Duity: A Low-cost and Pervasive Finger-count Based Hand Gesture Recognition System for Low-literate and Novice Users

Tusher Chakraborty
Microsoft Research
t-tucha@microsoft.com

Md. Taksir Hasan Majumder
Bangladesh University of Engineering and Technology
mythoss1092019@gmail.com

Md. Nasim
Purdue University, West Lafayette, IN, US
mnasim@purdue.edu

A. B. M. Alim Al Islam
Bangladesh University of Engineering and Technology
alim_razi@cse.buet.ac.bd

ABSTRACT

In the context of developed countries, a considerable amount of research studies have been done to make Human-Computer Interaction (HCI) as natural and intuitive as possible for the public computing devices. Given the proliferation of public computing devices in the developing regions, a context-aware HCI design, focusing the low-literate and novice users from these regions, has the utmost importance. Taking the hygiene issue with the usage of touch-based public computing devices into account, in-air hand gestures are considered as one of the most natural forms of non-verbal interaction in this regard. In this paper, we present `Duity` – a low-cost finger-count based hand gesture recognition system – which reliably tackles the hurdle faced by novice and low-literacy populations. Exploiting the reflective property of visible light spectrum, we develop a mechanism that can reliably count fingers and thus identify different gestures. The system is fine-tuned to such an extent that varying lighting conditions and environmental setups can not perturb the accurate identification of the gestures. Our claims are supported by a user evaluation, which takes multiple scenarios into accounts, confirming the robustness, usability, and acceptability of `Duity`.

CCS CONCEPTS

• **Human-centered computing** → **Gestural input**;

KEYWORDS

In-air Gesture, Novice and Low-literate Users, HCI4D

ACM Reference Format:
Tusher Chakraborty, Md. Taksir Hasan Majumder, Md. Nasim, and A. B. M. Alim Al Islam. 2018. Duity: A Low-cost and Pervasive Finger-count Based Hand Gesture Recognition System for Low-literate and Novice Users. In *EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '18), November 5–7, 2018, New York, NY, USA*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3286978.3287005

MobiQuitous '18, November 5–7, 2018, New York, NY, USA

ACM ISBN 978-1-4503-6093-7/18/11...$15.00
https://doi.org/10.1145/3286978.3287005

1 INTRODUCTION

In recent years, a surge in public interactive kiosks has been observed in developing regions, such as India, Bangladesh, Vietnam, etc., [22, 27, 35, 45]. A pervasive computing device such as an interactive kiosk for queue management, bill payment, shopping, hospital registration, etc., is a burgeoning concept in these regions. Most of these interactive kiosks, irrespective of their purposes, come with a touchscreen-based interface. The downsides of availing touchscreen-based interface in the public spheres of a low-income region are twofold. First, the high population density of developing regions exacerbates the hygiene problems caused by these touch-based public interfaces [25, 41]. Second, these interactive kiosks demand relatively higher cost for large touch-based screens [17]. An intuitive solution to these problems is leveraging touchless input modalities. In this regard, hand gestures are one of the primary forms of human expression and communication and thus, is a reliable, natural, and intuitive means of human-computer interaction.

In the way of devising a hand gesture-based touchless input modality focusing the public spheres of the low-income regions, a noteworthy challenge is that literacy is the lowest in least developed countries [30, 45, 46]. Even among poor populations from these countries, the literate are typically novice users of smart-technologies [30, 45]. Hence, insufficient knowledge of this large population, both tacit and explicit, of the kind that is required to adapt and incorporate new technologies is one of the important challenges. In addition, as the most part of illiteracy correlates strongly with the socio-economic condition, low-literate and semi-literate users are very different from the target users of typical touchless input modality designs [11]. On top of this, it is highly required to minimize the cost-to-benefit ratio of a pervasive system considering the context of a low-income region, with the benefit being over a certain threshold.

Several different approaches have been proposed for detecting in-air hand gestures in state-of-the-art literature. All these approaches can be classified into two major categories: vision-based and sensor-based. Vision-based approaches demand using cameras, while sensor-based approaches mainly utilize ultrasonic, infrared, and proximity sensors [2, 24, 32]. Additionally, some state-of-the-art gesture recognition approaches have exploited RF signal [1, 23, 47]. Some of the sensor-based approaches demand usage of additional tracking devices such as wearable gloves, markers, bodysuit, etc., which hinders ease and naturalness of interaction [6]. Vision-based techniques, through involving the use of cameras, can remove inconveniences introduced by the tracking devices. Typical methods for such gesture-based menu selection applications use camera and motion sensing devices [5, 10, 38, 43]. Most of the aforementioned techniques are capable of recognizing a large set of gestures. Applications such as input interfaces for most of the interactive kiosks, queue management systems, etc., which entail navigators or number pads to operate, require recognition of only a *small set of gestures* [17]. In these cases, the large gesture set recognized by most of the aforementioned techniques becomes mostly redundant considering their demand for expensive high-end devices and computational complexity (e.g., image processing). Besides, camera might present privacy risks involving the leaking of sensitive images [40]. Above all, to the best of our knowledge, none of these approaches considered the above-mentioned challenges from low-resource settings context. In addition, considering the novice users of technology, our on-field study reveals that invisible wave-based technologies are less adaptable and acceptable among them.

In this paper, we attempt to address the aforementioned challenges and come up with a low-cost (below $5 including all costs) and user-friendly finger-count based hand gesture recognition system exploiting visible light. We named our system as `Duity`, a Bengali word meaning gleam of light. Here, we use light-sensitive versatile devices which is available at extremely *lower cost* (below $1) compared to the devices pertinent to existing sensor-based and vision-based approaches. More importantly, our on-field study reveals that the use of visible light inherently provides a perceivable guidance to the users, especially who are novice, which facilitates natural user interaction. Besides, visible light enhances the user interaction through enabling the users to get explicit feedback (e.g., LED blinking on the success of performed gestures) from the system. On top of this, we develop a simplistic finger-count mechanism avoiding high computational complexity yet ensuring the robustness, to be a pervasive system, under any environmental lighting condition. `Duity` can also be incorporated as an `add-on` with the existing touchscreen-based interactive kiosk. We summarize the set of contributions, which we make in this paper, as follows:

- We propose a low-cost and pervasive finger-count based in-air gesture recognition system, exploiting visible light, for the novice and low-literacy populations from low-income regions.
- We develop an algorithm for making the gesture recognition system robust against environmental lighting changes. Additionally, we ensure detection of undefined noisy gestures and eliminate them accordingly.
- We investigate the impact of visible light in making an in-air gesture-based system more interactive and usable compared to other invisible wave-based techniques.
- Finally, we perform user evaluation, to confirm efficiency, usability, and robustness of `Duity`.

2 DESIGN PRINCIPLE

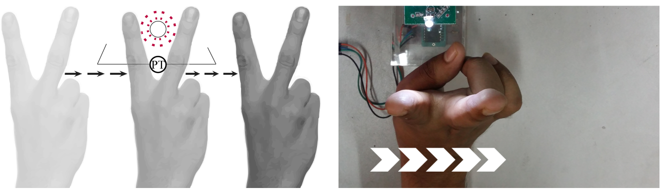
In this section, we delineate the design principles that drove us to the current design of `Duity`. Although our cardinal focus is on the input modalities here, the concrete affair between the UI and input modalities is also considered. In the way of devising a public interactive system for the people from low-income regions, the predominant focus group is always low-literate and novice users. Through the years, researchers have shown profound interest in devising design recommendations to make UI and input modalities more usable and adaptable for this focus group. All these studies emphasize on the simplicity of the UI and input modalities through avoiding complex navigation, menu scrolling, nonnumeric text input, soft-keys, etc., [21, 28, 30, 45]. Precisely in case of input modalities, user adaptation should be fast enough, which requires the input modalities to be natural, intuitive, and simple. Based on that, several studies highly recommend the use of numbers as the familiarity and speed with numeric input have been observed among the focus group [18, 30, 31, 34]. These design recommendations from state-of-the-art-literature gave us the insight of exploring finger-count menu. A finger-count menu [3, 4] is basically a two-handed technique where a multiple number of fingers are used for menu selection. Such free hand interaction does not require fingers to be distinguished, rather requires fingers to be counted. Finger-count menus have been found significantly faster and more natural than the other menu techniques [5, 26]. In addition, finger-count menu encourages ultra-simple navigation, no scrolling, fewer menus, and dedicated buttons, which are recommended by several studies for low-literate and novice users [21, 28, 30, 45].

Next, how to design a touchless finger-count technology for our above-mentioned focus group? Here, the major challenge lies in making the input system touchless precluding any haptic feedback, given that low-literate and novice users are highly comfortable with the tangibility of an input modality. Basically, in existing touch-based system, tangibility provides a perceivable (to be more specific, haptic) feedback to a user. This perceivable feedback is particularly essential considering the usability and acceptability of an input modality among the focus group. Now, the question arises, how to devise a touchless framework that inherently provides a perceivable feedback apart from the haptic sense? Here comes the visual perception offered by the visible light. Visible light ray is intangible, however, it still provides a perceivable feedback which is particularly essential for our focus group. For example, if a person is asked to wave the hand in front of a comparatively illuminated light source, the person naturally moves the hand through the illuminated region created by the light source through discerning the light rays incident upon the hand. This gave us the insight of exploring visible light in devising a touchless finger-count system. Next, we will delineate the working principle of exploiting visible light in devising `Duity`.

3 WORKING PRINCIPLE

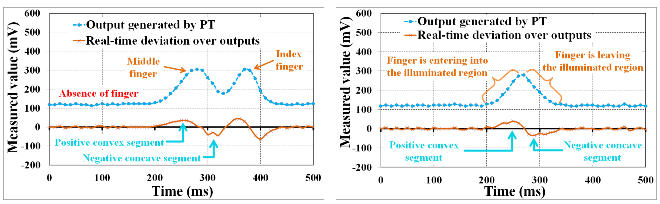
In this section, we show how we can identify the number of fingers exploiting the classical reflective properties of visible light. To do so, assume a pair of light source and light sensor, an LED and a phototransistor (PT) to be more specific, faced in the same direction. We move the finger(s) in front of the coupling for the purpose of finger counting. Fig. 1 shows such a placement and movement of two fingers.

Now, we move two fingers (index finger and middle finger) along a path that is *roughly* perpendicular to the direction of the coupling as shown in the Fig. 1. Fig. 2a shows the corresponding outputs of PT. In the PT response curve (i.e., the output generated by the PT), there are two convex segments and a concave segment between them, which as a whole represents two fingers and the gap between



(a) Front view (b) Top view

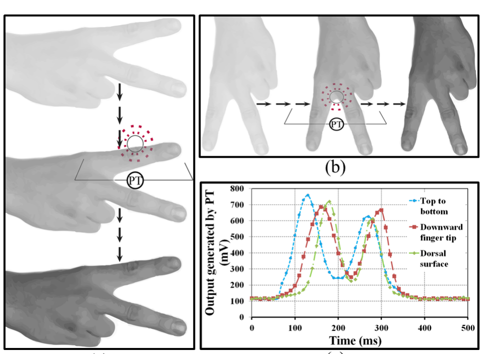
Figure 1: Movement of two fingers



(a) Movement of two fingers (b) Movement of one finger

Figure 2: Outputs of PT and real-time deviations in response to the movement of finger(s)

them. Similarly, in Fig. 2b, the convex segment of PT response curve represents a single finger, i.e., the index finger. While moving these two fingers from left to right in front of the coupling, at first, the middle finger enters into the region illuminated by the LED. When the middle finger starts entering into the illuminated region, the distance between the coupling and the middle finger decreases, and the illuminated area of the middle finger increases. Thus, we obtain a polynomial increment in the output generated by the PT. However, when the middle finger starts leaving the illuminated region we observe a decrement (polynomial in nature with respect to time) in the output. Now, when the index finger starts entering into the illuminated region, we again observe a increment (polynomial in nature) in the output generated by the PT. Thus, in Fig. 2a, a concave segment is created in between two convex segments, which represents the gap between two fingers. Here, the gap between two fingers determines the width of the concave segment. In these figures the flat segments represent the response of PT in absence of any finger in front of the coupling. Here, no significant real-time deviation[1] (absolute value < 10mV) over two consecutive outputs generated by the PT is observed.



(a) (c)

Figure 3: Different ways of movement and corresponding outputs

[1]Difference between two consecutive outputs generated by the PT is defined as the real-time deviation or first-order deviation.

Note that, it is not mandatory to move the finger(s) only from left to right, one may move the finger(s) along any path that is roughly perpendicular to the direction of the coupling such as vertically top to bottom movement, right to left movement keeping the fingertip downward, etc. Moreover, one may keep the dorsal surface of the hand in front of the PT instead of the palmar surface like the previous cases. In all these cases, the output curves exhibit similar pattern. Besides, the PT can perfectly sense the movement of the finger(s) within $\sim$ 30cm distance from it.

4 FINGER COUNTING MECHANISM

For pervasive applications, it is always recommended to design a solution with less computational complexity yet ensuring the service level over a certain standard. Complex computational mechanism demands high power consumption and often introduces hardware redundancy. Considering these issues, we attempt to design a simplistic finger counting mechanism exploiting the findings presented in the last section.

4.1 Basic Underlying Mechanism

To count the number of fingers in a simplistic manner, we exploit the real-time deviation over the outputs generated by the PT. Now, if we plot the real-time deviation over the outputs generated by the PT at the time of moving two fingers, we observe that the graph contains four notable curved segments- two convex and two concave (as shown in Fig. 2a). The first convex segment represents the entry of the first appearing, i.e., the middle finger into the illuminated region and the following supplementary concave segment represents its leaving. Next two convex and concave segments are created for the movement of next appearing, i.e., the index finger. Here, we can extrapolate that each finger creates two supplementary convex and concave segments (Fig. 2b). In a similar manner, movements of three, four, and five fingers introduce six, eight, and ten supplementary convex and concave segments respectively. To count the number of fingers in a reliable and simplistic manner, we exploit these supplementary convex and concave segments.

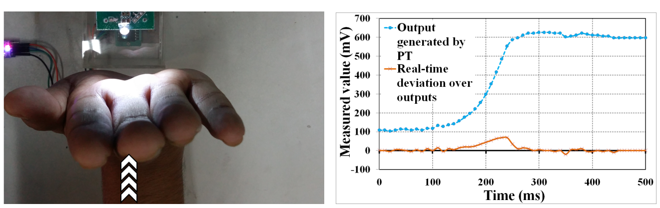
4.2 Algorithm

The algorithm collects the outputs generated by the PT at 10ms interval and stores four incremental data-points. Upon taking two successive readings for the dataset, our algorithm calculates the real-time deviation over these two readings. The algorithm then compares the *absolute value* of recently calculated real-time deviation against the *Idle deviation*, i.e., 10mV. The absolute real-time deviation lower than the *Idle deviation* is considered as an absence of any finger movement and reported as an *undetected situation*. The tiny short-lived spikes on real-time deviation curve represent such *undetected situations*. The algorithm always keeps track of the latest output of the PT in this situation. To do so, it keeps the current reading in and shifts the old reading out of the dataset. In this way, tiny short-lived spikes on the real-time deviation curve are identified and eliminated. Here, if the algorithm consecutively finds this *undetected situation* twice, it shifts all old readings out of the dataset and stores only the latest reading of the PT.

The algorithm always checks if the number of stored data-points is 4. If not, then the algorithm continues to take readings. Conversely, if the number of stored data-points is 4, then it is registered as a convex/concave segment on real-time deviation curve based on the positive/negative value of the deviations. Afterward, the algorithm does not store anymore reading in the dataset, rather observes the value of real-time deviation to identify when it crosses the horizontal axis. When the real-time deviation crosses the horizontal axis with an absolute value greater than the *Idle deviation*, the algorithm again starts to store readings in the dataset and continues the same aforementioned procedure. However, at some point in the observation period, if the algorithm finds *four* consecutive real-time deviations having the absolute values lower than the *Idle deviation*, then it is registered as the end of finger movement and the algorithm counts the number of fingers based on the number of consecutive supplementary convex and concave segments. After an interval of 200ms, the algorithm loops back to initial state. We confirm that the chosen values of the stated intervals, the *Idle deviation*, and the length of the dataset are suitable ones through experimentations involving different users and environmental lighting conditions. We skip detailing all these experimentations due to space limitation.

4.3 A New Gesture

According to the aforementioned discussion, movement of each finger introduces two supplementary convex and concave segments in the real-time deviation curve. Now, we intend to find a gesture that introduces only one convex segment in the real-time deviation curve. Unfortunately, it is nearly impossible to create this kind of gesture. When a hand/finger is moved closer to the coupling (creates a convex segment in the real-time deviation curve), the hand/finger has to be moved away from the coupling afterward (creates a concave segment on the real-time deviation curve), eventually always resulting in two supplementary convex and concave segments. However, a portion of our algorithm seemingly makes it possible.



(a) *Push* in real implementation **(b) Response of PT**

Figure 4: *Push* gesture and corresponding outputs

At one stage of our algorithm (the observation stage), we observe the values of real-time deviation rather storing them. This stage can lead to two possible subsequent stages. One of those two stages is the termination of finger movement, which is identified by measuring four consecutive real-time deviations having the absolute values lower than the *Idle deviation*. Now, if we make a gesture similar to 'pushing something' by moving the hand towards the coupling and then keeping it roughly steady (as shown in the Fig. 4a) for some moments ($\sim$ 60ms), our algorithm detects only one convex segment in the real-time deviation curve here. Next, after detecting the convex segment, the algorithm observes four consecutive real-time deviations having the absolute values lower than the *Idle deviation* (as shown in the Fig. 4b) at some point in the observation stage, and thus it is registered as the termination of movement ending up with the count of only one convex segment. Since the gesture exploits the natural action of pushing something, we name it as *push* gesture. Now, when should the user move the pushed hand back? For notifying the user to move the hand back, we take the advantage of visible light based interaction. Here, upon detecting the *push* gesture, LED starts blinking to notify the user to take the hand back.

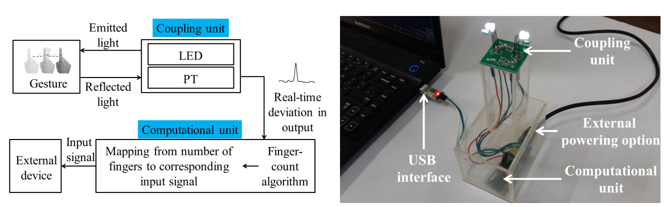
4.4 Robustness

A visible light based pervasive system needs to be capable of performing in diversified lighting conditions. In addition to the LEDs, the ambient light also illuminates the hand while performing a gesture. An abrupt change in the lighting condition (e.g., switching on/off a light) exhibits a key fact that this deviates the illuminance of light within a very short period of time. Such short-lived deviations introduce short-lived spikes in the real-time deviation curve and consequently eliminated by our proposed algorithm. This happens as our algorithm does not depend on only one sensed value, rather it depends on a set of values (four values spanning at least 40ms). Moreover, considering these scenarios, *spike detection* based signal processing is not directly a viable real-time option in our case.

It may happen that the PT is in a line of sight (LOS) with an external light source, which is highly-illuminated compared to LED. Here, while moving fingers through breaking the LOS between the PT and the external light source, the supplementary convex and concave segments on real-time deviation curve just swap their relative positions. To elaborate, since the PT is in an LOS with a highly-illuminated light source, the output generated by the PT decreases when the finger enters into the LOS region. Thus, a concave segment in the real-time deviation curve is first created. Conversely, when the finger starts to leave the LOS region, the output generated by PT increases. Since our algorithm does not consider the relative positions of supplementary convex and concave segments, the system does not get influenced by this scenario of having swapped supplementary convex and concave segments.

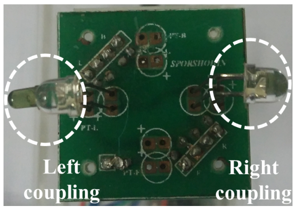
5 USER EVALUATION

We developed a custom-off-the-shelf touchless input device, named `Duity`, for interactive kiosk entailing the aforementioned visible light based finger-count methodology. Note that `Duity` can be adopted by any system which incorporates finger-count based menu selection interface. The pivotal purpose of developing this device is to study the usability, efficacy, and robustness of our proposed finger-count method. In this section, we present a user evaluation of `Duity` involving 30 users from two different low-income regions. For conducting user evaluation, we entailed two common applications of interactive kiosk: (1) interactive kiosk for shopping and (2) queue management system for banking. We connected `Duity` with a laptop, which was emulating the user interface of these two applications.

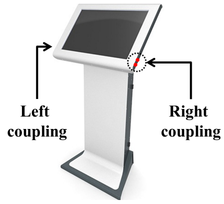


(a) Block diagram **(b) The device**

Figure 5: Simplified block diagram and snapshot of `Duity`



(a) In our prototype



(b) Off-the-shelf kiosk

Figure 6: Placements of Coupling unit in our prototype and off-the-shelf interactive kiosk

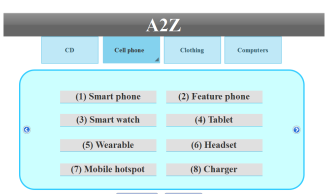
5.1 Design of Duity

Fig. 5a presents a simplified operational block diagram of `Duity`. Fig. 5b shows a snapshot of a `Duity` prototype. `Duity` consists of two main units: (1) Coupling unit and (2) Computational unit.

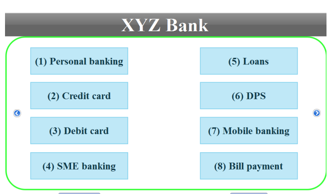
Coupling Unit: The Coupling unit consists of two couplings. Fig. 6 shows a snapshot of the Coupling unit. Here, two couplings are faced towards two opposite directions. Based on the placements, these two couplings are named as Right coupling and Left coupling. Each coupling is capable of recognizing aforementioned six gestures. Consequently, we can take $6 \times 2 = 12$ different inputs through `Duity`. Note that it is sufficient to subsume 12 gestures (10 fingers for counting and two more for special task) in order to operate a finger-count based menu selection interface [5, 26].

Note that it is not necessary to model the input device according to the design of our prototype. The design may change depending on the implementation context. For example, Fig. 6b shows a possible placement of the couplings in a real-world interactive kiosk. In addition, more couplings can be incorporated based on the application. Nevertheless, `Duity` can also be incorporated as an add-on input modality with the existing touch-based interactive kiosk. *In this case, we may persuade the users for utilizing touchless interaction through demonstrating awareness videos and graphical cues, which opens up a new dimension in persuasive and affective computing research.* However, in all the cases, the underlying method of finger-count remains same.

Computational Unit: A low-power-consuming microcontroller is incorporated for the computational purpose. This unit takes readings from each PT of the two couplings and invokes the finger-count algorithm. Upon recognizing the number of fingers or *push* gesture, the unit maps it to a corresponding input signal. Afterward, the unit feeds the input signal to a connected external device. Here, the input signal is fed to the external device through USB communication.



(a) Shopping interface



(b) Queue management

Figure 7: Interfaces for shopping and queue management

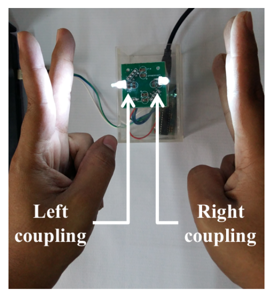
5.2 User Interface

We design two finger-count based menu selection user interfaces for two aforementioned applications of interactive kiosk emulated in a laptop. Fig. 7a presents a snapshot of the interface of an interactive kiosk for shopping purpose. Fig. 7b presents the user interface of the queue management system in a bank. Using this system, customers collect token for banking service. Each button of the interfaces contains a numerical prefix, enclosed in the round brackets, indicating the gesture to click the button. For example, '(1)' indicates the gesture of moving single finger in front of any coupling. We also separately designed the user interface considering the low-literate and novice users (See Fig. 8). In this particular interface, we enabled local language support, graphical cues, and voice annotation. We also provided graphical cues showing the number of fingers to select a particular tab (See Fig. 8b).

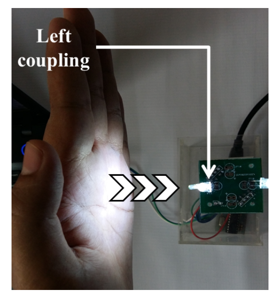
(a) Local language support and graphical cues

(b) Graphical cues for number of fingers

Figure 8: User interface for low-literate and novice users



(a) A possible way of performing '(5)' gesture



(b) Gesture to select the 'right-directed' arrow

Figure 9: Illustration of some gestures to operate the kiosk

The Left and Right couplings jointly count fingers, i.e., the sum of the number of fingers moved in front of two couplings is considered as a single input. For example, '(5)' indicates the gesture of simultaneously moving fingers in front of both couplings in such a way that the total number of fingers becomes 5. This can be done in several ways. For example, one can move three fingers in front of the Right coupling and two fingers in front of the Left coupling (Fig. 9a), or one can move all the five fingers in front of either Left or Right coupling. Note that it is not mandatory to maintain exact synchronism between moving fingers in front of both couplings. `Duity` can handle short manual asynchronism. This procedure enables counting up to ten fingers. Next, each arrow button in the interface is selected through performing a *push* gesture towards the corresponding direction. For example, the 'left-directed' and 'right-directed' arrow buttons used in the product browsing page (as shown in the Fig. 7a) are selected through performing *push* gestures in front of the Right and Left couplings respectively. Fig. 9b shows the *push* gesture performed to select the 'right-directed' arrow. The digit 0 is typed through moving all 10 fingers of two hands.

To make the device more *user interactive*, we added another feature taking the advantage of visible light. After every successful gesture detection, the Computational unit makes the corresponding

LED blinking, in front of which the gesture is performed. For example, upon detecting the movement of two fingers in front of the Left coupling, the LED of Left coupling blinks twice. Additionally, this feature enables us to observe from distance whether the gesture performed by the user is correctly recognized or not.

5.3 Experimental Setup

Our preliminary focus of the user evaluation was the learnability rather than the speed performance, because, finger-count based gesture, as well as the menu selection based on it, was a new concept to our users. Our next focus was to evaluate the efficacy of Duity in finger counting under diversified lighting conditions with the participation of our user group. In parallel, we evaluated the usability and adaptability of Duity.

Table 1: Demography of users

	Literate	Semi-literate	Low-literate
#Participants	10	10	10
Age range (years)	18 - 46	21 - 49	20 - 52
Gender	M: 5, F: 5	M: 6, F: 4	M: 4, F: 6
Hand size range (inch)	5.9 - 6.8	5.8 - 6.6	5.9 - 6.7
Palm size range (inch)	2.7 - 3.1	2.6 - 3	2.7 - 3
Training level	T: 3, ST: 3, UT: 4	T: 3, ST: 4, UT: 3	T: 4, ST: 3, UT: 3
Previous experience with interactive kiosk	5	1	0
Smartphone user	9	2	1
Active banking experience	10	9	7

5.3.1 Demography of users. To perform the user evaluation, we invited 30 participants with diversified literacy level, ages, genders, hand sizes [39], palm sizes [39], previous experience with interactive kiosk and smart-technology, and training levels. Table 1 presents a demography of the invited users. As the table suggests, we maintain the blend of diversified influential demographics – literacy level, ages, training level – in an *unbiased proportion*. Our low-literate users could write their names, read isolated words, do the basic counting, however, they did not have any formal education background. Our semi-literate users attained highest formal education up to the fifth grade of K-12 education system. Both of these two user groups have partial numeracy, and most of the users are the novice user of smart-technology. The adaptation of a new user interaction method may vary depending on the ages, and thus we invited users with diversified ages. Besides, we tried to make a combination of participants with and without having a previous experience of using a touch-based interactive kiosk, and we found the young group of participants more familiar with the touch-based interactive kiosk. In addition, a hand gesture recognition system should effectively perform irrespective of the hand size and palm size. We will delineate the training level criteria later.

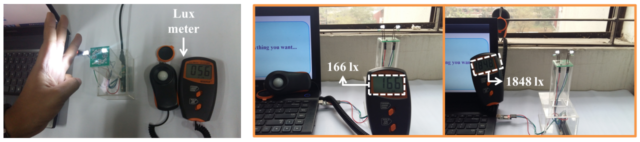
5.3.2 Introduction session with Duity. We first conducted an introductory session to make our users familiar with the complete system, i.e., the aforementioned finger-count menu interface and Duity. In this session, we initially presented a live demonstration to all users in order to make them familiar with the movements to be made and the whole operation of the emulated interactive kiosks. We then let *some selected* users directly interact with Duity. As the goal of this user evaluation was also to explore the experience of first-time users, we did not let all users interact with Duity in this introductory session. In this part, we conducted the session leveraging the classical think-aloud protocol. Think-aloud protocol enables the user to orally express their thoughts and opinions, more importantly, a specific problem which slows them down while exploring a system. We utilized paper and pencil method for protocol analysis.

In next part of the introductory session, we assigned the same selected user group some simple tasks such as selecting a button, changing the menu screen, etc. While attaining these tasks, we were standing around the user to guide them, and we directly intervened in case of any faulty move made by the user. Initially, users were carefully and slowly moving their fingers in front of the couplings. However, once they got the idea of moving the fingers through the illuminated region, they started to recover from the initial sluggishness. Another notable thing was the hesitation in making the combination of fingers. For example, in case of performing gesture '4', a user initially used to be confused with whether using one hand or both hands. Graphical cues showing the number of fingers associated with every selection tab (See Fig. 8b) exacerbated the situation. Most of the novice participants tried to mimic the finger combination shown on the display rather than the one that they feel comfortable. One of the participants said, *"I thought if I don't make the finger in the same way that is shown on the screen, it'd not work."* This introduced an unwanted delay in performing gesture. Here, the classical idea of using graphical cues to increase accessibility backfired. To eliminate any further confusion, we removed the graphical cues showing the number of fingers from our interface. As all our participants had partial numeracy, they were comfortable with the interface without this particular graphical cue (See Fig. 8a). Nevertheless, after performing the same gesture for several times, a user was able to figure out the own convenient combination. In most of the cases, users were comfortable in using their both hands instead of one.

The training level was measured based on the duration of gesture practice in the introductory session. For example, a trained user (T) performed gestures during the practice session for more than half an hour, and in case of a semi-trained user (ST), it was about ten minutes. On the other hand, an untrained user (UT) performed gestures directly in front of the device without any prior practice session, i.e., this group was not allowed to do any interaction with Duity in this introductory session.

5.3.3 Task and procedure. Our next focus was to evaluate the efficacy of Duity in gesture recognition and the gesture performing timing by the users. We provided each user a list of products and a dummy credit card PIN. The task of each user was to buy those products browsing the interactive kiosk emulated in the connected laptop and pay the bill through provided dummy credit card PIN. Note that we kept the whole session non-intrusive. We did not let them know that we were observing the timing, accuracy, etc., otherwise, it would engender a hurry-up sensation among them and spoil their spontaneous activities while interacting with Duity. To do so, in this session, we leveraged the video recording analysis method instead of paper and pencil analysis method. However, we kept a help-call option for the users – a user may ask for help or queries in the middle of a task completion.

5.3.4 Lighting setup. To study the robustness of Duity under different lighting conditions, we evaluated the accuracy of the device in



(a) Well-lit room (b) Asymmetric lighting condition

Figure 10: Different setups for user evaluation

three different lighting conditions. Moreover, during the experimentation, we generated different *pseudo accidental activities* such as an abrupt change in ambient lighting condition by switching the room light on/off, breaking the LOS between device and external light source by waving a hand, placing the device under highly illuminated light source, etc. Three lighting conditions that we considered are:

1 **Well-lit room:** This one is a common indoor lighting scenario at home and workplace. The illuminance in a well-lit room varies from 50lx to 250lx (as shown in the Fig. 10a). Here, the ambient light illuminance remains nearly uniform around the Coupling unit of the device.
2 **Asymmetric lighting condition:** Here, the environmental or ambient light illuminance varies around the Coupling unit. This lighting condition can be created by keeping the device nearby the window of a room when sunlight is present outside. Now, one side of the Coupling unit experiences high illuminance sunlight and the other side experiences relatively lower illuminance of room-light (as shown in the Fig. 10b).
3 **Daylight:** This lighting condition arises when the device is directly exposed to sunlight in an open outdoor space. Here, the light illuminance during the evaluation varied from 23000lx to 30000lx.

5.4 Result Analysis

To evaluate the performance of `Duity` and to compare it with existing touch-based and touchless interfaces, we subsume two metrics - accuracy and time taken to complete a task.

5.4.1 Accuracy and Timing. Each user performed on average 37 gestures under each lighting condition to complete the given task of shopping. Besides, we requested users to perform some non-specified random gestures in front of `Duity` after finishing the given task. Fig. 11 presents results found in different lighting conditions through confusion matrices. Here, a confusion matrix reveals mapping between the performed and recognized gestures. In an ideal case, the diagonal entries of the matrix should have the maximum values and other entries should have zero. All values of confusion matrices are shown in percentage. To further analyze the performance of `Duity`, we calculate classification accuracy. We define classification accuracy as the sum of correctly recognized gestures to the total number of gestures performed ratio. We observe that accuracy of `Duity` in recognizing gestures is 97.5% in well-lit room, 97.1% under asymmetric lighting condition, and 96.8% under daylight. Alongside, considering all the results as a whole, we find the overall accuracy of `Duity` in recognizing gesture to be 97.1%. Moreover, we found that `Duity` does not get influenced by any non-specified random gesture or any aforementioned pseudo accidental activity.

Table 2: Effect of training level on accuracy and timing

Training level	Average accuracy in % (SD)	Average task completion time in minutes (SD)
Untrained	97.1 (0.12)	3.12 (0.21)
Semi-trained	96.9 (0.2)	2.95 (0.13)
Trained	97.3 (0.17)	2.78 (0.06)

Table 2 shows the effect of training level on the accuracy and timing. As we mentioned earlier, a pervasive gesture-based system should be adaptable enough so that a user, without any prior experience, can make an accurate interaction in the first place. In case of `Duity`, Table 2 reveals that training level, i.e., prior interaction with `Duity`, does not have any influence on the accuracy in performing gesture. In addition, Table 2 suggests that users were making a faster movement with time. Note that, we tried to maintain these three sample sets, having different training levels, unbiased in terms of literacy level and age.

Table 3: Accuracy and timing with literacy level

Literacy level	Avg accuracy in % (SD)	Avg task completion time in minutes (SD)	#helps
Literate	97.5 (0.09)	2.11 (0.05)	2
Semi-literate (mostly novice users)	97 (0.21)	3.15 (0.13)	11
Low-literate (mostly novice users)	96.7 (0.66)	3.46 (0.21)	19

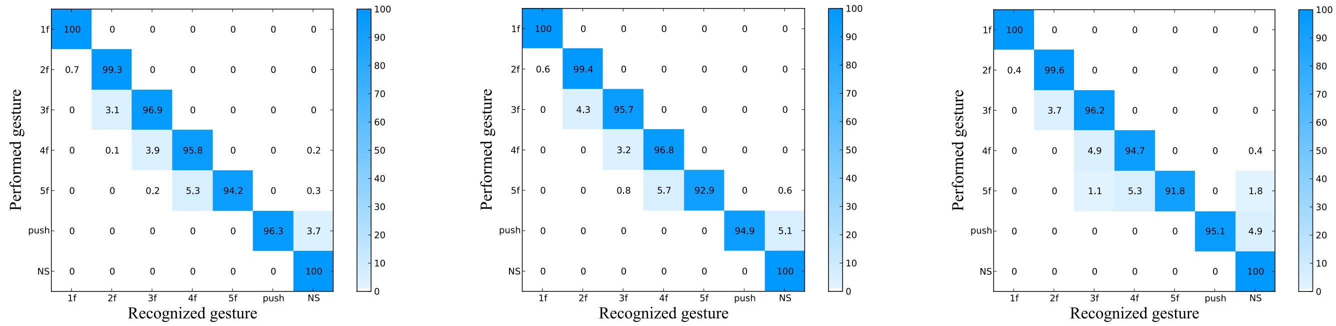
Table 3 shows the performance of users with different literacy level and experience with smart-technology. According to the table, there is no significant variation in gesture recognition accuracy with the literacy level and previous experience of using smart-technology. However, there is a noticeable contrast in case of timing. This was mostly because of the help-calls made by the semi-literate and low-literate users. In most of the cases, they were not understanding the options and menu hierarchy because of having less previous experience with the interactive kiosk or smart-technology (e.g., smartphone, smartwatch, etc.). Only two help-calls were made regarding the input modality, and those were basically regarding the confusion in making finger combination. As Table 1 presents, these three sample sets, having different literacy levels, subsumes an unbiased proportion of participants in terms of training level and age.

Table 4: Accuracy and timing with the variation of age

Age (years)	Average accuracy in % (SD)	Average task completion time in minutes (SD)	#helps
18 - 30	97.9 (0.22)	2.01 (0.12)	1
30 - 40	97.2 (0.23)	3.19 (0.57)	13
40+	96.1 (1.17)	3.63 (0.98)	18

Table 4 shows the performance of users with the variation in age range. Young group (18 - 30) has performed better compared to aged groups in terms of timing. Because, most of the members of young group had previous experience with interactive kiosk and smart-technology. Only one help-call was made by this group, and it was regarding the *push* gesture. However, aged groups comparatively made a significant number of help-calls, and we have discussed the rationale behind this earlier. However, there is no significant contrast in gesture recognition accuracy with different age ranges.

5.4.2 Comparison with off-the-shelf interfaces. To compare the performance of `Duity` with the existing touch-based and touchless interfaces of an interactive kiosk in terms of both accuracy and timing, we considered touchscreen, vision-based finger counting, and infrared (IR) proximity sensor-based finger counting. In our two focused low-income regions, most of the banks incorporate touch-based interactive kiosk for queue management. First, we collected



(a) Well-lit room **(b) Asymmetric lighting condition** **(c) Daylight**

Figure 11: Confusion matrix showing the accuracy of `Duity` under different lighting conditions. Here, $1f$ = 1 finger,..., $5f$ = 5 fingers, and NS = Non-specified gesture

Table 5: Comparison with existing interactive interfaces

Input modalities	Accuracy (%)			Timing (min)		
	Literate	Semi-literate	Low-literate	Literate	Semi-literate	Low-literate
Touch-screen	99.9	98.4	96.8	1.12	2.46	3.12
Vision-based	95.1	87.1	84.4	1.89	3.86	4.67
IR-based	88.3	85.1	81.9	2.07	4.14	5.02
`Duity`	97.6	97	96.9	1.34	2.14	2.68

the accuracy and time taken by our participants to complete a particular task using the touch-based interface of queue management system available in banks. Next, we requested each of them to perform the same task using our queue management interface. In the latter case, we incorporated three different input modalities - vision-based, IR sensor-based, and `Duity`. In case of vision-based input system, we tried to mimic the interactive camera setup described in [26] with our queue management interface. In case of IR sensor-based approach, we replaced each coupling of our device with an IR-based proximity sensor. Four of our users did not have any previous active banking experience, and the banking service provider of two more users had not incorporated a queue management system. Therefore, we conducted this session with the remaining 24 users. Since the banks usually maintain a well-lit indoor lighting, the evaluation of our system was also conducted in a well-lit room setup. Table 5 summarizes the results of this session. As the table suggests, the literate users, who had enough previous experience with smart-technology, attained nearly perfect accuracy and made expeditious interaction with the touchscreen-based system. Conversely, the semi-literate and low-literate users, who were mostly novice users of smart-technology, were comparatively unable to make an expeditious interaction due to the complex navigation, hierarchical menu, scrolling, etc. On top of this, they made faster interaction with `Duity` compared to the touch-based one due to the simplicity of finger-count menu interface. Besides, the accuracy attained with `Duity` is comparable with the touch-based interface irrespective of the literacy level and prior tech knowledge. However, most of the users, especially the novice users, *struggled with both vision-based and IR-based* input modalities. According to the users, although the menu interface (finger-count) was simple as before, they were basically facing trouble in making gestures in front of camera and IR sensor. They were not able to discern where and how to move or place hand, and this was particularly due to the absence of any perceivable feedback from those devices. In our video analysis, we also found the same on the basis of the think-aloud protocol. They were orally expressing their disquiet while interacting with camera and IR sensor.

5.5 Usability Analysis and User Feedback

To evaluate the usability of `Duity`, we first exploit classical System Usability Scale (SUS) and Single Ease Question (SEQ) scale. The average SUS score calculated for all participants is 88.1. In the SEQ scale of 7-point rating ('1' referred to 'very difficult' and '7' referred to 'very easy'), the average score is 5.9. In addition to these generic opinions, we let the users express our system specific feedback based on the 5-point Likert scale ('1' referred to 'strongly disagree' and '5' referred to 'strongly agree'). Table 6 shows the outline and the corresponding average scores of the questionnaire. To remove any potential bias, questions for SUS, SEQ, and Likert scale were dictated and graded by a third party in our absence. In addition to `Duity` specific feedback, users added some recommendations to improve the UI - adding animation, audible help message, etc.

During the user evaluation, we found aged people comparatively less likely in adopting this new technology, although they made quite good interaction with the device. The results of the Likert scale questionnaire also exhibit high interest of the young user group in using our system. Besides, in case of aged user group, there was an intelligible contrast between the time taken to complete a task using touch-based interface and our system. However, it does not imply that they are comfortable with touch-based interfaces. Four out of every five participants from the aged user group still feel comfortable and use feature phones instead of touch-screen based smartphones. This is a typical scenario in the context of a developing country.

6 DISCUSSION

In this section, we discuss the lessons that we have learned during our system design and study phases, and hopefully, these lessons will help the other researchers in designing a solution focusing towards the similar context. Additionally, we discuss the cost-effectiveness pertinent to `Duity`.

Visible light as a means of interaction: In a visible light-based hand gesture recognition system, light inherently adds an explicit guidance for a user to make a proper movement within the recognition region, which in return increase the accuracy in gesture recognition. One of our users opined, *"Once I used a proximity sensor based gesture operated device like you guys given me one today. But,*

Table 6: **Average scores of the questionnaire on 5-point Likert scale**

Statement	Literate	Semi-literate	Low-literate	Age (18 - 30 yr)	Age (30 - 40 yr)	40+ yr
The experience was enjoyable	5	5	5	5	5	5
Gestures were intuitive and easy	4.7	4.8	4.7	5	4.7	4.4
Operating the kiosk through finger-count was comfortable	4.6	4.5	4.7	4.8	4.5	4.5
You did not feel hurried or rushed while performing gestures	4.8	4.3	4.1	4.9	4.3	4
Visible light helped in performing gestures	4.9	4.9	5	5	5	4.8
Visible light has made the system more interactive	4.9	5	5	5	5	4.9
I will use this device in future	4.6	4.6	4.8	5	4.4	4.6
I prefer this device over touch-screen	4.3	4.2	4.3	4.9	4	3.9

in most of the cases, it didn't properly get my gestures. Because I was unable to get the region where to perform a gesture. But, in case of your device light made it easy. Here, I get the region by moving my hand through the light beam from LED." Another user added, *"It feels like we are touching the light."* Table 6 shows that all other participants agreed with this user. In addition, visible light can be utilized to convey a message to the user through blinking an LED, using different color combinations of an RGB LED, etc. One of our users suggested, *"That blinking of LEDs after gesture recognition is very interesting. By the way, you can use colorful lights to make it more interesting."*

Possible privacy risk with in-air gesture: We found that using in-air gesture poses a possible risk of visual tapping while inserting confidential information (e.g., PIN, password) in public spheres. One our users pointed out, *"I've found some security concern considering bill payment using this device. At the time of typing my PIN for bill payment, someone may tap my PIN by overlooking my gestures."*

Cost-effectiveness: The cost of our complete device is below $5 including the retail price of LED, PT, microcontroller, PCB, etc. In commercial production, materials are purchased in bulk amount, which reduces the cost per item of materials by at least 2 to 3 times than the retail price [14]. A typical breakdown shows that the material cost is 72% of the total product cost, which considers labor cost within the rest 28% [15]. Hence, the product cost including all other costs will still be less than $5 in commercial production. The main reason behind such a low-cost status is the absence of any expensive high-end device such as camera (*preferable* one is not available below $5), Kinect, depth sensor, etc. Besides, incorporation of these high-end devices demands complex computation (image processing) and comparatively heavyweight processor. Therefore, to the best of our knowledge, our system offers multiple times lower cost compared to the existing finger-count techniques.

7 RELATED WORK

Finger detection for diversified purposes such as number input, menu selection, etc., is a popular topic of interest to make interactions with computer interfaces more natural and easier [26]. Many contemporary touchless interaction systems [7, 29, 33, 44] are dependent on accurate finger or limb detection. Vision-based technology is prominent one for such detection.

In visual-based technology, skin color, shape, three-dimensional motion and three-dimensional anatomical models of hands, or in some cases, their various combinations are some of the most widely used features [12, 48] adopted in hand gesture detection. The most common method of tracking these features is to capture video using a camera, and then to detect if any gesture is being performed[6, 19]. Another choice for vision-based hand gesture detection is Kinect [16, 36, 37]. Its camera along with microphone and depth sensor can track the movement of a human's body parts. Using colored markers to keep track of different parts of hands is an alternative approach for gesture detection [8, 13]. Besides, few approaches use skin color and geometric analysis to isolate different hand regions. However, the sensitivity of such approaches to diversified operational backgrounds, more specifically in cluttered environments, can lead to inaccurate and faulty hand segmentation, resulting in a wrong detection.

Here, using cameras, as adopted in all these, for hand gesture detection is an expensive choice in case of input devices that demand *a small set of gestures* to operate. Moreover, these approaches demand a relatively heavy processing unit for their computational complexity. Additionally, if the privacy of the photos is not taken seriously, security problems may arise [20, 42], which is highly relevant for vision-based approaches to gesture detection using cameras. In sensor-based domain, Metzger et al., exploited infrared proximity sensor and a dual axis accelerometer for finger counting from one hand, which approximately costs more than $30 [32]. In our prior work, we exploited visible light to detect simple reciprocating hand movement and built a navigator for an interactive kiosk [9]. However, the predominant focus of that work was on cost-effectiveness and robustness of visible light-based system, not the user interaction by the novice and low-literacy users, which is also evident from the presence of complex menu navigation in our prior system. Note that, in that work, no specialized need related to finger counting was realized or discussed.

8 CONCLUSION

Considering the popularization of public computing devices in developing countries, approaches should be taken to make input modalities as cost-effective and natural as possible. Existing touchless technologies exploiting camera, radio frequency wave, IR, etc., are not reasonable choices, from the perspective of the novice and low-literacy populations from low-income regions. In this paper, we propose `Duity` – a novel low-cost finger-count based hand gesture recognition system exploiting visible light spectrum. We evaluate the performance of `Duity` through involving 30 users from two low-income countries. Overall 97.1% gesture recognition accuracy under different lighting conditions and satisfactory feedback from the user group confirm that `Duity` is natural, easy-to-use, and robust for pervasive purposes.

9 ACKNOWLEDGMENT

This work has been supported by Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh.

REFERENCES

[1] Fadel Adib, Zach Kabelac, Dina Katabi, and Robert C Miller. 2014. 3D tracking via body radio reflections. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. 317–329.

[2] Ahmad Akl and Shahrokh Valaee. 2010. Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing. In *Acoustics Speech and Signal Processing (ICASSP), IEEE International Conference on*. IEEE, 2270–2273.

[3] Gilles Bailly, Eric Lecolinet, and Yves Guiard. 2010. Finger-count & radial-stroke shortcuts: 2 techniques for augmenting linear menus on multi-touch surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 591–594.

[4] Gilles Bailly, JöRg MüLler, and Eric Lecolinet. 2012. Design and evaluation of finger-count interaction: Combining multitouch gestures and menus. *International Journal of Human-Computer Studies* 70, 10 (2012), 673–689.

[5] Gilles Bailly, Robert Walter, Jörg Müller, Tongyan Ning, and Eric Lecolinet. 2011. Comparing free hand menu techniques for distant displays using linear, marking and finger-count menus. In *IFIP Conference on Human-Computer Interaction*. Springer, 248–262.

[6] Kerry Bodine and Francine Gemperle. 2003. Effects of functionality on perceived comfort of wearables. In *International Symposium on Wearable Computers*. IEEE, 57.

[7] Marc Boillot. 2006. Method and system for touchless user interface control. US Patent App. 11/566,137.

[8] Lars Bretzner, Ivan Laptev, and Tony Lindeberg. 2002. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. IEEE, 423–428.

[9] Tusher Chakraborty, Md. Nasim, Sakib Md. Bin Malek, Md. Taksir Hasan Majumder Sami, Mohammed Samiul Saeef, and A. B. M. Alim Al Islam. 2016. Sporshohin: A Tale of Devising Visible Light Based Low-Cost Robust Touchless Input Device. In *Proceedings of the 7th Annual Symposium on Computing for Development (ACM DEV '16)*. ACM, New York, NY, USA, Article 1, 10 pages.

[10] Dustin B Chertoff, Ross W Byers, and Joseph J LaViola Jr. 2009. An exploration of menu techniques using a 3D game input device. In *Proceedings of the 4th International Conference on Foundations of Digital Games*. ACM, 256–262.

[11] Alan Cooper, Robert Reimann, and David Cronin. 2007. *About face 3: the essentials of interaction design*. John Wiley & Sons.

[12] Martin Côté, Pierre Payeur, and Gilles Comeau. 2006. Comparative study of adaptive segmentation techniques for gesture analysis in unconstrained environments. In *IEEE Int. Workshop on Imagining Systems and Techniques*. 28–33.

[13] Yuntao Cui and John J Weng. 1996. View-based hand segmentation and hand-sequence recognition with complex backgrounds. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, Vol. 3. IEEE, 617–621.

[14] The design trust. 2014. Cost price, trade price, wholesale price? http://goo.gl/XmdeX8. Accessed: Feb. 12, 2017.

[15] N. Dewhurst and D. Meeker. 2004. The True Cost of Oversea Manufacturing. http://goo.gl/nKNbuL. Accessed: Feb. 12, 2017.

[16] Jr Ding and Che-Wei Chang. 2015. An eigenspace-based method with a user adaptation scheme for human gesture recognition by using Kinect 3D data. *Applied Mathematical Modelling* 39, 19 (2015), 5769–5777.

[17] Khalid Djado, Claude Chapdelaine, Pierre-Luc St-Charles, and Maxime Derenne. 2014. Gesture interface for an interactive kiosk. In *Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*. ACM, 165–170.

[18] Kaushik Ghosh, Tapan S Parikh, and Apala Lahiri Chavan. 2003. Design considerations for a financial management system for rural, semi-literate users. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems*. ACM, 824–825.

[19] Elliott Greenfield. 2007. Touchless hand gesture device controller. US Patent App. 11/735,942.

[20] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. 2011. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 559–568.

[21] Matt Jones and Gary Marsden. 2006. Mobile interaction design. (2006).

[22] Journeyman. 2013. Kiosk Gallery. http://www.journeymanbd.com/?page_id=455. Accessed: June. 13, 2016.

[23] Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota. 2014. Bringing gesture recognition to all devices. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. 303–316.

[24] Jungsoo Kim, Jiasheng He, Kent Lyons, and Thad Starner. 2007. The gesture watch: A wireless contact-free gesture based wrist interface. In *Wearable Computers, 11th IEEE International Symposium on*. IEEE, 15–22.

[25] Mehmet Koroglu, Selami Gunal, Fatma Yildiz, Mehtap Savas, Ali Ozer, and Mustafa Altindis. 2015. Comparison of keypads and touch-screen mobile phones/devices as potential risk for microbial contamination. *The Journal of Infection in Developing Countries* 9, 12 (2015), 1308–1314.

[26] Arun Kulshreshth and Joseph J LaViola Jr. 2014. Exploring the usefulness of finger-based 3D gesture menu selection. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 1093–1102.

[27] V Lalith Kumar, AL Suseela, and Akashdeep Aharma. 2005. Taking Information Technology to the Heart of India. *Ubiquity* 2005, March (2005), 2–2.

[28] Sally Lehrman. 2007. Dialing in. Mobile phones target the world's nonreading poor. *Scientific American* 296, 5 (2007), 30–31.

[29] Shiguo Lian, Wei Hu, and Kai Wang. 2014. Automatic user state recognition for hand gesture based low-cost television control system. *IEEE Transactions on Consumer Electronics* 60, 1 (2014), 107–115.

[30] Indrani Medhi, Somani Patnaik, Emma Brunskill, SN Gautama, William Thies, and Kentaro Toyama. 2011. Designing mobile interfaces for novice and low-literacy users. *ACM Transactions on Computer-Human Interaction (TOCHI)* 18, 1 (2011), 2.

[31] Indrani Medhi, Aman Sagar, and Kentaro Toyama. 2006. Text-free user interfaces for illiterate and semi-literate users. In *Information and Communication Technologies and Development, 2006. ICTD'06. International Conference on*. IEEE, 72–82.

[32] Christian Metzger, Matt Anderson, and Thad Starner. 2004. Freedigiter: A contact-free device for gesture control. In *Wearable Computers, 2004. ISWC 2004. Eighth International Symposium on*, Vol. 1. IEEE, 18–21.

[33] Charilaos Papadopoulos, Daniel Sugarman, and Arie Kaufmant. 2012. Nunav3d: A touch-less, body-driven interface for 3d navigation. In *2012 IEEE Virtual Reality Workshops (VRW)*. IEEE, 67–68.

[34] Tapan Parikh, Kaushik Ghosh, and Apala Chavan. 2003. Design studies for a financial management system for micro-credit groups in rural India. In *ACM SIGCAPH Computers and the Physically Handicapped*. ACM, 15–22.

[35] Queue Pro. 2017. Queue management system in Bangladesh. http://www.queue-pro.com/news-events/. Accessed: Feb. 12, 2017.

[36] Kun Qian, Jie Niu, and Hong Yang. 2013. Developing a gesture based remote human-robot interaction system using kinect. *International Journal of Smart Home* 7, 4 (2013).

[37] Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. 2013. Robust part-based hand gesture recognition using kinect sensor. *IEEE transactions on multimedia* 15, 5 (2013), 1110–1120.

[38] Tao Ni Ryan, P McMahan, and Doug A Bowman. 2008. Tech-note: rapMenu: remote menu selection using freehand gestural input. In *3D User Interfaces, 2008. 3DUI 2008. IEEE Symposium on*. IEEE, 55–58.

[39] Choose Hand Safety. 2014. What is my hand size? http://www.choosehandsafety.org/choosing-hand-tools/hand-tool-size. Accessed: Sep. 26, 2016.

[40] Dragos Sbîrlea, Michael G Burke, Salvatore Guarnieri, Marco Pistoia, and Vivek Sarkar. 2013. Automatic detection of inter-application permission leaks in Android applications. *IBM Journal of Research and Development* 57, 6 (2013), 10–1.

[41] Heba Sayed Selim and Amani Farouk Abaza. 2015. Microbial contamination of mobile phones in a health care setting in Alexandria, Egypt. *GMS hygiene and infection control* 10 (2015).

[42] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. 2013. Real-time human pose recognition in parts from single depth images. *Commun. ACM* 56, 1 (2013), 116–124.

[43] Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. 2014. In-air Gestures Around Unmodified Mobile Devices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 319–329. https://doi.org/10.1145/2642918.2647373

[44] E. Stergiopoulou and N. Papamarkos. 2009. Hand Gesture Recognition Using a Neural Network Shape Fitting Technique. *Eng. Appl. Artif. Intell.* 22, 8 (Dec. 2009), 1141–1158.

[45] Indrani Medhi Thies et al. 2015. User interface design for low-literate and novice users: past, present and future. *Foundations and Trends® in Human–Computer Interaction* 8, 1 (2015), 1–72.

[46] UNICEF. 2016. Literacy among youth is rising, but young women lag behind. https://goo.gl/CcqkMP. Accessed: Jun. 17, 2017.

[47] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. 2016. 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 (UIST '16)*. ACM, New York, NY, USA, 851–860.

[48] Xenophon Zabulis, Haris Baltzakis, and Antonis Argyros. 2009. Vision-based hand gesture recognition for human-computer interaction. *The Universal Access Handbook. LEA* (2009), 34–1.