TRing: Instant and Customizable Interactions with Objects Using an Embedded Magnet and a Finger-Worn Device

Sang Ho Yoon, Yunbo Zhang, Ke Huo, Karthik Ramani
School of Mechanical Engineering, Purdue University
West Lafayette, IN 47907, USA
{yoon87, zhan2014, khuo, ramani}@purdue.edu

ABSTRACT

We present *TRing*, a finger-worn input device which provides instant and customizable interactions. *TRing* offers a novel method for making plain objects interactive using an embedded magnet and a finger-worn device. With a particle filter integrated magnetic sensing technique, we compute the fingertip's position relative to the embedded magnet. We also offer a magnet placement algorithm that guides the magnet installation location based upon the user's interface customization. By simply inserting or attaching a small magnet, we bring interactivity to both fabricated and existing objects. In our evaluations, *TRing* shows an average tracking error of 8.6 mm in 3D space and a 2D targeting error of 4.96 mm, which are sufficient for implementing average-sized conventional controls such as buttons and sliders. A user study validates the input performance with *TRing* on a targeting task (92% accuracy within 45 mm distance) and a cursor control task (91% accuracy for a 10 mm target). Furthermore, we show examples that highlight the interaction capability of our approach.

Author Keywords

Wearable Input Device; Embedded Input; Magnetic Sensing Technique; Design Toolkit; Customization

ACM Classification Keywords

H.5.2. [Information Interfaces and Presentation]: User Interfaces – Input devices and strategies

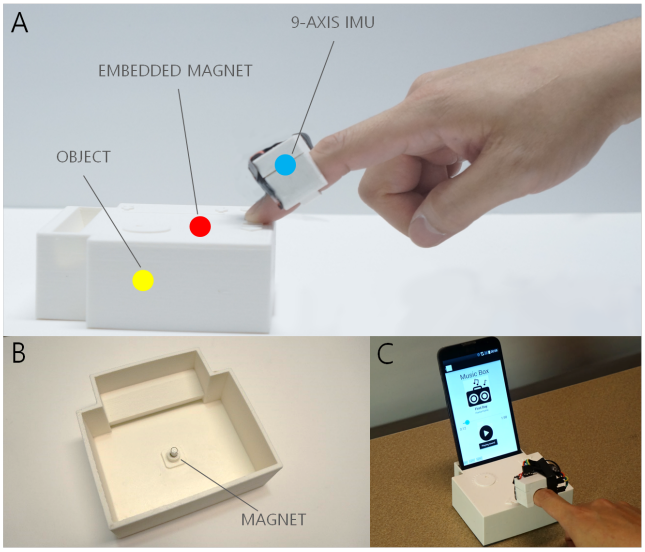

Figure 1. ***TRing*** **is (A) a finger-worn device with a magnetic sensing technique to track the fingertip around a permanent magnet. (B) By embedding a magnet in the object, (C) we bring interactivity to objects without placing electronic components or hardware in the object.**

INTRODUCTION

The rise of the Do-It-Yourself (DIY) and maker movements [1] has resulted in widespread use of additive manufacturing. In addition, the scaled up manufacturing of low cost embedded electronic hardware (microcontroller and sensors) has enabled rapid prototyping of interactive objects [42]. Recent approaches have focused on minimizing the post-processing stage for embedding interactive elements through novel sensing techniques [38] or printing methods [13]. A multi-material 3D electronics printer [43] is also in the process of commercialization. Utilizing these approaches, users can save time and money when making interactive objects. Moreover, these approaches support a growing number of novice "makers" who possess little background knowledge in dealing with electronic components and sensors.

Researchers also utilize the affordance and the availability of everyday objects to perform interactions [9, 24]. By utilizing everyday objects, users can lessen their dependence on dedicated input devices such that the object itself becomes the interface and the controller. Due to the nature of the existing physical structure of objects, previous works have adopted depth cameras, sensors, or tags to formulate interactions. These works demonstrate a natural way to bridge the gap between the physical environment and cyberspace as proposed in *Tangible Bits* [20].

However, existing methods of making objects interactive still require users to post-assemble basic electronic components including batteries, wires, and ICs on the object side [30]. Moreover, conventional approaches require battery management for each interactive object for the duration of use. In terms of sensing techniques, interactions with objects still rely on computer vision, for which users need to install a camera and attach visible tags to the object. These limita-

UIST 2016, October 16-19, 2016, Tokyo, Japan
 ISBN 978-1-4503-4189-9/16/10...$15.00
DOI: http://dx.doi.org/10.1145/2984511.2984529

tions increase manufacturing costs and reduce the flexibility of the interface customization for many users.

Recent magnetic sensing techniques have provided various input metaphors, from proximity sensing to position tracking. Previous works focused on using permanent magnets with magnetometers. However, these works embedded magnets in the tracked parts, such as a finger or stylus. Previously, putting a magnetometer in a tracked body was not feasible since high magnetic flux from a magnet distorted the geomagnetic field. This led to the inability of inertial measurement units (IMUs) to provide orientation estimation near the permanent magnet. In this work, we adopt a particle filter to obtain robust orientation estimation regardless of magnetic distortion. With our approach, we track the 3D position of the fingertip relative to the magnet. This allows us to make objects interactive by embedding magnets instead of installing electronic components.

In this work, we present *TRing*, which is an instant and customizable input based on magnetic sensing through a finger-worn device. Our prototype supports both fabricated and everyday objects as interaction mediums. Through our approach, users can instantly build and control interactive objects by simply inserting or attaching a magnet. Our contributions are as follows:

- A novel sensing technique providing real-time 3D position tracking around embedded magnets using a 9-axis IMU;
- A magnet placement algorithm that guides the magnet location based on a user's interface customization;
- A new approach to bring interactivity to objects using a finger-worn device that senses embedded magnets;
- Example applications demonstrating instant and customizable interactions with objects.

RELATED WORK

Bringing Interactivity to an Object

In practice, bringing interactivity to objects has been done through installing electronic hardware in existing products. In the post-processing stage, users generally assemble and test hardware modules manually. Recent approaches in fabricating interactive objects have focused on reducing the time and cost of this post-processing. These include multi-directional/multi-material 3D printing [13, 41], physical mark-up [39], embedded single-camera [38], structural electronics [28], acoustic sensing [16, 34], and layered fabrics [35]. These approaches still require substantial amounts of post-processing such as wiring and electronic module assembly, but it can be eliminated, saving significant time. In our work, we further suggest a method to fully remove the electronic hardware components from the object side, thereby enabling easy and low-cost installation and maintenance. Although previous approaches require less instrumentation [34, 40], the richness and sustainability of the interaction are still limited due to either the lack of position tracking or the requirement to install additional hardware. Our approach also introduces a simple way to modify the interface without iterating the whole post-processing or data training steps.

Everyday Object Interaction

Performing interaction using everyday objects offers task affordances while utilizing existing environments where these objects are used. This creates an opportunity for an instant and casual interaction in which naturally embedded behaviors with existing objects function as meaningful interactions [36]. To this extent, *iCon* and *Instant Interface* employ fixed cameras to transform everyday objects into an auxiliary controller and instant interface [9, 10]. Kranz et al. [24] demonstrate everyday object interfaces integrated with multimodal sensors to track human activity. *ReachMedia* and *IDSense* instrument everyday objects with RFID tags to detect object motions or related human activities [12, 25]. However, these approaches require a room/object-level hardware set-up that limits the availability and applicability of the interactions. *TRing* only requires embedding a small permanent magnet into existing objects to create an instant interface.

Position Tracking Using Magnetic Sensing Technique

Magnetic sensing has been explored extensively for position tracking. Tracking using active magnetic sources has been thoroughly explored in gaming [21], tangibles [26], and 3D interactions [18] based on its ability to track multiple points with high precision. Position tracking using passive magnetic sources and magnetometers has been also explored. This approach has demonstrated an acceptable performance as a wearable input for mobile applications [8]. Previous approaches required hardware installation on the reference part, which limited their deployment in everyday objects. In this work, we are interested in enabling instant and customizable interactions with a low-cost setup and minimal maintenance through only embedding a single magnet. We integrate a gradient descent algorithm [29] with a particle filter [45] to estimate finger orientation under high magnetic distortion. Our approach enables real-time position tracking around the magnet.

Interaction Through Finger-Worn Devices

The fingers have been explored widely as an interaction medium since they convey human intent precisely with high flexibility. To understand the interaction created by fingers, finger-worn devices have been explored using various sensing techniques. First, a vision-based approach using a camera [6] and infrared proximity sensor [22, 46] provides 2D tracking as well as context-aware interactions. However, these techniques require maintaining a clear line of sight to the fingers. Other sensing techniques utilize magnetic hall sensors [7] and accelerometers [32]. Although these works suggest a new way of interacting through fingers, they do not focus on interacting with an object. Here, we enable interaction with a physical object by defining a coordinate system around an embedded magnet. With *TRing*, users are welcome to customize an interface with arbitrary objects.

SENSING TECHNIQUE PRINCIPLE

Previous 2D and 3D position tracking methods using magnetic sensing have been explored using multiple magnetometers [8], hall sensor array [7], or a 9-axis inertial measurement unit (IMU) [47]. However, these works require placing

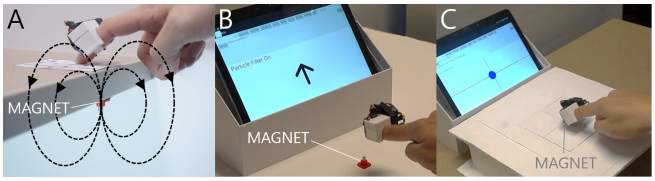

Figure 2. (A) We use a magnetic dip angle to detect magnetic distortion caused by the magnet. (B) Under magnetic distortion, we use a particle filter for orientation estimation. (C) By solving the magnetism equation with the estimated orientation, we achieve 3D fingertip tracking.

a magnetic source within the tracking body and a sensor installation in the reference object or platform. To this extent, all electronic hardware components should reside within the target object to support position tracking. The difficulty of embedding a magnet in the reference body remains twofold: 1) estimating the absolute orientation around the permanent magnet in spite of magnetic distortion and 2) recognizing the magnet's initial orientation. We explain how we overcome these bottlenecks through our tracking mechanism (Figure 2).

Orientation Estimation

A gradient descent algorithm provides a robust absolute orientation estimation using a 9-axis IMU [29]. However, this does not compensate for sudden changes in the magnetic field due to the slow responsiveness [45]. To provide robust orientation estimations under high magnetic distortion, a previous work [45] employed a particle filter [15, 23] along with a fast magnetic distortion detection based on magnetic dip angle. Outside of the high distortion region, we still adopt the gradient descent method because this method is computationally inexpensive in compensating for minor environmental distortions. As in previous works, we employ a sequential importance resampling/particle filter (SIR-PF) for orientation estimation under high magnetic distortion conditions. Thus, we maintain an accurate orientation both close to and far from the permanent magnet. The magnetic dip angle(θ_{dip}) is the angle between the lines of flux of the Earth's magnetic field and the surface of the Earth.

$$\theta_{dip} = (pi/2) - arccos(\vec{g} \cdot \vec{m}) \quad (1)$$

The dip angle reacts faster than the magnetic strength under magnetic distortion which is ideal for magnetic distortion detection. We compute θ_{dip} with readings from the accelerometer and magnetometer using Eq. 1. In our approach, we also include the rate of magnetic dip angle change ($\dot{\theta}_{dip}$) and the rate of magnetic strength change ($\dot{mag}$) for magnetic distortion detection. These parameters ensure that the distortion detection reacts against fast motions during finger interactions, the larger magnetic flux from the magnet, and the bigger sensor noises from constant motions.

Figure 3 illustrates the overall system flow of our proposed orientation estimation method:

1. Input accelerometer and magnetometer readings to calculate the magnetic dip angle using Eq. 1.
2. Detect magnetic distortion based on the following parameters: magnetic dip angle (θ_{dip}), threshold ($\theta_{min},\ \theta_{max}$), rate of magnetic dip angle change ($\dot{\theta}_{dip}$), magnetic strength threshold (mag_{init}), and rate of magnetic strength change ($\dot{mag}$).
3. Perform orientation estimation using the gradient descent method [29] if no severe magnetic distortion exists.
4. Under magnetic distortion, use particle filtering for orientation estimation [45] with full weight on the gyroscope readings.
5. Apply 0.5∼1 seconds delay before transitioning from the particle filter to the gradient descent method. This helps reduce erroneous transitions back to the gradient descent method in regions of magnetic distortion.
6. Obtain Euler angles from the computed quaternions.

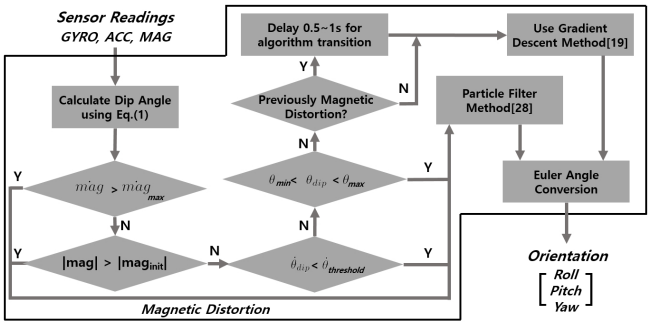

Figure 3. Orientation estimation system flow. Based on magnetic distortion status, we choose between the gradient descent algorithm and the particle filter for orientation computation.

In the SIR-PF, we choose quaternions to represent the orientation. Here, $\hat{q}_i^-$ and $\hat{q}_i^+$ represent prior and posterior estimated state at a time instant i, respectively. Then, we use the angular velocity (ω_i) from a gyroscope as input to the state space transition based on the quaternion rate ($\dot{q}$) as shown in Eq. 2 and Eq. 3.

$$\dot{q}_i = \frac{1}{2}\hat{q}_i^+ \begin{bmatrix} 0 \\ \omega_i \end{bmatrix} \qquad where, \quad \omega_i = [\omega_{ix}\ \omega_{iy}\ \omega_{iz}] \quad (2)$$

$$\hat{q}_{i+1}^- = \hat{q}_i^+ + \dot{q}_i \Delta t \quad (3)$$

The SIR-PF performs importance sampling and resampling recursively as following:

1. *Initialization:* Generate particles $q_i^j \sim p_{q_i}, i = 0, j = 1, \ldots, N$. q_i^j is the j^{th} particle at time instant i. In our case, we choose p_{q_i} as a uniform distribution centered at q_i. Further, we set the initial weights W_i^{j-} equally (N^{-1}).
2. *Updating Weights:* We update the weights using the likelihood. We first convert quaternion particles to direction cosine matrices (DCM) as shown in Eq. 4. Then, the columns of difference between the direction cosine matrices (DCM_{diff}^j) is used to calculate the axes of an ellipsoid. The ellipsoid volume is used for likelihood (L^j) computation (Eq. 5). Next, we update the posterior weight (W_i^{j+}) based on the prior weight (W_i^{j-}),

the likelihood (L^j), the transition probability ($p(q_i^j|q_{i-1}^j)$) and the importance density ($\pi(q_i^j|q_{i-1}^j, \omega_i)$) (Eq. 6). We choose one of the standard SIR-PF varieties, where $\pi(q_i^j|q_{i-1}^j, \omega_i) = p(q_i^j|q_{i-1}^j)$ [14].

$$DCM_{diff}^j = DCM_{q_i} - DCM_{q_i^j} \tag{4}$$

$$L^j := p(\omega_i|q_i^j) \propto \frac{1}{\Delta r_x \, \Delta r_y \, \Delta r_z} \tag{5}$$

$$W_i^{j+} = \frac{W_i^{j-} L^j p(q_i^j|q_{i-1}^j)}{\pi(q_i^j|q_{i-1}^j, \omega_i)} = W_i^{j-} \, L^j \tag{6}$$

3. *State Estimation:* The final estimated state $\hat{q}_i^+$ at time i is calculated as the weighted sum of the particles (Eq. 7).

$$\hat{q}_i^+ = \sum_{j=0}^{N} W_i^{j+} \, q_i^j \tag{7}$$

4. *Resampling:* Take N samples which are drawn from the set $\{q_i^j\}_{j=1,N}$ where the probability to take sample j is the updated weight W_i^{j+}. Then, reset the weights as $W_{i+1}^{j-} = N^{-1}$ for the next step.

5. *Prediction and Updating Particles:* To update each particle, we perform angular velocity sampling (a normal distribution with current gyroscope reading, ω_i, as the mean value, and the floor noise of the gyroscope as the standard deviation, σ_ω). In Eq. 8, the function $\mathcal{F}$ takes an angular velocity ω_i^j to update the particles.

$$q_{i+1}^j = \mathcal{F}(\hat{q}_i^+, \mathcal{N}\{\omega_i, \sigma_\omega\}) \tag{8}$$

6. Let $i := i + 1$ and iterate to item 2.

We apply full weights on the gyroscope readings whereas a previous work [45] employed partial weights on both magnetometer and gyroscope readings. By excluding the erroneous values from the magnetometer under distortion, our approach deals better with the higher magnetic flux (from a magnet) than that of previous works (ferromagnetic materials).

Position Tracking

From magnetism theory, the 3D position of the permanent magnet can be solved using the following equation:

$$\mathbf{H}(r) = \frac{K}{r^3}\left[\frac{3\mathbf{r}(\mathbf{m}\cdot\mathbf{r})}{r^2} - \mathbf{m}\right], r = |\mathbf{r}|, K = \frac{M}{4\pi} \tag{9}$$

H refers to the magnetic field vectors in the global frame, M denotes the magnetic moment, **m** is the directional vector of the magnet, and **r** is the magnet's location vector relative to the sensor. With known **m**, M, and **H**, **r** can be solved up to a sign ambiguity as follows:

Similar to *TMotion* [47], we first take the estimated orientations of **r** and 3-axis magnetometer readings (**H**). Then, we carry out a numerical approach to solve the nonlinear Eq. 9. Last, we remove the bias of the magnet orientation by fixating the magnet's orientation. Here, without loss of generality, we simply choose to use (0,0,1) as the directional unit vector, which means the north pole should face upward when it is installed on the reference object.

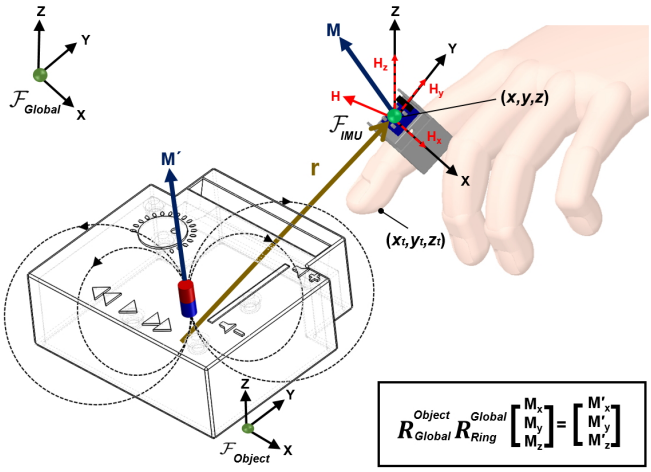

Figure 4. Estimated orientation is used to solve nonlinear magnetic equations. Magnetic direction vector (M) from the prototype is transformed to the object's frame (M') to provide relative position tracking.

Solving Eq. 9 with the above parameters leads to two possible solutions. We clear up this ambiguity by applying a half-plane restriction. Lastly, the computed magnet position enables the system to track the 3D position of the fingertip relative to the magnet in the object frame (Eq. 10).

$$\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + R_{Global}^{Object} R_{IMU}^{Global} T_{Fingertip}^{IMU} \tag{10}$$

To track the fingertip with respect to the magnet, we transform the directional unit vector from the IMU frame ($\mathcal{F}_{IMU}$) to the object frame ($\mathcal{F}_{Object}$) as shown in Figure 4. We further keep $\mathcal{F}_{Object}$ aligned with the local frame of the magnet. This allows us to maintain the input interface regardless of the object orientation. Moreover, we implement a simple gesture-based one-time calibration process which is explained in the **Software** section.

MAGNET PLACEMENT

In accordance with the tracking solution, we develop a magnet placement algorithm that computes the magnet installation location based on a user's interface customization. A key challenge is to find a location in which to place the embedded magnet such that all interface elements work. An algorithm for automatically determining the location of the magnets is explained and demonstrated using a proof-of-concept toolkit that supports 3D printed parts. To instrument pre-existing objects with our system, the magnet must be placed manually, but we can still compute the location if a 3D model is given.

Magnet Placement Formulation

Given a 3D triangular mesh model S and a set of n interface elements (I_i), the number (N) and locations (P_{M_j}) of magnets need to be determined. Each interface element I_i

Algorithm 1: Progressive Magnet Placement

Input: S is the input triangular mesh model; I_is is a set of user defined interface elements

Output: P_{M_j}s are a set of magnets associated with all interface elements

Function *ResultMagnetSet = FindMagnetPosition(S, I_is)*

```
MagnetPositionSet ← ∅;
InterfaceSet ← all I_is;
candidateSet ← ∅;
InitialUniformMeshResampling();
candidateSet ←
VertexClassification(S, InterfaceSet);
SortVertices(candidateSet);
while InterfaceSet ≠ ∅ do
    foreach candidate position P_Mj ∈ candidateSet do
        error = checkError(P_Mj, InterfaceSet);
        if error < δ then
            MagnetPositionSet+ = P_Mj;
            UpdateInterfaceSet(InterfaceSet);
return MagnetPositionSet;
```

is enabled by one magnet M_j, but multiple elements can be covered using the same magnet. Here, we describe the requirements for determining the magnet location and the corresponding objective function with constraints.

Number of Magnets To simplify the fabrication and assembly, the number of magnets N should be minimized.

Distance Deviation For each magnet M_j and its associated interfaces I_is, we define a function E_{M_j} which is the *mean squared error function* to measure the variation of distances between each interface element and magnet M_j.

Magnet Range Constraint From the viewpoint of magnet tracking, the distance $d_{i,j}$ between the interface I_i and its corresponding magnet M_j should satisfy the constraints defined in Eq. 11. Here, d_{min} and d_{max} are the minimum and maximum magnet sensing ranges respectively.

By combining all objective functions and constraints together, a constrained optimization can be formulated as

$$argmin_{[N,P_{M_1},P_{M_2},...,P_{M_N}]} \quad \sum_{j=1}^{N} E_{M_j}$$
$$s.t. \quad d_{min} < d_{i,j} < d_{max},\ i = 1, 2, ..., n,\ j = 1, 2, ...N. \tag{11}$$

Directly solving Eq. 11 could be challenging, as both the number and position of the magnets are unknown. Instead, we propose an algorithm to solve this issue by progressively adding magnets.

Progressive Magnet Placement Algorithm

The progressive magnet placement algorithm is shown in Algorithm 1. The input 3D model is represented by a triangular mesh model $S = (\mathcal{V}, \mathcal{E}, \mathcal{F})$. We choose to place the candidate magnets on the vertices for simplicity. Because the input mesh may have a highly non-uniform distribution of vertices, we conduct a uniform resampling on the mesh model. Then, we set up a candidate vertices set $V_c = \{\mathbf{v}_i | \exists j,\ s.t.\ d_{min} < d_{i,j} < d_{max}\}$. For the magnet position computation, we only consider vertices $\in V_c$. Next, we sort all candidate vertices in descending order according to the number of satisfied constraints. Subsequently, we start a loop to check each candidate vertex in this order and calculate the mean squared errors of the distance between this vertex and all the related interface elements. Once the mean squared error is below the preset threshold, δ, the position of this vertex is selected as a dedicated magnet position. Furthermore, a preset enclosure for housing the magnet is added onto the 3D model and oriented to the designated direction (north pole upward) automatically.

Afterwards, we iteratively check all interface elements to add magnets associated with them by repeating the aforementioned procedure. Based on the associated magnet position, the local coordinates of each interface element are computed. We transfer this information to a mobile device and map the function to each interface element.

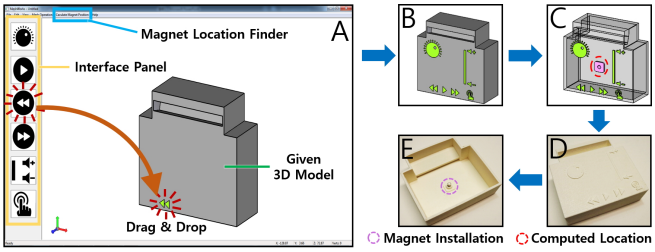

Figure 5. (A) Proof-of-concept toolkit where (B) user customizes interface. (C) The algorithm seeks for a magnet installation location and generates a magnet holder embedded 3D model. (D) With the 3D printed model, (E) user only needs to insert the magnet during the post-print assembly.

Proof-of-Concept Toolkit

As shown in Figure 5, we implement a proof-of-concept design toolkit. Users can customize different interface elements using a drag-and-drop approach. In the toolkit, we provide interface elements for a music player including a rotary knob, a linear slider, and buttons. We map each element to specific music player functions of the mobile device, such as volume control.

DESIGN RATIONALE

We consider several design factors. This helps us in formulating our design parameters to implement instant and customizable interactions with the target object.

Placement of the sensor Our approach requires sensor modules attached to the tracked part. We choose the index finger since it is most commonly used in surface computing control [11]. To use our approach, we need to keep the sensor module close to the interacting surface due to the limited tracking volume. Moreover, we need to limit the variations of the finger posture to achieve accurate fingertip tracking. We select the middle phalanx to satisfy these constraints while preserving the sense of touch in the fingertip.

Commitment Method To complete the interaction with the position tracking, a commitment action similar to a mouse-click is required. In a previous work [33], a physical tap was detected using 3-axis acceleration as the commitment method. Aligned with this approach, we also used the 3-axis accelerometer values to detect several motions including a physical tap, double-tap, and finger lift. This permits the use of finger motions in common with existing touch interfaces, such as touching a button to activate and lifting a finger to complete an adjustment of a slider or knob.

Magnet Selection A stronger magnetic dipole moment increases both the upper-limit and the saturation volume around the magnet. Here, sensor saturation prevents us from interacting very closely to the embedded magnet. Therefore, we need to choose the magnet size carefully to maximize the tracking volume while minimizing the saturation volume. To determine the proper magnet size, we measured the lower-limit and upper-limit of the magnetic sensing distance of different sizes. Based on our preliminary experiment, we chose a neodymium magnet (N42 grade) with a 5 mm diameter and 11.5 mm length. Using the magnet, we confirmed that no saturation occurs at $>$15 mm distance. Our system is capable of sensing a magnet at a 1500 mm range. With this small physical size, we can embed the magnet in most handheld objects seamlessly.

Object Identification To interact with a number of objects, the system should recognize different objects. Recent works have employed RFID technology for object identification [3], but this requires an attachment of an additional tag. To avoid using components other than the magnet, we focus on formulating object recognition through a simple gesture. Previous user-defined gesture sets for surface computing suggested a one-finger double-tap gesture as a candidate for an 'Open' task [44]. We employ this gesture since the procedure for selecting an object is similar to that of an 'Open' task. We use double-tapped locations to identify the object ID. Moreover, we can merge a calibration process into the same gesture to retrieve both the object's ID as well as the initial orientation.

IMPLEMENTATION

Figure 6 illustrates our prototype in detail. The overall prototype's dimensions are 35(W)$\times$22(D)$\times$25(H) mm. We consider the current dimensions as an upper limit: custom circuitry could be manufactured to be much smaller than the off-the-shelf modules we used in this prototype. We place the IMU at the center of the ring body and align it based on the given sensor axis. To provide a secure magnet attachment to an existing object, we provide a magnet holder (15(W)$\times$15(D)$\times$10(H) mm). Along with the holder, we use a cylinder-shaped, N42 grade, neodymium magnet with 5 mm in diameter and 11.5 mm in length.

Hardware

For our 9-axis IMU, we choose MPU-9250 due to its high magnetic sensing range and resolution ($\pm$4800 μT, 0.15 μT\bit). It also provides an accelerometer, gyroscope, and magnetometer in a small form factor (15 $\times$ 15 mm). A Bluetooth 4.0 Low Energy (BLE) module (Nordic nRF51822

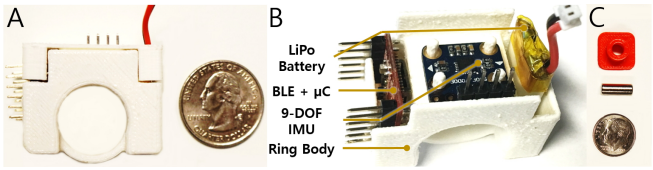

Figure 6. *TRing* (A) prototype and (B) its components. 9DOF-IMU is used to provide 3D tracking around the permanent magnet. (C) A magnet holder is provided for easy attachment to existing objects.

SoC) integrated microcontroller captures and transmits readings from sensors to the smartphone wirelessly. Each MPU-9250 requires a one-time soft+hard iron calibration for the magnetometer [4]. The accelerometer and the gyroscope are calibrated automatically in every power-up. We use a 110 mAh lithium polymer battery, which provides 6 hours of active operation with peak transmitting performance. We test and implement the prototype with two mobile devices: *LG Optimus G Pro* (1.7GHz quad-core, 2GB RAM) and *Galaxy Tab 10.1* (1.9GHz quad-core, 3GB RAM). This ensures that our approach does not require extra hardware for the tracking computation and it can be used in ubiquitous settings.

Software

In our prototype, the orientation is computed on a mobile device using mixed gradient descent and particle filter methods. The microcontroller streams nine channels of raw sensor values including 3-axis accelerations, angular velocities, and magnetic field readings. For tap and double-tap detection, we adopt the built-in algorithm from MPU-9250, which detects motion based on the accelerometer readings, dead time, and tap counts. A total of 19 bytes of data are streamed through the BLE module at a frequency of 45$\sim$50 Hz. The orientation estimation takes an average of $<$10 ms whereas the particle filter mode takes slightly longer ($\sim$5 ms). The total computation takes $<$20 ms on average, which results in an overall tracking rate of $>$30 Hz.

To compensate for geomagnetic field noises and to set the proper dip angle thresholds, we require a 2 second calibration. Here, we subtract the average magnetometer readings collected in a clean environment. To initiate interaction through *TRing*, the user performs a double-tap on the designated location. Upon performing the double-tap, the system acquires the object's orientation angles and the position of the tapped spot. Thus, we set the reference orientation and recognize the object through one double-tap action.

TECHNICAL EVALUATION

To fully understand the performance of our system, we devised our evaluation to investigate all attributes that affect system performance. In particular, we evaluated the accuracy of the orientation estimation, 3D position tracking, absolute targeting, and position tracking algorithms with different dwell time and speed. We used a *Galaxy Tab 10.1* for all experiments.

Experiment I: Orientation Estimation

We compared the angle difference between ground truth and the estimated values. We employed a rotating platform with

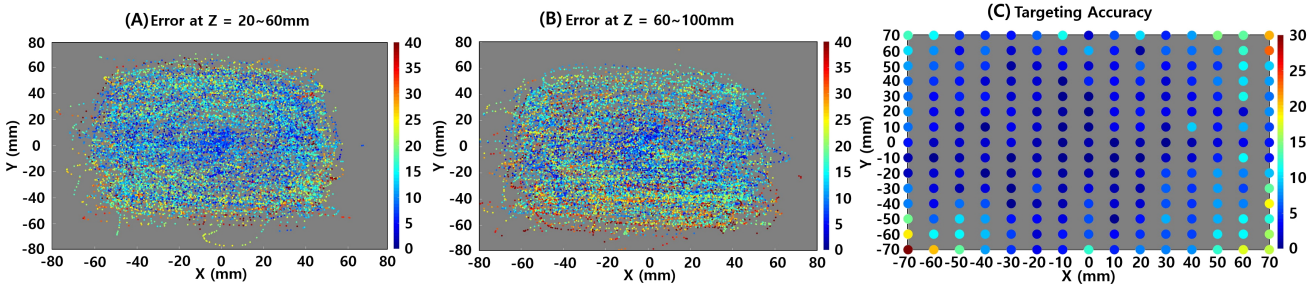

Figure 7. Tracking accuracy at (A,B) two different height ranges and (C) targeting accuracy are shown in mm. The color bar indicates the Euclidean distance between ground truth and our tracking. The magnet is located at the center (0,0). The mean tracking error is 8.6 mm in the area covering the conventional trackpad (120mm×120mm) with successful orientation estimation. In the same area, the mean targeting error is 4.96 mm.

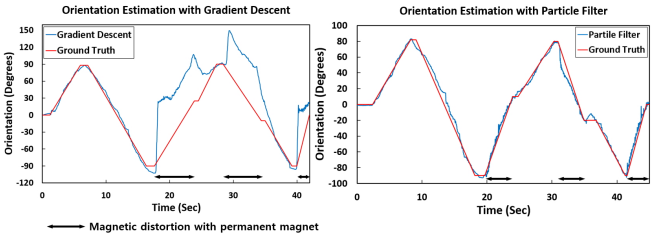

Figure 8. Heading estimation using gradient descent (Left) and gradient descent+particle filter (Right) during intermittent magnetic distortions.

a range of ±90° and a high angular velocity (>15 deg/s). We chose this range and speed to simulate plausible finger motion [19]. And we introduced magnetic distortion by intermittently placing the magnet near the prototype by hand.

Figure 8 illustrates the performance of the heading estimation under magnetic distortion. The gradient descent method showed high overall errors whenever magnetic distortion was introduced. On the other hand, the mixed method which combined gradient descent and particle filtering showed robust estimation throughout all sessions. Overall, the gradient descent approach worked well in a clean environment (RMS: 4.5°, Max: 10°), but performed poorly with high magnetic flux (RMS: 40°, Max: 100°). Using our approach, we achieved a comparable orientation estimation (RMS: 6°, Max: 15°) even under magnetic distortion.

Experiment II: 3D Position Tracking

For 3D position tracking measurement, we obtained ground truth using OptiTrack V120:Duo™. This device utilizes multiple cameras to capture the sub-millimeter movement of markers sampled at 120 Hz. We attached four markers on the testing platform to set up the reference plane and a single marker on the IMU's center. The magnet was affixed to the center of the reference plane. We attached the IMU to a plastic stick and moved it randomly in a 3D volume of ±80 mm (x-axis), ±80 mm (y-axis), and 20∼100 mm (z-axis). We used a minimum height of 20 mm in order to avoid magnetic saturation. We limited each set of motions to be less than 30∼40 seconds to remove bias from the particle filter performance. The ground truth data was saved on the desktop and the computed data was saved on a tablet. Both data sets were synchronized using a global time frame for off-line analysis. We collected a total of 30,213 data points.

In the total volume of 2,048,000 mm^3, the average Euclidean error was 15.61 mm (σ=9.05). There were two factors contributing to the high overall error: First was the outer region inaccuracy due to an increase in environmental magnetic field noises as the magnetometer moved away from the permanent magnet, as observed in previous works [8, 47]. Another factor was an orientation error caused by fast re-entry into the magnetic distortion region during the algorithm transition.

Considering only the data in the range of ±60 mm (x,y-axis) and 20∼60 mm (z-axis) with successful orientation estimation sessions (>95% of total data points), the overall error is only 8.6mm (σ=2.98). The mean error in the x-axis and y-axis were 4.44 mm (σ=3.12) and 4.38 mm (σ=3.04), respectively. This is sufficient to provide commonly-used touch interactions on planar surfaces. From the analysis, we noticed that the relative position tracking performed much better than the absolute tracking. By only comparing the deltas between data points, we observed errors lower than 3 mm in both the x-axis and y-axis. This implies that our prototype is also suited to fine cursor control.

Experiment III: Targeting Accuracy & Tracking Parameters

In order to evaluate the targeting accuracy and tracking with different parameters, ground truth was captured by tablet touch screen with virtual grids (10 mm spacing). This provided accurate ground truth data without occlusion. We attached a permanent magnet on the center of the tablet's backside and mounted our prototype on a 3D-printed fixture with a conductive tip. Before the evaluation, we confirmed that the magnetic field noise created by the tablet was minimal by comparing magnetometer readings with/without the tablet (>5000 data points). In order to measure the initial targeting accuracy, we collected data at each grid intersection point in ±60 × ±60 mm. To simulate all possible entering directions to the magnetic distortion region, we brought the prototype from left, right, up, down, and top. We collected a total of 42,250 data points (50 readings × 169 intersections × 5 directions). To measure the tracking with different parameters, we hovered the prototype randomly with different speeds on the tablet surface (Steady: 0 mm/s, Normal: > 100mm/s, Fast: > 200m/s) over at least 100 seconds.

As shown in Figure 7 (Right), the average targeting error was 4.96 mm (σ=3.1) in a region of ±60 mm (x-axis)× ±60 mm (y-axis). It showed better accuracy than the

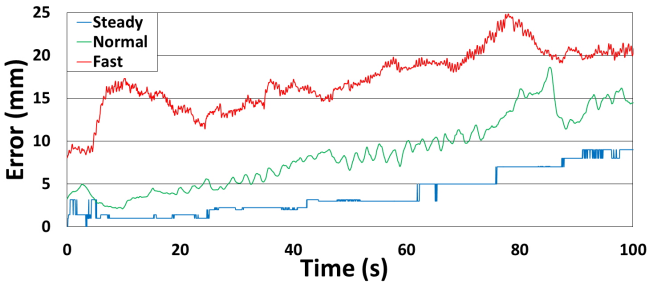

Figure 9. Tracking performance at different speeds and dwell times. Our prototype's low error rates at slower speeds indicate that it is suitable for use at typical speeds of finger movement.

position tracking experiment since the targeting always happened at the lower height (∼20 mm). With regard to the direction of entry, we did not observe any discrepancy in the accuracy. This demonstrates that *TRing* can be used with any finger motion.

As shown in Figure 9, higher motion speed and longer dwell time cause the tracking performance to deteriorate. This arises from the behavior of the particle filter, which accumulates errors due to the difficulty of dead reckoning under noisy sensor data. Based on this result, it is wise to limit the user interaction period to no more than 60 seconds to ensure a robust performance. This implies that the prototype would be a good fit for appliance/digital device control where average digital device interaction duration spans less than 60 seconds [2]. In terms of speed, our prototype can be used for general finger interaction since average finger-based control speed is within 150 mm/s [31].

TASK EVALUATION

To further verify the feasibility of our prototype for controlling basic user interfaces, we conducted two task evaluations. We explored user performance on absolute targeting and cursor control tasks. We recruited 12 male participants with a mean age of 28. The study duration was approximately 60 minutes. We took post surveys on the user experience as well as suggestions for system improvements and potential applications.

In our study, we set up a testing platform using a *Galaxy Tab 10.1* (Figure 10). We embedded a magnet in the center of the testing platform, which became the tracking coordinate origin. The participants wore our prototype on the middle phalanx of the index finger. We offered two ring body sizes with tape padding to ensure a good fit for all users. We measured each user's index finger to calibrate the space transformation parameters from the IMU to the fingertip. To maintain uniform performance, the initial calibration step was done before each task.

Task1: Targeting and double-tap calibration accuracy

We carried out two tasks: absolute targeting and double-tap calibration. We looked at whether the targeting accuracy was consistent across different user behaviors including different approaching orientations, speeds, and finger postures. The task aims to verify the accuracy of absolute position targeting,

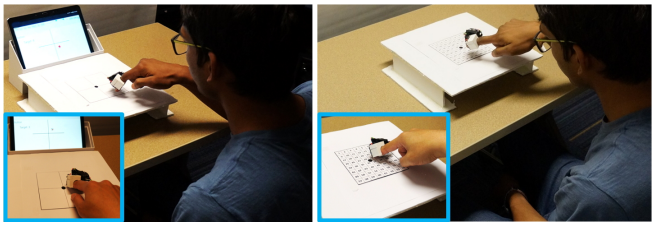

Figure 10. The overall study setup including (Left) targeting/double-tapping and (Right) cursor control tasks.

which can be used for placing interface elements relative to the magnet. The double-tap calibration task verifies whether we can use tapped information to set reference orientation as well as to recognize different objects.

Setup

Participants were asked to place and hold their fingertip on the printed grids, and the positions were then logged by researchers. We chose a grid size of 15 × 15 mm, the size of an average keycap. We guided users to approach from the outside of the magnetically distorted region for each trial. This ensured that each trial condition was independent, including the user's approaching motion and initial sensor conditions. A total of 768 data points were collected (64 positions × 12 users). Subsequently, the users carried out a double-tap and held their fingertip on the designated grid. During the double-tap task, we fixed the heading direction of the user's finger to compare it with ground truth. On a successful double-tap, we logged the orientation angles and computed the fingertip positions. A total of 192 data points were collected (16 positions × 12 users).

Results

Figure 11 shows that the error rates for targeting and double-tap increased with distance. This behavior was expected due to the degradation of the tracking performance over distance. The double-tap task exhibited a higher error rate than the targeting task at greater distances. We observed that the double-tap gesture often caused a fast in-and-out motion near the magnetic distortion boundary. Since our approach requires 0.5∼1 seconds delay for stabilization during transitions to or from magnetic distortion regions, this specific motion causes higher orientation and position errors. Within 45 mm distance from the embedded magnet, the overall accuracy was 92%, with 87% for the targeting and double-tap tasks. Meanwhile,

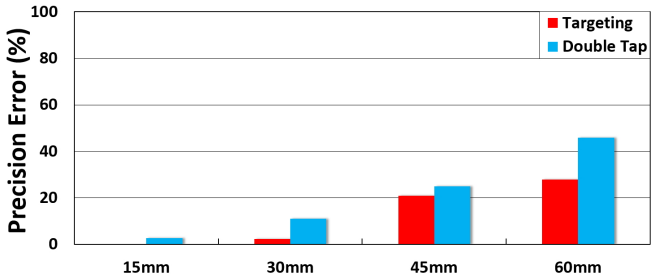

Figure 11. The position error rate for targeting and double-tap tasks for different distances.

the heading angle only showed a 3.24° error, small enough to be used in calibration.

Task2: Cursor control accuracy and reaction time

In Task 2, we evaluated users performing an absolute cursor control task. Since our goal was to evaluate user performance under different commitment methods and distances, we adopted a similar study design to previous work [7]. We employed surface tap and bimanual touch as a baseline and an upper-bound commitment method, respectively. Throughout the study, we explored how small a target and how far from the magnet users can control.

Setup

The study design for the task was 2×4×4×8 (Commitment Method×Distance×Target Size×Target Position). The participants were asked to move the cursor to the target region using a given commitment method. The target sizes were 3, 5, 10, and 15 mm. We categorized the distance into two groups: *Near* (0∼15 mm) and *Far* (30∼45 mm). We changed the cursor color when the cursor was located within the target. A random target was shown when users approached the magnetically distorted region. We randomized the order of the task parameters except for the commitment method, which was counterbalanced. We recorded task error, completion time, and all sensor data in each trial.

Results

For the error rate and completion time, we found a significant interaction effect between [Distance×Commitment Method]. Furthermore, there was a significant difference in the error rate for different target sizes.

For distance parameters, a pairwise t-test showed a significant difference in the error rate during tap commitment when the target size was below or equal to 10 mm. There was no significant difference in completion time caused by distance. The error was higher in the Far distance condition since participants had to carry out the tap gesture under greater tracking noise. For designers using our toolkit, we explicitly avoided fine-grained user interface components at far distances. For example, we limited slider resolution or button size according to the distance from the magnet. Furthermore, the results imply that a robust commitment method has the potential to minimize performance degradation due to distance.

A pairwise t-test shows that there is a significant difference in error rate between commitment methods. Furthermore, the bimanual touch method is significantly faster than tap gesture only in the Far distance condition. Performance can be improved by applying a user-specific calibration for the tap detection. In each user's accelerometer readings, we observed a difference in tap accelerations among different participants. For design improvements, we recommend setting a unique tap threshold for different users.

We carried out a pairwise t-test to explore the relationship between target sizes and overall performance. For both error and completion time, the smallest target size (3 mm) was significantly different from the other sizes regardless of distance and commitment method. There was a significant difference

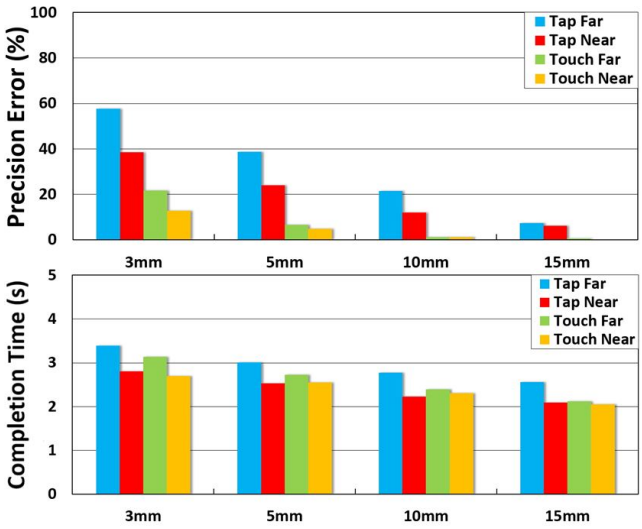

Figure 12. The error rate (Top) and completion time (Bottom) on different target sizes with tap and bimanual touch commitment methods under near and far distances from the magnet.

in error rates for tap commitment at the 5 mm target size when compared to bigger sizes. Thus, the results indicate that the target size should exceed 5 mm for robust performance.

Summary

From each task condition, that is, *Distance*, *Commitment Method*, and *Target Size*, we gained design insights for our prototype. First, the *Distance* parameter showed an increase in errors when combined with an inaccurate commitment method. However, increasing the target size (10 mm or higher) still provided a robust performance. Secondly, we observed that the *Commitment Method* significantly affected the performance. So, based on the user feedback, we designed alternative *Commitment Methods*. These were 1) 1∼2 seconds hold motion near the surface and 2) tap or touch on the side of the prototype using the thumb. The suggested methods can be instantly adopted without modifying the physical hardware or requiring a two-handed operation. Last, the target size should exceed 5 mm. Even with a bimanual touch commitment, smaller targets resulted in excessive error rates.

User feedback

We elicited qualitative feedback about the experience with our prototype in a post-survey. Participants were surprised by the fact that there was only a small magnet under the testing jig. They also reported that our prototype was easy to learn given their experiences with conventional trackpads. For improvements, participants recommended adding haptic feedback for commitment confirmation and to support for various fingers. For potential applications, participants suggested that our prototype would be a good fit for appliance control (TV, audio system, and indoor light) and eyes-free control using the fingers. We demonstrated some applications from these suggestions.

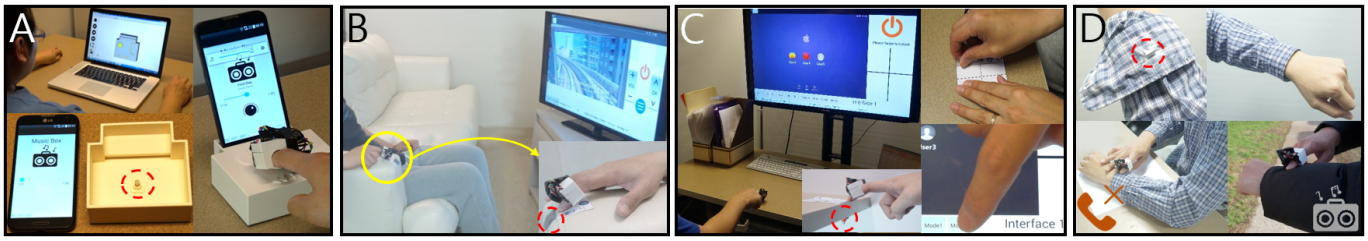

Figure 13. Example applications showing the capability of *TRing*. (A) Rapid prototyping of interactive object. (B) Making interactive furniture by attaching a magnet. (C) Easy customization process without requiring hardware changes. (D) Wearable interface connected to mobile device. (Red circles indicate the embedded magnets.)

INTERACTION TECHNIQUES

We devise our interaction techniques into continuous and discrete inputs. While discrete inputs support event-driven approaches, continuous inputs utilize on-going tracking to control a given interface. Thus, we combine fingertip tracking with motion events as an input metaphor.

Discrete Input

Button & Gesture By merging position tracking with commitment method, the system receives input signals with local coordinate information. This enables button interface around the embedded magnet such that distinctive functions are mapped to each button. To prevent false triggering, we discriminate between commitment events occurring near the surface from those occurring in mid-air using the 3D fingertip position. We can also detect a simple gesture based on the tracking history. Furthermore, more complex finger gestures are feasible with trained data sets.

Proximity & Tilt The detection of magnetic distortion using a dip angle provides a robust cue to determine the proximity of the tracked finger. Using proximity alone, we can instantiate the virtual interface as the user's finger approaches the object. Proximity can also be used to prevent false triggering by disregarding any input signals when the user's finger is out of the interaction region. On the other hand, *TRing* can sense a finger tilting under magnetic distortion. In combination with proximity sensing, a robust mode-shift function is achievable. For example, users can open different menus in a TV interface by simply tilting their fingers.

Continuous Input

Cursor Control *TRing* offers cursor control on an object's surface or in mid-air. This is achievable by utilizing continuous real-time position tracking. This technique is especially helpful when users are depending heavily on visual feedback to control a virtual interface. For instance, users do not focus on the object surface when controlling the virtual interface. During cursor control, the commitment method can be used to trigger a virtual interface.

Linear & Rotary Slider To implement linear and rotary sliders, we employ a multi-level trigger along with continuous position tracking. An on/off flag is needed to initiate and end continuous control. Here, we use a tap gesture in a slider region to initiate, and a finger lift to end the interaction. The linear and rotary sliders support interactions in both relative and absolute manners. The slider changes values based on the initial position in a relative mode whereas the value of a slider is fixed to the specific position in the absolute approach.

EXAMPLE APPLICATIONS

We demonstrate four applications to show the capability of using *TRing* in four contexts: rapid prototyping of interactive objects, bringing interactivity to existing objects, personalized interfaces, and wearable computing.

3D Printed Music Box Previously, post-print assembly was required to embed sensing capability in 3D printed parts [30]. Figure 13(A) demonstrates a simple process to bring interactivity to a 3D printed object through our approach. Here, users insert a magnet for building a music box with various controls including buttons, linear sliders, and rotary sliders. Combining 3D fingertip tracking with commitment methods such as tap and finger lift, we successfully operate all controls using *TRing*. We demonstrate a potential to improve the rapid prototyping process of interactive objects.

Furniture Remote Controller Cheng et al. [9] suggest that everyday objects are a good fit for auxiliary and instant interface to minimize distracting context-switching. To this end, we implement a TV remote control in existing furniture by attaching a magnet. As shown in Figure 13(B), we implement general functions provided in existing remote controllers. Furthermore, we implement a finger posture control which launches a *Gallery* by tilting a finger within the control region. This example showcases the simple approach of making existing objects act as an auxiliary and instant interface.

Personalized Office Desk To make a change in the interface, a previous approach [30] required users to go through hardware rearrangement. As shown in Figure 13(C), users can customize the interface with simple steps using *TRing*. They can 1) print a new cover which works as an overlay for an updated interface and 2) update local coordinates of the interface. We implement a personalized desk interface that provides office environment control with password typing. Here, a user can relocate the interface elements and change the password typing method to a swipe gesture using these two steps. This example illustrates *TRing*'s flexibility in interface customization.

Wearable Controller Existing wearable interface for clothing either requires placing circuit boards sewn with conductive threads [5] or a whole sensor array [37]. The high durability of magnets against heat (up to 80°C) and water makes a good fit for use with clothing. As shown in Figure 13(D),

a small magnet is placed in an unobtrusive manner. We employ a diametrically magnetized magnet to reduce the magnet's thickness. We demonstrate controlling a mobile device using a magnet embedded inside a shirt. This illustrates the potential of our prototype for use with mobile interaction.

LIMITATION AND FUTURE WORK

We show that *TRing* utilizes 3D position tracking to bring interactivity to objects with a simple customization process. We believe that the reduced post-processing stage for making an object interactive will benefit interaction designers and novice "makers" who do not possess deeper knowledge of sensors or electronic components. Here, we discuss the limitations and future work.

Magnetic sensing challenges

Our tracking exploits magnetic sensing. This suggests that *TRing* will not work properly with ferromagnetic materials. However, some interaction techniques including tilting and proximity can still be used. Since most 3D printers employ plastic materials, 3D printed parts would be a good fit for use with *TRing*.

Due to the constraints of our sensing mechanism, we require a magnet installation in a specific orientation (north pole facing upward). Also, the half-plane restriction used in the position tracking algorithm requires all interface elements to be on the same side of the magnet's polarity. In theory, one could eliminate these restrictions by employing multiple magnetometers [17] to obtain the magnet's orientation. With the known magnet orientation, no limitation will be imposed on the configuration of the magnet installation.

Scalability

Our system's current interaction area is limited to a 120×120 mm region with a single magnet. Although this is enough for making a small interactive object, it is not sufficient for larger objects such as a big table. If we can obtain the embedded magnet's orientation in future work, we can identify each magnet using a unique orientation. In this way multiple magnets can be employed for enlarging the interaction volume without requiring manual identification.

Our approach recognizes different objects through gesture, which is sufficient to implement the usage scenarios shown in our demonstration. However, the gesture approach cannot scale to the notion of hundreds of objects becoming interactive. A potential approach would be employing magnetic-RFID tag [27].

Due to the degradation in tracking performance over time, *TRing* is best suited for controlling tasks which require relatively short engagement. In the future, we will investigate different particle filter resampling techniques for potential improvement. Moreover, designing UIs with a larger target size helps maintain a robust performance over longer durations.

Commitment method

In this study, we only explored a tap gesture as a commitment method. Since robust commitment methods can improve the overall performance, we would like to explore different methods. A first step would be to verify the performance of classifying different motion events based on a 3-axis accelerometer with *TRing* [33]. In our user feedback, we received many suggestions to use the ring body as an interaction medium for the commitment method. Therefore, it is in our interest to develop commitment methods utilizing the ring body, such as adding a capacitive touch area.

CONCLUSION

We have proposed an instant and customizable interaction mechanism through a finger-worn device. Employing a magnetic sensing technique with a particle filter, we obtain 3D fingertip tracking around the magnet. To this extent, we bring interactivity to objects by simply embedding a magnet. Through evaluations, we have verified system accuracy (8.6 mm in 3D space) as well as user performance (button: 92% accuracy, cursor control: 91% accuracy). We also showcase our approach of making various objects interactive with high customization flexibility. We believe that our work will benefit novice interaction designers as well as general users who want to quickly implement a personalized physical interface without deeper knowledge of electronic components.

ACKNOWLEDGEMENTS

We gratefully acknowledge the valuable feedback from the Associate Chairs and the anonymous reviewers. This work was supported by the National Science Foundation (NSF) IGERT Grant #1144843. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views or opinions of the funding agency.

REFERENCES

1. Anderson, C. Makers: The new industrial revolution. 2012. *New York: Crown Business*.
2. Andrews, S., Ellis, D. A., Shaw, H., and Piwek, L. Beyond self-report: Tools to compare estimated and real-world smartphone use. *PLOS one 10*, 10 (2015), e0139004.
3. Berlin, E., Liu, J., Van Laerhoven, K., and Schiele, B. Coming to grips with the objects we grasp: Detecting interactions with efficient wrist-worn sensors. In *Proc. TEI '10*, ACM (2010), 57–64.
4. Bouchier, P. Razor 9DOF AHRS, 2013. Retrieved August 1, 2015 from `https://github.com/ptrbrtz/razor-9dof-ahrs`.
5. Buechley, L., Eisenberg, M., Catchen, J., and Crockett, A. The LilyPad Arduino: Using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proc. CHI '08*, ACM (2008), 423–432.
6. Chan, L., Chen, Y.-L., Hsieh, C.-H., Liang, R.-H., and Chen, B.-Y. CyclopsRing: Enabling whole-hand and context-aware interactions through a fisheye ring. In *Proc. UIST '15*, ACM (2015), 549–556.

7. Chan, L., Liang, R.-H., Tsai, M.-C., Cheng, K.-Y., Su, C.-H., Chen, M. Y., Cheng, W.-H., and Chen, B.-Y. FingerPad: Private and subtle interaction using fingertips. In *Proc. UIST '13*, ACM (2013), 255–260.
8. Chen, K.-Y., Lyons, K., White, S., and Patel, S. uTrack: 3D input using two magnetic sensors. In *Proc. UIST '13*, ACM (2013), 237–244.
9. Cheng, K.-Y., Liang, R.-H., Chen, B.-Y., Laing, R.-H., and Kuo, S.-Y. iCon: Utilizing everyday objects as additional, auxiliary and instant tabletop controllers. In *Proc. CHI '10*, ACM (2010), 1155–1164.
10. Corsten, C., Avellino, I., Möllers, M., and Borchers, J. Instant user interfaces: Repurposing everyday objects as input devices. In *Proc. ITS '13*, ACM (2013), 71–80.
11. Epps, J., Lichman, S., and Wu, M. A study of hand shape use in tabletop gesture interaction. In *Proc. CHI EA '06*, ACM (2006), 748–753.
12. Feldman, A., Tapia, E. M., Sadi, S., Maes, P., and Schmandt, C. ReachMedia: On-the-move interaction with everyday objects. In *Proc. ISWC '05*, IEEE (2005), 52–59.
13. Gao, W., Zhang, Y., Nazzetta, D. C., Ramani, K., and Cipra, R. J. RevoMaker: Enabling multi-directional and functionally-embedded 3D printing using a rotational cuboidal platform. In *Proc. UIST '15*, ACM (2015), 437–446.
14. Gordon, N. J., Salmond, D. J., and Smith, A. F. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F-Radar and Signal Processing*, vol. 140, IET (1993), 107–113.
15. Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing 50*, 2 (2002), 425–437.
16. Harrison, C., Xiao, R., and Hudson, S. Acoustic barcodes: Passive, durable and inexpensive notched identification tags. In *Proc. UIST '12*, ACM (2012), 563–568.
17. Hu, C., Li, M., Song, S., Yang, W., Zhang, R., and Meng, M. Q. A cubic 3-axis magnetic sensor array for wirelessly tracking magnet position and orientation. *Sensors Journal, IEEE 10*, 5 (2010), 903–913.
18. Huang, J., Mori, T., Takashima, K., Hashi, S., and Kitamura, Y. IM6D: Magnetic tracking system with 6-DOF passive markers for dexterous 3D interaction and motion. *ACM Transactions on Graphics 34*, 6 (2015), 217.
19. Hume, M. C., Gellman, H., McKellop, H., and Brumfield, R. H. Functional range of motion of the joints of the hand. *The Journal of Hand Surgery 15*, 2 (1990), 240–243.
20. Ishii, H., and Ullmer, B. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proc. CHI '97*, ACM (1997), 234–241.
21. Khalfin, I., and Jones Jr, H. S. Electromagnetic position and orientation tracking system with distortion compensation employing wireless sensors, Apr 2002. US Patent 6,369,564.
22. Kienzle, W., and Hinckley, K. LightRing: Always-available 2D input on any surface. In *Proc. UIST '14*, ACM (2014), 157–160.
23. Koo, W., Sung, S., and Lee, Y. J. Error calibration of magnetometer using nonlinear integrated filter model with inertial sensors. *IEEE Transactions on Magnetics 45*, 6 (2009), 2740–2743.
24. Kranz, M., Holleis, P., and Schmidt, A. Embedded interaction: Interacting with the internet of things. *Internet Computing, IEEE 14*, 2 (2010), 46–53.
25. Li, H., Ye, C., and Sample, A. P. IDSense: A human object interaction detection system based on passive UHF RFID. In *Proc. CHI '15*, ACM (2015), 2555–2564.
26. Liang, R.-H., Kuo, H.-C., Chan, L., Yang, D.-N., and Chen, B.-Y. GaussStones: Shielded magnetic tangibles for multi-token interactions on portable displays. In *Proc. UIST '14*, ACM (2014), 365–372.
27. Liang, R.-H., Kuo, H.-C., and Chen, B.-Y. GaussRFID: Reinventing physical toys using magnetic RFID development kits. In *Proc. CHI '16*, ACM (2016), 4233–4237.
28. MacDonald, E., Salas, R., Espalin, D., Perez, M., Aguilera, E., Muse, D., and Wicker, R. 3D printing for the rapid prototyping of structural electronics. *IEEE Access 2* (Dec 2014), 234–242.
29. Madgwick, S. O., Harrison, A. J., and Vaidyanathan, R. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics*, IEEE (2011), 1–7.
30. Mellis, D. A. Do-It-Yourself fabrication of electronic devices. *IEEE Pervasive Computing 13*, 3 (2014), 22–29.
31. Microsoft. MSDN library, 2013. Retrieved Mar 30, 2016 from `https://msdn.microsoft.com/en-us/library/dn915123(v=vs.85).aspx`.
32. Nguyen, A., and Banic, A. 3DTouch: A wearable 3D input device for 3D applications. In *Proc. VR '15*, IEEE (2015), 55–61.
33. Nirjon, S., Gummeson, J., Gelb, D., and Kim, K.-H. TypingRing: A wearable ring platform for text input. In *Proc. MobiSys '15*, ACM (2015), 227–239.
34. Ono, M., Shizuki, B., and Tanaka, J. Touch & Activate: Adding interactivity to existing objects using active acoustic sensing. In *Proc. UIST '13*, ACM (2013), 31–40.

35. Peng, H., Mankoff, J., Hudson, S. E., and McCann, J. A layered fabric 3D printer for soft interactive objects. In *Proc. CHI '15*, ACM (2015), 1789–1798.

36. Pohl, H., and Rohs, M. Around-device devices: My coffee mug is a volume dial. In *Proc. MobileHCI '14*, ACM (2014), 81–90.

37. Rekimoto, J. Gesturewrist and Gesturepad: Unobtrusive wearable interaction devices. In *Proc. ISWC '01*, IEEE (2001), 21–27.

38. Savage, V., Chang, C., and Hartmann, B. Sauron: Embedded single-camera sensing of printed physical user interfaces. In *Proc. UIST '13*, ACM (2013), 447–456.

39. Savage, V., Follmer, S., Li, J., and Hartmann, B. Makers' Marks: Physical markup for designing and fabricating functional objects. In *Proc. UIST '15*, ACM (2015), 103–108.

40. Savage, V., Head, A., Hartmann, B., Goldman, D. B., Mysore, G., and Li, W. Lamello: Passive acoustic sensing for tangible input components. In *Proc. CHI '15*, ACM (2015), 1277–1280.

41. Schmitz, M., Khalilbeigi, M., Balwierz, M., Lissermann, R., Mühlhäuser, M., and Steimle, J. Capricate: A fabrication pipeline to design and 3D print capacitive touch sensors for interactive objects. In *Proc. UIST '15*, ACM (2015), 253–258.

42. Tanenbaum, J. G., Williams, A. M., Desjardins, A., and Tanenbaum, K. Democratizing technology: Pleasure, utility and expressiveness in diy and maker practice. In *Proc. CHI '13*, ACM (2013), 2603–2612.

43. Voxel8. 3D electronics printer, 2014. Retrieved Febuary 1, 2016 from https://www.voxel8.co.

44. Wobbrock, J. O., Morris, M. R., and Wilson, A. D. User-defined gestures for surface computing. In *Proc. CHI '09*, ACM (2009), 1083–1092.

45. Yadav, N., and Bleakley, C. Accurate orientation estimation using AHRS under conditions of magnetic distortion. *Sensors 14*, 11 (2014), 20008–20024.

46. Yang, X.-D., Grossman, T., Wigdor, D., and Fitzmaurice, G. Magic Finger: Always-available input through finger instrumentation. In *Proc. UIST '12*, ACM (2012), 147–156.

47. Yoon, S. H., Huo, K., and Ramani, K. TMotion: Embedded 3D mobile input using magnetic sensing technique. In *Proc. TEI '16*, ACM (2016), 21–29.