SCALE: Enhancing Force-based Interaction by Processing Load Data from Load Sensitive Modules

Takatoshi Yoshida[1] **Xiaoyan Shen**[2] **Koichi Yoshino**[3] **Ken Nakagaki**[1] **Hiroshi Ishii**[1]
[1]MIT Media Lab, Cambridge, MA, {taka_y, ken_n, ishii}@media.mit.edu
[2]MIT Art Culture & Technology, Cambridge, MA, xyshen@mit.edu
[3]TOPPAN Printing Co., Bunkyo-ku, Tokyo, koichi.yoshino@toppan.co.jp

Figure 1. SCALE system and its application scenarios: (a) three load sensitive modules and a controller (b) Shelf augmentation with *Touch Detection* **(c) Workbench augmentation with** *Object Status Tracking* **(d) Floor augmentation with** *Motion Pattern Recognition*

ABSTRACT
SCALE provides a framework for load data from distributed load-sensitive modules for exploring force-based interaction. Force conveys not only the force vector itself but also rich information about activities, including way of touching, object location and body motion. Our system captures these interactions on a single pipeline of load data processing. Furthermore, we have expanded the interaction area from a flat 2D surface to 3D volume by building a mathematical framework, which enables us to capture the vertical height of a touch point. These technical invention opens broad applications, including general shape capturing and motion recognition. We have packaged the framework into a physical prototyping kit, and conducted a workshop with product designers to evaluate our system in practical scenarios.

Author Keywords
Force-based Interaction; Tangible Interaction; Load Sensitive Modules; 3D Touch; Activity Recognition

INTRODUCTION
Force conveys fundamental information in Human-Object Interaction, including force intensity, its direction, and object weight [12] - information otherwise difficult to be accessed or inferred from other sensing modalities. When force is captured during interaction, a wide range of activities can be reconstructed such as way of touch, movement of objects and patterns of body motion. Therefore, it is important to explore the design space of *Force-Based Interaction*, which we define here as 'contact based dynamic interaction between two objects or between an object and the human body based on force vector direction and amount'.

Force-based interaction is involved at different scales in terms of the intensity of loaded force and the size of the interaction area. For instance, force-based interaction can range from actions such as drawing minute letters on a piece of paper (∼1g, 1mm), to handling tools on a workbench (∼1kg, 10cm), to dancing in a room (∼100kg, 10m). Even though researchers have already tackled each respective task, [20, 27], it is ideal if interaction designers are able to explore the wide range of force-based interactions within a single integrated framework.

In this work, we propose a framework of processing load data from load sensitive modules to cover the three main categories of force-based interaction, including *Touch Interaction*, *Object Status Tracking*, and *Motion Pattern Recognition*, as shown in Fig.1. The modularity of our system expands two key aspects of load sensitive applications: ***scalability*** in weight tolerance by adding a number of modules to fit the target load capacity on demand, and ***variability*** in spatial configuration by reconfiguring the spatial placement depending on their respective objectives.

Specifically for *Touch Interaction*, we have expanded the interaction area from a flat 2D surface to 3D volume by developing a new algorithm, which is freed from the geometric shape information of an object, which is required in the previous method [11]. With a broadened set of applicable objects, this function allows us to utilize the information of a touch point in 3D space for further analysis, telling us which part of an object is currently being touched, or what kind of shape outline the object has.

UIST '19, October 20-23, 2019, New Orleans, LA, USA

ACM. ISBN 978-1-4503-6816-2/19/10. . . $15.00
DOI: http://dx.doi.org/10.1145/3332165.3347935

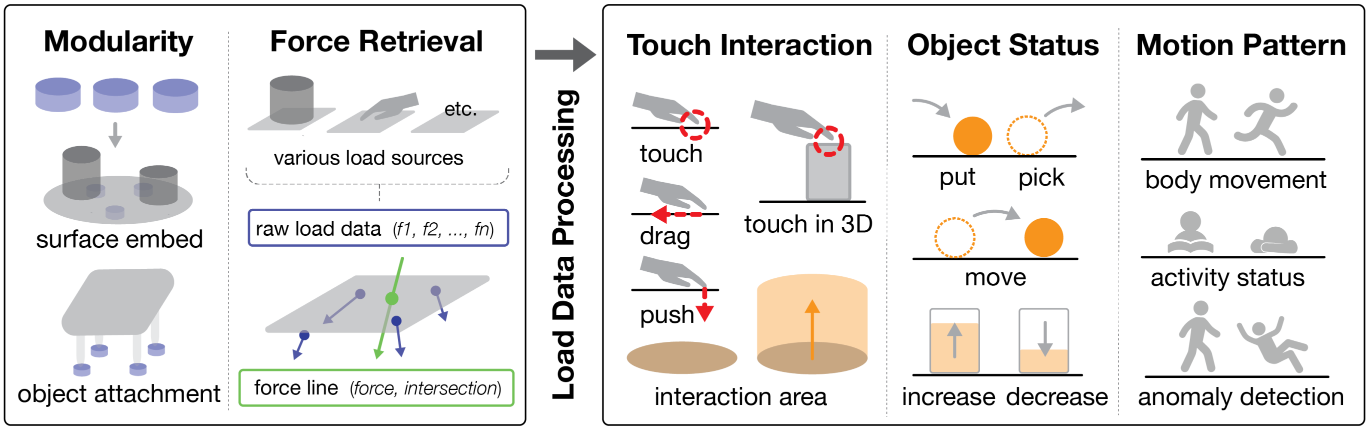


Figure 2. The concept of SCALE: (left) Modularity and Force Retrieval (right) Load Data Processing enables three types of functions: Touch Interaction, Object Status Tracking and Motion Pattern Recognition.

In addition to the algorithm improvements, we have implemented this framework into a physical prototyping kit with compact hardware and a GUI designed for novices. We additionally conducted a workshop with corporate designers and engineers to explore the application space enabled by the system, and evaluated its utility.

Our contributions described in this paper include;

- An architecture and design space for load sensitive modules to allow a range of force-based interactions, including touch interaction, object status tracking and motion pattern recognition.
- A new algorithm expanding interaction range from 2D to 3D above a load sensitive surface based on an inverse-matrix framework without prior shape knowledge.
- Technical implementation of hardware and GUI, and summarized findings from our workshop with corporate practitioners to explore the application space and to evaluate its utility.

RELATED WORK AND APPROACHES

The sensing technology for detecting the physical interactions between humans and objects is one of the primary research agendas in HCI. A number of contact sensing techniques using non force-based methods have been introduced, including vision-based [2, 14, 15], IR based [9], capacitive sensing [6, 16, 23, 28], swept frequency capacitive sensing [10, 25], EM based [33], microphone [13] and acoustic based method[21].

Among the techniques stated above, force-based sensing methods have the notable advantage of direct capturing of the contact force [31]. In the context of HCI, several force-based methods have been investigated, including Piezoelectric [7], and force-sensitive registers [5, 22]. In terms of *scalability* in weight, the methods with load cells show a wide range of applicability due to its high tolerance in maximum force [3, 19].

For the load-based methods, we have categorized the functionality into three parts, including *Touch*, *Object* and *Activity*. On the load-based approach, many systems have been proposed for touch detection purposes [20, 26, 30]. This approach naturally expands to variations of touch, including tap, press, drag and draw, however, the interaction area of these systems is constrained onto a 2D surface. Notably, INTACT pushes the interaction area to a 2D surface in 3D volume by assuming prior shape knowledge of the object on the geometrically-constrained surface [11]. Preliminary formulation of our approach is proposed previously [32], and we improved the algorithm in terms of mathematical stability with regularization terms, together with added design framework and workshop study.

Detection, or identifying objects, is another critical domain in the load-sensitive method. As the foundation of this category is regarding objects, the concept of *Weight as ID* conveys an essence that precise measurement of weight can be useful for identifying objects due to its occurrence in daily life [4]. Localization of the target has been a hot topic from fields such as Biology [24, 34] and Robotics [1, 17].

In addition to *Touch* and *Object*, load-based activity recognition has been investigated for many years. Context-aware systems have developed in combination with the algorithms of classifying signals [20, 26, 27]. Especially, the pose estimation for the human body has been a growing field [8, 29].

Among such broad applications on the load-based approach, our system as a prototyping tool kit unifies all the three application domains, including *Touch*, *Object* and *Activity*, into a single framework of load data processing. With the technical breakthrough being for detecting 3D touch, we expand the application field to everyday objects, freed from the requirement of having the geometric shape model in advance.

SCALE: A TOOL KIT FOR FORCE-BASED INTERACTION

Design Space

SCALE is a prototyping tool kit to encourage interaction designers and engineers to explore *force-based interaction*, which is uniquely enabled by capturing direct force information, with the architecture composed of load sensitive modules and a framework of load data processing. The key feature of SCALE is its modularity, aiming at *scalability* and *variability*, so that the users can increase the number of modules to be capable of accepting heavier load on demand, and place modules to reconfigure the spatial arrangement to fit their objectives, as shown in Fig.2 (left).

Furthermore, the modularity enables the system to cover a wide range of *force-based interaction* with the support with three functions in the load data processing, including *Touch Interaction*, *Object Status Tracking* and *Motion Pattern Recognition*, as shown in Fig.2 (right). Here we describe the design requirements for each process as following:

Touch Interaction

The system should be capable of capturing the interaction between a human and objects, and particularly *touch* is the common interaction seen in a wide range of situations. If the system captures both of the force intensity of a touch and the position of the touch, this information can be utilized for further analysis. For example the system could infer which part of an object is currently being touched. Furthermore, if the system has less constraints on an object, such as restrictions on a shape, the system could be applicable to many purposes. Therefore *Touch Interaction* of SCALE is designed to capture various types of touch interactions happening on 2D surfaces or in 3D volumes, freed from the shape constraint.

Object Status Tracking

The system should be capable of handling a large set of light and heavy objects in a single manner. A pen with 10 grams and an adult with 60 kg would represent the scalability seen around our life. Therefore *Object Status Tracking* has a function to track the object position and weight. By calculating total weight and center of mass, the five different status of an object can be classified: *pick*, *put*, *move*, *increase* and *decrease*.

Motion Pattern Recognition

The system should be capable of capturing what people are doing on a table, or how people are moving their body on a floor. When people walk or stretch, it causes different signal patterns on load sensors. So we designed *Motion Pattern Recognition* as a framework for recognizing different activities based on the signal pattern. Our simplest scheme is composed of feature extraction, and the support vector machine can distinguish between different user-defined activities.

On top of these processes, the user can develop their own applications in accordance to their purposes. Since this application space uniquely enabled by *force-based interaction* is thought to be broad, it is useful if the scope of the application space is being disclosed as a list of potential scenarios. Therefore we figured out the scope by having a workshop with corporate designers and engineers, as we describe the detail in the latter part of this paper.

System Architecture

The system architecture of SCALE is illustrated in the block diagram shown in Fig.3. There are three hardware components: *modules*, *controller* and *host computer*. Each module contains three-axis load cells and its peripheral circuits to transmit load data to controllers. All module data in the system is sent to one unified controller and pre-processed with a simple noise filtering. Since the raw data from load cells are sometimes polluted with sporadic saturated signals, we eliminate the outliers by applying a simple threshold on the absolute value of the raw data.

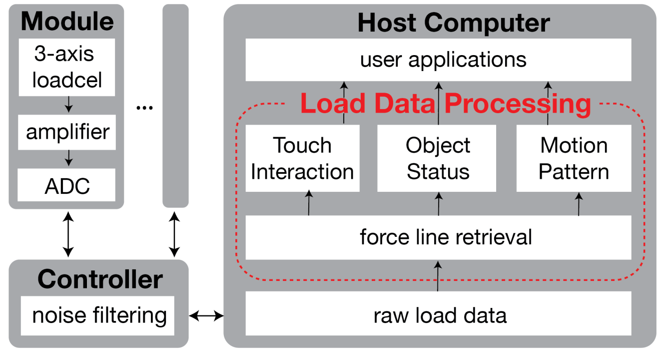


Figure 3. Architecture of SCALE: (left) Load sensitive modules and its controller (right) Applications on top of the *Load Data Processing* architecture on a host computer

The host computer receives raw load data from a controller through a USB serial bus. If we have N modules, s.t. $N \geq 3$, we receive an array of $3N$ load data. This load data is sent to the signal processing core called *Load Data Processing* and the system retrieves the force and its intersection as shown in Fig.2. This force information is exploited by following three different pipelines: *Touch Detection*, *Object Status Tracking* and *Motion Pattern Recognition*. After these three go through load data processing, the results are utilized to make user-defined applications.

LOAD DATA PROCESSING

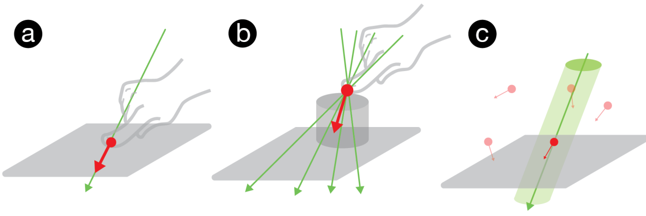
Force Line Retrieval

A *force line* is the key element of the architecture for the load data processing, which is mathematically represented as a set of *force* $\mathbf{f}$ and *intersection* $\mathbf{a}$ as shown in Fig.2. Here we describe how to retrieve a force line from raw load data. We assume the sets of measured force $\mathbf{f}_i$, sensed at i-th load module ($i = 1,2,...,N$). For simplicity, we could assume that all the sensors are placed at $\mathbf{p}_i$ on the same $z = 0$ plane. The touch force $\mathbf{f}$ and its torque τ is derived as $\mathbf{f} = \sum_i \mathbf{f}_i$ and $\tau = \sum_i \mathbf{p}_i \times \mathbf{f}_i$ by definition.

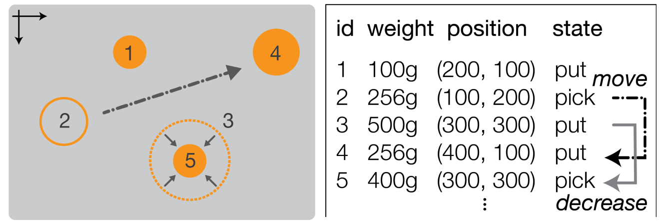
Here, the line of action for manual touch is expressed as $\mathbf{x} = \mathbf{a} + p\mathbf{d}$, parameterized by scalar p. The normalized direction vector $\mathbf{d}$ is $\mathbf{d} = \mathbf{f}/|\mathbf{f}|$ and the anchor point is $\mathbf{a}_0 = \mathbf{f} \times \tau/|\mathbf{f}|^2$. Since we can take an arbitrary point along the line as the anchor, we obtained the *intersection* $\mathbf{a}$ as the anchor point intersecting with the modular plane, where $\mathbf{a} = \mathbf{a}_0 - \frac{\mathbf{a}_{0z}}{\mathbf{d}_z}\mathbf{d}$. On this formulation, the scalar p becomes regularized by being zero at all times when the point is on the $z = 0$ plane.

Touch Interaction

We provide the algorithm to detect the touch point on a 2D surface or 3D object on load sensitive modules, as shown in Fig.4. Here especially, we describe a unique algorithm of *3D Touch Detection*, which exploits the unsteadiness of a hand during touch interaction. We assume enough rigidity in the



Touch Detection (c) Touch Classification



Objects (right) Database Manipulation

object, but it does not have to be composed of a single uniform material. Our framework accepts multi-material objects (e.g. a wooden desk with metal legs), as long as they convey force from a touch point to the sensors without internal dispersion.

2D Touch Detection

As illustrated in the previous section for *Force Line Retrieval*, we used the intersecting point **a** between the force line and $z = 0$ plane as the touch point, as shown in Fig.4(a). By constraining the existence area onto $z = 0$ plane geometrically, we can solve the mathematical ambiguity along the force line.

Another type of geometric constraint is investigated in a prior project called INTACT [11]. Instead of the $z = 0$ plane stated above, as the geometric constraint on the force line, they introduced the 2D surface envelope of an object. This approach was clever enough to expand the interaction area from a 2D surface to a 2D envelope in 3D volume, however, it is still less scalable since this approach requires prior shape information and its orientation of the object on the surface in advance. That means it is difficult to expand the application range of the method to an object with an unknown shape.

3D Touch Detection

To address the problem stated above, we propose an algorithm to localize the touch point in 3D space without any geometric constraints, with focus on the unsteadiness of a human hand. Even though our approach is still constrained on the 2D surface envelope of an object as well, this approach outstands since it does not require any prior shape information and can be applicable to any rigid object.

The key insight of our solution lies in the fact that when we touch an object with our hand, the touch is never stable. As illustrated in Fig.4 (b), when we aggregate the several recent lines they should have slight differences in direction. By looking at these lines, we can find the touch point as the most possible intersecting point of all the lines. Instead of assuming a geometric constraint, our approach equivalently introduces the temporal continuity of human touch. This assumption is thought to be valid when human touch is much slower than the frequency of load sensing, such as 80 Hz sensing with the sped-up ADC, which we introduced in the implementation section.

3D Touch Algorithm

Here we describe the detail of the algorithm to localize the 3D touch. Firstly for simplicity we transformed the equation for a force line $\mathbf{x} = \mathbf{a} + p\mathbf{d}$ into the form of a matrix equation, where $\mathbf{I}_3$ is a 3x3 unit matrix.

$$\begin{bmatrix} \mathbf{I}_3 & -\mathbf{d} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ p \end{bmatrix} = [\mathbf{a}] \quad (1)$$

This equation is apparently under-determined, so we must make the equation over-determined in order to calculate the touch point $\mathbf{x} = [x \quad y \quad z]$ by least square minimization. The touch point $\mathbf{x}$ can be assumed to be constant during a touch, and the system obtains different force lines $\mathbf{x} = \mathbf{a}_t + p_t\mathbf{d}_t$, where the discrete time stamp is denoted as t. When we collect the most recent T data during the touch, the matrix equation mentioned above naturally expands in the manner below:

$$\begin{bmatrix} \mathbf{I}_3 & -\mathbf{d}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{I}_3 & \mathbf{0} & -\mathbf{d}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{d}_T \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ p_1 \\ \vdots \\ p_T \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \vdots \\ \mathbf{a}_T \end{bmatrix} \quad (2)$$

Here we abbreviate the equation as $\mathbf{DX} = \mathbf{A}$ for simplicity, where $\mathbf{D} \in R^{3T\times T+3}$, $\mathbf{X} \in R^{T+3}$ and $\mathbf{A} \in R^{3T}$. Even though this equation has the worse condition number in terms of the inverse problem framework since the force lines are thought to be quasi-parallel, we can solve the equation by the support of appropriate regularization terms. Finally, we reach the least-squares solution $\mathbf{X}$ by using the Moore-Penrose pseudo-inverse matrix method. On this framework, the solution $\mathbf{x}$ tends to be constrained around the origin of the space, and slightly gets closer to the surface under the influence of the regularization on p_i as well.

Note that here we introduced *generalized Tikonov regularization*, rather than the standard Tikonov method with a uniform regularization parameter λ, to obtain a stable solution $\mathbf{X}$ by reducing the effect of sensing errors, which has introduced in a multi-modal sensing method [18]. This is because the regularization parameters, λ_x for $\mathbf{x}$ and λ_p for p_i, have different physical dimensions, such as $\mathbf{x}$ as a spatial position in mm and p_i as a dimensionless scalar. We experimentally adopted 20 for T, 0.1 for λ_x and 0.01 for λ_p. Here we finally reach the touch point $\mathbf{x} = [X1 \quad X2 \quad X3]$ as picking the first three components in $\mathbf{X}$:

$$\mathbf{X} = (\mathbf{D}^T\mathbf{D} + \mathrm{diag}(\lambda_x^2, \lambda_x^2, \lambda_x^2, \lambda_p^2, \cdots, \lambda_p^2))^{-1}\mathbf{D}^T\mathbf{A} \quad (3)$$

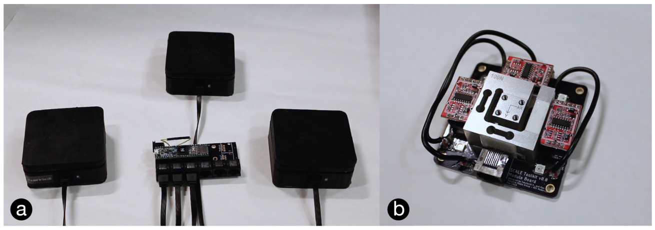


Figure 6. Load sensitive modules: (left) Modules with a 3d printed case, a detachable connector and a rubber for slip prevention (right) a 3d axis load cell and a peripheral circuit

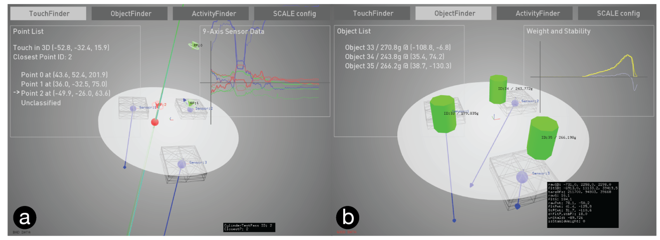


Figure 7. Interactive GUI for SCALE: (Left) It shows a current touch point and registered points for *Touch Detection* (Right) It shows the objects with its position and weight for *Object Status Tracking*

Touch Point Classification

The touch classification algorithm is shown to classify an immediate touch to a corresponding registered touch point. It takes T samples, typically 0.25 sec or more, to register a touch point as shown above. However, with the classification algorithm the system can detect the touch to registered points immediately with only a single sample of force line.

We will classify the green force line as the most possible registered point shown in Fig.4 (c). There are two steps of selection: *Cylindrical Search* and *Direction Similarity*. For the first step of Cylindrical Search, we will ignore all of the distant registered points from the force line with the threshold radius r. For the second step of Direction Similarity, we will calculate the inner product of normalized directions between input force line and that of the registered point. The appropriate parameter r heavily depends on the application, yet we generally adopt 30 mm for the threshold radius.

Object Status Tracking

To detect an object with weight and position and identify its status from load signals, there are two steps of load processing. The first step is called *Stability Check*, where the system determines the weight and position of a new object or the removal of an existing object. The second step called *Database Manipulation* is where the system accesses the internal database to identify the type of action. The core concept of the second part is *Weight as ID* insight, which claims weight information is useful to distinguish two or more different objects on a scale with required precision [4].

Stability Check

In the first part of Object Status Tracking, we focus on weight data $w_i = f_{iz}$, which is the equivalent z-component of load from each module. Here we have $w_{total} = \Sigma w_i$. To check the emergence or disappearance of objects, the system needs to distinguish *Stable* status, where every raw load data is almost static, from *Unstable* status. This stability check is conducted through simple thresholding by subtracting the slow LPF-ed (low pass filter) from the fast LPF-ed data.

$$stability = slow\text{-}LPF(w_{total}) - fast\text{-}LPF(w_{total})$$

If *stability* is small enough, it means the objects on the surface are *Stable*. This stability has a trade-off with response of the system. We experimentally adopted 2 grams as the threshold value for *stability*. Also, LPF is implemented as the exponentially weighed moving average, with the filter strength α at 0.04 for slow-LPF, and 0.25 for fast-LPF. Once the status is classified to *Stable*, the center of total weight x_{total} can be calculated as $x_{total} = \Sigma w_i x_i / \Sigma w_i$, where x_i is the position of i-th module.

Database Manipulation

In the second part of Object Status Tracking, the system handles the internal database and reflects the result to SCALE GUI, as shown in Fig.5. If the detected total weight w_{total} is above zero, the object is to be labelled as *put*. If the weight is not above zero, the object is labelled as *pick*. In either case, the objects are then added to the database. In Fig.5, the newly detected object #4 has the same weight as that of object #2, which is picked. Here these two objects are identified as the same, and merged into object #4. This operation is called *move*. If the new object #5 appears on the same position as the existing object #3, the system subtracts the weight from that of the existing object #5. This is *decrease* of the weight. The same procedure will apply for *increase*.

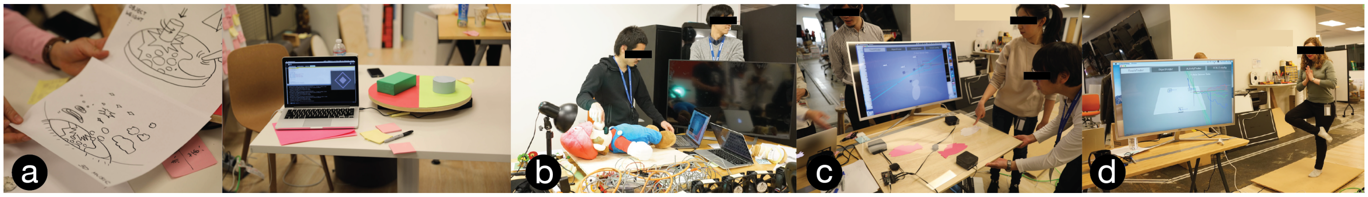
In our practical implementation, since the system faced errors in weight and position we need to set a tolerance to identify the values. We experimentally applied thresholds to identify two slightly different values to one value. As a result, we adopted 5 grams for the weight threshold and 3 cm for the position threshold.

Motion Pattern Recognition

Here we describe the pipeline to distinguish two or more activities from each other based on load signals. It is out of our scope to construct a pipeline to build *general* Motion Pattern Recognition framework, so we drew from activities that follow the same raw signal with periodic patterns.

In our pipeline, the incoming raw signals are converted into a feature vector, which expresses a specific type of motion by feature extraction. The user can choose any feature extraction method, including fast fourier transform, average, standard deviation and etc. The feature is fed to be classified by a support vector machine (SVM) algorithm.

Specifically for our applications, we record the force and torque vectors for 1 sec with 30 Hz sampling rate, and then we derived the standard deviation in each component as a 6 dimensional feature vector. Also we adopted the fine Gaussian kernel for the detailed algorithm for classification.



Application Space	Ideas	# of Ideas per Group				FBI types		
		A	B	C	D	Object	Touch	Motion
Healthcare	Daily Health Check	5		3	4		●	●
	Help Medical Examination	3	1				●	●
Surveillance	Personal Identification	2	1				●	●
	Tracking Living things / Objects	1	2	3	5	●	●	●
Cooking	Help Cooking	2	3			●	●	●
	Record Cooking	1	1	1	1	●	●	
Enter-tainment	Game UI		1	1		●	●	
	Help Creative Works		3			●	●	●
	Compose & Play Music	5	1	2	1	●	●	
Home	Control Air / Sound / Light / TV	2	6	5		●	●	●
	Help Non-Verval Communication		1				●	●
Learning	Storytelling with Figures				1	●	●	
	How to Use Device				1	●	●	
	Observe & Point to Living Things		1				●	●
	Dance / Instruments / Yoga	1		1			●	●

Figure 9. Classification and analysis of application scenarios from corporate designers and engineers

SCALE PROTOTYPE

Modular Hardware

The overall SCALE architecture is illustrated on Fig.3. We designed two types of hardware to maximize usability of the entire system: modules and a controller.

Each module contains a three-axis load cell (FNZ100N, Forsentek.inc) with a load capacity of 10 kg, and three amplifiers (HX711) with analog-digital converters in the fastest mode at 80 Hz. The module is cuboid with an 90 x 90 x 35 mm form factor. To make the entire module compact enough to fit on the palm of your hand, we designed an original PCB and put all of these elements inside of a 3d-printed cabinet, as shown in Fig. 6. To maximize the grip between the module and floor or object, we put layered rubber onto both sides of the module surface.

The load sensitive modules are to be connected to a single controller with ethernet cables, which has a detachable and regularized connector so that a user can easily reconfigure the number and the placement of modules. A single controller is capable of being connected with 8 modules at maximum, which leads to *scalability* in weight tolerance and *variability* in spatial configurations. A controller contains a micro processor (Teensy 3.6) to aggregate and pre-process all of the raw data from the modules, and transmit them to the host computer.

Software GUI

All of the software composed of real time signal processing and Graphical User Interfaces (GUI) is implemented on the open-source library (openFramework) by C++, as shown in Fig.7, except the *Motion Pattern Recognition* feature, which is implemented on Matlab environment.

The GUI provides three different primitive modes, including *Touch Detection*, *Object Status Tracking*, and *Motion Pattern Recognition* (only for capturing signals), and the user can develop an integrated system on top of these three basic functions. For all primitive modes, the user is capable of interactively registering a current touch point or object to the database and selectively serialize them for further analysis for other applications.

WORKSHOP FOR EXPLORING APPLICATION

We conducted a SCALE hands-on workshop to evaluate the utility and to explore potential applications which we had never expected. The workshop procedure was designed in a way participants can accomplish prototyping their ideas and present their narratives with the developed demonstrations.

Designing Workshop

With the support of a product corporation, 12 designers and 8 engineers attended the workshop and were divided into 4 teams to evenly distribute expertise in each group. There were three sessions in the workshop. The first 2-hour slot was designed to brainstorm new application scenarios. The participants were asked to come up with as many small use cases possible, to then merge them into a larger concept. The second session was 6 hours of hands-on participation to develop functional applications with the SCALE development kit. After providing detailed instructions to use the kit, each team that is composed of 5-6 people started to collaborate with colleagues to prototype their own ideas. We concluded with a one hour session to present the developed ideas and prototypes and to receive feedback from peers.

Exploring Application Space

We have compiled the ideas that corporate designers and engineers developed from the brainstorming session into Fig. 9. To catch the core interests of participants, we classified the ideas into six categories: Health-care, Surveillance, Cooking, Entertainment, Home and Learning. Among a range of promising scenarios, we picked some notable ideas worth sharing: (1) monitoring one's health through the analysis of posture changes while sitting, walking and sleeping; (2) tracking activity of pets or growth of babies and plants; (3) controlling home devices, including speakers, lights and air, through direct contact with furniture, walls or floor, rather than through digital interfaces. In addition to these ideas, from the user's perspective we received comments that mention a guideline on

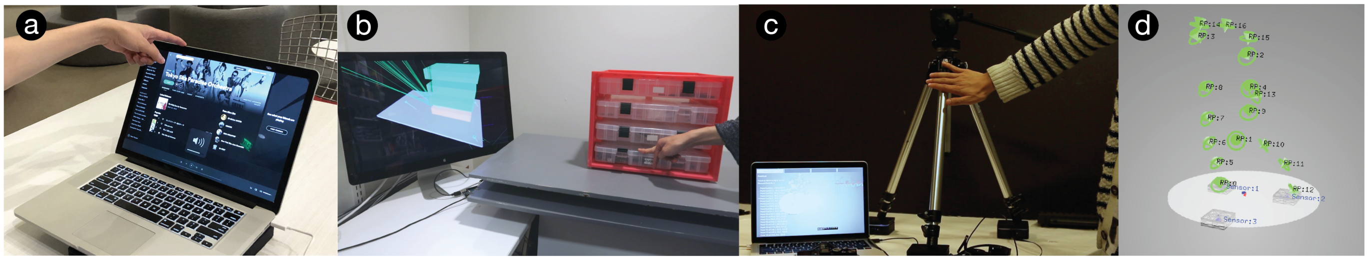


Figure 10. Demonstration for *Touch Detection*: (a) *Virtual Interface on Physical Objects* (b) *Embedded Usage Tracker* (c) *General Shape Capturing* with a tripod as a target (d) a close-up picture of captured shape of the tripods

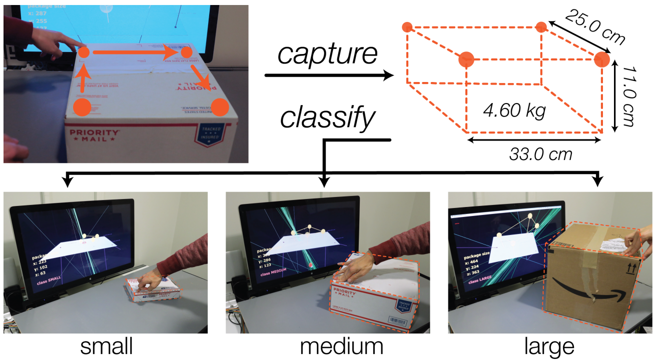


Figure 11. Demonstration for *Shape Capturing*: The system classifies the type of a package based on weight and its size.

how to develop or implement each idea on top of our software pipeline, and the best use cases for *Touch Interaction* where the system becomes *the best from a practical point of view* among all sensing technology.

From the hands-on session, we had four different functional prototypes, as shown in Fig. 8. We briefly describe them in the following list:

- *Group A*: The **tangible music composer** is implemented on *Object Status Tracking*, and allows the user to play and mix up music based on the placement of different types of objects on specified disk locations, as shown in Fig. 8(a).
- *Group B*: The **interactive story-telling** with voiced characters is designed for children to breathe life into their favorite toys through a pre-recorded voice-over triggered by touch interactions, which is implemented on *Touch Detection* and *Object Status Tracking*, as shown in Fig. 8(b).
- *Group C*: A **fish pointing system for future aquarium** utilizes the *3D Touch Detection* technique to select a specific fish swimming in the middle of a large tank with the assumption that the 3D position of all fish are tracked by computer vision, as shown in Fig. 8(c). This application provides detailed knowledge of the selected fish, such as the name, species, habitat and food, by touching on the load sensitive glass window.
- *Group D*: The last application is the **posture-aware floor for Yoga practitioners**, designed to identify individuals and analyze their posture and to allow the system to advise the individual on how to modify a post for safe practice, as shown in Fig. 8(d).

Evaluating Utility

To analyze the utility of the toolkit from a viewpoint of practicality, we conducted the subjective evaluation by distributing a questionnaire after the workshop. The questions were along the lines of, "How did you feel about SCALE as a ubiquitous sensitive system?" by using Likert's five points scale from "Very Good" to "Very Bad" and, "What are the pros and cons of the toolkit?" through open response. We received answers from 10 participants. The resulting scores from the first question are 4.6 / 5.0(*Average*), 5.0(*Median*) and 0.66(*SD*).

Regarding the comments from the second question, the positive comments are as follows: "It's really useful to be able to sense a variety of different things about the physical state of objects or people using a surface and invisible sensor" *(Female, Industrial Designer)*, "The interface is intuitive" *(Male, Chemical Engineer)* and "Detecting not only the single touchpoint but a series of touchpoints that translate into an activity" *(Female, Experience Designer)*. Among the negative comments were: "The threshold of SCALE should be adjusted so that people can act by elbow, body and so on" *(Male, Cognitive Psychologist)*, "The necessity of detection range and UI for ease to control" *(Male, Software Engineer)* and "Accuracy across large surfaces, sensitivity across multiple touch points at different densities" *(Female, Experience Designer)*.

The results of the questionnaire and brainstorming session as shown in Fig.9, which allow us to consider the following points. Firstly, we can see that an advantage of SCALE is the capability to recognize a wide variety of *Touch Interaction* with invisible forces. Secondly, SCALE is expected to use its *Motion Pattern Recognition* for grasping multiple interaction touch points. Finally, improvements on the versatility and application of SCALE are needed.

On the other hand, we found a issue regarding the constrain of the number of sensor module. While the reconfigurability of sensor modules made it easy for participants to quickly customize the layout of the modules, our prototype was constrained to use three module. This limitation made it mechanically unstable for some of the large scale interaction prototypes (e.g. body gesture detection). We plan to improve our User Interface software and force vector calculation algorithms to accommodate multiple (more than three) sensor modules placements.

Figure 12. Demonstration for *Object Status Tracking*: **(a)** *Retail Automation* **enables to capture object movement and liquid consumption (b)** *Smart Workspace* **is monitoring the location and usage of the tools**

SCALE APPLICATIONS

Reflecting on the concluding remarks from the workshop, we identified 4 application areas where we felt that SCALE could have a potential impact - either as a useful enhancement to an established application or a novel application, uniquely enabled by our approach:

- making everyday objects and surfaces force sensitive
- capturing the general shape of an object by touching it
- locating objects, including liquids, through weight identification
- making home fixtures an activity tracking platform (eg. floors)

In the rest of this section we propose a few exemplary applications for each category, shedding light on the utility and scope of our sensing approach.

Volume Slider on PC Monitor

If everyday objects can be sensitive to touch, including touch position, direction, and intensity, they can configure functions in productive ways. The canonical example would be a PC monitor with a user-defined touch point, as shown in Fig. 10(a). When a user touches the top-right corner, the audio volume changes from low to high according to pressing force. A user can also assign a power button just next to the mute button, since the system can differentiate two overlapped registered points with the classification algorithm.

Shelf Usage Tracker

In addition to enhancing a PC monitor, *making everyday objects force sensitive* can be useful for objects with no feedback system inside. A user can easily augment a tool shelf containing different types of screws into a trackable activity tool by putting only three modules beneath the shelf or table surface, as shown in Fig. 10(b). When a user opens the third drawer and grabs some screws, the quantity and the type of screws are distinguished immediately.

Shape Capturing By Touch

Our 3D touch algorithm allows a user to capture the general shape of an object, like a notebook PC, by touching its outer points. After a user repeatedly touches multiple points around the object, the detected points are connected, and a contour of the object is captured, as shown in Fig.10(c).

Since our system is capable of capturing the general shape of an object from only load data, the system classifies an object into the user-defined categories based on its weight and estimated size, as shown in Fig.11. This could be useful for the application requiring simultaneous acquisition of weight and rough shape, including the measurement of packages at postal offices, or the airport counter to check-in the bags for flight, to estimate its cost and rough volume.

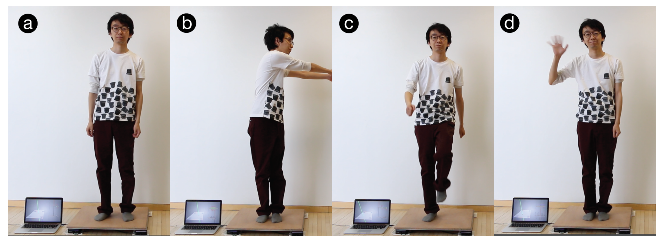


Figure 13. Demonstration for *Motion Pattern Recognition*: **The system classifies four different activities: (a)***stand* **(b)***stretch* **(c)***walk* **(d)***wave*

Retail Automation

On top of the *object status detection* mode, combined with an external database of product information, it is possible to prototype an automated checkout system on a load sensitive table as shown in Fig. 12(a). Recently, this type of application has been well-investigated on machine-vision systems, yet our load sensitive approach is adding an essential value of weight-based interaction, including selling-by-weight. In addition to discrete objects, liquids or granular products are under coverage of the SCALE system. A customer can take as much coffee as they want, and be charged according to the exact amount of consumption, since the change in weight is captured with its position.

Smart Workspace

The workbenches or tables enhanced by load sensitive modules are becoming smart enough to track the usage and positions of tools, like a handy drill, as shown in Fig. 12(b). The system remembers the previous position of a handy drill, so that the user can indicate the current location of the tool through other display techniques. Additionally, if the user forgets the place where the drill should be returned, the system will notify you of the location by searching in its database.

Posture Estimation on a load sensitive floor

Once load sensitive modules are embedded beneath the room floors, the surface immediately becomes capable of *motion pattern recognition*. From the different wave shape of load signals, the system classifies the type of movement (running)

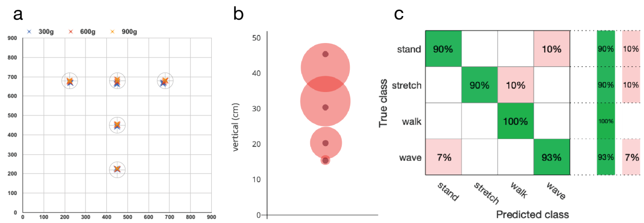


Figure 14. Results from Technical Evaluations: (a) Horizontal Accuracy from *Object Status Tracking* **(b) Vertical Accuracy from** *3D Touch Detection* **(c) Confusion Matrix for** *Motion Pattern Recognition*

and displays a caution to stop running inside the room, as shown in Fig. 13. Further analysis including affection inference or user recognition could be implemented on top of the load processing framework we have proposed in this paper.

TECHNICAL EVALUATION

Here we provide the performance of our prototype we experimentally evaluated to support the viability of the system. We setup the measurement on accuracy and precision concerning spatial position, and conducted two different experiments for the *horizontal* plane and the *vertical* axis, as shown in Fig. 14.

The *horizontal* accuracy is measured on *xy* plane, especially related to 2D touch detection or object localization. As shown in Fig. 14(a), we achieved less than 1 cm accuracy in the prototype, tested with the three different weights (300, 600, 900 gram) to check the weight consistency of the algorithm.

The *vertical* accuracy is measured along the z axis to evaluate 3D touch point detection, shown in Fig. 14(b). We put a fixed size shelf on the SCALE platform, and keep touching a point on each surfaces for 1 sec. We repeatedly obtain the estimated height for 10 times. In the figure, we illustrated the tested height as a small red dot and the standard error as a bigger red circle. At most we have 7cm accuracy at the height of 50 cm. While the error seemingly expands according to the height, it would be useful to distinguish two different surfaces in a shelf.

Also, as shown in Fig.14(c), we classified four motion patterns according to the proposed pipeline, and evaluated the accuracy of the prediction by making a confusion matrix. For the specific four different body motion, the prototype successfully classified them with more than 90 % accuracy.

DISCUSSION AND LIMITATION

Multi-touch Inability

Our system is not designed to accept multi-touch input in parallel to other load sensitive systems. If two different people are interacting on the surface or handling the objects simultaneously, the data processing framework would fail. This is because as shown in Fig.2 we combine all of the signals from load cells into one force line at the very beginning of the processing pipeline. Thanks to the modularity of our system, we can apply different modules beneath two areas where a user would like to separately detect multi-touches.

Database

An additional database about *product*, which is composed of product name, sale price, or materials would be useful to build wider applications, especially for *Object Status Tracking*. For example, our basic system stores the set of (*weight*, *position*) as shown in Fig.5. Once we assign the initial relation between weight and product, or position and product, the system is capable of tracking all changes during its execution.

Speed

This system has 80 Hz throughput of touch point detection, yet we are facing an unavoidable latency of at least 0.25 sec, since the system requires this for the acquisition of a bundle of quasi-parallel action lines, and it usually takes more than 20 samples. Although our system could apply to 3D input, there are limitations in expanding to temporal critical applications, such as making instant musical instruments with pieces of cardboard.

Scalability

We can deploy a much larger system, such as a load sensitive floor on an architecture scale, with the advantage of area and weight scalability. Thanks to the modularity of our system, we can put as many modules as a user requires to meet the maximum load requirement. If the user exceeds the load tolerance of the system, they have another option of using higher-capacity load cells, such as ones with 100kg tolerance, in turn sacrificing the minimum distinguishable weight on the platform.

CONCLUSION

We proposed a load processing framework with load sensitive modules for enhancing force-based interaction, and explored its design space with scalable and variable architecture. The workshop with corporate designers shows a range of applications and the utility of a modular prototyping kit with the algorithm including 3D touch detection. We envision the SCALE framework provide ubiquitous interactive surfaces with scalable load sensitive architecture to capture scalable Force-based Interactions of everyday activities for further analysis of human object interaction.

ACKNOWLEDGMENTS

We are deeply grateful to Panasonic β Research Lab (CA, USA) for the workshop opportunity and their insightful feedback. We also would like to thank Ms. Deema Qashat for her proofreading and comments.

REFERENCES

[1] Mihai Andries. 2015. *Object and human tracking, and robot control through a load sensing floor*. Ph.D. Dissertation. Université de Lorraine.

[2] Patrick Baudisch, Torsten Becker, and Frederik Rudeck. 2010. Lumino: tangible blocks for tabletop computers based on glass fiber bundles. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1165–1174.

[3] Dunn Brian. 2016. *Load Cells Introduction and Applications*. BD Tech Concepts LLC.

[4] Andrew Carvey, Jim Gouldstone, Pallavi Vedurumudi, Adam Whiton, and Hiroshi Ishii. 2006. Rubber Shark As User Interface. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*. ACM, New York, NY, USA, 634–639. DOI: http://dx.doi.org/10.1145/1125451.1125582

[5] D. Antony Chacon, Eduardo Velloso, Thuong Hoang, and Katrin Wolf. 2019. SpinalLog: Visuo-Haptic Feedback in Musculoskeletal Manipulation Training. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '19)*. ACM, New York, NY, USA, 5–14. DOI: http://dx.doi.org/10.1145/3294109.3295626

[6] Liwei Chan, Stefanie Müller, Anne Roudaut, and Patrick Baudisch. 2012. CapStones and ZebraWidgets: sensing stacks of building blocks, dials and sliders on capacitive touch screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2189–2192.

[7] Gustav Gautschi. 2002. Piezoelectric sensors. In *Piezoelectric Sensorics*. Springer, 73–91.

[8] Daniel J. Goble, Brian L. Cone, and Brett W. Fling. 2014. Using the Wii Fit as a tool for balance assessment and neurorehabilitation: the first half decade of "Wii-search". *Journal of NeuroEngineering and Rehabilitation* 11, 1 (08 Feb 2014), 12. DOI: http://dx.doi.org/10.1186/1743-0003-11-12

[9] Jefferson Y. Han. 2005. Low-cost Multi-touch Sensing Through Frustrated Total Internal Reflection. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST '05)*. ACM, New York, NY, USA, 115–118. DOI: http://dx.doi.org/10.1145/1095034.1095054

[10] Colin Honigman, Jordan Hochenbaum, and Ajay Kapur. 2014. Techniques in Swept Frequency Capacitive Sensing: An Open Source Approach.. In *NIME*. 74–77.

[11] Charles Hudin, Sabrina Panëels, and Steven Strachan. 2016. INTACT: Instant Interaction with 3D Printed Objects. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2719–2725.

[12] Hiroshi Ishii and Brygg Ullmer. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. ACM, 234–241.

[13] Hiroshi Ishii, Craig Wisneski, Julian Orbanes, Ben Chun, and Joe Paradiso. 1999. PingPongPlus: design of an athletic-tangible interface for computer-supported cooperative play. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 394–401.

[14] Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. 2007. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*. ACM, 139–146.

[15] Yasuaki Kakehi, Kensei Jo, Katsunori Sato, Kouta Minamizawa, Hideaki Nii, Naoki Kawakami, Takeshi Naemura, and Susumu Tachi. 2008. ForceTile: tabletop tangible interface with vision-based force distribution sensing. In *ACM SIGGRAPH 2008 new tech demos*. ACM, 17.

[16] Sven Kratz, Tilo Westermann, Michael Rohs, and Georg Essl. 2011. CapWidgets: tangile widgets versus multi-touch controls on mobile devices. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. ACM, 1351–1356.

[17] M.H Lee and H.R Nicholls. 1999. Review Article Tactile sensing for mechatronics âĂŤ a state of the art survey. *Mechatronics* 9, 1 (1999), 1 – 31. DOI: http://dx.doi.org/https://doi.org/10.1016/S0957-4158(98)00045-2

[18] Leo Miyashita, Ryota Yonezawa, Yoshihiro Watanabe, and Masatoshi Ishikawa. 2015. 3D Motion Sensing of Any Object Without Prior Knowledge. *ACM Trans. Graph.* 34, 6, Article 218 (Oct. 2015), 11 pages. DOI: http://dx.doi.org/10.1145/2816795.2818133

[19] I. Muller, R. M. de Brito, C. E. Pereira, and V. Brusamarello. 2010. Load cells in force sensing analysis – theory and a novel application. *IEEE Instrumentation Measurement Magazine* 13, 1 (February 2010), 15–19. DOI:http://dx.doi.org/10.1109/MIM.2010.5399212

[20] Kazuya Murao, Junna Imai, Tsutomu Terada, and Masahiko Tsukamoto. 2015. Recognizing activities and identifying users based on tabletop activities with load cells. In *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services*. ACM, 39.

[21] Makoto Ono, Buntarou Shizuki, and Jiro Tanaka. 2013. Touch & activate: adding interactivity to existing objects using active acoustic sensing. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 31–40.

[22] Leonel Paredes-Madrid, Arnaldo Matute, Jorge Bareño, Carlos Parra Vargas, and Elkin Gutierrez Velásquez. 2017. Underlying physics of conductive polymer composites and force sensing resistors (FSRs). a study on creep response and dynamic loading. *Materials* 10, 11 (2017), 1334.

[23] Jun Rekimoto. 2002. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 113–120.

[24] Roy E Ritzmann and Sasha N Zill. 2017. Control of locomotion in hexapods. In *The Oxford Handbook of Invertebrate Neurobiology*.

[25] Munehiko Sato, Ivan Poupyrev, and Chris Harrison. 2012. Touché: Enhancing Touch Interaction on Humans, Screens, Liquids, and Everyday Objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 483–492. DOI: http://dx.doi.org/10.1145/2207676.2207743

[26] Albrecht Schmidt, Martin Strohbach, Kristof Van Laerhoven, Adrian Friday, and Hans-Werner Gellersen. 2002b. Context acquisition based on load sensing. In *International Conference on Ubiquitous Computing*. Springer, 333–350.

[27] Albrecht Schmidt, Martin Strohbach, Kristof Van Laerhoven, and Gellersen Hans-W. 2002a. Ubiquitous interaction using surfaces in everyday environments as pointing devices. In *ERCIM Workshop on User Interfaces for All*. Springer, 263–279.

[28] Martin Schmitz, Jürgen Steimle, Jochen Huber, Niloofar Dezfuli, and Max Mühlhäuser. 2017. Flexibles: Deformation-Aware 3D-Printed Tangibles for Capacitive Touchscreens. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 1001–1014. DOI:http://dx.doi.org/10.1145/3025453.3025663

[29] Julien Tripette, Haruka Murakami, Katie Rose Ryan, Yuji Ohta, and Motohiko Miyachi. 2017. The contribution of Nintendo *Wii Fit* series in the field of health: a systematic review and meta-analysis. *PeerJ* 5 (Sept. 2017), e3600. DOI: http://dx.doi.org/10.7717/peerj.3600

[30] Geoff Walker. A review of technologies for sensing contact location on the surface of a display. *Journal of the Society for Information Display* 20, 8 (????), 413–440. DOI:http://dx.doi.org/10.1002/jsid.100

[31] Yuzhang Wei and Qingsong Xu. 2015. An overview of micro-force sensing techniques. *Sensors and Actuators A: Physical* 234 (2015), 359 – 374. DOI: http://dx.doi.org/https://doi.org/10.1016/j.sna.2015.09.028

[32] Takatoshi Yoshida, Xiaoyan Shen, Tal Achituv, and Hiroshi Ishii. 2018. 3D Touch Point Detection on Load Sensitive Surface Based on Continuous Fluctuation of a User Hand. In *SIGGRAPH Asia 2018 Posters (SA '18)*. ACM, New York, NY, USA, Article 39, 2 pages. DOI: http://dx.doi.org/10.1145/3283289.3283339

[33] Yang Zhang, Chouchang Yang, Scott E. Hudson, Chris Harrison, and Alanson Sample. 2018. Wall++: Room-Scale Interactive and Context-Aware Sensing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. 273. DOI: http://dx.doi.org/10.1145/3173574.3173847 Exported from https://app.dimensions.ai on 2019/04/04.

[34] Sasha Zill, Josef Schmitz, and Ansgar BÃijschges. 2004. Load sensing and control of posture and locomotion. *Arthropod Structure Development* 33, 3 (2004), 273 – 286. DOI:http://dx.doi.org/https://doi.org/10.1016/j.asd.2004.05.005 Arthropod Locomotion Systems: from Biological Materials and Systems to Robotics.