Gesture Recognition Using Visible Light on Mobile Devices

Zimo Liao, Zhicheng Luo, Qianyi Huang, *Member, IEEE*, Linfeng Zhang, Fan Wu, *Member, IEEE*, Qian Zhang, *Fellow, IEEE*, and Guihai Chen, *Fellow, IEEE*

***Abstract*— In-air gesture control extends a touch screen and enables contactless interaction, thus has become a popular research direction in the past few years. Prior work has implemented this functionality based on cameras, acoustic signals, and Wi-Fi via existing hardware on commercial devices. However, these methods have low user acceptance. Solutions based on cameras and acoustic signals raise privacy concerns, while WiFi-based solutions are vulnerable to background noise. As a result, these methods are not commercialized and recent flagship smartphones have implemented in-air gesture recognition by adding extra hardware on-board, such as mmWave radar and depth camera. The question is, can we support in-air gesture control on legacy devices without any hardware modifications? To answer this question, in this work, we propose *SMART*, an in-air gesture recognition system leveraging the screen and ambient light sensor (ALS), which are ordinary modalities on mobile devices. For the transmitter side, we design a screen display mechanism to embed spatial information and preserve the viewing experience; for the receiver side, we develop a framework to recognize gestures from low-quality ALS readings. We implement and evaluate *SMART* on both a tablet and several smartphones. Results show that *SMART* can recognize 9 types of frequently used in-air gestures with an average accuracy of 96.1%.**

***Index Terms*— Gesture recognition, visible light sensing, device-free, non-intrusive visible communication.**

I. Introduction

GESTURE control is a natural and user-friendly way to interact with devices. It extends the traditional keyboard/touch screen and provides users with great freedom. In home scenarios, smart TV can be directly controlled with gestures, instead of using a remote controller; when driving, the driver can adjust the volume of music using simple gestures, which is less distracting than controlling with touch screen or buttons. Besides, gesture control prevents our hands from physically touching the device which may carry the harmful virus. This is of vital importance for devices in public areas, such as self-service machines at the airport and vending machines at shopping malls. According to [1], the gesture recognition market is expected to grow at a compound annual rate of 27.0%, from 9.8 billion USD in 2020 to 32.3 billion in 2025.

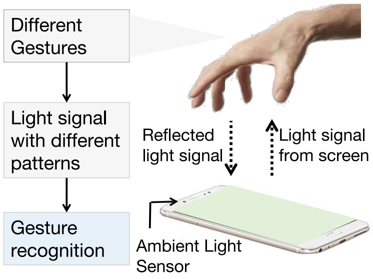


Fig. 1. "Screen-Hand-ALS" light path. Light from screen is reflected by the hovering hand, and the ALS can sense the intensity of the reflected light. We analyze the received light signal and recognize different gesture.

Although prior works have implemented gesture recognition via hardware on commercial devices like camera [2], microphone [3], [4], [5], and Wi-Fi radio [6], [7], none of them is commercialized on mobile devices. Solutions based on cameras and microphones raise privacy concerns, resulting in low user acceptance. Wi-Fi signals have low spatial resolution and thus Wi-Fi-based solutions are sensitive to background noise, such as people/object movement in users' surroundings. Furthermore, solutions based on Wi-Fi mainly rely on specialized NIC models (e.g., Intel 5300) and thus lack generality. Recently, several flagship smartphones are released on the market and they are equipped with specialized hardware to support in-air gesture recognition. For example, Google Pixel 4 [8] relies on Soli [9], a 60GHz mmWave radar, to sense human gestures in the air; Huawei Mate 30 Pro [10] supports the similar functionality, but it relies on an extra depth camera on the front panel; similarly, LG Q8 ThinQ [11] has a ToF camera on-board to support in-air gesture recognition. We ask the question that, can we support gesture recognition on legacy devices without any hardware modification?

We observe that we can leverage the "Screen-Hand-ALS (Ambient Light Sensor)" light path to recognize hand gestures, as shown in Figure 1. When a user is performing hand gestures over the screen, the light signal transmitted from the screen is reflected by hand to the ALS on the mobile phone. The

Manuscript received 19 July 2023; revised 29 December 2023; accepted 7 February 2024; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. P. Jue. This work was supported in part by the National Key Research and Development Program of China under Grant 2022ZD0119100 and in part by China NSF under Grant 62025204 and Grant 62002150. This paper is an extended version of the work that first appeared with title "SMART: Screen-based Gesture Recognition on Commodity Mobile Devices" at ACM MobiCom 2021 [DOI: 10.1145/3447993.3483243]. *(Corresponding author: Qianyi Huang.)*

Zimo Liao, Fan Wu, and Guihai Chen are with the School of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zimo_liao@sjtu.edu.cn; fwu@cs.sjtu.edu.cn; gchen@cs.sjtu.edu.cn).

Zhicheng Luo and Qianyi Huang are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China (e-mail: luozhch23@mail2.sysu.edu.cn; huangqy89@mail.sysu.edu.cn).

Linfeng Zhang is with the Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China (e-mail: zhang-lf19@mails.tsinghua.edu.cn).

Qian Zhang is with the Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong (e-mail: qianzh@cse.ust.hk).

Digital Object Identifier 10.1109/TNET.2024.3369996

amplitude of the reflected light signal received by ALS is relative to the position of the user's hand. Thus, it is possible to infer the hand gesture through analyzing the time-series of ALS readings. Screen and ALS are both ordinary modalities on mobile devices. The ALS is widely deployed on mobile devices (e.g., mobile phones, tablets, and smartwatches [12], [13], [14], [15]), which can sense the ambient light intensity and then adjust the brightness of the screen accordingly. Thus, the solution is compatible with commercial-off-the-shelf mobile devices. Different from camera, ALS can only gain the intensity of ambient light, which contains little sensitive information.

Although the idea sounds straightforward, we are faced with three challenges. The first challenge is to embed spatial information into the "Screen-Hand-ALS" light path so that we can recognize the gesture direction. Since most mobile devices have only one ALS, which is a one-pixel sensor, the receiver has a low spatial resolution. To overcome the limitation of the receiver, the light emitted from the screen must provide as much spatial information as possible. To address this challenge, we model the "Screen-Hand-ALS" channel using lambert's cosine law [16] and conduct experiment to validate the theoretical model. We study the influence of the flickering block's position on the light intensity received by a light sensor. Based on the model, we arrange the position of blocks on the screen.

The second challenge is to hide spatial light signals into screen contents and to preserve the viewing experience. To hide the spatial light into the original screen content, we change the original frame into a pair of switching, complementary frames. To overcome the limitation of screen refresh rate, we use the screen's line-by-line refreshing scheme to generate high-frequency signals, which are invisible to human eyes. Besides, we propose the color decomposition algorithm to select the RGB value of each frame's pixel according to the chromatic additive rule and flicker fusion rule [17] to ensure the visual effect after frame fusing is nearly the same as the original frame.

The third challenge is to recognize gestures from low-quality ALS data. Since the power of the light signal is restricted by screen brightness, and the diffuse reflection on the user's hand can cause large signal loss, the signal received by ALS is weak. Furthermore, the sampling rate of ALS is even lower than the flickering frequency of the line-by-line refreshing scheme, which causes the under-sampling problem. It is challenging to extract effective features from the low-SNR ALS readings. Based on our analysis, we address this challenge through a signal segmentation and feature selection mechanism. We design a segmentation algorithm, which can extract the high-SNR signal part and omit possible low power intervals. After that, we carefully choose effective features for classification.

In this paper, we propose *SMART*, which leverages the screen on mobile devices for air gesture recognition. We design the screen update mechanism (the transmitter side) and the gesture recognition framework (the receiver side). We implement and evaluate *SMART* on commercial mobile devices, including a tablet and several smartphones. We evaluate the gesture recognition accuracy, human perception, and processing latency. We also compare *SMART* with depth camera based approach. Key findings are as follows:

- Recognition Accuracy: We test *SMART* under 5 different lighting conditions with 14 users. *SMART* can recognize 9 types of frequently used in-air gestures with an average recognition accuracy of 96.1%.
- Subjective Viewing Experience: We invite 15 volunteers to evaluate the color difference, flickering effect and visual fatigue. We conclude that the design of *SMART* transmitter greatly relieves the flickering effect and visual fatigue and the viewing color is quite close to the original display.
- Processing Latency: We implement both transmitter and receiver of *SMART* on different mobile devices. We find that the time to process each frame is shorter than the frame-to-frame interval for 60 FPS (frame per second) display, and the time of signal preprocessing and gesture recognition is below 5ms, which verifies that *SMART* can run in real time on these commodity mobile devices.
- Comparison with depth camera: We compare *SMART* with the gesture recognition functionality of Huawei Mate 30 Pro. Results show that *SMART* has comparable gesture recognition performance with depth camera but lower power consumption.

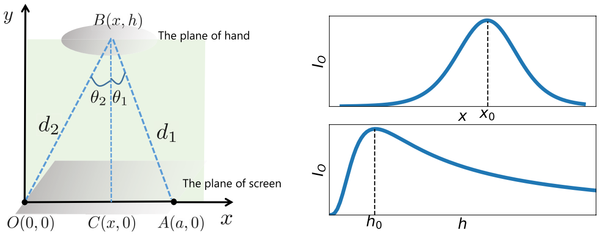
We highlight our main contributions as follows:

- We propose a new design paradigm based on screen and ALS for gesture recognition on mobile devices. Using a screen as the transmitter, we design a frame switching mechanism to embed spatial information into the original screen content. We also develop a gesture recognition framework based on specific features extracted from the light intensity data collected by ALS.
- We model the "Screen-Hand-ALS" channel to explore the theoretical relationship between the received light power and hand gesture. The model guides us to determine the position of the flickering blocks on the screen.
- We implement and evaluate *SMART*. Evaluation results demonstrate that *SMART* can recognize nine types of frequently used in-air gestures with an average accuracy of 96.1%. We expect that *SMART* provides the legacy device with the same in-air gesture control capability as the expensive flagship smartphones.

II. Modeling "Screen-Hand-ALS" Channel

In this section, we first model the "screen-hand-ALS" channel thoretically, where light emitted by the screen is reflected by the hand and then the reflected light goes to the ALS. This is the fundamental working principle under *SMART*. We also conduct experiments to validate the model.

To achieve gesture recognition based on the "Screen-Hand-ALS" light path, we want to know the relationship between the received light power and hand gesture. *SMART* uses a model to estimate the intensity of reflected light when hand is placed at a specific position over the screen. As the screen is a combination of discrete light sources (i.e., a large number of



(a) Geometric model (b) Mathematical result

Fig. 2. "Screen-Hand-ALS" light path modeling. (a) The geometric illustration of "Screen-Hand-ALS" model. 'A' denotes a point light source (a pixel); 'B' denotes a reflection point on hand; 'O' denotes the ALS; 'C' is the projection of 'B' on the screen plane. (b) The mathematical relationship between the intensity of reflected light (I_O) and hand's position (x and h).

pixels) and the shape of hand is complicated, the mathematical model of the light propagation and reflection procedure in the system is intractable. Thus, we build a simplified model to illustrate the basic mechanisms.

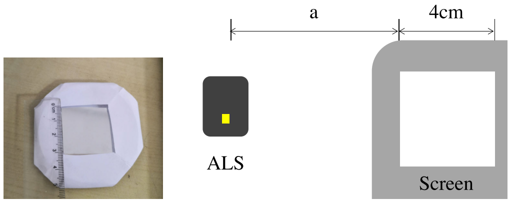
In this model, we use the coordinate system shown in Figure 2(a) to illustrate the power loss. Here we assume that the transmitter is a point light source. Although the screen is apparently not a point light source, we can take it as a combination of many discrete points (pixels). The original point O and A are two points on the screen plane and represent the ALS and the point light source (i.e., a pixel), respectively; B is a reflection point on the user's hand, and C is the projection of B on the screen plane.

We first describe the light traveling process: the light from A propagates to B, reflected by B and received by O. The traveling path can be decomposed into 4 parts: the screen-to-hand path ($A \rightarrow B$) in free space, reflection by hand (point B), hand-to-sensor ($B \rightarrow O$) path in free space, and reception at the receiver (point O). Next, we calculate the propagation loss for each part based on Lambertian radiation pattern [16].

We denote $\angle ABC$ by θ_1 and $\angle OBC$ by θ_2. For light power loss in the free-space, illuminating path follows the inverse-square law for visible light propagation. The loss from A to B, denoted by l_{AB}, is inversely proportional to $|AB|^2$ (i.e., $l_{AB} \propto \frac{1}{|AB|^2}$) and the loss from B to O follows $l_{BO} \propto \frac{1}{|BO|^2}$. When an area element is radiating as a result of being illuminated by an external source, the irradiance landing on that area element will be proportional to the cosine of the angle between the illuminating source and the normal. A Lambertian scatter will then scatter this light according to the same cosine law as a Lambertian emitter. Thus, the loss at point B (i.e., l_r) caused by reflection follows $l_r \propto (\cos\theta_1 * \cos\theta_2)$. Loss at receiver O follows $l_O \propto \cos\theta_2$.

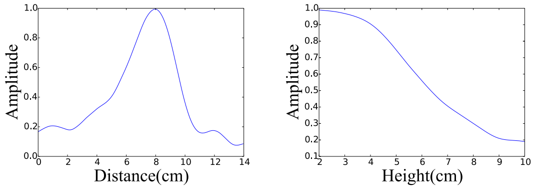
We denote $\overrightarrow{BC}=(0,-h)$, $\overrightarrow{BA}=(a-x,-h)$, $\overrightarrow{BO}=(-x,-h)$, $d_1=|\overrightarrow{BA}|$, $d_2=|\overrightarrow{BO}|$. According to the calculation above, the light intensity of signal from A to O is

$$\begin{aligned} I_O &= I_A \cdot l_{AB} \cdot l_r \cdot l_{BO} \cdot l_O \\ &= I_A \cdot c \cdot \frac{\cos\theta_1}{d_1^2} \cdot \frac{\cos\theta_2^2}{d_2^2} \\ &= I_A \cdot c \cdot \frac{h^3}{((x-a)^2+h^2)^{\frac{3}{2}}(x^2+h^2)^2}, \end{aligned} \tag{1}$$



(a) Reflecting object (b) Relative position setting

Fig. 3. Experimental setup for model verification. (a)The mirror we used as the reflecting object. (b) The experimental setting of transmitter and receiver's position.



(a) Horizontal movement (b) Perpendicular movement

Fig. 4. Experimental results for model verification. (a)The relationship between received light intensity and horizontal movement of reflecting object. (b) The relationship between received light intensity and perpendicular movement of reflecting object.

where c is a constant. Although c may depend on the user's skin color, its effect can be eliminated by some normalization techniques.

Now we know the light signal's amplitude at point O is related to x and h, that is, the hand's position. Particularly, we analyze the relationship between the amplitude and x, h. When $h > 0.5a$,[1] for the same value of h, when x is becoming larger, I_O is first monotonic increasing and then monotonic decreasing. I_O has the maximum value when $x = x_0$[2], $x_0 \in (0, a)$. For the same value of x, I_O increases first and then decreases with increasing h. When $h = h_0$,[3] the value of I_O is maximum. How I_O changes with x and h theoretically is shown in Figure 2(b).

We conduct experiments to verify the theoretical relationship we've analyzed above. The screen on a mobile phone serves as the light transmitter, and an ALS (TEMT6000) serves as a receiver. Both of them are placed on a horizontal plane (a desk). As shown in Figure 3(a), we take a $3cm \times 3cm$ mirror as the reflecting object to directly observe the relationship between the light intensity I_O and the position of reflection point (x, h). The size of the lighting block on the screen is $4cm \times 4cm$ and the distance between the block and the ALS (i.e., a) is $10cm$. Their relative position is shown in Figure 3(b).

To verify the relationship between I_O and x, the mirror is moved above the desk from the ALS to the lighting block at a constant speed. The height of the mirror above the desk is $h = 10cm$. The relationship between the I_O and x from the experiment is shown in Figure 4(a). For the second experiment, we set a shorter distance ($a = 2cm$) between the ALS and the lighting block to explore the relationship

[1] The condition is always true in our system since the largest a value in *SMART* represents the width of our mobile devices. h is the height of the hand above screen, which is about 10cm, while the width of screen is usually less than 20 cm.

[2] The analytical solution of x_0 is unsolvable.

[3] $h_0 = ((\frac{x^4}{64} + \frac{3}{4}x^2(x-a)^2)^{\frac{1}{2}} - \frac{x^2}{8})^{\frac{1}{2}}$

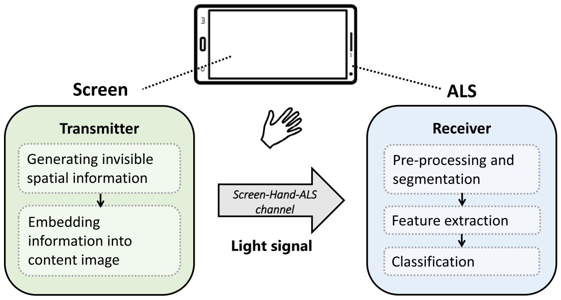

Screen
ALS
Transmitter
Generating invisible spatial information
Embedding information into content image
Screen-Hand-ALS channel
Light signal
Receiver
Pre-processing and segmentation
Feature extraction
Classification

Fig. 5. Overview of *SMART*.

between I_O and h. The vertical distance h changes from $10cm$ to $2cm$ at a constant speed. The experimental results are shown in Figure 4(b). We can conclude that both experimental results are consistent with our theoretical analysis. The hand's position affects the path loss during the propagation of light signal from screen to ALS. Therefore, it is feasible to exploit "Screen-Hand-ALS" light path to design *SMART*.

III. Overview

In this section, we provide the system overview of *SMART*. As we il lustrated above, *SMART* leverages the "Screen-Hand-ALS" light path to implement in-air gesture control on legacy mobile devices. As shown in Figure 5, the design of *SMART* mainly contains the transmitter side and the receiver side.

For the transmitter side (the screen display), *SMART* embeds spatial information into light signal while preserving viewing experience. *SMART* designs a mechanism to decouple the original frame into a pair of switching, complementary frames. As human eyes have persistence-of-vision effect, it looks the same as the original frame. Different blinking blocks are arranged at different positions on the screen to convey spatial information. In order to overcome the restriction of screen refresh rate, *SMART* exploits the line-by-line refreshing mechanism of screen [18] to provide high-frequency signals, which is above the frequency that human eyes can perceive.

For the receiver side (the ALS), *SMART* proposes a framework to recognize gestures from low-quality ALS data. The signal quality of the received light is poor, as the light from screen is attenuated during propagation and reflection, while the noise level is high. We carefully analyze the sensor data and identify distinguishing feature for the gesture recognition task. After feature extraction, *SMART* builds a lightweight classifier for gesture recognition.

IV. *SMART* Transmitter

In this section, we present the design of *SMART* transmitter. The main challenge is how to embed spatial information into the screen so that such changes remain invisible to human eyes while is obvious to an ALS. It is challenging for two reasons. First, the screen has a limited refresh rate and thus it can only generate low-frequency, human perceivable light signals. Second, the ALS is a one-pixel hardware. It can only sample the combined light intensity from all light sources around with no spatial resolution. To address these challenges, we first analyze the main differences between light sensors and human eyes.

A. Comparison Between Light Sensors and Human Eyes

To guide the design of *SMART*, we consider the differences between light sensors and naked eyes by focusing on two aspects: (i) what kind of information can be sensed by light sensors? (ii) how can it be ignored by human eyes?

Human eyes' structure is sophisticated and can perceive abundant information of the images displayed on screens. However, light sensors can simply convert light signal into electrical signal, which means that they can only sense light intensity. Following are the two main differences between them. First, a human eye can be seen as a linear low-pass filter and averages the high-frequency blinking light [19]. In other words, when a light source is flickering at a frequency above a certain threshold, human eyes will not perceive the flickering. However, light sensors can sample at a frequency higher than eyes can perceive. Second, human eyes can discriminate colors with different chromaticities, even if they have the same brightness. However, ALSs on mobile devices usually can only sense the brightness of light. We try to enlarge the signal that can be sensed by light sensors, and diminish the flickering effect that can be perceived by human eyes.

Based on the analysis, we design complementary frames to maximize the light signal received by the ALS while letting the signal be unobtrusive to human eyes. The design of *SMART* transmitter has the following three main components:

1) To avoid the flickering effect, we take advantage of the line-by-line refreshing principle of screen display, so that switching complementary blocks provides high-frequency light signals, which is above the frequency that human eyes can perceive (Section IV-B);
2) To keep the perceived colors of complementary frames look the same as the original image, we use color mixture principle to hide the blocks into screen content (Section IV-C);
3) We smooth the edges of complementary blocks to relieve the phantom array effect (Section IV-D).

B. Complementary Block Structure Design

Specifically, as a low pass filter, human eyes are sensitive to low-frequency light signals (the frequency below 50Hz [20]). When the refreshing rate of the screen is f_s, switching between two complementary frames can provide a light source with a maximum frequency of $\frac{f_s}{2}$. Assume that the refresh rate of the screen is 60Hz, the maximum light frequency can be 30Hz. The screen can also generate light signal with frequency 20Hz, 15Hz, 10Hz, which are all lower than the threshold. In order to overcome this frequency constraint, we take advantage of the line-by-line refreshing principle.

Similar to the rolling shutter effect of camera, screens on the mobile devices are updated line by line [18]. The pixels in a frame are updated from top to bottom. We design a pair of n-line complementary frames (frame A and frame B) as shown in Figure 6, and switch them continuously. The luminance of the whole screen alternates every $\frac{1}{n \cdot f_s}$, which is shorter than the original refreshing cycle $\frac{1}{f_s}$.

We notice that only when n is odd, by switching between two complementary frames, we can get a reliable

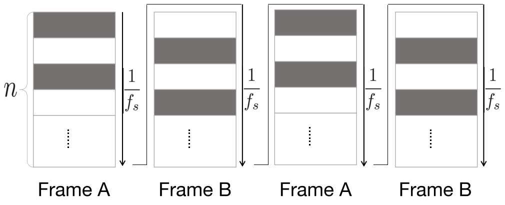


Fig. 6. Complementary frames.

high-frequency flickering light. The explanation is shown in Figure 7. If n is an odd number, the number of bright blocks increases and decreases alternatively all the time with a consistent pattern, generating a light signal with frequency $\frac{n \cdot f_s}{2}$. However, when n is even, the change from frame A to frame B is not symmetric with the change from frame B to frame A. Thus, the signal with frequency $\frac{n \cdot f_s}{2}$ does not exist. Therefore, an odd n should be chosen in *SMART*.

We do some experimental measurements to verify our analysis above and determine the proper n value. We use iPad Pro $11 (f_s = 120fps)$ as the transmitter and an ALS (TEMT6000) as the receiver. We divide the screen into n lines as shown in Figure 6. We set $n = 1$ as the control group. We play a pre-processed video to switch a pair of complementary frames (frame A and B). Light signals are sampled by the ALS over the screen. The horizontal distance between the ALS and the screen is about 10cm. The frequency spectra of different n values are shown in Figure 8. We can observe that: (i) The signal of frequency $\frac{n \cdot f_s}{2}$ is indeed generated as we've analyzed above. However, someone may be concerned that the frequency is generated by harmonic effect, since $\frac{n \cdot f_s}{2}$ is $\frac{f_s}{2}$'s harmonic frequency. By comparing the figures, we can observe that when the main frequency power is lower, the harmonic frequency power becomes larger, which is caused by the line-by-line refreshing mechanism. Then we conclude that the peak power of $\frac{n \cdot f_s}{2}$ is mainly caused by the line-by-line refreshing mechanism rather than the high-frequency harmonics of $\frac{f_s}{2}$. Here we take $n = 5$ as an example. When $n = 5$, $\frac{n \cdot f_s}{2} = 300Hz$ and the relative amplitude is 0.21. The amplitude at $\frac{f_s}{2}$ of $n = 1, 3$ is larger than the amplitude at the same frequency of $n = 5$. However, the amplitude at $\frac{5 \cdot f_s}{2}$ of $n = 1, 3$ is lower than the amplitude at the same frequency of $n = 5$. Thus, although $\frac{5 \cdot f_s}{2}$ is a harmonic frequency of $\frac{f_s}{2}$, the signal power at frequency $\frac{5 \cdot f_s}{2}$ is not mainly generated by harmonic effect. (ii) When n is smaller, the power of frequency $\frac{n \cdot f_s}{2}$ is larger. We notice that when $n = 3$, the amplitude of signal at frequency $\frac{n \cdot f_s}{2} = 180Hz$ is 0.61. However, with a larger n, the power of $\frac{n \cdot f_s}{2}$ is weaker. When $n = 7$, the amplitude of $\frac{n \cdot f_s}{2} = 420Hz$ becomes quite low (the value is 0.12). Thus, in addition to signal frequency, the value of n also affects the power of the flickering light. With a smaller n, the flickering area is larger, which brings a larger light intensity change since the light intensity is proportional to the flickering area. As we prefer a large signal energy to get a higher SNR, we choose $n = 3$.

Flickering blocks are arranged on one side of the screen, as shown in Figure 9(a). We design the complementary frames in this way, as we expect the following three frequency components in the received light signal can be used in gesture recognition: (i) $f_0 = 0$ Hz: ambient light has a low frequency and the main power falls around 0 Hz [21]. When hand is approaching the light sensor, DC current from the light sensor decreases under bright conditions or increases under dark conditions. This is because if the user is in a bright room, low-frequency light is mainly from the ambient light, which will be blocked by the palm. However, if the room is dark, low-frequency light signal is mainly from the screen backlight reflected by the user's palm. (ii) $f_1 = \frac{f_s}{2}$ Hz: the two complementary frames alternate at f_s results in f_1. According to the model we build in Section II, the power of f_1 becomes larger if the hand is coming closer to the whole flickering zone. (iii) $f_2 = \frac{n \cdot f_s}{2}$ Hz: this frequency component is caused by the line-by-line refreshing mechanism. If the hand is approaching the middle part (vertically) of the complementary block area, more light generated by the complementary blocks will be reflected. Thus, the power of f_2 will also change with the hand position.

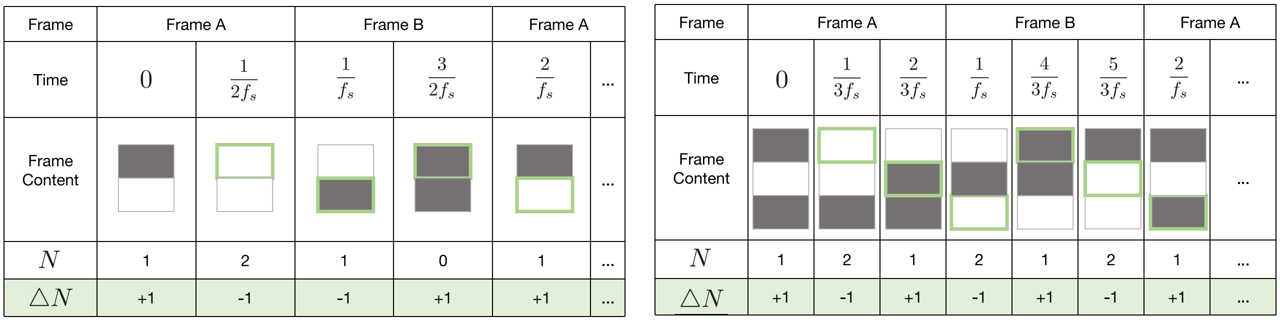
C. Hiding Complementary Blocks Into Screen Content

In this subsection, we propose a color decomposition algorithm to hide the complementary blocks in the screen content, so that users will not perceive the color difference from the original display. According to the analysis in Section IV-A, our goal is to maintain chromaticities while maximize the luminance difference between complementary frames.

RGB color space is widely used on mobile devices. However, it is not designed according to human color vision. Considering human eye perception, we need to convert RGB images into another space that suits human vision system [17]. Several color spaces are created to quantify human color vision. The CIE 1931 XYZ color space is one of the first defined quantitative links between distributions of wavelengths in the electromagnetic visible spectrum, and physiologically perceived colors in human color vision, which has a linear relationship with RGB color space. Specifically, the color of each pixel is converted from (R, G, B) into (X, Y, Z), in which Y parameter determines the luminance of a color. The chromaticity can be specified by the two derived parameters $x = \frac{X}{X+Y+Z}$ and $y = \frac{Y}{X+Y+Z}$ [22].

As mentioned before, light sensors can only sense light intensity while ignoring chromaticity. However, human eyes are sensitive to chromaticity change. In order to minimize the eye's perception of the image distortion caused by mixing two frames, we keep chromaticity (x, y) of the complementary blocks the same as the original pixel (x_0, y_0); on the other hand, we maximize the luminance change of complementary pixels so that the light sensor can receive stronger flickering light. We model the problem as an optimization problem. Here we denote colors of complementary pixels as (x_1, y_1, Y_1) and (x_2, y_2, Y_2) and the original color is (x_0, y_0, Y_0). ΔY is the luminance difference between a pair of pixels. The optimization problem is shown as follows:

$$\max \ \Delta Y = |Y_1 - Y_2|$$



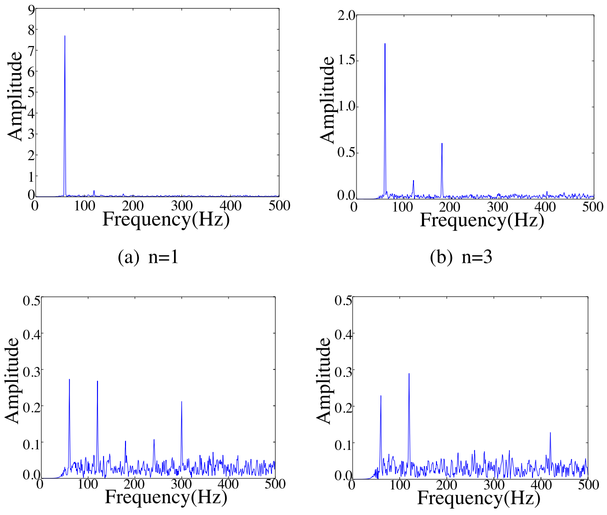
Frame	Frame A		Frame B		Frame A	
Time	0	$\frac{1}{2f_s}$	$\frac{1}{f_s}$	$\frac{3}{2f_s}$	$\frac{2}{f_s}$	...
Frame Content						...
N	1	2	1	0	1	...
$\triangle N$	+1	-1	-1	+1	+1	...

(a) $n = 2$

Frame	Frame A			Frame B			Frame A	
Time	0	$\frac{1}{3f_s}$	$\frac{2}{3f_s}$	$\frac{1}{f_s}$	$\frac{4}{3f_s}$	$\frac{5}{3f_s}$	$\frac{2}{f_s}$	...
Frame Content								...
N	1	2	1	2	1	2	1	...
$\triangle N$	+1	-1	+1	-1	+1	-1	+1	...

(b) $n = 3$

Fig. 7. Illustration of the high-frequency light signal generation based on line-by-line refreshing scheme of the screen in detail. Here N denotes the number of bright blocks in the current frame.



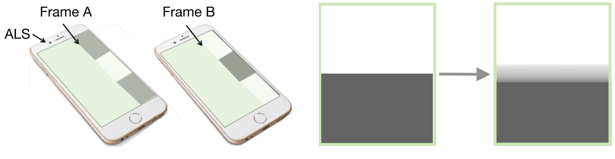
(a) n=1

(b) n=3

(c) n=5

(d) n=7

Fig. 8. Frequency spectra of different n values.



(a) Block's positions

(b) Edge smoothing

Fig. 9. The specific complementary frames on the screen. (a)Positions of blocks and ALS on the mobile device. (b)The edge smoothing scheme to attenuate phantom array effect.

$$s.t. \begin{cases} x_1 = x_2 = x_0, \\ y_1 = y_2 = y_0, \\ Y_0 = \dfrac{Y_1 + Y_2}{2} \end{cases}$$

We formulate the problem as a linear optimization problem. Given x_0 and y_0, we calculate the best combination of Y_1 and Y_2 theoretically. First, we calculate the maximum luminance value Y_{max} of (x_0, y_0). The luminance difference between the original pixel and the modified pixel is $\Delta Y = min(Y_{max} - Y_0, Y_0)$. Then we gain $Y_1 = Y_0 + \Delta Y$ and $Y_2 = Y_0 - \Delta Y$. In order to reduce the processing latency, we can compute Y_{max} offline for each pair of (x_0, y_0) and store the results in a lookup table.

However, we find that the hiding effect of the complementary blocks in practice is unsatisfactory when the original pixel color's luminance Y_0 is small. It means that the theoretical methodology we've described above is not feasible to all colors displayed on screen. Next we design a color calibration method to gain better vision perception.

We test the fusion effect of colors with different luminances. Specifically, We test RGB values in different intervals. The range $[0, 255]$ of RGB value is uniformly divided into three intervals, including $I_1 = [0, 85)$, $I_2 = [85, 170)$, $I_3 = [170, 255]$. Since each RGB value includes the 3-dimensional value (R, G, B), $27(3 \times 3 \times 3)$ RGB values are selected randomly. We select a value in each interval randomly every time. We find if $Y_0 < r \times Y_{max}$, the vision perception becomes worse.

After trial-and-error procedure, we find a threshold ratio $r = 0.79$ which can utmostly hide blocks in the screen content and correct the color decomposition algorithm. Specifically, the original methodology is effective when $Y_0 > r \times Y_{max}$. However, if $Y_0 < r \times Y_{max}$, mixing $Y_1 = \frac{Y_0}{r}$ and $Y_2 = 0$ can

Algorithm 1 Color Decomposition With Maximum Luminance Change

```
Input: 1)$Y_{max} \in R^{M \times N}$, the maximum luminance
       values for different chromaticity values; 2)$r$,
       the threshold ratio; 3)$R_0G_0B_0$, RGB value of a
       pixel
Output: Color decomposition result $R_1G_1B_1$ and
        $R_2G_2B_2$
$X_0Y_0Z_0 \leftarrow f_{RGBtoXYZ}\ (R_0G_0B_0)$
$x_0 \leftarrow \frac{X_0}{X_0+Y_0+Z_0}$
$y_0 \leftarrow \frac{Y_0}{X_0+Y_0+Z_0}$
if $Y_0 < r \times Y_{max}[x_0][y_0]$ then
    $Y_1 \leftarrow \frac{Y_0}{r}$
    $Y_2 \leftarrow 0$
else
    $\Delta Y \leftarrow Y_{max}[x_0][y_0] - Y_0$
    $Y_1 \leftarrow Y_0 + \Delta Y$
    $Y_2 \leftarrow Y_0 - \Delta Y$
$X_1 \leftarrow x_0 \times \frac{Y_1}{y_0}$, $X_2 \leftarrow x_0 \times \frac{Y_2}{y_0}$
$Z_1 \leftarrow (1 - x_0 - y_0) \times \frac{Y_1}{y_0}$, $Z_2 \leftarrow (1 - x_0 - y_0) \times \frac{Y_2}{y_0}$
$R_1G_1B_1 \leftarrow f_{XYZtoRGB}\ (X_1Y_1Z_1)$
$R_2G_2B_2 \leftarrow f_{XYZtoRGB}\ (X_2Y_2Z_2)$
```

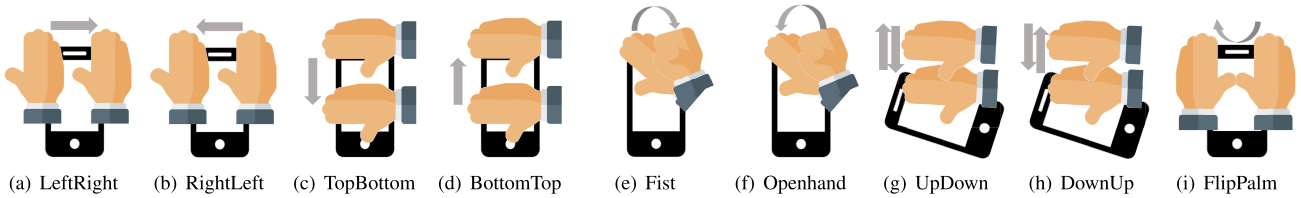

(a) LeftRight (b) RightLeft (c) TopBottom (d) BottomTop (e) Fist (f) Openhand (g) UpDown (h) DownUp (i) FlipPalm

Fig. 10. Nine gestures of *SMART* .

achieve better user perception. Algorithm 1 gives the details of the color decomposition procedure.

D. Edge Smoothing

After hiding the well-designed spatial information into the screen content, we find that the high-frequency flickering at the edge of the flickering blocks is still visible when human eyes is blinking or moving. The phenomenon has been mentioned in [19]. It is because human eyes are more sensitive to the flickering of moving light source than the static light source. The phenomenon is called phantom array effect. To mitigate the effect, we smooth edges of flickering blocks by scattering the switching pixels near the edges. As shown in Figure 9(b), when approaching to the boundary of the flickering blocks, the density of complementary pixels gradually decreases. Then the abrupt change between two different blocks becomes smooth. In this way, *SMART* relieves the phantom array effect.

V. Receiver

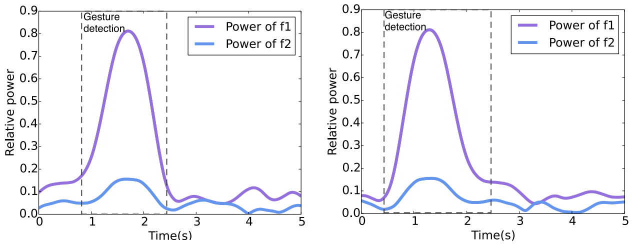
SMART receiver extracts features from ALS readings and recognizes different gestures. Our goal is to distinguish 9 in-air gestures that are frequently used in commodity mobile devices. The gestures are illustrated in Figure 10. Specifically, "Left-Right", "Right-Left", "Top-Bottom", "Bottom-Top" can be used for page turning. "Fist" and "Open hand" are commonly used for screenshot and zooming.

The main design challenge is to extract distinguishing features from the light signal. The strength of signal from the screen becomes weak after the propagation and reflection loss. Besides, the noise is large since the sampling rate of ambient light sensor is limited [13], [14]. The receiver needs to extract reliable features for gesture recognition from the low-quality, down-sampled signal. To address the challenge, we design strategies to segment exact gestures from the time-series of signal and extract distinguishable features strongly related to gestures.

A. Pre-Processing

As present in Section IV-B, we are interested in f_1 and f_2, which are produced by the complementary blocks displayed on screen. Since the sampling rate of ALS (denoted by f_l) on mobile devices is usually low, e.g., 100Hz, according to Nyquist-Shannon sampling theorem, it can only sample up to $\frac{f_l}{2}$. In order to recover frequency above $\frac{f_l}{2}$, we use frequency aliasing [23].

When sampling a high frequency signal at sub-Nyquist rate, the high frequency component will be aliased to low frequency spectrum as follows:



(a) Left-Right (b) Top-Bottom

Fig. 11. f_1 power and f_2 power during different gesture. Here we use "Left-Right" and "Top-Bottom" gesture as two examples to compare power of f_1 and f_2.

$$f_a = \begin{cases} (N+1)f_l - f, & f_l/2 < f - N \cdot f_l < f_l \\ f - N \cdot f_l, & 0 \le f - N \cdot f_l \le f_l/2 \end{cases} \tag{2}$$

where f is the signal frequency and f_a is the aliasing frequency, $N = 0, 1, 2, \cdots$. According to Equation (2), we can get aliasing frequency of f_1 and f_2, denoted by f_1^a and f_2^a, respectively. We use FFT to extract signal energy of f_1 and f_2 (f_1^a and f_2^a actually). Besides, we also utilize power on f_0 to show the relative hand position with ALS. Energy of f_0, f_1 and f_2 are denoted by E_0, E_1 and E_2, respectively.

B. Segmentation

We segment light signal based on the FFT results. A gesture is detected when the power of f_1 and f_2 is large. It is because only when a user is making gesture, the "Screen-Hand-ALS" channel can be built and the light signal from screen can be reflected to ALS. Our target is to cut and analyze the segment of light signal with high power (which also means high SNR). As shown in Figure 11, for each gesture, we can get two observations: (i) the power of f_1 is always larger than f_2 during the relatively high light signal period; () time period with large power is nearly the same for both f_1 and f_2. Thus, f_1's power is a reliable indicator for the signal power.

To detect the gesture, a simple solution is to select an empirical threshold for f_1's power. A gesture start is detected when f_1's power becomes higher than the threshold, and end is detected when the power fall below the threshold. However, the f_1's power may falls below the threshold briefly during a gesture. The phenomenon mainly happens with several gestures composing of complex steps (e.g., UpDown, FlipPalm). Thus, we propose a interval-omitted segmentation method, which can extract the high power zone and omit possible low power intervals.

Next we describe the segmentation algorithm in detail. We use a variable $State$ to mark the current segmentation status. The initial value of $State$ is "Init". When the power

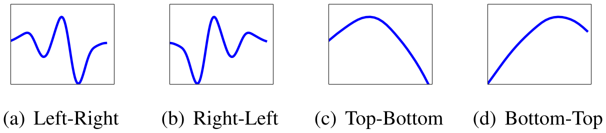

(a) Left-Right (b) Right-Left (c) Top-Bottom (d) Bottom-Top

Fig. 12. The patterns of additional features added according to our observation. (a) The pattern of $E_0' \cdot E_1'$ during "Left-Right" gesture. (b) The pattern of $E_0' \cdot E_1'$ during "Right-Left" gesture. (c) The pattern of R_{12} during "Top-Bottom" gesture. (d) The pattern of R_{12} during "Bottom-Top" gesture.

of f_1 (denoted as E_1 in the following text) is larger than the empirical threshold $EThre$, $State$ is set as "Start", which represents a gesture start. Then, once E_1 falls below the threshold, it represents either a finished gesture or a procedure during a gesture. If the low power period is smaller than $0.5s$, it is treated as a part of a gesture. Otherwise, the gesture is thought to be done and the $State$ is set as "End".

To reduce the probability of false detection, we add two segmentation constraints: (i) Ambient light standard deviation. We observe the segmentation result and find that some random electrical noise (caused by hardware imperfections/electromagnetic interference) can cause false trigger. To address this problem, we use standard deviation of ambient light power as a constraint. When a user is performing a gesture, ambient light power (denoted as E_0 since ambient light is mainly conposed of the frequency $0Hz$) received by ALS fluctuates significantly since hand motion can block/unblock ambient light. Thus, we select a threshold value $SThre$ for the standard deviation of E_0. If the standard deviation of E_0 is smaller than $SThre$, we discard the segmentation result. (ii) Gesture length. The instant change of light intensity in the environment (e.g., light on/off) can generate complicated light frequencies, containing the target frequencies (e.g., $\frac{f_s}{2}$, $n \cdot \frac{f_s}{2}$). However, the duration is much shorter than a gesture. By setting a threshold on the gesture duration, *SMART* can discard the detected segments which are shorter than the threshold. After trial-and-error procedure, we set the threshold to be $0.5s$ in *SMART*.

C. Feature Analysis and Selection

As mentioned in Section IV-B, the time-series of 0 Hz, f_1, f_2 energy are closely related to hand position. The 3 sequences after segmentation are shown in Figure 13. However, these time series cannot be directly used as features for gesture recognition. According to our observation, we need to address the following two problems.

First, as we have mentioned before, $f_1 = \frac{f_s}{2}$, $f_2 = \frac{n \cdot f_s}{2}$. Since f_2 is f_1's harmonic frequency, a part of f_2's energy is from f_1. Thus, E_1 and E_2 are coupled and E_2 depends on E_1. How to decouple E_1 and E_2 and amplify the difference between E_1 and E_2? We have an observation that the ratio of E_1 and E_2, $R_{12} = \frac{E_1}{E_2}$, is relevant with hand position. When hand is approaching the middle of the complementary block area (for example, when the hand is performing "Top-Bottom" gesture), R_{12} increases; on the contrary, R_{12} decreases if hand is moving reversely. To illustrate the observation, Figure 12(c) and Figure 12(d) show the pattern of R_{12} for gesture "Top-Bottom" and "Bottom-Top" as two examples. Thus, we use R_{12} as a feature.

Second, different gestures may have similar patterns of E_0, E_1. For example, for both gestures "Left-Right" and "Right-Left", each pair of patterns are similar, but temporal relationship between E_0 and E_1 is different. For gesture "Left-Right", E_0 decreases first and then E_1 increases. For gesture "Right-Left", E_0 decreases after E_1 increasing. Figure 10 shows the phenomenon directly.

The temporal relationship between E_0 and E_1 can reflect the relative position among user's hand, light sensor, and blinking block, which is an important feature for gesture recognition. In order to capture the inherent temporal relationship, We take the derivative of E_0 and E_1 with respect to time, to show the changing trend separately. As shown in Figure 12(a) and Figure 12(b), $E_0' \cdot E_1'$ can clearly discriminate between "Left-Right" and "Right-Left" gestures.

To sum up, four feature are used in classification: E_0, E_1, R_{12} and $E_0' \cdot E_1'$. The effect of the later two features on *SMART* is further evaluated in Section VII-B.

D. Classification

After obtaining four feature series, we build classification model for gesture recognition. Our method is similar with [21]. First, we apply Z-score normalization on features such that every feature stream has zero mean and unit variance. After that, we train a k-nearest neighbour (KNN) classifier using DTW distance as the distance metric in order to eliminate the effect of different gesture speeds. In specific, the distance from the test point to its neighbour points is the sum of DTW distances between each pair of feature series. In Section VIII, we compare the performance of different models, including recurrent neural networks (RNN), long short-term memory networks (LSTM), multilayer perceptrons (MLP), convolutional neural networks (CNN), gated recurrent networks (GRN) and KNN.

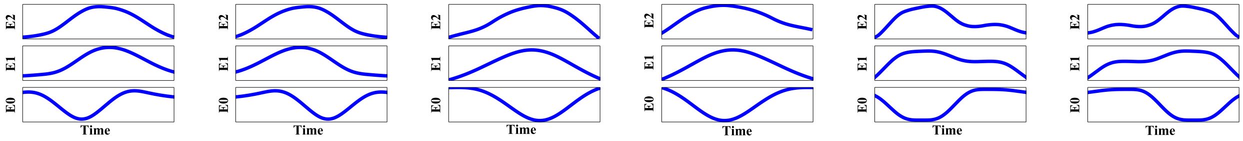
The details of these neural networks are as follows. The MLP model used in our experiments is composed of two fully connected layers and a linear classifier. ReLU is used as the activation function. The dimension of intermediate layers is 400. The RNN model used in our experiments is composed of two RNN layers, followed by a linear classifier. Sigmoid is used as the activation function. The dimensions of the hidden states of RNN layers are 64. The LSTM and GRN models have the same architectures except replacing the RNN layers in the RNN model with LSTM and GRN layers. The CNN model used in our experiments is composed of a convolutional layer, followed by a max pooling layer and a linear classifier. The input and output dimensions of the convolutional layer are 1 and 64, respectively. Its kernel size is 4×3.

VI. Prototype

In this section, we describe the prototype of *SMART*.

A. Transmitter

We implement *SMART* on a commercial off-the-shelf tablet, i.e., iPad Pro with an 11-inch screen. As the operating system restrains the operation access to the screen driver, we use



(a) Left-Right (b) Right-Left (c) Top-Bottom (d) Bottom-Top (e) Fist (f) Open hand

Fig. 13. Signal patterns for several gestures.

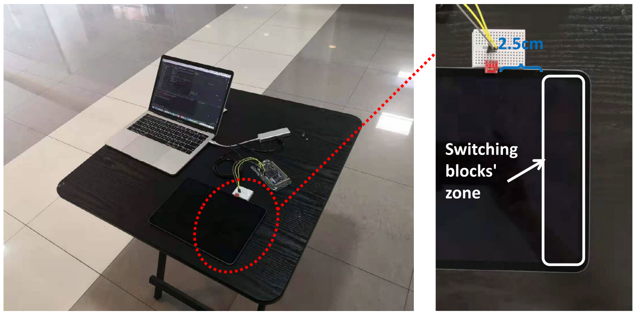


Fig. 14. Experiment setup.

TABLE I
EXPERIMENT SETTING

Item	Number	Value
User	14	9 males, 5 females
Gesture	9	LeftRight, RightLeft, TopBottom, BottomTop, Fist, Openhand, UpDown, DownUp, Flip
Environment	5	0lux, 150lux, 350lux, 700lux, 2000lux

pre-processed videos to emulate the switching between complementary frames. The blinking blocks are positioned on one side of the screen. The width of the blinking zone is about 5cm, which can fit onto the screens of the majority of mobile phones [24]. Thus, *SMART* can not only be implemented on tablets, but also on smartphones. By default, the brightness of the screen is 100% and the screen displays a coffee shop picture.

We use a standalone ambient light sensor (i.e., TEMT6000) as the receiver since the operating system also restricts the sampling rate of light sensors on commercial off-the-shelf devices [14]. ALS is connected to an Arduino DUE microcontroller as the receiver. We place the ALS just above the screen as shown in Figure 14 to emulate the relative position between screen and light sensor on commercial devices. The distance between the light sensor and the blocks' left edge is 2.5cm. The default sampling rate of ALS is set to 250Hz, since the integration time of most ALSs are below 4ms [25], [26], [27].

VII. Evaluation

In this section, we evaluate *SMART* in terms of gesture detection accuracy, gesture recognition accuracy, user perception, and processing latency. We also conduct an in-depth comparison between *SMART* and the gesture recognition functionality on COTS smartphones.

We test *SMART* in 5 environments with 14 users (9 males and 5 females) in the age range of 20 to 30. Users perform gestures at approximately 10cm above the screen. Our experiments are conducted in five typical environments. Table I summarizes the experiment settings.

A. Detection Accuracy

We evaluate the detection accuracy of *SMART*. First, we test the mis-detection rate of *SMART* under 150lux lighting environment. We ask 4 users to perform each gesture for 20 times and the system is tested for 720 times in total ($4 \times 20 \times 9.720$). The received light signals of *SMART* for the 4 users are sampled and recorded. We separately use both the threshold-based segmentation method and the interval-omitted segmentation method to segment the data and evaluate the mis-detection rate. According to the experimental results, for the threshold-based segmentation, the mis-detection rate is 6.39%. However, the mis-detection rate of the interval-omitted segmentation algorithm is 2.22%. The interval-omitted segmentation algorithm is valid according to the experimental result.

Besides, we investigate that whether *SMART* has false positives. According to our segmentation scheme, *SMART* detects a gesture only when it detects the frequency which is the same as the frequency emitted from the screen, i.e., $\frac{f_s}{2}$. Thus, it can not be triggered without the light signal reflected by hand.

To test whether *SMART* will be triggered by other confounding factors, we test *SMART* under two scenarios: (i) Human interference. We put our prototype on the desk (without user making gesture). A subject is asked to walk around the desk. We run *SMART* for 5 minutes and we found that there is no false positive. Besides, we let a user's face come close to the prototype. We find that when the user's face is within 15cm from the screen, it will cause false positives. However, such a small distance between eyes and screen can cause visual fatigue [28]. Thus, we believe that face will not falsely trigger *SMART* in daily use. (ii) Light on/off. We put our prototype on the desk. A lamp on the desk is switched on/off alternately every 5 seconds, causing 200lux light intensity change each time. We also run *SMART* for 5 minutes. We find that although the instant change of light intensity will generate complicated light frequencies, containing the target frequencies (e.g., $\frac{f_s}{2}$, $n \cdot \frac{f_s}{2}$), the signal lasts for just 0.1s-0.3s, which is much shorter than a gesture. By setting a threshold on the gesture duration (e.g., 0.5s), *SMART* can discard the detected segments which are shorter than the threshold.

B. Recognition Accuracy

We evaluate the classification accuracy of *SMART* with different design choices and different environment settings. We ask 14 users to perform each gesture for 20 times.

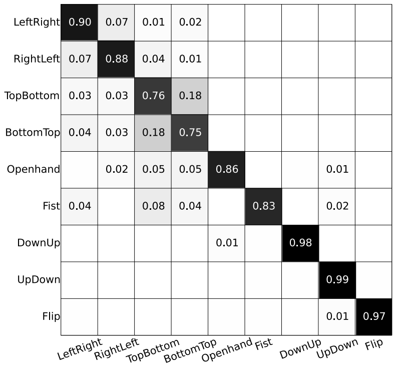


Fig. 15. The confusion matrix for feature set 1.

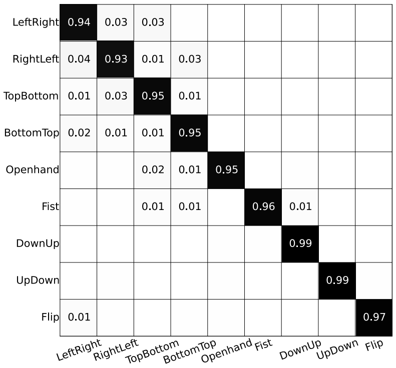


Fig. 16. The confusion matrix for feature set 2.

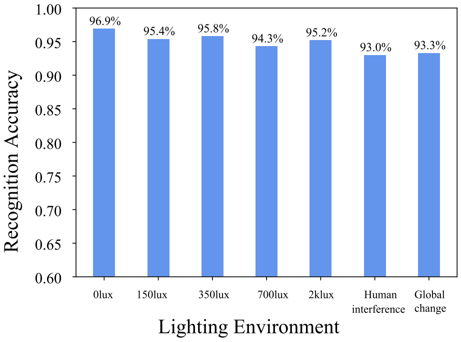


Fig. 17. Recognition accuracy in different lighting environments.

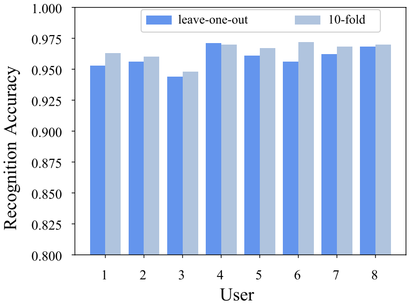


Fig. 18. Recognition accuracy of different users.

Besides, to investigate the robustness of *SMART* for various environments, one user is asked to perform each gesture for 50 times in each environment. By default, we use the average of 10-fold cross-validation as the final result.

Different Feature Sets. In order to show the effectiveness of the features we've selected in Section V-C, we compare the recognition accuracy when the system is trained with different sets of features. Feature set 1 only includes the time series of E_0, E_1 and E_2, which are the power of DC, f_1, f_2, respectively. Feature set 2 contains the four features present in Section V-C. Figure 15 and Figure 16 show the confusion matrices of the recognition framework trained with the two feature sets, separately. We can see that feature set 2 achieves 96.1% accuracy compared to that of 87.3% for the feature set 1. Especially, for the four gestures "TopBottom", "BottomTop", "Openhand", "Fist", the accuracy is improved from 79.6% to 95.3% with the two carefully designed features, i.e., R_{12} and $E'_0 \cdot E'_1$.

Different Lighting Environments. We test 5 static environments that correspond to common lighting conditions: (i) A completely dark room. The light intensity is 0 lux. (ii) A conference room with lighting infrastructure on at night. The average light intensity is about 150 lux. (iii) A lounge environment in day time. The average light intensity in the room is about 350 lux. (iv) A normal office in day time with sunlight and lighting infrastructure. The average light intensity is about 700 lux. (v) A bright corridor besides a window in the afternoon. The average light intensity is about 2000 lux.

To examine the influence of light fluctuations on recognition accuracy, we also investigate two common dynamic light environments: (i) Human interference: We ask one subject to perform the nine gestures and another subject is commanded to walk around the place. Each type of gesture is tested for 20 times in 4 light environments (except for the 700lux normal office, since there is no space around the testbed allowing a subject to walk around). (ii) Global light intensity variation: We conduct the experiment in the office with multiple light sources. A user performs each gesture for 20 times, while one lamp, on the same desk as the testbed, is switched on/off every 3s. The ALS measures the light intensity changes between 600lux and 750lux.

Figure 17 presents the recognition accuracy under the different light conditions. We can observe that (i) the recognition accuracies under the static environments range from 94.3% to 96.9%, which means that *SMART* works well under static environments. (ii) the accuracies in the two dynamic light environments are above 93%. Thus, *SMART* is able to work at various ambient light intensities, from dark (0lux) to bright (2000lux) indoor environment, and is robust under dynamic changing light conditions.

User diversity. To investigate the robustness of *SMART* for unseen users, we use both leave-one-out and 10-fold cross validation to evaluate the accuracy of each user. With leave-one-out, the test user's samples are excluded in the training set. The average leave-one-out accuracy for all 14 users is 95.6%. The recognition accuracy of the 8 users among the total 14 users are shown in Figure 18. The leave-one-out and 10-fold cross validation results of each user are similar, which means that *SMART* is a generic model, not a personalized model.

Unseen Scenarios. We consider the performance of *SMART* for unseen environments. We use leave-one-out cross validation. We find that we can achieve 96% accuracy with KNN if

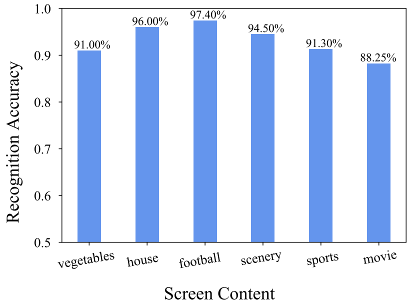


Fig. 19. Evaluation with different contents.

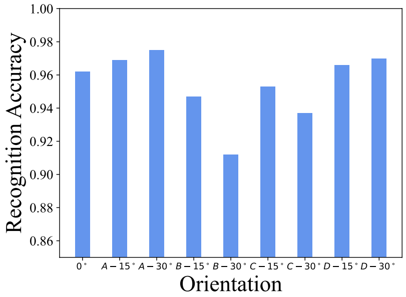


Fig. 20. Recognition accuracy for different orientations.

tested environment's samples are included in the training set, while we achieve 88.7% accuracy for unseen environments.

To improve the performance of unseen scenarios, we further propose to replace the KNN classifier with a gated recurrent neural network (GRN) to achieve better performance. This model is built with two bi-directional gate recurrent layers with dropout for feature extraction and one fully connected layer for classification. Our experiments show that it achieves 93.45% average accuracy on "unseen" environments. Besides, the performance of GRN can be improved with model ensemble, which jointly considers the output of multiple models and determines the final label. Usually, model ensemble can promote accuracy at the price of more computation and storage consumption. Our experiments demonstrate that the ensemble of 2 GRNs and 5 GRNs achieve 94.27% and 95.61% average accuracy on "unseen" scenarios, respectively.

Different Screen Contents. We evaluate the recognition accuracy of both static and dynamic contents. (i) Static contents: We test the gesture recognition accuracy of 3 different static contents (vegetables, coffee house, football field). The three contents separately corresponds to three levels of average ΔY: $(20, 40)$, $(40, 60)$, $(60, 80)$. As shown in Figure 19, we can observe that with a larger ΔY, the recognition accuracy becomes higher. It is easy to understand since larger ΔY means higher SNR of light signal from screen, leading to more distinguishable features. (ii) Dynamic contents: We also test the gesture recognition accuracy of 3 types of dynamic contents including scenery video, sports, and movies. They respectively represent video with minor, medium, and drastic frame transition. For each video type, we choose 3 video clips, each about 30-90s. During the test for each video clip, we play the video clip on a loop and the subjects perform each gesture for 10 times at random moments. As shown in Figure 19, we can see that the gesture recognition accuracy of *SMART* is acceptable when the screen is displaying dynamic content. Although the dynamic content changes the light intensity, for the majority of time, it changes in a smooth and slow way. Furthermore, the duration of a gesture is usually short (around 1-2s [21]) and screen light will not change significantly within such a short interval. Thus, hand gesture plays the dominant role in the received light intensity.

Different Orientations. We evaluate the recognition accuracy of the smartphone's different orientations. We conducted tests on four different orientations, specifically angles of 15° and 30°. The orientations are shown in Figure 24. Since the gesture is not parallel to the screen, the vertical distances between hand and different part of the screen are not same. Here we set the vertical distance between hand and the ambient light sensor about $10cm$. For each angle, we test 9 gestures and each gesture is performed 20 times. The results are shown in Figure 20. Our observation is that the accuracy is mainly influenced by the vertical distance between the flickering block on the screen and the hand. If the vertical distance between the flickering block and the plane of gesture becomes larger, the light signal power received by the ambient light sensor becomes lower which causes the recognition accuracy to decline.

Different classification algorithms. In order to compare with state-of-the-art models, we have added more experiments with 5 different neural network models, including recurrent neural networks (RNN), long short-term memory networks (LSTM), multilayer perceptrons (MLP), convolutional neural networks (CNN) and gated recurrent networks (GRN). Each model is trained by 300 epochs and optimized with Adam. The learning rate is set to 1×10^{-3} and decayed by 10 at the 150th epoch. Their 10-fold cross validation accuracies and FLOPs have been shown in Table III.

We have the following three observations: (i) LSTM, GRN and MLP achieve higher accuracy than the KNN classifier at the expense of more storage and computation cost. (ii) The accuracy of CNN is low. The possible reason may be that although CNN is good at capturing local features, it is weak in sequence processing. (iii) The accuracy of RNN is lower than 90% but its two variants LSTM and GRN achieve more than 97% accuracy. The main reason may be that RNN does not perform well on long sequences but many signals in our datasets have long time steps. LSTM and GRN address this issue by introducing additional control gates and thus perform much better.

Improved Techniques for Neural Networks. The aforementioned neural network for gesture classification achieves clearly higher accuracy than KNN. However, these neural networks are directly borrowed from image classification and text classification and not specifically optimized for gesture classification. To further improve their performance for gesture classification, in this paragraph, we study them from the perspectives of architecture design, data augmentation, and the training methodology.

Network Architecture The architecture of deep neural networks has a substantial impact on the performance of neural networks. We further investigate the performance of GRN with different architectures, including (i) different numbers of gated

TABLE II

PERCEPTION SCORES

Score	Image color difference	Flicker	Visual fatigue
1	Completely the same	Completely no flicker	No visual fatigue
2	Almost the same	Almost no flicker	A little visual fatigue
3	A little different	A little flicker	Evident visual fatigue
4	Evidently different	Evidently flicker	Strong visual fatigue

TABLE III

EXPERIMENTS OF NEURAL NETWORK MODELS AND THE KNN CLASSIFIER ON 10-FOLD CROSS VALIDATION. "PARAMETERS" AND "FLOPS" INDICATE THE NUMBER OF PARAMETERS AND THE FLOATING POINT OPERATIONS IN THE MODEL

Model	Accuracy	Parameters	FLOPs
RNN	88.62	135690	13365760
LSTM	97.50	535050	53660160
GRN	97.85	401930	40296960
CNN	89.18	1482	82176
MLP	97.60	323206	322400
KNN	96.00	–	–

TABLE IV

THE RECOGNITION ACCURACY OF GRN MODELS WITH DIFFERENT DATA AUGMENTATION TECHNIQUES IN UN-SEEN SCENARIOS AND CROSS-VALIDATION SETTINGS

Random Cropping	×	✓	×	×	✓
Gaussian Noise	×	×	✓	×	✓
Mix-up	×	×	×	✓	×
Un-seen scenarios	93.5	93.7	93.8	87.5	**94.0**
Cross-Validation Accuracy	97.9	98.2	98.1	90.0	**98.5**

TABLE V

THE ACCURACY OF DIFFERENT NEURAL NETWORKS WITH IMPROVED TECHNIQUES

Model	RNN	LSTM	GRN	CNN	MLP
Un-seen Scenario Accuracy	91.8	93.7	94.2	89.4	94.0
Cross-validation Accuracy	90.4	97.9	98.5	93.1	98.2

recurrent layers, (ii) different numbers of the intermediate channels, (iii) the usage of attention layers [29], (iv) the usage of dropout layers, and (v) the usage of batch normalization layers [30].

Our experimental results show that (i) GRN with 3, 4, 5 and 6 gated recurrent layers achieve 91.4%, 91.9%, 91.4% and 89.3%, respectively. (ii) GRN with 64, 128, 256, 512, 1024 channels achieve 87.4%, 90.2%, 93.5%, 91.0% accuracy, respectively. (iii) GRN with attention layer, dropout layers, and batch normalization layers achieve 93.7%, 93.6%, and 93.4% accuracy, respectively.

These results indicate that (i) The performance of neural networks is sensitive to the size (number of parameters) of neural networks. On one hand, with few layers and channels, the neural networks does not have enough representation ability for recognition. On the other hand, with too many layers and channels, the neural network tend to be overfit and thus leads to poor performance. (ii) Usage of the attention layer, dropout layers, batch normalization layers tends to result in slight performance improvement, which may be caused by their regularization effect.

Data Augmentation Since the gesture data collection is time consuming and costly, here we use data augmentation, which can extend the diversity of the training set by applying hand-crafted transformations. However, most of existing data augmentation techniques are designed for image processing or natural language processing. Here we explore data augmentation techniques for time-series signal. We investigate the performance of the following three data augmentation techniques, including Random cropping, Gaussian noise, and mix-up. *Random cropping* is inspired by the random cropping data augmentation in image processing. It randomly samples a continuous sub-sequence of original sequence as the training sample. *Gaussian noise* randomly adds Gaussian noise to the original signal sequence. *Mix-up* indicates generating new training samples and the corresponding labels as the linear interpolation between two existing training samples [31].

The recognition accuracy of GRN models with these data augmentation is shown in Table IV. We gain two observations: (i) In un-seen scenarios, 0.2%, 0.3% accuracy boosts and 6.0% accuracy decrements can be observed when Random cropping, Gaussian noise and Mix-up are individually applied, respectively, indicating the modern data augmentation techniques for images and languages does not work well on time-series signal while the traditional data augmentation techniques generalize better. The experiment results in cross-validation setting also shows the similar trend. (ii) By combining the two beneficial data augmentation techniques together, 0.5% and 0.6% accuracy boosts in un-seen scenarios and cross-validation setting can be obtained, respectively. These results have demonstrated the effectiveness of data augmentation in gesture recognition based on time-series signal.

Training Methodology The training methodology of neural networks play a fundamental role in deep learning, even with the same neural network. Besides training the neural networks with the standard cross entropy loss, we further evaluate the performance of GRN models with *deep mutual learning (DML)* and *Knowledge Distillation (KD)*. DML is proposed by Zhang et al. which trains two GRN models together and additionally minimizes the Kullback-Leibler divergence between their prediction [32]. KD is proposed by Hinton et al. which firstly pre-train a GRN model as the teacher, and then train a student GRN model to minimize the Kullback-Leibler divergence with the teacher [33].

Our experimental results demonstrate that GRNs with the DML and KD achieves 93.6% and 93.7% accuracy respectively, which outperforms the standard training method by 0.1% and 0.2% accuracy, respectively.

To sum up, by combining the optimal choice of network architectures, data augmentation, and training methodology, our best GRN model achieves 94.2% recognition accuracy, without using inference-time inefficient methods such as model ensemble. Besides, Table V shows the accuracy of all kinds of neural networks with more layers and channels, data augmentations, and knowledge distillation training. Experimental results show that these techniques effectively improve the accuracy of all the neural networks. Besides, GRN still achieves the highest accuracy in both un-seen scenarios and cross-validation.

C. Subjective Perception

We conduct a user study to evaluate the subjective perception quality of the screen display. We invited 15 volunteers (5 female and 10 male) in the age range of 18 to 28 to evaluate 6 different images. The images belong to various types, including natural scenery, sport, food, and building. They can be classified according to the range of ΔY: (i)ΔY of cliff and vegetables belongs to $(20, 40)$; (ii)ΔY of coffee house and grassland belongs to $(40, 60)$; (iii)ΔY of a football field and tennis court belongs to $(60, 80)$. Each image is shown to the participants in 4 versions. The first version is the original static image without any modification; the second version is dynamically switching between two complementary frames as we designed in Section IV (without color calibration method and smoothing); the third version adds the edge smoothing scheme (Section IV-D) to the second version; the fourth version adds the color calibration method to the third version. We showed the four versions of each image one by one to each volunteer and ask them to rate the images based on three aspects: image color difference, continuous flicker, and visual fatigue. To be specific, we use scores 1 to 4 to indicate the degree for the three aspects. Table II shows the meaning for each score. We treat 1 and 2 as satisfactory scores, which means acceptable viewing experience.

Figure 21 shows the average scores rated by the volunteers for different versions. We get three conclusions. First, from the score result of version 4, we conclude that the visual effect of *SMART* transmitter is acceptable since the average scores for the three aspects are all between 1 and 2. Second, comparing version 2 and version 3, we notice that the visual effect of version 3 is much better than version 2 on the aspect of "flicker" and "visual fatigue". Without the smoothing scheme, the screen flickering is mostly evident, causing obvious or even strong visual fatigue. After edge smoothing, the scores improve dramatically. Third, comparing version 3 and version 4, we notice that the visual effect of version 4 is much better than version 3 on the aspect of "color difference". Our conclusion is that the color calibration method can dramatically mitigate color distortion.

D. Processing Latency

To evaluate *SMART* 's ability to support real-time display, we deployed the frame processing algorithm on both the Android and iOS platform. We run *SMART* transmitter on 5 Android devices (Xiaomi MI9 Pro, Samsung A90, Samsung Galaxy S10, ZTE AXON10Pro), and 2 iOS devices (iPhone 11Pro, iPhone XS) and measure the processing time for each frame.

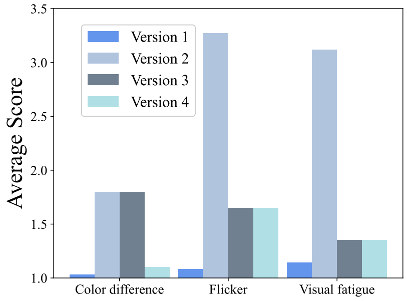


Fig. 21. Perception scores of four image versions.

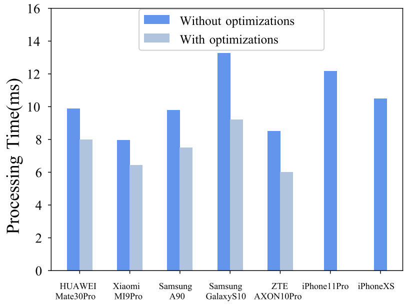


Fig. 22. *SMART* 's transmitter processing latency per frame on different commercial devices.

We test 10 1080p images and 2 videos on different devices. Each image/video is tested on each device 10 times. We perform some simple optimizations to reduce the computation load, including both spatial domain and time domain: (i) Spatial domain: if a block in the frame is of single color (same RGB values), *SMART* does the processing (Section IV-C) only once; (ii) Time domain: if pixels in a frame share the same color with the previous frame, *SMART* reuses the results from the previous frame.

The average result of the processing time for each device is calculated and shown in Figure 22. We can observe that the average processing time of different devices is 6-9ms after optimizations. Thus, it is possible for each frame to be processed and rendered in real time to support 60 FPS dynamic displaying.

E. Comparison With Depth Camera

We compare *SMART* with depth camera in terms of both accuracy and power consumption.

Accuracy. We test the gesture recognition of Huawei Mate 30 Pro, which has a gesture sensor (i.e. a depth camera) on the front panel. As Huawei Mate 30 Pro supports 6 gestures (i.e., "LeftRight", "RightLeft", "TopBottom", "BottomTop", "UpDown", "Fist"), we test each gesture for 30 times in a static light environment. The average accuracy is 93.8%. For *SMART*, the average accuracy for recognizing 9 gestures is 93.0%- 96.9%. Thus, *SMART* has comparable accuracy with the commercial system.

Power consumption. To evaluate the power consumption, we run *SMART* on Huawei Mate 30 Pro. The power consumption of *SMART* comes from two parts: (i) *SMART* transmitter: It mainly refers to the power consumption for frame processing (Section IV-C). The power consumption for screen display is not included in the measurement, as the screen is always on when the smartphone is in use, no matter *SMART* is

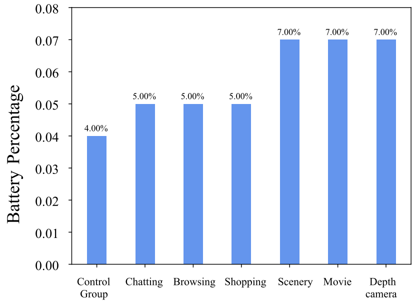


Fig. 23. Power consumption of *SMART* and depth camera.

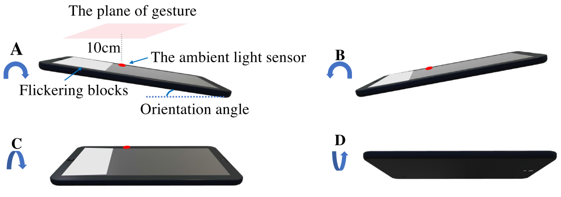


Fig. 24. Four tilted orientations.

running or not. (ii) *SMART* receiver: It mainly refers to the the power consumption for running gesture recognition algorithm. Similar to screen, ALS is always on when the smartphone is in use, and thus we do not include the power consumption of ALS in the measurement.

We tested 5 types of displaying content: online chatting, web browsing, online shopping, playing scenery video and watching movie. We also set a control group (without running the algorithm, but with the screen and ALS on). By taking the difference between the experimental group and the control group, we can measure the power consumption of *SMART*.

For Huawei Mate30 Pro, we use BatteryManager.BATTERY_PROPERTY_CAPACITY [34] for reading the battery percentage. In order for the results to be accurate, we let *SMART* run for 1.5 hours for each test. Each case is repeated for 3 times. The average battery drop of each type of scene is shown in Figure 23. To measure the power consumption of depth-camera, we use the API function CameraManager.open() [35] to keep the depth-camera on for 1.5 hours and examine the battery drain of the mobile phone. We repeat the experiment for 3 times and the battery drop is 7%.

Comparing the power consumption of SMART and depth-camera, we have two observations. First, we found that the power consumption of *SMART* is lower than depth-camera in most cases. It is mostly benefited from the time domain optimization, as a large portion of pixels in subsequent frames share a lot of similarity. Second, we found that the power consumption for more drastic frame transition is higher. The reason is that drastic transition leads to more different pixels between the adjacent frames, which means more pixels in new frames need to be processed.

Jointly considering accuracy and power consumption, *SMART* has comparable gesture recognition performance with depth camera but lower power consumption.

VIII. Discussion

Degradation in the perception scores. In *SMART*, although we make great efforts to reduce the influence to user's viewing experience, there is still a little degradation in perception scores. Despite the little degradation, *SMART* can serve as a low-cost, privacy preserving alternative for depth camera, providing users with a choice to realize gesture control on various commercial off-the-shelf mobile devices. To further minimize the perception degradation, we propose two possible solutions. First, *SMART* can be triggered when detecting hand proximity. This can be achieved with proximity sensor, which is widely available on smartphones. The current use of proximity sensor is to detect face proximity when making phone calls and shut off the screen to avoid unintended operation. Similarly, only when detecting hand proximity, the system can trigger *SMART* to run the screen modulation algorithm, preventing long-term perception degradation. Second, users can choose whether to run *SMART* or not in different application scenarios, so that users can balance between visual perception and convenience. For example, when users are driving, they may put driving safety at the first place and choose in-air gesture control instead of using touch screen or buttons. In this scenario, a little degradation of visual effect is acceptable since driving safety is more important.

View blockage caused by gesture. When the user is watching a video while using gestures to control the speed or volume, the gesture above the screen may block the views of the user. Fortunately, we find that most gestures can be completed within 1s [21], which means that the period of view blockage is mostly shorter than 1s. Thus, when the user is watching a video, the blockage time and range could be accepted.

Switched-off screen. In-air gesture recognition is still required in some scenarios when the screen is off, such as using gestures to wake up the phone or control the volume of music when it is played background. Fortunately, *SMART* can recognize several gestures without light signals from screen, since the ambient light change (light signals with low frequencies) received by ambient light sensor can provide partial gesture features. We evaluate the recognition accuracy of four gestures (including “Fist”, “Openhand”, “UpDown” and “DownUp”) without light signal from screen. We exploit the original evaluation dataset of the four gestures, but only use ambient light (E_0) as the feature. The average accuracy is 95.25%. Thus, we can realize gesture recognition for a few types of gestures. These gestures can be used in various scenarios without the screen on.

IX. Related Work

Device-free in-air hand gesture recognition. Most existing in-air gesture recognition systems use customized hardware. In industry, some companies start to manufacture mobile devices supporting gesture recognition such as Huawei [10], LG [11], Google [8], etc. However, most of them need customized hardware such as radar and depth camera. Soli [9] is a gesture recognition system developed by Google based on 60GHz wireless signal with mm-level wavelength. Leap Motion [2] uses infrared cameras to sense hand gestures.

In academia, different sensing media are used to sense human hand gestures. Cameras are widely used in the field of gesture recognition [36], [37], [38]. However, such systems may cause privacy problems and heavy computation overhead. Since WiFi signals are ubiquitous in our daily environment, some prior works have also studied the use of WiFi to sense hand gestures [4], [6], [7], [39], [40], [41]. They can be easily affected by electromagnetic interference and can not be used in some RF-inappropriate environment, like hospital, underground mines and gas station. Besides, most of them are based on the measurement of CSI to gain accurate gesture recognition. However, only specific Wi-Fi NIC models provide CSI information, while the majority of mobile devices do not provide such information. WiGest [6] uses RSS information which can be achieved on commercial devices. However, it needs a special preamble gesture for gesture detection. Acoustic signal is commonly used to recognize in-air hand gestures [3], [5], [42], [43]. They also have privacy problems and can be effected by ambient sound interference. LLAP [3] is a device-free gesture tracking system that can be deployed on mobile devices. However, it can be interfered by nearby moving objects and other devices deployed with LLAP. Light signal is also used to realize device-free gesture recognition. SMART [44] is a screen-based gesture recognition system on commodity mobile devices.

Visible light based human gesture sensing. Existing studies have explored various gesture sensing modalities based on visible light [21], [45], [46], [47], [48], [49], [50]. Okuli [45] is proposed to realize fine-grained finger tracking with a LED and two light sensors. Some human sensing systems [46], [47], [48] based on visible light can reconstruct human gestures. Specifically, LiSense [46] can reconstruct human skeleton postures using several LEDs and photodiodes embedded in the floor and StarLight [47] can gain a more fine-grained sensing posture. Aili [48] can reconstruct hand poses with a table lamp with an LED panel and an array of photodiodes. Some works propose visible light human gesture recognition methods. LiGest [49] uses a grid of light sensors deployed on the floor to build an ambient light based gesture recognition system. Reference [50] develops a gesture recognition system using small, low-cost photodiodes for both energy harvesting and sensing. SolarGest [21] is designed to recognize hand gestures near a solar-powered device. However, they can not be used on mobile devices directly since they all need customized devices or previous deployment in environment.

Hidden Screen-Camera Communication. Prior work [19], [20], [51], [52], [53], [54] utilizes the gap of perception ability between human eyes and cameras to hide unobtrusive information in a given screen content while realizing the screen-camera communication. InFrame++ [19] convert barcodes into complementary frames to enable screen-camera communication. They leverage flicker fusion property of HVS to keep the visual effect of switching complementary frames seems like the original content. HiLight [51] encodes data into translucency change and enables any-scene communication. Since HiLight change the translucency instead of the RGB value of each pixel, it realizes real screen-camera communication. Chromacode [20] excels in its adaptive embedding in uniform color space and realizes imperceptible, high rate, and reliable communication. AIRCODE [55] is a hidden screen-camera communication system built on visual and audio dual channel. It ensures unobtrusiveness of users and achieves robust communication with high rate. Our techniques are relevant to screen-camera communication methods, but also different from them. While existing methods realize hidden screen-camera communication, we try to realize the unobtrusive communication between screen, hand and light sensor. Unlike cameras collecting complete vision information, ambient light sensor can only sample light intensity, which makes unobtrusive communication non-trival.

X. Conclusion

In this paper, we proposed *SMART*, a screen-based gesture recognition scheme, which enables in-air gesture control on legacy mobile devices. *SMART* exploits the "Screen-Hand-ALS" light path to recognize the in-air gesture. We have implemented and evaluated *SMART* on both a tablet and several smartphones. Results show that *SMART* can recognize 9 frequently used gestures with 96.1% accuracy, meanwhile it preserves the viewing experience of the screen. We expect that *SMART* provides the legacy device with the same in-air gesture control capability as the flagship smartphones.

Acknowledgment

The opinions, findings, conclusions, and recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

References

[1] Markets and Markets, *Gesture Recognition and Touchless Sensing Market—Global Forecast to 2025*, MarketsandMarkets Res. Private Ltd., Pune, India, 2020.

[2] (2021). *Leap Motion*. [Online]. Available: https://www.leapmotion.com/

[3] W. Wang, A. X. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2016, pp. 82–94.

[4] N. Yu, W. Wang, A. Liu, and L. Kong, "QGesture: Quantifying gesture distance and direction with WiFi signals," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–23, 2018.

[5] S. Yun, Y. C. Chen, H. Zheng, L. Qiu, and W. Mao, "Strata: Fine-grained acoustic-based device-free tracking," in *Proc. Int. Conf. Mobile Syst.*, 2017, pp. 15–28.

[6] H. Abdelnasser, M. Youssef, and K. A. Harras, "WiGest: A ubiquitous WiFi-based gesture recognition system," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 1472–1480.

[7] L. Sun, K. H. Kim, S. Sen, and D. Koutsonikolas, "WiDraw: Enabling hands-free drawing in the air on commodity WiFi devices," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 77–89.

[8] (2021). *Google Pixel 4*. [Online]. Available: https://store.google.com/product/pixel_4a

[9] (2021). *Google Project Soli*. [Online]. Available: https://www.google.com/atap/project-soli/

[10] (2020). *Huawei Mate30*. [Online]. Available: https://consumer.huawei.com/en/phones/mate30/

[11] (2021). *LG G8 Thinq*. [Online]. Available: https://consumer.huawei.com/en/phones/mate30/

[12] C. Zhang and X. Zhang, "Pulsar: Towards ubiquitous visible light localization," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2017, pp. 208–221.

[13] L. Yang, Z. Wang, W. Wang, and Q. Zhang, "NALoc: Nonlinear ambient-light-sensor-based localization system," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 4, pp. 1–22, Dec. 2018.

[14] L. Li, P. Hu, C. Peng, J. Shen, and A. F. Zhao, "Epsilon: A visible light based positioning system," in *Proc. USENIX Assoc.*, 2014, pp. 331–343.

[15] Z. Wang, Z. Yang, Q. Huang, L. Yang, and Q. Zhang, "ALS-P: Light weight visible light positioning via ambient light sensor," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 1306–1314.

[16] J. M. Kahn and J. R. Barry, "Wireless infrared communications," *Proc. IEEE*, vol. 85, no. 2, pp. 265–298, Feb. 1997.

[17] Z. Lan et al., "Kaleido: You can watch it but cannot record it," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 372–385.

[18] Z. Yang, J. Zhang, Z. Wang, and Q. Zhang, "Lightweight display-to-device communication using electromagnetic radiation and FM radio," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–19, Mar. 2018.

[19] A. Wang, Z. Li, C. Peng, G. Shen, G. Fang, and B. Zeng, "InFrame++: Achieve simultaneous screen-human viewing and hidden screen-camera communication," in *Proc. 13th Annu. Int. Conf. Mobile Syst., Appl., Services*, May 2015, pp. 181–195.

[20] K. Zhang et al., "ChromaCode: A fully imperceptible screen-camera communication system," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2018, pp. 575–590.

[21] D. Ma et al., "SolarGest: Ubiquitous and battery-free gesture recognition using solar cells," in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, Aug. 2019, pp. 1–15.

[22] (1931). *Cie 1931 Color Space*. [Online]. Available: https://en.wikipedia.org/wiki/CIE_1931_color_space

[23] C. Zhang and X. Zhang, "LiTell: Robust indoor localization using unmodified light fixtures," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2016, pp. 230–242.

[24] (2018). *Smartphone Unit Shipments Worldwide By Screen Size From 2018 to 2022*. [Online]. Available: https://www.statista.com/statistics/684294/global-smartphone-shipments-by-screen-size/

[25] (2021). *TSL2740 Ambient Light Sensor*. [Online]. Available: https://ams.com/tsl2740

[26] (2021). *TSL2540 Ambient Light Sensor*. [Online]. Available: https://ams.com/zh/tsl2540

[27] (2021). *APDS-9253–001*. [Online]. Available: https://ams.com/zh/tsl2540

[28] (2023). *Digital Eye Strain: How Much Time Should You Spend on Your Phone?* [Online]. Available: https://safe365.com/blog/en/digital-eye-strain-how-much-time-should-you-spend-on-your-phone/

[29] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.

[31] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.

[32] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proc. CVPR*, 2018, pp. 4320–4328.

[33] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS*, 2014, pp. 1–9.

[34] (2021). *Measuring Device Power*. [Online]. Available: https://source.android.com/devices/tech/power/device

[35] (2021). *Camera2*. [Online]. Available: https://developer.android.com/training/camera2

[36] R. Y. Wang, S. Paris, and J. Popovic, "6D hands: Markerless hand tracking for computer aided design," in *Proc. ACM Symp. User Interface Softw. Technol.*, 2011, pp. 549–558.

[37] T. Sharp et al., "Accurate, robust, and flexible real-time hand tracking," in *Proc. 33rd Annu. ACM Conf. Human Factors Comput. Syst.*, Apr. 2015, pp. 3633–3642.

[38] S. Izadi et al., "KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. 24th Annu. ACM Symp. User Interface Softw. Technol.*, Oct. 2011, pp. 559–568.

[39] B. Kellogg, V. Talla, and S. Gollakota, "Bringing gesture recognition to all devices," in *Proc. 11th Usenix Conf. Networked Syst. Design Implement.*, 2014, pp. 303–316.

[40] A. Virmani and M. Shahzad, "Position and orientation agnostic gesture recognition using WiFi," in *Proc. 15th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2017, pp. 252–264.

[41] Y. Zheng et al., "Zero-effort cross-domain gesture recognition with Wi-Fi," in *Proc. 17th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2019, pp. 313–325.

[42] S. Gupta, D. Morris, S. Patel, and D. Tan, "SoundWave: Using the Doppler effect to sense gestures," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2012, pp. 1911–1914.

[43] W. Ruan, Q. Z. Sheng, L. Yang, T. Gu, P. Xu, and L. Shangguan, "AudioGest: Enabling fine-grained hand gesture detection by decoding echo signal," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Sep. 2016, pp. 474–485.

[44] Z. Liao et al., "SMART: Screen-based gesture recognition on commodity mobile devices," in *Proc. 27th Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2021, pp. 283–295.

[45] C. Zhang, J. Tabor, J. Zhang, and X. Zhang, "Extending mobile interaction through near-field visible light sensing," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 345–357.

[46] T. Li, C. An, Z. Tian, A. T. Campbell, and X. Zhou, "Human sensing using visible light communication," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2015, pp. 331–344.

[47] T. Li, Q. Liu, and X. Zhou, "Practical human sensing in the light," *GetMobile: Mobile Comput. Commun.*, vol. 20, no. 4, pp. 28–33, Apr. 2017.

[48] T. Li, X. Xiong, Y. Xie, G. Hito, X.-D. Yang, and X. Zhou, "Reconstructing hand poses using visible light," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 1–20, Sep. 2017.

[49] R. H. Venkatnarayan and M. Shahzad, "Gesture recognition using ambient light," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–28, Mar. 2018.

[50] Y. Li, T. Li, R. A. Patel, X.-D. Yang, and X. Zhou, "Self-powered gesture recognition with ambient light," in *Proc. 31st Annu. ACM Symp. User Interface Softw. Technol.*, Oct. 2018, pp. 595–608.

[51] T. Li, C. An, X. Xiao, A. T. Campbell, and X. Zhou, "Real-time screen-camera communication behind any scene," in *Proc. 13th Annu. Int. Conf. Mobile Syst., Appl., Services*, May 2015, pp. 197–211.

[52] V. Nguyen et al., "High-rate flicker-free screen-camera communication with spatially adaptive embedding," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[53] M. Izz, Z. Li, H. Liu, Y. Chen, and F. Li, "Uber-in-light: Unobtrusive visible light communication leveraging complementary color channel," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.

[54] D. Wan, J. C. Liando, and L. Mo, "Softlight: Adaptive visible light communication over screen-camera links," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[55] K. Qian et al., "AIRCODE: Hidden screen-camera communication on an invisible and inaudible dual channel," in *Proc. 18th USENIX Symp. Networked Syst. Design Implement.*, 2021, pp. 457–470.