Lift: Using Projected Coded Light for Finger Tracking and Device Augmentation

Shang Ma[1,2], Qiong Liu[2], Chelhwon Kim[2], Phillip Sheu[1]

EECS, UC Irvine[1]
Irvine, California
{shangm, psheu@uci.edu}

FX Palo Alto Laboratory[2]
Palo Alto, California
{liu, kim@fxpal.com}

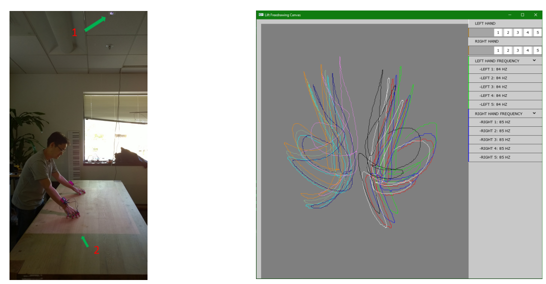
***Abstract*—We present Lift, a visible light-enabled finger tracking and object localization technique that allows users to perform freestyle multi-touch gestures on any object's surface in an everyday environment. By projecting encoded visible patterns onto an object's surface (e.g. paper, display, or table), and localizing the user's fingers with light sensors, Lift offers users a richer interactive space than the device's existing interfaces. Additionally, everyday objects can be augmented by attaching sensor units onto their surface to accept multi-touch gesture input. We also present two applications as proof of concept. Finally, results from our experiments indicate that Lift can localize ten fingers simultaneously with an average accuracy of 1.7 millimeter and an average refresh rate of 84 Hz with 31 milliseconds delay on WiFi and 23 milliseconds delay on serial communication, making gesture recognition on non-instrumented objects possible.**

Keywords*—*Coded light; projector-based tracking; finger tracking; device augmentation

I. Introduction

Multi-touch gesture-based interaction on an instrumented surface is well suited for a variety of applications because of its simplicity and intuitiveness. Most of the laptops currently on the market have inbuilt touchpads providing a user-friendly gesture interface. For desktop computers, products like Logitech K400 [1] and T650 [2] are also available to allow multi-touch interaction besides the traditional mouse and keyboard interface. These examples and the popularity of the touchscreen-enabled smartphone demonstrate the usability and acceptability of such an interface among users. In this work, we present Lift, a visible light-based finger tracking technique that employs an off-the-shelf projector and light sensors to enable direct multi-touch gesture control on any device's surface. Fig. 1 presents a simple scenario where a user directly performs freestyle drawing on an uninstrumented table using ten fingers.

Lift is implemented by embedding location information into visible patterns and projecting these patterns onto a target surface. Light sensors are attached on both a device's surface and the user's fingers for localization and gesture tracking. In doing so, Lift not only removes the need for a dedicated touchscreen on the device or an external touchpad for gesture interaction, but it also enables users to have direct control on the device's surface, whether it is originally instrumented or not. More importantly, Lift features the richness of interaction space and simplicity in physical size, hardware complexity, and computation load, meaning Lift is compact and easy-to-use while allowing users to interact with any device. This is made possible by using a projection-based interaction method and encoding position information into each pixel of the projection area. The perspective of the projection-based encoded position is desirable for sensing, positioning, and processing in a variety of scenarios where computation resources are limited. Because all pixels in the projection area have been encoded with location information and the only step needed to restore the corresponding position is to decode a sequence of sensor readings, Lift can locate and track light sensors without the need of heavy computation and therefore enable a fast response in finger tracking and gesture operation in many applications. This advantage sets it apart from all existing approaches.



(a) (b)

Fig. 1. (a) 1. A DLP projector behind the ceiling; 2. An office table as interaction space. (b) The user performed freestyle drawing on a non-instrumented table and the traces of all fingers were recorded.

II. Related Work

A. Finger Tracking

Finger tracking has been explored by a number of previous research projects. In particular, Kramer's glove [3] was one of the first to track a user's finger postures using a strain gauge attached to a glove. However, the position of the strain garget on the glove may shift when the hand changes its pose, which could reduce tracking accuracy.

Visual tracking approaches, in which the camera is either the only or the main sensor, are generally considered to be a dominating technique covering a wide field of applications [4, 5, 6, 7]. The main disadvantages of this technique include the significant power consumption and being subject to various factors, such as camera resolution, lighting condition, and the reflective properties of target surfaces. All of these add a substantial amount of infrastructure, cost, and complexity to achieve the desired effect and can be challenging to implement in practice.

Sensor-based approaches can also provide a greater range of data and create good gesture recognition systems. For example, infrared proximity sensor [8], magnetic sensing [9], electric field [10], acoustics [11], ultrasound imaging [12], wrist-worn camera + laser projector [13], ultra-wideband signal [14], muscle and tendon sensing [15], and data glove [16] have all been explored in previous studies. However, some of these systems operate in a relatively short range, and others only support a small set of hand gestures. In contrast, Lift supports fine-grained continuous tracking in a much larger physical space.

B. *Encoded Projection*

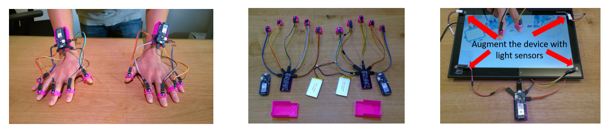
Encoded projection is used in a variety of applications, including projection calibration [17], tracking movable surfaces/cars [18, 19, 20], GUI adaptation and device control [21], and ambient intelligence [22]. These works employ visible pattern projection through a Digital Light Processing (DLP) projector to transmit location information and restore original position data by sensing light intensity via light sensors. This mechanism leverages one important property of a DLP projector: the fast-flipping tiny micro mirrors inside can be used to encode and modulate the projected light.

However, none of these technologies is fast enough to track multiple fast-moving objects. Lee et al. [18] demonstrated that an update rate of 11.8 Hz was achieved when 10 patterns were used to resolve a 32×32 unit grid. Summet et al. [20] projected 7.5 packets per second in their system, which is "just enough to track slow hand motions, approximately 12.8 cm/sec when 1.5 m from the projector". Instead, Lift has achieved a tracking speed of 84 Hz for fingers moving at a speed of 46.1 cm/sec. Ma et al. [19] used an Android phone to decode received light signals increasing the refresh rate to 80 Hz. However, in their system, only one light sensor can be decoded at a time. In contrast, our implementation has achieved an average update rate of 84 Hz while tracking ten light sensors simultaneously. This is a significant improvement over previous works. [23] and [24] did achieve a higher system update rate, but they required a specially made beamer for one bit of spatial resolution. Lift, however, only needs a single off-the-shelf projector for all 1,039,680 (1140×912) pixels.

III. Prototype Implementaion

To assess the feasibility of our tracking approach, and to experiment with different interaction applications, we constructed a prototype platform (Fig. 1 & 2). Our setup consists of four components: (1) an off-the-shelf DLP projector from Texas Instruments [25] that can project gray-coded binary patterns onto the target surface with a projection area of 1380 mm × 860 mm at a distance of 2 m, (2) tiny sensors which will be attached to a user's ten fingers and the surface of a target device, (3) two Arduino MKR1000 boards [26] with batteries, one for each hand, and (4) two custom printed circuit boards for basic signal conditioning.

Gray-coded patterns are used to encode every pixel in the projection area and to provide the unique coordinates so that when a light sensor detects the presence or absence of the projected light, a bit sequence representing the coordinates of this specific pixel can be received and decoded, retrieving the *x* and *y* pixel coordinates. Because projecting images into the environment is the inbuilt function of our projector, Lift does not require any augmentation on the projector itself. Additionally, communication in Lift currently takes place in a unidirectional fashion (from the projector to light sensors), and position decoding is performed on the Arduino board. Therefore, Lift does not need central infrastructure for heavy computation. This simplicity allows us to maintain a minimalist system design, while still being able to provide a rich interaction space on different object surfaces. Finally, the two Arduino boards used in the current design have inbuilt WiFi communication modules and therefore make Lift suitable for mobile applications.



(a) (b) (c)

Fig. 2. (a) The user wears ten sensors on his fingers. (b) Electronics components: light sensors on rings, Arduino and signal conditioning boards, and batteries. (c) Light sensor units on a target device's surface.

A. *Projector & Encoded Patterns*

Our current projector has a native resolution of 1140×912 pixels and a projection frequency of 4 kHz. Given that gray-codes have a $O(log_2 n)$ relationship between the number of required patterns and the total number of pixels, *n*, inside the projection area, 21 ($log_2 912 + log_2 1140$) gray-coded images are needed to uniquely locate each pixel of our interaction space. Fig. 3 demonstrates an example in which 3 horizontal (left three) and 3 vertical patterns (right three) are used to resolve an 8×8 unit grid.

Fig. 3. An example of gray code images for an 8×8 grid.

B. *Encoding & Decoding Scheme*

In previous work [17, 18, 20], the absolute light intensity value of light sensors is used to determine the bit value for each projection frame. This design is subject to ambient lighting, the variance of the light sensors, and the analog-to-digital converter inside the microcontroller on which the decoding software is executed. Instead, we use Manchester coding [27] to transmit the data patterns. That is, each of our 21 gray-coded images is projected onto the target device followed by its reverse pattern. This removes the dependency on the DC voltage of the received light intensity, which can be influenced severely by lighting conditions and light sensor reception efficiency. However, this completes the necessary number of patterns for location discovery from 21 to 42. In addition, we add 5 framing bits at the beginning of each packet, allowing sensor units to be synchronized with the data source, further increasing the necessary patterns to 47. Unsurprisingly, this encoding scheme also helps with the flickering issue of visible light communication, which is usually due to long runs of *black* or *white* and may cause serious detrimental physiological changes in humans if the fluctuation in the brightness of the light is perceivable to human eyes [28]. In Lift, each *black* or *white* bit is followed by its reverse bit, and the number of the *white* in a packet is only one more than the number of *black*. This design frees the system from fluctuation so that users always see the projection area as a static image without flickering (Fig. 1a).

In Lift, two Arduino MKR1000 boards are used to decode the received light signals. More specifically, five digital I/O pins on each board are configured as input to evaluate the voltage level using a threshold approach. The firmware running on the microcontroller keeps collecting the light intensity every 250 microseconds and restoring the original position data for all fingers simultaneously.

C. *Projector-Light Sensor Homography*

Lift is designed to provide fine-grained 2D physical coordinates of a user's fingers on an uninstrumented surface. To obtain physical coordinates of a user's fingers, the position, orientation, and optical parameters of the projector relative to the interaction surface should be known. Consider that there exists a point (x', y') on the DMD (Digital Micromirror Device) mirror array inside the projector, and it is projected to a point on a projection screen, for instance a flat table, with physical coordinates (x, y) (in mm). (Here we assume that the origin on the table is already known.) The relationship between (x', y') and (x, y) is determined by a planar projective transformation whose parameters further depend on the position and orientation of the projector relative to the table, and the optical parameters of the lens inside the projector.

In Lift, (x', y') coordinates can be observed and decoded by a light sensor while (x, y) can be measured relative to a user-defined origin. Therefore, the perspective transformation between the projector plane and the table can be calculated with the following steps: (1) marking a certain number of points in the projection area, (2) measuring the distances between these points and the origin of the physical plane, which, in our case, is the center of an office table shown in Fig. 4, and (3) collecting the pixel coordinates of these points using a light sensor. The inbuilt *cv::findHomography* function in OpenCV can be used to find the transformation matrix, and this result could be applied to future sensor readings to obtain the Euclidean coordinates of a projected point.

IV. Evaluation & Results

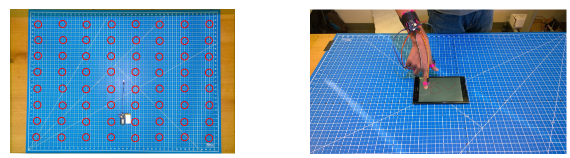
A projector-based finger tracking system offers the capability to directly compute the location, moving speed and orientation of ten fingers in a large interaction space. These quantities can be used in a variety of applications, such as gaming control, augmented interaction space, and more. Here, we evaluate several parameters in the system, including: (1) tracking accuracy, (2) tracking resolution, (3) communication delay, (4) system refresh rate, and (5) system robustness under different lighting conditions.

A. *Tracking Accuracy*

1) *Study Design for Discrete Points*

To evaluate the position estimation accuracy of our proposed system for different points, a test arena was developed as shown in Fig. 4a. In this experiment, the projector was installed behind the ceiling, and a flat office table was used as an interaction space. The distance between the projector and the table is 2 meters, and the projection area on the table is 1380 mm × 860 mm (54.3 inch × 33.85 inch). A 48×36 inch cutting mat with 0.5 inch grid was used to provide high-accuracy ground truth data. Since the cutting mat does not cover the whole projection area, we performed our experiment using points inside the cutting mat area.

Fig. 4 shows that totally 64[1] points (black dots at the center of red circles) were marked for data collecting, and they were selected to be uniformly distributed in the projection area. A sensor unit was placed at these points to collect pixel coordinates. Their physical distances with regard to the center were also measured and the homography was calculated.



(a) (b)

Fig. 4. (a) A test arena for calculating the homography. (b) Tracking the user's gestures using both an Android tablet and Lift.

We also collected the pixel locations of another 56 points, which were different from the 64 points from the first step, and applied the homography to them. The physical coordinates of these 56 points relative to the center were measured as ground truth. We compared the difference of these two quantities for all 56 points.

2) *Results*

Fig. 5 shows the results of tracking errors at different points in the projection space. The difference is defined as the Euclidean distance between the computed coordinates from Lift and the ground truth. An average error of 1.707 mm for the entire projection space was reported, and the standard deviation was 0.918 mm. This demonstrates that Lift achieves its goal of millimeter-level finger tracking in practice.

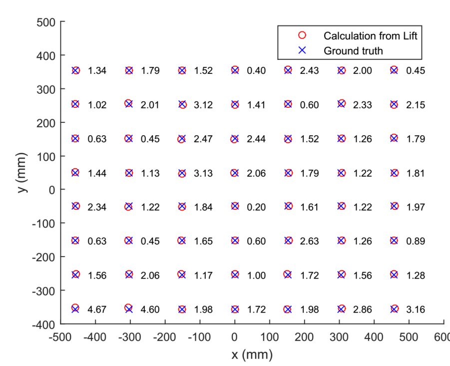


Fig. 5. Tracking error of all 56 discrete points.

3) *Study Design for Continuous Gestures*

[1]Lift achieves an average accuracy of 1.761 mm if 16 points are used. About ten seconds is needed for one point during this calibration process.

We also evaluated the tracking accuracy for continuous gestures, as shown in Fig. 4b. We divided the projection area into 4 × 4 grids, where each one has the size of 267 × 178 mm. At the center of each grid, we placed an Android tablet (Google Nexus 9) and asked participants to draw freely using one of their fingers on the tablet. The screen size of the tablet is about 180 × 134 mm and the resolution is 2048 × 1536. A program was developed to obtain the pixel coordinates when participants made a touch gesture on the screen and to draw the traces of these touch points as well. Since this location data represents pixel coordinates on the screen, we converted this quantity into their corresponding physical locations (in mm) by scaling the pixel value with the number of pixels per millimeter and also offset the result with the distance between the origin on the table and the origin on the tablet, which is the top left corner in our design. The resulting locations were used as the ground truth data for this experiment. Simultaneously, the physical location of finger movements collected by Lift was also calculated based on sensor readings and the homography.

We also noticed that there is certain offset between the position of sensor units on fingers and the fingertips for touch interaction on the tablet. Therefore, we offset all the data points from Lift with the distance between the first point of touch interaction and the first point of data collection from Lift. Here, we assume that this offset is consistent during the entire motion of a gesture once the gesture starts and we used the physical coordinates of these two points to calculate their distance.

Six participants were invited for this experiment. Three of these participants are female, and other three are male; none of them were provided with any monetary benefits. Each participant was asked to draw 3 traces in a grid, meaning 192 trials for each participant and 1152 trails in total.

4) Results

To compute the tracking error for continuous gestures, we calculated the average least perpendicular distance of each point collected by touch events on the tablet with the corresponding trajectory formed by the outputs from Lift, and we performed this measurement for each grid by averaging across trials. Fig. 6 shows the average tracking error for each grid. The average continuous tracking error for the whole interaction area is about 1.631 mm, the standard deviation is 0.35 mm, and the maximum deviation is 1.032 mm.

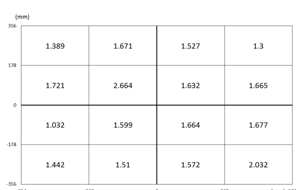


Fig. 6. Average tracking error of continuous gestures for the 16 grids.

B. Tracking Resolution

From the same dataset used in previous section, we also calculated the resolution, PPI (pixels per inch), for each grid of the projection area. As shown in Fig. 4a, 64 points divide the cutting mat into 49 grids and the size of a single grid is 6 inch by 4 inch. We first applied the homography to these 64 points to get their physical coordinates, and then calculated the resolution of each grid using (1):

$$PPI = \sqrt{w^2 + h^2} \,/\, d \qquad (1)$$

where w is the width resolution, h is the height resolution in pixel, and d is the diagonal size in inch. The result is shown in Fig. 7. Additionally, the average PPI for the whole projection space is about 25.22.

25.0	25.2	25.3	25.3	25.1	25.2	25.4
25.0	25.0	25.1	25.4	25.4	25.2	25.1
25.3	25.2	25.5	25.2	25.4	25.5	25.4
24.9	25.3	25.5	25.0	25.3	25.3	25.4
25.0	25.3	25.0	25.2	25.3	25.2	25.3
25.2	24.9	25.0	25.3	25.2	25.2	25.3
25.0	25.1	25.5	25.3	25.5	25.1	25.1

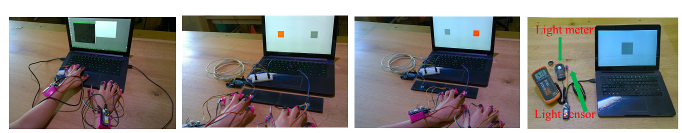
Fig. 7. The projection area was divided into 6 inch × 4 inch grids, 49 grids in total. The resolution (pixel per inch) was calculated for each grid.

C. Latency

The total system performance is also affected by latency at multiple levels. First, the optical packet itself is 11.75 milliseconds long. Second, the microcontrollers add a certain delay while decoding all the positions and packing data for transmission. Finally, to make Lift portable and compatible with existing mobile/IoT devices, we chose WiFi for communication between Lift and target devices and Open Sound Control (OSC) [29] is used as the data transmission protocol because of its simplicity and robustness. This WiFi connection is the communication bottleneck of our system; the WiFi bandwidth is shared between two Arduino boards in Lift with other devices in the testing environment, including desktops, laptops, mobile phones, and a variety of other wireless devices. Therefore, we designed the following experiment to measure the latency of the proposed system.

We used a simple setup to measure the delay between the time when Lift receives a “*start*” command for data collection and the time when a laptop (2.2 GHz CPU, 8 GB memory) receives the decoded position through the WiFi connection and applies the homography on the sensor readings to get the final position. Fig. 8a demonstrates our setup for this experiment.

A program was developed to run on a laptop recording a timestamp of a keypress event from the keyboard. Whenever the user pressed the “*space*” key, the corresponding timestamp was recorded by the program, which also notified Lift to start the position detection process through serial communication running at 115200 baud. In turn, Lift will send position data of all ten fingers to the same laptop through WiFi connection once it finishes data processing. The time interval between this program detecting the keyboard event and it computing final finger positions was considered as the system latency.



(a) (b) (c) (d)

Fig. 8. Experiment setup to evaluate (a) system latency, (b, c) system refresh rate, and (d) system reliability under different light condition.

The same six participants were invited to collect 1200 groups of time difference, and an average latency of 31 milliseconds was reported while using WiFi connection. It is worth mentioning that when serial communication (115200 baud) was used to send data from Lift to the host PC, the delay decreased to 23 milliseconds.

D. Refresh Rate

1) Study Design

To ensure smooth finger tracking, the maximum speed that fingers can move in the space also plays an important role in a finger tracking system. The same six participants were invited for the following evaluation to find out the number of positions that Lift can decode at different movement speeds.

The setup of this evaluation is shown in Fig. 8b & 8c, and the procedure is demonstrated as follows. All participants were asked to wear 10 sensor units on their fingers and touch two touchpads (29 mm by 16 mm) in a fixed order (left one first, then right one) at different speed modes: *normal*, *medium*, and *as fast as possible*. The time when these two touch sensors are activated is used as timestamps indicating the beginning and ending of a single test. Based on the time difference and the distance between these two pads, the movement speed of user's finger can be measured. To make our experiment as accurate as possible, the participants were asked to move their fingers in a straight line. They were given a demonstration of the system first and were allowed to practice with it. Once they were familiar with the procedure, we started the experiment.

In our experiment, the distance between these two pads was chosen to be 20 cm, which is long enough so that participants have to move their fingers for a measurable period of time, but not too long that they have to move or stretch their body to access the second touchpad. The time interval between two touch events was detected by an Arduino microcontroller and sent to a laptop through serial communication (115200 baud) for data logging. A program running on the same laptop was used to count the number of position packages Lift collects through WiFi during this time interval. Each speed case was repeated 10 times. Participants can use any finger they wish for the touch action, but they have to use the same finger during a single test. 30 tests were performed for each user and 180 tests in total for all six participants. Participants can take a rest and change to another finger between the tests.

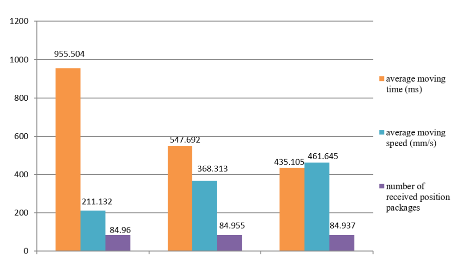


Fig. 9. Refresh rate of Lift at different movement speeds.

2) Results

The average moving time between these two touchpads is *955.504*, *547.692*, and *435.105* milliseconds for the three speed modes respectively, shown in Fig. 9. Fig. 9 also shows average speeds, v, calculated by (1):

$$v = \frac{\textit{the distance between two touchpads}\,(20\ cm)}{\textit{the time interval between two touch events}} \tag{2}$$

which are *211*, *368*, and *461* mm/second. The average number of position packages received at these three different speeds is *84.365*, *84.424*, and *84.087*, and the standard deviation is 0.179, 0.325, and 0.170, respectively. Each package contains the position data of all ten fingers. This agrees with the theoretical maximum update rate of Lift system, which is 85.1 Hz and decided by the total number of patterns our projector can send every second and the length of one position packet. This quantity, r, can be calculated simply by (2):

$$r = \frac{4000\,(\textit{total number of patterns per second})}{47\,(\textit{number of patterns for a single position})} \approx 85.1 \tag{3}$$

The results demonstrate that Lift can maintain high refresh rate even when users move their fingers at a high speed.

E. Lighting Conditions

1) Study Design

We also tested the robustness of our current implementation under different lighting conditions by putting a light sensor at a known place and logging its decoded coordinates at different indoor ambient light (Fig. 8d). The light intensity was measured by a light meter placed next to the sensor unit. The main metric we used here to quantify the system reliability is the percentage of correct readings.

2) Result

Table I below shows the percentage of correct readings at different lighting conditions. The first column is the light intensity of the environment under different conditions. Since the current implementation of Lift is based on a visible light projector, the projector used in the system will increase the light intensity of the projection area by a certain range. Here, the second column in the table shows the light intensity of the projection space after Lift is activated. The results indicate that our tracking technique can work reliably when ambient light is in the range of 0 Lux to 345 Lux. However, if the environment is brighter than this threshold, Lift fails to work. We will explore how to solve this problem in section VI.

TABLE I. LOCALIZATION ACCURACY UNDER DIFFERENT LIGHTING CONDITIONS

Ambient Light Intensity (Lux)	***Ambient Light + Lift Intensity (Lux)***	***Accuracy Percentage (%)***
21	150	100
232	384	100
312	444	100
336	464	98.7
345	472	94.4
349	482	22
355	487	0
400	538	0
483	614	0

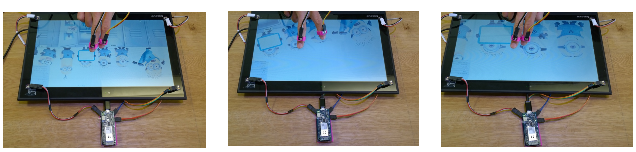
V. Example Application

As part of our investigations, we also implemented demo applications on two interaction surfaces with different form factors and input styles. For the tablet-size application, we developed a simple photo browsing album on a non-touchscreen laptop (Fig. 10). Users can navigate through a collection of pictures on an uninstrumented display directly using multi-touch gestures. We also implemented a ten-finger freestyle drawing application (Fig. 1), with which a user can draw any freeform traces on a common office table.

A. *Gesture Tracking on An Uninstrumented Display*

For the album application, a projector was used to project the aforementioned patterns onto a table with the non-touchscreen laptop inside the projection area. Four light sensors were fixed at the corners of the display of the laptop, one in each corner, indicating the boundary of the target device (Fig. 2c). A user was then asked to wear two sensor units, one on the index finger and the other on the middle finger (Fig. 10).

As a proof of concept, we designed four simple multi-touch gestures: *pan*, *zoom in/out*, and *rotation*. To pan an image that is larger than the display, the user moves two fingers on the screen at the same time to view different areas of the image (Fig. 10a). As the fingers move, their position data are saved, and decoded. The distance the data travels is used to drag the image around. The rotation gesture is similar to Mac or iPhone products, where users can move two fingers around each other to rotate an image (Fig. 10b). To zoom in/out of images, users need to put two fingers on the screen and pinch them to zoom in or out (Fig. 10c). Other gestures, such as swipe and scroll, can be implemented based on these sensor data as well.



(a) (b) (c)

Fig. 10. With Lift, a user can (a) pan an image, (b) rotate an image, (c) zoom in/out an image on an uninstrumented display.

B. *Freestyle Drawing on An Uninstrumented Tabletop*

Another application we have implemented is a freestyle drawing tool. Thanks to our projection-based tracking mechanism, it gives us a very large interaction area: 1380 mm × 860 mm at a projection distance of 2 m (Fig. 1a). However, this can be further increased by adding a wide angle lens in front of the projector to allow a much larger interaction space, such as a wall. We will leave this for future explorations.

In this experiment, ten light sensors were attached to a user's ten fingers. The user can freely move his fingers in the interaction space at a relatively high speed. The output signals of all these sensors were then collected and decoded on the two Arduino boards, one for each hand, and sent to a laptop computer for visualization purpose. A GUI application was used to display the traces of the user's ten fingers, which are color coded as well (Fig. 1b).

VI. Limitations & Future Directions

The experimental results suggest that projection-based location discovery is a promising approach for fine-grained finger tracking in different interaction scenarios. However, our study and exemplary applications brought to light several limitations and challenges, which include the following.

A. *Interference from ambient light*

Our current implementation projects binary patterns through visible light. To make position decoding possible, the light sensor we use, TSL13T, is more responsive to visible light, around 750 nm. However, this also brings up the problem of interference from ambient light. In section IV, we have tested the proposed system in different lighting conditions and found that Lift fails to work with strong ambient light. This is, however, not a fundamental limitation of our approach and can be addressed by modulating the visible light with high frequency carriers, which would allow us to track fingers under various challenging conditions.

B. *Interacting with Projected Content*

The projection-based finger tracking mechanism only leverages the pattern projection mode of the projector. But the projector itself also supports video projection as a normal video projector. We envision a scenario where users can directly and intuitively manipulate projected multimedia contents using Lift. This application combines video projection and pattern projection modes, providing finger tracking, content manipulation, and visual feedback all at the same time. We will leave the development of such an application for future work.

C. *2D versus 3D*

Currently, Lift only supports 2D gesture interaction, meaning users need to place their fingers on top of the calibrated surface and keep all the light sensors facing upwards before performing any gestures. We are interested in exploring 3D finger tracking using projected patterns. In principle, 3D depth information can be available if multiple projectors are used. We are also curious about how different types of inertial sensors can be incorporated into the system to automatically detect the starting and ending of gestures.

VII. Conclusion

We introduce Lift, a coded visible light-based object localization and finger tracking technique in an everyday environment. Use of projected coded light enables computation-efficient finger tracking with an extended interaction space on any device's surface. We evaluated our design through a series of experiments, validating that Lift can track ten fingers simultaneously with high accuracy (1.7 mm), high refresh rate (84 Hz), and low latency (31 ms for WiFi, and 23 ms for serial) under different ambient light conditions. We also developed two applications where Lift makes use of the projection area on uninstrumented surfaces to track fast moving fingers and perform gesture recognition to control the devices. With this work, we successfully demonstrate that Lift holds significant promise for future research direction.

REFERENCES

[1] Wireless Touch Keyboard K400. http://www.logitech.com/en-us/product/wireless-touch-keyboard-k400r

[2] Wireless Rechargeable Touchpad T650 http://support.logitech.com/en_us/product/touchpad-t650

[3] J. P. Kramer, P. Lindener, and W. R. George, "Communication system for deaf, deaf-blind, or non-vocal individuals using instrumented glove," U.S. Patent 5,047,952. Filed October 14, 1988, issued September 10, 1991.

[4] W. Hürst, and C. Van Wezel, "Gesture-based interaction via finger tracking for mobile augmented reality," Multimedia Tools and Applications, Vol. 62, No. 1, pp. 233-258, 2013.

[5] H. Koike, Y. Sato, and Y. Kobayashi, "Integrating paper and digital information on EnhancedDesk: a method for realtime finger tracking on an augmented desk system," ACM Trans. Computer-Human Interaction, Vol. 8, No. 4, pp. 307-322, 2001.

[6] S. Henderson and S. Feiner, "Opportunistic tangible user interfaces for augmented reality," IEEE Trans. Visualization and Computer Graphics, Vol. 16, No. 1, pp. 4-16, 2010.

[7] C. Harrison, H. Benko, and A. D. Wilson, "OmniTouch: wearable multitouch interaction everywhere," 24th Annual ACM Symp. User Interface Software and Technology, p. 441-450, 2011.

[8] W. Kienzle and K. Hinckley, "LightRing: always-available 2D input on any surface," 27th Annual ACM Symp. User Interface Software and Technology, p. 157-160, 2014.

[9] K. Y. Chen, S. Patel, and S. Keller, "Finexus: Tracking Precise Motions of Multiple Fingertips Using Magnetic Sensing," CHI Conf. Human Factors in Computing Systems, p. 1504-1514, 2016.

[10] M. Le Goc, S. Taylor, S. Izadi, and C. Keskin, "A low-cost transparent electric field sensor for 3d interaction on mobile devices," CHI Conf. Human Factors in Computing Systems p. 3167-3170, 2014.

[11] A. Mujibiya, X. Cao, D. S. Tan, D. Morris, S. N. Patel, and J. Rekimoto, "The sound of touch: on-body touch and gesture sensing based on transdermal ultrasound propagation," ACM Conf. Interactive tabletops and surfaces, p. 189-198, 2013.

[12] M. Sagardia, K. Hertkorn, D. S. González, and C. Castellini, "Ultrapiano: A novel human-machine interface applied to virtual reality," IEEE Conf. Robotics and Automation, p. 2089, 2014.

[13] D. Kim, O. Hilliges, S. Izadi, A. Butler, J. Chen, I. Oikonomidis, and P. Olivier, "Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor," 25th Annual ACM Symp. User Interface Software and Technology, p. 167-176, 2012.

[14] S. Wang, J. Song, J. Lien, I. Poupyrev, and O. Hilliges, "Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum," 29th Annual ACM Symp. User Interface Software and Technology, p. 851-860, 2016.

[15] S. Scott, D. S. Tan, D. Morris, R. Balakrishnan, J. Turner, and J. A. Landay, "Enabling always-available input with muscle-computer interfaces," 22nd Annual ACM Symp. User Interface Software and Technology, p. 167-176, 2009.

[16] J.H. Kim, N.D. Thang, and T.S. Kim, "3-d hand motion tracking and gesture recognition using a data glove," IEEE Symp. Industrial Electronics, p. 1013-1018, 2009.

[17] J. C. Lee, P. H. Dietz, D. Maynes-Aminzade, R. Raskar, and S. E. Hudson, "Automatic projector calibration with embedded light sensors," 17th Annual ACM Symp. User Interface Software and Technology, p. 123-126, 2004.

[18] J. C. Lee, S. E. Hudson, J. W. Summet, and P. H. Dietz, "Moveable interactive projected displays using projector based tracking," 18th Annual ACM Symp. User Interface Software and Technology, p. 63-72, 2005.

[19] S. Ma, Q. Liu, and P. Sheu, "On hearing your position through light for mobile robot indoor navigation," IEEE Conf. Multimedia and Expo Workshops, 2016.

[20] J. Summet, and R. Sukthankar, "Tracking locations of moving hand-held displays using projected light," Conf. Pervasive Computing, p. 37-46, 2005.

[21] D. Schmidt, D. Molyneaux, and X. Cao, "PICOntrol: using a handheld projector for direct control of physical devices through visible light," 24th Annual ACM Symp. User Interface Software and Technology, p. 379-388, 2012.

[22] R. Raskar, P. Beardsley, J. Van Baar, Y. Wang, P. Dietz, J. Lee, D. Leigh, and T. Willwacher, "RFIG lamps: interacting with a self-describing world via photosensing wireless tags and projectors," ACM Trans. Graphics, Vol. 23, No. 3, pp. 406-415, 2004.

[23] R. Raskar, H. Nii, B. Dedecker, Y. Hashimoto, J. Summet, D. Moore, Y. Zhao, J. Westhues, P. Dietz, J. Barnwell, and S. Nayar, "Prakash: lighting aware motion capture using photosensing markers and multiplexed illuminators," ACM Trans. Graphics, Vol. 26, No. 3, pp. 36, 2007.

[24] J. Kim, G. Han, IJ. Kim, H. Kim, and S.C. Ahn, "Long-Range Hand Gesture Interaction Based on Spatio-temporal Encoding," 5th Conf. Distributed, Ambient and Pervasive Interactions, p. 22-31, 2013.

[25] TI DLP Product. http://www.ti.com/lsds/ti/dlp/advanced-light-control/products.page

[26] Arduino MKR1000. https://www.arduino.cc/en/Main/ArduinoMKR1000

[27] Manchester code. https://en.wikipedia.org/wiki/Manchester_code

[28] S. M. Berman, D. S. Greehouse, I. L. Bailey, R. D. Clear, and T. W. Raasch, "Human electroretinogram responses to video displays, fluorescent lighting, and other high frequency sources," Optometry & Vision Science, Vol. 68, No. 8, pp. 645-662, 1991.

[29] M. Wright, and A. Freed, "Open sound control: A new protocol for communicating with sound synthesizers," Conf. Computer Music, Vol. 2013, No. 8, p. 10, 1997.