

MIT Open Access Articles

*SensorTape: Modular and Programmable
3D-Aware Dense Sensor Network on a Tape*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Dementyev, Artem, Hsin-Liu (Cindy) Kao, and Joseph A. Paradiso. "SensorTape: Modular and Programmable 3D-Aware Dense Sensor Network on a Tape." in Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, UIST '15, November 08-11, 2015, Charlotte, NC, USA.

As Published: <http://dx.doi.org/10.1145/2807442.2807507>

Publisher: Association for Computing Machinery (ACM)

Persistent URL: <http://hdl.handle.net/1721.1/103645>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of use: Creative Commons Attribution-Noncommercial-Share Alike



SensorTape: Modular and Programmable 3D-Aware Dense Sensor Network on a Tape

Artem Dementyev
MIT Media Lab
Cambridge, MA, USA
artemd@media.mit.edu

Hsin-Liu (Cindy) Kao
MIT Media Lab
Cambridge, MA, USA
cindykao@media.mit.edu

Joseph A. Paradiso
MIT Media Lab
Cambridge, MA, USA
joep@media.mit.edu

ABSTRACT

SensorTape is a modular and dense sensor network in a form factor of a tape. SensorTape is composed of interconnected and programmable sensor nodes on a flexible electronics substrate. Each node can sense its orientation with an inertial measurement unit, allowing deformation self-sensing of the whole tape. Also, nodes sense proximity using time-of-flight infrared. We developed network architecture to automatically determine the location of each sensor node, as SensorTape is cut and rejoined. Also, we made an intuitive graphical interface to program the tape. Our user study suggested that SensorTape enables users with different skill sets to intuitively create and program large sensor network arrays. We developed diverse applications ranging from wearables to home sensing, to show low deployment effort required by the user. We showed how SensorTape could be produced at scale using current technologies and we made a 2.3-meter long prototype.

Author Keywords

Sensor tape; cuttable; dense sensor network; self-sensing; flexible electronics

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

We envision a sensate material in the form factor of a tape. Like a roll of Scotch™ tape, it is *ubiquitous* and *inexpensive*. It is *flexible* and *cuttable* to any length and can be attached to a curved surface. Unlike Scotch tape, it has *dense arrays* of different sensors and embedded computation. The tape has inputs and outputs, and can be *programmed* and left in the environment. The tape is *modular*, as one can get different types of tape: for example, with inertial measurement units (IMUs), or proximity sensors, or even sensors directly printed on the tape as in [25, 10]. Tapes with different sensors can be joined together for mixed functionality. Such tape can do *self-sensing*, so it knows its length, sensor locations, and 3D shape.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

UIST '15, November 08-11, 2015, Charlotte, NC, USA

Copyright © 2015 ACM. ISBN 978-1-4503-3779-3/15/11...\$15.00.

DOI: <http://dx.doi.org/10.1145/2807442.2807507>



Figure 1. a) Current flexible circuit prototype, showing electronics of two nodes. Each node has a microcontroller, inertial measurement unit (IMU), proximity and light sensors. b) SensorTape can be cut anywhere and rejoined. We develop algorithms to automatically determine the order and the number of nodes. c) Form factor of SensorTape is of 1-inch wide masking tape. d) SensorTape sensing its deformations with IMUs e) The 2.3 meter long testbed with 66 sensor nodes. Proximity sensors are visualized on the laptop, while hovering foot over the tape.

The abilities of sensors to deal with complex phenomena are greatly increased if they are in dense arrays. For example, an array of capacitive sensors can track finger on a touchpad or an array of pressure sensors can detect walking patterns in a room or make an electronic drum. Such sensor arrays have many applications ranging from engineering to arts, but they are out of reach for most users. To create them using current electronics platforms, many discrete modules have to be physically wired, and a custom communication network has to be created. This is a time consuming task, and does not allow easy experimentation, as the network has to be manually reprogrammed and rewired when it changes.

A sensor network in the shape of a tape allows for freedom of experimentation. Since tapes and ribbons are familiar everyday materials: the manipulations with the tape are *intuitive*. Actions such as cutting and rejoining have direct digital

metaphors. The tape form factor yields well to the current flexible electronics manufacturing methods. Also, it enables precise localization of all sensors, since all the nodes are in a linear array. Sensor nodes can be placed on single communication and power buses, thus removing the need for multiple batteries and wireless communications.

Current electronics technologies are sufficiently mature for such a tape to be made, especially as flexible electronics are becoming more affordable and prevalent. For example, we have witnessed emergence of usable and inexpensive LED strips e.g. [4]. But so far, LED strips provide only output, and do not have distributed sensing or localization. As electronics are becoming less power hungry, more powerful, and affordable, it is possible to add such capacities. Even before the emergence of such systems, we need to understand their potential capacities and how to design them. To do so, we developed the SensorTape platform which enables anyone to experiment with such a sensate strips. In our implementation, we place programmable and interconnected sensor node modules on a flexible substrate, as shown in Figure 1a, and in Figure 4 in detail.

The contributions of the paper are as follows:

1. We introduce a novel sensor network platform in the form factor of a tape. Such a platform enables fast prototyping and deployment of continuous sensors, as it is modular and can be cut or joined.
2. We develop a prototype 2.3m long (66 nodes), and a novel self-configurable network architecture. We explain how SensorTape can be manufactured in large quantities.
3. We develop algorithms to demonstrate that SensorTape can be used for deformation self-sensing (e.g. bend and twist), and measure the accuracy.
4. We evaluate a number of parameters such as cost, speed, power, and maximum length. We conduct user studies to understand the usability and possible applications.
5. We explore ways to program such a tape, and implement four applications to explore use cases.

DESIGN CONSIDERATION

The sensate tape should meet the following design considerations:

1. Employs tape form-factor: The SensorTape should inherit physical properties of masking tape: it should be *cuttable*, *flexible*, and *thin*. We are inspired by masking tape because it is commonly used in craft applications, and thus familiar to most users. Masking tape is shown in Figure 1a.

2. Easily configurable: We envision that the tape should be *intuitive* enough to be manipulated as a traditional craft material. When cutting and joining, the tape should be *self-configurable*, and know the total number of sensor nodes and the location of each one. Furthermore, the user should be able to connect the pieces of tape at different angles, for example to create a rectangular surface. To enable direct user feedback, the tape should be *programmable* and have input

and output. A new user interface (UI) should be used to easily program the input and output. The UI can be graphical, leveraging the intuitive tape form factor.

3. High usability: The tape should encourage user's experimentation. The tape should be able to support a large number of sensor nodes without significant bandwidth and power compromises. As a core sensing modality, the tape should sense *proximity* to physical objects and its own *bend* and *twist*. Such modalities allow for real-time and tangible human interactions, which facilitate more user experimentation. The tape should be *modular* to allow addition of nodes with other sensors. To facilitate physical modifications, the tape should be *inexpensive*. To reduce the cost, it has to yield well to the current manufacturing methods. Communication protocol should be *robust* and able to bypass a broken node. Also, the tape should be physically durable, and able to flex easily and repeatedly.

PREVIOUS WORK

We are not aware of any projects that explore the tape form factor to make a modular sensor network. LED strips are cuttable and employ the same form factor, but do not have distributed computation or sensing capacities. This project takes the LED strip concept further to create a modular sensor network. Popular electronics platforms (e.g. Arduino) are mostly designed to build discrete devices. Networking 100s of them is beyond the skill and the time frame of many users. Even current modular electronics platforms are not flexible or cuttable and do not have deformation sensing. They are mostly composed of rigid blocks that snap together. This work draws from four areas, that we categorize below:

Customizable flexible sensors. The technologies [14, 1] to inexpensively and quickly prototype flexible electronics created a substantial interest in customizable arrays of flexible sensors. Using paper-like printed circuits, cuttable capacitive and resistive sensors were demonstrated in [9, 20, 10, 12] for HCI applications. The work in customizable sensors is mostly concerned with customization of discrete modules.

Dense sensor networks. Most of the research in sensor networks was done for wireless networks such as Smart Dust [13] or Berkeley's Mica platform [11]. Wired networks received less attention. The ChainMail [19] is a sensate-skin sensor network. It is made of rigid circuit board nodes, networked with wires. The system does not have deformation sensing ability. Pushpin [17] is a platform for prototyping sensor networks. To provide power each node is pressed into a multilayer foam-like substrate. Nodes communicate to each other with infrared signals and can estimate their location.

Substantial work focused on embedding arrays of sensors into a floor to detect footsteps. For example, Z-tiles [24] used modules with pressure sensors, that can network and self-localize. Another work [8] used networked capacitive sensing nodes that were placed on a flexible printed substrate. The sensor networks above are mostly application specific (e.g., floor sensing) and do not allow customization and general-purpose use.

Self-sensing surfaces. There has been attempts to create deformation self-sensing surfaces using piezoelectric sensors [23], resistive bend sensors [15] and capacitance [9]. ShapeTape (Measurand) is a tape that can sense its bend and twist, using an array of bundled fiber optic sensors spaced at 6 cm. ShapeTape is useful for 3D curve mapping [6], but it is not cuttable or programmable. Many projects in this space use machine learning; therefore require retraining if the number of sensing elements changes. Our approach is purely algorithmic, therefore is scalable to any size. Commercial IMU-based systems (e.g., Xsens MVN, MotionNode) for 3D motion tracking are not customizable, bulky, and expensive.

Modular electronics platforms. There has been great interest in creating modular electronics platforms [7, 18, 22, 26], where one can construct a complex system from modular building blocks. In platforms such as Syftio [18] and Littlebits [7] the rigid modules are snapped together with magnets, and communicate through radios or a common bus. Furthermore, in the Circuit Stickers platform [21], flexible circuit modules are attached to paper with copper tape. Seeduino Film [3] is an Arduino-based platform composed of rigid-flex PCBs that can be linked together. We are not aware of any platforms that leverage the flexible tape form factor, or achieve self-sensing.

NETWORK ARCHITECTURE

Overview: Our architecture is outlined in Figure 2. The network has one master, and all the sensor nodes on the tape are slaves. The master is concerned with coordinating the communication and shuttling data to the PC. Each slave contains a microcontroller (MCU), LEDs and sensors. We used the I^2C (Inter-Integrated Circuit) communication bus running at 100 kHz to talk between the master and slaves. In I^2C slaves and the master are daisy chained by clock and data wires. Slaves can receive and send data. I^2C is often used to talk between chips, and was used in some previous work [19, 8, 24].

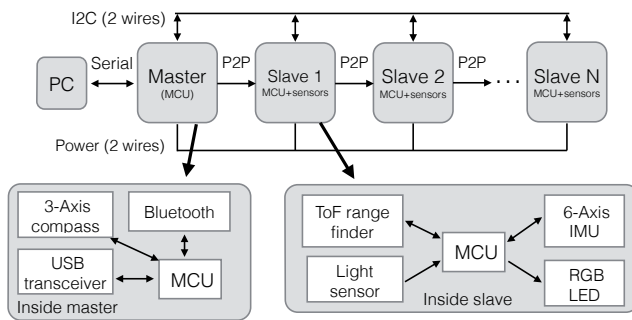


Figure 2. The network is composed of daisy chained slave nodes and one master. Each node is connected to the next by 5 wires: clock and data for I^2C , peer-to-peer serial, 5V power and ground.

We were mainly concerned with making the network robust, fast and self-configurable. Although a sequential peer-to-peer (P2P) network is available, we avoided using this topology for data communications. With the daisy chain bus topology, if one node goes down, others continue to function. Also, the data rates can be faster since data does not have to hop

through multiple nodes. Specifically we chose I^2C bus because the microcontroller has a hardware I^2C module, which makes it more reliable, since it has a memory buffer and generates accurate clock. Furthermore, I^2C only uses two wires, and keeping the number of wires between the nodes to the minimum makes it easier to join pieces of the tape. 1-Wire protocol (Dallas) requires only one wire for communications, but was avoided due to lack of dedicated hardware and the protocol’s proprietary standard.

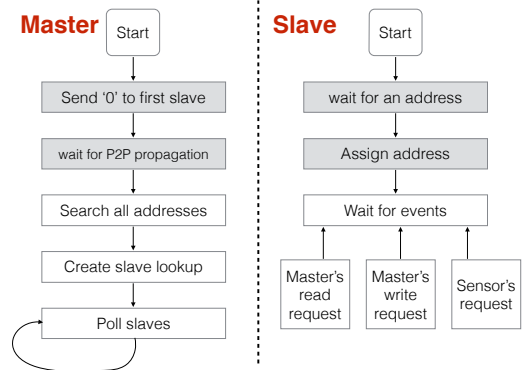


Figure 3. Communication flowcharts for the master and a slave. The events in gray boxes happen over peer-to-peer communications. Later events happen over I^2C .

Communication protocol: The communications are outlined in Figure 3. Each slave has a unique address of 7 bits, allowing 128 total devices. Since each slave needs to have a unique address and the network size can change, the addresses are assigned dynamically. We used the peer-to-peer serial communication to assign the addresses. The master initiates communication by sending 0 to the first slave. The first slave address is assigned to 0, and the first slave sends 1 to the second slave. This hopping repeats until the last slave is reached.

After slave address assignment, the master does not know how many slaves are present, so it sends a command to all possible addresses from 0 to 127. If a slave is present, it will reply with the number and type of sensors it has. That allows the master to create a directory of all present slaves. After the initiation, the master polls the slaves in a consecutive order at 20 ms intervals. Data is immediately sent to the PC over a serial link. Slaves can receive commands from the master, which we use to change slave’s LED colors. The slave code is interrupt (event) driven, as it reacts to requests from the master or its IMU or proximity sensor. Such sensors generate an interrupt when new data is ready.

PROTOTYPES

We constructed two prototypes (Figure 4), each with unique advantages and disadvantages. Both have the same software and electronics, and only differ in the construction techniques. Both work with the same master node, which can be clipped or soldered to the tape. The conductive inkjet prototype 1 is easier to experiment with and reproduce by researchers and the DIY community. The flexible circuit prototype 2 uses the same technology as LED strips. It fits better

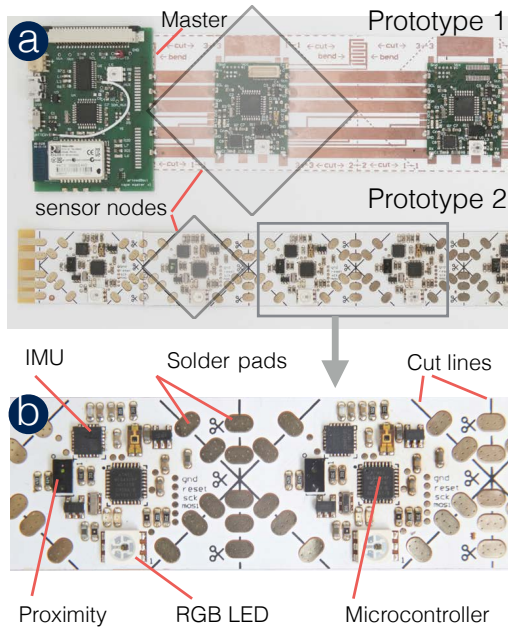


Figure 4. (a) The conductive inkjet prototype 1 (top) and the flexible circuit prototype 2 (bottom) are shown next to each other. The electronics are exactly the same in both prototypes. Master node is connected to prototype 1. The electronics of each node fit into diamond shaped cells, which are shaded in gray. (b) Close up of the electronics in prototype 2, where solder pads on the cut lines allow easy reconnection. The solder pads expose the five global tracks that run on the backside.

with our future vision of the sensate tape. The prototype 2 is more fit for mass production, more robust, and has better performance, but requires more expensive and longer manufacturing process.

Prototype 1: conductive inkjet technology

We first used conductive inkjet technology [1] to make the flexible substrate. In this technology copper is electroplated on flexible polyester film, and etched using inkjet-printed mask. Similar technique was used in our other research projects [9, 20, 10, 8]. Sensor nodes were made from rigid circuit boards and soldered to the flexible substrate. Nodes were spaced at 7cm. We could not attach components directly to the substrate, because it only allows one layer, and does not have enough resolution for fine-pitched chips, such as the proximity sensor.

Prototype 2: Flexible Printed Circuit

The second prototype was made using flexible printed circuit (FPC) technology. We used a 2-layer polyimide-based substrate. The electronics were directly soldered onto the FPC. The nodes were spaced at 3.5 cm, and tape was 1-inch wide. We applied a white coverlay to make the tape more aesthetically appealing, and make the cut lines more visible. The numerous solder pads on both sides of the tape allowed easy reconnection of the five tracks that run on the backside. We used a South China based circuit manufacturer, since the costs for flexible circuit technology in the USA and Europe are unfeasibly high for many research budgets. Also, we were able

to get ready advice, since the manufacturer had a lot of experience manufacturing LED strips. We employed an assembly service to solder the components. The components can be soldered by hand using an inexpensive reflow oven. We would not recommend such a path, as it is time consuming and error-prone to solder small components on a flexible circuit board. We achieved only 20% yield with hand soldering.

Cuttable architecture

Using three cuts: vertical (90°), diagonal left to right (45°) and diagonal right to left (135°), allows for versatile arrangements. For example 45° and 135° cuts put together will create a 90° degree bend, as can be seen in Figure 6. Soldering the pieces together is the most reliable, while attaching them with conductive Z-axis tape (3M) is the fastest for rapid prototyping. For ease, we labeled the cut lines on the tape. Electronics were put in the diamond shaped cells between the cuts, as seen graphically in Figure 6, and physically in Figure 4b.

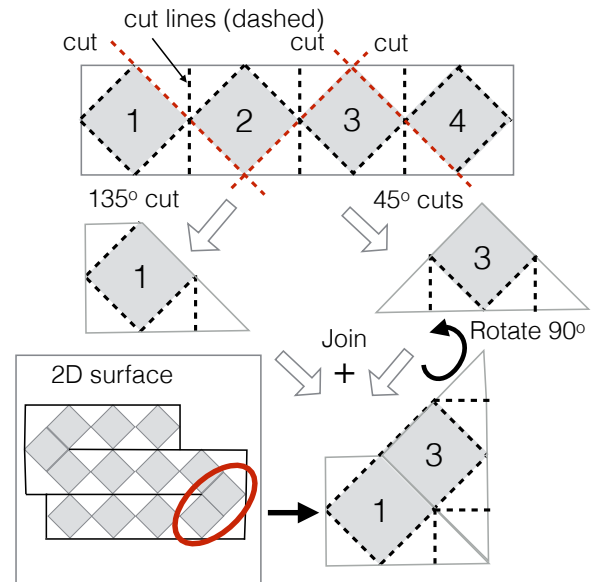


Figure 6. The tape is layed out based on diamond shaped cells. Each node is shown in shaded gray. Diamonds allow 45° , 90° and 135° cuts. In this figure we show how the tape can be cut and rejoined to make a 90° bend. Using such bends, the tape can be made into a 2D surface.

Components

One sensor node is composed of following parts:

- 1. Microcontroller** Each node has an ATmega328p (Atmel) microcontroller (MCU). We chose this microcontroller because it is compatible with the Arduino platform, which has a large support base in the DIY community and an easy-to-use programming interface. We wanted the users to be able to easily program the sensor nodes.
- 2. Sensors:** There are three sensors on-board each node. We added as many sensors as possible, to explore various use cases: 1) TEMA6000 (Visday) ambient light sensor. 2) A MPU6050 (Invensense), which is a three-axis accelerometer and three-axis gyro chip. It includes integrated motion processor, so the chip outputs orientation data in quaternions format. We used this feature to sense deformations. 3), VL6180

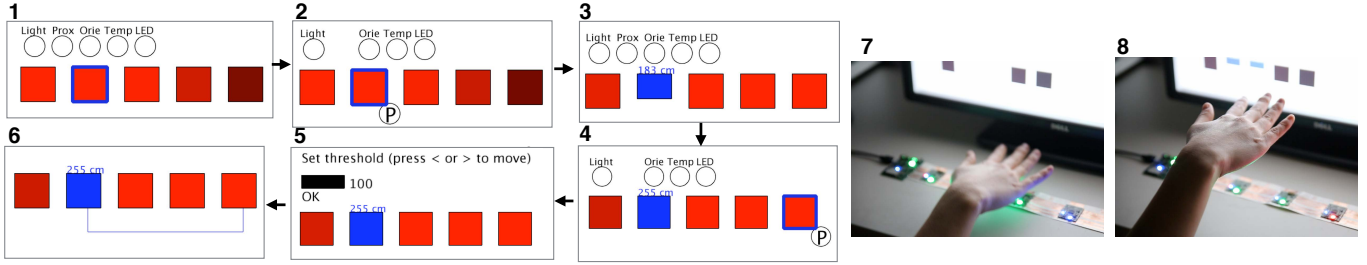


Figure 5. Example interaction using the graphical programming interface. 1) Tape with each sensor node shown as a square. Square brightness changes based on light. Clicking on one of the squares shows available sensors as circles on top. 2) The user can drag and drop the selected sensor. 3) Dragging to the same square, visualizes that sensor. 4) Dragging to another square creates a relation between two sensors. 5) User is asked to set threshold for the sensor. 6) The relationship is shown by the line on the bottom. 7) By moving hand close to the sensor, LED becomes blue. 8) Red when above threshold.

(STMicroelectronics), Time-of-flight (ToF) distance sensor, which measures the time infrared light takes to bounce from an object. The proximity range is up to 20cm, and it is not influenced by the color and type of the surface. We did not see any interference with neighboring proximity sensors.

3. Miscellaneous: Each node has a voltage regulator. It regulates the input voltage that can range from 5V to 3.5V to a stable 3.3V. A second voltage regular provides a stable 2.8V supply for the proximity sensor. Furthermore, nodes contain an LED (WS2812S, Worldsemi) that can output red, green and blue. The LED can be controlled from the master and provides visual feedback to the user.

4. Master node: The master node contains the following parts: The same ATmega328p, a serial-to-USB converter, and a Bluetooth transceiver. The tape can be clipped to the master node without soldering using a flexible circuit connector.

DEFORMATION SELF-SENSING

An important feature of the tape is that it can perform self-sensing. In this section, we will describe the algorithm we developed to sense the tape’s position in space. The algorithm maps the orientation data from IMUs to a kinematic chain model and deformations of 3D surface. To create a reference frame and to remove gravity we convert original quaternions into roll pitch, and yaw. We breakdown all deformations into bending (movement in XY-plane) and twisting (YZ-plane), which can be analyzed independently. The geometrical representation is shown in Figure 7. Our approach allows scaling to arbitrary lengths, without retraining, as needed in many machine learning approaches. Initially, we experimented with printed capacitive sensors to detect bend, but we did not find them reliable.

First, lets look at bending. Here we had to make a key assumption that the distance (h) between the nodes is fixed. This allowed us to create a simple kinematic chain, where each node is a joint. The X-Y position of each node is determined by the position of the previous node, and trigonometric relations between the pitch angle (τ_n) and the distance h .

Twisting is determined by the rotation in the Z-axis. Here we assume that the width (w) of the tape remains the same. As with bending, the Y-Z position is determined by geometric relationships between the width and the roll angle (θ_n).

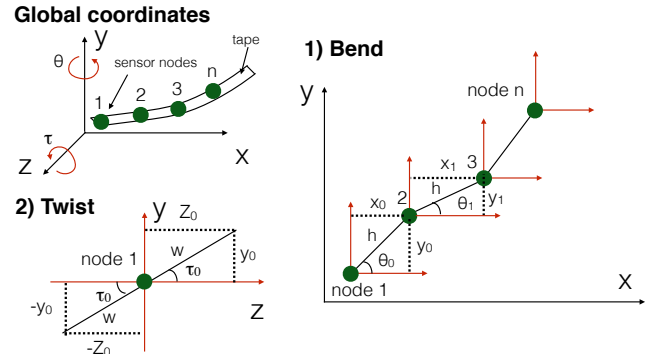


Figure 7. The graphical representation of bending and twisting calculations. The bending only happens in the X-Y plane and twisting in the Y-Z plane.

The x, y, z coordinates for each node n can be mapped as shown below. The distance between the IMUs is h , and width of the tape is w , τ_n is rotation in Z-axis, and θ_n is rotation in Y-axis

$$x_n = x_{n-1} + h \cos(\theta_n) \quad (1)$$

$$y_n = y_{n-1} + h \sin(\theta_n) + w \sin(\tau_n) \quad (2)$$

$$z_n = w \cos(\tau_n) \quad (3)$$

PROGRAMMABILITY

To achieve low deployment effort by the end user, we created framework that abstracts the basic protocol and physical layers. For the end user, programming SensorTape does not require any hardware or firmware modifications of the master or the nodes. The configuration and communications on the tape happen on their own. We built a Java library that automatically creates and updates the nodes on the PC. Any high-level programming language can be used as PC only sees simple serial data from the master. Our library allows the user to build more complex applications, without worrying about communications and housekeeping of the nodes. For example, it allows easy visualization of the nodes, or computation of the 3D surfaces.

Since using our Java library still require programming skills, we developed a purely graphical UI to program and visualize the tape. Using drag and drop interface, it allows simple but limited operations. We show an example interaction in

Figure 5. This UI leverages the fact that all the nodes can be visualized as squares on a screen, that directly corresponds to their physical layout. Clicking on an individual square (node) brings up its available sensors represented by circles above the node. The circles can be dragged and dropped in two ways. First, dropping in the same node changes the visualization to that sensor. Second, dropping in a different square creates a relation between the sensor and the node where the sensor was dropped. For example, a user can link the light sensor on the first node to the LED on the second, so the LED color changes based on the light sensor of the first node.

TECHNICAL CONSIDERATIONS

Cost

Cost is an important consideration, since SensorTape involves a large number of electronic components, and has more value at a large scale. The price of electronics is the major cost: one node with the IMU is about \$5 (US Dollars). Fabrication of the flex circuit for one node is about \$1.6. So, the price is about \$100 per meter. The price of electronic components was about half of the cost of the large electronics distributors (e.g. DigiKey), because we ordered in quantity of 100 from a China-based supplier, found on www.alibaba.com. On the downside, we spent a lot of time communicating with the suppliers, as there were language barriers. The proximity sensor we are using is novel, and therefore costs almost as much as all other parts (\$4), increasing the price to \$200 per meter. It is possible to bring the cost down by integrating the computation and sensing functions into a single chip. Also, it may make economical sense to have different tapes, each with just proximity or IMU, as users might need only one sensor.

Power

The power consumption of each node is 22 mA (76mW). This is the maximum power consumption, as it was measured with all sensors turned on and operating in real-time. The power can be greatly reduced if the nodes are put to sleep when they are not performing a task.

Communications

The throughput of the network is 6.6 kbps, as the master gets 17 bytes from the slave nodes every 20ms (about 50Hz). The throughput is spread over the nodes and is the same regardless of the number of nodes. With an increased number of nodes, data from each node is requested less often. For example, with one node, sampling is 50Hz, but with 10 nodes, each will be sampled slower at 5 Hz.

The tape startup process takes 3.5 seconds; mostly taken up by the IMU's initialization. The master takes about 700ms to find who is in the network. In the current version, the network has to be power cycled after cutting or rejoining, as the number of nodes is only determined at startup. It is possible to make the network change without restarting, by making the master periodically determine the number of nodes.

Maximum length

It is important to understand how many sensor nodes can be supported by the tape. As seen in Figure 1e, the flexible

circuit prototype worked with 66 nodes. We could not add more, since this was the number of nodes we manufactured. To understand impact of more nodes, we conducted electrical simulations as shown in Figure 8. Our analysis suggests that the maximum length is 15 nodes (1.1 meter) for the conductive inkjet prototype, which is limited by the resistance of the power traces. The maximum length of the flexible circuit prototype is limited by the parasitic capacitance to 108 nodes (3.8 meter). We discuss the three main limitations below:

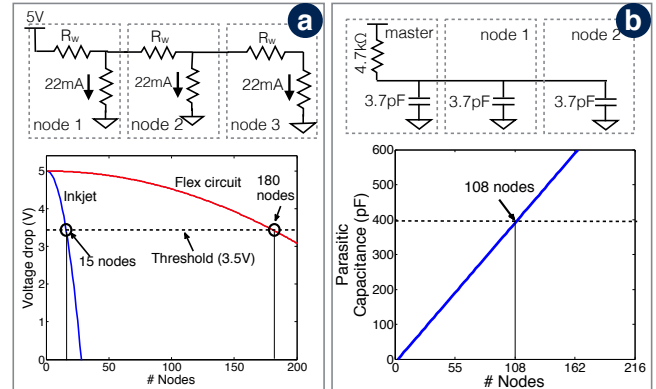


Figure 8. a) Example electrical model and the results of the voltage drop estimation. Below the 3.5V threshold, nodes cannot function. b) Example electrical model and the results of the parasitic capacitance estimation. Above a threshold of 400pF, the capacitance will interfere with communications.

First limitation is the voltage drop caused by the resistance of the traces. More nodes will increase the current the traces are carrying, thus the voltage drop becomes higher ($V=IR$). The nodes are powered by 3.3V and the power rail is 5V. The rail voltage can drop by 1.5 V before nodes stop working. The resistance of the traces is the largest limitation for conductive inkjet prototype. The resistance (R_w) between the nodes is 0.6Ω , as measured by a multimeter. In contrast, the flexible circuit prototype has resistance of $4.4 m\Omega$, which is 136 times less. As a result, the voltage will drop to 3.5 V with 15 nodes for the conductive inkjet prototype and with 180 nodes with the flexible circuit prototype. The best way to minimize the trace resistance is to increase the width of the power trace.

Second limitation is the capacitance of the I^2C bus wires. As the wires get longer, they pick up parasitic capacitance. Each connected node adds about $3.7 pF$ of capacitance. The capacitance was derived from the RC constant, where R is the $4.7 k\Omega$ I^2C pull-up resistor. The RC constant was measured directly by probing rise time of the signals. If the capacitance becomes too large, the I^2C signals will rise and fall too slowly, thus disturbing the timing. The I^2C bus can support capacitance up to $400 pF$ [2]. As a result, maximum of 108 nodes can be connected. This is the main limitation of the flexible circuit prototype, but can be alleviated by switching to a differential signaling communications.

Last limitation is the I^2C address space. Since each node on the bus needs a unique address, current 7-bit addressing allows 128 nodes. The address space can be increased programmatically by writing a custom communications protocol.

Deformation sensing

For the ground truth, we printed reference angles from 0° to 165° in 15° increments on a cardboard, and placed it next to the tape. To get the error, we bent the tape to those angles, and compared the detected angles with the reference angles. The mean error was 6.4° . The error can be compensated for as it increased with the angle as shown in Figure 9.

With MEMS gyroscopes, drift can be a problem. We did not see any noticeable drift for two hours of continuous operation, despite the lack of a magnetometer. The drift compensation is done on-chip using data from an accelerometer.

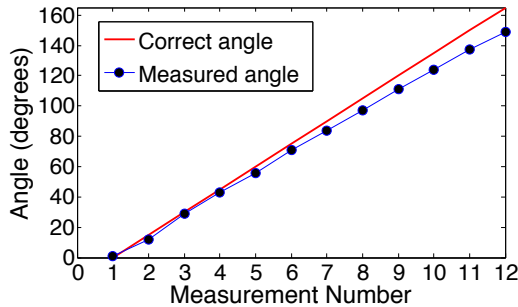


Figure 9. Error between the reported angle and true angle, as measured by one node.

Physical robustness

In the first prototype, the most common problem was delimitation of the copper from the polyester substrate. This happened at the solder joints with rigid PCBs, due to mechanical stress caused by different material stiffness. We did not experience any copper or component delamination in the second prototype; we continuously bent the tape for deformation sensing and could roll it into a spool. Damage might happen if the bend angle is large (e.g., right angle) or tape is bent repeatedly over long time.

USER STUDY

Protocol

To understand the user experience and expressivity of SensorTape, we conducted an informal first-use study with 7 participants (3 female, 4 male). Ages ranged from 23 to 32 (mean 25.5). We envision that SensorTape will be used by people with varying skill levels and backgrounds. So, we purposely selected diverse participants: Engineers, Artists, Designers and Biologists, with different expertise in electronics, programming and fabrication. Each study took 40 minutes, with three phases:

(1) Study 1: Programming interface

This study tests the programming user interface, and the mapping between on-screen visualization to the physical tape. We first demonstrated the basic UI functions: menu selection and making connections to the participants. They were then instructed to change the displayed visualization from the menus, and to make connections between sensor nodes.

(2) Study 2: cuttable form factor

This study tests the physical manipulations of the tape. We first explained how SensorTape could be cut, reoriented and reconnected with Z-tape or solder. Participants were then given the task of making a 90° angle connection with a strip of SensorTape, and to first connect it with Z-tape, then with solder.

(3) Post-study interview

We elicited qualitative feedback towards both the programming interface and the tape form factor. Participants were given a longer roll of SensorTape to play with and imagine how they would use it in everyday life. This feedback assisted us in the design of applications for SensorTape.

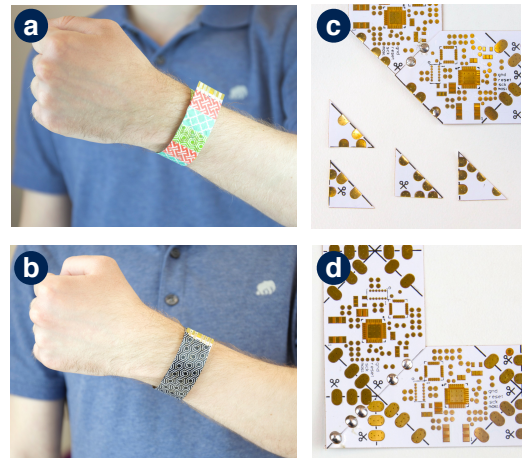


Figure 10. Results from the user study: a) and b) using the SensorTape as a bracelet with customized tape cover, as suggested by some users. c) Incorrectly soldered 90° angle from one participant, as only two of the five traces are connected. d) Correctly soldered 90° angle.

Study results

Programming interface

Overall, participants found the interface to be intuitive. However, making connections between sensor nodes was confusing for participants from non-engineering backgrounds; they expressed the need for additional UI cues, such as arrows to show directionality, color coding, or additional text. Furthermore, while input sensors (e.g., IMU) required visualization, output devices (LED) were directly visible on the tape. A few were confused by this physical/digital separation, and suggested more explicit UI guides. We observe the need for more discreet feedback for participants with limited engineering experience.

Cutting

Participants found the tape, in its form of a flexible circuit, to be easy to cut. However, several made suggestions on improving the guides, such as using dotted lines as in origami, and color-coding to separate diagonal versus straight lines. A few participants ($N=3$), confused by the mono-colored guides, cut out small triangles that could not be used. Figure 10c,d shows correctly and incorrectly joined pieces of tape from the user study.

Reconnecting

Participants were instructed to first reconnect SensorTape with the Z-tape, then to remove the Z-tape and reconnect by soldering. In general, participants found Z-tape to be easy to use, yet were concerned about flimsy connections. Almost all participants (N=6) had trouble understanding how Z-tape worked, using it as they would normal tape. Soldering resulted in secure connections, yet it was time consuming, and especially challenging for participants from non-engineering backgrounds. One participant suggested replacing the pads with holes, and using wire or conductive thread to tie the holes together to enable different angles. Others suggested a standalone connector tape module, or tiny, conductive alligator clips for connection. We observe the need for a robust reconnection method that is accessible to participants from various backgrounds.

Form factor

Many participants compared the form factor of SensorTape to a ribbon. They preferred a width of half an inch, comparable to that of tapes and ribbons used in arts and crafts.

Applications

Participants wanted to use SensorTape for an array of wearable applications, such as sports and injury monitoring, gesture tracking, and bio-sensing. The wrist and arm were the most popular locations, as they are easily accessible and affords various gestures. Beyond miniaturizing the tape, several (N=4) participants wanted to personalize the appearance of SensorTape by attaching a cover layer. This can be easily done by attaching a layer of fabric, or colored masking tape on the SensorTape surface. Some of the suggestions are shown in Figure 10a,b. Other suggested applications include inventory tracking, home activity sensing, etc.

APPLICATION SCENARIOS

In each application scenario we will highlight two or more core properties of SensorTape: cutting and joining two tapes straight and at 90° angles, bending, and sensing shape and proximity. Applications emphasize low deployment effort that can be achieved by the end user. Applications did not require any hardware modifications except cutting and joining, and did not need any node firmware changes. The applications only required programming visualizations on the PC; the basic functions such as finding and updating the nodes was done automatically.

Shape-measuring tool. Core properties explored: *joining*, *bending*, and *shape sensing*. In this application, SensorTape is used to digitize curves of physical objects. Tape can be made longer or shorter with minimum effort to enable a custom shape-measuring tool. As shown in Figure 11, we use a small six-node piece of tape to measure radius of a glass bottle; radius is estimated from the tape's curvature. The small piece of tape is too short to record the profile of the bottle. Adding six more nodes enabled the user to capture the profile of the bottle. This requires little time, and no programming as the new piece is automatically added to the 3D surface. This high-dimensional input tool is similar in spirit to

ShapeTape [6] and HandSCAPE [16], but offers more customization. The 3D surfaces are controlled directly by moving control points of a NURBS surface in OpenGL.

Wearable for posture monitoring. Core properties explored: *cutting*, *bending*, and *shape sensing*. In this application we use deformation sensing algorithms to monitor posture. Sensing is done by a piece of tape attached to the user's back. The tape is powered by a battery and data is sent to the phone over Bluetooth to alert if the user has bad posture. Because of flexibility issues, most modular electronics platforms are not comfortable to be used as wearable devices. SensorTape is flexible, and it can be cut to the right size to be comfortably worn by the user. SensorTape allows quick prototyping and deployment of wearable sensors. As shown in Figure 11, setup step only involves cutting a roll of SensorTape to custom clothing. By placing separate pieces of tape on the arms, legs, and torso, it is possible to do body motion tracking.

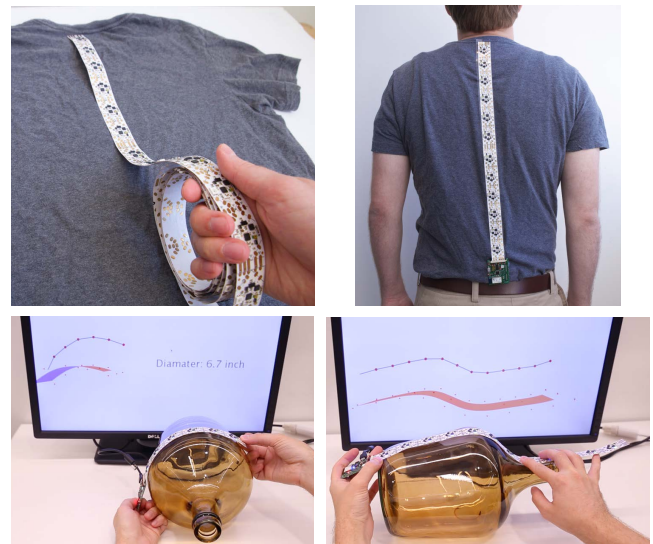


Figure 11. Applications using self-sensing. Top: SensorTape can be attached to a clothing and tailored to custom length. Attached to the back, the tape is used to monitor posture. Bottom: the tape is used as a custom 3D ruler. A smaller piece of tape is used to measure the diameter of a glass bottle, and a larger piece can measure the profile of the bottle.

Height measurement. Core properties: *cutting* and *proximity sensing*. We attached SensorTape vertically to the wall to measure the height of the person standing next to it. It is not currently possible to digitally measure height without a bulky and expensive height gage. Furthermore, it is not possible to easily customize the size and shape of such gages. We used proximity sensors, and interpolated the information from neighboring sensors to obtain 1 cm resolution. Linear arrays of sensors are well suited for height and displacement measurements, but are hard to implement quickly using current discrete electronics platforms. Implementing this application with the SensorTape took under 1 hour. Physical setup required only unrolling and cutting the a roll of SensorTape to a custom wall size. This application can be extended to many home sensing applications for presence/movement detection.

Interactive surface. Core properties: *joining at 90°* and *proximity sensing*. We created a coarse proximity-based touch screen by placing SensorTape around the edges of a rectangular screen. A cursor on the screen can be controlled by simultaneously hovering the hands over horizontal and vertical edges. The size of the cursor can be controlled by the height of the hands. The rectangular tape was created by cutting and jointing the segments at 90° angles. The deployment effort is significantly lower than that of prior proximity-aware touch screen tables [5]. With SensorTape no firmware modifications or hardware wiring was required, and the touchscreen UI was built fully in high-level Java. SensorTape allows quick prototyping of complex user interfaces with sensor arrays.

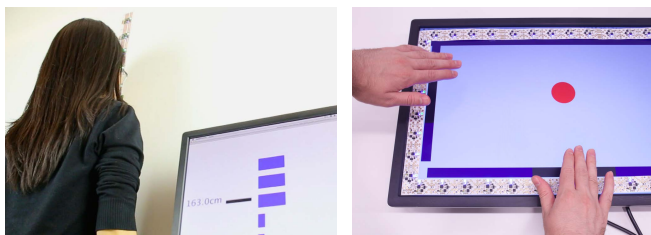


Figure 12. Applications using proximity sensing. Left: SensorTape is attached to the wall for height sensing. The proximity sensors can estimate the height of the person standing next to the tape. Right: by using proximity sensors, a cursor can be controlled by hovering hands over horizontal and vertical edges.

LIMITATIONS AND FUTURE WORK

Communications: I^2C is designed for short distances, limiting the maximum length and the speed. Also, I^2C can stop working when the signal wires are touched, as humans have large capacitance. In the future we plan to use a differential signaling protocol, such as RS-485. This can be accomplished by using a serial to RS-485 converter such as MAX481 (Maxim) at each node. Differential signals are robust to parasitic capacitance, and allow for greater data rates.

The communications are not fully protected against node failures; the peer-to-peer network requires each node to be operational. The address assignment will fail for all nodes downstream of the broken node. In future work, we will create a protocol that combines a global I^2C bus and a local P2P network to bypass broken nodes. By assigning a random address at the start, the master can talk to nodes beyond a broken link.

Programming: For the platform to become more customizable, programming will need to be simplified on multiple levels. Currently the firmware on slave nodes is loaded by directly connecting a programmer. If the user updates the firmware, this can take a long time, as the number of slaves can be large. In the future we plan to write a bootloader that can load the software through the I^2C bus.

If someone attempts to make their own version of the tape, it might take a lot of effort to write firmware for new digital sensors. Nodes are not running an operating system, so the core functionality is not abstracted from the sensors. Since the sensor nodes are driven by interrupts from the master and the sensors, a simple event-driven node operating system will greatly simplify development.

Usability: In the future work we will create a more intuitive interface. We found from the user study that both the UI and the physical tape caused some confusion. Specifically, in the UI the participants found linking the sensors to be confusing. In the future we will label the inputs and the outputs clearly by arrows, and display the type of relationship. In the physical tape, the cut lines were not clear, so many participants cut out small rectangles, making the tape hard to connect. We will provide more visual cues such as colored lines and text, so the users only cut diagonal and vertical lines. Many participants found it hard to reconnect the tape because solder pads are too small. In the future, we will increase the size of the pads and reduce their number from 5 to 4. The peer-to-peer line can be removed by switching one of the bus wires as peer-to-peer during initialization.

We only conducted limited user testing of the tape. To gain a greater understanding of advantages and disadvantages of SensorTape, a larger workshop-based evaluation is required. To do so, we will need to make more pieces of tape, and streamline the manufacturing process. Also, we released design files at github.com/ResEnv, so the users can modify and build their own versions.

CONCLUSION

In this work we presented SensorTape, a sensor network on a tape. It allows users to intuitively create and program large sensor network arrays. This will enable applications, that are not possible with the current discrete electronics platforms. As with normal tape, SensorTape is flexible and cuttable, as it is made from flexible electronics. The SensorTape can be cut and rejoined to form a linear or a 2D surface. Furthermore, it is modular, as SensorTapes with different functionalities can be joined together. We developed the network architecture that automatically determines the length of the tape and the locations of all sensors. We developed algorithms to do the tape's surface reconstructing using IMUs, enabling a multitude of motion-related applications.

We demonstrated that it is feasible to manufacture SensorTape using existing conductive inkjet and flexible electronics technologies. We show applications that leverage the unique tape form factor. Since it is flexible, it can be comfortably used for wearable applications, for example we attached it to the back to monitor posture. The SensorTape can be used as high-dimensional input, such as tangible 3D curve manipulation. Furthermore, it can make a surface interactive, such as making a wall that can measure height or making a proximity-based touchscreen. To enable more custom applications we developed an interface to graphically program the tape.

In the user study, we found that most participants saw the SensorTape as a wearable device, that is easy to customize. This was unexpected, since we originally envisioned it as something that is placed on objects or in the environment. Furthermore, the users suggested that SensorTape feels like material for arts and crafts, as it can be changed by cutting and joining. We believe that this project provides groundwork for future materials, which are very technologically sophisticated, and yet can be manipulated as a traditional tape or fabric.

ACKNOWLEDGMENTS

This project is partially supported by National Science Foundation award 1256082. We thank Nan-Wei Gong, Felix Heibeck, Nan Zhao for help and advice.

REFERENCES

1. Conductive ink jet. <http://www.conductiveinkjet.com/>.
2. i2c-bus specifications. http://www.nxp.com/documents/user_manual/UM10204.pdf, 2014.
3. Seeduino film. http://www.seeedstudio.com/wiki/Seeeduino_Film, 2014.
4. Adafruit neopixel digital rgb led strip. <http://www.adafruit.com/products/1138>, 2015.
5. Annett, M., Grossman, T., Wigdor, D., and Fitzmaurice, G. Medusa: a proximity-aware multi-touch tabletop. In *Proc. of UIST*, ACM (2011), 337–346.
6. Balakrishnan, R., Fitzmaurice, G., Kurtenbach, G., and Singh, K. Exploring interactive curve and surface manipulation using a bend and twist sensitive input strip. In *Proc. of i3D*, ACM (1999), 111–118.
7. Bdeir, A. Electronics as material: littlebits. In *Proc. of TEI*, ACM (2009), 397–400.
8. Gong, N.-W., Hodges, S., and Paradiso, J. A. Leveraging conductive inkjet technology to build a scalable and versatile surface for ubiquitous sensing. In *Proc. of Ubicomp*, ACM (2011), 45–54.
9. Gong, N.-W., Steimle, J., Olberding, S., Hodges, S., Gillian, N. E., Kawahara, Y., and Paradiso, J. A. Printsense: a versatile sensing technique to support multimodal flexible surface interaction. In *Proc. of CHI*, ACM (2014), 1407–1410.
10. Gong, N.-W., Wang, C.-Y., and Paradiso, J. A. Low-cost sensor tape for environmental sensing based on roll-to-roll manufacturing process. In *Proc. of Sensors*, IEEE (2012).
11. Hill, J. L., and Culler, D. E. Mica: A wireless platform for deeply embedded networks. *Micro, IEEE* 22, 6 (2002), 12–24.
12. Holman, D., Fellion, N., and Vertegaal, R. Sensing touch using resistive graphs. In *Proc. of DIS*, ACM (2014), 195–198.
13. Kahn, J. M., Katz, R. H., and Pister, K. S. Next century challenges: mobile networking for smart dust. In *Proc. of MobiCom*, ACM (1999), 271–278.
14. Kawahara, Y., Hodges, S., Cook, B. S., Zhang, C., and Abowd, G. D. Instant inkjet circuits: lab-based inkjet printing to support rapid prototyping of ubicomp devices. In *Proc. of Ubicomp*, ACM (2013), 363–372.
15. Lahey, B., Girouard, A., Burlison, W., and Vertegaal, R. Paperphone: understanding the use of bend gestures in mobile devices with flexible electronic paper displays. In *Proc. of CHI*, ACM (2011), 1303–1312.
16. Lee, J., Su, V., Ren, S., and Ishii, H. Handscape: a vectorizing tape measure for on-site measuring applications. In *Proc. of CHI*, ACM (2000), 137–144.
17. Lifton, J., Seetharam, D., Broxton, M., and Paradiso, J. Pushpin computing system overview: A platform for distributed, embedded, ubiquitous sensor networks. In *Pervasive Computing*. Springer, 2002, 139–151.
18. Merrill, D., Kalanithi, J., and Maes, P. Siftables: towards sensor network user interfaces. In *Proc. of TEI*, ACM (2007), 75–78.
19. Mistree, B. F., and Paradiso, J. A. Chainmail: a configurable multimodal lining to enable sensate surfaces and interactive objects. In *Proc. of TEI*, ACM (2010), 65–72.
20. Olberding, S., Gong, N.-W., Tiab, J., Paradiso, J. A., and Steimle, J. A. A cuttable multi-touch sensor. In *Proc. of UIST*, ACM (2013), 245–254.
21. Qi, J., and Buechley, L. Sketching in circuits: designing and building electronics on paper. In *Proc. of CHI*, ACM (2014), 1713–1722.
22. Rekimoto, J., Ullmer, B., and Oba, H. Datatiles: a modular platform for mixed physical and graphical interactions. In *Proc. of CHI*, ACM (2001), 269–276.
23. Rendl, C., Kim, D., Fanello, S., Parzer, P., Rhemann, C., Taylor, J., Zirkl, M., Scheipl, G., Rothländer, T., Haller, M., et al. Flexsense: a transparent self-sensing deformable surface. In *Proc. of UIST*, ACM (2014), 129–138.
24. Richardson, B., Leydon, K., Fernstrom, M., and Paradiso, J. A. Z-tiles: building blocks for modular, pressure-sensing floorspaces. In *Proc. of CHI*, ACM (2004), 1529–1532.
25. Vyas, R., Lakafosis, V., Lee, H., Shaker, G., Yang, L., Orecchini, G., Traille, A., Tentzeris, M. M., and Roselli, L. Inkjet printed, self powered, wireless sensors for environmental, gas, and authentication-based sensing. *Sensors Journal, IEEE* 11, 12 (2011), 3139–3152.
26. Watanabe, R., Itoh, Y., Asai, M., Kitamura, Y., Kishino, F., and Kikuchi, H. The soul of activecube: implementing a flexible, multimodal, three-dimensional spatial tangible interface. *Proc. of CIE* 2, 4 (2004), 15–15.