When VLC Meets Under-Screen Camera

Ye, Hanting; Xiong, Jie; Wang, Qing

DOI
10.1145/3581791.3596855

Publication date
2023

Document Version
Final published version

Published in
MobiSys 2023 - Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services

Citation (APA)
Ye, H., Xiong, J., & Wang, Q. (2023). When VLC Meets Under-Screen Camera. In *MobiSys 2023 - Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services* (pp. 343-355). (MobiSys 2023 - Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services). ACM. https://doi.org/10.1145/3581791.3596855

When VLC Meets Under-Screen Camera

Hanting Ye
Delft University of Technology
Delft, The Netherlands
h.ye-1@tudelft.nl

Jie Xiong
University of Massachusetts Amherst
Amherst, United States
jxiong@cs.umass.edu

Qing Wang
Delft University of Technology
Delft, The Netherlands
qing.wang@tudelft.nl

ABSTRACT

While radio communication still dominates in 5G, light and radios are expected to complement each other in the coming 6G networks. Visible Light Communication (VLC) is therefore attracting a tremendous amount of attention from both academia and industry. Recent studies showed that the front camera of pervasive smartphones is an ideal candidate to serve as the VLC receiver. While promising, we observe a recent trend with smartphones that can greatly hinder the adoption of smartphones for VLC, i.e., smartphones are moving towards **full-screen** for the best user experience. This trend forces front cameras to be placed *under* the devices' screen—leading to the so-called **Under-Screen Camera (USC)**—but we observe a severe performance degradation in VLC with USC: the transmission range is reduced from a few meters to merely 0.04 m, and the throughput is decreased by more than 90%. To address this issue, we leverage the unique spatiotemporal characteristics of the rolling shutter effect on USC to design a pixel-sweeping algorithm to identify the sampling points with minimal interference from the translucent screen. We further propose a novel slope-boosting demodulation method to deal with color shift brought by the leakage interference. We build a proof-of-concept prototype using two commercial smartphones. Experiment results show that our proposed design reduces the BER by two orders of magnitude on average and improves the data rate by 59×: from 914 b/s to 54.43 kb/s. The transmission range is extended by roughly 100×: from 0.04 m to 4.2 m.

CCS CONCEPTS

• **Networks** → **Wireless access networks**; **Mobile networks**; • **Human-centered computing** → **Ubiquitous computing**; **Smartphones**; **Mobile devices**.

KEYWORDS

Through-Screen VLC, Full-Screen, Translucent Screen, Under-Screen Camera, Optical Camera Communication, Color-Shift Keying

ACM Reference Format:
Hanting Ye, Jie Xiong, and Qing Wang. 2023. When VLC Meets Under-Screen Camera . In *The 21st Annual International Conference on Mobile Systems, Applications and Services (MobiSys '23), June 18–22, 2023, Helsinki, Finland*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3581791.3596855

MobiSys '23, June 18–22, 2023, Helsinki, Finland

ACM ISBN 979-8-4007-0110-8/23/06.
https://doi.org/10.1145/3581791.3596855

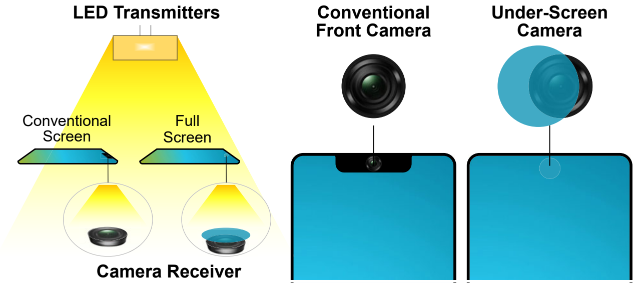


Figure 1: Illustration of the proposed through-screen VLC with Under-Screen Cameras (USC). USC makes VLC on full-screen devices challenging.

1 INTRODUCTION

While radio communication still dominates in 4G/5G, light and radios are expected to complement each other in 6G networks. The key advantage of light-based communication is that it mitigates the severe issue of spectrum shortage, and its wide frequency band can be utilized to support a high data rate [14, 15]. Visible Light Communication (VLC) is therefore considered a key component for future 6G networks [11], attracting attention from both academia and industry. The protocols related to VLC have been standardized by IEEE 802.11 and 802.15 Task Groups [1, 22], and by the International Telecommunication Union (ITU) in G.9991 [24].

While the first-generation VLC systems mainly employ LED as the transmitter and photodiode as the receiver, recently, cameras widely available on commodity smartphones are utilized as the receiver for VLC. So in terms of hardware, both VLC transmitter (i.e., LED) and receiver (i.e., smartphone) are already pervasive in our everyday life without incurring any extra hardware burden. Most smartphones nowadays have two cameras, i.e., a front camera and a back camera. Between them, the front camera is considered a better option to serve as the VLC receiver because during our routine use of the smartphone, the front camera faces the LEDs in our surrounding environment most of the time. The communication modality of LED-to-front-camera can be a good supplement to existing wireless technologies. Several LED-to-front-camera applications, such as museum narration, have been proposed in recent works [21, 27, 29, 57, 60, 61]. Another exciting application is to transfer sensitive data which needs to be confined within an area of interest such as a room.

While promising, the growing interest in maximizing the screen-to-body ratio of consumer devices, such as smartphones and laptops, has led to the evolution of full-screen designs in recent years [39]. These designs significantly reduce the bezel area occupied by the front camera and other sensors, making them attractive options for a wide range of flagship laptops (e.g., Thunderobot T-BOOK

14), smartphones (e.g., ZTE AXON 20/30/40, Xiaomi MIX4, and Samsung Galaxy Z Fold 3/4), and even TVs [13, 23]. However, this trend poses challenges for the adoption of the front camera in mobile devices as the VLC receiver, as it causes the placement of the front camera under the screen, resulting in the so-called Under-Screen Camera (USC).[1] Although the under-screen camera design has little impact on photo and video quality due to a dedicated translucent screen layer, it severely degrades the performance of VLC: the transmission range is reduced from a few meters to merely 0.04 m, and the data rate is decreased by more than 90%.

We dig deep to identify the causes of this performance degradation. The first cause is signal attenuation brought in by the translucent screen. The second cause, which is more severe, is the interference induced by the translucent screen. This translucent part is still part of the screen, and therefore, the dynamically changing screen content can severely affect the VLC performance. In this paper, we ask the following research question: *Can we utilize the camera under the screen for VLC without incurring a performance degradation?*

To make VLC work with under-screen cameras, we must deal with the two causes of performance degradation. Note that the first issue is also the main challenge for the original function of camera, i.e., image and video taking. Therefore, a tremendous amount of effort has been devoted to addressing this issue. The state-of-the-art solution uses a small piece of translucent screen on top of the camera. However, even with the state-of-the-art solutions, we still observe a significant decrease in signal amplitude [30, 48, 67].

To address this issue, we propose to use Color Shift Modulation (CSK) instead of conventional ON-OFF modulation. Unlike ON-OFF-based modulation, color-based CSK modulation utilizes the color positions, which are much less sensitive to light intensity. Also, by utilizing color modulation, more bits can be encoded to achieve a higher data rate.

To address the second issue, we need to tackle multiple challenges. As cameras on commodity devices such as smartphones usually have a low image rate, i.e., less than 200 images per second, to achieve a high data rate, a unique property of camera imaging, i.e., the rolling shutter effect [6, 16] is utilized. The basic idea is that within one single image, all the pixels are not captured simultaneously but are recorded row by row at different timestamps. We can therefore zoom in to extract the color of light at the granularity of row level. When there is no screen on top of the camera, the light source can be easily identified on the image as the brightness of the light source area is much higher compared to other areas. When the camera is under the screen, a lot of other areas also have high brightness as the screen itself is also a light source. For accurate demodulation, we need to make sure we are extracting the color information of LED light but not other areas on the image. Therefore, the first challenge is *how to accurately identify the light source, i.e., the LED in the captured images in the presence of strong interference?*

To address this challenge, we propose a color compensation method leveraging an observation of the color composition of the translucent screen: when CSK-modulated images are converted into grayscale for detecting the VLC transmitter, the grayscale values of the red and blue components are lower than that of the green component. We thus increase the brightness of the blue and the red components at captured image, which helps us detect the VLC transmitter area in the captured image of under-screen camera. We further apply the vertical averaging scheme to distribute the same brightness to all three colors. This vertical average scheme also strengthens the low-frequency data while weakening the high-frequency noise. The VLC transmitter has the same color for pixels in a row, while the other area has random colors for pixels in a row. After this step, the VLC transmitter area is further highlighted. We can then apply a median filter to remove the background and keep the identified VLC transmitter for further processing.

Although the LED (transmitter) area is detected, the image still contains interference from the translucent screen. The second challenge is *how to remove the interference caused by the translucent screen?* We propose two novel steps to address the interference. The first step is based on the key fact that the translucent screen (the piece of glass) on top of the camera has a unique design which is very different from ordinary screens. While ordinary screen has a much larger pixel area for each pixel point shown in Figure 4, the translucent screen has a much small pixel area in order to let more light go through the screen. Because of this design, the image taken with an under-screen camera has two groups of area, i.e., screen-pixel-interference area and non-screen-pixel-interference area. We thus identify the non-screen-pixel-interference area for more accurate demodulation. The good news is that the translucent screen design for each manufacturer is fixed and we just need to obtain the pixel layout design once for each model of smartphone.

The second step is to identify the optimal pixel point for demodulation. This is because due to the slow speed of camera, the exposure duration can be long enough to have two different colors from LED. Note that the camera does not show both colors (one color in the first part and the other color in the second part) but just shows a mixed color for the row.[2] We, therefore, need to identify the optimal pixel point without color mixing for demodulation. To solve this problem, we propose a vertical scanning method. The basic idea is to model the color information as a transition graph. While mixed colors are on the transition links, the pure colors are the transition nodes. We can thus identify those nodes with an obvious direction change for the best demodulation performance.

After we apply the methods described above to identify those non-screen-pixel-interference area, we still find some interference leaked from those pixel-interference area and this leakage does affect the performance. The leakage can cause the color to vary. So the third challenge we encounter is *how to address the leakage interference to improve the performance.*

To solve this challenge, we propose a series of methods. First, we observe that the color change due to the leakage is not random. For example, if a red color is interfered by blue color leakage, then the red color will be shifted to a different color between red and blue. We can therefore use a line to connect these two colors and the color only changes along the line. This property enables us to recover the original color we transmitted even the color is shifted. A more challenging scenario is that a random screen color leakage can be on the line connecting two colors used for communication. In

[1] It is also referred as Under-Display Camera (UDC) in the literature [28, 42, 45, 67].

[2] The camera can be understood as a sensor that integrates incoming light signals within a specified exposure time window, resulting in a mixed color.

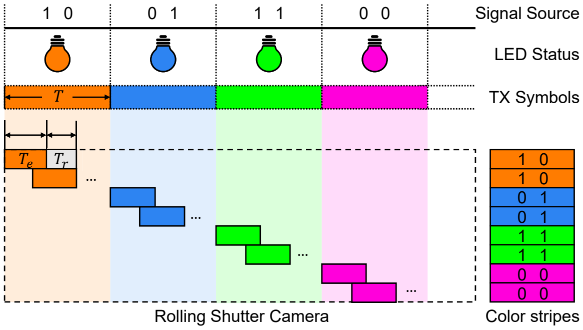


Figure 2: Rolling-shutter effect principle.

this case, the two colors change following the same line, confusing the method above to recover the original color. Interestingly, we find that in this case, while the colors' absolute values (locations) are not accurate, the relative information between the two colors is still reserved, i.e., if one shifted color is further away from the interfering color, the original color is also further away from the interfering color. We can thus utilize the relative information for demodulation.

With the design components, we successfully implement a proof-of-concept system for the proposed through-screen VLC system with under-screen camera using off-the-shelf smartphones. Our system is able to achieve low BER and long transmission range with under-screen camera for the first time. Below we summarize our contributions.

- We propose the concept of VLC with under-screen camera for emerging full-screen devices. We analyze the unique spatiotemporal characteristics of the rolling shutter effect on under-screen camera. We design a pixel-sweeping algorithm based on the spatial characteristics of the image/frame captured by the under-screen camera and identify the sampling points with the least amount of interference for communication. We further propose a novel slope-boosting demodulation method to deal with color shift brought by the leakage interference.
- We build a proof-of-concept testbed and thoroughly evaluate the system performance in different scenarios. The results show that our system can achieve a maximum throughput of 54.43 kb/s with four LED transmitters and one under-screen camera receiver. Compared to state-of-the-art solutions, our methods can reduce the BER by two orders of magnitude on average, and improve the throughput by 59× (914 b/s vs. 54.43 kb/s). The transmission range is extended from 0.04 m to 4.2 m, with an improvement of roughly 100×.

2 BACKGROUND

Rolling-Shutter effect. The rolling-shutter effect, which is a fundamental property of CMOS sensors commonly used in smartphone cameras, allows exposure to be performed on a per-row basis [6, 16]. A rolling-shutter camera controls the exposure row-by-row (or column-by-column), as shown in Figure 2. The first row is exposed for T_e and the second row starts being exposed after T_r. T_r is a measure of the time required to capture and transfer image data from the camera to processor, which is fixed for each camera. Note that T_r is usually smaller than T_e and therefore to fully utilize the channel, the adjacent exposures are overlapped. In the traditional rolling-shutter (without the effect of screen), all pixels in a row contain the *same modulated information*. The strip width (w) is defined as the number of pixel rows occupied by the same color strip in a captured image. Therefore, the width of the stripe is determined by $w = T/T_r$, where T donates the symbol period. Thus, the higher the frequency of the transmitted symbol, the narrower the stripes (i.e., the less number of rows) with the same color will be in the image.

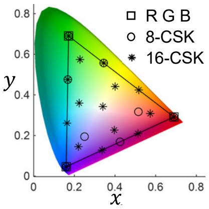


Figure 3: CSK.

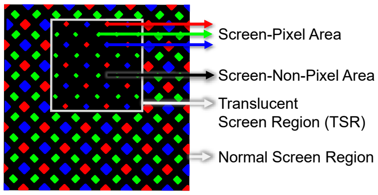


Figure 4: Screen-pixel layout.

Standard Color Space. The CIE 1931 chromaticity diagram, developed by the International Commission on Illumination (CIE), is a popular representation of the color chromaticity perceived by the human eye. As shown in Figure 3, *the diagram maps all colors perceivable by human eyes to two chromaticity parameters - x and y* - with the monochromatic light with different wavelengths forming a line on the periphery of the horseshoe shape.

Color-Shift Keying Modulation. The Color-Shift Keying (CSK) modulation scheme was proposed in the IEEE 802.15.7 standard for VLC. It uses three (Red (R), Green (G), and Blue (B)) separate LEDs to generate white light [63]. By configuring these luminaires to provide a variety of colors using the mixture of R, G, and B LEDs, CSK modulates the signal by modifying the output power combinations of the three colors. A standard color space is used to define the modulation triangle formed by the operating frequencies of the red, green, and blue LEDs, as shown in Figure 3. The constellation symbols for 8-CSK and 16-CSK calculated directly from the specification [63] are marked in Figure 3.

Color-Pulling effect. This phenomenon was reported in a recent work [62]: the transmitted CSK symbols are pulled closer to the display color when the screen is illuminated, as shown later in Figure 6(a). The color-pulling effect has important implications for CSK-based VLC systems, as it can potentially affect the color accuracy of transmitted symbols and the overall demodulation performance of the system.

3 SYSTEM OVERVIEW

3.1 Architecture

Transmitter. We adopt the standard Color-Shift Keying (CSK) modulation [63] at the receiver. The CSK constellation symbols are chosen inside a *modulation triangle* that is formed within the standard color space, as shown in Figure 3, where 8-CSK and 16-CSK constellation symbols are marked. To transmit CSK symbols, the transmitter modulates the signal by varying the output power combinations of the three channels R/G/B of the LED.

Receiver. We consider a **full-screen** device (e.g., a smartphone) as the receiver. It does not have any bezels on its screen to host line-of-sight sensors such as camera and ambient light sensor. Instead, full-screen devices introduce a special region on the screen, i.e., a small **Translucent Screen Region (TSR)**. This region is composed of translucent substrates and cathodes and is usually placed at the top section of the screen. The camera is placed directly under TSR, leading to the so-called **Under-Screen Camera (USC)** [5, 36, 44, 68]. The TSR *is part of the screen and can be lit up to display various contents.* Compared to other screen regions, the translucent screen region utilizes a translucent cathode material, and the pixel layout is also redesigned. A common TSR's structure used by commercial translucent OLED screens is shown in Figure 4. The TSR can be further divided into two parts:

- **Screen-pixel area**: It hosts specially designed R/G/B pixels that can be lit up to display various contents. A large screen-pixel area benefits the displaying function of TSR but sacrifices the imaging quality of under-screen camera as well as the performance of through-screen VLC.
- **Screen-non-pixel area** (i.e., the black area in Figure 4): It allows visible light to pass through – but still at the cost of strong attenuation – to reach the under-screen camera. A large screen-non-pixel area brings more light to the under-screen camera, which benefits the under-screen camera's imaging quality and the performance of through-screen VLC, but reduces the TSR's displaying performance.

Industrial effort has been devoted to optimize the pixel layout to balance the performance of screen displaying and imaging quality [2, 53]. The density and size of the screen-pixels in the TSR have been carefully adjusted to increase the proportion of screen-non-pixel area in TSR, as shown in Figure 4. However, no matter how the TSR is designed, it still requires the screen-pixel area to display content, and the screen-non-pixel area cannot be made fully transparent. It thus still has large impacts on through-screen VLC. Next, we present the major impacts.

3.2 TSR's Impact on USC Communication

As a passive 'blocker' (The corresponding solution will be presented in Section 4.1). The TSR blocks a significant portion of the incident light. The amount of light that can travel through the translucent TSR is only 2.9% [67]. This discourages us adopting intensity-based modulations widely used in state-of-the-art VLC [47, 55, 58]. We leverage CSK modulation to mitigate this impact since the demodulation of color-based CSK signals does not depend much on signal strength. Instead, it relies on mapping the CSK signals to the CIE color space. However, the signal strength attenuation still makes it difficult to detect the position of the VLC transmitter (i.e., region of interest) on the image captured by the under-screen camera.

As an active interferer. The pixels in the TSR are light sources, inducing diffractive blur and color shifts on under-screen camera's captured images. This interference can significantly affect the whole image and lead to a low Signal-to-Interference-plus-Noise Ratio (SINR) of the captured image.

- *Screen-pixel interference (The corresponding solutions will be presented in Sections 4.3 and 4.4).* An under-screen camera, like traditional cameras, is composed of a lens and a CMOS sensor.

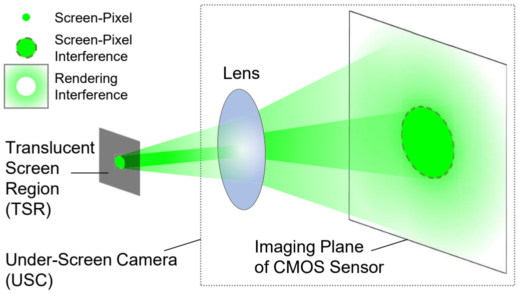

Screen-Pixel
Screen-Pixel Interference
Rendering Interference
Lens
Translucent Screen Region (TSR)
Under-Screen Camera (USC)
Imaging Plane of CMOS Sensor

Figure 5: Illustration of screen-pixel and rendering interferences, using Green pixel as an example.

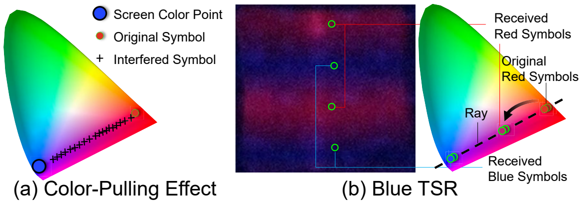


Figure 6: Color-pulling effect caused by the rendering.

  Figure 5 is a micrograph illustration of a *Green* pixel in the TSR and its mapping to the imaging plane of the under-screen camera. This illustration is based on the principle of pinhole imaging [18]. The *dashed circle* illustrates the interference area on the camera's imaging plane, where the interference comes from the TSR's Green pixel because this area is exactly under the Green pixel. We term it as *screen-pixel interference*, which maintains the original screen-pixel shape but is larger in size on the imaging plane. Note that the area of this screen-pixel interference caused by the TSR's R/G/B pixels is different, which depends on the color and size of the pixels in the TSR.
- *Rendering interference.* On the camera's imaging plane, there is also leakage from the TSR's illuminating pixel, causing *rendering interference* as denoted by the *solid square* in Figure 5. This rendering interference is induced by the diffusion effect of the screen-pixel. Compared to the screen-pixel interference, the strength of the rendering interference is weaker, but the interference area on the camera's imaging plane is larger.

This rendering interference leads to the so-called *color-pulling effect (cf. Section 2)* on the received CSK symbols. When the screen is lit up, the original color of a CSK symbol will be pulled closer to the corresponding TSR's pixel color, as shown in Figure 6(a). To give an example, let us consider when we send the red and blue symbols alternately, and the TSR is displaying blue. We pick some sampling points from the area (on the camera's imaging plane) that is affected by rendering interference, and show their color coordinates in CIE1931 in Figure 6(b). We can see the location of blue light is not affected when the screen color is blue. However, the received red coordinates are pulled closer to the blue screen from the original red coordinate, leading to decoding errors. *We will present our solution to this problem in Section 5.*

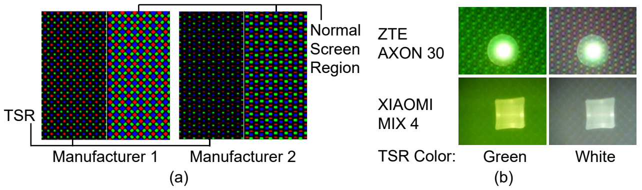


Figure 7: Screen diversity: (a) The comparison of translucent screen region and normal screen region for different manufacturers. (b) The interference of different pixel layouts on images captured by the under-screen camera.

Screen diversity. The TSR designs on different smartphones are not the same, bringing different levels of impact on through-screen VLC. Due to patent protection, different manufacturers adopt various pixel designs and arrangements in the TSR, as shown in Figure 7(a). These unique TSR designs bring different interference to the images captured by under-screen camera, as shown in Figure 7(b). *In Section 4.3 (Steps 1 & 2), we will present a solution that can work with different TSR designs.*

Pipeline of our solution. Next, we present how to address these challenges. First, we propose a method to extract the Region of Interest from the captured images (Section 4.1). Then, we design a pixel-sweeping algorithm, including a horizontal scanning module to address screen diversity issue and remove screen-pixel interference (Section 4.3) and a vertical scanning module to deal with CSK's inter-symbol interference (Section 4.4). Finally, we propose a slope-boosting algorithm to demodulate the CSK symbols by removing the color shift caused by rendering interference (Section 5).

4 PIXEL-SWEEPING ALGORITHM

4.1 Region of Interest (RoI) Extraction

To decode data, the prerequisite is to detect the transmitter's centroid and size – the *Region of Interest (RoI)* that contains transmitted information – on captured images. Existing methods such as grayscale threshold-based mask methods [57] and computer vision-empowered methods [27] do not work well on detecting the RoI from the CSK-modulated images captured by under-screen camera. The reasons are as follows: 1) When CSK-modulated images are converted into grayscale, the grayscale values of each color component are different. The grayscale values of the red and blue components are lower than that of the green component. Thus, the mask method does not work in through-screen VLC; 2) The TSR weakens the overall intensity of the RoI and brings extra interference. As a result, the RoI's contour has more jagged edges and can be easily submerged in TSR's display light.

To extract the RoI in through-screen VLC, we propose a new method as shown in Figure 8. First, we compensate the blue and red components. Then, we apply a vertical blur filter on the whole image. The kernel size of the blur filter is at least one stripe width to average the grayscale between all stripes. To cancel TSR's impact, we add median and binary OTSU filters [37] to filter out the pixels of the TSR, allowing us to obtain a smooth contour of the RoI. Finally, we use a shape filter with a circle area of $4\pi N_r/C_r^2$, where N_r is the number of pixels in the area and C_r is the length of the area boundary. When the output of the shape filter is greater than 0.8, the shape of the detected RoI is regarded as a circle.

4.2 Rolling Shutter on USC: Analysis of the Spatiotemporal Characteristics

We continue to detect the positions of CSK symbols within the RoI. We first analyze the spatiotemporal characteristics of the rolling-shutter effect on under-screen camera.

Spatial characteristics. Compared to traditional rolling-shutter effect (cf. Section 2), for image captured by the under-screen camera, which is affected by TSR (Translucet Screen Region), the pixels in different parts of a row are exposed to *different levels of interference* (screen-pixel interference and rendering interference). Figure 9(a) shows that the pixels are subject to different R/G/B interference with different intensities. The main reasons are i) the interference intensities of the screen-pixel interference and the rendering interference are different; ii) the pixel layouts are different due to screen diversity. This spatial characteristic of the sampling points of under-screen camera makes it impossible to arbitrarily pick a pixel from each row to decode the transmitted information.

Temporal characteristics. Due to the uncontrollable read-out duration (T_r), it is difficult to control the exposure duration (T_e) of under-screen camera to synchronize with the transmitted CSK symbols (T). As illustrated in Figure 9(b), this results in inter-symbol interference on both sides of a stripe, caused by the mix of current stripe and the previous/next stripe. Also, due to different R/G/B screen-pixel layout designs of TSR in different smartphones, the total levels of interference induced by screen light on pixels of each row are different. Thus, for the rolling shutter on under-screen camera, *pixels are subject to double interference from the TSR's display light and also from the inter-symbol interference.* Also, this temporal characteristic makes the symbol width and the symbol boundary vague for detection. Therefore, it is infeasible to average the color values of the entire stripe to decode the transmitted CSK symbol represented by a stripe.

Based on the above spatiotemporal characteristics, we design a pixel-sweeping scheme for demodulation including two scanning modules as shown in Figure 11. Next, we present the details.

4.3 Horizontal Scan in the Spatial Domain

To decode information from the stripes, we should first identify in a stripe where to sample a pixel that has less TSR interference in RoI. Compared to pixels in the screen-pixel area, pixels in the screen-non-pixel area are interfered by the same color (e.g., white when all R/G/B pixels are lit up) with less intensity. Note that the light penetration from the screen-pixel area is lower than that from the screen-non-pixel area [56]. The characteristics of TSR require higher penetrated light intensity to facilitate under-screen camera photographing, and thus the ratio of the screen-non-pixel area is increased in TSR. Thus, we can always find a sampling point that is not in the screen-pixel area when we scan each row in the RoI, as detailed below:

Step 1: Obtain the placements of TSR's R/G/B pixels. We need first to know the layout of the pixels in the TSR. The layouts from different manufacturers mainly differ in the following aspects: *location*, *shape*, and *size*. We use a **one-time calibration** to identify the

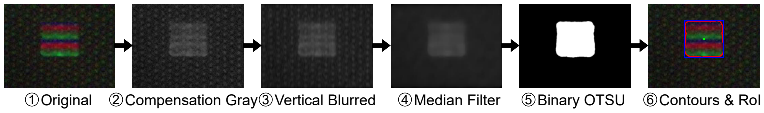


Figure 8: Improved RoI extraction mechanism.

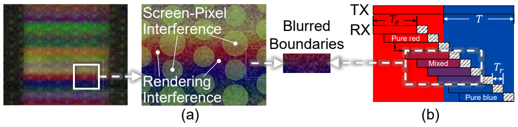


Figure 9: Rolling-shutter effect of USC.

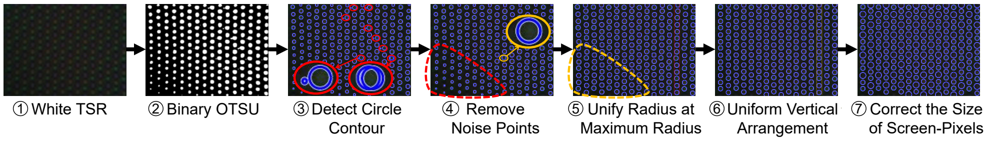

① White TSR
② Binary OTSU
③ Detect Circle Contour
④ Remove Noise Points
⑤ Unify Radius at Maximum Radius
⑥ Uniform Vertical Arrangement
⑦ Correct the Size of Screen-Pixels

Figure 10: Obtain placement of R/G/B screen-pixels mask.

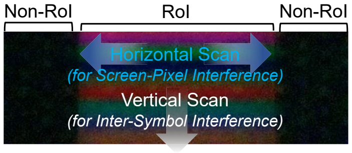

Non-RoI
RoI
Non-RoI
Horizontal Scan
(for Screen-Pixel Interference)
Vertical Scan
(for Inter-Symbol Interference)

Figure 11: Pixel-sweeping workflow.

TSR's R/G/B pixels to handle the TSR diversity challenge: a) For *location*, since a screen must light up R/G/B screen-pixels together to display white color, we first let the TSR display white color and configure a large exposure period on the under-screen camera before communication. Since different color screen-pixels have different luminous efficiency, we use a multi-level OTSU threshold algorithm [32] to obtain the exact position of each screen-pixel for each R/G/B channel. b) For *shape*, we find that the final aperture formed by screen-pixels of different shapes is still approximately circular due to the isotropy of screen-pixel emission. Therefore, we unify the minimum circumcircle of the interference range of each screen-pixel as the standard screen-pixel shape shown in Figure 10, and identify the circumcircle location using Hough transform method [7]. There are some smaller noise points after Hough transformation. We filter out the noise points by setting the mean of all contour radii as the radius threshold and the mean distance between contour centroids as the distance threshold. c) For *size*, we light up the R/G/B screen-pixels in turn, and apply procedure ②-⑥ in Figure 10 to each single R/G/B channel. Finally, we take the maximum radius of the detected contour as the radius of the R/G/B screen-pixel. Note that since the relative positions of the TSR and the under-screen camera are fixed, each smartphone only needs to perform this calibration procedure once.

Step 2: Establish the correct screen-pixel mask. After obtaining all the R/G/B screen-pixels, we need to know the current color of the TSR to establish the correct screen-pixel mask *since the screen's interference on decoding depends on screen's displayed colors.* To achieve this, we filter the pixels in the non-RoI by setting a threshold based on the average brightness level of these pixels, as shown in Figure 11. This step is necessary to remove the effect of black image noise. After converting the filtered pixels from the RGB color space to the standard CIE 1931 XYZ space, we obtain the screen-non-pixel area color point (x_s, y_s) corresponding to the current color of TSR. We then compare it with the screen color space used in smartphones [43]. For example, if the screen color point is on the vertex of the screen color triangle, we only need to set the screen mask with one set of R/G/B pixels. The sampling points under the screen-pixels which are not lit up, can still benefit demodulation. Step 2 presents us more space to pick up better sampling points in Horizontal scanning.

Step 3: Pick up pixels outside the R/G/B screen-pixel area. We set the middlemost pixel of the first row of RoI as the origin to start our scan. We scan each pixel in the row from left to right until one pixel located outside of the screen-pixel area is identified and this pixel is selected as the output sampling point. Then we jump to the closest pixel in the next row and repeat the above scanning process. Note that when the scanning process reaches the end of a row, we resume from the beginning of the same row.

4.4 Vertical Scan in Time Domain

According to the Rolling-Shutter effect, the maximum number of rows in one stripe without inter-symbol interference can be expressed as $n = \lceil (T - T_e)/T_r \rceil$, where T is the period of the transmitted symbol. As shown in Figure 9(b), when $\lceil (T - T_e)/T_r \rceil = 3$, there are at most[3] three pure CSK symbols (i.e., red color) in a stripe without inter-symbol interference. However, we can also see that, even after the previous horizontal scan step, there are still CSK symbols with mixed colors as shown in Figure 9(b). We therefore need to pick those CSK symbols with a pure color[4] for best communication performance.

We propose a vertical scan method to identify the pure-color CSK symbols. Specifically, we model the color information as a transition graph in Figure 12. The pure colors are denoted as nodes while mixed colors are denoted as lines connecting the nodes. Note that without inter-symbol interference, we can obtain the color transition graph in Figure 12(a) with all the nodes (i.e., pure colors) clearly separated. When there is more inter-symbol interference, we have more mixed colors and the color transmission graph is shown in Figure 12(b). Our objective is to clearly identify the nodes (i.e., pure-color CSK symbols) even in the presence of inter-symbol interference. We show all the possible color transition cases in Figure 13. We can see that while there exist clear direction changes at the nodes in the first three cases, there is no obvious direction change on the color transition graph in Case 4.

As shown in Figure 13, we denote the color change in the x coordinate as f_x, and the change in the y coordinate as f_y. For the first three cases, we have either $f'_x = 0$ or (and) $f'_y = 0$ at the nodes. In the challenging Case 4, there is no clear direction change at the transition nodes. However, due to the continuity of the color changes, we have $f''_x = 0$ and $f''_y = 0$. We thus leverage these

[3] When the initial exposure timestamp of the first red stripe is not synchronized with the symbol's starting point, the number of pure red CSK symbols is less than 3.

[4] Here pure color means one of the colors used in our CSK modulation/demodulation without a mixture of other colors.

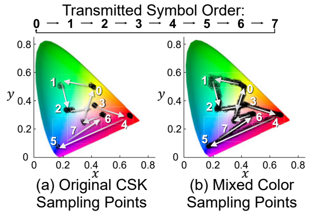


Figure 12: Color mixing.

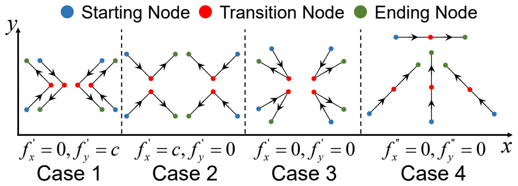


Figure 13: Transition graph in CIE1931 (Node: pure colors; Edge: mixed colors).

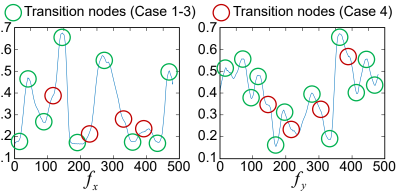


Figure 14: Illustrating for transition nodes found in vertical scan process.

properties to identify the transition nodes on the graph. We present an identification result using data collected in our experiment. As shown in Figure 14, we can accurately identify those transition nodes. We circle those transition nodes for the first three cases in green and those transition nodes for Case 4 in red.

5 DEMODULATION

After pixel-sweeping, the original two-dimensional sampling points in the RoI are significantly reduced to a $N \times 1$ sampling points vector, where N is the number of transmitted symbols in the corresponding frame. Even if we have identified the pure-color sampling points for each stripe, the sampling points are still affected by the *color-pulling effect* of *rendering interference* (cf. Section 3.2).

5.1 Classification of CSK symbols

The received CSK symbols (colors) can be divided into two groups, i.e., 1) one CSK symbol close to the screen color and 2) other CSK symbols. For the CSK symbol in 1), we can directly use distance detection to establish symbol mapping because they are minimally interfered by the screen. For the CSK symbols in 2), we define the line connecting the currently displayed screen color point and a CSK constellation point as a *ray*. Except for the screen color point, if there are multiple CSK symbols on a ray, we define this ray as *overlapping ray*, and these CSK symbols are regarded as overlapping symbols on this ray. If there is no overlapping, we can apply slope detection for demodulation. The slope here refers to the slope of the ray connecting the currently displayed screen color point and a CSK constellation point in the CIE1931 standard color coordinate system. We denote the screen color point and CSK symbol as $u_s = (x_s, y_s)$ and $u_m = (x_m, y_m)$, respectively. The slope of ray is then calculated as $(y_m - y_s)/(x_m - x_s)$.

5.2 Analysis of Overlapping Cases

We take the case of two overlapping CSK symbols (red symbol 1 and magenta symbol 2) on a ray as an example to illustrate the concept, and the rest cases–when there are multiple overlapping CSK symbols on a ray–can be deduced by analogy. The final positions of the two CSK symbols affected by the color-pulling effect of the screen light are analyzed, as shown in Figure 15. We can first rule out Case 1, as the color-pulling effect can only cause unidirectional movement. Cases 2, 3, and 4, however, are all possible. In Case 2, both symbols 1 and 2 experience minor interference. In Case 3, symbols 1 and 2 are both subjected to significant interference from the screen. Case 4 arises when symbol 1 experiences substantial interference while symbol 2 encounters minor interference. By applying the pixel-sweeping algorithm, we filter out non-uniform and strong interference within the screen-pixel area. Consequently, all $N \times 1$ sampling points undergo uniform rendering interference, which is either the same strong or weak, depending on the screen's brightness. As a result, only Cases 2 and 3 hold true. This implies that *the CSK symbols on the same ray can always be accurately distinguished under rendering interference.* We leverage the relative positions of these CSK symbols rather than the absolute positions for precise demodulation. For instance, we can map sampling points closer to the screen color point in Cases 2 and 3 to symbol 2, and the more distant points to symbol 1.

5.3 Slope-boosting Demodulation

Motivated by reserved relative position information on the overlapping ray, we design a slope-boosting scheme for demodulation. With the slope of ray, we can correctly demodulate the CSK symbols on non-overlapping rays, and distinguish those CSK symbols on overlapping rays. The details are as follows:

Step 1: Color classifier. First, we need to obtain the TSR's color coordinates in the captured frames. Accurate demodulation of the overlapping ray using the relative information between CSK symbols can only be achieved with rendering interference of the same color. To acquire the interference color coordinates in a frame, we refer to step 2) in Section 4.3 to determine the color of the TSR.

When the screen content changes at a high frequency, the color transitions within the TSR occur frequently. Consequently, each frame captured by the under-screen camera may contain multiple color interferences.[5] However, owing to the screen's row-by-row scan mechanism [26], we can still detect color changes within the same frame/image and differentiate the sampling points in an image affected by different TSR colors.

Step 2: Slope classifier. We denote all preset M-CSK symbols as $u_m = (x_m, y_m)$, $m \in \mathcal{M} \triangleq \{1, \cdots, M\}$. If the distance from the screen color point to preset CSK symbol is less than the threshold d_{th}, we determine that this preset CSK symbol is a screen color symbol, i.e., $u_s = (x_s, y_s)$. The threshold $d_{th} = 0.05$ is determined empirically that can minimize the decoding errors. All the rest of the preset CSK symbols are applied with the slope classifier defined as

$$\frac{\left|\frac{y_m-y_s}{|x_m-x_s|}-\frac{y_n-y_s}{|x_n-x_s|}\right|+\left|\frac{|y_m-y_s|}{x_m-x_s}-\frac{|y_n-y_s|}{x_n-x_s}\right|}{\left|\frac{y_m-y_s}{x_m-x_s}\right|+\left|\frac{y_n-y_s}{x_n-x_s}\right|}<s_{th}, \qquad (1)$$

where the slope is normalized to prevent slight jitter from causing severe slope fluctuation due to different distances from each preset CSK symbol to the screen color origin, and $s_{th} = 1$ is determined empirically as the slope change tolerance. For the output of the slope classifier, we have two types of sets $\mathcal{A}$ and $\mathcal{B}$. The non-overlapping

[5] Currently, due to limitations in the output frame rate of USC and refresh rate of the screen, there are at most two different TSR colors in a single captured frame.

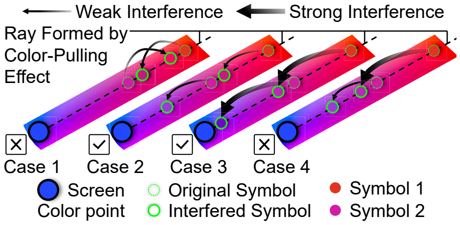
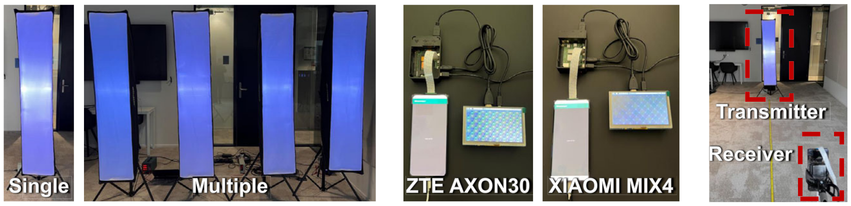


Figure 15: The overlapping cases.



(a) Our Implemented LED Transmitters (b) Our Implemented Receivers (c) Basic Setup

Figure 16: The evaluation setup.

CSK symbols are stored in $\mathcal{A}_i, i = \{1, \cdots, I\}$, where $I \leq M$ is the number of non-overlapping rays, and the number of CSK symbols in $\mathcal{A}_i$ is denoted as $|\mathcal{A}_i| = 1$. Also, $\mathcal{B}_j, j = \{1, \cdots, J\}$ stores all the CSK symbols with at least one overlapping case, where $J \leq M$ is the number of the overlapping rays, and $|\mathcal{B}_j| \geq 2$ represent the number of overlapping CSK symbols on the ray j. When the slope difference between symbol m and symbol n is smaller than s_{th} in Equation (1), the rays formed by the color-pulling effect of symbol m and symbol n overlap at ray j. Thus, m and n are put into the set $\mathcal{B}_j$, i.e., $\mathcal{B}_j = \mathcal{B}_j \cup \{m, n\}$, otherwise, $\mathcal{A}_i = \mathcal{A}_i \cup m$ and $\mathcal{A}_{i+1} = \mathcal{A}_{i+1} \cup n$.

Step 3: Distance decoder. We denote the input sampling point sequence as $r_l = (x_l^r, y_l^r), l = \{1, \cdots, L\}$, where L is the length of sequence. The distance decoder is expressed as $\|r_l - u_s\|^2 < d_{th}$. Thus, all transmitted symbols with least color-pulling effect are detected. Note that when the color of TSR is detected as black, all sample point sequences are input into the distance decoder for demodulation. The optimal minimum distance decoder is calculated as

$$r_l^* = \arg\min_{m \in \mathcal{M}} \|r_l - u_m\|_2, \tag{2}$$

where r_l^* is the decoded symbol.

Step 4: Slope-boosting decoder. We do slope-boosting detection on all remaining sample points except those processed by the distance decoder. The optimal slope decoder can be expressed as

$$r_l^* = \arg\min_{m \in \mathcal{M}} \frac{\left| \frac{y_l^r - y_s}{|x_l^r - x_s|} - \frac{y_m - y_s}{|x_m - x_s|} \right| + \left| \frac{|y_l^r - y_s|}{x_l^r - x_s} - \frac{|y_m - y_s|}{x_m - x_s} \right|}{\left| \frac{y_l^r - y_s}{x_l^r - x_s} \right| + \left| \frac{y_m - y_s}{x_m - x_s} \right|}. \tag{3}$$

If $r_l^* \in \mathcal{A}_i$, then it means that the ray where the demodulated symbol is located does not overlap, and the sampling point is directly mapped to r_l^*. Otherwise, if $r_l^* \in \mathcal{B}_j$, we put this sample point into set C_j, i.e., $C_j = C_j \cup r_l^*$. The sampling points in set C_j correspond to those sample points falling on the overlap ray represented by set $\mathcal{B}_j$.

The slope-boosting procedure is shown in Algorithm 1. Based on Step 1, we can group the CSK symbols interfered by the same TSR color to increase the number of symbols used for clustering. A larger number of sample points can improve the decoding accuracy.

Algorithm 1 Slope-boosting demodulation

```
Input: $\mathcal{B}_j$: preset overlapping CSK symbols on ray $j$, $C_j$: overlapping sampling points on ray $j$, $K$: the number of clusters, $D_{m,n} \triangleq \|p_m - p_n\|_2$
Output: $\boldsymbol{r}^*$
Sort $C$ in descending order of $|C|$
for $j = 1$ to $J$ do
    $K = |\mathcal{B}_j|$
    while $\boldsymbol{r}_j^* == \emptyset$ do
        $\boldsymbol{e}_i, i = \{1, \cdots, K\}$ with centroid $p_i$ are obtained by K-means clustering [4] on the set $C_j$
        if $D_{m,n} > d_{th}, \forall m, n \in \{1, \cdots, K\}$ then
            $r_l^* \in \boldsymbol{e}_i \leftarrow \mathcal{B}_j$, sort $\boldsymbol{e}_i$ and $\mathcal{B}_j$ in ascending order of $D_{m,n}$
        else if $D_{m,n} < d_{th}$ then
            $K = K - 1$
```

6 IMPLEMENTATION

LED transmitter. A snapshot of our implemented transmitter is presented in Figure 16(a). It includes a full-color LED chip, a lampshade, an LED driver board, and a control unit. The full-color LED chip has a maximum power consumption of 5 Watts and a maximum brightness of 450 lumen, powered by a 24V DC voltage. We use a long-strip lampshade with a dimension of 35 cm×160 cm in our experiment. The shape and length of the lampshades are similar to the commonly-seen fluorescent luminaires in offices. An Arduino DUE—a low-cost embedded platform with an 84 MHz processor—is used as the control unit at the LED transmitter. We leverage three independent PWM ports of the Arduino DUE to control the full-color LED separately for generating different CSK symbols. The LED driver has three transistors (ON MOSFET 20N06L) and three transistor drivers (TC4420). The PWM ports of the Arduino DUE trigger the transistors to modulate the full-color LED.

Receiver. The receivers we implement for the through-screen VLC are presented in Figure 16(b). We use two different models of full-screen smartphones that are equipped with under-screen cameras, i.e., ZTE AXON30 and Xiaomi MIX4, both are available in the market since late 2021. AXON30 employs an AMOLED screen that can reach a maximum brightness of 475 cd/m^2. MIX4 also has an AMOLED screen but with a maximum brightness of 903 cd/m^2. The screen refresh rate of the two smartphones both supports 60 Hz and 120 Hz, and use the color gamut of DCI-P3. For both smartphones, the pixel density of the TSR is 400 pixels per inch (PPI), but *the shape, size, and layout of pixels in the translucent screen regions are different.* We develop an APP to control the displayed color in the translucent screen region to evaluate the performance of our system under different screen colors. The APP can run with Android 4.4.2 and above.

For the under-screen camera, a 16-megapixel front camera and a 20-mega-pixel front camera are originally used in the AXON30 and the MIX4, respectively. Though the cameras' resolutions are very high which is good for VLC, we cannot obtain sufficient control/information over/of the cameras from the APIs provided by the smartphones' operating systems, such as shutter speed and output frame rate setting. Thus, we remove the back covers of the smartphones and replace the front cameras with Raspberry camera V2 (IMX219), which has 8 million pixels with a pixel size of 1.12 μm and a cost of around $25. This camera utilizes the Sony IMX series imaging sensor which is the same as that used in the front cameras of full-screen smartphones. Note that the schemes and algorithms we designed here can work with cameras of different quality levels, including the original front camera of the smartphone. We use Raspberry Pi 4 as our controller to connect the camera because it can run the OpenCV library and support the CSI-2 interface. The total power consumption of Raspberry Pi 4 and camera V2 is less than 1 W. The recorded frames at the Raspberry Pi 4 are transferred to a laptop for further processing.

7 PERFORMANCE EVALUATION

We evaluate the performance of our proposed through-screen VLC in various scenarios. We use Bit Error Rate (BER), data rate, and transmission range as the metrics for performance evaluation.

7.1 Preliminary Evaluation

The default experiment setup is shown in Figure 16(c). We place the transmitter and receiver with a distance of 1 m in between. At the transmitter, we adopt the traditional CSK design specified in the IEEE 802.15.7 standard [63]. The frequency of the PWM signals is set to 1 MHz to support 5 kHz CSK symbol rate. At the receiver, the screens are set to 100% full brightness. We configure the camera V2 to work in mode 7 and turn off automatic exposure/gain control to capture frames with a size of 640×480 at the highest frame rate of 200 frames per second (FPS). We set the ISO (sensitivity of the camera) to the highest value of 1600. We fix the camera exposure time to 1/5400 s, slightly lower than the period of the transmitted CSK symbols, which is 200 μs. The read-out duration of the camera is estimated to be around 10 μs, using the method presented in [29]. We mainly use AXON30 for evaluation (except in Section 7.4). To evaluate the performance of our system, we compare the performance of four solutions:

- *Ours:* As presented in the previous sections, our solution leverages *i*) the proposed pixel-sweeping algorithm to obtain the pixels that represent each color stripe (cf. Section 4), and *ii*) the proposed slope-boosting detection algorithm to decode the CSK symbols (cf. Section 5).
- *Original:* It uses conventional method to sample pixels at a fixed width instead of our pixel-sweeping algorithm; It also only adopts conventional minimum Euclidean distance detection for CSK symbol decoding [21, 66].
- *Only pixel-sweeping:* It only adopts our pixel-sweeping algorithm to obtain the pixels; For decoding the CSK symbols, it uses the conventional minimum Euclidean distance detection method [21, 66] instead of our proposed slope-boosting detection algorithm.
- *Only slope-boosting:* It samples the pixels at a fixed width instead of our proposed pixel-sweeping algorithm; It adopts our proposed slope-boosting detection algorithm to decode the CSK symbols.

BER results versus screen color. We evaluate the system performance under eight different colors of the screen: *black,*[6] *red, blue, green, magenta, cyan, yellow, white.* We use our designed APP to switch the smartphone's screen color among these eight colors. The BER results under 8-CSK are shown in Figure 17. We first observe that without our proposed algorithms, i.e., with the 'Original' solution, the BER under all the eight screen colors goes beyond 10^{-1}, which fails the through-screen VLC link. However, with our proposed solution ('Ours'), the BER can be as low as 10^{-4} when the smartphone's screen is not lit up ('black'), and around 10^{-3} when the smartphone displays other seven colors. An interesting observation is that among the three colors: red, green and blue, the BER under the green screen is the highest. This is because in OLED, the number of green pixels is much larger than that of red and blue pixels. Therefore, more interference is generated when the smartphone displays green color. We observe that the BER under 'Only pixel-sweeping' is lower than the BER under 'Only slope-boosting'. Also for both of them, the BERs under different screen colors (except 'black') are always higher than 10^{-2}. We further evaluate the system performance when 16-CSK is used at the transmitter. The results are presented in Figure 18. We can observe that under all the eight different screening colors, our proposed solution can still achieve a BER below 10^{-2}.

In the rest of the evaluations in this section, unless otherwise specified, we use 8-CSK at the transmitter and consider the most challenging scenario, i.e., the smartphone displays white color (all the R/G/B pixels of the screen are lit up).

Impact of transmitter size/shape. We next investigate the impact of transmitter shape and size on the achieved data rate. As shown in Figure 20(a), we use four lampshades with distinct shapes and sizes at the transmitter: (i) a circular lampshade with a diameter of 40 cm (Circle-40cm); (ii) a square lampshade with a side length of 40 cm (Square-40cm); (iii) a rectangular lampshade with a size of 30 x 120 cm (Rectan-120cm); and (iv) a rectangular lampshade with a size of 35 x 160 cm (Rectan-160cm). Figure 20(b) shows the impact of these lampshade shapes and sizes on the achieved system data rate. Rectangular lampshades of 160 cm and 120 cm exhibit a data rate drop when the communication distance exceeds 4 m and 3 m, respectively, while square and circular lampshades show a data rate drop beyond 1 m. This is due to the rolling-shutter effect, as the system's data rate is significantly influenced by the number of pixel rows of the light source in the image captured by the under-screen camera. The smaller the captured light source in the image, the less information can be decoded at the receiver. The circular lampshade achieves a slightly lower data rate compared to the square lampshade due to a smaller light source area. Note that even when small lampshades (e.g., a circular lampshade with a diameter of 40 cm) are employed, a data rate larger than 4 kb/s can still be achieved within a range of 2 m.

Impact of screen brightness and transparency. To avoid the possible damage to human eyes, smartphones' screens adopt DC dimming instead of PWM dimming [12]. Therefore, the brightness of the screen determines how much interference it causes on the

[6] This is achieved when the smartphone's screen is not lit up.

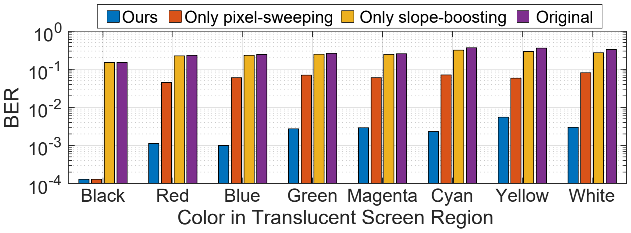


Figure 17: BER results vs. screen color (8-CSK).

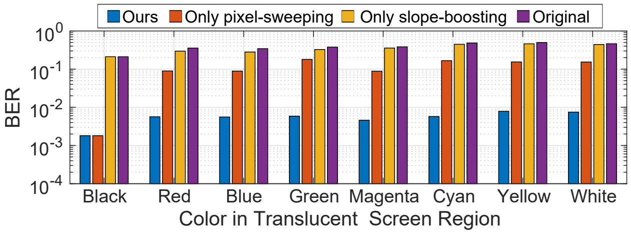


Figure 18: BER results vs. screen color (16-CSK).

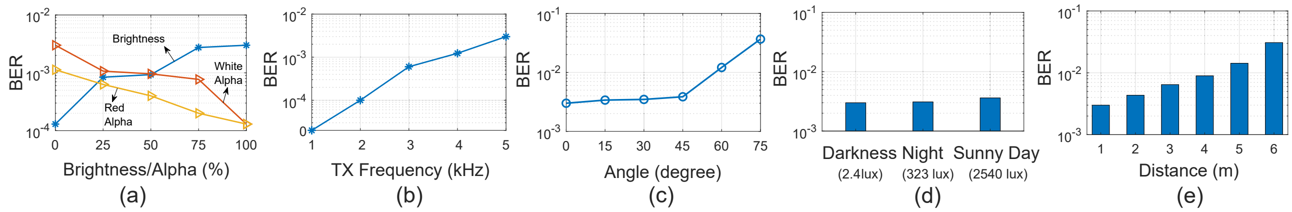
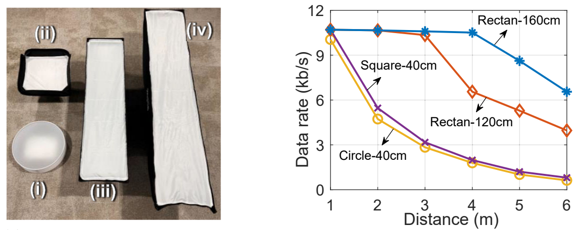


Figure 19: BER results vs. (a) brightness/alpha, (b) TX frequency, (c) TX-camera angle, (d) ambient light, (e) distance.



(a) Different lampshade setup

(b) Data rate vs. distance

Figure 20: System data rate vs. size and shape of the light source.

transmitted CSK signals. We carry out experiments to evaluate the effect of screen brightness on through-screen VLC. The BER results are shown in Figure 19(a). Overall, as the brightness increases, the BER gradually increases. When the brightness is lower than 50%, the BER is lower than 10^{-3}. When the brightness is higher than 50%, the BER is higher but still lower than 10^{-2}. On the other hand, as a well-known concept in computer graphics, the *alpha channel* is widely used to form a composite image with partial or full transparency [46]. It stores a value between 0 and 1 to indicate pixel translucency: 0 means that the pixel is fully opaque and 1 means that it is fully transparent. We also evaluate the impact of screen transparency on the performance of our system by changing the alpha value. The results are shown in Figure 19(a). We observe that changing the transparency of a pixel does not affect the color coordinates. The BERs under red and white screens both gradually decrease when the transparency increases, while the screen color coordinates detected do not change. This result is interesting because it means changing pixel transparency can be equivalent to adjusting screen's brightness.

Impact of transmitter frequency. To capture the effect of transmitter frequency (CSK symbols transmitted per second), we vary it from 1 kHz to 5 kHz at a step size of 1 kHz. Figure 19(b) shows the BER results when 8-CSK is used. We observe that as we increase the transmitter frequency, the BER increases. At 1 kHz, the BER is 0; at 2 kHz, the BER is around 10^{-4}; and at 4 kHz, the BER increases to 10^{-3}. This is because the width of the color stripe decreases with a higher transmitter frequency. This increases the inter-symbol interference as it becomes more difficult to distinguish color with fewer pixels.

7.2 Robustness Evaluation

Impact of angle. We evaluate the robustness of our system when the receiver is placed in different directions with respect to the LED transmitter. We fix the distance between the transmitter and screen to 1 m and vary the relative angle between them. The results are shown in Figure 19(c). We can observe that the BER maintains almost at the same level when the angle between the transmitter and the receiver does not exceed 45°. At the angle of 60°, the BER increases to around 10^{-2}. The results show that our system can work well when the transmitter-receiver angle is below 60°.

Impact of ambient light conditions. Next, we evaluate the system robustness under different ambient light conditions. Three ambient light conditions are considered: (1) Darkness (average light intensity: 2.4 lux); (2) Night with indoor illumination (average light intensity: 323 lux); and (3) Sunny day (average light intensity: 2540 lux). The impact of ambient light condition on BER is presented in Figure 19(d). First, we observe that the BER uder all three conditions is lower than 10^{-2}. Thus, the performance of the system is not affected much by the ambient light level. This is because the exposure time of the camera is usually set very small to exploit the rolling-shutter effect of the camera to detect the rapidly changing stripes. This makes the impact of interference from ambient light very limited on our system. In addition, even in a bright indoor environment (sunny day), with our proposed RoI detection algorithm, we can still accurately detect the light source, which further alleviates the impact on the BER performance.

Impact of distance. Lastly, we evaluate the system robustness when the transmitter is placed at different distances from the receiver. The BER results are shown in Figure 19(e). As the incident light attenuates with distance, a longer distance causes a lower

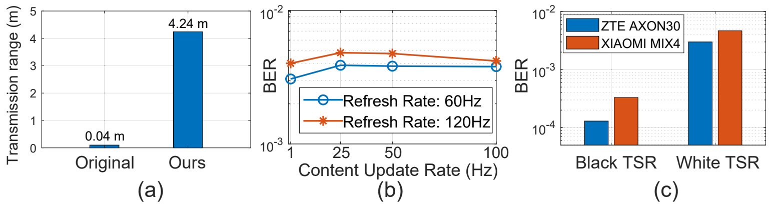


Figure 21: Results: (a) Range; (b-c) BER under different setups.

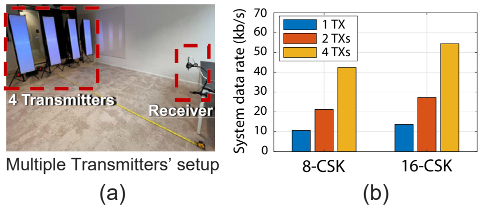


Figure 22: Multiple transmitters Results.

signal strength at the receiver. The variance of the cluster formed by CSK symbols on CIE1931 becomes larger due to the decrease in SINR. Therefore, the larger the distance, the higher the BER. However, even at a distance of 4 m, the achieved BER is still lower than 10^{-2}.

Following the state-of-the-art studies [52, 54], we define the *transmission range* as the maximal communication distance between the transmitter and receiver with a BER below 10^{-2}. The transmission ranges under different solutions as shown in Figure 21(a). We can observe that in the presence of the translucent screen that covers the under-screen camera, the 'Original' solution can only achieve a transmission range of 4 cm. On the other hand, by adopting the proposed pixel-sweeping and slope-boosting detection algorithms, the transmission range is extended to 4.24 m, bringing a gain over 100×.

7.3 Dynamic Contents on Screen

Next, we display dynamic contents on the TSR and evaluate the impact on through-screen VLC. We change TSR's displayed color periodically following the order of *black, red, blue, green, magenta, cyan, yellow, and white.* We test four content update rates: 1 Hz, 25 Hz, 50 Hz, and 100 Hz, all of which are within the content update rate range of commonly used APPs.[7] The achieved BERs under different content update rates and different screen refresh rates are shown in Figure 21(b). We can observe that at all the content update rates, the BERs are always below 10^{-2}. We note that when the content update rate is 25 Hz, the BER is slightly higher. This is because the frame rate of the screen is extremely unstable at 25 Hz.

7.4 Screen Diversity

We also evaluate the system performance with different commercial smartphones at the receiver. We test two commercial smartphones, ZTE AXON30 and Xiaomi MIX4, as previously shown in Figure 16(b). The translucent screen regions of the two smartphones have dramatically different pixel layouts, sizes, and shapes. Also, their screens differ in maximum brightness. We fix the distance between the transmitter and receiver as 1 m and measure the BER of the through-screen VLC link when different smartphone screens are adopted. The results are shown in Figure 21(c). We can observe that although there are some performance variations under different smartphone screens, the BERs are always below 10^{-2}, demonstrating the effectiveness of our proposed pixel-sweeping algorithm. Interestingly, we find that the BER under the screen of MIX4 is higher than that of AXON30, even when the screen is not lit up (i.e., 'black'). We believe this is because MIX4 have a lower light penetration rate compare to AXON30 due to intrinsic screen material and thickness diversity. When smartphones display white color at full screen brightness, MIX4 causes higher interference to the captured frames because it has a higher brightness.

7.5 Multiple Transmitters

Finally, we carry out experiments to evaluate the system performance under multiple transmitters. The under-screen camera is a multi-pixel receiver. Therefore, a single receiver with an under-screen camera can detect the CSK signals from multiple LED transmitters by splitting a captured frame into several slices and detecting the RoI in each slice independently. With multiple transmitters, the system data rate can be significantly increased. In our evaluation, we test up to four transmitters. We place the transmitters in a row at a distance of 3 meters from the receiver. A snapshot of the experiment setup with four transmitters is shown in Figure 22(a). We test both 8-CSK and 16-CSK. The evaluation results are shown in Figure 22(b). We can observe that under 8-CSK, the date rates achieved in our through-screen VLC system can reach 10.59 kb/s, 21.19 kb/s and 42.38 kb/s with one transmitter, two transmitters and four transmitters, respectively. Under 16-CSK, the maximum system data rate can go up to 54.43 kb/s with four transmitters.

8 DISCUSSION AND RELATED WORK

Multi-color screen contents. Our system works well when the entire Translucent Screen Region (TSR) displays a single color (the color can change over time), as shown in our evaluation. The TSR is located at the top of the smartphone and this part usually hosts the status bar. Thus, the TSR usually only displays a single color at a time, such as white in many APPs. However, for APPs running on a full screen such as games, complex contents with different colors could be displayed in the region. In this case, every single frame captured by the Under-Screen Camera (USC) is disturbed by a different color, making decoding more challenging. For this scenario with multi-color screen contents, one potential solution is to cancel the effect of the screen color at each single screen-pixel. We will investigate this in the future.

Through-screen visible light communication and sensing. Through-screen VLC was proposed in a recent work SpiderWeb [62]. SpiderWeb discovered the color-pulling effect and proposed a new modulation scheme termed SWebCSK to address the challenges brought by the screen. However, SpiderWeb requires the transmitters to know the receiving device's current screen color and

[7] In the experiment, we find that even when we set the screen refresh rate to 60 Hz and the content update rate to 100 Hz, due to the screen optimization mechanism in Android OS [3], the phone still displays contents at 60 Hz.

Table 1: Summary of LED-to-Camera communication systems (in the one-transmitter & one-receiver setup).

Name	Modulation	Throughput	Range
VLandmark[40]	Binary FSK	10 b/s	1.2 m
RollingLight [29]	FSK	90.56 b/s	Not Specified
CamCom [41]	OOK	15 b/s	Not Specified
Seminal[6]	OOK	148 b/s	9 cm
ReliableVLC[10]	OOK	700 b/s	3 m
IoTorch [20]	PWM	2.92 kb/s	1 m
CeilingTalk [57]	OOK-PWM	1 kb/s	5 m
ReflexCode [59]	GSK	1.07 kb/s	3 m
Martian [8]	Prefix code	1.6 kb/s	25.4 cm
ColorBars [21]	CSK	5.2 kb/s	3 cm
Our work with USC	**CSK**	**13.61 kb/s**	**4.2 m**

optimize its modulation constellation points accordingly, introducing a large overhead on transmitter-receiver synchronization and constellation optimization. In contrast, our design does not require any modifications on LED transmitters, allowing us to achieve a much higher data rate compared to SpiderWeb. A recent work [35] also proposed through-screen visible light sensing for fingertip air-writing. In contrast, our work focuses on through-screen communication with visible light.

Image restoration on under-screen cameras. Under-screen cameras for smartphones have spurred interest in techniques that can restore images captured by them. A recent work [9] presented a method to recover dimmed and blurred images by restoring angular frequencies in the scene. Other works [28, 33, 42, 45, 67] also applied deep neural networks to handle the blur and low SNR issues associated with under-screen images. However, these works mainly addressed the passive interference on the camera induced by the screen when the screen is off. Instead, our work aims to remove the impact of active interference from the screen on VLC when the screen is on, which has never been considered in previous works.

LED-to-camera communication. It utilizes the existing lighting infrastructure as transmitters and pervasively available rolling shutter cameras as the receivers [8, 21, 57, 59]. The work in [57, 59] investigated a high-order intensity modulation by encoding data into different luminance levels. ColorBars exploited CSK to improve the data rates [21]. The data rate using pulse-based optical camera communication modulation methods such as OOK and FSK does not exceed 4.2 kb/s as shown in Table 1. ColorBars uses CSK modulation to reach a data rate of 5.3 kb/s, but the communication distance is only 3 cm. In our work, the dynamic contents on screens bring unique challenges for realizing through-screen VLC. Still a higher data rate and a longer communication range can both be achieved with our designed algorithms.

Screen-to-camera communication. This communication modality employs images or videos on a standard monitor to transmit data information [17, 19, 38, 50, 64]. However, these coded images are typically visible to users, comprising the confidentiality of the data content. Thus, hidden screen-to-camera communication was proposed to achieve communication without comprising the confidentiality [25, 31, 34, 49, 51, 65]. In through-screen VLC, the screens become receivers with cameras under the screen as antennas. Transmitters are those ceiling lamps which are used for both illumination and communication. We utilize the color information and apply dedicated signal processing methods to achieve a throughput of 54.43 kb/s and a range of 4.2 m with four transmitters.

9 CONCLUSION

In this paper, we studied how to enable through-screen VLC with under-screen cameras on full-screen devices such as smartphones. We identified key challenges associated with the translucent OLED screen that covers the under-screen camera. We proposed a pixel-sweeping algorithm and a slope-boosting demodulation method to address the challenges. Our results from comprehensive experiments demonstrated the feasibility of through-screen VLC. We envision that our work could stimulate follow-up studies on communication & sensing using under-screen cameras.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers and the shepherd for their valuable comments and suggestions. This work has been funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska Curie grant agreement ENLIGHT'EM No. 814215. We would like to thank Pei Tian for her suggestions on figure optimization.

REFERENCES

[1] IEEE 802.11. 2019. 802.11 Light Communications Amendment. (2019).

[2] Adrian Wong. 2020. Xiaomi 3rd Generation Under-Display Camera Technology! *https://www.engadget.com/xiaomis-under-display-camera-tech-is-coming-to-phones-next-year-130012911.html* (September 2020).

[3] Android. 2022. Slow rendering. *https://developer.android.com/topic/performance/vitals/render* (August 2022).

[4] Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning.*

[5] Sam Byford. 2021. Xiaomi announces Mix 4 with under-display camera. *https://www.theverge.com/2021/8/10/22617979/xiaomi-mi-mix-4-announced-under-display-camera-cup-specs-price* (Aug 2021).

[6] Christos Danakis, Mostafa Afgani, Gordon Povey, Ian Underwood, and Harald Haas. 2012. Using a CMOS camera sensor for visible light communication. In *Proceedings of the IEEE Globecom Workshops.*

[7] E Roy Davies. 2017. *Computer vision: principles, algorithms, applications, learning.* Academic Press.

[8] Haohua Du, Junze Han, Xuesi Jian, Taeho Jung, Cheng Bo, Yu Wang, and Xiang-Yang Li. 2017. Martian: Message broadcast via LED lights to heterogeneous smartphones. *IEEE Journal on Selected Areas in Communications* (2017).

[9] Neil Emerton, David Ren, and Tim Large. 2020. 28-1: Image Capture Through TFT Arrays. In *SID Symposium Digest of Technical Papers.* Wiley Online Library.

[10] Julia Ferrandiz-Lahuerta, Daniel Camps-Mur, and Josep Paradells-Aspas. 2015. A reliable asynchronous protocol for VLC communications based on the rolling shutter effect. In *Proceedings of the IEEE GLOBECOM.*

[11] 6G Flagship. 2019. Key Drivers and Research Challenges for 6G Ubiquitous Wireless Intelligence. (2019).

[12] Florian Schmitt. 2019. Analysis: DC Dimming vs. PWM – Can you dim AMOLED displays without the flickering? *https://www.notebookcheck.net/Analysis-DC-Dimming-vs-PWM-Can-you-dim-AMOLED-displays-without-the-flickering.423121.0.html* (June 2019).

[13] Gijong Lee. 2021. LG Display starts under display camera tech development. *https://www.thelec.net/news/articleView.html?idxno=3485&replyAll=&reply_sc_order_by=I* (October 2021).

[14] A. Gomez, K. Shi, C. Quintana, M. Sato, G. Faulkner, and et. al. 2015. Beyond 100-Gb/s Indoor Wide Field-of-View Optical Wireless Communications. *IEEE Photonics Technology Letters* (2015).

[15] Mark Halper. 2021. Laser light pioneer now moving in on Li-Fi. *https://www.ledsmagazine.com/specialty-ssl/article/14074310/laser-light-pioneer-now-moving-in-on-lifi* (January 2021).

[16] Jie Hao, Yanbing Yang, and Jun Luo. 2016. CeilingCast: Energy efficient and location-bound broadcast through LED-camera communication. In *Proceedings*

of the IEEE INFOCOM.

[17] Tian Hao, Ruogu Zhou, and Guoliang Xing. 2012. COBRA: Color barcode streaming for smartphone systems. In *Proceedings of the ACM MobiSys.*

[18] Richard Hartley and Andrew Zisserman. 2003. *Multiple view geometry in computer vision.* Cambridge university press.

[19] Frederik Hermans, Liam McNamara, Gábor Sörös, Christian Rohner, Thiemo Voigt, and Edith Ngai. 2016. Focus: Robust visual codes for everyone. In *Proceedings of the ACM MobiSys.*

[20] Yuki Hokazono, Aoi Koizuka, Guibing Zhu, Makoto Suzuki, Yoshiaki Narusue, and Hiroyuki Morikawa. 2019. IoTorch: Reliable LED-to-Camera Communication Against Inter-Frame Gaps and Frame Drops. *IEEE Transactions on Mobile Computing* (2019).

[21] Pengfei Hu, Parth H. Pathak, Xiaotao Feng, Hao Fu, and Prasant Mohapatra. 2015. ColorBars: Increasing Data Rate of LED-to-Camera Communication Using Color Shift Keying. In *Proceedings of the ACM CoNEXT.*

[22] IEEE 802.15 WSN Task Group 7a (TG7a). 2020. Standard for Local and Metropolitan Area Networks - Part 15.7: Short-Range Optical Wireless Communications Amendment: Higher Speed, Longer Range Optical Camera Communication (OCC). (2020).

[23] Ilse Jurrien. 2021. Samsung Under Panel Camera for smartphones and TVs. *https://nl.letsgodigital.org/smartphones/samsung-under-panel-camera/* (January 2021).

[24] ITU-T. 2019. G.9991: High-speed indoor visible light communication transceiver - System architecture, physical layer and data link layer specification. (2019).

[25] Mostafa Izz, Zhongyuan Li, Hongbo Liu, Yingying Chen, and Feng Li. 2016. Uber-in-light: Unobtrusive visible light communication leveraging complementary color channel. In *Proceedings of the IEEE INFOCOM.*

[26] Keith Jack. 2011. *Video demystified: a handbook for the digital engineer.* Elsevier.

[27] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta. 2014. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of the ACM MobiCom.*

[28] Kinam Kwon, Eunhee Kang, Sangwon Lee, Su-Jin Lee, Hyong-Euk Lee, ByungIn Yoo, and Jae-Joon Han. 2021. Controllable image restoration for under-display camera in smartphones. In *Proceedings of the IEEE CVPR.*

[29] Hui-Yu Lee, Hao-Min Lin, Yu-Lin Wei, Hsin-I Wu, Hsin-Mu Tsai, and Kate Ching-Ju Lin. 2015. Rollinglight: Enabling line-of-sight light-to-camera communications. In *Proceedings of the ACM MobiSys.*

[30] LG. 2021. LG Transparent OLED Signage. (Oct. 2021). https://www.lg-informationdisplay.com/oled-signage/brand

[31] Tianxing Li, Chuankai An, Xinran Xiao, Andrew T. Campbell, and Xia Zhou. 2015. Real-Time Screen-Camera Communication Behind Any Scene. In *Proceedings of the ACM MobiSys.*

[32] Ping-Sung Liao, Tse-Sheng Chen, Pau-Choo Chung, et al. 2001. A fast algorithm for multilevel thresholding. *J. Inf. Sci. Eng.* (2001).

[33] Sehoon Lim, Yuqian Zhou, Neil Emerton, Tim Large, and Steven Bathiche. 2020. Image Restoration for Display-Integrated Camera. In *SID Symposium Digest of Technical Papers.* Wiley Online Library.

[34] Hongbo Liu, Bo Liu, Cong Shi, and Yingying Chen. 2017. Secret key distribution leveraging color shift over visible light channel. In *Proceedings of the IEEE CNS.*

[35] Hao Liu, Hanting Ye, Jie Yang, and Qing Wang. 2021. Through-Screen Visible Light Sensing Empowered by Embedded Deep Learning. In *Proceedings of the ACM SenSys Workshop on AIChallengeIoT.*

[36] Michael Simon. 2022. Apple and Samsung are developing under-display camera tech for the iPhone 15 Pro. *https://www.macworld.com/article/625530/iphone-15-pro-under-display-face-id.html* (March 2022).

[37] Nobuyuki Otsu. 1979. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics* (1979).

[38] Samuel David Perli, Nabeel Ahmed, and Dina Katabi. 2010. PixNet: Interference-free wireless links using LCD-camera pairs. In *Proceedings of the ACM MobiCom.*

[39] P.Gagnon. 2019. Presentation at OLED World Summit 2019. *AMOLED Market & Technology Trend* (2019).

[40] Niranjini Rajagopal, Patrick Lazik, and Anthony Rowe. 2014. Visual light landmarks for mobile devices. In *Proceedings of the IEEE IPSN.*

[41] Richard D Roberts. 2013. Undersampled frequency shift ON-OFF keying (UFSOOK) for camera communications (CamCom). In *Proceedings of the IEEE WOCC.*

[42] Hrishikesh Panikkasseril Sethumadhavan, Densen Puthussery, Melvin Kuriakose, and Jiji Charangatt Victor. 2020. Transform Domain Pyramidal Dilated Convolution Networks for Restoration of Under Display Camera Images. In *Proceedings of the ECCV.*

[43] Gaurav Sharma and Raja Bala. 2017. *Digital color imaging handbook.* CRC press.

[44] Shikhar Mehrotra. 2022. iPhone 15 Pro Updates: Apple Could Use An Under-display Camera Module. *https://www.republicworld.com/technology-news/mobile/iphone-15-pro-updates-apple-could-use-an-under-display-camera-module-developed-by-samsung-articleshow.html* (March 2022).

[45] Varun Sundar, Sumanth Hegde, Divya Kothandaraman, and Kaushik Mitra. 2020. Deep atrous guided filter for image restoration in under display cameras. In *Proceedings of the ECCV.*

[46] Richard Szeliski. 2010. *Computer vision: algorithms and applications.* Springer Science & Business Media.

[47] Zhao Tian, Kevin Wright, and Xia Zhou. 2016. The DarkLight Rises: Visible Light Communication in the Dark. In *Proceedings of the ACM MobiCom.*

[48] Takatoshi Tsujimura. 2017. *OLED display fundamentals and applications.*

[49] Anran Wang, Zhuoran Li, Chunyi Peng, Guobin Shen, Gan Fang, and Bing Zeng. 2015. Inframe++ achieve simultaneous screen-human viewing and hidden screen-camera communication. In *Proceedings of the ACM MobiSys.*

[50] Anran Wang, Shuai Ma, Chunming Hu, Jinpeng Huai, Chunyi Peng, and Guobin Shen. 2014. Enhancing reliability to boost the throughput over screen-camera links. In *Proceedings of the ACM MobiCom.*

[51] Anran Wang, Chunyi Peng, Ouyang Zhang, Guobin Shen, and Bing Zeng. 2014. Inframe: Multiflexing full-frame visible communication channel for humans and devices. In *Proceedings of the ACM HotNets.*

[52] Purui Wang, Lilei Feng, Guojun Chen, Chenren Xu, Yue Wu, Kenuo Xu, Guobin Shen, Kuntai Du, Gang Huang, and Xuanzhe Liu. 2020. Renovating Road Signs for Infrastructure-to-Vehicle Networking: A Visible Light Backscatter Communication and Networking Approach. In *Proceedings of the ACM MobiCom.*

[53] Zhibin Wang, Yilu Chang, Qi Wang, Yingjie Zhang, Jacky Qiu, and Michael Helander. 2020. 55-1: Invited Paper: Self-Assembled Cathode Patterning in AMOLED for Under-Display Camera. In *SID Symposium Digest of Technical Papers.*

[54] Yue Wu, Purui Wang, Kenuo Xu, Lilei Feng, and Chenren Xu. 2020. Turboboosting visible light backscatter communication. In *Proceedings of the ACM SIGCOMM.*

[55] Xieyang Xu, Yang Shen, Junrui Yang, Chenren Xu, Guobin Shen, Guojun Chen, and Yunzhe Ni. 2017. PassiveVLC: Enabling practical visible light backscatter communication for battery-free iot applications. In *Proceedings of the ACM MobiCom.*

[56] Anqi Yang and Aswin C Sankaranarayanan. 2021. Designing Display Pixel Layouts for Under-Panel Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[57] Yanbing Yang, Jie Hao, and Jun Luo. 2017. CeilingTalk: Lightweight indoor broadcast through LED-camera communication. *IEEE Transactions on Mobile Computing* (2017).

[58] Yanbing Yang, Jun Luo, Chen Chen, Zequn Chen, Wen-De Zhong, and Liangyin Chen. 2021. Pushing the Data Rate of Practical VLC via Combinatorial Light Emission. *IEEE Transactions on Mobile Computing* (2021).

[59] Yanbing Yang, Jiangtian Nie, and Jun Luo. 2017. Reflexcode: Coding with superposed reflection light for LED-camera communication. In *Proceedings of the ACM MobiCom.*

[60] Zhice Yang, Zeyu Wang, Jiansong Zhang, Chenyu Huang, and Qian Zhang. 2015. Wearables Can Afford: Light-Weight Indoor Positioning with Visible Light. In *Proceedings of the ACM MobiSys.*

[61] Zhice Yang, WANG Zeyu, Jiansong Zhang, Chenyu Huang, and Qian Zhang. 2018. Polarization-based visible light positioning. *IEEE Transactions on Mobile Computing* (2018).

[62] Hanting Ye and Qing Wang. 2021. SpiderWeb: Enabling Through-Screen Visible Light Communication. In *Proceedings of the ACM SenSys.*

[63] SH Yoon, KS Lee, JS Cha, V Mariappan, KE Young, DG Woo, and JU Kim. 2019. IEEE Standard for Local and Metropolitan Area Networks–Part 15.7: Short-Range Optical Wireless Communications. *IEEE Std 802.15.7-2018* (2019).

[64] Bingsheng Zhang, Kui Ren, Guoliang Xing, Xinwen Fu, and Cong Wang. 2015. SBVLC: Secure barcode-based visible light communication for smartphones. *IEEE Transactions on Mobile Computing* (2015).

[65] Kai Zhang, Yi Zhao, Chenshu Wu, Chaofan Yang, Kehong Huang, Chunyi Peng, Yunhao Liu, and Zheng Yang. 2021. ChromaCode: A Fully Imperceptible Screen-Camera Communication System. *IEEE Transactions on Mobile Computing* (2021).

[66] Xiao Zhang and Li Xiao. 2020. RainbowRow: Fast Optical Camera Communication. In *Proceedings of the ACM ICNP.*

[67] Yuqian Zhou, David Ren, Neil Emerton, Sehoon Lim, and Timothy Large. 2021. Image restoration for under-display camera. In *Proceedings of the IEEE CVPR.*

[68] ZTE. 2021. ZTE launches its new-generation under-display camera smartphone Axon 30. *https://www.zte.com.cn/global/about/news/20210727e1.html* (July 2021).