, I., and Perlin, K. The UnMousePad: An Interpolating Multi-touch Force-sensing Input Pad. In *ACM Transactions on Graphics (TOG)*, vol. 28, ACM, 2009, 65.
25. Roudaut, A., Karnik, A., Löchtefeld, M., and Subramanian, S. Morphees: Toward High "Shape Resolution" in Self-Actuated Flexible Mobile Devices. In *CHI'13*, ACM, Apr. 2013, 593.
26. Sato, T., Mamiya, H., Koike, H., and Fukuchi, K. PhotoelasticTouch: Transparent Rubbery Tangible Interface Using an LCD and Photoelasticity. In *UIST'09*, ACM, 2009, 43–50.
27. Schölkopf, B., Herbrich, R., and Smola, A. A Generalized Representer Theorem. In *Conference on Computational Learning Theory*, 2001.
28. Schwesig, C., Poupyrev, I., and Mori, E. Gummi: A Bendable Computer. In *CHI'04*, ACM, Apr. 2004, 263–270.
29. Smith, R. T., Thomas, B. H., and Piekarski, W. Digital Foam Interaction Techniques for 3D Modeling. In *VRST'08*, ACM, 2008, 61–68.
30. Sorkine, O., and Alexa, M. As-rigid-as-possible surface modeling. In *SGP'07*, 2007.
31. Steimle, J., Jordt, A., and Maes, P. Flexpad: Highly Flexible Bending Interactions for Projected Handheld Displays. In *CHI'13*, ACM, Apr. 2013, 237.
32. Tajika, T., Yonezawa, T., and Mitsunaga, N. Intuitive Page-turning Interface of E-books on Flexible E-paper based on User Studies. In *MM'08*, ACM, Oct. 2008, 793.
33. Tarun, A. P. et al. PaperTab: An Electronic Paper Computer with Multiple Large Flexible Electrophoretic Displays. In *CHI EA'13*, ACM, 2013, 3131–3134.
34. Taylor, J. et al. User-specific hand modeling from monocular depth sequences. In *CVPR*, 2014.
35. Tikhonov, A., Leonov, A., and A.G., Y. Nonlinear Ill-Posed Problems. In *Kluwer Academic Publishers*, 1998.
36. Warren, K., Lo, J., Vadgama, V., and Girouard, A. Bending the Rules: Bend Gesture Classification for Flexible Displays. In *CHI'13*, ACM, Apr. 2013, 607.
37. Watanabe, J., Mochizuki, A., and Horry, Y. Bookisheet: Bendable Device for Browsing Content Using the Metaphor of Leafing Through the Pages. In *UbiComp'08*, ACM, Sept. 2008, 360.
38. Wellner, P. Interacting with paper on the DigitalDesk. *Communications of the ACM 36*, 7 (1993), 87–96.
39. Ye, Z., and Khalid, H. Cobra: Flexible Displays for Mobile Gaming Scenarios. In *CHI EA'10*, ACM, 2010, 4363–4368.
40. Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S., and Harvill, Y. A Hand Gesture Interface Device. In *CHI'87*, ACM, 1987, 189–192.
41. Zirkl, M. et al. An All-Printed Ferroelectric Active Matrix Sensor Network Based on Only Five Functional Materials Forming a Touchless Control Interface. *Advanced Materials, Volume 23, Issue 18* (2011), 2069–2074.