3DTouch: A Wearable 3D Input Device for 3D Applications

Anh Nguyen* Amy Banic†

Department of Computer Science, University of Wyoming

Abstract

3D applications appear in every corner of life in the current technology era. There is a need for an ubiquitous 3D input device that works with many different platforms, from head-mounted displays (HMDs) to mobile touch devices, 3DTVs, and even the Cave Automatic Virtual Environments. We present 3DTouch, a novel wearable 3D input device worn on the fingertip for 3D manipulation tasks. 3DTouch is designed to fill the missing gap of a 3D input device that is self-contained, mobile, and universally works across various 3D platforms. This paper presents a low-cost solution to designing and implementing such a device.

Our approach relies on a relative positioning technique using an optical laser sensor and a 9-DOF inertial measurement unit. The device employs touch input for the benefits of passive haptic feedback, and movement stability. On the other hand, with touch interaction, 3DTouch is conceptually less fatiguing to use over many hours than 3D spatial input devices. We propose a set of 3D interaction techniques including selection, translation, and rotation using 3DTouch. An evaluation also demonstrates the device's tracking accuracy of 1.10 mm and 2.33 degrees for subtle touch interaction in 3D space. We envision that modular solutions like 3DTouch opens up a whole new design space for interaction techniques to further develop on. With 3DTouch, we attempt to bring 3D applications a step closer to users.

Index Terms: H.5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces—Input devices and strategies;

1 Introduction

3D applications appear in every corner of life in the current technology era. The 3D technology, a key element behind Virtual Reality (VR) systems, has reached beyond traditional computer games, and modeling applications to also browsers (e.g. WebGL), touch devices, home theaters, and even large visualization platforms such as the Cave Automatic Virtual Environment (CAVE). However, each of these platforms require users to learn and operate a different set of input devices. Recent years have witnessed a wide variety of input devices [5]. Desktop input devices such as traditional mice, keyboards, or 3D mice (e.g. 3Dconnexion SpaceNavigator) provide stability and accuracy; however, they are not portable for spatial environments such as the CAVE. Tracked multi-touch mobile devices [28] are portable devices that enable intuitive and direct input in a VR environment, but their working space is limited within the screen area. While voice input is convenient, it is not intuitive for users to give voice commands for performing complex 3D interaction tasks (e.g. rotate the red cube 60^o around z-axis). Although these input devices have their unique advantages, they are usually designed for a single certain platform.

One of the challenges of bringing VR applications to broader user groups is the cumbersome infrastructure setups (e.g. CAVE) and non-portable input devices (e.g. mouse). The advent of portable VR commodity display solutions such as Oculus Rift and Google Cardboard, addresses this problem allowing VR applications to be used in mobile settings outside VR laboratories. However, there is not yet a portable 3D input device that is self-contained and can be used across different platforms.

Figure 1: 3DTouch - a novel 3D input device worn on the fingertip

One input method to interact with 3D applications is using 3D mid-air gestures, which are popularized by commodity devices like Kinect and Wiimote. However, serving as a type of 3D input beyond the purpose of entertainment, they are subject to a major drawback of fatigue [19]. A recent study showed that performing 3D mid-air gestures with bare hands is more tiring than performing 1D and 2D gestures on hand-held input devices (e.g. smartphones, or remote controls) [19].

Touch interaction is another way to interact with 3D applications. Unlike spatial interaction, touch interaction has a subtle neat advantage that users can feel natural passive haptic feedback on the skin via sense of touch. Touch gestures are conceptually less fatiguing than 3D mid-air gestures. Moreover, the touch surface keeps the hand steady and thus increasing the stability and accuracy of finger movements. A variety of creative research works have then brought touch interaction to surfaces that are not inherently touch-sensing capable such as tables [4], walls [18], clothes [24], skin [14, 12], or virtually any flat surface using a combination of a depth-sensing camera and a projector [12].

In this paper, we present 3DTouch, a thimble-like 3D touch input device worn on the user's fingertip. 3DTouch is self-contained, and universally working on various VR platforms (e.g. desktop and CAVE). The device employs touch input for the benefits of passive haptic feedback, and movement stability. On the other hand, with touch interaction, 3DTouch is conceptually less fatiguing to use over many hours than spatial input devices. With such an ubiquitous input device, users wearing HMDs can interact with VR environments *anywhere*.

3DTouch allows users to perform touch interaction on many surfaces that can be found in an office environment (e.g. mousepad, jeans, wooden desk or human skin). When mounted on the tip of index finger, the user can perform touch interaction on the other hand's palm, which serves as the touch pad (Fig. 1). 3DTouch fuses data reported from a low-resolution, high-speed laser optical sensor

*e-mail: anguyen8@uwyo.edu
†e-mail: abanic@cs.uwyo.edu

IEEE Virtual Reality Conference 2015
23 - 27 March, Arles, France
978-1-4799-1727-3/15/$31.00 ©2015 IEEE

(OPS), and a 9-DOF inertial measurement unit (IMU) to derive relative position of a pointer in 3D space. The OPS, usually found in computer mice, determines the direction and magnitude of movement of the pointer on a 2D plane. The 3D orientation of this plane is determined by an IMU.

The contributions of this paper are: (1) A novel, low-cost technique to turn a finger or a thumb into 3D touch input device using an OPS and a 9-DOF IMU; (2) A set of interaction techniques showing how 3DTouch can be utilized in 3D applications; (3) An evaluation demonstrating the accuracy of 3DTouch across various surfaces of different materials, sizes and shapes.

2 Related Work

3DTouch is an interdisciplinary research project that crosses various fields. In this section, we review the related literature in the areas of VR, finger-worn interfaces, and touch interaction.

2.1 Virtual Reality

Motion tracking systems are widely used in VR community because of their capability of sensing position, orientation, and velocity of one or more objects. These 6-DOF position trackers can be based on many different technologies, such as those using electromagnetic fields (e.g. Polhemus Liberty), optical tracking (e.g. NaturalPoint OptiTrack [21]), or hybrid ultrasonic/inertial tracking (e.g. Intersense IS900). All of these, however, share the limitation that some external fixed reference (e.g. a base station, a camera array, or an emitter grid) must be used. While ultrasonic and electromagnetic tracking techniques are susceptible to environment interference, optical tracking is subject to the inherent problem of occlusion [27].

Inertial tracking systems, on the other hand, can be self-contained and require no external reference. They use technologies such as accelerometers, gyroscopes, and compasses to sense their change in orientation [16]. While devices equipped with such sensors (e.g. Wiimote, smartphones) are capable of serving as a 3D pointing device, they have only been used to translate objects on a fixed 2D plane (e.g. the TV screen). 3DTouch, with 5 DOFs, does not only serve as a 3D pointing device, but also enables users to rotate and translate objects in 3D space.

2.2 Finger-worn Interfaces

Early work in instrumenting the human finger was conducted in the 3DUI community. Ring Mouse [6] is a small, ring-like device, with two buttons, worn along the index finger. It uses ultrasonic tracking, but generates only position information. With a similar design to that of Ring Mouse, FingerSleeve uses a 6-DOF magnetic tracker to report position and orientation [32]. The drawback of these devices is that they are not self-contained, relying on an external tracking system.

Using magnetic field sensing techniques, several projects have explored augmenting the finger with a small magnet. With Abracadabra [13], users wear a magnet on their finger to provide 1D and 2D input to a mobile device. On the other hand, FingerFlux [26] provides simulated haptic feedback to the fingertip when operating above an interactive tabletop. While these devices bring more functionality to the finger, they do not support 3D and always-available input. By mounting a Hall sensor grid on the index fingernail, and a magnet on the thumbnail, FingerPad turns pinched fingertips into a touch pad [8]. However, the input space enabled by FingerPad is only 2D. uTrack [9] turns the fingers and thumb into a 3D input device. As a magnet is worn on the thumb, and two magnetometers are worn on the fingers, uTrack is a self-contained 3D input device. However, it is not a full 6-DOF input device and can only serve as a 3D pointing device.

Other researchers explored mounting cameras on the body [12, 20, 31] for truly ubiquitous use. Logisys's Finger Mouse, a cylinder-shaped optical mouse, brings the traditional mouse control to the finger [10]. Extending this concept, Magic Finger [31] allows users to recognize 32 different textures for contextual input by augmenting the finger with a high-resolution OPS. While these two projects are closely related to 3DTouch in using OPS, none of them had the goal of turning the finger into a 3D input device.

2.3 Extra Dimensions of Touch Interaction

Many mobile touch devices only utilize the 2D position of a touch contact being made on the surface. However, other auxiliary information of touch interaction has also proved to be useful such as: the shape [29, 7] or size [3] of the contact region; the orientation of the finger making contact [25]; and the touch pressure [23]. While the size of the contact region was used to improve the precision of selection techniques [3], attributes such as the shape of the contact region [29, 7], orientation of the finger [25], and touch pressure [23] were additional inputs for the application to deliver pseudo-haptic feedback to users.

Using a 9-DOF IMU mounted on the fingernail, 3DTouch leverages the finger orientation to augment the 2D input from the OPS into 3D input. And the pressure dimension is used to enable *press gesture*, conceptually similar to a mouse-click gesture. Unlike the popular tap gesture on touch devices, press gesture allows the user to make selection commands without lifting finger off the surface, thus reducing workload for the finger joint.

Note that multi-touch mobile devices can be used in coordination with a tracking system [28] to enable an intuitive and direct touch input for an VR environment. 3DTouch can also be utilized in a similar fashion on a touch surface being human skin or physical props [11].

3 Hardware Prototype

An open problem of spatial tracking is how to build a 6-DOF system that is self-contained, and capable of tracking its own position and orientation with high levels of accuracy and precision [5]. With 3DTouch, our approach is to fuse data from a 9-DOF IMU and a laser OPS to derive position and orientation.

3.1 Inertial Measurement Unit

Pololu MinIMU-9 v2 is a 9-DOF IMU that packs an L3GD20 3-axis gyro, an LSM303DLHC 3-axis accelerometer and 3-axis magnetometer onto a tiny 0.8” $\times$ 0.5” board. We selected such an IMU with 9 degrees of freedom because when applying Kalman filter [17], the estimates of orientation would be more precise than those based on a single measurement alone.

3.2 Optical Flow Sensor

We used Pixart ADNS-9800 laser optical flow sensor with a modified ADNS-6190-002 lens. The reason we chose a laser OPS is that they work on a larger number of surfaces than LED-based OPS. ADNS-9800, often found in modern laser gaming mice, comprises a sensor and a vertical-cavity surface-emitting laser (VCSEL) in a single chip-on-board (COB) package. The sensor is a high resolution (up to 8200 cpi), black-and-white camera (30×30 pixels). However, for the purpose of tracking movements, we programmed the resolution down to 400 cpi for a higher frame rate. This OPS is then wired to an application printed circuit board (PCB) designed according to the schematic diagram in the datasheet [1]. The PCB streams data from the OPS to an Arduino Uno R3 (Fig. 7).

3.3 Physical Form

The device needs to be small enough to be worn on the user's finger. We mounted the IMU on top of the fingernail so that we can utilize the finger's orientation. Fig. 2 illustrates three possible form factors of 3DTouch. In form factor 1 (Fig. 2a), if small enough, the OPS could be mounted on the fingertip. In form factor 2 (Fig. 2b), the OPS could be placed on the fingerpad. In form factor 3 (Fig. 2c), 3DTouch, worn as a ring, can be used as a pointing device, and the finger does not have to perform touch interaction with a surface. The third form factor enables users to turn their finger into a pointing device, and use the thumb to perform touch gestures such as tap and double-tap with the OPS. Beyond these three proposed form factors, 3DTouch can be usable when worn on the thumb as well.

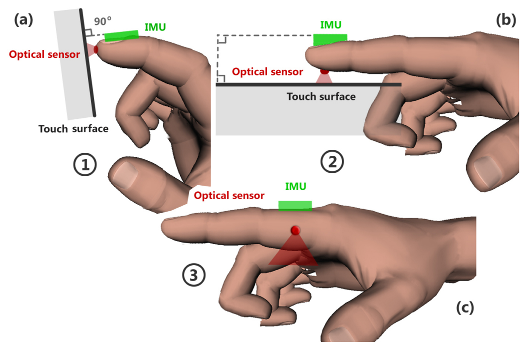

Figure 2: Three possible form factors of 3DTouch: (a) Form factor 1: The OPS is mounted on the fingertip, below the fingernail. (b) Form factor 2: The OPS is placed on the fingerpad. (c) Form factor 3: The finger can serve as a pointing device with 3DTouch worn as a ring. In all three form factors, the IMU is placed on the finger for the finger orientation to be utilized.

Our prototype presented in this paper (Fig. 1) was implemented in form factor 2. An user can transform from form factor 2 into form factor 3 by simply pushing 3DTouch further towards the palm. 3DTouch has the shape of a thimble, which is an adjustable Velcro strap used to hold the sensors. The IMU is mounted on top of the fingernail, and the OPS is on the fingerpad (Fig. 1).

3.4 Computer Interfacing

The IMU and OPS stream data to an Arduino Uno board. The Atmega16U2 microcontroller on Arduino then applies Kalman filtering to the data from the IMU, and synchronizes the orientation result with relative position data from the OPS. The fused data are then streamed to a computer running Ubuntu 12.04. An USB cable is used to connect Arduino Uno to the computer for evaluation purposes. This wired connection later could be replaced by a wireless solution using a pair of XBee modules.

3.5 How to fuse sensory data and derive 3D position?

The OPS provides a pair of 2D position increments (x, y) at a time t. These values (x, y) represent the distance (calculated from the speed measured in *counts per inch* or *cpi*) that the sensor has traveled on a 2D plane over a delta time $t_2 - t_1$. The OPS basically determines the direction and magnitude of touch movements on a 2D surface.

When combined with the OPS (Fig. 2), the IMU is used to measure the 3D orientation (relative to the ground that the user is standing on) of the 2D plane that the OPS moves on. In form factor 1 (Fig. 2a), the IMU is perpendicular to the touch surface (Fig. 5a,c), the orientation of the 2D plane is the orientation of the IMU plus 90^o *pitch* angle. In form factor 2 (Fig. 2b), the IMU is parallel with the touch surface, the orientation of the 2D plane is the orientation of the finger. Given the relative 2D position of a point on the plane and the absolute 3D orientation of the plane, it is easy to calculate the relative 3D position of the point (Fig. 3).

4 Gesture Detection

For the device to be usable, we decided to implement the basic touch gestures of tap and double-tap. A novel *press gesture* is also proposed. This section explains the algorithms used to enable the tap, double-tap, and press gestures.

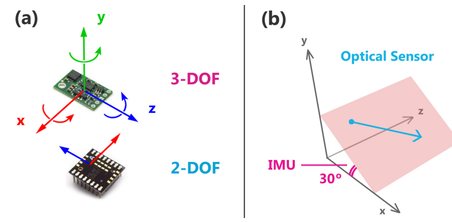

Figure 3: Relative 3D position can be derived by fusing data from an IMU and an OPS: (a) The IMU (top) and OPS (bottom); (b) The OPS determines the direction and magnitude of movement on the 2D plane. The orientation of this plane is not fixed and determined by the IMU.

4.1 Sensing Contact

To sense contact with a surface, Magic Finger [31] relies on rapid changes in the pixel contrast level of the sensor image. This approach requires continuous reading of the image pixels, and performing the calculation to derive the change in contrast level. However, we took a simpler, yet effective approach by monitoring the surface quality (SQUAL) values reported directly by the ADNS-9800 sensor board. As described in the datasheet [1], SQUAL ranges from $0 - 169$, and becomes nearly zero if there is no surface below the sensor.

However both approaches of using image contrast level, and SQUAL are still optical techniques to sense contact. Hence, the sensing accuracy is affected by variables such as environment lighting condition, surface texture, and the lift detection (Z-height) setting programmed to the OPS. Different surfaces will have different lift detection values with the same setting due to different surface characteristic [1].

4.2 Tap Gesture

The SQUAL, and (x, y) increments (X_DELTA, and Y_DELTA) values are used to measure tap gestures. A tap gesture is recognized when there is a rapid change, within 300ms timespan, in SQUAL from 0 to 40, and in (x, y) movements between ± 5 units. These settings are specific values for mousepad texture only. When the texture can be recognized by a pattern recognition algorithm [31], it is possible to load the specific settings for corresponding textures.

4.3 Double-Tap Gesture

Similar to a double-click gesture, we needed to continuously monitor the tap gestures. If two tap gestures take place within a certain pre-defined time span, then a double-tap gesture is fired. Microsoft Windows 7 sets 500ms as the default time span for a double-click [30]. However, this should be an adjustable setting for users, and for the purposes of testing, we set it to be 200-500ms.

Furthermore, for a double-tap gesture to be recognized, two subsequent taps need to occur at the same position. This is difficult to achieve with optical sensing because there is always noise when the sensor is lifted off the surface. After pilot testing 300 double-tap gestures, we defined the offset distance for two subsequent taps to be recognized as a double-tap to be ± 15 for mousepad texture.

4.4 Press Gesture

We apply a thin layer of elastic rubber of 2.0 mm height around the curvature of the ADNS-6190-002 lens of the OPS. The lift detection distance for ADNS-9800 ranges from $1 - 5$ mm [1]. As the fixed height of the ADNS-6190-002 is 2.4 mm, the 2.0 mm thin layer of rubber allows the sensor to still recognize the surface within a $2.4 - 4.4$ mm range. For the mousepad texture, an average SQUAL value of 40 corresponds to 2.4 mm lift-off distance under normal indoor light condition. We continuously monitor and detect a press gesture when the SQUAL values are ≥ 40.

This gesture reduces workload for the finger joint as users do not have to lift their finger off the surface. However, it is subject to many other environmental factors such as surface texture, and lighting condition. A mechanical push button may be a reliable alternative.

5 3DTouch Interaction Techniques

This section describes how a single 3DTouch device, worn on a finger or thumb, can be used to perform 3D interaction techniques of selection, translation, and rotation. Interaction techniques utilizing more than one finger are discussed in Section 7, and are not within the scope of this paper. The interaction techniques presented in this section are implemented in Virtual Reality User Interface (Vrui) framework [15], which allows 3D applications to run on a wide variety of platforms such as desktops, wall displays and CAVEs. We recorded a video (attached with this submission) to demonstrate how our 3DTouch prototype is currently being used to interact with VR applications based on Vrui. For reproduction, our code for integrating 3DTouch with Vrui will be available upon request.

5.1 Selection

3DTouch is capable of sensing the absolute 3-DOF orientation of the finger wearing the device. Hence, we propose to use the traditional Ray-Casting technique [6] to select an object in 3D space. With ray casting, the user points the finger wearing 3DTouch at objects with a virtual ray that defines the direction of pointing (Fig. 4b). More than one object can be intersected by the ray; however, only the one closest to the user should be selected. On a 2D plane such as the TV screen, the user can point up-down and left-right to move the 2D pointer around (Fig. 4a).

After a ray is pointed at an object, a tap gesture can be performed to make the selection command. For the selection technique, the form factor 3 (Fig. 2c) is the most suitable because the user can use their finger as a pointing device and give selection commands by performing tap and double-tap gestures. Since 3DTouch does not support absolute positioning, the casted ray always starts from a pre-configured point (e.g. the center bottom of the screen or a tracked body when used in combination with a tracking system).

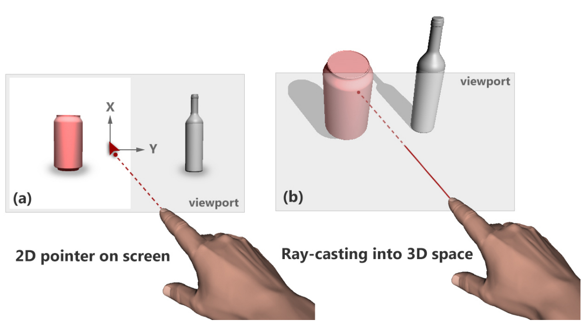

Figure 4: (a) Moving the 2D pointer to the left half of the 2D plane to select the soda can. (b) Pointing at the soda can in 3D space to select it.

5.2 Translation

With an OPS, 3DTouch is capable of drawing or translating an object on a 2D plane. However, this plane's orientation is adjustable by the 3-DOF orientation of the user finger. Fig. 5 illustrates two examples of how the actual touch movements map to a 3D virtual environment (VE). This interaction technique can be applied to both object and screen translation. With 3DTouch, touch interaction can be performed on flat surfaces as well as curved surfaces (Fig. 5c).

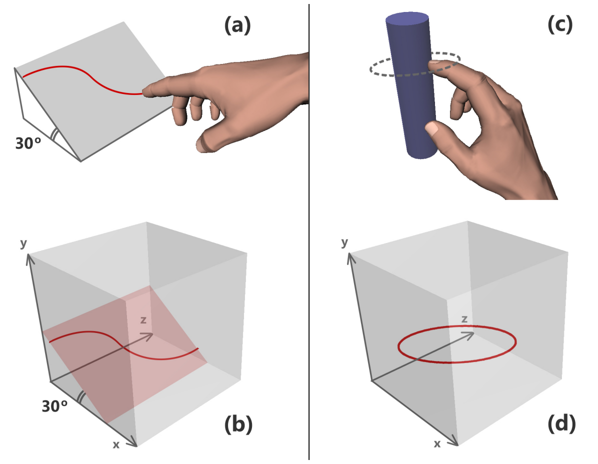

Figure 5: (a) The 3DTouch user is drawing a curve (red) on a flat surface, which makes 30° with the ground. (b) In the 3D VE, a curve is generated on a 2D plane, which also makes 30° with the XZ plane. (c) The 3DTouch user is touching around the curved surface of a cylinder. (d) In the 3D VE, a circle with diameter proportional to that of the cylinder is generated.

5.3 Rotation

Similar to translation, the user draws a vector on a surface to rotate a virtual object in focus. The object will be rotated around the rotation axis, which is perpendicular to the drawn vector on the same 2D plane. The length of the vector drawn is proportional to the rotation angle. Also, the direction of the vector determines the rotation direction. Fig. 6 illustrates an example of how the drawn vector is used to derive the rotation in a 3D VE.

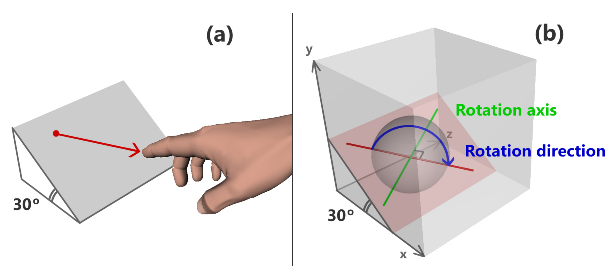

Figure 6: (a) The user is drawing a vector (red) on a flat surface, which makes 30° with the ground. (b) In the 3D VE, the sphere is rotated around the rotation axis by an angle proportional to the vector length. The rotation axis is perpendicular to the vector on the 2D plane, which also makes 30° with the XZ plane.

6 Evaluation of Tracking Accuracy

We conducted an experiment to evaluate the 3D tracking accuracy of our device across multiple surfaces. We compared the 3D position and 3D orientation reported by 3DTouch, against the data obtained using NaturalPoint OptiTrack motion tracking system [21]. In this experiment, we assumed the data obtained from the OptiTrack to be the ground truth. OptiTrack reported a maximum mean error $= 0.8$ mm throughout the whole experiment.

6.1 Setup

3DTouch and OptiTrack both streamed their data via wired connections to a Linux machine with a dual-core 2.1GHz CPU with 4GB of RAM. A program written in C++ synchronized and logged the samples at 50Hz. Our device was configured in the form factor 1 (Fig. 2a), and worn by the first author on the index finger. 12 Flex-13 cameras sampling at 120Hz were used to capture the movements of 3DTouch. A rigid body, composed of three reflective markers, was mounted on top of 3DTouch for it to be tracked by OptiTrack.

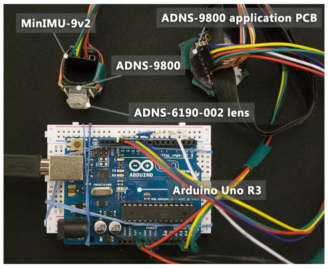

Figure 7: 3DTouch setup: an Arduino Uno R3, an OPS, an IMU, and a sensor application PCB (purple).

6.2 Experimental Design

Since the surface texture is the factor affecting the optical sensing accuracy, we tested the device across 3 textures: mousepad, wooden desk, and jeans. These are three of the environmental textures used as contextual input for Magic Finger [31]. For each texture, we designed 4 different target sizes: 12×12mm, 21×21mm, 42×42mm, and 84×84mm (Fig. 8). We chose 12×12mm as the smallest size because that is the smallest touch area usable by a previous work [8]. The largest area is designed according to the average human palm size [2], which is the touch area for the target mobile applications of 3DTouch.

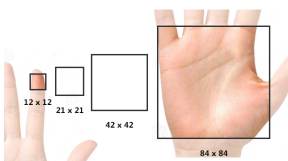

Figure 8: 4 different target sizes: 12×12 mm, 21×21 mm, 42×42 mm, and 84×84 mm.

For each target size, we performed drawing 6 basic shapes: horizontal line, vertical line, diagonal line, triangle, square, and circle. These basic shapes are the building blocks for users to perform 3D interaction techniques and 2D gestures. In total, the experiment design was $3 \times 4 \times 6$ (Texture $\times$ Size $\times$ Shape) with five repetitions for each cell to minimize the human error factor. For each drawing trial, the touch surface is tilted at a random angle within 0 to 90° from the ground.

6.3 Results

There were above 72,000 data points collected in total. We measured the Euclidean error in 3D position and 3D orientation of the directional vector of the data points reported by 3DTouch and OptiTrack (Fig. 10). The mean position error is 1.10 mm (σ = 0.87), and the orientation error is 2.33 degrees (σ = 2.58). As a relative reference, optical mice with similar resolution of 400 cpi, and frame rate of 1500 fps used in mobile robot odometry measurement had the maximum error ≤ 0.8 mm in a 50 mm range [22].

The results showed that the mean position and orientation errors increase with the target sizes (Fig. 9a, b). The three textures tested all have a high SQUAL values between $[50, 90]$. The position errors across the textures (Fig. 9c) did not show significant difference ($F = 2.227$, $p = 0.12$ via Analysis of Variance test).

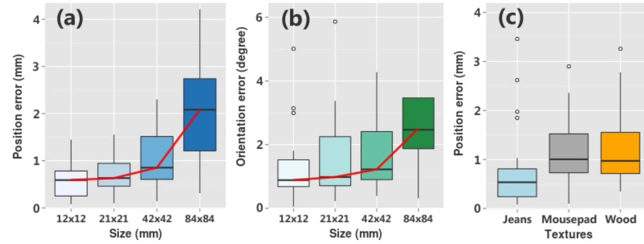

Figure 9: (a) The mean position errors (mm) across 4 target sizes. (b) The mean orientation errors (degree) across 4 target sizes. (c) The mean position errors across 3 textures.

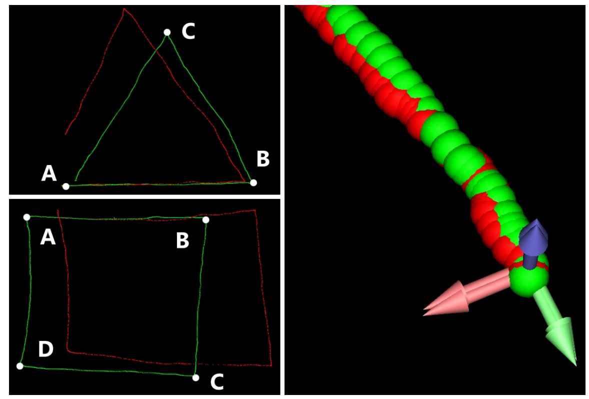

Figure 10: Visualization of the 3D data points reported by 3DTouch (*red*) and OptiTrack (*green*) in Vrui. *Right*: Screenshot of two data clouds (in a random trial) shows very small position and orientation differences. Three vectors (RGB corresponds to x/y/z axes) show the orientation of a pair of points in comparison. *Left*: We show two outlier error cases (top and bottom). These are from two trials (mousepad, 84×84mm, triangle & square) where the errors are highest (≥ 15mm) suggesting acceleration issues with the OPS. The shapes were drawn in the order ABCD.

Visually inspecting the results with the highest errors (Fig. 10), we found that the data points reported by 3DTouch are off but still form a shape similar to that by OptiTrack. There are variable distances from point to point found in the 3DTouch trajectories. This observation suggests that our position error is partly due to the inherent acceleration (up to 30g) in the OPS. We confirmed this acceleration issue by a follow-up experiment (data not shown). In this study, 3DTouch is moved within a physical restricted distance of 1 inch and the result confirmed there is an acceleration issue as the reported distances are falsely increased beyond actual 1 inch as the speed increases within the range $[0, 7]$ inch per second. An OPS with low acceleration such as ADNS-2030 (0.5g) may ameliorate the problem. Therefore, the results presented in this paper should be the baseline performance.

7 Discussion and Future Work

Our system evaluation showed that 3DTouch is capable of performing 3D translation with the mean errors of 1.10 mm and 2.33 degrees. However, a user study will be further conducted to measure usability feedback, especially fatigue and comfort level of our device. In a follow-up study (Anonymous, In prep.), we further evaluate the performance of 3DTouch against the existing VR input devices such as a tracked Wiimote, and a tracked touch tablet [28] across different VR settings (e.g. desktop, wall displays, and CAVE). We also would like to support human skin as the next touch surface and test 3DTouch accuracy on curved surfaces.

3DTouch has a flexible design, supporting multiple form factors. This allows users to wear the device at his comfort finger configuration as desired. Several potential interaction techniques with 3DTouch are: (a) In a CAVE, with 3DTouch worn on the index finger, users can use the palm of the other hand, or the thumb of the same hand as the touch surface; (b) Two or more fingers wearing pieces of 3DTouch would enable multi-touch interaction (essentially becoming a glove but with modular finger pieces); (c) 3DTouch users can interact with curved surfaces (Fig. 5c). This allows users to interact with spherical and other non-flat displays; (d) 3DTouch can be used in combination with a tracking system (similar to a tracked Wiimote [28]) for more robust tracking.

As VR systems evolve, we envision the following applications of our solution: (a) Users wearing wireless HMDs interacting with the 3D VE using 3DTouch in an everyday setting (e.g. in classroom, on the train); (b) 3DTouch can be used in both desktop and mobile settings, meaning users do not have to acquire and learn a new input device as they switch their work to a different VR platform; (c) Wearing 3DTouch means the user has an additional input dimension on the fingertip, our device thus can be combined with other input devices (e.g. touch device and Wiimote) to enable more interaction capabilities.

8 Conclusion

In this paper, we present a novel 3D wearable input device using a combination of a laser OPS, and a 9-DOF IMU. 3DTouch enables users to use their fingers or thumb as a 3D input device with the capability of performing 3D selection, translation, and rotation. 3DTouch is designed to fill the missing gap of a 3D input device that is self-contained, mobile, and universally working across various 3D platforms. This paper presents a low-cost solution to designing and implementing such a device. Modular solutions like 3DTouch opens up a whole new design space for interaction techniques to further develop on. With 3DTouch, we attempted to bring VR applications a step closer to users in everyday life, across both mobile and desktop settings.

Acknowledgements

The authors would like to thank George Janack, Dr. Jerry Hamann and John Kicklighter for their helpful advice regarding the hardware implementation. We also thank Dr. Steven Barrett and Dr. Jeff Clune for their helpful feedback and UW School of Energy Resources Fellowship for funding (Anh Nguyen).

References

[1] ADNS-9800 datasheet. `http://www.pixart.com.tw`.

[2] A. Agnihotri, B. Purwar, N. Jeebun, and S. Agnihotri. Determination of sex by hand dimensions. *Internet Journal of Forensic Science*, 1(2), 2006.

[3] H. Benko, A. D. Wilson, and P. Baudisch. Precise selection techniques for multi-touch screens. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1263–1272. ACM, 2006.

[4] X. Bi, T. Grossman, J. Matejka, and G. Fitzmaurice. Magic desk: bringing multi-touch surfaces into desktop work. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 2511–2520. ACM, 2011.

[5] D. A. Bowman. *3D User Interfaces*. The Interaction Design Foundation, Aarhus, Denmark, 2013.

[6] D. A. Bowman, E. Kruijff, J. J. LaViola Jr, and I. Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley, 2004.

[7] X. Cao, A. D. Wilson, R. Balakrishnan, K. Hinckley, and S. E. Hudson. Shapetouch: Leveraging contact shape on interactive surfaces. In *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*, pages 129–136. IEEE, 2008.

[8] L. Chan, R.-H. Liang, M.-C. Tsai, K.-Y. Cheng, C.-H. Su, M. Y. Chen, W.-H. Cheng, and B.-Y. Chen. Fingerpad: private and subtle interaction using fingertips. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 255–260. ACM, 2013.

[9] K.-Y. Chen, K. Lyons, S. White, and S. Patel. utrack: 3d input using two magnetic sensors. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 237–244. ACM, 2013.

[10] Finger Mouse. `http://www.logisyscomputer.com`.

[11] B. Frohlich, J. Plate, J. Wind, G. Wesche, and M. Gobel. Cubic-mouse-based interaction in virtual environments. *Computer Graphics and Applications, IEEE*, 20(4):12–15, 2000.

[12] C. Harrison, H. Benko, and A. D. Wilson. Omnitouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 441–450. ACM, 2011.

[13] C. Harrison and S. E. Hudson. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 121–124. ACM, 2009.

[14] C. Harrison, D. Tan, and D. Morris. Skinput: appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 453–462. ACM, 2010.

[15] O. Kreylos. Environment-independent vr development. In *Advances in Visual Computing*, pages 901–912. Springer, 2008.

[16] H. J. Luinge, P. H. Veltink, and C. T. Baten. Estimating orientation with gyroscopes and accelerometers. *Technology and health care*, 7(6):455–459, 1999.

[17] J. L. Marins, X. Yun, E. R. Bachmann, R. B. McGhee, and M. J. Zyda. An extended kalman filter for quaternion-based orientation estimation using marg sensors. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 4, pages 2003–2011. IEEE, 2001.

[18] N. Matsushita and J. Rekimoto. Holowall: designing a finger, hand, body, and object sensitive wall. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 209–210. ACM, 1997.

[19] M. Nancel, J. Wagner, E. Pietriga, O. Chapuis, and W. Mackay. Mid-air pan-and-zoom on wall-sized displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 177–186. ACM, 2011.

[20] T. Ni and P. Baudisch. Disappearing mobile devices. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 101–110. ACM, 2009.

[21] OptiTrack. `www.naturalpoint.com/optitrack/`.

[22] J. Palacin, I. Valganon, and R. Pernia. The optical mouse for indoor mobile robot odometry measurement. *Sensors and Actuators A: Physical*, 126(1):141–147, 2006.

[23] G. Ramos, M. Boulos, and R. Balakrishnan. Pressure widgets. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 487–494. ACM, 2004.

[24] T. S. Saponas, C. Harrison, and H. Benko. Pockettouch: through-fabric capacitive touch input. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 303–308. ACM, 2011.

[25] F. Wang, X. Cao, X. Ren, and P. Irani. Detecting and leveraging finger orientation for interaction with direct-touch surfaces. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 23–32. ACM, 2009.

[26] M. Weiss, C. Wacharamanotham, S. Voelker, and J. Borchers. Fingerflux: near-surface haptic feedback on tabletops. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 615–620. ACM, 2011.

[27] G. Welch and E. Foxlin. Motion tracking survey. *IEEE Computer graphics and Applications*, pages 24–38, 2002.

[28] C. B. Wilkes, D. Tilden, and D. A. Bowman. 3d user interfaces using tracked multi-touch mobile devices. In *Joint Virtual Reality Conference of ICAT-EGVE-EuroVR*, pages 65–72. The Eurographics Association, 2012.

[29] A. D. Wilson, S. Izadi, O. Hilliges, A. Garcia-Mendoza, and D. Kirk. Bringing physics to the surface. In *Proceedings of the 21st annual*

ACM symposium on User interface software and technology, pages 67–76. ACM, 2008.

[30] Windows 7. http://msdn.microsoft.com/en-us/library/windows/desktop/bb760404(v=vs.85).aspx.

[31] X.-D. Yang, T. Grossman, D. Wigdor, and G. Fitzmaurice. Magic finger: always-available input through finger instrumentation. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 147–156. ACM, 2012.

[32] R. C. Zeleznik, J. J. LaViola, D. Acevedo Feliz, and D. F. Keefe. Pop through button devices for ve navigation and interaction. In *Virtual Reality, 2002. Proceedings. IEEE*, pages 127–134. IEEE, 2002.