2D Input for Virtual Reality Enclosures with Magnetic Field Sensing

Kent Lyons
Technicolor Research
175 S. San Antonio Rd. Suite 200, Los Altos, CA 94022
kent.lyons@technicolor.com

ABSTRACT

Virtual Reality enclosures are inexpensive devices that create a virtual reality experience using a mobile phone. For example, Google Cardboard lets a user put their phone inside the device and is made from cardboard, some simple lens and a magnet. Because the phone is encased, there is no way to interact with the touch screen. Cardboard uses its magnet, the phone's magnetometer and an algorithm to create a binary input [10]. In this paper, we extend the capabilities of these devices to provide continuous 2D input. In particular, we use magnetic field sensing to track the magnet in 2D on the side of the enclosure. We provide background on magnetic field sensing and show how it can apply to a VR enclosure. We examine several parameters that impact calculating the magnet's position and we focus on the challenge of dealing with the ambient geomagnetic field. Finally, we present a solution which uses the phone's inertial sensors and some user interactions that take advantage of 2D input with the magnet.

Author Keywords

Magnetic field sensing; input; Google Cardboard; virtual reality

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation: User Interfaces. - Graphical user interfaces

INTRODUCTION

Virtual Reality (VR) enclosures for smartphones have become an inexpensive way to provide simple VR experiences to end users. Such enclosures are relatively cheap and therefore allow users to gain access to virtual reality without investing in dedicated hardware. One challenge with these types of devices is how to obtain input from the user. The phone is surrounded by the enclosure so the touch screen is not available. Likewise, the inertial sensors are used to track head rotations and are largely unavailable for direct user input. An additional complication is in keeping the enclosure as simple and cheap as possible. For example, while it would be possible to incorporate a touch panel in the side of the enclosure (similar to Google Glass), doing so would make the enclosure more expensive and complex because it would require adding electronics to a device that is otherwise just a simple mechanical housing and cheap lenses.

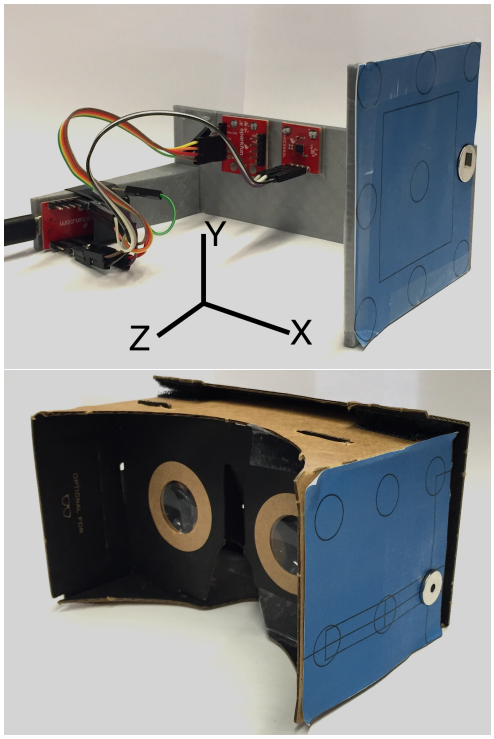

Figure 1. Our test apparatus with HMC588L magnetometer, LSM6DS3 accelerometer/gyroscope and Arduino (top). A Google Cardboard modified to allow the magnet to move across the entire side panel (bottom).

The first generation of Google Cardboard offers a unique solution to address the problem of input. Cardboard uses permanent magnets placed on the enclosure, and with the phone's magnetometer, it senses the user sliding a magnet to create a binary button press input [10]. While this approach is effective, it offers very minimal input capabilities.

In this paper, we extend the idea of using a magnet attached to the outside of a VR enclosure, but provide for continuous 2D input. As we will detail, previous work has tracked a magnet

ISWC '16, September 12–16 2016, Heidelberg, Germany

ISBN 978-1-4503-4460-9/16/09...$15.00
DOI: http://dx.doi.org/10.1145/2971763.2971787

using multiple magnetometers. However, only one of these sensors is available in smartphones. We provide an analysis of parameters impacting magnetic field (MF) sensing for VR enclosures and highlight the issue of accounting for the ambient geomagnetic field. We demonstrate an approach which uses a single magnetometer and the other sensors available in mobile phones. We implement this using stand alone sensing hardware, as well as on an iPhone inside a Unity3D cardboard app. Finally, we propose some interactions that utilize 2D magnetic field tracking.

RELATED WORK

There has been a variety of work that uses magnetometers to track a magnet for direct user input. Smus and Riederer present an algorithm for detecting the movement of a magnet to simulate a button press for Google Cardboard [10]. Others looked at extracting simple parameters out of magnet movement. Abracadabra [7] allows input of a continuous 1D parameter by tracking the orientation of the magnetic field. It also provides a click input by monitoring the transition between polarities which causes an inversion in field orientation. Nenya uses a magnetic ring that is rotated about the finger and also provides a continuous 1D input [1].

There is other research which offers more degrees of freedom. For example, GaussSense uses a dense 2D array of Hall Effect sensors to infer the position of a magnet [9]. uTrack in contrast uses two magnetometers and a brute force search algorithm to track the position of a magnet in 3D [2]. Other work extends this line of research looking at different configurations and more efficient solutions (TMotion [13]) or the ability to track multiple magnets (Finexus [3]).

The research of Han *et al.* is close to our target solution [5, 6]. In particular, they present a system that tracks the 2D position of a magnet. Their approach has the constraint that the magnet must move in a single plane and tracking degrades with rotations or movement out of that plane. These constraints fit well with our target use case. In particular, our magnet will move on the surface of the VR enclosure and thus will only move in 2D. Unfortunately, Han *et al.* use two magnetometers for their solution. The second magnetometer is used to cancel out the impact of the Earth's magnetic field. A key contribution of our work is to provide similar 2D tracking of a magnet; however, we do not use a second magnetometer.

MAGNETIC FIELD SENSING

Magnetometers sense magnetic fields and have become cheap and ubiquitous with the rise of smartphones. These sensors are capable of measuring the strength of the magnetic field vector $\hat{\boldsymbol{H}}$ on three orthogonal axes. However in a phone, they are typically used as a compass. They measure the Earth's magnetic field but only the orientation of the field is of interest. The magnitude of the magnetic field is discarded. Abracadabra [7] and Nenya [1] similarly only use orientation.

Fundamentally, a magnetic field $\boldsymbol{H}$ (μT) can be decomposed into two orthogonal component vectors, tangential (H_r) and radial (H_θ):

$$H_r = \frac{2K\cos\theta}{r^3}, H_\theta = \frac{K\sin\theta}{r^3}, K = \frac{M}{4\pi} \quad (1)$$

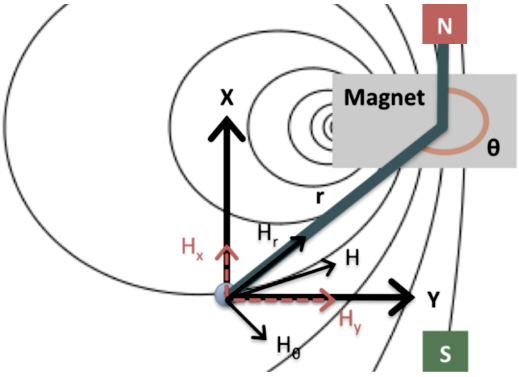

Figure 2. A magnet generates a field composed of the tangential (H_r) and radial (H_θ) components. A sensor placed distance r away at the angle θ measures this field, $\hat{\boldsymbol{H}}$, in three dimensions $[\hat{H}_x, \hat{H}_y, \hat{H}_z]$.

where K is a constant (in units of $\mu T \cdot cm^3$ for this paper) related to the magnetic moment and depends on the specific permanent magnet used [8]. r (cm) is the distance from the magnet to the sensor and θ is the angle from the north pole of the magnet to the sensor (Figure 2). The magnetic field is two dimensional because it is rotationally symmetric about the magnetic pole. Using trigonometry we can convert this polar representation into a Cartesian one:

$$r = \sqrt{x^2 + y^2}, \cos\theta = -x/r, \sin\theta = -y/r \quad (2)$$

$$H_x = H_r\cos\theta - H_\theta\sin\theta = K \cdot \frac{2x^2 - y^2}{r^5} \quad (3)$$

$$H_y = H_r\sin\theta + H_\theta\cos\theta = 3K \cdot \frac{xy}{r^5} \quad (4)$$

which allows us to calculate a 2D coordinate $[x, y]$ for our user interface from the magnetic field reading $[H_x, H_y]$.

Note that this derivation assumes the magnet and sensor are aligned with the pole parallel to the x-axis and $H_z = 0$. If the configuration is different, the known rotation T needs to be applied to the raw sensor readings $\hat{\boldsymbol{H}} = [\hat{H}_x, \hat{H}_y, \hat{H}_z]$ such that $\boldsymbol{H} = T\hat{\boldsymbol{H}} = [H_x, H_y, 0]$. Similar derivations can be found in Han *et al.* [5, 6] and Chen *et al.* [2].

Application to Cardboard

The basic idea we present in this paper is to use the side of the VR enclosure for 2D input. The user moves the magnet across the surface similar to a trackpad. However with VR enclosures, different phones have different sizes and make different design choices for their electronics. As such, the position of the magnetometer relative to the VR enclosure's magnet will vary between phone models. This geometry is fixed for a given phone and enclosure pair and could be specified with a one time configuration (for instance looking up the key geometry values in a database).

In this paper, instead of using several different phones, we use the 3D printed test apparatus shown in Figure 1 (top) that allows us to control for the position of the magnetometer relative to the interaction surface. The design of this apparatus mimics a VR enclosure for a phone (Figure 1, bottom).

The interaction surface where the magnet moves is the right hand vertical surface (the YZ plane). In a VR enclosure, the phone is mounted in the back of the device (away from the user) in the XY plane. Our test apparatus has a bar that allows us to mount a magnetometer in a similar configuration. Furthermore, we can place the magnetometer in one of three different locations for our experiments which allows us to have a distance from the interaction surface (along the x axis) of 1.5cm, 7.0cm, or 12.5cm. Our interaction surface has an area of 8cm x 8cm and the magnet can be moved vertically (y between -3.5cm and +4.5cm) and horizontally (z between 8cm and 0cm). Finally, our test apparatus uses a stand alone magnetometer (Honeywell HMC5883L) and accelerometer/gyroscope (STMicroelectronics LSM6DS3). We connect directly to the sensors with I2C and have access to the raw data and all of the sensor settings.

Given this configuration, we can apply Equations 3 and 4. The user moves the the magnet in the YZ plane which is the known distance x from the magnetometer. Furthermore, the magnet is axis aligned with the sensor. Since the magnetic field is rotationally symmetric about the pole, we only need to consider the plane coincident with the x-axis that passes through the pole of the permanent magnet.

Let $\hat{\boldsymbol{H}}$ be the magnetic field measured by the magnetometer. We rotate the reading about the x-axis by the angle $\alpha = \text{atan2}(\hat{H}_z, \hat{H}_y)$. Therefore for Equations 3 and 4, $H_x = \hat{H}_x$ and $H_y = \sqrt{\hat{H}_y^2 + \hat{H}_z^2}$. Since we know x, we have an over-determined system with only one unknown, y, and we can use least squares to solve for it[1]. Finally, we transform y back into the Cartesian coordinate frame of the magnetometer: $\hat{y} = y\cos\alpha$, $\hat{z} = y\sin\alpha$.

Unfortunately, the above only holds for a single magnet. In addition to the magnet we are using to create the field for input, there is the magnetic field of the Earth and other ambient sources. The impact of this field must be accounted for otherwise the distance calculations will be incorrect. Han *et al.* present some data indicating this problem [5]. They show the impact of the external field on calculating position in an area of 9cm x 9cm. In some locations the error appears rather large ($>$ 3cm). Unfortunately, there are very few details about the experimental setup used and limited data.

We examine the nature of these errors in more detail next. Furthermore, using magnetic field sensing in this way has other parameters we must consider. In particular, the strength of the magnet, distance from the magnet to the sensors, and sensor gain all play a role in our system.

SENSITIVITY ANALYSES

Geometry and Magnet

Figure 3 shows the relationship between the magnitude of the magnetic field $\boldsymbol{H}$, the strength of the magnet K, orientation of the magnet, and distance d. First, we can see the non-linear nature of Equation 1. When the magnet is far away, the magnetic field is very weak and a small change results in much larger change in d. This region provides better sensitivity to magnet movement, but the impact of sensor noise would also be more pronounced. Conversely, when the magnet is close to the sensor, the field strength is relatively large and small changes in $\boldsymbol{H}$ result in small changes in d. However, sensors have maximum operating ranges. For instance, the HMC5883L can only read values ± 8 Gauss ($\pm 800\mu T$) and the sensor will saturate if too close to the magnet.

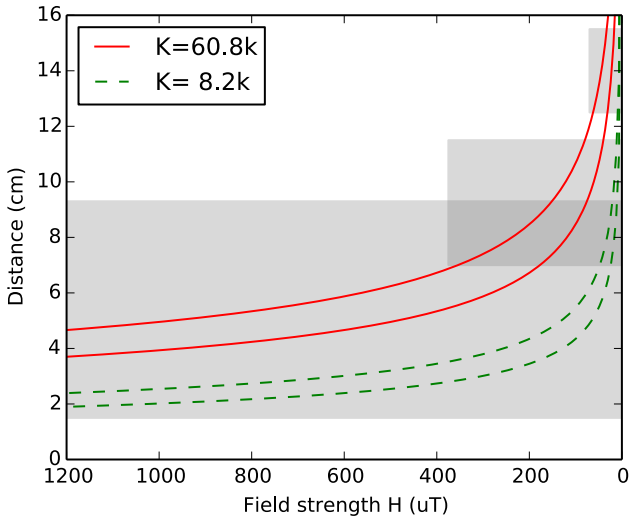

Figure 3. Relationship between field strength, distance, strength of magnet (K) and magnet orientation (parallel vs perpendicular).

This figure also shows the target operating regions for the different configurations of our test apparatus (grey rectangles in Figure 3). When the sensor and magnet are close (*e.g.*, our $x = 1.5cm$ position (bottom rectangle)) a large span of distances must be sensed. Furthermore, it may not be possible to sense when the magnet is close to the sensor. For example, even with a weaker magnet where K=8200, the HMC5883L saturates when closer than about 2cm. Conversely when the magnet is far away ($x = 12.5cm$, top rectangle), the span of distances we need to sense is smaller albeit at a larger absolute distance. This setup also means we are in the region where small differences in magnetic field readings result in large changes in calculated distance. Finally, the orientation of the magnet shifts the curves. When the magnet is parallel the field is tangential (H_r) so the distance is larger for a given field strength compared to when the magnet is perpendicular and the field is radial (H_θ).

Ambient Field

We also examine the impact of the ambient magnetic field using a sensitivity analysis. The intensity of the geomagnetic field varies based on location from 22-67 μT [4]. And as above, the orientation of the permanent magnet, sensor and geomagnetic field must be considered. The worst case is when the geomagnetic field is directly aligned with the magnet so the total magnitude of the geomagnetic field is either added to or subtracted from the magnitude of the permanent magnet. The error is also greatest when the angle θ between the magnetometer and pole is 0 (or 180) degrees. In this case,

[1]In this paper, we use the scipy.optimize.leastsq function.

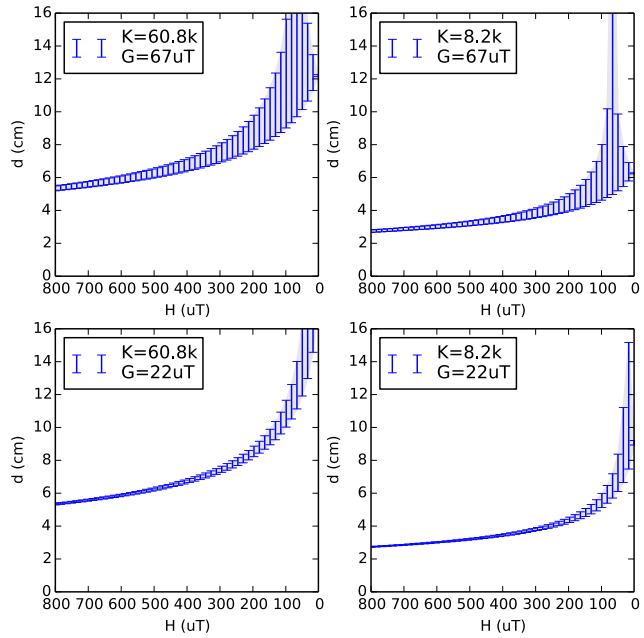

Figure 4. Analysis of the impact of ambient magnetic field on distance calculation.

$H_r = 2K/r^3$ and $H_\theta = 0$ therefore $r = \sqrt[3]{2K/|H|}$. We use this configuration for our analysis.

Figure 4 shows the possible uncertainty in position for two different magnets and different ambient geomagnetic field strengths. When the above conditions are met, a given magnetic field reading ($\boldsymbol{H}$) can be off by as much as +/- the magnitude of the geomagnetic field. This uncertainty in the actual value of $\boldsymbol{H}$ results in an associated uncertainty in position. For example, if we measure $H = 700\mu T$ and are in a region where the ambient geomagnetic field is $67\mu T$ (Figure 4, top-left), the distance from the sensor to the magnet would be somewhere between 5.69cm and 6.07cm. However, when the magnet is moved farther away and we measure a field of $H = 100\mu T$, the possible position is between 9.46cm and 16.2cm. The uncertainty in position approaches infinity when the ambient magnetic field exactly cancels out the magnetic field of the permanent magnet ($H = 0$).

If we are in a location where the ambient magnetic field is weaker (Figure 4, bottom), then the impact on error is smaller. Unfortunately, the magnitude of the geomagnetic field is outside our control. Also, the strength of the magnet impacts the error where stronger magnets result in larger error bounds due to the larger value of K in Equation 1 (Figure 4 left vs right). This finding is counter to Chen *et al.*'s reasoning for selecting a strong permanent magnet to counteract the ambient magnetic field [2].

EXPERIMENTS CHARACTERIZING MF SENSING

Static Known Ambient Field

Our first experiment demonstrates applying these equations to sensor data in a VR enclosure configuration. We perform a static test on a table with our apparatus so that we have a constant ambient magnetic field, $\boldsymbol{G}$. Given the above analysis, we

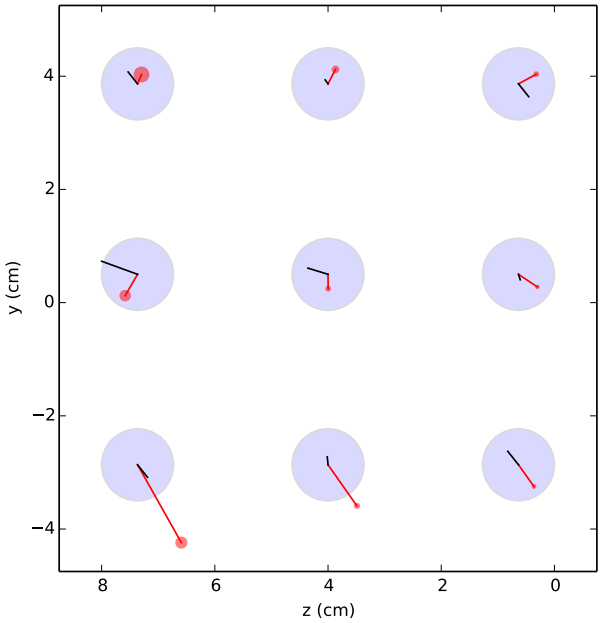

Figure 5. Visualization of our static test results. Red indicates the mean error for each position where the red circle is the dispersion (mean of difference from the central point). Black indicates the mean position after calibration using an affine transform. The blue circles represent the position for each test and the magnet.

(cm)	z=7.365	z=4.000	z=0.635
y= 3.865	0.179, 0.268	0.287, 0.086	0.357, 0.290
y= 0.500	0.435, 0.675	0.254, 0.372	0.401, 0.101
y=-2.865	1.578, 0.283	0.887, 0.142	0.471, 0.304

Table 1. Errors for static test (original value, after calibration).

only report data for our middle sensor position ($x = 7.0cm$) and a magnet where K was measured to be approximately 8200. With the distance we are operating at, we also increase the sensitivity of the HMC5883L to $\pm 88\mu T$ (Figure 3).

For all of the magnetic field readings in the paper, we first calibrate for hard and soft iron effects. We remove the permanent magnet from the area and rotate the device around all axes. If the magnetometer were perfectly calibrated, all of these points would lie on a sphere centered at (0,0,0) with the radius being the magnitude of the ambient magnetic field. In practice, there can be hard iron effects (the center is offset) and soft iron effects (the sphere is deformed into an ellipsoid). To compensate for these effects, we fit an ellipsoid to the data after removing outliers. The parameters of the ellipsoid (centroid and major/minor axes) provide the needed information to transform the raw magnetometer readings into ones where these effects are corrected. We perform this calibration procedure once and apply it to every sensor reading taken from the magnetometer.

At the beginning of our static test, we captured 500 readings of the ambient field and took the mean, $\boldsymbol{G}$. During the test, we subtract this vector from each magnetic field reading:

$$\boldsymbol{H} = \hat{\boldsymbol{H}} - \boldsymbol{G} \tag{5}$$

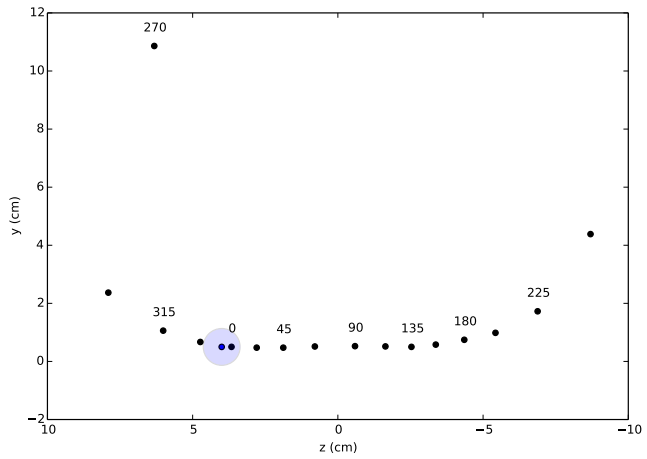

Figure 6. Errors resulting from rotating the device and not taking into account the corresponding counter rotation of the ambient field. The blue point and circle are the ground truth position. Each black point represents the calculated position at a given angle.

On our 8cm x 8cm interaction area, we attached a printed template with 9 positions and manually placed the magnet in each location (Figure 1, top). We collect 500 readings for each position and use the previous equations to calculate the location of the magnet. Figure 5 (red) and Table 1 show the difference between the mean calculated position and ground truth. The maximum error across these nine positions is about 1.6cm. These errors are relatively systematic and probably result from our 3d printed apparatus and sensor not being exactly aligned. However, a cheap VR enclosure (for instance made of cardboard) might suffer from this problem as well. A simple calibration could improve this error. To demonstrate, we compute an affine transformation using the known location of the four corners. After applying the transformation, the errors are reduced to less than 0.7cm (Figure 5, black). For comparison, the magnet itself is 1.27cm in diameter.

Overall, this data shows that if we could somehow remove the ambient magnetic field, we can obtain good 2D position measurements for user input on a VR enclosure. Unfortunately, as our previous sensitivity analysis shows, the ambient field can have a very large impact on our position measurement.

Device Rotation

Our VR enclosure will not be stationary and therefore the strategy from the previous experiment of estimating the ambient field first and assuming it remains constant is unrealistic. To demonstrate the impact of not accounting for the direction of the ambient field, we conduct a second experiment similar to the previous one. Here, we orient the device so that the ambient field is aligned with the Z axis (0 degrees) and measure the ambient magnetic field at the beginning of the experiment as before ($\boldsymbol{G} = [2.35, -26.39, 20.38], |\boldsymbol{G}| = 33.42\mu T$).

We place the magnet in a single position (in the middle at z=4cm, y=0.5cm) and manually rotate the whole apparatus about the Y axis at 22.5 degree increments through a full circle. At each position, we again collect 500 readings and subtract the initially determined magnetic field vector $\boldsymbol{G}$ from each (Eq. 5). We calculate the mean position of the magnet. As we see in Figure 6, at zero degrees the position is reasonably accurate. However, as we rotate through the circle, there is significant error (much larger than the interaction area).

COMPENSATING FOR THE AMBIENT FIELD

Fundamentally, the challenge with using magnetic field sensing for our VR enclosure is that the field can change for one of two reasons. It varies as the user moves the magnet to provide input as we intend. It also changes as the user moves the VR enclosure around to look at different elements in the virtual world. The above data and analyses show the error introduced by movement can be enough to make our distance calculations useless for input. While carefully selecting the strength of the magnet or position of the magnetometer might mitigate the impact, it does not fundamentally solve this issue. It is impossible to differentiate between these two with just the one magnetometer in a phone. Han *et al.* use a second magnetometer to solve this problem [5], but that solution is not available to us. Instead, we use the inertial sensors in the phone.

Tracking Phone Movement

If we could let the user only move the magnet or only the VR enclosure, but not both at the same time, we could solve this issue. When just the magnet is moving, we use the above solution where we first estimate and subtract out the geomagnetic field (Eq. 5). We treat G as a constant to determine H and apply Equations 3 and 4. When the VR enclosure moves and the magnet is stationary, we can do the opposite:

$$\boldsymbol{G} = \hat{\boldsymbol{H}} - \boldsymbol{H} \tag{6}$$

The field generated by the magnet $\boldsymbol{H}$ is treated as constant since the user is not moving it and we redetermine $\boldsymbol{G}$. Having a mode for tracking phone movement versus allowing input could be permissible depending on the interactions exposed to the user. However, asking the user to stay perfectly stationary while providing input is not feasible.

We need a way to track the changing orientation of $\boldsymbol{G}$ while also allowing the user to move the magnet. Luckily, VR enclosures must already estimate the rotation of the phone so it can create the right viewport into the virtual environment. We can use this same approach to track the orientation of the ambient magnetic field as the user moves.

In other situations, orientation can be tracked using the accelerometer and the magnetometer. These sensors form a basis (measuring gravity pointing down and the Earth's magnetic field pointing north) for determining absolute orientation. However, since devices like Google Cardboard use the magnet for input, they forego the magnetometer and instead must rely on fusing the accelerometer and gyroscope. For example, WebVR Polyfill[2] is an implementation of Google Cardboard for mobile web browsers which uses a complimentary filter to fuse accelerometer and gyroscope sensor readings. By integrating the measurements of rotational velocity over time and fusing them with the gravity vector, the complimentary filter provides an estimate of the rotation matrix

[2]https://github.com/borismus/webvr-polyfill

R that transforms the starting reference frame to the current one. The graphics software uses this rotation to create the right view port into the virtual world. We can use it to track $\boldsymbol{G}$. As such, we added a gyroscope and accelerometer to our test apparatus (Figure 1) and use the tracking code from WebVR Polyfill in our implementation.

We can start with an estimate of $\boldsymbol{G}$ as we have been doing; or more likely, we start with the magnet in a known position so that we know what the associated magnetic field reading $\boldsymbol{H}$ should be (for example stored in the same database as the relative position of sensor and interaction surface). Therefore we use Equation 6 to obtain an initial measurement $\boldsymbol{G_i}$. As the user rotates the VR enclosure, we re-estimate the orientation of the phone (R) using the inertial sensors and apply the associated transform:

$$\boldsymbol{G} = R \cdot \boldsymbol{G_i} \tag{7}$$

This approach lets the user move both their head (and the VR enclosure) as well as the magnet simultaneously. The inertial sensors track $\boldsymbol{G}$ as the user moves. We then use Equation 5 and in turn Equations 3 and 4 to obtain the 2D position of the magnet.

With a really good gyroscope and filtering, this solution might be enough. However today's gyroscopes have both drift and noise. Given how sensitive we are to incorrect magnetic field readings (Figures 4 and 6), it is not feasible to track the orientation indefinitely. The solution we adopt in this paper is to rely upon user interactions that have modal input.

By default, the magnet is not being used for input and the user is looking around the virtual environment. The magnet starts in a known position so we can continually calculate the ambient magnetic field (Eq. 6). At a certain point, the user transitions into the input mode where we save the current measurement $\boldsymbol{G_i} = \boldsymbol{G}$ and then start tracking $\boldsymbol{G}$. The user can move both the magnet and the enclosure as desired. When the user exits the input mode, the last value of $\boldsymbol{H}$ is saved, and the system goes back to calculating $\boldsymbol{G}$. This approach can suffer from IMU drift; however, the drift only accumulates while in the input mode. And since drift is proportional to tracking time, this solution works for VR applications that utilize short bursts of 2D input.

Another interaction approach to minimize the impact of drift is to have the magnet return to a known position after each input session. The original Google Cardboard uses a second fixed magnet on the inside of the device so that the input magnet would recenter. Unfortunately, this option is not available to us as there is no known way to determine the location of the two separate magnets (and associated magnetic field) with just the one magnetometer. Instead, our interaction techniques are designed so the user positions the magnet in a known location. In this case, there is the potential for some drift during input, but it does not accumulate from session to session. For instance, a detent could be created on the surface of the device so the user can return the magnet to a known position. In the next section, we describe how this type of interaction works for a D-Pad like 4-way input widget. EdgeWrite [12] would also be suitable for this approach.

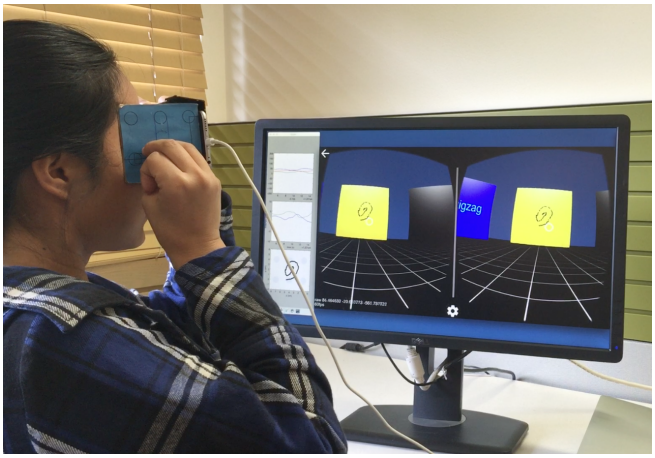

Figure 7. A user drawing a spiral on our iOS Cardboard app. The phone screen is mirrored to the monitor and raw magnetometer and position information is displayed in the window on the left.

The user makes gestures along the edge of the device to enter letters of the alphabet. However by design, the user also always stops a given gesture in a known corner. That position would let the system reset to a known magnetic field reading $\boldsymbol{H}$ and determine the current ambient field reading $\boldsymbol{G}$.

IMPLEMENTATION

We implemented two versions of our system for 2D input (Figure 1). The first is on the apparatus used for our experiments with discrete sensors and an Arduino. Sensor data is sent to a laptop over USB serial for processing and the code to solve the magnetic field equations is written in Python. For tracking, we use the complementary filter implementation provided by WebVR Polyfill. This software is written in Javascript and runs in Node.js. We also built a simple desktop user interface that shows the magnetometer readings and two representations of the calculated magnet position.

Our second version runs on iOS for use in a Google Cardboard VR enclosure. In particular, we use an iPhone 6 and the Cardboard SDK which is built on top of Unity3D. We added Objective C code to obtain "raw" sensor data from the Core Motion API. For example, we use *startMagnetometerUpdatesToQueue:withHandler:* to obtain *CMMagnetometer* sensor events. The documentation indicates that that these are readings that come straight from the sensor. However, there is clearly some processing going on at the sensor, firmware or OS level as strong magnetic fields in close proximity seem to change the calibration of the sensor. Also, since we are using the raw events, we perform our own hard and soft iron corrections for the iPhone just as we do for our test apparatus. This step is very much required as there are significant hard iron effects, likely due to the close proximity of the magnetometer to the ear speaker.

Unlike our test apparatus, we do not know the exact position of the iPhone 6 magnetometer. Looking at tear downs of the phone we can see that the sensor is in the top left. Examining magnetometer data, we estimate that it is approximately 1.1cm from the top of the phone and 1.9cm from the left.

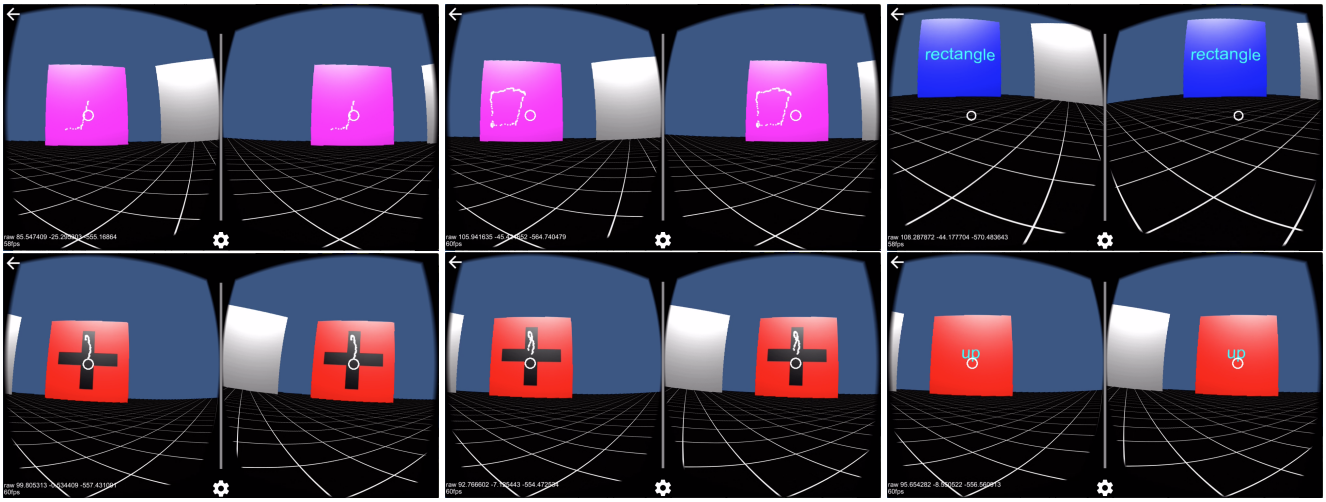

Figure 8. Examples of using the $P gesture recognizer to input a rectangle (top) and the D-Pad widget to input "up" (bottom).

Since the magnetometer is so close to the edge, we can see from Figure 3 that our permanent magnet would likely saturate the sensor if the top of the phone is placed on the end of our VR enclosure used for input. Our preliminary testing indicated this is indeed the situation and therefore we use the reverse configuration. Our estimated distance from the sensor to the interaction surface has $x = 12.3cm$. Also, since the sensor and magnet are far apart and we have no direct control of sensitivity, we use a much stronger magnet than we used in our test apparatus where K is approximately 44000. We actually use two magnets forming a sandwich similar to the original Google Cardboard [10]. One magnet is on the outside and manipulated by the user. The second magnet is on the inside and moves with the first. It holds the magnet on the outside in place when the user lets go. Finally, we noticed a lot of sensor noise in our readings at this distance. We apply an exponential moving average to the magnetometer data for smoothing out the noise.

We developed a demonstration user interface for the iPhone that is written in C# for Unity3D. Currently, our software to calculate the position of the magnet from the magnetic field readings, and to estimate device rotations, has not been ported to native code and instead we perform an RPC over the network to the same software that runs our test apparatus. A fully native solution is left as future work and should greatly improve overall latency in the system.

Interaction Widgets

We created three different interaction widgets for our Unity3D Cardboard app. Each is a plane positioned in the 3D world. To enter the input mode, the user places the "gaze" cursor on the widget by rotating Cardboard. After two seconds, the system switches modes which is visually indicated by changing the color of the surface. At this point, the system starts tracking device rotation to estimate $\boldsymbol{G}$ and also calculates the 2D position of the magnet which is used by the user interface. When the gaze cursor leaves the widget, the system returns to tracking only mode.

Our first widget is a simple drawing surface that paints a point for each position that is calculated. Figure 7 shows a novice user drawing a spiral with our system. Our second widget performs gesture recognition. In particular, we use a Unity3D port[3] of the $P gesture recognizer [11]. For debugging, we render the individual points as they are calculated. Once the user exits the widget, the recognizer processes the points and displays the result of gesture recognition (Figure 8, top). Our final widget is similar to a simple 4-way D-pad (Figure 8, bottom). The user activates the widget, moves the magnet in any of the 4 cardinal positions, then returns it to the center. The direction detected is displayed onscreen. This widget is an example of an interaction that does not have the cumulative error of tracking the magnet from input session to input session. It assumes the user starts with the magnet in the center of the input area and finishes in the same position.

We had a volunteer from our organization use our system. She had no prior experience with VR enclosures such as Google Cardboard and was a novice at using our device (less than 30 minutes of experience in total). Figures 7, 8 and our Video Figure show examples of her using our system with our three different interaction widgets.

DISCUSSION AND FUTURE WORK

In using magnetic field sensing for 2D input, the ambient magnetic field can have a significant impact. However, as we detailed, tracking device orientation using the inertial sensors allows us to successfully use the magnet for user input. The version of our system that runs on the iPhone uses the "raw" sensor values. However, the APIs available do not provide the same level of access to the sensors as a direct hardware API would. Also, mobile phone platforms offer several soft sensors that utilize a variety of filtering and sensor fusion algorithms to overcome some of the limitations of the sensors or compensate for known but proprietary factors impacting the sensors. Unfortunately, the existing software makes assumptions that are not valid with our approach. In particular, the

[3] https://github.com/DaVikingCode/PDollar-Unity

presence of a large permanent magnet that can move results in useless sensor readings from these more advanced APIs. A future opportunity would be to create APIs that allow a developer to leverage the processing the phones are doing on the sensor data but that also takes into account the presence of our relatively large moving magnetic field we use for input.

Another area of future work would be to develop a holistic system that tracks all of the unknown variables in our system. Given the above API issues, our current system uses a standalone tracking system based on a complementary filter. Likewise, our magnetic field equations are over-determined but we are ignoring possible errors in solving those equations by just using function minimization. There are also constraints on the magnet movement we are not using (the bounding box of the interaction surface), and the current system does not model the dynamics of user movement for either the VR enclosure or the magnet. It would seem that there is opportunity in combining all of these constraints into one system that tracks the needed variables.

It would be useful to characterize the overall performance of the input system. However, since our method uses the inertial sensors to track device orientation and has modal input, one would need to simulate realistic device rotation as well input durations to get a sense of real world performance. The findings would also likely be sensitive to the exact sensors and tracking implementations used. An alternative would be to perform tests with users. However, this approach confounds system performance and user performance.

Because our iPhone configuration has the sensor placed relatively far away from the interaction surface, we use a strong pair of magnets to create a sufficiently large signal. As such, it can be a bit difficult to slide the magnet over the surface due to the force exerted between the magnets. Applying a coating with a lower coefficient of friction or developing a handle for the magnet might provide for easier input. The handle could both reduce friction between the magnet and the cardboard surface, as well as provide a better point for grasping the magnet with the fingers allowing for easier actuation.

Finally, we developed some simple interaction widgets to demonstrate the capability of our system to provide 2D input. In future work, it would be interesting to explore interactions where the position of the magnet at the start or end of a movement is known. Doing so would minimize any cumulative tracking error similar to our D-Pad example or adapting EdgeWrite as mentioned above. There are other options to investigate for switching between tracking and input modes beyond using Cardboard's gaze pointer. Likewise, there is opportunity to create or adapt existing VR interaction techniques to make use of our continuous 2D pointing capability.

CONCLUSION

Virtual Reality enclosures provide a cheap and simple way to experience VR content. However, given that cost and simplicity is such a driving factor in their design, there are limited opportunities for input. We extend the original idea from Google Cardboard of using a magnet for input. However instead of treating it as binary, we use magnetic field sensing to provide continuous 2D input. By tracking the orientation of the ambient magnetic field using the inertial sensors in the phone, we can successfully calculate the 2D position of the magnet. We showed our system working on a smartphone and created a Unity3D Cardboard app to explore different interactions that are enabled with our new 2D pointing capability.

REFERENCES

1. Ashbrook, D., Baudisch, P., and White, S. Nenya: Subtle and Eyes-free Mobile Input with a Magnetically-tracked Finger Ring. In *Proceedings CHI '11* (2011), 2043–2046.
2. Chen, K.-Y., Lyons, K., White, S., and Patel, S. uTrack: 3D Input Using Two Magnetic Sensors. In *Proceedings of UIST '13* (2013), 237–244.
3. Chen, K.-Y., Patel, S., and Keller, S. Finexus: Tracking Precise Motions of Multiple Fingertips Using Magnetic Sensing . In *Proceedings of CHI '16* (2016), 1504–1514.
4. Chulliat, A., Macmillan, S., Alken, P., Beggan, C., Nair, M., Hamilton, B., Woods, A., Ridley, V., Maus, S., and Thomson, A. The US/UK World Magnetic Model for 2015-2020. Tech. rep., NOAA National Geophysical Data Center, 2015.
5. Han, X., Seki, H., and Hikizu, M. Wearable Handwriting Input Device Using Magnetic Field. In *SICE, 2007 Annual Conference*, IEEE (2007), 365–368.
6. Han, X., Seki, H., Kamiya, Y., and Hikizu, M. Wearable Handwriting Input Device Using Magnetic Field: 2nd Report: Influence of Misalignment of Magnet and Writing Plane. *Precision Engineering 34*, 3 (2010), 425–430.
7. Harrison, C., and Hudson, S. E. Abracadabra: Wireless, High-precision, and Unpowered Finger Input for Very Small Mobile Devices. In *Proceedings of UIST '09* (2009), 121–124.
8. Kraichman, M. B. *Handbook of Electromagnetic Propagation in Conducting Media*. Headquarters Naval Material Command, 1970.
9. Liang, R.-H., Cheng, K.-Y., Su, C.-H., Weng, C.-T., Chen, B.-Y., and Yang, D.-N. GaussSense: Attachable Stylus Sensing Using Magnetic Sensor Grid. In *Proceedings of UIST '12* (2012), 319–326.
10. Smus, B., and Riederer, C. Magnetic Input for Mobile Virtual Reality. In *Proc. of ISWC '15* (2015), 43–44.
11. Vatavu, R.-D., Anthony, L., and Wobbrock, J. O. Gestures As Point Clouds: A $P Recognizer for User Interface Prototypes. In *Proceedings of ICMI '12* (2012), 273–280.
12. Wobbrock, J. O., Myers, B. A., and Kembel, J. A. EdgeWrite: A Stylus-based Text Entry Method Designed for High Accuracy and Stability of Motion. In *Proceedings of UIST '03* (2003), 61–70.
13. Yoon, S. H., Huo, K., and Ramani, K. TMotion: Embedded 3D Mobile Input Using Magnetic Sensing Technique. In *Adjunct Proc. of UIST '15* (2015), 71–72.