ABSTRACT
As a popular approach to implementing Visible Light Communication (VLC) on commercial-off-the-shelf devices, LED-Camera VLC has attracted substantial attention recently. While such systems initially used reflected light as the communication media, direct light becomes the dominant media for the purpose of combating interference. Nonetheless, the data rate achievable by direct light LED-Camera VLC systems has hit its bottleneck: the dimension of the transmitters. In order to further improve the performance, we revisit the reflected light approach and we innovate in converting the potentially destructive interferences into collaborative transmissions. Essentially, our ReflexCode system codes information by superposing light emissions from multiple transmitters. It combines traditional amplitude demodulation with slope detection to "decode" the grayscale modulated signal, and it tunes decoding thresholds dynamically depending on the spatial symbol distribution. In addition, ReflexCode re-engineers the balanced codes to avoid flicker from individual transmitters. We implement ReflexCode as two prototypes and demonstrate that it can achieve a throughput up to 3.2kb/s at a distance of 3m.

CCS CONCEPTS
• Computer systems organization → Special purpose systems;
• Networks → Wireless access networks;
• Hardware → Design reuse and communication-based design;
• Human-centered computing → Mobile computing;
• Theory of computation → Error-correcting codes;

KEYWORDS
Visible Light Communication; Collaborative Transmissions; Grayscale Modulation

1 INTRODUCTION
Deemed as a potential "role" in the 5G era, Visible Light Communication (VLC), using visible light spectrum as the communication media, has been attracting increasing attentions. While high-power Light Emitting Diodes (LEDs) and high-sensitivity Photo-Diodes (PD) may achieve a data rate at Gbit/s level within very short distance [17], readily deployable VLC systems need to be based on commercial-off-the-shelf (COTS) devices such as LED luminaires, screens, and smartphone cameras [13–16, 19, 20, 23, 34]. They are mostly applied for low-rate data communication services such as information broadcasting, where availability, efficiency, and reliability are more important than channel capacity. Among the latter category, the LED-Camera communication is particularly promising thanks to the widespread usage of LED lighting and smartphones [13, 14, 19, 20]. On one hand, LED luminaires, of high efficiency and long lifespan, are becoming a mainstream lighting apparatus [37] and are thus good choices of VLC transmitters. On the other hand, smartphone cameras, with the rolling-shutter effect [9], are deemed as convenient VLC receivers. More importantly, as a smartphone camera has a limited angle-of-view, using it as a receiver can largely avoid interference that severely challenges the potential deployment of LED-PD VLC [45].

Many innovative systems have been proposed for realizing LED-Camera VLC [9, 13, 14, 19, 20, 28]. These systems initially made use of reflected light as their communication media, but reflections may blend emissions from various interfering LED transmitters and hence these early systems were confined to a single transmitter [9, 28]. Recent developments shifted to direct light, i.e., taking photos of LED luminaires directly, in order to avoid interference [19, 20] and also to take the advantage of collaborative transmissions [13, 14]. We illustrate and compare these two approaches qualitatively in Figure 1; it shows the spatial division advantage of direct light. However, direct light LED-Camera VLC...
has already hit its bottleneck of data rate [13]; as the data rate is strongly affected by the dimension of a transmitter, achieving a kbit/s level of data rate at a reasonable distance would require a luminaire dimensionally similar to the outdated fluorescent tube. Reflected light, on the contrary, does not depend on the dimension of the transmitter, so it has the potential to serve as the stepping-stone for achieving higher data rates.

From a practical point of view, down light LED illumination faces severe issues with user experiences due to its glaring effect that may cause stress and anxiety [18, 32]. Although modifying the structure of LED may partially alleviate glare [8, 24], raising smartphones towards luminaires (hence directly looking at them) can hardly be a comfortable action and thus may discourage users from using such a VLC service. At the meantime, reflected light LED illumination has been gaining momentum in commercial usage [1, 4], thanks to its lower cost and aesthetical pleasingness. As a result, an LED-Camera VLC system, in particular its camera-based receiver, has to be fully functional under scenarios where the indoor environment is partially or even totally lit by reflected LED lighting.

Nevertheless, using reflected light as the communication media poses two significant challenges to a VLC system design: severe interference and low SNR (Signal-to-Noise Ratio). Because reflected light blends multiple transmissions and other light emissions together, specific mechanisms are needed to cope with the resulting mess. Solutions for Radio-Frequency (RF), such as FD (Frequency Division) and TD (Time Division), may not work, given that LED-Camera VLC can only support an optical clock rate at no more than 8kHz [13, 20]: FD entails a color-dependent channelization that significantly reduces signal strength and hence the transmission range [14], while TD effectively reduces the time sharing for individual luminaires, hence substantially lowering the already limited data rates. One may borrow the conventional cell-based design from cellular networks (as theoretical proposals for LED-PD VLC do [45]), yet a cellular partition successfully constructed for reflected light LED-Camera VLC would allow only a single transmission in each cell, and a single transmitter suffers a low SNR inherent to reflected VLC [13]. If the interfering co-cell transmissions can be made constructive, SNR may be elevated to support a reasonable transmission distance.

Therefore, we intend to answer the following question in our paper: would it be possible to constructively use the multiple transmissions within a cell of a reflected light LED-Camera VLC system? Apparently, a satisfactory answer would boost data rate for individual cells while extending the system to cover a large area in an interference-free manner through cell partition. To this end, we propose ReflexCode that codes information by superposing light emissions from multiple transmitters. With a close synchronization among these transmitters, the new modulation, termed Grayscale-Shift Keying (GSK), achieves a higher bit-per-symbol ratio than the commonly used On-OFF Keying (OOK), thus potentially resulting in a higher bit rate. In other words, ReflexCode aims to convert the originally destructive interferences into constructive signals. However, this idea has further imposed on us several challenges.

First of all, demodulating GSK with a high symbol rate is highly non-trivial even under ideal reflection. Experiments show that when the transmitting frequency reaches 5kHz or above, a symbol can barely maintain stable before changing to the next one, producing “blurred” boundaries between consecutive symbols. Secondly, the tonal range of grayscale may vary spatially within a frame captured by the camera due to the spatial distribution of the transmitters; this further complicates the demodulation as one cannot use fixed thresholds to identify respective symbols. Last but not least, as our higher order modulation renders the traditional run-length limited or DC balanced codes for OOK not applicable anymore, flicker may appear even under a high transmitting frequency due to the potentially introduced low-frequency components.

To cope with these issues, we devise a robust demodulation scheme to extract bits from a frame accurately: it combines traditional amplitude demodulation with slope detection to tackle the “boundary-blur” issue, and it also computes decoding thresholds dynamically depending on the spatial symbol distributions within a frame. In addition, we re-engineer balanced codes so as to suppress flicker for individual transmitters while maintaining the efficiency of GSK. In summary, our contributions can be summarized as:

- The innovative idea of GSK to constructively use multiple LED transmissions for boosting data rate.
- A robust demodulation algorithm to effectively extract information from GSK modulated transmissions.
- Novel balanced codes adapted to GSK for flicker mitigation and reliability improvement.
- Two prototypes for validating the effectiveness of ReflexCode, and extensive field experiments to showcase the promising performance of ReflexCode.

The remainder of our paper is organized as follows. We first set up the background for the design of ReflexCode in Section 2. Then we describe the system architecture of ReflexCode, as well as its major components, in Section 3. The extensive evaluations on ReflexCode prototypes are reported in Section 4. We survey related literature in Section 5 and finally conclude the paper in Section 6.

2 PRELIMINARY AND MOTIVATION

We set up the background for developing ReflexCode in this section. We first briefly introduce the rolling-shutter effect of a CMOS camera. Then we explain the advantage of the reflected light VLC in detail, followed by the basics of GSK.

2.1 Rolling Shutter

The rolling shutter effect stems from the sequential exposure of CMOS cameras. As the pixels are exposed and sampled in a column-by-column manner, they may record time-varying information [9, 20]. In particular, if the light emitted from an LED luminaire (transmitter hereafter) flickers in a high frequency unnoticeable by human users, the rolling shutter will lead to bright-dark bands in a frame depending on the ON-Off state of the transmitters when individual columns are exposed. How a transmitter flickers is dictated by the modulation scheme, for which OOK (where bright band stands for “1” and dark one for “0”) is widely used thanks to its simplicity and robustness. The bit rate of such a transmission depends on several factors including the signal frequency and the camera features, mostly importantly on the Region of Interest (RoI) [7]. As shown in Figure 2, direct light LED-Camera VLC has spatially separated RoIs, yet the dimensions of individual RoIs are determined by those
of the transmitters and their distances from the receiver [13]. This makes it quite easy to handle interference but extremely difficult to scale-up data rate.

2.2 Reflection Advantage

As RoI is the major bottleneck to improve data rate, increasing RoI serves as a quick solution for boosting VLC performance. According to Figure 3, for a given camera (receiver hereafter) with a fixed Angle-of-View (AoV) characterized by $\beta$, we have the relation between the dimension of an RoI $s_{RoI}$ and that of the transmitter $s_{tx}$ by neglecting the small value of focal length:

$$s_{RoI} \propto \frac{s_{tx}}{d_c}, \quad (1)$$

where $d_c$ is the distance from the transmitter to the receiver. Apparently, increasing $s_{RoI}$ can be achieved by increasing $s_{tx}$ and/or decreasing $d_c$. Nevertheless, as we discussed in Section 1, none of these values can be easily manipulated as they are determined by common lighting constraints. Fortunately, we may achieve both through reflected light VLC. Again according to Figure 3, the image of the transmitter on a reflector (e.g., wall or ceiling free of major blockage) is amplified due to the Field of View (FoV) of the transmitter, characterized by the angle $\alpha$. In particular, we have another relation between the dimension of this image $s_{image}$ and $s_{tx}$:

$$s_{image} = s_{tx} + 2d_r \times \tan \alpha,$$

where $d_r$ is the distance from the transmitter to the reflector. Now, if we replace $s_{tx}$ in (1) with $s_{image}$, we would have an increase in $s_{RoI}$ as $\Delta s_{RoI} \propto 2d_r \tan \alpha / d_c$. Note that $d_r$ now becomes the distance between the reflector and the receiver, which may be controlled (thus reduced) by a user. Based on our experience, this increase can easily amplify the RoI to the full width of the receiver (i.e., the camera width), allowing us to obtain the maximum achievable data rate for LED-Camera VLC. However, this amplification comes at two major costs: i) signal attenuation due to the long propagation distance and the absorption of the reflector, and ii) the interference from other transmitters due to the absence of spatial division: the “barcodes” could be a mess as shown in Figure 2. As we briefly explained in Section 1, both low SNR and interference demand special attention and are thus the major target of this paper.

2.3 Grayscale-Shift Keying (GSK)

Though omitted in Section 2.1, the brightness of the bands within an RoI can actually be individually controlled: the brightness of a given band is proportional to the light intensity when the band is exposed. Using the level of brightness (hence grayscale in digital image) to form symbols, a special Amplitude-Shift Keying (ASK) can be constructed. We term this modulation Grayscale-Shift Keying (GSK); it extends OOK to have a higher bit-per-symbol ratio. Although GSK applies to direct light VLC, it can be more complicated and less robust than OOK. However, as shown in Figure 2, the mutual-interfering transmissions under reflected light VLC naturally form bands with a various grayscale, albeit too messy to be demodulated without coordination. If we can constructively superpose these transmissions by re-designing the modulations for individual transmitters, we may end up with a GSK that potentially leads to a higher data rate, due to its higher SNR [44] and modulation order. Note that, in order to successfully superpose multiple transmissions on a reflector, the usable image size $\hat{s}_{image}$ has to be the overlap of all images of the concerned transmitters, which is smaller than individual $s_{image}$. Nonetheless, $\hat{s}_{image}$ could still generate a sufficiently large RoI at the receiver if the "amplifier" $(2d_r \tan \alpha / d_c)$ is large enough, which should be guaranteed within a single cell (i.e., a small area). Therefore, we hereafter focus only one single-cell VLC, where we convert destructive transmissions from multiple co-cell transmitters into GSK modulation.

3 ReflexCode: Modulation by Superposing Transmissions

In this section, we first briefly introduce the overall design of ReflexCode, then we discuss its individual components in detail.
Figure 4: Superposition under asynchronous transmissions (left) and that under synchronized transmissions (right).

3.1 Overview of ReflexCode

The system is partitioned by default into two parts, namely transmitter and receiver. As shown in Figure 2 (the lower part), the transmitter consists of a group of coordinated LED luminaires, while the receiver uses a smartphone camera as its front-end. Although the reflector is part of the communication media, we do not involve it in the design of ReflexCode, yet we shall evaluate ReflexCode against various reflector colors and textures in Section 4.5.

On the transmitter side, GSK is obtained by superposing different ON-OFF states of the synchronized transmitters. Moreover, the emissions from individual transmitters have to be first encoded to achieve DC-balance and bounded disparity, so as to confine bit errors and avoid flicker. Finally, FEC is added to combat possible packet loss. On the receiver side, demodulation is proceeded by a tonal range identification process; this calibration is needed to cope with the nonuniform tone distribution. The demodulation process takes the dynamically calibrated thresholds and the local differentials as input to convert grayscale bands into bits.

3.2 GSK from Synchronized Emissions

Considering a common area on a reflector lit by \(N\) transmitters. Depending on how many transmitters are ON at a given moment, the reflected light intensity has \(N + 1\) scales due to superposition. In conventional VLC systems, these transmitters may not be closely synchronized, hence the resulting superposition may not be decodable. However, synchronizing these transmitters is rather straightforward as they are connected to a common power grid [26]. In our prototypes, these transmitters share the same drivers and are thus inherently synchronized. Consequently, each grayscale lasts for a fixed amount of time, causing a determined width of a band on the receiver side. Therefore, we can treat each distinct grayscale as a symbol. We compare an asynchronous superposition with a synchronized one (hence GSK) in Figure 4. Although GSK may still lead to variations in the widths of bands due to consecutive identical symbols, asynchronous superposition results in a high variance in width and also very blurred boundaries between bands, making it almost impossible to demodulate.

As GSK encodes bits to these \(N + 1\) symbols, the bit-per-symbol ratio is \(b = \log_2(N + 1)\) in theory. Note that \(b\) does not have to be an integer. For example, if \(N = 2\), we can code information in a ternary system instead of binary. However, we currently only consider the cases where \(b\) is an integer, so as to simplify the system design. In particular, our prototypes involve three transmitters to form at most a 4-GSK, with each symbol representing 2 bits in the following manner:

\(N\) be an integer. For example, if a ternary system instead of binary. However, we currently only consider the cases where \(b\) is an integer, so as to simplify the system design. In particular, our prototypes involve three transmitters to form at most a 4-GSK, with each symbol representing 2 bits in the following manner:

i) All three transmitters OFF to generate a totally dark band for \(00\), i.e., symbol-0 (or S-0),

ii) The middle transmitter \(t_x2\) ON while others OFF for 01, i.e., symbol-1 (or S-1),

iii) The complement of symbol-1 (\(t_x1\) and \(t_x3\) ON and \(t_x2\) OFF) for 10, i.e., symbol-2 (or S-2), and

iv) All ON to form symbol-3 (or S-3) for 11.

This balanced allocation aims to produce a symmetric tonal range within an RoI, thus facilitating the demodulation. The principle behind this allocation scheme can be readily extended to a larger \(N\), yet we shall not dwell on this technique detail as we would most probably not have a very large \(N\) under normal lighting constraints.

3.3 Encoding with Flicker Mitigation

As a communication channel, DC balancing and bounding bit disparity are important to VLC, and a DC balance code is also crucial for mitigating visible flicker (one of the major side effects of LED-VLC). The existing Run-Length Limited (RLL) codes (as recommended in IEEE 802.15.7 [30]) are not designed for our collaborative transmissions that lead to GSK, as they are based on the assumption of a single transmitter with OOK modulation (2-GSK under our framework). However, ReflexCode employs a higher order GSK, so it is highly nontrivial to keep the overall brightness energy balanced and thus the transmitters free from flicker. In ReflexCode, the flicker can be caused at both symbol and transmitter levels, due to its superposing nature. For example, as shown in Figure 5, if we drive the transmitters by GSK symbols directly mapped from the original data bits, flicker may appear due to, for example, symbol stream of S-3, S-1, S-3, S-1 (marked in red) forces \(t_x2\) to stay ON during successive four time slots.

To this end, we propose \(N\)'s-complement (NC) coding to mitigate flicker at both symbol and transmitter levels based on the following definitions.

Figure 5: NC coding for flicker mitigation.
Definition 3.1. Given symbol $S-n$ produced by putting $n$ transmitters ON, the complement of $S-n$, $S-n$, is produced by turning the original $n$ transmitters OFF while switching the remaining $N-n$ transmitters ON.

The NC coding essentially splits a symbol into a pair of original and complement symbols to maintain a symbol level energy balance. At the transmitter level, this scheme reduces to normal Manchester coding at individual transmitters, as shown in Figure 5. We did a user study with 10 volunteers to observe the flicker of LED luminaires, and the results show no perceivable flicker on both individual transmitters and the superposed reflection. We omit the report of this user study in Section 4 due to its simplicity.

### 3.4 FEC to Combat Data Loss

Due to lack of feedback channel, LED-Camera VLC offers only one-way communication. Moreover, high data loss of LED-Camera VLC due to, for example, the inter-frame loss of a camera has been verified in [20]. All these necessitate a Forward Error Correction (FEC) scheme. We adopt rateless codes, in particular Raptor codes [31], as FEC for ReflexCode, due to their recent successful application to VLC [13, 40]. Under Raptor coding, a message with $k$ original packets will generate $\gamma \times k$ encoded packets, where $\gamma$ is the encoding overhead fixed as $\gamma \approx 1.25$ empirically. Therefore, ReflexCode divides an arbitrary message into blocks with $k$ packets and generates Raptor encoded packets, for further modulated into GSK symbols for transmission.

Remarks: The aforementioned modules allow us to construct ReflexCode transmitter, whose outcomes are illustrated in Figure 6. To demodulate these symbols, we face two major challenges: i) the nonuniform tonal range distribution due to the spatial distribution of transmitters, and ii) the transience of intermediate symbols especially under high frequency.

### 3.5 Adapting to Nonuniform Distribution

As ReflexCode is devised for using reflected light, existing ambient light from other non-VLC luminaires or the sun (through windows) may shine upon the reflector affecting on ReflexCode’s demodulation. Therefore, we employ a high-pass filter to remove low-frequency noises before demodulation. Moreover, the tonal range of grayscale symbol stream within a received frame is not uniformly distributed due to the spatial distribution of the transmitters. Basically, the reflection causes a brightest frame center and a gradual reduction of brightness on both sides as shown in Figure 6, which makes it difficult to set thresholds for detecting symbols with varying spatial location in a frame. Earlier proposals suggest using polynomial regression to fit the change of tonal range [9], yet it is too computational intensive for COTS VLC.

In ReflexCode, we dynamically measure the tonal range for each symbol so as to properly set the thresholds. Inspired by the pilot channel of CDMA, we put a few successive $S-N$ $(N = 3$ in our current implementations) in a packet header. This acts as reference for measuring the tonal range, as a header represents the brightest pixels of a tonal range locally. Having multiple headers in a frame further enables us to detect the tonal range variation across the frame. ReflexCode starts by identifying headers, using a rough threshold determined by brightest pixels of a frame (the header closest to the center) and those at both edges of the frame, as shown by Figure 7(a). Upon locating all headers in a frame (according to their special width and the rough threshold), ReflexCode uses their brightness levels to determine a piecewise linear function as the envelope of the tonal range across the frame, as shown in Figure 7(b) (the envelope). And the thresholds (i.e., lower bounds) are set as 92%, 62%, 25%, and 0 for S-3, S-2, S-1, and S-0, respectively. We determine these thresholds empirically to minimize the error rate.
3.6 Demodulation Driven by Differential

In addition to the demodulation thresholds, the procedure explained in Section 3.5 also extracts individual packets out of a frame. In the following, we present the core demodulation process to retrieve the data bits from a certain packet.

3.6.1 Width-Driven Demodulation: A Baseline. Existing LED-Camera VLC systems mostly apply OOK and demodulate it by examining the widths of grayscale bands in a frame [9, 13, 14, 20], while detecting the width of a band using a bisection threshold. In addition, by reasoning on the widths of bright/dark bands in a combinatorial manner, [13] is able to well handle the blooming effect, i.e., brighter gets wider while darker gets thinner. However, the GSK demodulated by ReflexCode is fundamentally different, as shown by the comparison in Figure 6. Essentially, the higher order nature of GSK makes the symbols more prone to the blooming effect, hence the symbol width and boundary become very vague for detection. Therefore, a width-driven demodulation may fail when facing a cluster of bands with various brightness levels; it is likely to deem bands for intermediate symbols as noise and hence discard them. This issue is particularly evident under a higher frequency, where intermediate symbols can hardly get stable before transitioning to the next brightness level. As a result, we seek to obtain a precise symbol detection by examining the brightness gradient and the relation between consecutive symbols.

3.6.2 Judging Points Determination. With the thresholds obtained in Section 3.5, we still need to determine how to correctly sample pixels for comparing against the thresholds. For example, if we send S-0 and S-3 consecutively, there must exist some pixels with brightness levels equal to S-1 and/or S-2. Therefore, if we directly apply the thresholds throughout a whole frame, there would be a large number of detection errors. A correct logic should first determine where to sample pixels that can be part of a symbol, then use these pixels for demodulation by comparing them against the three thresholds.

Let us consider the function $G(i), i = 1, 2, ..., m$ of pixel brightness in a frame, where $m$ is the horizontal pixel size and $G$ returns the average pixel brightness for a given column. Theoretically, the first-order derivative of $G(\cdot)$ should go to zero where a certain symbol appears. In reality, the value may not be exactly zero due to the transience of intermediate symbols under a high frequency. Moreover, $G$ is discrete given pixels as input, so we should detect where the value of the finite difference of $G(\cdot)$ goes across zero. We denote the indices of such pixels as judging points, and use Algorithm 1 for their determination. Basically, the algorithm makes use of Criterion 1 (i.e., first-order derivative crossing zero) to detect the local minimal and maximal, which potentially representing S-0 and S-3, and it utilizes the other two criteria (i.e., second-order derivative crossing zero) to detect positive and negative inflection points, which are candidates for S-1 and S-2. Here the derivatives are computed by finite differences. As one symbol may cause a couple of separated judging points that are still close to each other, the algorithm filters such redundant judging points belonging to the same symbol using two thresholds: i) $y_{\text{gap}}(i)$ for the tonal range, and fixed $x_{\text{gap}}$ representing the minimum width of a grayscale band. Examples of the determined judging points in a packet are illustrated in Figure 8.

3.6.3 ReflexCode Demodulation. Given the thresholds and judging points determined in Section 3.5 and 3.6.2, ReflexCode may better handle the intermediate symbols that cannot be dealt with by the baseline width-driven demodulation mentioned in Section 3.6.1. Essentially, ReflexCode first locates all headers using brightness and
width information, which results in the potential packets between two consecutive headers. For each potential packet, ReflexCode scans through judging points and demodulates them into bits by comparing them against thresholds. The resulting candidate packets are then given to Raptor decoding for recovering the original packet stream.

However, demodulation faces a challenge incurred by the inherent nonlinear variance of the tonal range across a frame and the piecewise linear thresholds used to approximate this nonlinearity. For example, the second judging point from the left identified in Figure 8 is actually representing S-3, but it gets wrongly demodulated to S-2 due to a local variance in tonal range (thus its value barely missing the threshold). Fortunately, the wrongly demodulated symbol stream becomes S-0, S-2, S-0, · · · , yet based on the NC coding explained in Section 3.3, the only possible combination would be S-0, S-3, · · · . Therefore, ReflexCode’s NC coding has the ability to rectify these symbol errors.

More generally, ReflexCode checks every pair of symbols based on the property of NC coding to correct potential errors, using Algorithm 2. As NC coding results in a finite number possible combinations of two consecutive symbols (4 under N = 3), we take the set of these combinations C_b as an input. The algorithm checks though every two symbols sequentially: it accepts a pair if this pair belongs to C_b; otherwise symbol correction is needed. The correction procedure correct takes the ratio between the brightness levels of the two symbols as input (retrieved from G), and it searches for the closest symbol pair in C_b as the candidate. In our implementation, this whole algorithm is merged with the demodulation procedure, avoiding the need to go through every packet twice: the checking and correction take place upon every pair of symbols are identified but before translating them to bits.

4 PERFORMANCE EVALUATIONS

In this section, we evaluate the performance of ReflexCode using two prototypes shown in Figure 9.

4.1 Experimental Setup

We build two prototypes so as to put our evaluations into more realistic perspective:

SS We construct this small-scale prototype using three LED spotlights [2], as shown in Figure 9(a). This prototype is meant to emulate the reflected lighting [1, 4], as the spotlights are facing directly to the reflector (a billboard). Due to the small dimension and the facing direction of these spotlights, direct light VLC is impossible, which is exactly the scenario where ReflexCode is supposed to work. Our current fixture puts those three spotlights at an adjacent distance of 7cm and a 20cm distance from the wall.

LS This large-scale prototype, shown in Figure 9(b), is mimicking a normal direct lit room (3.4 × 3.2 × 3m$^3$ in dimension), where direct light VLC is possible but throughput confined by the dimension of the luminaires. Applying ReflexCode to

![Figure 8: Illustrating for determined judging points in a packet.](image)

![Figure 9: Two prototypes of ReflexCode.](image)
this scenario demonstrates the advantage of reflected VLC even under direct lit environments. We put the 3 LED luminaries on the ground and receive data through the reflected light on the wall; this has exactly the same effect as putting the luminaries on the ceiling but it allows for our easy access to the system for debugging. We assemble each luminaire using COTS LED strips [3]: it is made of 144 LED chips with an overall dimension of $30 \times 8\text{cm}^2$. The luminaires are arranged in a row with an 80cm adjacent distance and a 100cm distance from the wall.

For both prototypes, we employ the self-built LED driver consisting of low-cost transistors and a MSP430 MCU to drive the LED luminaries. We fix the transmitter-reflector distance as it is not a parameter to be controlled by our VLC service that piggybacks on an existing lighting infrastructure.

The two prototypes share most system parameters, in particular packet size and transmission frequency (up to 6kHz). A ReflexCode packet has a preamble of five successive S-3 and a single S-0 to indicate the header, followed by 24 data bits (8-bit packet sequence number and 16-bit payload), and finally ends with a S-0. We employ a Nexus 6 smartphone as the receiver, and build demodulation due to the significantly shrunk width of bands, as shown in Figure 6. Therefore, we first evaluate performance of ReflexCode under different frequency in terms of Packet Reception Ratio (PRR) and Packet Error Rate (PER): while the former is the ratio between the successfully identified packets (those between two consecutive headers) and total transmitted ones, the latter is the percentage of wrongly demodulated packets out of all successfully identified ones. Each experiment consists of 10 sessions and every session contains 300 packets (before FEC); we report the average outcome over all sessions, except for throughput where maximum values are reported too.

### 4.2 Frequency Impact on Demodulation

As mentioned in 3.2, a higher transmission frequency (thus a higher symbol rate) may yield a higher data rate because one frame/RoI can carry more data bits. However, higher frequency complicates demodulation due to the significantly shrunk width of bands, as shown in Figure 6. Therefore, we first evaluate performance of ReflexCode under different frequency in terms of PRR and PER with SS. We vary the frequency from 3kHz to 7kHz. We compare the results of ReflexCode with that obtained by a baseline of width-driven scheme described in Section 3.6, and comparisons are shown in Figure 10.

Whereas it is understandable that increasing frequency may cause complications in decoding individual symbols and hence result in a reduced PRR, it is less intuitive that lowering frequency may again reduce PRR. In fact, lowering frequency can reduce the chance of capturing a whole packet within a frame due to the widened symbols and the temporal asynchrony between transmitters and receivers. Consequently, the number of incompletely received packets gets increased and thus causing a lower PRR, though decoding individual symbols are made easier. The joint effect of these two aspects causes the concave shape of PRR as shown in Figure 10(a). PER increases with the frequency due to the increased transience of the symbols, yet ReflexCode manages to maintain a relative low PER even at 7kHz. In general, ReflexCode outperforms the baseline especially at a higher frequency. We fix the transmission frequency at 6kHz hereafter.

### 4.3 Performance under Ambient Light

As variance in ambient light (caused by sunlight through window or other non-VLC luminaires) may affect the contrast of a frame, we study the impact of ambient light on the performance of ReflexCode by introducing an extra luminaire beside SS. We vary the illuminance from 700lux (ReflexCode only) to 1200lux (combined effect of ReflexCode and the extra luminaire), which is monitored by a smartphone with a light meter [12] at a 40cm distance from the reflector. Adopting a common method to deal with ambient light in VLC, ReflexCode employs a high-pass filter to filter low frequency noises (in particular the DC component) caused by ambient light as mentioned in Section 3.5. We compare the results with and without such a filter and report PRR and PER in Figure 11. Apparently, using the high-pass filter makes ReflexCode largely insensitive to the interference of ambient light.

### 4.4 Effect of Transmitter Spacing

Since the light overlapping area on the reflector is crucial to the quality of GSK, we study how the adjacent distance between the transmitters affects the performance of ReflexCode based on LS. We vary the adjacent distance between transmitters from 0.2m to 0.8m,
while the receiver is fixed at a 2.0m distance from the reflector. We also compare camera auto-ISO with a constant 350 ISO. Intuitively, both PRR and PER degrade slightly due to the increased propagation distance from side transmitters (thus a reduced signal strength). These intuitions are fully corroborated by the results shown in Figure 12. Though a denser deployment may potentially achieve a higher throughput, we hereafter fix the adjacent distance of LS to 0.6m to closely emulate a real indoor scenario.

### 4.5 Impact of Various Reflectors

In this section we test the robustness of ReflexCode against various reflectors, as realistic scenarios may not always offer us a plain-white reflector, although a ReflexCode user is recommended to properly choose a reflector (e.g., wall or ceiling) free of major blockage. We fix the receiver at a 40cm distance from the reflector for SS, and we compare camera auto-ISO with a constant 120 ISO.

We first test ReflexCode under five typical colors for the reflector, namely light green (LG), pink (PK), sky blue (SB), white (WH) and yellow (YL). As shown in Figure 13, ReflexCode is quite robust against color differences from WH except PK (much less common in our daily life), which seems to suggest that the red tone is particularly harmful to the tonal range in grayscale. In addition, auto-ISO always outperforms the fixed ISO, suggesting that auto-ISO is maintaining the maximum tonal range in a frame under various colors.

We then test ReflexCode with a textured reflector by attaching strips with various colors on the white reflector. In particular, we arrange these strips into six typical patterns, namely horizontal (HT), square grid (SG), left hatching (LH), cross hatching (CH) and right hatching (RH) and vertical (VT). The results reported in Figure 14 are rather intuitive concerning HT and VT: ReflexCode is insensitive to HT but is subject to the interference from VT, as GSK’s grayscale bands are inherently vertical. The performances under the three hatching patterns are more curious and hence tricky to interpret: they seem to be strongly affected by the ISO setting, and we suspect that this might be an artifact introduced by individual camera designs. In general, ReflexCode should avoid operating under a reflector with visible textures to guarantee its VLC service quality, although it can still manage to transmit data at a lower rate with the help of Raptor coding.

### 4.6 Channel Property

Based on the standard settings determined by the aforementioned studies, we hereby investigate ReflexCode’s channel property (in
As the transmitter-reflector distance is fixed, we vary the reflector-receiver distance from 0.2m to 1.2m for SS and from 1m to 3m for LS, and the results in terms of PRR and PER are shown in Figure 15. As the reflected light attenuates with distance, a longer distance causes a lower signal strength at the receiver and hence shrinking the dimension of effective RoI. Consequently, it is natural to observe that the performance of ReflexCode degrades with an increased distance for both prototypes. Since LS employs more powerful transmitters, the performance degradation is made less severe, allowing the transmission distance to reach up to 3m. The performance of ReflexCode does not appear to be obviously sensitive to ISO settings, possibly because the tonal range decreases with distance in a rather consistent manner.

4.6.2 Degradation with Viewing Angle. We test the performance of ReflexCode by varying the viewing angle between $[-60, 60]^\circ$ yet maintaining the same distance from the reflector center (0.4m for SS and 1.5m for LS), emulating realistic scenarios where a user may not face the reflector perpendicularly. As we would expect, Figure 16 shows that the channel quality degrades due to the deformed RoI that in turn affects the quality of the grayscale symbols. As the transmitters of LS offer a higher signal strength, this degradation appears to be minor for LS but becomes rather prominent in SS. The asymmetry of both PRR and PER may be caused by the minor inconsistency in deploying the transmitters and also in positioning the receiver during a test. Generally, we could conclude that ReflexCode maintains a rather good channel quality within a wide viewing angle. In reality, users may not bother to make a few more steps if they feel uncomfortable with the system performance.

4.7 Throughput

We finally investigate the throughput offered by ReflexCode. Whereas the channel property is evaluated right after the demodulation, the throughput is studied from the application perspective: it is calculated as totally recovered data bits after Raptor decoding divided by the total transmission time. The transmission time is defined as the time span from receiving the first frame till all data bits of a message get decoded. We conduct experiments with $k = 26$ and $k = 50$ for Raptor coding, and report both peak and average throughput for each experiment. We also compare the performance of ReflexCode (RC) with a baseline (BL) in which all three transmitters transmit identical Raptor coded packets synchronously with OOK/Manchester modulation at a frequency of 6kHz. We refrain from comparing ReflexCode with direct LED-Camera VLC systems, because the outcome is very expectable: the dimension of an LED transmitter used in ReflexCode is rather small, leading to a very low data rate under LS or even unsuccessful communications under SS, according to the throughput computed in [40]. Since there is no obvious difference between auto-ISO and a fixed ISO during the previous studies, we adopt auto-ISO in all the following tests.

4.7.1 Throughput vs. Distance. We first evaluate the throughput of ReflexCode under a varying distance for both SS and LS, and we report the results in Figure 17. It is expectable that both ReflexCode and the baseline have their throughput (both peak and average) reduced with an increasing distance, which accords with the trend of channel quality reported in Section 4.6.1. ReflexCode can achieve a maximum throughput of 3.5kb/s and 3.8kb/s under...
both SS and LS, respectively, which are significantly higher than the corresponding values achieved by the OOK baseline. Moreover, the benefits of GSK over OOK become more evident with a larger distance, possibly because the higher bit-per-symbol ratio of GSK shortens the packet length that in turn allows more packets in one RoI. Again due to the more powerful transmitters, LS maintains a rather stable throughput with an increasing distance, whereas SS has a more evident decrease in throughput. Nevertheless, SS has a lower variance in performance (peak and average throughputs are close to each other) at a fixed distance, thanks to the better light focusing offered by the optical condenser used in the spotlight transmitters. We also observe that the throughput of ReflexCode with \( k = 50 \) is slightly higher than that with \( k = 26 \), indicating that the error rate of ReflexCode’s demodulation is good enough to allow for a less aggressive FEC setting. Overall, ReflexCode can deliver an average throughput up to 3kb/s in a room scale scenario, strongly demonstrating its potential for realistic LED-Camera VLC applications, such as advertisement/coupon delivery in a shopping mall or emergency message broadcast in an indoor public area.

4.7.2 Throughput vs. Viewing Angle. We then report in Figure 18 the evaluation results under a varying viewing angle. Similar to what we have observed with a varying distance, we have the following observations from this set of experiments: i) the outcome agrees with the channel property studied in Section 4.6.2, ii) LS is much more robust than SS against the viewing angle due to the more powerful transmitters and the resulting larger reflecting area, and iii) more importantly, ReflexCode significantly outperforms the baseline under all viewing angles. We actually observe some abnormal trends in throughput that appear to be counter-intuitive: Figure 18(c) shows a slightly lower throughput at zero viewing angle (counter-intuitive results in a similar vein appear in Figure 17(c) too). We believe that this anomaly may be caused by the auto-ISO setting that can potentially respond to lighting variation in an unpredictable manner. Nevertheless, we would still recommend using auto-ISO as it does not really cause a major problem to ReflexCode, and we believe that the manufacturers should have configured it for delivering the best overall performance.

5 RELATED WORK
There are mainly two trends of COTS-based VLC from the system development perspective, namely Screen-Camera [10, 11, 16, 23, 34, 35, 42] and LED-Camera [9, 13, 14, 19, 20, 40]. Whereas the former requires a specific device, the screen, but has a potential to achieve a higher throughput (at a cost of a reduced communication range), the latter only requires pervasively available lighting infrastructure. Due to the fundamental differences between them, we only focus our discussions on the latter that is relevant to ReflexCode. Note that, although the theoretically well-studied LED-PD VLC has got a few system avatars recently [6, 21, 33], we do not count them as COTS-based VLC because interfacing PD to COTS user devices (e.g., smartphones) is nontrivial beyond receiving simple messages [21].

LED-Camera VLC started with reflected light as the communication media [9, 28]. These seminal systems make use of only one LED transmitter to illuminate a reflector, and the applied modulation scheme (OOK in particular) is rather rudimentary. These lead to a very low data rate at around 100bps. Note that these earlier proposals also noticed the nonuniform distribution of reflected tonal range within a frame, yet they resort to polynomial regression to fit envelope, entailing unnecessary computational cost at the receiver side. Later proposals turned to direct light as an alternative. RollingLight [20] pioneered in demonstrating the possibility of constructing such a system, inspired by Luxapose [19] devised as a position system. To further improve the data rate, ColorBars [14] proposes to modulate data with Color-Shift Keying (CSK) at a cost of shortened communication range, while CeilingCast [13] suggests to combine multiple transmissions using rateless codes and also derives a simple model to characterize the performance bound of LED-Camera VLC.

As a communication system, coding and signal processing are crucial to VLC. Unlike LED-PD VLC where sophisticated RF techniques such as MIMO and UWB can be applied [6, 33], LED-Camera VLC is largely confined by the ability of its receiver (the rolling shutter camera). As a result, CSK [14] and GSK used by ReflexCode appear to be two major upgrading schemes from basic OOK. Another aspect of coding is DC balance. As VLC systems are piggybacking on existing lighting infrastructure, one has to guarantee that VLC transmission would not break lighting constraints, especially not to cause flicker. As the RLL codes recommended by IEEE 802.15.7 [30] are taken from existing RF systems, recent proposals also aim to improve the performance of RLL for VLC: while [36] and [25] explore the serial concatenation of RLL and FEC codes to have a unified coding solution, [27] applies finite state machines to designing RLL codes so as to simultaneously achieve coding gain and flicker mitigation. Unfortunately, RLL codes may not always apply to a realistic VLC system. For example, Colorbars [14] still needs specially white light as inserted illumination symbols.
to make up for color offset resulted from the random combination of symbols, even when different symbols are distributed with the same proportion. Similarly, ReflexCode’s GSK faces flicker at both symbol and transmitter levels, necessitating the need for a new coding mechanism.

Instead of acting as a data service, VLC finds its applications in other areas such as indoor localization [19, 21, 29, 38] and motion tracking [22, 41, 43]. In fact, the Visible Light Positioning (VLP) systems can be deemed as basic VLC systems from the communication perspective, as light beacons used by VLC are transmitting data at a relatively low rate to identify themselves. Therefore, the innovations of VLP lie in their ability of deriving the receiver’s location based on the received messages from multiple beacons. As ReflexCode does not require any down light beacons, it has a potential to extend VLC. Motion tracking actually utilizes on the dual of VLC, namely Visible Light Sensing (VLS) [22, 41, 43]: it receives transmissions modulated by the variance in communication media and deduces information from them. Existing proposals all sense the direct light, so the advantage in reflected light taken by ReflexCode may broaden the scope of VLS applications.

6 CONCLUSION

In this paper, we have presented ReflexCode, a novel yet practical LED-Camera VLC system that adopts reflected light as its communication media. Compared with the existing peers, Reflex has improved on throughput substantially, mainly due to its innovation on exploiting collaborative transmissions to form GSK, a modulation scheme based on grayscale. In order to mitigate flicker and also to demodulate GSK efficiently, we have equipped ReflexCode with a new DC balanced coding adapting to GSK and a robust demodulation scheme driven by local variations in grayscale. Using two prototypes with different scales, we have validated the practicality of ReflexCode under various application scenarios, and we have also demonstrated its efficacy in throughput: it achieves a peak rate of almost 4kb/s at 2m distance and maintains an average rate of around 3kb/s at various distances up to 3m.

We are on the way to test higher order GSKs for further boosting data rate. However, one needs to be careful here, because a higher-order modulation means more LED luminaires in a VLC cell, which, on one hand, may not be very realistic and the other hand, increases the distance from some LED transmitters to the reflector. This latter effect causes a lower SNR and a higher error rate in demodulation. Therefore, although it is theoretically sound to raise the modulation order, we would anticipate such a system to be more complicated and also offers a much shorter transmission distance between the reflector and the receiver.

Two other issues remain on our agenda to further study: i) energy efficiency and ii) mobility support. While the transmitter side is known to be energy efficient, camera-based VLC is known to have a relatively high consumption on the receiver side. However, as LED-Camera VLC is not using the full functionality of the camera (as opposed to Screen-Camera VLC), there is a room for optimizing the energy consumption by only activating those necessary functions. As a wireless communication technology, VLC must also offer mobility support [5, 39]. Since ReflexCode is supposed to collaboratively operate multiple transmitters within a cell, mobility cross cell boundaries requires an adequate handover. We are planning to deploy a large testbed with multiple ReflexCode prototypes to facilitate our investigation on supporting mobility under VLC.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their constructive feedback and valuable input. Special thanks to our shepherd, Prof. Zhang Xinyu, for his insights and time. This work was supported in part by the AcrRF Tier 2 Grant MOE2016-T2-2-022 and the DSAIR Center at NTU.

REFERENCES

[2] Low Cost LED Lamp. https://item.taobao.com/item.htm?spm=a1z09.2.0.0.rUBQRc&id=520120735357&u=95pl6ee7c.