Internet of Things 6 (2019) 100044

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

Projected visible light for 3D finger tracking and device augmentation on everyday objects

Shang Ma[a], Qiong Liu[b], Mingming Fan[c], Phillip Sheu[a,*]

[a] Electrical Engineering and Computer Science, The Henry Samueli School of Engineering University of California, Irvine, CA 92697-2625
[b] FXPAL, 3174 Porter Drive, Palo Alto, California 94304, United States
[c] Department of Computer Science, University of Toronto, Toronto, ON, Canada, M5S 2E4

ARTICLE INFO

Article history:
Received 18 December 2018
Revised 15 February 2019
Accepted 15 February 2019
Available online 1 March 2019

Keywords:
Coded visible light
Finger tracking
Projector-based interaction
Object augmentation

ABSTRACT

Recent advances on the Internet of Things (IoT) lead to an explosion of physical objects being connected to the Internet. These objects sense, compute, interpret what is occurring within themselves and the world, and preferably interact with users. In this work, we present a visible light-enabled finger tracking technique allowing users to perform freestyle multi-touch gestures on everyday object's surface. By projecting encoded patterns onto an object's surface (e.g. paper, display, or table) through a projector, and localizing the user's fingers with light sensors, the proposed system offers users a richer interactive space than the device's existing interfaces. More importantly, results from our experiments indicate that this system can localize ten fingers simultaneously with an accuracy of 1.7 mm and an refresh rate of 84 Hz with only 31 ms delay on WiFi or 23 ms delay on serial communication, easily supporting multi-finger gesture interaction on everyday objects. We also develop two example applications to demonstrate possible scenarios. Finally, we conduct a preliminary exploration of 3D depth inference using the same setup and achieve 2.43 cm depth estimation accuracy.

1. Introduction

As a result of the ongoing popularity of the Internet of Things, more and more everyday objects are being transformed into smart objects that can interact with users and react to their environment. These objects enable novel interactive applications, being the building blocks for the IoT paradigm. However, with these benefits come new challenges [1]. More specifically, most of everyday objects have their own original functions other than interacting with users or their surroundings. This diversity leads to variation on their interfaces. Further, many existing IoT devices have been designed to fulfill a specific need (e.g. thermometer, activity tracking, etc.) and have their own customized user interfaces. For example, smart TVs are usually packed with "smart" remotes for users to control their TVs from a distance, and most tablets leverage a multi-touch screen to enable multi-finger gesture interaction. Meanwhile, many IoT devices are being packaged in a small format that a touchscreen cannot fit inside, meaning users have to use their mobile phones to control them, which lacks direct interaction. A few examples of this would be the Philips Hue [2] and the Nest Learning Thermostat [3], which are used to control lighting and comfort in the user's home using a smartphone application. Due to this diversity, users normally have to learn different interaction styles to operate different objects in their environment. For example, they need to

[*] Corresponding author.
E-mail addresses: shangm@uci.edu (S. Ma), liu@fxpal.com (Q. Liu), mfan@cs.toronto.edu (M. Fan), psheu@uci.edu (P. Sheu).

https://doi.org/10.1016/j.iot.2019.02.004
2542-6605/© 2019 Elsevier B.V. All rights reserved.

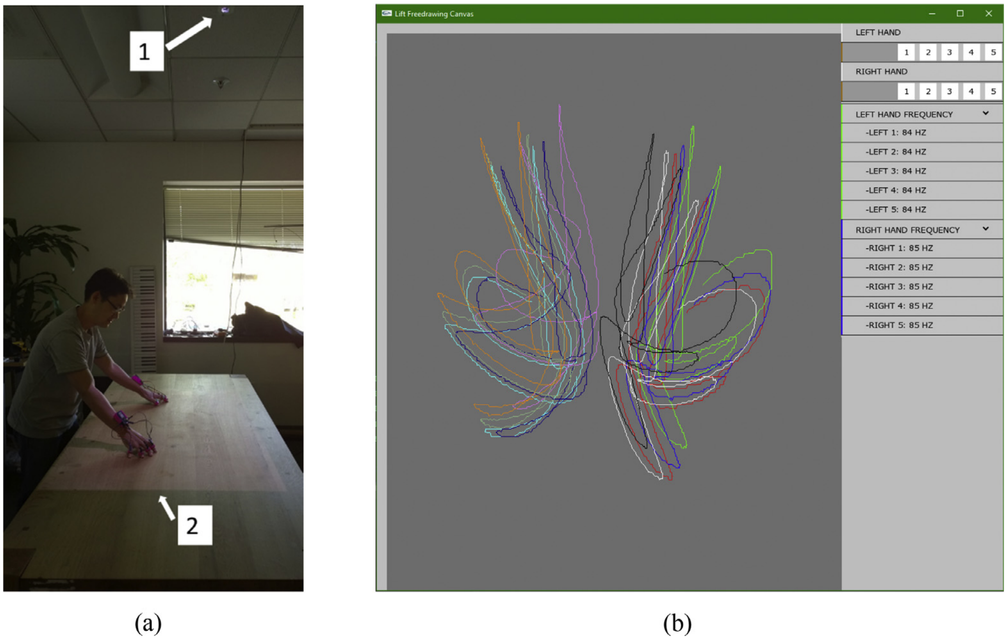

Fig. 1. (a) The user performs freestyle drawing on a regular office Table: (1) A projector behind the ceiling; (2) An office table as interaction space; (b) The traces of the user's fingers are recorded and color coded.

know how to interact with their smart microwave oven, how to manipulate icons on a tablet, and they also need to learn how to configure a smart thermometer using their mobile phones. Considering that a user's surroundings is populated by a huge amount of smart devices, this would cause a heavy cognitive load to learn and memorize how to interact with so many different objects [4].

In the meantime, the simplicity and intuitiveness of multi-touch gesture interaction on an instrumented surface makes itself well suited to be used with a variety of applications. The greater part of the laptops on the market contain inbuilt touchpads to provide a user-friendly gesture interface while desktop computers have products like Logitech K400 [5] and T650 [6] also available to permit multi-touch interaction beyond the conventional mouse and keyboard interfaces. These cases, along with the popularity of touch-screen enabled smartphones, demonstrate the usability and acceptability of such an interface among today's users.

Both the lack of multi-touch capability in many IoT objects/devices and the user's familiarity and preference on gesture interaction developments pose challenges to researchers and developers in designing appropriate interaction techniques for users to interact with these devices naturally and consistently regardless of whether they have touchscreens. A common solution to these challenges is to use universal and external controllers to communicate with these devices in a consistent way. For example, many IoT devices can often be controlled through the user's smartphone, which normally has a touch-enabled screen. Devices such as Logitech K400/T650 have also been very popular for desktop users who want to use multi-touch gestures on a regular desktop. Even though these devices seem to provide a universal solution, it presents drawbacks. First of all, the interaction between the user and the device is not direct. Users have to rely on a separate controller to interact with the device they want to control, meaning they will have to switch their attention between the controller (e.g. a touchpad) and the controlled device (e.g. a desktop). Secondly, these controllers may not be supported by many existing IoT/mobile devices. This is highly possible, since most of these controllers require dedicated operating systems and drivers, while many IoT devices are designed to perform specific tasks with limited computation resources.

Driven by these challenges, we present Lift, a visible light based finger tracking system that utilizes an off-the-shelf projector and light sensors to enable direct multi-touch gesture interaction on the surface of everyday objects. Fig. 1 presents a simple scenario where a user directly performs freestyle drawing on a regular office table using ten fingers. Lift is implemented by embedding location information into visible images and projecting these images onto a target object. Light sensors are then attached to both the object's surface and a user's fingers for object augmentation and gesture tracking. Thereby, Lift not only eliminates the requirement for a dedicated touchscreen on the object itself or even an external touchpad for gesture interaction, but it also enables users to have direct control on the object's surface, regardless of whether it is originally instrumented. More importantly, Lift features the richness of interaction space while remaining simplistic in physical size, hardware complexity, and computation load. Thus, Lift allows users to interaction with any object inside the projection area while remaining compact and easy-to-use. This is made possible by using a projection-based interaction method and encoding position data into each pixel of the projection.

For one thing, by projecting an interaction space into the environment, anywhere on the surface of an object can be allowed to accept a user's gesture input. This extends the original interfaces (e.g. buttons and switches on a device) to the whole body of the device, or even its surrounding space. Additionally, 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. Since all pixels in the projection area have been encoded with location information, the only remaining step required to restore the corresponding position is decoding a sequence of sensor readings. Therefore Lift can find and track light sensors without substantial computation and enable a fast response in finger tracking and gesture interaction for many applications. This advantage makes Lift easy to integrate into various existing mobile/IoT devices which normally have limited computation capability and sets it apart from all existing approaches, which often rely on complex recognition algorithms or require numerous computing resources to achieve a desirable effect. To the best knowledge of the authors, it is the first time that encoded visible light is utilized to build a fine-grained finger tracking system for augmenting everyday objects and IoT devices.

This paper extends our previously 2D finger tracking system [7] and is organized as follows: Related work in the literature is described in Section 2. Section 3 depicts technical details of the key components in our implementation. In Section 4, we present an experimental evaluation of the proposed system conducted in a real-world environment. This is followed by two example applications of the proposed system on everyday objects in Section 5. We then extend the proposed system to 3D space and examine this idea with one more experiment. Several limitations of the current system and future works will be explored in Section 7. Finally, we conclude our investigation in Section 8.

2. Related work

2.1. Finger tracking

Finger tracking has been investigated by a number of previous research projects. Kramer's glove [8], one of the first projects to track user's finger postures, functioned by attaching a strain gauge to a glove. Since then, novel finger tracking systems have been a research vision for decades and various types of technologies have been explored to build systems that can capture the movement of human fingers and identify the user's gestures based on collected data.

Visual tracking approaches, in which the camera is either the sole or the main sensor, are generally considered to be a dominating technique covering a wide field of applications [9–12]. The main disadvantages of these systems include the significant power consumption and being subject to various factors, such as lighting condition, the reflective properties of target surfaces, and camera resolution. These issues add a substantial amount of infrastructure, cost, and unpredictability to accomplish the desired effect and can be challenging to implement in practice.

Additionally, sensor-based systems provide a greater range of data and create decent finger tracking systems. Data glove [13], magnetic tracking [14], acoustics [15], wrist-worn camera and laser projector [16], muscle and tendon sensing [17], infrared proximity sensors [18], electric field [19], ultra-wideband signal [20], and ultrasound imaging [21], for instance, have all been explored in previous studies. Nonetheless, some of these systems operate in a moderately short range while others simply detect a small set of hand gestures. In comparison, Lift supports fine-grained and continuous finger tracking within a much larger space.

2.2. Encoded projection

A variety of applications, such as ambient intelligence [22], GUI adaptation and device control [23], projection calibration [24], and tracking movable surfaces/cars [25–27], have used encoded projection. These applications employ visible pattern projection through a Digital Light Processing (DLP) projector to transmit location information and restore position data by sensing light intensity via light sensors. This design 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, no previous technology has proven fast enough to track multiple rapidly-moving objects. Lee et al. [25] demonstrated that an update rate of 11.8 Hz was achieved when 10 patterns were used to resolve a 32 × 32 unit grid. Additionally, Summet et al. [27] projected 7.5 packets per second in their system, which is "just enough to track slow hand motions, approximately 12.8 cm/s when 1.5 m from the projector". On the other hand, Lift has achieved a tracking speed of 84 Hz in monitoring finger movement at the speed of 46.1 cm/s. Ma et al. [26] also increased the refresh rate to 80 Hz using an Android phone to decode received light signals. However, only one light sensor was decoded at a time, as opposed to our implementation, which has achieved an average update rate of 84 Hz while tracking ten light sensors simultaneously with two entry-level microcontrollers running at 48 MHz. This is a significant improvement over previous work. Research from [28] and [29] showed a higher system update rate, but both required a specially made beamer for one bit of spatial resolution whereas Lift only needs a single projector for all 1039,680 (1140 × 912) pixels, making Lift's implementation more practical. Goc et al. [30] leveraged a similar encoded projection system but the tracking speed and accuracy were obtained only through calculation, not through user studies or system evaluation. In contrast, we have conducted formal experiments to investigate how the proposed finger tracking scheme performs in the real-world scenarios. This provides empirical evaluation for future design of encoded projection based tracking system and demonstrates how everyday objects can be augmented to build interactive applications.

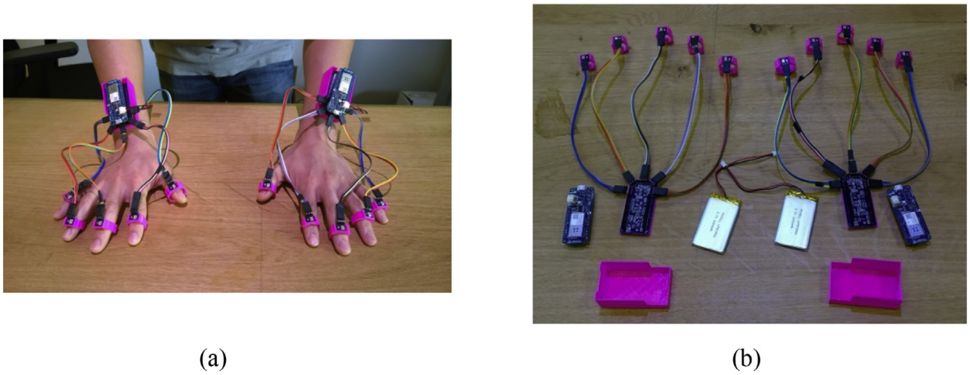

Fig. 2. (a) The user wears ten sensors on his fingers; (b) System components: light sensors on rings, Arduino boards, signal conditioning circuits, and batteries.

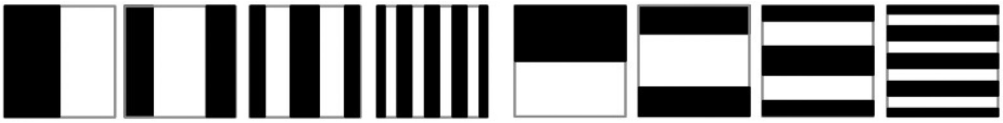

Fig. 3. An example of gray code images for 16 × 16 grid.

3. Prototype implementation

To assess our proposed tracking approach, we constructed a prototype platform in a typical office environment, as shown in Fig. 1(a). We then conducted a series of experiments to evaluate the performance of the proposed system. However, it is worth mentioning that Lift also applies to other surfaces, for example, mobile phones, paper, and whiteboards, on which users can perform gesture operations.

More specifically, our setup consists of four main components: (1) an off-the-shelf DLP projector from Texas Instruments [31] that can project gray-coded binary patterns onto a target surface with a projection area of 1380 mm × 840 mm from a distance of 1.9 meters; (2) light sensors attached to a user's ten fingers (see Fig. 2(a)); (3) two Arduino MKR1000 boards [32] with batteries, one for each hand; and (4) two custom printed circuit boards for signal conditioning (seen in Fig. 2(b)).

Gray-coded patterns are employed to encode each pixel in the projection and give each of these pixels a unique pair of coordinates so that when a light sensor detects the presence or absence of the projected light, a bit sequence representing the coordinates of a particular pixel can be received and decoded, thus recovering its original coordinates. Considering that projecting images into the environment is the inbuilt function of our projector, Lift does not depend upon any augmentation or extension on the projector itself. Moreover, data communication in Lift takes place in a unidirectional fashion (from the projector to light sensors), and position decoding is performed on the Arduino boards. As a result, Lift does not need central infrastructure for heavy computation or data transmission, allowing for a minimalist system design while still presenting a rich interaction space on different object surfaces. Finally, the two Arduino boards used as part of the current design contain WiFi communication modules, making Lift ready for many IoT applications.

3.1. Projector & encoded projection

As with other encoded projection systems, we rely on a DLP projector to overlay the required patterns onto a target object. Our current projector has a native resolution of 1140×912 pixels and a projection frequency of 4 kHz. Because gray-codes have a $O(\log_2 n)$ relationship between the number of required patterns and the total number of pixels inside the projection, a minimum of 21 $(\log_2 912 + \log_2 1140)$ gray-coded images are required to uniquely locate each pixel of the projection area. Fig. 3 demonstrates an example in which 4 horizontal (left four) and 4 vertical patterns (right four) are used to resolve a 16×16 unit grid.

3.2. Inbuilt Synchronization with High Reliability

In previous work [24,25,27], the absolute light intensity is employed to compute the bit value for each projection frame. However, this design is subject to ambient lighting, variance of the light sensors, and the analog-to-digital converter inside the microcontroller on which the decoding software is executed. Alternatively, Lift adopts a more robust encoding scheme. We use Manchester coding [33] to transmit the binary patterns. That is, each of our 21 gray-coded images is projected onto the target device followed by its reverse pattern. For example, when a '0' (black) is needed, we project the original 0 followed by its reverse value '1' (white). In this way, '0' is expressed by a low-to-high transition ('01'), while '1' is represented

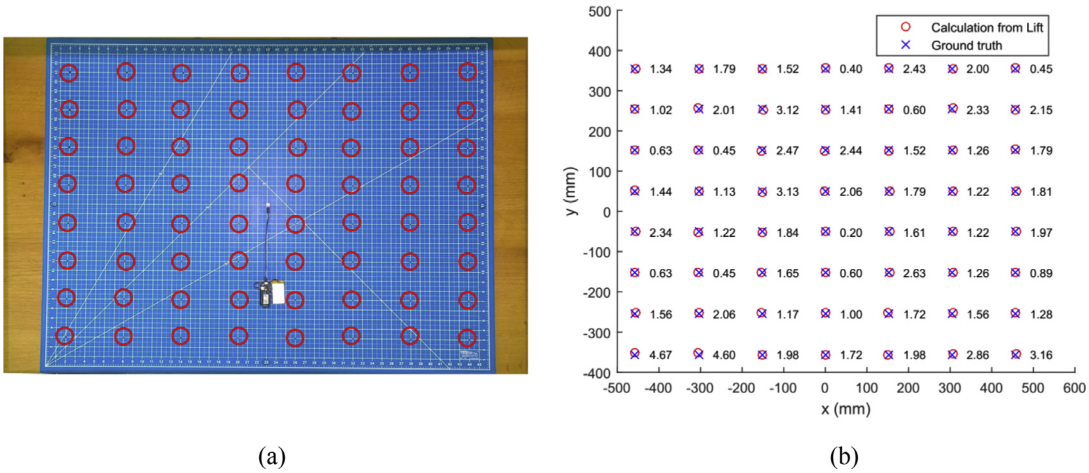

Fig. 4. (a) A test arena for computing the homography; (b) Localization error for 56 discrete points.

by a high-to-low transition (‘10’). This eliminates the dependency on the DC voltage of the received light intensity, which can be severely influenced by ambient lighting conditions and light sensor reception efficiency, and significantly improves the signal-to-noise ratio of the overall system. As such, Lift can be used in a variety of indoor environments. However, this completes the necessary number of patterns for location discovery from 21 to 42. Beside, we also add five framing bits (‘01110’) at the beginning of each packet bundle, allowing sensor units to be synchronized with the projector automatically. This further increases the necessary patterns to 47. Given that the exposure duration for a single frame is 250 μs for our projector, the total length of a packet is 11.75 (0.25 × 47) ms.

3.3. Projector-light sensor homography

Lift is devised to produce fine-grained physical coordinates of a user’s fingers on the surface of everyday objects for interactive IoT applications. In attempting to answer this challenge, another key component in our current implementation is projector-sensor calibration.

To obtain physical coordinates of a user’s fingers in the space, the position, orientation, and optical parameters of the projector relative to the target object need to be taken into account. For example, imagine that there exists a point (*x*’, *y*’) on the DMD (Digital Micromirror Device) mirror array inside the projector, and this particular point is projected to a point on a target object, for instance a flat table, with physical coordinates (*x*, *y*) (in mm). (An assumption we make here is that the origin on the table has already been defined.) The relationship between (*x*’, *y*’) and (*x*, *y*) is dictated by a planar projective transformation, which is also known as the homography between the pixel coordinates in the DMD array and the Euclidean coordinates in the physical space. Its parameters entirely rely upon the position and orientation of the projector with respect to the table, and the optical parameters of its internal lens.

In Lift, (*x*’, *y*’) coordinates are collected and decoded by a light sensor, whereas the (*x*, *y*) coordinates are measured relative to a user-defined origin. As a result, the transformation matrix can be computed using three key procedures: (1) identifying and marking a few points in the projection area (64 points in our current setup); (2) measuring the physical coordinates of these marked points relative to an origin on the target object, which, in our experiments, is the center of an office table as shown in Fig. 4(a); and (3) using a light sensor to collect the pixel coordinates of these points. Then the inbuilt *cv::findHomography* function [34] in OpenCV is used to find the homography, and the outcome can be applied to future sensor readings to identify the Euclidean coordinates of a projected point on the object. We have followed these steps to calculate the homography in our current setup and also evaluate the performance of this design in the next section.

4. Evaluation & results

Besides introducing the idea of encoded visible light for finger tracking on everyday objects, this work also aims to experimentally evaluate such technology in a realistic environment. The first research question that this paper would like to answer is: can Lift accomplish real-time high-resolution finger tracking on a non-instrumented surface? In this section, we assess several parameters of the proposed system: (1) tracking accuracy, (2) system delay, (3) system refresh rate, and (4) system reliability under various ambient lighting conditions.

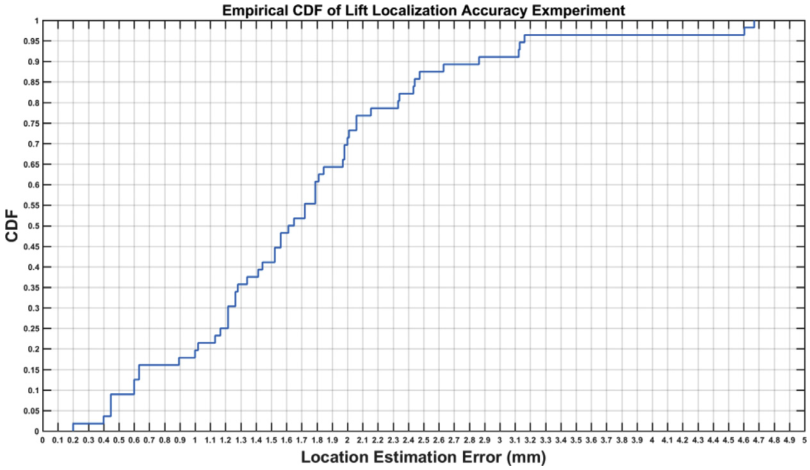

Fig. 5. The empirical cumulative distribution function of localization error for 56 discrete points inside the projection.

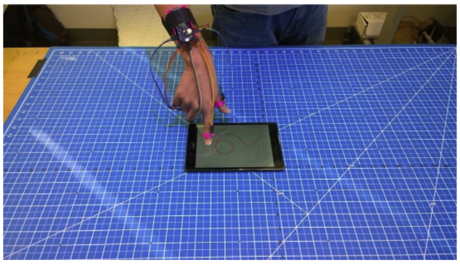

Fig. 6. Experiment setup to evaluate Lift's performance on continuous gesture tracking.

4.1. Localization/Tracking accuracy

4.1.1. Study design for discrete points

To assess localization accuracy of our proposed system for discrete points, a test arena was developed as shown in Fig. 4(a). In this setup, a DLP projector was installed behind the ceiling, and a flat office table was used as a target object (see Fig. 1). The distance from the projector to the table was 1.9 m, and the projection area on the table was 1380 mm × 840 mm. A 36 × 48 inch cutting mat was used to provide high-accuracy ground truth because of its convenience. However, because the cutting mat did not cover the whole projection range, we performed our experiment using points within the cutting mat area.

Fig. 4(a) demonstrates that totally 64 points (denoted by the black dots at the center of the red circles) were marked for calculating homography as previously described. They were uniformly distributed in the projection area. Pixel coordinates of these points were collected using a sensor unit that was placed at each of them, their physical distances with regards to the center were then measured, and the homography was calculated. The pixel locations of additional 56 points (not shown in the figure), which differed from the first 64 points, were also collected. The homography was then applied to these pixel coordinates. The physical locations of these 56 points were also measured as ground truth. We compute the difference of these two quantities as localization error.

4.1.2. Results

Fig. 4(b) outlines the results of localization accuracy calculated at these 56 points inside the projection. The deviation represents the Euclidean distance between the computed coordinates from Lift and the ground truth. The empirical cumulative distribution function (CDF) for this experiment is also illustrated in Fig. 5. For 90% of these points, Lift has achieved a localization deviation less than 2.875 mm. For all 56 points, the mean is 1.707 mm with a standard deviation of 0.918 mm. The maximum deviation is 4.669 mm and the minimum deviation is 0.2 mm. These results affirm Lift's goal of accomplishing millimeter-level finger localization in practice. From this figure, we can see that it is statistically significant to claim that location error is within 3.2 mm.

4.1.3. Study design for continuous gestures

The tracking accuracy for continuous gestures, as shown in Fig. 6, was also evaluated by dividing the projection space into 4 × 4 grids, each measuring 267 × 178 mm. We then placed an Android tablet (Google Nexus 9) at the central of

Table 1
Mean values of tracking error for continuous gestures at different locations.

x, *y* ranges (mm)	−534–267	−267–0	0–267	267–534
178–356	1.389	1.671	1.527	1.3
0–178	1.721	2.664	1.632	1.665
−178–0	1.032	1.599	1.664	1.677
−356–178	1.442	1.51	1.572	2.032

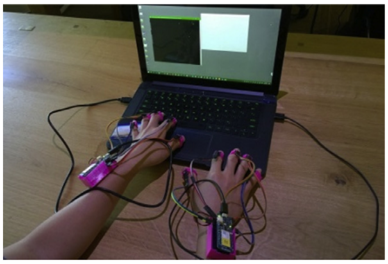

Fig. 7. Experiment setup for evaluating system latency.

each grid and requested our participants to draw freely on the tablet using only one finger. The tablet has a screen size of 180 × 134 mm and a screen resolution of 2048 × 1536 pixels. An Android program was created to acquire the pixel coordinates when participants are performing the drawing on the screen. 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 mm and further offsetting the result with the distance between the origin on the table and the origin on the tablet, which, in our design, was located at the top left corner of the tablet. The resulting locations were used as the ground truth in this experiment. Concurrently, the physical location of finger movements gathered by Lift was also computed based on sensor readings and the homography calculation.

Six participants were involved in this study, of which three are male and three female. No monetary compensation was provided for participation in the experiment. Every participant was asked to draw three traces in a grid, which translates to 192 trials for each participant and 1152 trials altogether.

4.1.4. Results

To calculate the tracking error for continuous gestures, we computed the average least perpendicular distance of each point collected by touch events on the tablet with the corresponding trajectory shaped by the outputs from Lift. This calculation was performed by averaging values across all trials inside each of 16 grids. Table 1 illustrates the mean error for each of them. The mean error for the entire projection area is 1.631 mm, and the standard deviation is 0.35 mm.

4.2. System latency

System latency significantly affects the total performance of a finger tracking system. For Lift, the latency comes from multiple levels. To begin with, the position packet through visible light channel itself is 11.75 ms. Secondly, the microcontrollers would add a delay when decoding the position and packing data in preparation for transmission. Finally, to make Lift portable and compatible with existing mobile/IoT devices, we chose WiFi for communication between Lift and target devices. Open Sound Control (OSC) [35] was used as the data transmission protocol because of its simplicity and robustness. This WiFi connection presented a major communication hurdle in our system; its bandwidth was shared between two Arduino boards in Lift and other devices in the testing environment, such as smart phones, tablets, desktops, and laptops. Thus, we designed the following experiment to measure the latency in this proposed system.

We used a simple setup to identify the delay between the time when Lift receives a command to initiate location data processing and the time when a laptop (2.2 GHz CPU, 8 GB memory) acquires the decoded position through the WiFi network and triggers the application of homography on the sensor readings, obtaining the final position. Fig. 7 illustrates our setup for this experiment.

A program was implemented to run on a laptop and record the timestamp of a keypress event from the space key on the keyboard. When the user presses the key, the corresponding timestamp is instantaneously recorded by the program. A notification is then sent to Lift via serial communication running at 115,200 baud, initiating the position detection process. Next, Lift will send the position data of all ten fingers to the same laptop through a WiFi connection once the data processing is complete. The time interval between the period when this program detects the keyboard event (the pressing of the space key) and when it receives the final finger positions from Lift and applied homography on them is referred to as the system latency. The six participants were invited to for this experiment where 1200 groups of time differences were collected. An

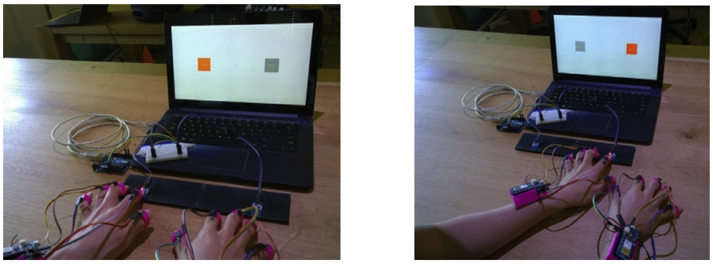

Fig. 8. Experiment setup for evaluating system refresh rate.

average latency of 31 ms was reported while using WiFi connection. When serial communication (115,200 baud) was used to transmit data from Lift to the laptop, the delay decreased to 23 ms.

4.3. System refresh rate

4.3.1. Study design

To ensure smooth finger tracking, the maximum speed that fingers can move in the projection space plays an important role in Lift. The six participants were invited for the following experiment purposely structured to identify the number of positions that Lift can decode at various movement speeds. Fig. 8 illustrates the setup of this experiment.

The participants were requested to wear 10 sensor units on their fingers and touch two touchpads (29 mm by 16 mm) in a predetermined order (left one first, then right one) at three speed modes: normal, medium, and as fast as possible. The time when these two touch sensors are activated was used as timestamps representing the beginning and ending points of a single test run. Knowing the time difference and the distance between the two test pads, it is easy to calculate the movement speed of the participant's finger. To increase the accuracy of our experiment, participants were asked to move their fingers in a straight line at the three aforementioned speeds. They were provided with a demonstration of the system first and allowed to practice with it. Once all participants were familiar with the procedure, we began the experiment.

In this experiment, the two pads were strategically placed at 20 cm apart, giving the participants enough space to move their fingers for a measurable period of time, but not too long that they had to extend or stretch their bodies to reach out to the second touchpad. The spacing was designed to allow maximum movement from a relatively stationary position, and the time interval between the two touch events was detected by an Arduino microcontroller and sent to a laptop via serial communication (115,200 baud) for data logging. A program running on the same laptop was used to simultaneously count the number of position packages Lift collects through WiFi network during the entirety of this time interval. Each speed case was repeated ten times. Participants could use any finger they liked for the touch action, but they had to use the same finger throughout a single test. Each participant was asked to perform 30 tests, amounting 180 tests in total for the six participants. For convenience and accuracy, participants were allowed to rest and change to another finger after completing a test run.

4.3.2. Results

The average moving time between these two touchpads was 955.504, 547.692, and 435.105 ms for the three speed modes respectively, as shown in Fig. 9. The same figure also outlines the average movement speed of user's fingers: 211, 368, and 461 mm/s. The average number of position packages received at these three different speeds was 84.365, 84.424, and 84.087, and the standard deviation was 0.179, 0.325, and 0.170, respectively. Each package contained the position data of all ten fingers. These results provide concrete proof that Lift can sustain a high refresh rate even when the users move their fingers at a relatively higher speed.

4.4. Lighting conditions

4.4.1. Study design

It is also important to test the robustness of the Lift system under different lighting conditions. We accomplished this by setting up a light sensor at a specified point inside the projection and logging its decoded coordinates at different indoor ambient light (Fig. 10). A light meter was placed next to the sensor unit to measure the light intensity. The primary metric utilized to quantify the system reliability was the percentage of correct readings.

4.4.2. Results

Table 2 below demonstrates the percentages of correct readings noted down at different lighting conditions. The first column represents the light intensity of the environment under different conditions measured in lux. Given that the current implementation of the Lift system is based on a visible light projector, the projector in the system will increase the light intensity of the projection area. The second column in the same table represents the light intensity of the projection space

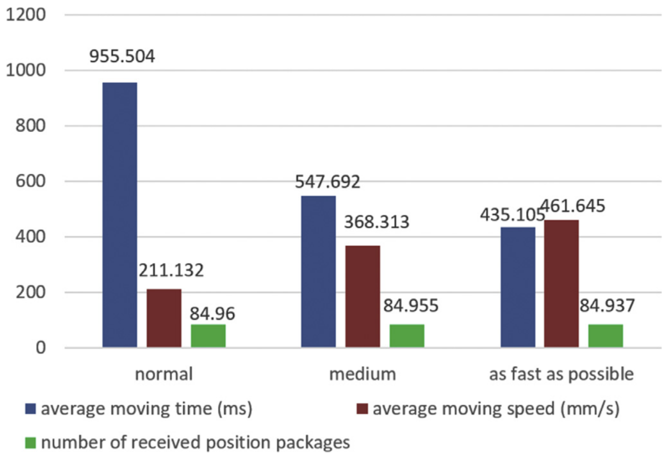

Fig. 9. Experiment setup for evaluating system refresh rate.

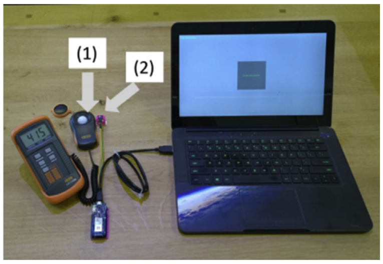

Fig. 10. Experiment setup for evaluating system reliability under different ambient light conditions: (1) a light meter; (2) a light sensor.

Table 2
Localization accuracy under different lighting conditions.

Ambient Light Intensity (lx)	Ambient Light + Lift Intensity (lx)	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

after Lift was turned on. The results illustrates that our tracking technique can operate reliably when ambient light is in the range of 0 to 345 lx. Ambient light level higher than 345 lx undermines the reliability of Lift's operation. Section 7 discusses how this problem can be solved to increase the robustness of Lift.

5. Example applications

To demonstrate how our proposed system can be used to augment everyday objects, we developed demo applications on two everyday objects with different form factors and input styles. In the first tablet-sized application, we developed a handy photo browsing album on a non-touchscreen laptop (Fig. 11). It allowed users to effortlessly browse through a collection of pictures on a non-instrumented display directly using multi-touch gestures. In the second application, we developed and installed a ten-finger freestyle drawing application (Fig. 1), which allowed a user to draw freeform traces on a common office table. Furthermore, we discuss how our proposed tracking technique can be used to allow users to interact with cultural heritage sites in VR/AR environments, and to assist older adults to monitor their physical activities in the home.

5.1. Gesture tracking on uninstrumented displays

A projector was utilized to project the patterns onto the flat surface of a table. The non-touchscreen laptop was placed inside the projection area and four light sensors were then placed at the four corners of the laptop's display to outline the

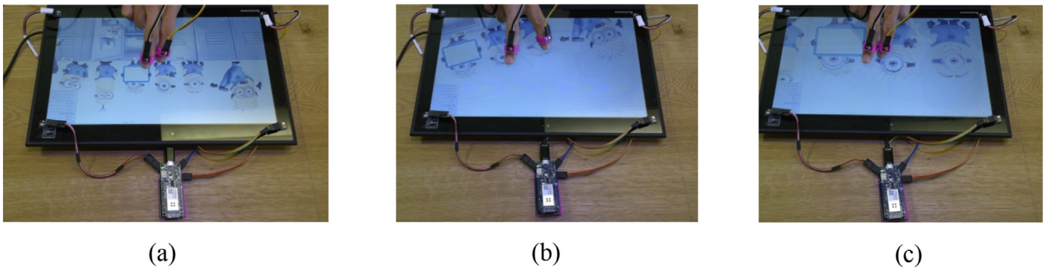

Fig. 11. With lift, a user can (a) pan an image, (b) rotate an image, and (c) zoom in/out an image on a non-instrumented display.

boundary of the target device (see Fig. 12). A participant was then requested to wear two sensor units on the index and middle finger, respectively.

As a proof of concept, we developed four common multi-touch gestures: *pan, zoom in/out*, and *rotating* that people regularly use on their touchscreen-enable devices. The participant had to move two fingers on the screen at exactly the same time, to trigger the pan gesture as shown in Fig. 12(a). As the fingers move to perform a gesture, their position data are simultaneously saved and decoded for transmission. The distance the data travels is utilized in dragging the image around. The rotation gesture perfectly conforms to its equivalent integrated in Mac or iPhone products, where users move two fingers around each other to initiate a rotating motion on an image (Fig. 12(b)). To zoom in/out of images, users must place two fingers on the screen and pinch them to either zoom in or out (Fig. 12(c)).

5.2. Freestyle drawing on uninstrumented tables

We have also implemented a freestyle drawing tool on a regular office table. With the help of encoded projection mechanism, we created an expansive interaction space of 1380 mm × 860 mm at a projection distance of 1.9 m (Fig. 1(a)). Note that it is also possible to extend the area of the projection to the size of, for example a wall or a whiteboard, by adding a wide angle lens in front of the projector. This holds great potential for future researchers exploring this technology.

In the freestyle drawing tool application, ten light sensors were attached to the participant's ten fingers. The participant was able to freely move his fingers in the interaction space at a relatively higher speed. The output signals of all the ten sensors were then collected and decoded on the two Arduino boards, one for each hand, and transmitted to a laptop computer to be visualized. A GUI application was used to display the color coded positions of the participant's ten fingers (Fig. 1(b)).

5.3. Enriching user experience when interacting cultural heritage sites in AR/VR

Experiencing cultural heritage sites of a group or a society can increase people's awareness of the history and culture of the group or the society and promote the protection of these cultural heritages. In recent years, the advancement in virtual and augmented reality technology has begun to offer people opportunities to experience the cultural heritage sites through head-mounted VR helmets or see-through AR devices. Although VR/AR displays allow users to view cultural heritage sites, interacting with the content in the VR/AR environment is still limited to simple hand or head gestures [40]. In contrast, the high-speed and high-accurate ten finger tracking capability of our system allows users to interact with the digital cultural heritage site rendered in VR/AR freely with ten fingers. Furthermore, our system also allows users to interact with the virtual content with the gestures proposed in previous research [41–43] and new gestures that need more fingers. In the future, it is worth exploring how to design such gesture sets that leverage more fingers but are also natural to users when they interact with digital content, such as expiring cultural heritage sites in VA/AR environments.

5.4. Assisting older adults in monitoring physical exercise in the home

Previous research on physical activity and health pointed out the importance of a physically active lifestyle in the prevention of chronic diseases and the promotion of health and well being [38]. Despite the importance of the physical and psychological benefits from regular activity, previous research shows that more than 60% of the adult population do not exercise regularly [39]. One way to help older adults adhere to their scheduled exercise plans is to monitor the amount of exercise that they have performed and how well they have performed the scheduled exercise advised by doctors.

With the high-resolution and high-speed ten-finger tracking technique, our proposed system can accurately track the real-time motion of an older adult's fingers and hands. With such information, we can quantify whether and for how long older adults are physically active and. Furthermore, by comparing older adults' hand and finger motion trajectories to the target motion trajectories in a particular type of exercise, we might be able to develop systems to assess how well older adults have performed the exercise and to provide feedback for older adults to reflect on their fingers, hands, and body motions. Furthermore, it is worth exploring visualization techniques to visualize the fine-grained fingers and hands motion

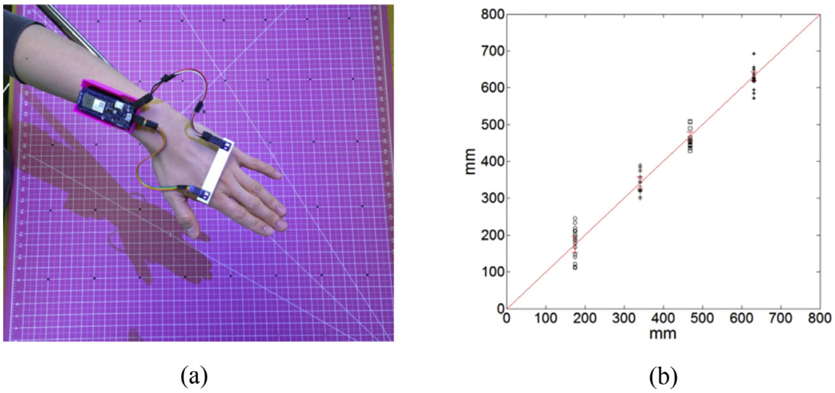

Fig. 12. (a) Two sensors on a single hand for depth estimation; (b) The 20 estimation trials for each height level.

data of older adults captured by our system to their doctors. This might allow the doctors to better monitor the ability of older adults' fingers and hands continuously and to better spot potential issues with their motor skills in order to prevent the onset of motor-related diseases.

6. Extend to 3D

As we have demonstrated, our implementation can achieve real-time high-resolution 2D tracking of ten fingers simultaneously on a non-instrumented surface. However, the system could be extended to provide 3D depth information thereby allowing more degrees of freedom and added potential for new application designs. Therefore, the next fundamental question we aim to answer in this work is: *can we obtain depth information of a user's hands using our current setup*? In other words, we would like to explore whether we can measure the distance between a user's hands and the table using only the DLP projector without any other tools, such as LIDARs, or time-of-flight cameras.

6.1. Depth inference

Lift achieves this goal based on two basic facts: First, the size of a projection area varies accordingly as the distance between the projection plane and the projector changes. This relationship is determined by the so-called throw ratio [36] and can be illustrated by the following equation:

$$t = \frac{D}{W} \tag{1}$$

where t is the throw ratio normally fixed for a particular projector, D is the distance from the projector to the projection plane, and W is the width of the resulting projection. The projector we used has a throw ratio of 1.4.

Further, the total number of pixels inside the projection is determined by the DMD chip of the projector, which is known as 1,039,680 (1140 × 912) pixels for our projector. Therefore, from these two quantities, we can represent the relationship between the width of a single pixel and the distance from the projector to, for example, a user's hands by the following equations:

$$\begin{aligned} D_{handToprojector} &= (PixelWidth \times 912) \times t \\ D_{handTotable} &= 1.9 - D_{handToprojector} \end{aligned} \tag{2}$$

from which we can see that if a user moves his hand closer to the table (*therefore away from the projector*), pixels that fall on his hand would become wider and vice versa. Eventually, we could infer the distance from the projector to a user's hand for a given pixel width, which in turn communicates the distance of the hand above the table, as shown in the Eq. (2). Note that here we also use the total distance between the table and the projector, which has already been measured in our previous experiments (1.9 m).

6.2. Preliminary evaluation & results

We conducted the following experiment to evaluate the proposed algorithm for computing the 3D depth of a user's hands: Two plastic plates were made with a 3D printer, each of which contains two light sensors situated 6 cm apart. One participant from previous experiments was recruited to wear these plates on the back of his hands so that all four sensors remained facing upward, as shown in Fig. 12(a). The two sensors on each hand were connected to the Arduino board as before, allowing them to obtain their locations with respect to the gray code coordinates in high resolution. Given that the physical distance between the two light sensors on the user's hand is fixed and they would detect two separate sets of

Table 3
Estimation error for 3D depth calculation.

Target depth (mm)	Mean error (cm)	Standard deviation (cm)
175	3.097	2.175
341	2.479	1.541
468	2.366	1.234
631	1.779	1.855

pixel coordinates from the coded visible light, we can calculate the width of a single pixel at any given distance of the hand above the table using this equation:

$$PixelWidth = \frac{60\,\text{millimeters} * \cos\left(\arctan\left(abs\left(\frac{y_2-y_1}{x_2-x_1}\right)\right)\right)}{abs(x_2-x_1)} \tag{3}$$

where (x_1, y_1) and (x_2, y_2) are two pairs of coordinates from the two sensors on the user's hand. To compute the distance between the hand and the table, we then apply the result from (3) to (2).

The participant was asked to hold up his hand at four different height levels where he felt comfortable. A central stand was used to help the user keep his hand still and parallel to the table (see Fig. 12(a)). The physical distance between the hand and the table was measured and used as ground truth. They were 175, 341, 468, and 631 mm for this particular user. In our experiment, 20 measurements were collected from Lift to calculate the depth using the above algorithm for each height level. The difference between the ground truth and the estimation from Lift demonstrates the depth estimation error, as shown in Fig. 12(b). The mean error and standard deviation across all 20 trials for each case is also illustrated in Table 3. The results show that Lift can achieve centimeter-level depth estimation using the proposed algorithm with 2.43 cm accuracy.

7. Contribution & future work

Our work makes two main contributions. To begin with, this is the first time that the encoded projection technique is exploited as a tracking mechanism to enable high-resolution multi-finger tracking and object augmentation on everyday objects. In the proposed system, wearable sensor units, signal processing firmware, desktop demonstration applications are designed and implemented to provide concrete proof of concept that supports the use of encoded projection based visible light communication in everyday environment. Additionally, we conducted an extensive set of experimental tests to provide detailed evaluation on the performance of the proposed system in terms of tracking accuracy, system latency, refresh rate, and system robustness. This provides empirical evaluation of encoded projection based location discovery scheme and therefore lays the foundations for future development of interactive smart objects and IoT applications in an everyday environment. Nevertheless, our study and experiments also revealed certain limitations and challenges of the current design.

7.1. Interference from ambient light

The system we developed projects binary patterns through visible light. To successfully decode position data, the light sensor we used has the best spectral response around 750 nm. This presents the problem of potential interference from ambient light in the setting. In Section 4.4, we tested and proved that Lift fails to work with strong ambient light. However, this is not a fundamental limitation of the proposed system since modulating the visible light with high frequency carriers or changing to infrared channel would significantly increase the signal-to-noise ratio allowing for finger tracking under challenging lighting conditions.

7.2. Power consumption

In our current implementation, the hardware system has a power consumption of about 0.5watts while decoding the received light signals at the frequency of 84 Hz. This includes the power consumption of the WiFi module for data transmission, which in practice consumes ~0.4 watts. Other low-powered transmission module [37] can be utilized to reduce this part to around 50 mW. Finally, the DLP projector itself still consumes 15 W. Although it can be connected to the grid without the worry of running out of power, more effort is needed to reduce its energy consumption.

7.3. High-resolution depth data

Although our exploration on depth inference is preliminary because we only use a single projector in our system, the results are promising. It is important to recognize that our types of sensing technologies, such as miniature infrared laser distance sensors and various inertial sensors, can be combined to provide high-resolution depth data and more degrees of freedom in the 3D space for interaction application design.

8. Conclusion

We introduce Lift, a visible light-based 3D finger tracking technique applied in an everyday environment. Encoded projection based visible light communication enables computation-efficient finger tracking with an extended interaction space on the surface of everyday objects, such as a common office table. To evaluate our design, a series of experiments were conducted to validate that Lift can track ten fingers simultaneously in 2D with high accuracy (1.7 mm), high refresh rate (84 Hz), and low latency (31 ms for WiFi, and 23 ms for serial cable) under various ambient light conditions. Finally, we further extend our design to obtain 3D depth information and achieve a depth resolution of 2.43 cm. With this work, we successfully demonstrate the feasibility of this new technique and present the first exploration of its rich design spaces for multi-finger tracking and object augmentation, which reveals that Lift holds significant promise for future research.

References

[1] UIST 2.0 Interviews – Dan Olsen. https://uist.acm.org/archive/uist2.0/DanOlsen.html.
[2] Philips Hue. http://www2.meethue.com/en-us/.
[3] Nest Learning Thermostat. https://nest.com/thermostat/meet-nest-thermostat/.
[4] Minimize Cognitive Load to Maximize Usability. https://www.nngroup.com/articles/minimize-cognitive-load/.
[5] Wireless Touch Keyboard K400. http://www.logitech.com/en-us/product/wireless-touch-keyboard-k400r.
[6] Wireless Rechargeable Touchpad T650. http://support.logitech.com/en_us/product/touchpad-t650.
[7] S. Ma, Q. Liu, C. Kim, P. Sheu, Lift: using projected coded light for finger tracking and device augmentation, in: Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications (PerCom),, IEEE, 2017, pp. 153–159.
[8] J.P. Kramer, P. Lindener, W.R. George, Communication system for deaf, deaf-blind, or non-vocal individuals using instrumented glove (1991). http://www.freepatentsonline.com/5047952.html.
[9] W. Hürst, C. Van Wezel, Gesture-based interaction via finger tracking for mobile augmented reality, Multimed. Tools Appl. 62 (1) (2013) 233–258.
[10] S. Henderson, S. Feiner, Opportunistic tangible user interfaces for augmented reality, IEEE Trans. Vis. Comput. Graph. 16 (1) (2010) 4–16.
[11] H. Koike, Y. Sato, Y. Kobayashi, Integrating paper and digital information on EnhancedDesk: a method for realtime finger tracking on an augmented desk system, ACM Trans. Comput.-Hum. Interact. 8 (4) (2001) 307–322.
[12] C. Harrison, H. Benko, A.D. Wilson, OmniTouch: wearable multitouch interaction everywhere, in: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, ACM, 2011, pp. 441–450.
[13] J.-H. Kim, N.D. Thang, Kim T-S 3-d hand motion tracking and gesture recognition using a data glove, in: Proceedings of the IEEE International Symposium on Industrial Electronics, 2009. ISIE 2009, IEEE, 2009, pp. 1013–1018.
[14] K.-Y. Chen, S.N. Patel, S. Keller, Finexus: tracking precise motions of multiple fingertips using magnetic sensing, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, ACM, 2016, pp. 1504–1514.
[15] A. Mujibiya, X. Cao, D.S. Tan, D. Morris, S.N. Patel, J. Rekimoto, The sound of touch: on-body touch and gesture sensing based on transdermal ultrasound propagation, in: Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces, ACM, 2013, pp. 189–198.
[16] D. Kim, O. Hilliges, S. Izadi, A.D. Butler, J. Chen, I. Oikonomidis, P. Olivier, Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor, in: Proceedings of the 25th annual ACM Symposium on User interface Software and Technology, ACM, 2012, pp. 167–176.
[17] T.S. Saponas, D.S. Tan, D. Morris, R. Balakrishnan, J. Turner, J.A. Landay, Enabling always-available input with muscle-computer interfaces, in: Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology, ACM, 2009, pp. 167–176.
[18] W. Kienzle, K. Hinckley, LightRing: always-available 2D input on any surface, in: Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology, ACM, 2014, pp. 157–160.
[19] M. Le Goc, S. Taylor, S. Izadi, C. Keskin, A low-cost transparent electric field sensor for 3D interaction on mobile devices, in: Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems, ACM, 2014, pp. 3167–3170.
[20] S. Wang, J. Song, J. Lien, I. Poupyrev, O. Hilliges, Interacting with soli: exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum, in: Proceedings of the 29th Annual Symposium on User Interface Software and Technology, ACM, 2016, pp. 851–860.
[21] M. Sagardia, K. Hertkorn, D.S. González, C. Castellini, Ultrapiano: a novel human-machine interface applied to virtual reality, in: Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014 2089–2089.
[22] Raskar R., Beardsley P., van Baar J., Wang Y., Dietz P., Lee J., Leigh D., Willwacher T. RFIG lamps: interacting with a self-describing world via photosensing wireless tags and projectors. In: Proceedings of the ACM Transactions on Graphics (TOG), 2004. vol 3. ACM, pp 406–415
[23] D. Schmidt, D. Molyneaux, X. Cao, PICOntrol: using a handheld projector for direct control of physical devices through visible light, in: Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, ACM, 2012, pp. 379–388.
[24] J.C. Lee, P.H. Dietz, D. Maynes-Aminzade, R. Raskar, S.E. Hudson, Automatic projector calibration with embedded light sensors, in: Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology, ACM, 2004, pp. 123–126.
[25] J.C. Lee, S.E. Hudson, J.W. Summet, P.H. Dietz, Moveable interactive projected displays using projector based tracking, in: Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology, ACM, 2005, pp. 63–72.
[26] S. Ma, Q. Liu, P. Sheu, On hearing your position through light for mobile robot indoor navigation, in: Proceedings of the 2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), IEEE, 2016, pp. 1–6.
[27] J. Summet, R. Sukthankar, Tracking locations of moving hand-held displays using projected light, Pervasive, Springer, 2005, pp. 37–46.
[28] R. Raskar, H. Nii, B. Dedecker, Y. Hashimoto, J. Summet, D. Moore, Y. Zhao, J. Westhues, P. Dietz, J. Barnwell, Prakash: lighting aware motion capture using photosensing markers and multiplexed illuminators, in: Proceedings of the ACM Transactions on Graphics (TOG), 3, ACM, 2007, p. 36.
[29] J. Kim, G. Han, I.-J. Kim, H. Kim, S.C. Ahn, Long-range hand gesture interaction based on spatio-temporal encoding, in: Proceedings of the International Conference on Distributed, Ambient, and Pervasive Interactions, Springer, 2013, pp. 22–31.
[30] M. Le Goc, L.H. Kim, A. Parsaei, J.-D. Fekete, P. Dragicevic, S. Follmer, Zooids: building blocks for swarm user interfaces, in: Proceedings of the 29th Annual Symposium on User Interface Software and Technology, ACM, 2016, pp. 97–109.
[31] TI DLP Product. http://www.ti.com/dlp-chip/advanced-light-control/products.html.
[32] Arduino MKR1000. https://store.arduino.cc/arduino-mkr1000.
[33] Manchester code. https://en.wikipedia.org/wiki/Manchester_code.
[34] OpenCV findHomography. https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html?highlight=findhomography#findhomography.
[35] M. Wright, A. Freed, Open soundcontrol: a new protocol for communicating with sound synthesizers, in: Proceedings of the ICMC, 1997.
[36] Throw ratio. https://en.wikipedia.org/wiki/Throw_(projector).
[37] BlueCreation BC118. https://www.digikey.com/product-detail/en/bluecre%20ation/BC118/1495-1003-1-ND/4860037.
[38] U.S. Department of Health and Human Services, Physical activity and health; A Report of the Surgeon General. www.cdc.gov/nccdphp/sgr/pdf/chap5.pdf, 1996.
[39] R.A. Hahn, S.M. Teutsch, R.B. Rothenberg, et al., Excessive deaths from nine chronic diseases on the United States, 1986 JAMA 264 (1990) 2654–2659.

[40] G. Caggianese, P. Neroni, L. Gallo, Natural interaction and wearable augmented reality for the enjoyment of the cultural heritage in outdoor conditions, in: Proceedings of the International Conference on Augmented and Virtual Reality, Springer, Cham, 2014, pp. 267–282.

[41] P. Mistry, P. Maes, SixthSense: a wearable gestural interface, in: Proceedings of the ACM SIGGRAPH ASIA 2009 Sketches, ACM, 2009, p. 11.

[42] J. Kim, J. He, K. Lyons, T. Starner, The gesture watch: a wireless contact-free gesture based wrist interface, in: Proceedings of the 2007 11th IEEE International Symposium on Wearable Computers, IEEE, 2007, pp. 15–22.

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