ColorBars: Increasing Data Rate of LED-to-Camera Communication using Color Shift Keying

Pengfei Hu, Parth H. Pathak, Xiaotao Feng, Hao Fu and Prasant Mohapatra
University of California, Davis
Email: {pfhu, phpathak, xtfeng, haofu, pmohapatra}@ucdavis.edu

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 ($\approx$ 12 bytes per second) 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 5.2 Kbps on Nexus 5 and 2.5 Kbps on iPhone 5S, which is significantly higher than previous approaches. It is also shown that lower CSK modulations (e.g. 4 and 8 CSK) provide extremely low symbol error rates ($< 10^{-3}$), making them a desirable choice for reliable LED-to-camera communication.

CCS Concepts

•Networks → Network protocol design; •Human-centered computing → Ubiquitous and mobile computing systems and tools; Smartphones;

Keywords

Visible Light Communication, LED-to-Camera Communication, Rolling Shutter, Color Shift Keying, Smartphones

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 detailed information about the products on the rack including advertisements, promotions etc. In augmented reality applications, for example, the LED lights in an office can broadcast the floor map and directions to a specific room.

The visual association property of LED-to-camera communication is difficult to achieve using Radio Fre-

CoNEXT '15, December 01-04, 2015, Heidelberg, Germany

DOI: http://dx.doi.org/10.1145/2716281.2836097

quency (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, low achievable data rate remains one of its biggest limitations. In recent works such as [1] and [2], the achievable data rate of LED-to-camera link is shown to be no more than 11.32 and 1.25 bytes per second respectively. Although such data rate might be enough for transmitting identification codes etc., it is certainly not sufficient for most of the practical applications such as advertisements and augmented reality which require transfer of small images and textual content. 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 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.

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 by adding illumination symbols of white light such that 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 a Nexus 5 smartphone. 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 -

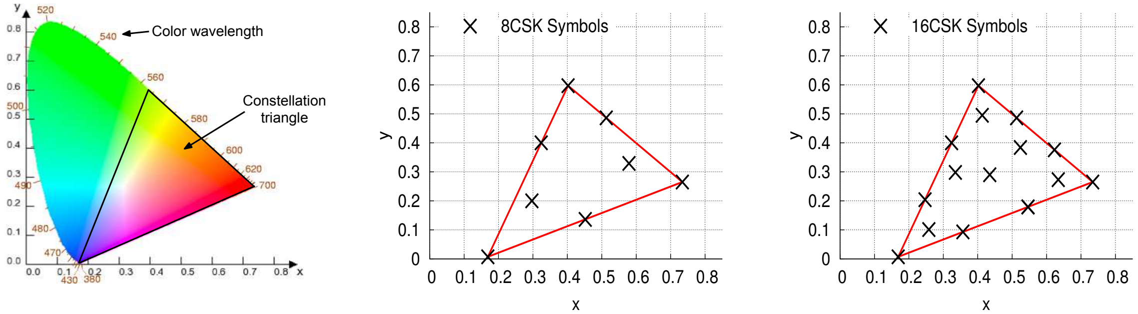
(1) This paper shows that modulating data using different color symbols - Color Shift Keying (CSK) - with the use of tri-LED is a feasible and well-suited technique for LED-to-camera communication. Due to the ability of commodity cameras to detect a wide range of colors, it is possible to use higher constellations with CSK. 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 embedded board with tri-LEDs and evaluate it using Android and iOS smartphones. Our evaluation shows that ColorBars can achieve a data rate of 5.2 Kbps on Nexus 5 and 2.4 Kbps on iPhone 5S which is significantly higher than previous approaches (11.32 and 1.25 bytes per second in [1] and [2] respectively). We also show that ColorBars, when using lower CSK schemes (i.e. 4 and 8 CSK), can achieve moderate to high data rate along with extremely low symbol error rate ($< 10^{-3}$) which can provide reliable LED-to-camera communication.

The remainder of the paper is organized as follows. In Section 2, we provide the necessary background. Section 3 details various challenges that arise in using CSK

(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 [3] (e) 8 CSK Constellation Design (f) 16 CSK Constellation Design

Figure 1: The rolling shutter effect, different modulations as received by camera, and CSK constellation design

for LED-to-camera communication and provides an overview of our system. Section 4 shows how ColorBars eliminates the color flicker problem. Section 5 and 6 outline the use of error correction codes for inter-frame loss and transmitter-assisted calibration process respectively. Section 7 describes the demodulation procedure and Section 8 evaluates our system. Section 9 discusses the related work and we conclude in Section 10 with discussion on open challenges.

2. BACKGROUND

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.

2.1 Rolling Shutter, OOK and FSK

Rolling Shutter: The CMOS (Complementary Metal-Oxide Semiconductor) image sensor (or camera), most commonly used in today's smartphones, tablets, laptops and other mobile devices, exhibit a phenomenon referred as rolling shutter. The 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 the rolling shutter effect. Once all the scanlines of the matrix are read, their output are concatenated to produce an image. Fig. 1(a) 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. 1(b)). Because OOK only utilizes LED's white light, it is less 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. 1(b) 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.

2.2 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 [4] 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 [5]. The diagram maps all colors that are perceivable by human eye to two chromaticity parameters - x and y - as shown in Fig. 1(d). 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. 1(e) and 1(f) show the constellation symbols for 8 and 16 CSK respectively as provided by the IEEE 802.15.7 standard.

Pulse Width Modulation: The LED transmitter varies the intensity of the three LEDs to produce a desired color symbol. Pulse Width Modulation (PWM) is used to vary the intensity of LED's emitted color. PWM is a technique to use the digital signals to control the power supplied to electrical devices such as the LEDs, motors etc. The digital signal could switch between on (full voltage) and off (no voltage). The percentage duty cycle describes the percentage of the time the digital signal is on over a period of time. In the case of an LED, different duty cycles result in a steady voltage between 0 and full, allowing us to control the brightness of the LED. If we use three PWM signals to control the inner red, green and blue LEDs of one tri-LED with different duty circles respectively, the tri-LED will generate an accumulated color.

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. 1(c). 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.

3. OVERVIEW OF COLORBARS

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

3.1 Design Challenges

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

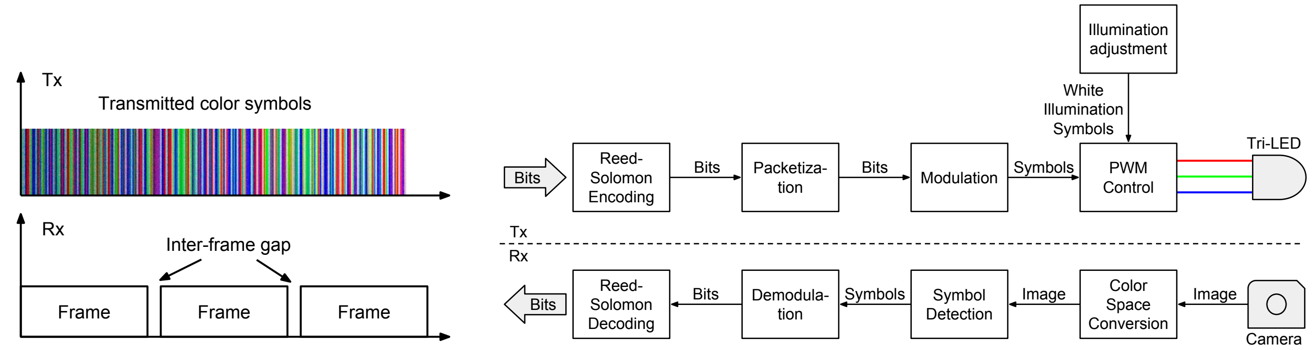
(1) Color Flicker: It is necessary that when an LED is used for data communication, it continues to serve its primary purpose of illuminating the indoor space. Different from OOK or FSK which utilizes only white light during the ON period, if the data symbols are transmitted in the form of different 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 a detrimental physiological effects on humans [6]. 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 require a certain amount of time to process the captured frame as shown in Fig. 2(a). The symbols 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 a 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.

3.2 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(b). 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 CSK modulation. For example, when 8CSK is



(a) Data loss incurred by inter-frame gap

(b) ColorBars System Block Diagram

Figure 2: (a) Inter-frame gap of camera sensor and (b) ColorBars system block diagram

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. At this point, additional white light symbols are added to the data symbols in order to guarantee flicker-free operation of the LED. The 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 a receiver, the achievable data rate is known to be much higher (refer to [7–9] for survey). However, in this work, we are only interested in using rolling shutter cameras as receivers due to their availability in most commonly used mobile devices. 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.

4. 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* [12].

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 the *critical duration*. According to Bloch's law of vision [13],

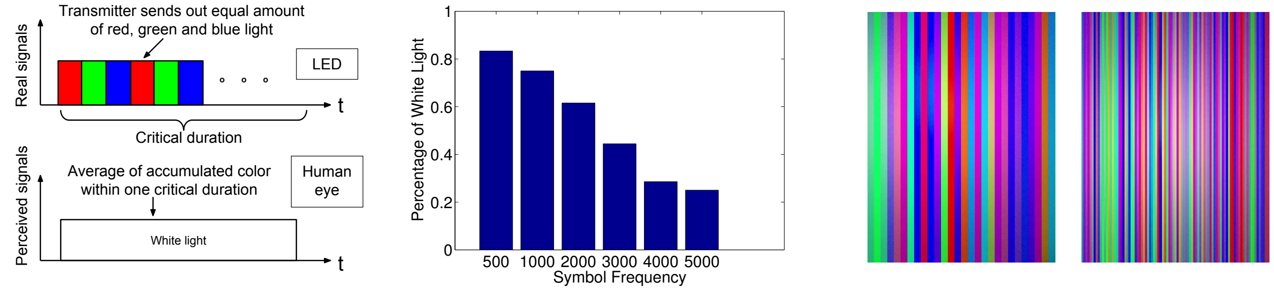
$$\Psi = I \cdot t, \ (t \leq t_c) \tag{1}$$

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, additional stimuli light will not change 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

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

where $I_r(t), I_g(t), I_b(t)$ are the intensity functions of red, green and blue light respectively. This is further demonstrated with an example in Fig. 3(a). Here, a tri-LED emits pure red, green and blue light *in a sequence* at 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.



(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)

Figure 3: Relationship between CSK symbol frequency and color perceived by human eye

Note that in our CSK constellation design, the symbols are equally spread within the R, G and B areas of the constellation triangle (Figs. 1(d), 1(e) and 1(f)). 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 the users.

To address this issue, ColorBars inserts dedicated illumination symbols of white light. When 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. We empirically derive how many white light symbols are necessary to guarantee white light perception for different symbol frequency. We perform an experiment where we increase the symbol frequency from 500 Hz to 5000 Hz and 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 combinations of symbol frequencies and percentage white light. Fig. 3(b) shows the minimum (as observed by the 10 volunteers) percentage of white light necessary to eliminate any effect of color flicker.

From Fig. 3(b), 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. 3(b) to determine the necessary percentage of white light in the 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 computational 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 camera decreases (see Fig. 3(c)). 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. INTER-FRAME DATA LOSS

As we mentioned before, since 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 bitstream 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 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 [14] 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. 2(a). 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 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 $\eta_S \cdot C \cdot L_S$.

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 chooses $k = n - 2t = \eta_S \cdot C \cdot (F_S - L_S)$. 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, 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.

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.

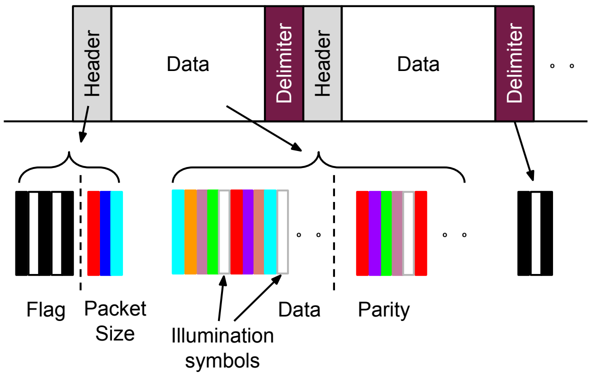


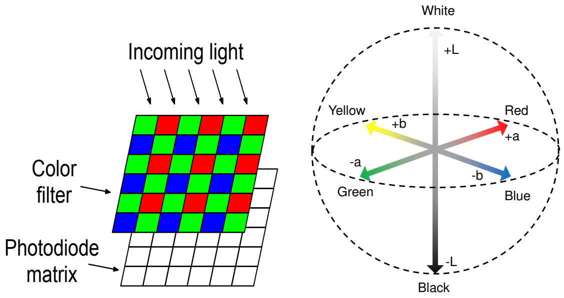
Figure 4: Data Packet Structure

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. 4.

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 6) 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.

6. 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



(a) The Bayer Color Filter (b) CIELab Color Space

Figure 5: (a) Different color filters affect the color perceived by camera, (b) Once image is converted to CIELab, impact of brightness could be reduced.

broadcast by the LED transmitter should be correctly demodulated by all the camera receivers. This requires that each of the camera receivers is able to calibrate the captured color and the true transmitted color with the help from the transmitter. 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.

6.1 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 are 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. 5(a)). 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.

Note that in ColorBars, the transmitter uses CIE color space (Fig. 1(d)) as its basis for constellation design. This is because it allows to choose a set of symbols that are equally distributed among RGB wavelengths and together can produce a white light (necessary for illumination). 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. We will discuss in Section 7 that 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.

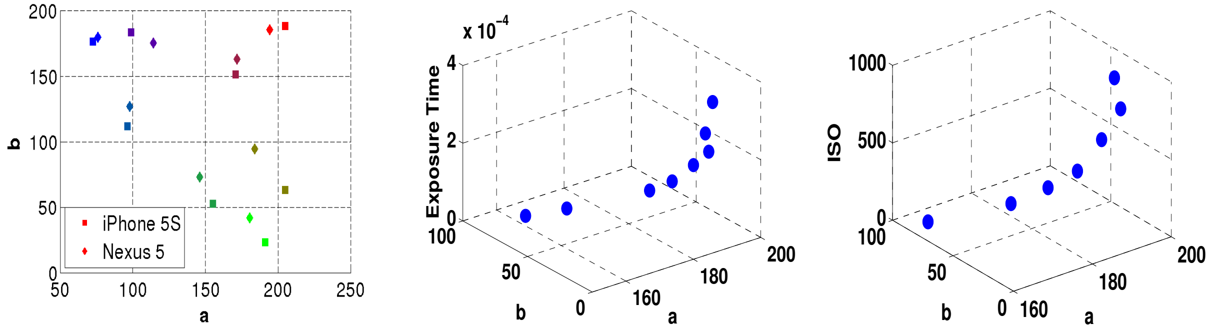
ColorBars uses CIELab color space for demodulation on the receiver side. CIELab is a three channel color space with one channel for lightness L and two channels for colors (a and b). It was designed to overcome the limitations of RGB space in which the distance between two colors does not always correspond to separation perceived by the human eye. The CIELab color space is shown in Fig. 5(b). The a dimension spans from green ($-a$) to red ($+a$) and the b dimension spans from blue ($-b$) to yellow (b). The vertical axis L captures the brightness by spanning from black to white. After removing the lightness dimension, any color can be represented by $\{a, b\}$ as shown in Fig. 5(b).

Fig. 6(a) 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.

6.2 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 depending on the current ambient light conditions. 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 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. Fig. 6(b) and Fig. 6(c) 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 calibration packets. The calibration packets are sent out by the transmitter periodically. A calibration packet includes a flag and all symbols of current modulation scheme in a sequence (e.g. 8 symbols for



(a) Same color can be perceived differently by different cameras (b) Effect of exposure time on the perceived color (c) Effect of ISO on perceived color

Figure 6: Camera's perceived color can vary substantially due to variations in camera filter and settings

8CSK). Similar to the data packets, the flag is used to identify the calibration packets. We use "owowowo" sequence as the flag indicator for 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 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.

7. DEMODULATION

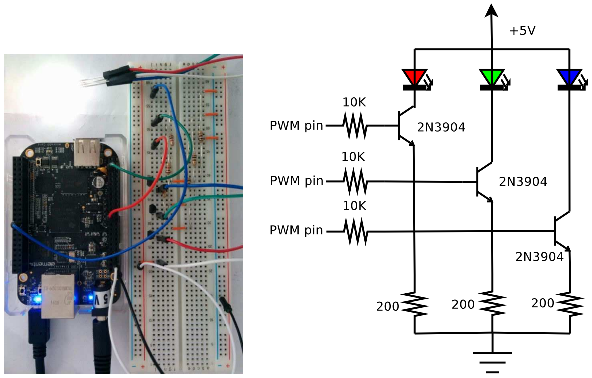
The color symbols transmitted by an LED are received by the image sensor which can demodulate them to receive the data. The receiver captures a continuous set of frames through video recording and then extracts the symbols in each frame. It carries out the following steps in order to demodulate the data from each frame.

Step 1 - Convert to CIELAB color space: Fig. 8(a) 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 i_{th} row and j_{th} 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 j_{th} 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 matching. Once the packets are split, the illumination symbols (white light) are also removed. 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 scheme. These symbols are used as reference symbols for demodulating the symbols in the data packets. We use ΔE [15] to measure the difference between the colors of two pixels in CIELab color space. ΔE is the euclidean distance between two colors in the a, b-plane of the CIELab space. It is known that the difference between two colors is noticeable when $\Delta E \geq 2.3$ [15]. We use 2.3 as the threshold to match the color of a symbol to reference colors received in the calibration packet.

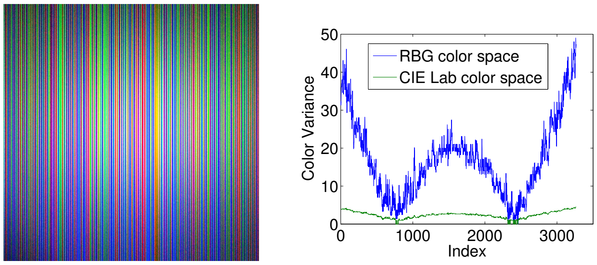
For each data packet, the first essential information to be decoded after demodulation is the size of the packet available in the header. If the number of symbols received in the packet matches the size mentioned in the header, it means that the packet did not suffer any inter-frame loss. In this case, 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.



(a) Transmitter Components

(b) Transmitter Circuit

Figure 7: 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

Figure 8: Impact of non-uniform brightness can be reduced by conversion to CIELab space

8. PERFORMANCE EVALUATION

Experiment Setup: The ColorBars transmitter is implemented using BeagleBone Black board [16]. 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 [17]. We empirically find the maximum frequency of color change supported by the BeagleBone board to be less than 4500 Hz. Various components of the transmitter are depicted in Fig. 7(a) and the circuit diagram of the transmitter module is shown in Fig. 7(b).

The ColorBars receiver module is implemented on smartphones. In order to evaluate multiple commodity cameras, we use Android (Nexus 5) and iOS (iPhone 5S) smartphones. For the Nexus 5, 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 and perform color space conversion along with dimension reduction. The preprocessed 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. As the lumens of the tri-LED is low in our experiments, the phone camera is kept close to the LED (within 3cm).

We evaluate the ColorBars transmitter and receiver modules using three performance metrics - symbol error rate, throughput and goodput.

Inter-frame Loss Ratio: We first measure the inter-frame loss ratio of the two smartphone receivers. The Nexus 5 camera sensor has a resolution of 2448 × 3264 with 30 frames per second. The iPhone 5S camera has a resolution of 1080 × 1920 with 30 frames per second. Note that we use the back camera of both the devices in our experiments. In order to determine the inter-frame loss ratio, the transmitter sends out symbols at different rates and the device cameras record the received symbols. Table 1 shows the number of symbols received per second for the two devices. Based on this, we can observe that iPhone 5S has a higher inter-frame loss ratio (0.37) compared to Nexus 5 (0.23).

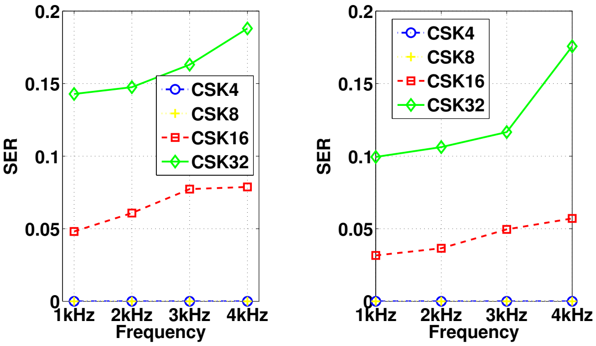
Color Space Conversion: The advantage of converting the received frame from RGB color space to CIELab space is to remove the majority of the effect of brightness variation within the same color symbol. Fig. 8(b) compares how well CIELab space can remove the brightness variations compared to the RGB space. For this, we choose a color symbol in the center of the frame shown in Fig. 8(a), 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.

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 the SER in CIELab space for both Nexus 5 and iPhone 5S smartphones for different CSK schemes. To capture the effect of symbol frequency (symbols per second), we vary it from 1000 Hz to 4000 Hz in the increments of 1000 Hz for each of the CSK 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

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

Table 1: Average inter-frame loss ratio of Nexus 5 and iPhone 5S

practical scenarios.



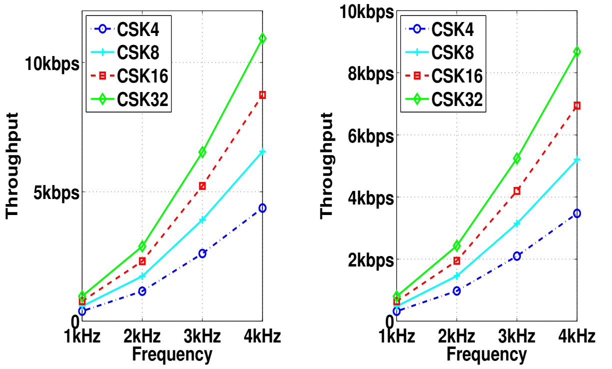
(a) Nexus 5 (b) iPhone 5S

Figure 9: SER vs. Frequency in CIE Lab color space

Figs. 9(a) and 9(b) show the observed SER for Nexus 5 and iPhone 5S respectively. Here, the SER is the fraction of symbols that incorrectly demodulated by the receiver. We observe that for both the smartphones, as we increase the symbol frequency the SER increases for higher CSK schemes (i.e. 16 and 32 CSK). This is because with higher symbol frequency, the size of the symbol (width of the band) decreases. This increases the inter-symbol interference as it becomes more and more difficult to distinguish symbols with fewer pixels of true representation of the colors. It is also observed that the SER in lower CSK schemes (i.e. 4 and 8 CSK) is close to 0, which means that both the camera devices can accurately distinguish (4 or 8) colors even at a very high symbol frequency. This means that lower CSK modulation schemes can be used in the applications where reliable LED-to-camera communication is desirable.

It is also observed from Fig. 9 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.

Throughput and Goodput: We now evaluate the throughput and goodput that is achievable through ColorBars system. Similar to the SER results, we vary the CSK modulation and symbol frequency to investigate their impact on throughput and goodput, with automatic camera settings of exposure time and ISO. It should also be noted 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).



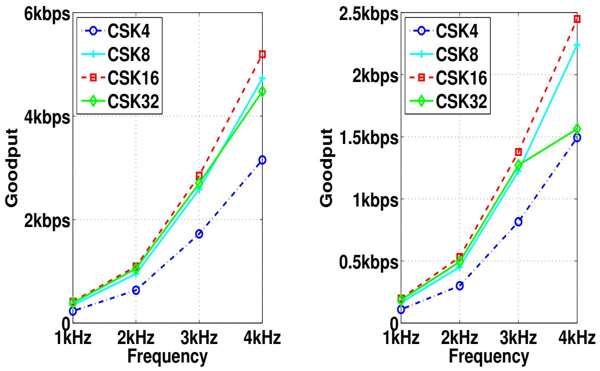
(a) Nexus 5 (b) iPhone 5S

Figure 10: Throughput of different CSK modulations and symbol frequency

Figs. 10(a) and 10(b) show the raw achievable throughput data rate by the Nexus 5 and iPhone 5S respectively. For the calculation of raw throughput, we do not perform any error correction at the receiver but simply measure the number of symbols received excluding the illumination symbols of white light. 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 modulation schemes achieve higher throughput. The maximum observed throughput with 32 CSK and 4KHz symbol frequency is over 11 Kbps and 9 Kbps for Nexus 5 and iPhone 5S respectively.

The reason why iPhone 5S achieves lower throughput compared to Nexus 5 is because the throughput depends on two factors - SER and inter-frame loss ratio. Even though iPhone 5S has lower SER, its inter-frame loss ratio is much higher (Table 1) in comparison with Nexus 5. This means that more number of transmitted symbols are lost for the iPhone, resulting in overall throughput reduction.

Figs. 11(a) and 11(b) show the observed goodput for Nexus 5 and iPhone 5S respectively. 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



(a) Nexus 5

(b) iPhone 5S

Figure 11: Goodput of different CSK modulations and symbol frequency

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. The maximum observed goodput of approximately 5.2 Kbps and 2.5 Kbps occur at 16 CSK with 4 KHz symbol rate for both Nexus 5 and iPhone 5S respectively.

The lower goodput of iPhone is attributed to its higher inter-frame loss ratio. For the higher inter-frame loss ratio of 0.37 for iPhone, the transmitter has to apply to RS 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.

9. RELATED WORK

LED-to-camera communication has been recently studied in [1, 2, 18, 19]. In [18], 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 [19] 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 a recent work [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. Compared to this, 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] and also used in the IEEE 802.15.7 standard [4]. Different from these, ColorBars utilizes rolling shutter cameras as the receivers which introduces many novel challenges. A recent work [20] has proposed to use RGB LED and CSK for data communication with mobile phones, however, it does not utilize the rolling shutter scheme of CMOS cameras or address the issue of color flicker.

In another set of research works, screen-to-camera communication has been studied in [21], [22], [23], [24]. Authors in [21] proposed visual-MIMO for enabling communication between pixels of LCD/LED arrays and cameras. Authors in [22] proposed to use blur-adaptive OFDM coding to encode the data in the pixels of LCD display. [23] 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 [24] using the transmitter and receiver frame synchronization. In recent work [25, 26], 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 [27] and [28] 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.

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 which can 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. We also show that when ColorBars uses lower CSK modulation, extremely low symbol error rate can be guaranteed for a reliable communication.

There are many open challenges which provide the directions for our future work. First, the LEDs utilized in our experiments provide low lumens, requiring the phone camera to be in close proximity while communicating. We plan to extend our work to utilize an array of tri-LEDs to provide high lumens and enable communication from farther distances. Second, in this work, we used the CSK constellation as suggested by the 802.15.7 standard, which is not necessarily optimized for the rolling shutter camera receivers. In the future, we plan to optimize the CSK constellation design to minimize the inter-symbol interference.

11. 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 *Mobisys*, ACM, 2015.

[2] N. Rajagopal, P. Lazik, and A. Rowe, "Visual light landmarks for mobile devices," ACM IPSN '14.

[3] "Cie 1931 chromaticity diagram."

[4] "Ieee standard for local and metropolitan area networks–part 15.7: Short-range wireless optical communication using visible light," *IEEE Std 802.15.7-2011*, pp. 1–309, Sept 2011.

[5] CIE, "Commission Internationale de lEclairage Proc.," *Cambridge University Press*, 1931.

[6] 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.," *Journal of Optometry & Vision Science*, 1991.

[7] A. Sevincer, A. Bhattarai, M. Bilgi, M. Yuksel, and N. Pala, "Lightnets: Smart lighting and mobile optical wireless networks - a survey," *Communications Surveys Tutorials, IEEE*, 2013.

[8] D. Karunatilaka, F. Zafar, V. Kalavally, and R. Parthiban, "Led based indoor visible light communications: State of the art," *Communications Surveys Tutorials, IEEE.*

[9] P. Pathak, X. Feng, P. Hu, and P. Mohapatra, "Visible light communication, networking and sensing: A survey, potential and challenges," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1–1, 2015.

[10] E. Monteiro and S. Hranilovic, "Design and implementation of color-shift keying for visible light communications," *Lightwave Technology, Journal of*, 2014.

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

[12] C. Landis, "Determinants of the critical flicker-fusion threshold," *Physiological Reviews*, 1954.

[13] D. Regan and C. Tyler, "Temporal summation and its limit for wavelength changes: an analog of blochâĂŹs law for color vision," *JOSA*, 1971.

[14] V. Guruswami and M. Sudan, "Improved decoding of reed-solomon and algebraic-geometric codes," in *Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on*, pp. 28–37, IEEE, 1998.

[15] G. Sharma and R. Bala, *Digital color imaging handbook*. CRC press, 2002.

[16] G. Coley, "Beaglebone black system reference manual," *Texas Instruments, Dallas*, 2013.

[17] Q. Wang and D. Giustiniano, "Communication networks of visible light emitting diodes with intra-frame bidirectional transmission," in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pp. 21–28, ACM, 2014.

[18] R. D. Roberts, "Undersampled frequency shift on-off keying (ufsook) for camera communications (camcom)," in *IEEE WOCC 2013*.

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

[20] S.-H. Chen and C.-W. Chow, "Color-shift keying and code-division multiple-access transmission for rgb-led visible light communications using mobile phone camera," *Photonics Journal, IEEE*, vol. 6, no. 6, pp. 1–6, 2014.

[21] A. Ashok, M. Gruteser, N. Mandayam, J. Silva, M. Varga, and K. Dana, "Challenge: Mobile optical networks through visual mimo," in *Mobicom*, ACM, 2010.

[22] S. D. Perli, N. Ahmed, and D. Katabi, "Pixnet: interference-free wireless links using lcd-camera pairs," in *Mobicom*, ACM, 2010.

[23] T. Hao, R. Zhou, and G. Xing, "Cobra: color barcode streaming for smartphone systems," in *Mobisys*, ACM, 2012.

[24] W. Hu, H. Gu, and Q. Pu, "Lightsync: Unsynchronized visual communication over screen-camera links," in *Mobicom*, ACM, 2013.

[25] T. Li, C. An, X. Xiao, A. T. Campbell, and X. Zhou, "Real-time screen-camera communication behind any scene," in *Proc. of MobiSys*, 2015.

[26] 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. of MobiSys*, 2015.

[27] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor positioning with mobile phones and visible light," in *Mobicom*, ACM, 2014.

[28] Z. Yang, Z. Wang, J. Zhang, C. Huang, and Q. Zhang, "Wearables can afford: Light-weight indoor positioning with visible light," in *Mobisys*, ACM, 2015.