High Speed LED-to-Camera Communication using Color Shift Keying with Flicker Mitigation

Pengfei Hu, Parth H. Pathak, Huanle Zhang, *Member, IEEE*, Zhicheng Yang, *Member, IEEE*, and Prasant Mohapatra, *Fellow, IEEE*

Abstract—LED-to-camera communication allows LEDs deployed for illumination purposes to modulate and transmit data which can be received by camera sensors available in mobile devices like smartphones, wearable smart-glasses, etc. Such communication has a unique property that a user can visually identify a transmitter (i.e., LED) and specifically receive information from the transmitter. It can support a variety of novel applications such as augmented reality through mobile devices, navigation using smart signs, fine-grained location specific advertisement, etc. However, the achievable data rate in current LED-to-camera communication techniques remains very low to support any practical application. In this paper, we present ColorBars, an LED-to-camera communication system that utilizes Color Shift Keying (CSK) to modulate data using different colors transmitted by the LED. It exploits the increasing popularity of Tri-LEDs (RGB) that can emit a wide range of colors. We show that commodity cameras can efficiently and accurately demodulate the color symbols. ColorBars ensures flicker-free and reliable communication even in the presence of inter-frame loss and diversity of rolling shutter cameras. We implement ColorBars on embedded platform and evaluate it with Android and iOS smartphones as receivers. Our evaluation shows that ColorBars can achieve a data rate of 7.7 Kbps on Nexus 5, 3.7 Kbps on iPhone 5S, and 2.9 Kbps on Samsung Note8. It is also shown that lower CSK modulations (e.g., four and eight CSK) provide extremely low symbol error rates ($< 10^{-3}$), making them a desirable choice for reliable LED-to-camera communication.

Index Terms—Visible light communication, LED-to-camera communication, rolling shutter, color shift keying

1 Introduction

Increasing adoption of LEDs for common lighting applications in indoor environment has provided a unique opportunity to utilize them for Visible Light Communication (VLC). In VLC, information can be transmitted by the LEDs using different modulation techniques, and can be received by either a high-speed photodiode or a camera sensor commonly available in today's mobile devices. The LED-to-camera communication holds special importance because it enables any mobile device (smartphone, tablet, wearables like smart-glasses etc.) with a camera to receive the information transmitted by the LED. One major advantage of LED-to-camera communication is that it allows a user to visually locate a transmitter (i.e., LED) and receive the information specifically transmitted by that transmitter. This association of the transmitter identity and the transmitted information has a huge potential to create many novel applications such as augmented reality through cameras of mobile devices, fine-grained location specific services such as advertisements and navigation etc. For example, in a retail store, a consumer can visualize an LED on top of a merchandise rack through her smartphone or smartglasses, and receive small text objects (for example, hyperlinks to advertisements, which can then be downloaded over WiFi or cellular connection) or images (e.g., dynamically retrieved images that can be used as coupons/promotions). In augmented reality applications, for example, the LED lights in an office can broadcast real-time location and direction information, which can then be combined with visual information retrieved over high-speed WiFi or cellular connection to present users with complete floor map and navigation information.

The visual association property of LED-to-camera communication is difficult to achieve using Radio Frequency (RF) or Near Field Communication (NFC). Since data can be received from multiple transmitters in RF communication, it is difficult to associate the data to a visually identifiable transmitter. The communication range of NFC is much shorter compared to VLC making it unsuitable for such applications. The LED-to-camera communication has great potential, however, high achievable data rate with off-the-shelf transmitter implementation remains one of its biggest challenges. The rolling shutter phenomenon of the camera which allows it to receive the data transmitted by LEDs also imposes limits on the type of modulation that can be used by the LEDs, resulting in low link data rates. In recent works such as [1] and [2], the achievable data rate of LED-to-camera link is within tens of bytes per second. More recently, authors in [3] demonstrate an achievable data rate of 10.32 Kbps for LED-to-camera links.

- *P. Hu is with VMWare, Palo Alto, CA 94304. E-mail: pengfeih@vmware.com.*
- *P. H. Pathak is with the Computer Science Department, George Mason University, Fairfax, VA 22030. E-mail: {dtczhang, zcyang, pmohapatra}@ucdavis.edu.*
- *H. Zhang, Z. Yang, and P. Mohapatra are with the Department of Computer Science, University of California, Davis, CA 95616. E-mail: phpathak@gmu.edu.*

Manuscript received 24 Feb. 2018; revised 26 Jan. 2019; accepted 11 Apr. 2019. Date of publication 30 Apr. 2019; date of current version 3 June 2020. (Corresponding author: Pengfei Hu.)
Digital Object Identifier no. 10.1109/TMC.2019.2913832

However, the scheme requires waveform generation with sample rate of 2 giga-samples per second and bandwidth of 240 MHz, making the system difficult to be implemented on off-the-shelf devices and practical applications.

In this paper, we present ColorBars, a system that is designed to improve the data rate of LED-to-camera communication. ColorBars utilizes Color Shift Keying (CSK) for modulating the data where LED transmits different data symbols by emitting light of different colors. It exploits the increasing popularity of tri-LED lights which use three separate red, green and blue LEDs to produce white light. The advantage of the tri-LED is that it allows the generation of a large number of colors by varying the RGB intensity. We show that such color-based modulation of data is perfectly suitable for today's mobile devices given that their cameras can easily capture a large variety of colors, making it feasible to use higher CSK modulation schemes. The use of CSK reduces the symbol duration significantly in comparison with Frequency Shift Keying (FSK) based schemes. The shorter symbol duration along with feasibility of higher CSK modulations result in improved data rate in ColorBars. Compared to [3], ColorBars achieves comparable throughput but only requires a low-profile microcontroller for the transmitter and off-the-shelf smartphones as receivers.

Although CSK is well-suited for LED-to-camera communication, there are three main challenges in design of ColorBars. First, since the LEDs serve a dual purpose of illumination and communication, it is necessary that the color-based modulation does not impact the human perceivable color of the LED. ColorBars eliminates the color flicker problem through development of two schemes (1) adding illumination symbols of white light, (2) Color Sequence Shift Keying (CSSK) modulation. Both schemes are designed to ensure that the perceived light in the critical duration of human eye remains white. Second, current rolling shutter cameras suffer from inter-frame gap where symbols transmitted during the gap are lost. Given shorter symbol duration of ColorBars, this can result in loss of a large number of symbols. We show how error correction coding and packetization can be used to recover the symbols lost due to the inter-frame gap. Third, in order for ColorBars to support commodity camera devices as receivers, it is necessary to address the diversity of the design of their camera sensors. Specifically, different camera sensors interpret the same transmitted color symbol differently due to differences in the type of color filter, its manufacturer and arrangement. To address this issue, ColorBars proposes to use a calibration process with the use of additional management packets sent out by the LED transmitter.

We implement ColorBars on BeagleBone Black embedded platform and tri-LEDs as the transmitter module. We develop an Android application that implements all receiver functionality and evaluate the system using the Nexus 5 and Samsung Note 8 smartphones. We also evaluate ColorBars using iPhone 5S as the receiver. Our implementation confirms that ColorBars can enable reliable and flicker-free LED-to-camera communication at a higher data rate.

The contributions of our paper can be summarized as follows

(1) This paper shows that modulating data using either Color Shift Keying (CSK) or Color Sequence Shift Keying (CSSK) with the use of tri-LED is a feasible and well-suited technique for LED-to-camera communication. Due to the ability of commodity cameras to detect a wide range of colors, it is possible to use higher constellations with CSK and CSSK. This along with shorter symbol duration provides higher data rates compared to previously studied modulation approaches like FSK.
(2) We design ColorBars, an LED-to-camera communication system which addresses many challenges that arise when utilizing CSK modulation. ColorBars ensures *color flicker-free* operation of the LED where transmitted color symbols do not change LED's human perceivable color for consistent white illumination. It shows how error correction codes and packetization can be used to provide reliable communication in the presence of data loss due to inter-frame gap. ColorBars is designed to support commodity cameras available on mobile devices as receivers. It addresses the design diversity in the camera sensors with the use of transmitter-assisted calibration to reduce demodulation errors.
(3) We implement ColorBars on the embedded board with tri-LEDs and evaluate it using Android and iOS smartphones. Our evaluation shows that ColorBars can achieve a data rate of 7.7 Kbps on Nexus 5, 3.7 Kbps on iPhone 5S and 2.9 Kbps on Samsung Note 8. We also show that ColorBars, when using lower CSK and CSSK schemes (i.e., 4 and 8 CSK/CSSK), can achieve moderate to high data rate along with a very low symbol error rate ($< 10^{-3}$) which can provide reliable LED-to-camera communication.

The remainder of the paper is organized as follows. Section 2 discusses the related work. In Section 3, we provide the necessary background. Section 4 details various challenges that arise in using CSK for LED-to-camera communication and provides an overview of our system. Section 5 shows how ColorBars eliminates the color flicker problem. Sections 6 and 7 outline the use of error correction codes for inter-frame loss and transmitter-assisted calibration process respectively. Section 8 describes the demodulation procedure and Section 9 evaluates our system. We conclude in Section 10 with discussion on open challenges.

2 Related Work

LED-to-camera communication has been recently studied in [1], [2], [4], [5], [6], [7], [8], [9]. In [4], authors proposed the use of undersampled OOK (one symbol per frame) in order to ensure reliable data delivery even in highly noisy environment. The use of OOK was also explored in terms of rolling shutter cameras in [5] where the use of shorter symbol duration was proposed. Compared to CSK, OOK suffers severe noise due to ambient light. Also, undersampled OOK can result in flickering effect due to very slow symbol rate. In [1], authors proposed the use of FSK and addresses various issues such unsynchronization, inter-frame gap loss etc. Similarly [2], proposed to use FSK for non-LOS communications. In both works, the achievable throughput in this case is also shown to be no more than 12 bytes per second. The recent works [6], [7] explore the collaborative transmission

TABLE 1
Comparison of LED-to-Camera Communication Schemes, Modulations, Data Rates, and Flicker Considerations

Scheme	Data Rate	Modulation	Flicker
UFSOOK [4]	15 bps	OOK	N/A
[2]	10 bps	BFSK	N/A
RollingLight [1]	90 bps	FSK	N/A
[5]	1 Kbps	OOK	N/A
CeilingTalk [6]	1Kbps	OOK-PWM	N/A
[14]	2.88 Kbps	WDM	Yes
[3]	10.32 Kbps	OOK	N/A
ColorBars	7.7 kbps	CSK	No

among multiple LED luminaries to decrease the destructive interference. [8] presents a VLC protocol for delivering message to a group of random arriving users through careful designed physical layer and link layer. In the recent work [9], authors exploit the polarized light intensity modulation to achieve flicker-free communication. Compared to these, ColorBars uses CSK modulation which reduces the symbol duration and in turn increases the data rate. CSK has been the focus of research in some recent papers [10], [11], [12] and also used in the IEEE 802.15.7 standard [13].

In order to overcome the frame rate limit on CMOS image sensor, [3], [14], [15] apply rolling shutter property of CMOS image sensor to improve the data rate. [14] applies multi-input multi-output (MIMO) technique to mitigate the inter-channel interference for RGB VLC transmission. [15] presents a novel thresholding scheme (named EVA) to synchronize and demodulate the rolling shutter scheme based VLC. A packet reconstruction scheme is proposed and implemented to increase the data rate of LED-to-camera communication in [3], which could achieve a recorded data rate of 10.32 Kbps. Table 1 summarizes the state-of-the-art VLC LED-to-camera communication schemes, modulations used, data rates reported and whether the scheme tackles color flicker or not. In comparison, ColorBars achieves a lower data rate, however, the difference is not due to underlying technique or modulation, but due to our implementation. We use a general-purpose off-the-shelf microcontroller and Linux development board for implementing transmitter. This reduces our achievable symbol rate (maximum number of symbols per second) to 5000 CSK symbols. If this limitation is removed, our scheme is capable of achieving higher throughput.

In another set of research works, screen-to-camera communication has been studied in [16], [17], [18], [19]. Authors in [16] proposed visual-MIMO for enabling communication between pixels of LCD/LED arrays and cameras. Authors in [17] proposed to use blur-adaptive OFDM coding to encode the data in the pixels of LCD display. [18] is designed to achieve communication between smartphone screen and camera using 2-D color barcodes. The limited throughput of previous schemes were improved by LightSync [19] using the transmitter and receiver frame synchronization. In recent work [20], [21], the screen-to-camera communication is shown to be feasible in parallel with screen-human viewing. Many design challenges introduced by LED-to-camera are different from that in screen-to-camera communication. As an example, ColorBars requires the LED to provides white illumination while enabling the communication. Rolling shutter cameras have also been used in visible light localization in [22] and [23] where a camera can receive beacons from multiple LEDs and localize itself using angle or arrival calculation. Compared to these, the primary focus of ColorBars is to provide high data rate and reliable communication in LED-to-camera links.

Authors in [24] present DarkLight system which uses low-cost photodiodes to detect ultra-short, imperceptible light pulse. It could support 1.3m distance with 1.6Kbps data rate. Some works [22], [23], [25], [26], [27] focus on visible light positioning (VLP) which have demonstrated better performance than the RF counterparts. [22], [23], [25] use customized light hardware to transmit visual symbols as landmarks for localization. To be specific, [25] employ the photodiodes to retrieve beacons information and measure the received signal strength from multiple ambient bulbs to calculate the relative distance. [22] uses the camera as the receiver to retrieve landmarks information through processing images of surrounding LEDs. [23] tries to relieve the heavy processing burden on receiver side to let it more capable for IoT device. It proposes the polarization-based modulation, which is flicker-free, to enable a low pulse rate VLC.[26], [27], [28] investigate VLP solution without modifying light fixtures. They extract the intrinsic hidden visual feature of unmodified LEDs and fluorescent lamps as landmarks to achieve high precision indoor localization. Recently, the idea of visible light passive communication has been introduced and exercised in [29], [30], [31]. They investigate the feasibility of practical backscatter communication using visible light for battery-free IoT applications.

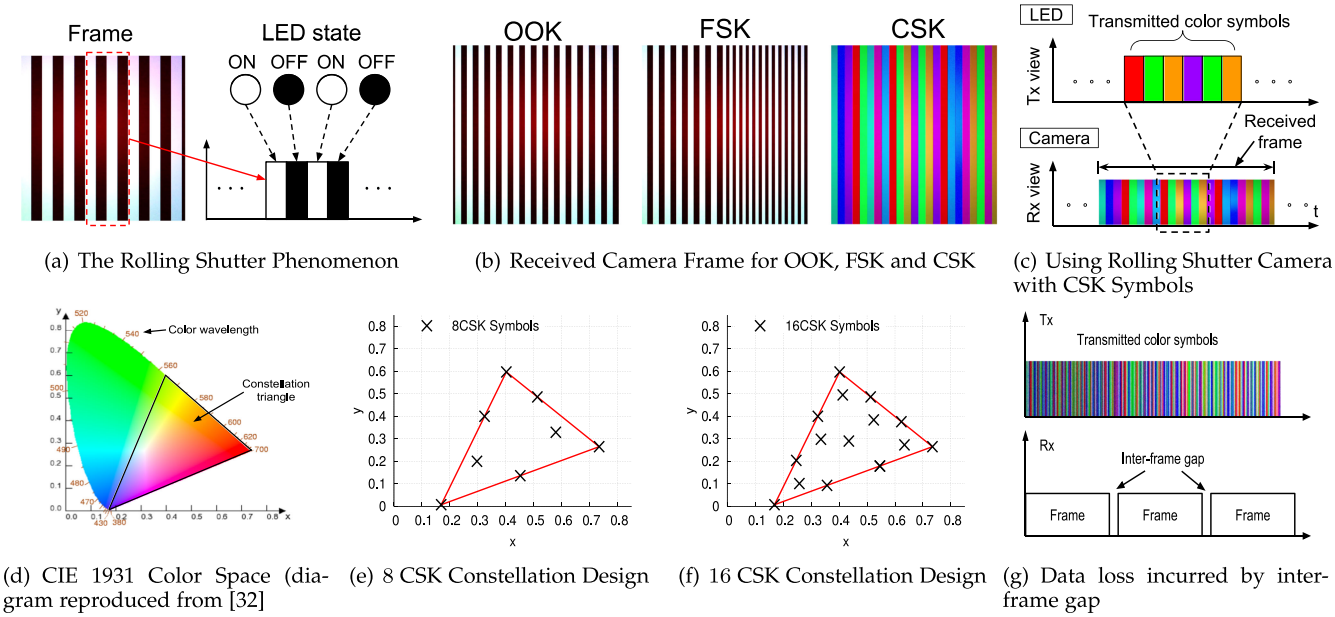
3 Background and Motivation

In this section, we first provide a primer on camera sensor's rolling shutter effect and then discuss how CSK can overcome the limitations of current low-throughput modulation schemes.

3.1 Rolling Shutter, OOK, and FSK

Rolling Shutter. The CMOS image sensor (or camera) most commonly used in today's smartphones, tablets, laptops and other mobile devices exhibit a phenomenon referred as rolling shutter. An image sensor consists of a matrix of photodiodes where each photodiode converts the incident photons to voltage. This voltage is then used to obtain the pixel value of the image frame. In order to reduce the overhead of caching, design complexity, power consumption and cost, the rolling shutter image sensors expose only one scanline of photodiodes at a time and read the output. This scanning of photodiodes, one scanline after another in sequence, is referred as rolling shutter effect. Once all the scanlines of the matrix are read, their output are concatenated to produce an image. Fig. 1a shows an LED that alternates between ON and OFF states, and correspondingly how a camera produces an image frame with alternating bands of pixels with bright and dark shades due to the rolling shutter effect.

On-Off Keying (OOK). The rolling shutter phenomenon enables communication between an LED and a camera where multiple data symbols can be transferred within one camera frame. In OOK modulation, LED ON and OFF states are used to communicate 1 and 0 respectively (Fig. 1b). Because OOK only utilizes LED's white light, it is less



(a) The Rolling Shutter Phenomenon (b) Received Camera Frame for OOK, FSK and CSK (c) Using Rolling Shutter Camera with CSK Symbols

(d) CIE 1931 Color Space (diagram reproduced from [32] (e) 8 CSK Constellation Design (f) 16 CSK Constellation Design (g) Data loss incurred by inter-frame gap

Fig. 1. Rolling Shutter effect, different modulations as received by camera, CSK constellation design, and inter-frame gap.

robust to ambient light noise. OOK can also produce human perceivable LED flickering in the case of long runs 0s or 1s in the transmission data.

Frequency Shift Keying (FSK). To address the limitations of OOK, [2] and [1] have proposed to used FSK where different symbols consist of many ON-OFF bands at different frequencies. Fig. 1b shows a frame with two FSK symbols. FSK reduces the demodulation error due to longer symbol duration and multiple On-Off bands in each symbol. FSK has shown to provide a throughput of 11.32 and 1.25 bytes per second in [1] and [2] respectively.

3.2 Color Shift Keying (CSK)

To overcome the limitations of low achievable throughput, we propose to use CSK modulation for LED-to-camera communication in this paper. CSK was originally proposed by IEEE 802.15.7 standard [13] for visible light communication.

CSK exploits the design of many current commercial LED luminaires which use three separate (red, green and blue) LEDs to generate white light in place of the traditional phosphorescent white LED. With three LEDs, such luminaires can be configured to provide a variety of colors using R, G and B mixture. CSK modulates the signal by modifying the intensity of the three colors. It utilizes color space chromaticity diagram as defined by CIE 1931 [33]. The diagram maps all colors that are perceivable by human eye to two chromaticity parameters – x and y as shown in Fig. 1d. Depending on the operating frequency of the red, green and blue LEDs of the source, a constellation triangle can be formed within the color space. The constellation symbols are then chosen inside the triangle such that inter-symbol distance is maximized for reduced inter-symbol interference. Figs. 1e and 1f show the constellation symbols for 8 and 16 CSK respectively as provided by the IEEE 802.15.7 standard.

On the receiver side, the image sensor receives the color symbols in the form of different color bands in a frame as shown in Fig. 1c. The receiver can map the received color to reference symbol's color for demodulation. When a higher CSK modulation is chosen, each symbol can represent many bits. This can increase the data rate compared to FSK where multiple bands have to be used for one symbol. Depending on the receiver hardware and how many colors the image sensor can capture, higher CSK modulations can be implemented to dramatically improve the data rate. This paper identifies and addresses the major challenges involved in using CSK with rolling shutter camera receivers.

4 Overview of ColorBars

In this section, we first identify the design challenges and then provide the overview of our ColorBars system.

4.1 Design Challenges

When CSK is used for LED-to-camera communication, there are three major challenges.

(1) *Color Flicker*. It is necessary that when an LED is used for data communication, it continues to serve its primary purpose of illuminating the indoor space. Different from OOK or FSK which utilize only white light during the ON period, if the data symbols are transmitted in the form of different color light, the color changes can be perceived by the human eye. Such non-white illumination is undesirable as it changes the rendered colors of surrounding objects, causing a great discomfort to users. Apart from this, fast, human perceivable, changes in color is known to have detrimental physiological effects on humans [34]. Hence, it is required that in ColorBars, even when the color symbols are transmitted, the human perceivable color of illumination remains white.

(2) *Inter-frame Data Loss*. The commodity cameras available in commonly used mobile devices such as smartphones cannot continuously capture image frames. They requires a certain amount of time to process the captured frame as shown in Fig. 1g. The symbols

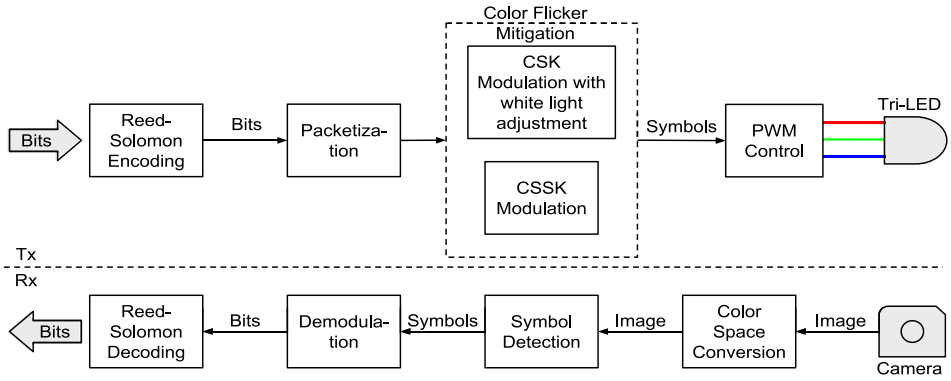


Fig. 2. ColorBars system block diagram.

transmitted by the LED during this inter-frame gap are not received by the camera. In the absence of any uplink communication from the camera to the LED, it is necessary to design techniques that can recover the symbols to ensure reliable communication.

(3) *Receiver Diversity*. When supporting commodity cameras as receivers, it is necessary to take into account the diversity of these cameras in terms of their color filters, their type and arrangement. Due to the diversity, the same transmitted color symbol can be perceived differently by different cameras. It is essential to design a mechanism by which these differences are minimized to reduce the demodulation errors.

4.2 System Overview

We now provide a brief overview of our system. The block diagram of various modules of ColorBars system is shown in Fig. 2. On the transmitter side, the input data bit stream is divided into blocks of bits and error correction coding is applied on each block. ColorBars uses Reed-Solomon encoding to deal with the data loss due to inter-frame gap and inter-symbol interference errors. The blocks of data and parity bits are then used to form packets where a header and a packet delimiter are added to each packet. The encoded bits of the packets are then modulated into a stream of symbols using either CSK or CSSK modulation. For example, when 8CSK is used, the bits are split into pieces of 3-bits and each of the piece is mapped to a color symbol according to the CSK constellation design. For CSK, additional white light symbols are added to the data symbols at this time in order to guarantee flicker-free operations. For CSSK, the bits are mapped to sequence of colors (CSSK symbols). Since the sequences are designed to balance the R(ed), G(een) and B(lue), no additional white light symbols are necessary. The CSK or CSSK symbols are transmitted by the tri-LED which is controlled by a PWM module. The PWM controller generates three separate pulse signals to control the intensity of red, green and blue LEDs in order to produce a specific color.

The ColorBars receiver (camera sensor) captures the symbols transmitted by the tri-LED transmitter in image frames. An image frame contains bands of different colors each representing a transmitted symbol. Each of the captured image is first converted from RGB color space to CIELab color space to reduce the impact of nonuniformly distributed brightness. To reduce the computational overhead of image processing on resource-constrained smartphones, each image is reduced to a single dimension during symbol detection phase. The symbols are identified using color matching process and demodulated using the constellation design. The delimiter sequences are used to form packets. At the end, Reed Solomon decoder is used on the bits of the packets to recover the errors due to inter-frame gap and inter-symbol interference.

Note that when utilizing a high-speed high-gain photodiode as the receiver, the achievable data rate is known to be much higher (refer to [35], [36] for survey). However, in this work, we use rolling shutter cameras as receivers due to their availability in most commonly used mobile devices. More importantly, the use of camera enables a user to visually locate a transmitter LED and receive information specifically from that transmitter. This visual association makes cameras more suitable as receivers. Also, minimizing inter-symbol interference in the CSK constellation design has been studied in [10], [11] with respect to high-speed photodiode receivers. However, designing such optimization for rolling shutter cameras is beyond the scope of this paper. For this work, we adopt the CSK constellation designs provided by the IEEE 802.15.7 standard.

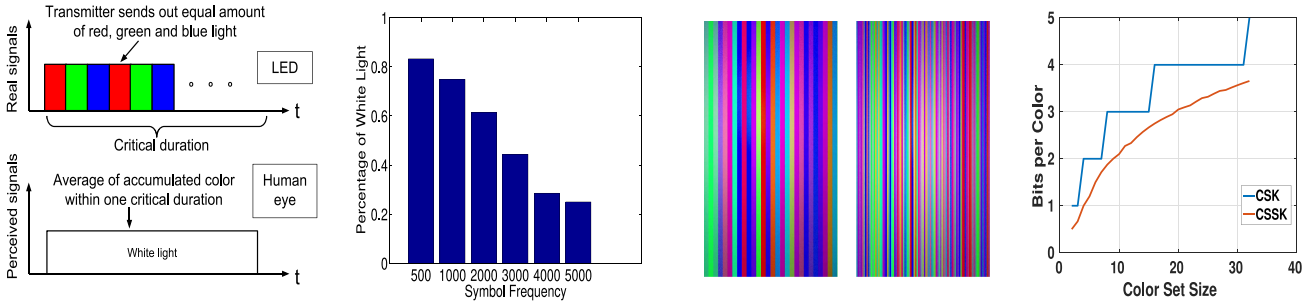
5 Avoiding Color Flicker

In this section, we discuss how ColorBars addresses the color flicker problem. Our eyes can perceive the surrounding objects through the images projected onto the retina. As the brain needs a certain amount of time to "process" the received images, our visual system cannot respond to the immediate change of stimuli, resulting in a delay in observing change in luminance or color. If the intermittent stimuli is presented below a specific rate, our visual system can perceive the changes (an effect referred as *flicker*). Above the rate at which the flicker effect ceases is called the *flicker fusion threshold* [37].

To perceive the surrounding objects, our eyes experience a process called *temporal summation*. During the temporal summation, the eye will accumulate the incoming photons for a period of time until it saturates. This time period is referred as *critical duration*. According to Bloch's law of vision [38],

$$\Psi = I \cdot t, \ (t \leq t_c), \tag{1}$$

where I is the intensity of stimulus (LED in our case) and t_c is the critical duration, the perceived intensity Ψ is a linear function of the duration t. Once the threshold is reached,



(a) Average color perceived by the eye in critical duration

(b) Percentage of white light symbols necessary to prevent color flicker

(c) Width of color bands at 1000 sym./sec. (left) and 3000 sym./sec. (right)

(d) Efficiency of CSK vs. CSSK

Fig. 3. Relationship between CSK symbol frequency and color perceived by human eye, efficiency of CSK versus CSSK.

additional stimuli light does not affect the perception of the visual system.

The perception of the color is the average of the temporal summation during the critical duration. According to the Bloch's law, the perceived color (ψ) is

$$\psi = \frac{\int I_r(t)dt + \int I_g(t)dt + \int I_b(t)dt}{t}, \tag{2}$$

where $I_r(t), I_g(t), I_b(t)$ are the intensity function of the red, green and blue light respectively. This is further demonstrated with an example in Fig. 3a. Here, a tri-LED emits pure red, green and blue light *in sequence* at a very high frequency. Given that the three lights are emitted with the same proportion, the human eye will perceive a white light due to temporal summation within the critical duration.

Note that in our CSK constellation design, the symbols are equally spread within the R, G and B areas of the constellation triangle (Figs. 1d, 1e, and 1f). This means that irrespective of which CSK modulation is chosen, when these symbols are transmitted in equal proportion within the critical duration, they can in fact provide white light. However, the challenge with ColorBars is that the symbols within one critical duration might not provide an average white light perception. This is because, depending on the data, any random symbols can be chosen for transmission and it is possible that the combined effect of symbols might create a color offset that is visible to users.

To address this issue of color flicker, we design two solutions. Both solutions address the problem in different ways. We discuss their advantages and limitations.

5.1 Inserting Illumination Sysmbols of White Light

As shown in Figs. 2 and 4, once the modulation module receives the data stream, it will first calculate how much white light should be inserted to mitigate the color flicker. Then it will group the bit stream to data symbol based on the modulation order (such as 4-CSK, 2 bits will be group to be a data symbol). After that, the ColorBars inserts a dedicated number of illumination symbols of white light in the data symbols.

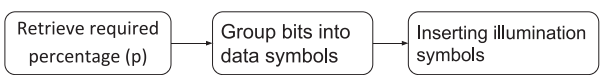


Fig. 4. Insertion of illumination symbols.

Here, for example, when 1 data symbol is followed by 1 illumination symbol, the percentage of illumination symbols is 0.5; or 3 data symbols followed by 1 illumination symbol, the percentage is 0.25. When a sufficient number of periodic illumination symbols are inserted, the perception of white light can be guaranteed. Since the illumination symbols do not serve any purpose for communication, it is desirable to reduce their proportion to increase the achievable data rate. An obvious question here is that how much white light illumination symbols should be inserted. We empirically derive how many white light symbols are necessary to guarantee the white light perception for different symbol frequencies. We perform an experiment where we increase the symbol frequency from 500 Hz to 5000 Hz where each symbol is a randomly chosen color from the constellation triangle. We vary the percentage of white light symbols, and ask 10 volunteers to observe the LED light for color flicker for all symbol frequency and percentage white light combinations. Fig. 3b shows the minimum (as observed from the 10 volunteers) percentage of white light necessary to eliminate any effect of color flicker.

From Fig. 3b, it is interesting to note that as the symbol frequency increases, the percentage of white light necessary decreases. This is because, at a higher symbol frequency, it is more likely that the symbols in each critical duration are more uniformly distributed within the constellation triangle, resulting in a combined effect of white light. At lower symbol frequencies, longer symbol duration might create a color offset from the white light, requiring more white light symbols for illumination adjustment. This result shows that operating an LED at higher symbol frequency is better for two reasons: (i) as the symbol frequency increases, the achievable data rate increases and (ii) at a higher symbol frequency, fewer white light symbols are necessary for illumination which in turn also increases the data rate as more data symbols can be transmitted within a unit time. We use the results of Fig. 3b to determine the necessary percentage of white light in rest of the work.

Although higher symbol rate is desirable for higher data rate, there are two limiting factors (1) the hardware limitations of the transmitter, such as the computation capacity of controller, the maximum frequency of LED etc.; (2) the limitation of receiver's hardware and the data processing algorithm. With the use of rolling shutter cameras as receivers, the second factor imposes a major limitation on LED's symbol frequency. As the symbol frequency increases, the size of the symbol received (width of the color band in the frame) by the

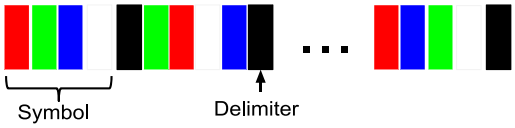


Fig. 5. CSSK symbols separated by delimiters.

camera decreases (see Fig. 3c). Once the width falls below a specific value, it becomes extremely difficult to correctly demodulate the symbol. From our experiments, we empirically find that minimum width of the band should be 10 pixels to avoid any symbol detection error.

5.2 Color Sequence Shift Keying (CSSK)

Inserting white illumination symbols creates a simple flicker mitigation solution, but it has a few drawbacks. First, since the white light symbols don't serve any purpose in communication, they waste the available bandwidth and reduce the achievable data rate. Second, the proportion of additional white light symbols needed is at odds with the symbol frequency and demodulation error rate. At lower color frequency, the demodulation rate is lower but more white light symbols are needed to ensure flicker mitigation. In order to reduce the necessary white light symbols, we need to use higher color rate but that increase the demodulation error rate. To address these limitations, we present a new modulation referred as Color Sequence Shift Keying (CSSK).

CSSK is based on CSK but instead of using each color as a symbol, CSSK uses a sequence of colors as a symbol. The idea behind the design of CSSK is that it is difficult to achieve a balance between R, G and B using CSK because it is dependent on which symbols are transmitted within the critical duration. Since each symbol is composed of multiple color in case of CSSK, such a balance is easier to achieve even with any arbitrary set of symbols. Formally, let N be the size of a color set. With N colors, the total number of the permutation sequence is $N!$. Instead using each color for a symbol as in CSK, here we adopt each sequence as one symbol. For example, if the color set is $\{RED, GREEN, BLUE, WHITE\}$, the total number of permutation of these 4 colors would be $4! = 24$, i.e., there are 24 symbols. As shown in Fig. 5, the sequence *RED-GREEN-BLUE-WHITE* could be one symbol. In case of CSSK, an additional symbol is needed to delimit the two sequences. We use an OFF (BLACK) symbol as a delimiter as shown in Fig. 5. Note that in CSSK, a per-symbol delimiter is used. Due to the inter-frame loss (Section 6), it is possible that the transmitted symbols are completely lost or only partially received. In this case, per-symbol delimiter enables the receiver to synchronize with the transmitter.

Since CSSK uses a sequence of colors for a symbol, the number of bits represented by a color decreases compared to CSK. As the size of symbols space is $N!$ in CSSK, each symbol could represent $log_2(N!)$ bits. Hence, the number of bits n each color could represent is

$$n = \frac{log_2(N!)}{N}. \tag{3}$$

For CSK, each color could represent

$$n' = log_2 N, \tag{4}$$

It's not hard to find that $n' > n$, i.e., the efficiency of the CSSK scheme is worse than that of CSK. This is shown in Fig. 3d.

Although CSSK has lower spectral efficiency than CSK, it has two benefits in comparison. First, CSK requires inserting white light illumination symbols which in turn reduces the data rate. With CSSK, no additional white light symbols are required. Second, the ability of CSSK to achieve white balance is independent of the color rate. Compared to CSK, the length of a CSSK symbol is longer (number of colors in a CSSK symbol plus one OFF symbol for delimiting). However, without the additional white light symbols, throughput of both the schemes is comparable as we will evaluate in Section 9. Note that our objective for designing CSSK is to maintain the data rates offered by the original CSK scheme while guaranteeing no color flicker. As we will observe in evaluation, sequence-based symbols reduce the symbol demodulation errors, resulting in CSSK outperforming CSK in terms of goodput.

6 Inter-Frame Data Loss

As we mentioned before, a camera requires a certain amount of time to process the each captured frame, there exists a time gap between the two consecutive frames when the information transmitted by the LED is lost. We refer to this as inter-frame data loss. In this section, we show how we can use error-correction coding and packetization to ensure reliable data transfer.

Error Correction Coding. The inter-frame data loss can be recovered using an error correction coding scheme. To apply an error correction coding, ColorBars first divides the bit-stream into blocks of k bits. Due to unidirectional communication from LED to camera and unsynchronization between the two, the inter-frame data loss can occur at any part of the block of k bits. To handle such error characteristics, ColorBars uses Reed-Solomon (RS) codes for error correction. RS codes are block-based error correction codes that are widely used in wired and wireless communication as well as the storage systems. In $RS(n, k)$ coding, a codeword of n bits is generated by adding $n - k$ parity bits to the k data bits. Such an RS encoding can detect errors in up to $2t$ bits and can correct up to t bits where $2t = n - k$. RS codes are especially suitable for ColorBars as it can detect and correct bit errors anywhere within the codeword of n bits.

It is noted that the computational overhead of encoding and decoding increases sharply [39] as k increases. In ColorBars, the size of k and n should depend on the inter-frame gap of the receiver. Let us say that for a symbol rate of S symbols per second (sym./sec.), the inter-frame loss ratio is l and the frame rate is F. The inter-frame loss ratio is the ratio of size of inter-frame gap to the total size of a frame and an inter-frame gap. Hence, the number of symbols received in one frame by the receiver is $F_S = (1 - l) \cdot S/F$. This is shown in Fig. 1g. The additional illumination symbols of white light are added with illumination ratio of η_S. The illumination ratio is the ratio of number of useful data symbols to total number of data and white light symbols. Also, let us say that for the given CSK/CSSK scheme in use, the size of the symbol is C bits. This way, the number of symbols lost between two consecutive frames is $L_S = l \cdot S/F$, and the total number of data bits lost is $\eta_S \cdot C \cdot L_S$.

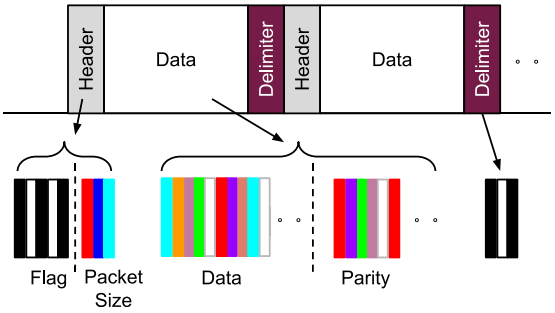


Fig. 6. Data packet structure.

We can get the size of codeword $n = \eta_S \cdot C \cdot (F_S + L_S)$. In order for recovering these bits, the RS coding with $t = \eta_S \cdot C \cdot L_S$ should be used, which means the parity size should be $2t = 2\eta_S \cdot C \cdot L_S$. Hence, ColorBars choose $k = n - 2t = \eta_S \cdot C \cdot (F_S - L_S)$. For example, if there are 150 bands within one receiver frame ($F_S = 150$ symbols) and the number of lost bands is 30 (the loss ratio is $1/6$), the transmitter is using 8CSK ($C = 3$ bits) and $\eta_S = 4/5$ (20% illumination symbols), the size of the message is calculated to be 36 bytes. This way, the error correction coding used by ColorBars can be indicated by $RS(\eta_S C(F_S + L_S), \eta_S C(F_S - L_S))$. Note that in ColorBars, if the CSK modulation scheme is adopted, the bitstream is first mapped to symbols based on the chosen CSK modulation scheme, and the "white" symbols are added afterwards to the encoded symbols, whereas the bitstream is mapped directly to symbols if CSSK modulation is adopted.

Packetization. The data and parity symbols along with the illumination symbols are encapsulated in a data packet. A header is attached to each of the packet which includes the size of the packet. Note that if the size of the packet is too small, the entire packet can be lost during the inter-frame gap and additional techniques might be necessary to recover the packet. On the other hand, if the packet is too large, even when only the header of the packet is lost during the inter-frame gap, the resultant data loss can be much larger. In this case, a natural choice of size of the packet $p = C(F_S + L_S)$ which is the total size of a frame and inter-frame gap.

Since the receiver is required to perform decoding and demodulation on each packet, it is necessary to delimit each packet using a pre-defined delimiter sequence. ColorBars uses "owo" sequence as delimiter between packets where "o" and "w" are LED OFF and white light symbols respectively. During the OFF symbol, the LED is turned off to produce a dark symbol that can be easily identified and distinguished from other data symbols. The symbol duration of the OFF symbol is also chosen to be the same as other symbols based on the symbol rate. The delimiter along with the packet structure is illustrated in Fig. 6.

Apart from the packet delimiter, each packet is assigned a header which includes two fields - (1) a flag indicating that the packet is either a data packet or calibration packet (Section 7) and (2) size of the packet. A data packet is indicated by a flag of sequence of five symbols "owowo", while the size of the packet is indicated using 3 data symbols. The size of the packet is required in the header as it allows the receiver to determine how many bits were lost in the inter-frame gap and apply RS decoding accordingly. If either the delimiter or the packet header is lost inter-frame gap, the packet is discarded.

7 Receiver Diversity

One of the biggest challenges in utilizing color-based modulation scheme is to address the diversity in camera receivers. The camera image sensor used in different smartphones or other devices can vary significantly in their characteristics, especially, in how they capture different colors. For ColorBars, the different color symbols broadcast by the LED transmitter should be correctly demodulated by all the camera receivers. Since ColorBars is a unidirectional broadcast communication system, it is impossible to know the configuration of a receiver. There are two issues here. First, receiver cameras can be vastly diverse in terms of how they perceive/capture the transmitted color. We address this issue using calibration packets in this section. Second, camera receivers have different inter-frame loss ratios (Section 6). We believe that this is a compatibility issue where the system designer can choose to support a certain inter-frame loss ratio, and all camera receivers with inter-frame ratio lower than that can receive the transmitted data (albeit at a slightly lower data rate). In the evaluation, we assume that inter-frame loss ratio is known and transmitter setting (coding rate) is configured accordingly. Hence, it is not required to have different transmitter settings for different receivers, but doing so can achieve higher throughput. In this section, we first identify the issues that arise due to camera diversity and how transmitter can enable receiver-side calibration for improved demodulation.

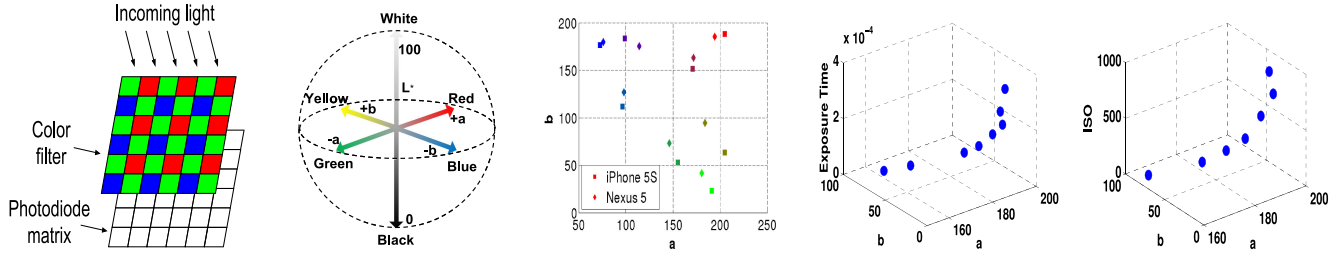
7.1 Different Cameras, Different Symbols

An image sensor consists of a matrix of photodiodes. Since a photodiode can only perceive the intensity of the light and not the color itself, the images captured using the sensor will be grayscale images. In order to estimate the colors in the image, each photodiode is covered with a color filter. A commonly used color filter is Bayer filter (shown in Fig. 7a). A Bayer filter is matrix of filters with alternating rows of green-red and green-blue filters. The higher number of green filters is due to the fact that human eye is more sensitive to the green color wavelength. Depending on the intensity of different colors after filtering and demosaicing procedure, the true color of the pixel is estimated in the image. For different image sensors, the color filters (technology and manufacturer), their arrangement and the demosaicing procedure can be different which results in different cameras estimating the same true color differently.

Fig. 7c shows how 8 different color symbols (8CSK) transmitted by an LED transmitter are received by the camera sensor of two different smartphones (Nexus 5 and iPhone 5S). It can be observed that there is a noticeable difference between how the same color is perceived by two different cameras. As we discussed, this is attributed to different color filters used by the camera sensors.

7.2 Same Camera, Different Symbols

There is another challenge introduced by the camera sensor diversity issue. Because an image capture is controlled by many different parameters of the camera sensor (e.g., exposure time, ISO etc.), the same symbol when transmitted at a different time can be received differently by the same camera. The camera sensor in most modern smart devices adjust the exposure time and ISO automatically and dynamically



(a) The Bayer Color Filter (b) CIELab Color Space (c) Same color can be perceived differently by different cameras (d) Effect of exposure time on the perceived color (e) Effect of ISO on perceived color

Fig. 7. Camera's perceived color can vary substantially due to variations in camera filter and settings.

depending on the current ambient light condition. This can lead to significant variation in received symbols spatially as well as temporally. The exposure time is the length of the time the camera shutter is open to allow the light into the photodiode matrix. Larger exposure time means that each photodiode will have more time to accumulate photons until it saturates. ISO is a parameter that determines how many photons are enough to saturate the photodiode. For a higher ISO, lesser number of photons are needed to saturate the photodiodes. Figs. 7d and 7e show how the same transmitted color symbol (pure blue) can be perceived differently with different exposure time and ISO respectively.

Calibration Packet. ColorBars handles the receiver diversity issues with the use of dedicated management packets called the calibration packets. The calibration packets are sent out by the transmitter periodically (n times per second). A calibration packet includes a flag and all symbols of current modulation scheme in a sequence (e.g., 8 symbols for 8CSK). Similar to the data packets, the flag is used to identify the calibration packets. We use "owowowo" sequence as the flag indicator for the calibration packet where "o" and "w" are LED OFF and white light symbols respectively. The choice of OFF and white symbols ensures that a receiver can infer an incoming calibration packet even before it has received the color sequence for calibration. Once a receiver receives a calibration packet, it stores each of the symbols and its color for future matching. Since the calibration packets are sent out periodically, the receivers can quickly adapt to the changing channel condition (i.e., ambient light) to reduce the symbol demodulation error. A new receiver joining the system can wait till the reception of the first calibration packet to start demodulating the data.

8 Demodulation

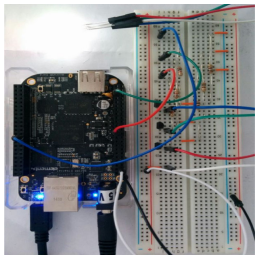
The color symbols transmitted by an LED are received by the image sensor which can demodulate them to receive the data. The receiver captures a continuous set of frames through video recording and then extracts the symbols in each frame. Note that in ColorBars, the transmitter uses CIE color space (Fig. 1d) as its basis for constellation design because it allows to choose a set of symbols that are equally distributed among all RGB wavelengths and together can produce white light which is necessary for illumination constraint. However, the receiver can demodulate the transmitted symbols using any color space that can reduce the symbol error. A naive way of matching the color of symbols is to use RGB color space and apply a distance metric. Although it is intuitive, this method has severe limitations in terms of removing brightness from the received color. The use of CIELab color space is better in demodulation as it can distill symbol's color by removing most of the effects of brightness. CIELab is a three channel color space with one channel for lightness L and two channels for colors (a and b). CIELab was designed to overcome the limitations of RGB space in which the distance between two colors does not always correspond to separation perceived by the human eye. The CIELab color space is shown in Fig. 7b. The a dimension spans from green ($-a$) to red ($+a$) and the b dimension spans from blue ($-b$) to yellow (b). The scaling of a and b axes depends on the specific implementation and normally range from ± 100 or -128 to $+127$. We adopt the latter in this paper. The vertical axis L captures the brightness by spanning from black to white. Hence, ColorBars uses CIELab color space for demodulation on the receiver side. After removing the brightness dimension, any color can be represented by $\{a, b\}$ as shown in Fig. 7b.

Fig. 8. Calibration packet.

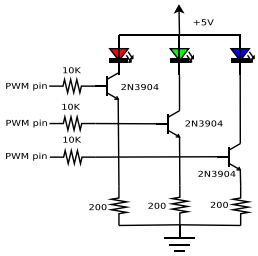
It carries out the following steps in order to demodulate data from each frame -

Step 1 - Convert to CIELAB Color Space. Fig. 10a shows a received frame to demonstrate the brightness within the frame is not uniformly distributed. The center of the frame is observed to be brighter compared to the peripheral region. As we discussed before, the RGB values vary considerably within a band due to the non-uniform brightness. To address this, ColorBars converts the received color to CIELAB color space and eliminates the brightness dimension to distill the symbol color using $\{a, b\}$.

Step 2 - Preprocessing Packets. In order to decrease the computational overhead of demodulation, the receiver reduces a 2D frame containing the color bands to a single dimension. Let $P[i, j]$ be the color of the pixel at ith row and jth column of the frame, and let M be the number of pixel rows in the image. Note that $P[i, j] = \{a, b\}$ in the CIELab space without the lightness dimension. We first calculate the mean color of jth column by averaging the a and b of all M pixels in the column. The mean color of each column is used as a way to reduce the frame to an array of pixels. After this, any delimiters between the packets are detected based on sequence

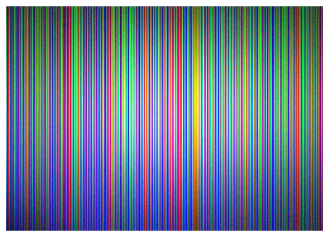


(a)Transmitter Components

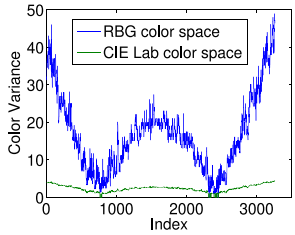


(b) Transmitter Circuit

Fig. 9. LED transmitter platform.



(a) Brightness is non-uniformly distributed in received image frames



(b) Variance of color at different positions in RGB and CIE Lab

Fig. 10. Impact of non-uniform brightness can be reduced by conversion to CIELab space.

matching. After splitting the packets, if CSK modulation is used, the illumination symbols (white light) are also removed. In case of CSSK, the black color delimiters are used for separating the color sequences. Note that detecting and distinguishing OFF and white symbols is possible with very high accuracy. Each packet is then marked as either a data or a calibration packet using the flag information in the packet header.

Step 3 - Decoding and Demodulation. For the calibration packets, the sequence of color symbols are stored by the receiver along with the number of symbols used by the CSK or CSSK scheme. These symbols are used as reference symbols for demodulating the symbols in the data packets. We use ΔE [40] to measure the color difference between each pixel and reference symbol. ΔE is the euclidean distance between two colors in the a, b-plane of the CIELab space.

$$\Delta = \sqrt{(P_a[i] - R_a[j])^2 + (P_b[i] - R_b[j])^2}, \tag{5}$$

where where $P_a[i], P_b[i]$ are the two dimensions (a and b of CIELab color space) of the mean color of pixel $P[i]$, while $R_a[j], R_b[j]$ are the two dimensions of the reference color symbol. It is known that the difference between two colors is noticeable when $\Delta E \geq 2.3$ [40]. We use 2.3 as the threshold to match the color of a symbol to reference the colors received in the calibration packet.

For each data packet, the first essential information to be decoded after demodulation is the size of the packet available in the header. If the number of symbols received in the packet matches the size mentioned in the header, it means that the packet did not suffer any inter-frame loss. In this case, the RS decoding fixes any symbol error induced by color matching procedure. If the number of received symbols are lesser than the header size, the RS decoding recovers the data lost due to the inter-frame gap. Recall that our RS encoding was designed to ensure that the data can be recovered correctly in the presence of inter-frame loss.

9 Performance Evaluation

Experiment Setup. The ColorBars transmitter is implemented using BeagleBone Black board [41]. BeagleBone Black is a low-cost, open hardware embedded development platform with 1 GHz processor and 512 MB of memory. The BeagleBone platform is especially suitable for implementing ColorBars as it provides sufficient number of PWM controls necessary for generating different colors through off-the-shelf RGB tri-LED. It has also been recently used for developing open-source VLC testbed in [42]. We empirically find the maximum frequency of color change supported by the BeagleBone board to be less than 5000 Hz. Various components of the transmitter are depicted in Fig. 9a and the circuit diagram of the transmitter module is shown in Fig. 9b. Because the lumens of the tri-LED used in our experiments is low, the smartphone camera is kept very close to the LED (within 3 cm) in order to allow the LED to fill the entire view of camera. However, this distance limitation can be removed if a LED with strong luminance is used. Whether the light could fill the camera's view or not depends on the Received Signal Strength (RSS) and cameras settings (ISO and exposure time). For example, if both the RSS and ISO are high, the light will fill the entire camera even though transmitter is at a farther distance. We further explore the relationship between illuminance and distance analytically later in this section.

The ColorBars receiver module is implemented on smartphones. In order to evaluate multiple commodity cameras, we use Android (Nexus 5, Samsung Note 8) and iOS (iPhone 5S) smartphones. The Nexus 5 camera sensor has a resolution of 2448×3264 and frames per second is 30. The resolution setting of Samsung Note 8 is 1080×1920 with 30 frames. The iPhone 5S camera has a resolution of 1080×1920 with 30 frames per second. Note that we use the same resolution settings and the back camera of all the devices in our experiments. For the Nexus 5 and Samsung Note 8, we develop an Android app that implements the functionality of ColorBars receiver including decoding, image space conversion, symbol detection, demodulation and error correction. We use multiple threads to speed up the real-time decoding process. In the implementation, one thread is used to read frames from the camera[1] and perform color space conversion along with dimension reduction. The pre-processed image is then added to a queue, from where another thread receives the frames. This thread performs the symbol detection, demodulation and error correction. In the case of iPhone 5S, we capture the video using the device and perform the decoding procedure offline.

Color Flicker Mitigation. We evaluate how effectively CSSK can mitigate the color flicker compared to CSK. For a quantitative comparison of flicker mitigation performance, we introduce a new metric called *flicker distance variance*. The flicker distance is calculated as the distance between the observed accumulated color from the white color on the CIELab color space The flicker distance is calculated for

1. Android (started from Lollipop 5.0) allows smartphones to take raw images. However, the compressed format is more suitable for our application. The read-out time of raw data is shorter than compressed data, but the processing time will be much longer.

TABLE 2
Flicker Distance Variance

Color Frequency	Color Space	CSK with white light	CSSK
500 Hz	4	111.34	0 (N)
	8	81.32	2.83 (N)
	16	65.92	2.79 (N)
	32	57.02	8.02 (Y)
1000 Hz	4	55.56	0 (N)
	8	42.15	2.15 (N)
	16	34.40	2.45 (N)
	32	29.92	2.64 (N)
2000 Hz	4	27.84	0 (N)
	8	22.58	0 (N)
	16	18.60	0 (N)
	32	16.33	0.52 (N)
3000 Hz	4	18.56	0 (N)
	8	15.99	0 (N)
	16	13.41	0 (N)
	32	11.84	0 (N)
4000 Hz	4	13.87	0 (N)
	8	12.65	0 (N)
	16	10.66	0 (N)
	32	9.52	0 (N)

time windows of critical duration size (30 ms [43]) and a variance is calculated over the windows.

Table 2 shows the comparison of the two schemes – CSK with white light symbols and CSSK in terms of flicker distance variance. We also vary the color rate from 500 Hz to 4000 Hz to understand its impact on flicker distance. For the symbol rate of 500 Hz and 4 color (RED, GREEN, BLUE, WHITE), the CSK scheme has the variance of 111.34 while variance of CCSK is 0. As expected, the variance decreases as the symbol rate increases in the case of CSK with white light symbols. While in case of CSSK, the variance is low at all symbol rates. This is because the symbols are already a sequence of component colors, the R, G and B balance brings the accumulated color very close to the white color.

We also perform subjective evaluation to check if CSSK with a given configuration mitigates color flicker from user's perspective. As before, we ask 10 volunteers to visually identify any color flicker that they observe while the LED transmits the color sequence symbols. Table 2 shows the Y(es) and N(o) based on whether majority of volunteers observe a color flicker or not for each CSSK configuration. Note that after adding white light symbols, CSK scheme only occasionally shows color flickers (depending on the color symbols selected for transmission). We observe no color flickers in case of CSSK except for $500\ Hz$ 32-CSSK configuration. This exception is because in this configuration only partial 32-CSSK symbol can be transmitted within the the critical duration of 30 ms, resulting in a high flicker distance variance. This shows that CSSK can mitigate color flicker without the need of any additional white light symbols.

We evaluate the ColorBars transmitter and receiver modules using three performance metrics - symbol error rate, throughput and goodput. All the performance experiments are conducted under the normal indoor illumination condition ($250\ lux$). The measured illumination of the RGB LED transmitter ($1\ cm$ from transmitter) is $2000\ lux$.

Inter-frame Loss Ratio. We first measure the inter-frame loss ratio of the three smartphone receivers. In order to determine the inter-frame loss ratio, the transmitter sends out color-band symbols at different rates and the device cameras record the received symbols. Table 3 shows the number of color-band symbols received per second for the three devices. Based on this, we can observe that Samsung Note 8 has the highest inter-frame loss ratio (0.39) compared to Nexus 5 (0.23) and iPhone 5S (0.37).

Color Space Conversion. The advantage of converting the received frame from RGB color space to CIELab space is to remove the majority of the effect of brightness variation within the same color symbol. Fig. 10b compares how well CIELab space can remove the brightness variations compared to the RGB space. For this, we choose a color symbol in the center of the frame shown in Fig. 10a, and calculate its mean color in both spaces. We then calculate the variance of euclidean distance from each pixel's color in the symbol column to the mean color of the column to observe the variation from the mean due to brightness artifacts. CIELab space observes much smaller variance due to removal of most of the brightness effects compared to the RGB space.

Energy Consumption. We measure the energy consumption of ColorBars on Nexus 5 and Samsung Note 8 devices. We use Android system's default battery monitor to measure the energy. Given that smartphone energy consumption measurement can be extremely complex, we put the smartphones in airplane mode to exclude any energy consumption from network and turn off the screen. We then run the ColorBars application which involves camera, image processing and CPU. Since it is not possible to measure these components separately, we measure battery depletion when the application is running for 1, 2 and 3 hours. Fig. 11 shows the battery consumption in percentage. Although it is extremely challenging to isolate consumption just due to ColorBars application, these measurements provide important information about how the battery depletes when ColorBars is running. Based on this measurement, we can estimate that with a fully charged battery, ColorBars can run more than 9 hours continuously.

Symbol Error Rate. The evaluation of Symbol Error Rate (SER) shows the demodulation errors experienced by the receivers due to the inter-symbol interference. We measure

TABLE 3
Average Inter-Frame Loss Ratio of Nexus 5, iPhone 5S, and Samsung Note 8

Transmission Symbol Rate	1000 Hz	2000 Hz	3000 Hz	4000 Hz	Avg. Inter-frame Loss Ratio
Nexus 5	772.84	1506.11	2352.65	3060.67	0.2312
iPhone 5S	640.55	1263.56	1887.73	2431.01	0.3727
Samsung Note 8	602.34	1206.42	1803.83	2407.87	0.3978

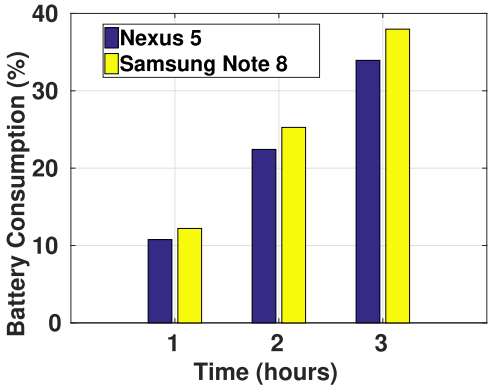


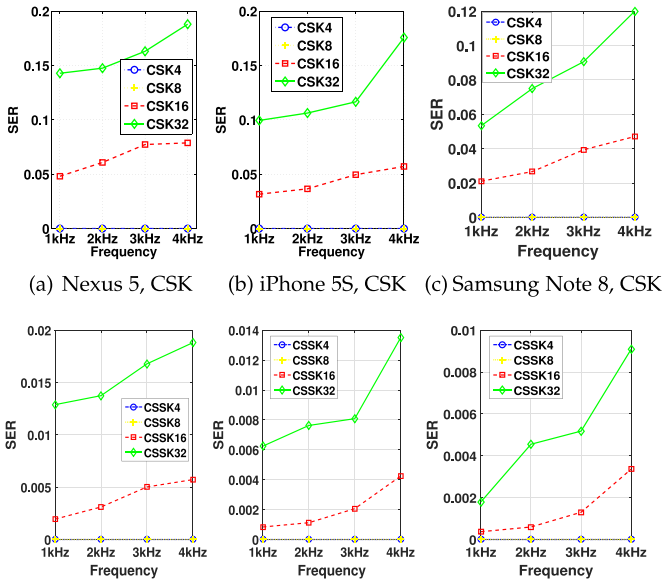
Fig. 11. Energy Consumption of ColorBars.

the SER in CIELab space for Nexus 5, Samsung Note 8 and iPhone 5S smartphones for both CSK and CSSK schemes with different orders. To capture the effect of color frequency (color per second), we vary it from 1000 Hz to 4000 Hz in the increments of 1000 Hz for both CSK and CSSK schemes. We do not modify the exposure time or ISO settings in either of the cameras to allow it to adjust these parameters automatically as it happens in most practical scenarios.

Figs. 12a, 12c, and 12b show the observed SER for Nexus 5, iPhone 5S and Samsung Note 8 respectively when CSK modulation scheme is used. While Figs. 12d, 12e, and 12f show the SER when CSSK modulation scheme is used. Here, the SER is the fraction of symbols that are incorrectly demodulated by the receiver. We observe that for all the smartphones, as we increase the color frequency, the SER increases for higher CSK/CSSK schemes (i.e., 16 and 32). This is because with higher color frequency, the size of the color (width of the band) decreases. This increases the inter-symbol interference as it becomes more and more difficult to distinguish color with fewer pixels of true representation of the colors. It is also observed that the SER in lower CSK/CSSK schemes (i.e., 4 and 8) is close to 0, which means that all the camera devices can accurately distinguish (4 or 8) colors even at a very high color frequency. This means that lower CSK/CSSK modulation schemes can be used in the applications where reliable LED-to-camera communication is desirable.

We observe that CSSK modulation achieve much lower SER compared to the CSK modulation. This is because demodulation is CSSK matching a sequence of colors as compared to CSK where only one color is matched. It is also observed from Fig. 12 that iPhone 5S achieves a lower SER compared to the Nexus 5 smartphone. This can be purely attributed to how well a camera sensor captures the true color emitted by the LED. We observe that iPhone 5S better captures the true color, allowing the receiver to better distinguish the symbols.

In order to evaluate the effect of distance on the SER performance of CSK and CSSK, we vary the illuminance of the transmitter to mimic the distance change. The direct result of distance change is the illuminance changes on receiver side, which could also be achieved by varying the transmission illuminance. The reason why we vary illuminance instead of distance is that in our current implementation, it is difficult for a smartphone camera to focus on tiny LED used in our implementation. The limitation can be removed by using a larger LED or equipping the camera with an optical lens. If the camera is equipped with an external optical lens, the



(a) Nexus 5, CSK (b) iPhone 5S, CSK (c) Samsung Note 8, CSK

(d) Nexus 5, CSSK (e) iPhone 5S, CSSK (f) Samsung Note 8, CSSK

Fig. 12. SER versus frequency.

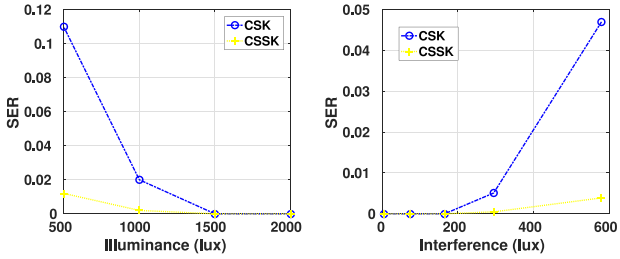
transmission distance between the light source and the camera can be significantly extended. The illuminance and distance relationship can be calculated as follows. Let us assume that the luminous flux emited by an LED is F_T. The received flux (F_R) depends on receiver's relative position and orientation as shown in Fig. 14. Here, the angle between transmitter's normal axis and the transmitter-receiver line is referred as irradiation angle (β). The angle between receiver's normal and the transmitter-receiver line is called incident angle (α). Let the distance between transmitter and receiver be D. Since a light sensor is a photodiode with a concentrator lens, the voltage generated by the photodiode is proportional to the area of the photodiode where the photons are collected. It is known from [44], [45] that the received F_R can be calculated using the following equation

$$F_R = F_T \times \frac{(m+1)A}{2\pi D^2} cos\,\alpha\ cos^m\beta, \tag{6}$$

where m is the order of Lambertian emission. Most commonly used LEDs are pure Lambertian emitters where $m = 1$. In our experiments since user's camera is close to the LED, we can assume $\alpha = 0$ and $\beta = 0$. In this case, the received luminance decreases with increasing distance.

Based upon the illuminance-distance relationship, we vary the illuminance of transmitter from 500 to 2000 lux to mimic the distance change, and measure the SER on Samsung Note 8 with CSK and CSSK. The experiment is in the dark environment to prevent the interference (we study ambient light interference separately below). Fig. 13a shows the result. With high illuminance, both CSK and CSSK achieve lower SER. However, the SER of CSK drops sharply when illuminance decreases under 1000 lux. The CSSK shows much better SER performance even under low illuminance. This can be attributed to the fact that sequences of colors used as symbols provide better demodulation performance as the distance increases or illimunance decreases.

Interference is one of the most challenging issue for VLC, as there can be many ambient interference sources in a real



(a) SER vs. Illuminance (b) SER vs. Interference

Fig. 13. SER with varying illuminance and interference.

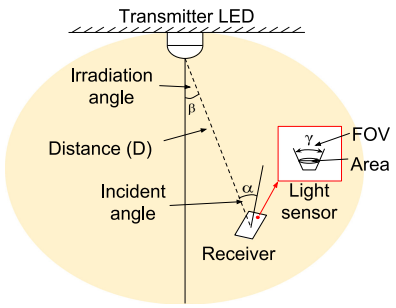
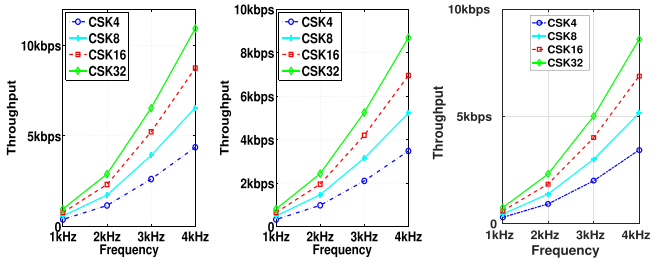
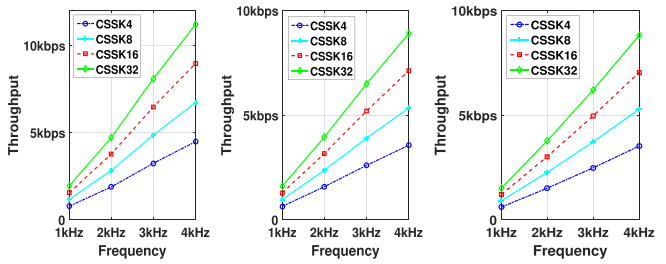


Fig. 14. Impact of distance on received illuminance.



(a) Nexus 5, CSK (b) iPhone 5S, CSK (c) Samsung Note 8, CSK



(d) Nexus 5, CSSK (e) iPhone 5S, CSSK (f)Samsung Note 8, CSSK

Fig. 15. Throughput of different CSK/CSSK modulations and color frequency.

world. To evaluate the effect of interference on the performance of CSK and CSSK, we placed one LED besides the transmitter as an interferer. The illuminance of the interferer is varied from 5 to 580 *lux*. We measure the SER on Samsung Note 8 with CSK and CSSK. Fig. 13b shows that the SER will increase along with the increase of interference as expected. We could observe that SER of both CSK and CSSK are extremely low even when the interference is high. This demonstrates that both CSK and CSSK can provide reasonable demodulation performance even in presence of surrounding interference. We also note that mitigating interference requires frequency domain analysis (calculation and removal of lower frequency components) [46]. Currently, ColorBars does not employ any such interference removal, but incorporation of such schemes in our system can substantially improve the performance of our system.

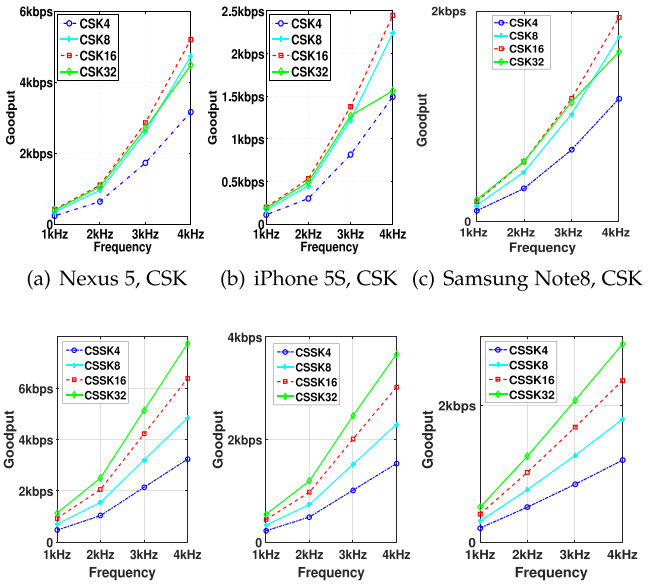
Throughput and Goodput. We now evaluate the throughput and goodput that is achievable through ColorBars system. Similar to the SER results, we vary the modulation and symbol frequency to investigate their impact on throughput and goodput, with automatic camera settings of exposure time and ISO. It is also note that currently ColorBars transmitter sends out 5 calibration packets per second. The overhead of the calibration packet is observed to be negligible due to very small size of the packets (e.g., less than 50 symbols for 32 CSK/CSSK).

Figs. 15a, 15b, and 15c show the raw achievable throughput data rate for the Nexus 5, iPhone 5S and Samsung Note 8 devices respectively with CSK modulation while Figs. 15d, 15e, and 15f show the same for CSSK modulation. For the calculation of raw throughput, we do not perform any error correction at the receiver but simply measure the number of symbols received. In case of CSK with white illumination symbols, we do not count the illumination symbols towards the throughput. Similarly, in case of CSSK, we do not count the black delimiter symbols towards the throughput. As shown in Fig. 5, the encoding efficiency of CSSK is lower than CSK, however, this low efficiency is offset by the waste bandwidth of white light in CSK modulation, which makes the throughput of these two kinds of modulation similar to each other. We can observe that the throughput increases with increase in the symbol frequency as expected. Similarly, in the absence of any error correction, higher CSK/CSSK modulation schemes achieve higher throughput.

The reason why iPhone 5S and Samsung Note 8 achieve lower throughput compared to Nexus 5 is because the throughput depends on two factors - SER and inter-frame loss ratio. Even though iPhone 5S and Samsung Note 8 have lower SER, their inter-frame loss ratio is much higher (Table 3) in comparison with Nexus 5. This means that more number of transmitted symbols are lost for the iPhone and Samsung Note 8, resulting in overall throughput reduction.

Figs. 16a, 16b, and 16c show the observed goodput for three smartphones when CSK modulation scheme in use while Figs. 16d, 16e, and 16f show the goodput of three smartphones when CSSK modulation in use. For the goodput measurement, we perform the RS error correction on the receiver and measure only the correctly received or recovered symbols. Different from throughput, in the case of goodput, we observe that higher CSK modulation does not always increase the goodput. This is because at higher CSK modulation, such as 32-CSK, the higher value of SER starts to decrease the goodput. However, higher CSSK modulation can consistently increase the goodput due to lower SER. The maximum observed goodput of approximately 5.2 Kbps, 2.5 Kbps and 2 Kbps occur at 16-CSK with 4 KHz symbol rate for Nexus 5, iPhone 5S and Samsung Note 8 respectively. The maximum observed goodput of approximately 7.7 Kbps, 3.7 Kbps and 2.9 Kbps occur at 32-CSSK with 4 KHz symbol rate for Nexus 5, iPhone 5S and Samsung Note 8 respectively.

The lower goodput of iPhone and Samsung Note 8 is attributed to their higher inter-frame loss ratio. For the higher inter-frame loss ratio of 0.37 for iPhone and 0.39 for Samsung Note 8, the transmitter has to apply to RS



(a) Nexus 5, CSK (b) iPhone 5S, CSK (c) Samsung Note8, CSK

(d) Nexus 5, CSSK (e) iPhone 5S, CSSK (f) Samsung Note8, CSSK

Fig. 16. Goodput of different CSK/CSSK modulations and symbol frequency.

encoding accordingly which increases the encoding overhead (more number of parity bits). This shows that in practice, where a single ColorBars transmitter has to support different smartphones, the achievable goodput remains bounded by the the slowest (highest inter-frame loss ratio) smartphone that needs to be supported as a receiver.

10 Discussion and Conclusions

In this work, we presented ColorBars, an LED-to-camera communication system that leverages tri-LED's ability to provide a variety of colors as a way to modulate information to be received by rolling-shutter cameras. We identified three challenges (1) color flicker (2) inter-frame data loss and (3) receiver diversity and addressed them in the design of ColorBars. Our testbed-based evaluation showed that it can achieve a data rate of 5.2 Kbps on Nexus 5 and 2.5 Kbps on iPhone 5S devices. The maximum observed goodput of approximately 7.7 Kbps, 3.7 Kbps and 2.9 Kbps occur at 32-CSSK with 4 KHz symbol rate for Nexus 5, iPhone 5S and Samsung Note 8 respectively. We also show that when ColorBars uses lower CSK modulation, extremely low symbol error rate can be guaranteed for reliable communication.

There are many open challenges that provides the direction for our future work. First, the LEDs utilized in our experiments provide low lumens which requires the camera to be close proximity or equipped with external optical lens while communicating. We plan to extend our work to utilize an array of tri-LEDs to provide high lumens and enable communication from more farther distances. Second, In this paper, we used the CSK constellation as suggested by the 802.15.7 standard, which is not necessarily optimized for rolling shutter camera receivers. In the future, we plan to optimize the CSK constellation design to minimize the inter-symbol interference. Third, CSK is implemented with Pulse Width Modulation (PWM). Any dimming level from 0 to 100 percent can be obtained with high PWM frequency [46], [47]. However, the dimming support in ColorBars and its impact on throughput will be evaluated in future work.

Acknowledgments

This work was supported by the University of California, Davis.

References

[1] H.-Y. Lee, H.-M. Lin, Y.-L. Wei, H.-I. Wu, H.-M. Tsai, and K. C.-J. Lin, "Rollinglight: Enabling line-of-sight light-to-camera communications," in *Proc. 13th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2015, pp. 167–180.

[2] N. Rajagopal, P. Lazik, and A. Rowe, "Visual light landmarks for mobile devices," in *Proc. 13th Int. Symp. Inf. Process. Sensor Netw.*, 2014, pp. 249–260.

[3] W.-C. Wang, C.-W. Chow, C.-W. Chen, H.-C. Hsieh, and Y.-T. Chen, "Beacon jointed packet reconstruction scheme for mobile-phone based visible light communications using rolling shutter," *IEEE Photon. J.*, vol. 9, no. 6, pp. 1–6, Dec. 2017.

[4] R. D. Roberts, "Undersampled frequency shift on-off keying (ufsook) for camera communications (camcom)," in *Proc. IEEE 22nd Wireless Optical Commun. Conf.*, 2013, pp. 645–648.

[5] C. Danakis, M. Afgani, G. Povey, I. Underwood, and H. Haas, "Using a cmos camera sensor for visible light communication," in *Proc. IEEE Globecom Workshops*, 2012, pp. 1244–1248.

[6] Y. Yang, J. Hao, and J. Luo, "Ceilingtalk: Lightweight indoor broadcast through led-camera communication," *IEEE Trans. Mobile Comput.*, vol. 16, no. 12, pp. 3308–3319, Dec. 2017.

[7] Y. Yang, J. Nie, and J. Luo, "Reflexcode: Coding with superposed reflection light for led-camera communication," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, 2017, pp. 193–205.

[8] H. Du, J. Han, X. Jian, T. Jung, C. Bo, Y. Wang, and X.-Y. Li, "Martian: Message broadcast via led lights to heterogeneous smartphones," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1154–1162, May 2017.

[9] C.-L. Chan, H.-M. Tsai, and K. C.-J. Lin, "Poli: Long-range visible light communications using polarized light intensity modulation," in *Proc. 15th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2017, pp. 109–120.

[10] E. Monteiro and S. Hranilovic, "Design and implementation of color-shift keying for visible light communications," *J. Lightw. Technol.*, vol. 32, no. 10, pp. 2053–2060, 2014.

[11] E. Monteiro and S. Hranilovic, "Constellation design for color-shift keying using interior point methods," in *Proc. IEEE Globecom Workshops*, Dec. 2012, pp. 1224–1228.

[12] W. Huang, P. Tian, and Z. Xu, "Design and implementation of a real-time CIM-MIMO optical camera communication system," *Opt. Express*, vol. 24, no. 21, pp. 24 567–24 579, 2016.

[13] *IEEE Standard for Local and Metropolitan area Networks–Part 15.7: Short-Range Wireless Optical Communication Using Visible Light*, IEEE Standard 802.15.7, 2011.

[14] K. Liang, C.-W. Chow, and Y. Liu, "RGB visible light communication using mobile-phone camera and multi-input multi-output," *Opt. Express*, vol. 24, no. 9, pp. 9383–9388, 2016.

[15] C.-W. Chen, C.-W. Chow, Y. Liu, and C.-H. Yeh, "Efficient demodulation scheme for rolling-shutter-patterning of CMOS image sensor based visible light communications," *Opt. Exp.*, vol. 25, no. 20, pp. 24 362–24 367, 2017.

[16] A. Ashok, M. Gruteser, N. Mandayam, J. Silva, M. Varga, and K. Dana, "Challenge: Mobile optical networks through visual MIMO," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 105–112.

[17] S. D. Perli, N. Ahmed, and D. Katabi, "Pixnet: Interference-free wireless links using LCD-camera pairs," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 137–148.

[18] T. Hao, R. Zhou, and G. Xing, "Cobra: Color barcode streaming for smartphone systems," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Services*, 2012, 85–98.

[19] W. Hu, H. Gu, and Q. Pu, "Lightsync: Unsynchronized visual communication over screen-camera links," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 15–26.

[20] 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*, 2015, pp. 197–211.

[21] 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*, 2015, pp. 181–195.

[22] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor positioning with mobile phones and visible light," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 447–458.
[23] Z. Yang, Z. Wang, J. Zhang, C. Huang, and Q. Zhang, "Wearables can afford: Light-weight indoor positioning with visible light," in *Proc. 13th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2015, pp, 17–330.
[24] Z. Tian, K. Wright, and X. Zhou, "The darklight rises: Visible light communication in the dark," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 2–15.
[25] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A visible light based positioning system," in *Proc. 11th USENIX Conf. Netw. Syst. Design Implementation*, 2014, vol. 14, pp. 331–343.
[26] C. Zhang and X. Zhang, "Litell: robust indoor localization using unmodified light fixtures," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 230–242.
[27] S. Zhu and X. Zhang, "Enabling high-precision visible light localization in today's buildings," in *Proc. 15th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2017, pp. 96–108.
[28] C. Zhang and X. Zhang, "Pulsar: Towards ubiquitous visible light localization," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, 2017, pp. 208–221.
[29] J. Li, A. Liu, G. Shen, L. Li, C. Sun, and F. Zhao, "Retro-vlc: enabling battery-free duplex visible light communication for mobile and iot applications," in *Proc. 16th Int. Workshop Mobile Comput. Syst. Appl.*, 2015, pp. 21–26.
[30] S. Shao, A. Khreishah, and H. Elgala, "Pixelated vlc-backscattering for self-charging indoor IoT devices," *IEEE Photon. Technol. Lett.*, vol. 29, no. 2, pp. 177–180, Jan. 2017.
[31] X. Xu, Y. Shen, J. Yang, C. Xu, G. Shen, G. Chen, and Y. Ni, "Passivevlc: Enabling practical visible light backscatter communication for battery-free iot applications," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, 2017, pp. 180–192.
[32] "Cie 1931 chromaticity diagram," [Online]. Available: http://commons.wikimedia.org/wiki/File:Cie_chromaticity_diagram_wavelen gth.png. Lat accessed: 2019
[33] CIE, *Commission Internationale de lEclairage Proc*," Cambridge, U.K.: Cambridge Univ. Press, 1931.
[34] S. M. Berman, D. S. Greenhouse, I. L. Bailey, R. D. Clear, and T. W. Raasch, "Human electroretinogram responses to video displays, fluorescent lighting, and other high frequency sources," *J. Optometry Vis. Sci.*, vol. 68, pp. 645–662, 1991.
[35] A. Sevincer, A. Bhattarai, M. Bilgi, M. Yuksel, and N. Pala, "LIGHTNETs: Smart lighting and mobile optical wireless networks - A survey," *IEEE Commun. Surveys Tutorials*, vol. 15, no. 4, pp. 1620–1641, Oct.-Dec. 2013.
[36] D. Karunatilaka, F. Zafar, V. Kalavally, and R. Parthiban, "Led based indoor visible light communications: State of the art," *IEEE Commun. Surveys Tutorials*, vol. 17, no. 3, pp. 1649–1678, Jul.-Sep. 2015.
[37] C. Landis, "Determinants of the critical flicker-fusion threshold," *Physiological Rev.*, vol. 34, pp. 259–286, 1954.
[38] D. Regan and C. Tyler, "Temporal summation and its limit for wavelength changes: An analog of blochs law for color vision," *J. Optical Society America*, vol. 61, pp. 1414–1421, 1971.
[39] V. Guruswami and M. Sudan, "Improved decoding of reed-solomon and algebraic-geometric codes," in *Proc. 39th Annu. Symp. Foundations Comput. Sci.*, 1998, pp. 28–37.
[40] G. Sharma and R. Bala, *Digital Color Imaging Handbook*. Boca Raton, FL, USA: CRC Press, 2002.
[41] G. Coley, "Beaglebone black system reference manual," *Texas Instruments, Dallas*, 2013.
[42] Q. Wang and D. Giustiniano, "Communication networks of visible light emitting diodes with intra-frame bidirectional transmission," in *Proc. 10th ACM Int. Conf. Emerging Netw. Experiments Technol.*, 2014, pp. 21–28.
[43] P. Mulholland, T. Redmond, D. Garway-Heath , M. Zlatkova, and R. Anderson, "Estimating the critical duration for temporal summation of standard achromatic perimetric stimuli," *Investigative Ophthalmology Visual Sci.*, vol. 56, no. 1, p. 431, 2014.
[44] T. Komine and M. Nakagawa, "Fundamental analysis for visible-light communication system using LED lights," *IEEE Trans. Consum. Electron.*, vol. 50, no. 1, pp. 100–107, Feb. 2004.
[45] K. Cui, G. Chen, Z. Xu, and R. Roberts, "Line-of-sight visible light communication system design and demonstration," in *Proc. 7th Int. Symp. Commun. Syst. Netw. Digital Signal Process.*, Jul. 2010, pp. 621–625.
[46] P. H. Pathak, X. Feng, P. Hu, and P. Mohapatra, "Visible light communication, networking, and sensing: A survey, potential and challenges," *IEEE Commun. Surveys Tutorials*, vol. 17, no. 4, pp. 2047–2077, Oct.-Dec. 2015.
[47] S. Rajagopal, R. D. Roberts, and S.-K. Lim, "IEEE 802.15. 7 visible light communication: modulation schemes and dimming support," *IEEE Commun. Mag.*, vol. 50, no. 3, pp. 72–82, Mar. 2012.

Pengfei Hu received the PhD degree in computer science from the University of California, Davis, in 2018. He is a researcher at VMWare xLab. In 2012, he received the National Scholarship which is the top award for graduate students in China. His research interests include mobile computing, visible light communication, and cyber security and data privacy.

Parth H. Pathak received the MS and PhD degrees in computer science from North Carolina State University, and the PhD degree in computer science from North Carolina State University, in 2012. He is an assistant professor with the Computer Science Department at George Mason University. Before joining at George Mason University, he was a postdoctoral scholar at the University of California, Davis. His research interests include mobile and ubiquitous computing, energy-efficient sensing, Internet-of-Things systems, wireless networking, high-speed millimeter wave wireless networks, and network analytics. He received the Award for Excellence in Postdoctoral Research at the University of California, Davis in 2015. He has also received the Best Paper Award at the IFIP Networking 2014 conference.

Huanle Zhang received the BS degree from Hangzhou Dianzi University, China, in 2011, and the MS degree from the University of Electronic Science and Technology of China, in 2014. He was a project officer with the School of Computer Science and Engineering, Nanyang Technological University, Singapore, from 2014-2016. He is currently working toward the PhD degree with the Department of Computer Science, University of California, Davis. His research interests include mobile systems and applications, cellular communication systems, and Internet of Things. He is a member of the IEEE.

Zhicheng Yang received the BS degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2010, and the MS degree in computer science from Washington University, in Saint Louis, Saint Louis, Missouri, in 2012. He is currently working toward the PhD degree in computer science at the University of California, Davis, CA. His current research interests include millimeter wave sensing, 60 GHz WLAN, visible light communications, and mobile computing. He is a member of the IEEE.

Prasant Mohapatra is a professor with the Department of Computer Science and is serving as the vice chancellor for research at the University of California, Davis. He was the editor-in-chief of the *IEEE Transactions on Mobile Computing*. He has served on the editorials boards of the *IEEE Transactions on Computers*, the *IEEE Transactions on Mobile Computing*, the *IEEE Transaction on Parallel and Distributed Systems*, *ACM WINET*, and *Ad Hoc Networks*. He is a fellow of the IEEE and a fellow of AAAS.