SmartVLC

Co-Designing Smart Lighting and Communication for Visible Light Networks

Wu, Hongjia; Wang, Qing; Xiong, Jie; Zuniga, Marco

DOI
10.1109/TMC.2019.2915220

Publication date
2020

Document Version
Final published version

Published in
IEEE Transactions on Mobile Computing

Citation (APA)
Wu, H., Wang, Q., Xiong, J., & Zuniga, M. (2020). SmartVLC: Co-Designing Smart Lighting and Communication for Visible Light Networks. *IEEE Transactions on Mobile Computing*, *19*(8), 1956-1970. Article 8708935. https://doi.org/10.1109/TMC.2019.2915220

SmartVLC: Co-Designing Smart Lighting and Communication for Visible Light Networks

Hongjia Wu, Qing Wang, *Member, IEEE*, Jie Xiong, *Member, IEEE*, and Marco Zuniga, *Member, IEEE*

Abstract—Visible Light Communication (VLC) based on LEDs has been a hot topic investigated for over a decade. However, most of the research efforts assume the intensity of LED light is constant. This hypothesis is *not* true when *Smart Lighting* is introduced to VLC, which requires LEDs to adapt their brightness based on the intensity of natural ambient light. Smart lighting saves power consumption and improves user comfort. However, intensity adaptation severely affects the throughput performance of data communication. In this paper, we propose *SmartVLC*, a system that can maximize the throughput (benefit communication) while still maintaining the LEDs' illumination function (benefit smart lighting). A novel Adaptive Multiple Pulse Position Modulation (AMPPM) scheme is proposed to support fine-grained dimming levels to avoid flickering while maximizing the throughput under each dimming level. SmartVLC is implemented on off-the-shelf commodity hardware. Several real-life challenges in both hardware and software are addressed to make it a robust real-time system. Comprehensive experiments are carried out to evaluate the system performance under multifaceted scenarios. Experimental results demonstrate that SmartVLC supports a communication distance up to 3.6m, and improves the throughput achieved with two state-of-the-art approaches by 40 and 12 percent on average, respectively, without bringing any flickering to users.

Index Terms—Visible light communication, illumination, flickering-free, AMPPM, system design, implementation, evaluation

1 Introduction

Lighting consumes around one-fifth of the world's electricity and produces carbon emissions that are comparable to the global automobile fleet. [2] An effective way to reduce this high energy footprint is to use *smart lighting systems*, whose market is projected to exceed 47 billion US dollars by 2020. [3] These systems adjust the illuminance of artificial lights (usually, LED lights) based on the contribution of ambient natural sunlight in our environments. [4] Smart lighting systems are expected to maintain a constant illumination (and should not fall below 500 lux according to the ISO 8995-1 illumination standard [5]) within the considered areas, e.g., an office, to improve energy-saving on illumination as well as user-comfort. As the intensity of the natural sunlight changes continuously, *a key requirement for smart lighting systems is to have fine-grained dimming levels*. This fine granularity allows maintaining a constant illumination when natural light changes and to make sure the intensity of artificial lights is changed gracefully, without causing flickering to users.

Smart lighting systems are suitable to achieve user comfort and energy saving at the same time, but they are facing a new challenge: nowadays artificial lighting is not only expected to provide *illumination*, but also *wireless communication*. Over the past decade, Visible Light Communication (VLC) has attracted significant attention from both industry and academia. In VLC, data is transmitted by modulating optical sources such as standard LED luminaries. [6], [7] LED lights can be turned on and off rapidly, in the order of million times per second. At this high modulation frequencies, human eyes do not perceive any flickering effects (the basic service of illumination is not disturbed), but photosensors are able to decode information at throughput ranging from Kbps to Mbps. VLC is a promising technology that is enabling a new generation of services, such as LiFi (Internet connectivity through luminaries) [8] and accurate indoor localization. [9], [10] *Most VLC systems, however, assume that the intensity of the light emitted from LEDs is constant, [11], [12], [13] but this assumption is not correct for smart lighting systems.*

Few studies look at the intersection of smart lighting and visible light communication. Contrary to RF communication, where the carrier is modulated *solely* for data communication; with visible light, the carrier is modulated to achieve the dual goal of providing controlled illumination for users and high throughput for devices. However, we observe that the more we control the carrier to achieve fine-grained dimming for user-comfort, the less control we have for data communication (lower throughput); and vice versa.

Research Problem. There is a trade-off between fine-grained dimming levels and high throughput.

To clarify this trade-off, let us have a birds-eye-view of the limitations of the state-of-the-art approaches. In a plain vanilla smart lighting system, the light intensity is controlled solely to adjust the dimming level. To convey information via

- *H. Wu is with the Simula Research Laboratory, Fornebu 1364, Norway. E-mail: hongjia@simula.no.*
- *Q. Wang is with KU Leuven, Leuven 3000, Belgium. E-mail: qing.wang@kuleuven.be.*
- *J. Xiong is with the University of Massachusetts at Amherst, Amherst, MA 01003. E-mail: jxiong@cs.umass.edu.*
- *M. Zuniga is with the Delft University of Technology, Delft 2628, CD, the Netherlands. E-mail: m.a.zunigazamalloa@tudelft.nl.*

Manuscript received 7 Aug. 2018; revised 16 Apr. 2019; accepted 26 Apr. 2019. Date of publication 7 May 2019; date of current version 1 July 2020. (Corresponding author: Qing Wang.)
Digital Object Identifier no. 10.1109/TMC.2019.2915220

VLC, light fixtures need to be further enhanced to modulate information in frames. Broadly speaking, the efforts at the intersection of smart lighting and VLC can be classified into two groups: (i) *compensation-based*, which favor fine-grained dimming levels, [14], [15], [16] and (ii) *compensation-free*, which favor high throughput. [17], [18], [19][1] The pros and cons of these two groups are depicted in Fig. 1. In the *first* group of studies, the frames are divided into two fields: a *data* field, to modulate information; and a *compensation* field, to adjust the dimming level to the required level. This approach can control the dimming level in a fine-grained manner, but the throughput is low because only a fraction of the frame is used to modulate information and the compensation field conveys no information. The *second* group of studies aims at increasing the throughput by removing the compensation field. These studies integrate dimming control with data modulation. The limitation of the second type of approaches is that the dimming level becomes a function of the encoding scheme exclusively. Encoding schemes can provide different dimming levels, but these levels follow a step-wise function that can be very coarse.

Key Insight and Contributions. Our work builds on the top of compensation-free methods, but we provide a new perspective. Compensation-free methods set their encoding parameters to *a single set of fixed* values to reach the required dimming level as close as possible. Having a single set of fixed encoding parameters limits the ability to modulate the visible light either for communication or for illumination (the encoding parameters change only if the dimming level needs to be adjusted due to ambient light changes). Our *key insight* is to let one frame have multiple sets of *different* encoding parameters. This flexibility allows us to use interpolation techniques to combine different sets of parameters to achieve a more fine-grained control of illumination and also able to maximize the system throughput under each dimming level.

Based on the above key insight, we propose *SmartVLC*: a system that co-designs Smart lighting (illumination) and VLC (communication) for visible light networks. The concrete contributions of our work are three-fold:

- *Method [Sections 3, 4, and 5].* We propose Adaptive Multiple Pulse Position Modulation (AMPPM): a practical scheme to assign optimal encoding parameters to individual frames. The key component in AMPPM is the concept of super-symbol, in which we dither over individual symbols to achieve a dimming granularity comparable to *compensation-based* methods, and throughput higher than both *compensation-free* methods (no compensation is needed in AMPPM. And thus, the whole frame of AMPPM, except for the frame header, can be used to convey information) and *compensation-based* methods (in AMPPM multiple sets of different encoding parameters are interpolated to maximize the throughput). We efficiently utilize the property of the human eye's non-linear reaction to light intensities to reduce the number of brightness adjustments, and meanwhile guarantee flickering-freenss to users. To maximize the throughput at each dimming level, we identify the throughput envelope with a novel slope-based method at a low computational cost.
- *Platform [Section 6].* To validate the generality of our approach, we implement SmartVLC with cheap off-the-shelf commodity hardware (BeagleBone Black). The main challenges we solve are to provide high sampling rate with cheap commodity hardware and limit the search space with real-life constraints including symbol error rate upper-bound and non-flickering requirement. With these methods, we significantly reduce the amount of computational load to enable SmartVLC's real-time performance on cheap commodity hardware.
- *Evaluation [Section 7].* We present a multifaceted evaluation. First, we compare the performance of SmartVLC with state-of-the-art approaches. Our results show that our throughput is on average 40 and 12 percent higher compared to compensation-based and compensation-free methods, respectively. Second, we assess the dimming-granularity of SmartVLC (i) by exposing it to static and changing illumination conditions, and (ii) by performing user study with 20 subjects to validate that no flickering effects are present in SmartVLC.

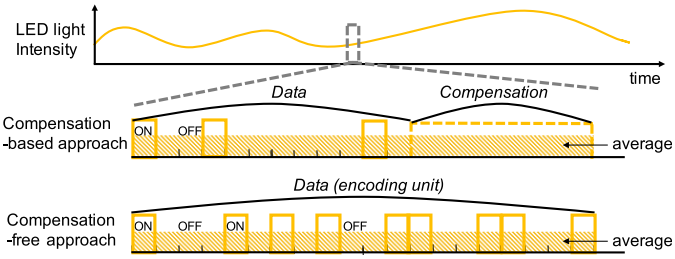


Fig. 1. Illustration of state-of-the-art approaches for co-design of illumination (with dimming support) and VLC systems.

2 Background

In this section, we introduce some background information on dimming schemes and the concept of flickering.

2.1 Dimming Schemes

To support smart lighting in VLC, some modulation schemes are proposed to enable the dimming level control of LEDs. They are divided into two categories: *analog dimming* and *digital dimming*. [15], [20] The former one increases/decreases LEDs' brightness by adjusting the forward current through LEDs which is simple to implement. However, a variation of the forward current alters the emitted wavelength of the light source, causing the color shift of LEDs. [14], [20] The latter one is based on Pulse Width Modulation (PWM), where dimming is achieved by adjusting LEDs' duty cycles. Thus, it bypasses the problem of color shift. Next, we introduce two popular digital dimming methods that are closely related to our work.

1. The compensation-based classification is dedicated to the payload and error check parts of the frame. As the header part contains the corresponding demodulation information at various dimming levels, the unified compensation approach is shown in standard, [14] i.e., the compensation bits are appended at the end of the header to avoid intra-frame flickering.

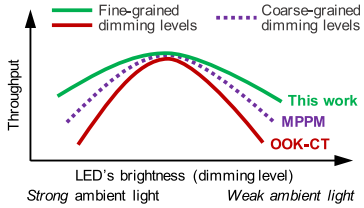


Fig. 2. High-level comparison of the communication and dimming performance of state-of-the-art approaches with our work.

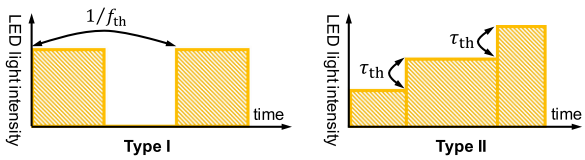


Fig. 3. Two types of flickering. *Type I:* Caused by slow changes of the ON and OFF states of an LED. *Type II:* Caused by a "big" change of an LED's light intensity in a short period.

On-Off-Keying with Compensation Time (OOK-CT). In OOK, bits 1 and 0 are modulated by turning on/off the LED, respectively. The brightness of the LED depends on the percentage of '1's in the data, and is thus fixed. To support a wide range of dimming levels, modulated data is appended with compensated consecutive ONs or OFFs, as illustrated in Fig. 1 with label *'compensation-based approach'*. This approach is also referred to as Variable On-Off Keying (VOOK). [14], [15], [16] The advantage and disadvantage of OOK-CT is illustrated in Fig. 2. The *advantage* is that it supports any dimming level by simply changing the duration of compensation period. The *disadvantage* is that the throughput degrades significantly if the targeted dimming level is either low or high.

Multiple Pulse Position Modulation (MPPM). In MPPM,[18], [19], [21], [22] bits are modulated by the positions of ONs in a symbol, as shown in Fig. 1 with label *'compensation-free approach'*. The ONs do not have to be consecutive. Under most of the dimming levels, MPPM can achieve a higher throughput than OOK-CT. The *disadvantage* of MPPM is that the supported dimming levels are restricted. It can provide fine-grained dimming level by simply increasing the duration of a symbol; however, this can increase the symbol error rate greatly, and thus, can degrade the system throughput. Detailed explanations of this performance degradation is presented in Section 4.1.1. These advantage and disadvantage of MPPM are also illustrated in Fig. 2, with a comparison to OOK-CT and our proposed scheme in this work.

2.2 Flickering

Providing flickering-free support in traditional VLC systems and smart lighting systems are both important because the primary function of LEDs is for illumination. There are generally two types of flickering, as described below.

Type-I Flickering (Frequency Domain). This type of flickering is caused by slow change of ON and OFF states at the LED, [14]. [16] If the frequency of the ON/OFF changes is high enough, i.e., exceeding a threshold f_{th} as illustrated in Fig. 3, people's eyes will not perceive the change and there will be no flickering. According to the IEEE 802.15.7 standard, [16] the minimum frequency f_{th} of turning on/off an LED is 200 Hz, above which people will not observe the flickering.

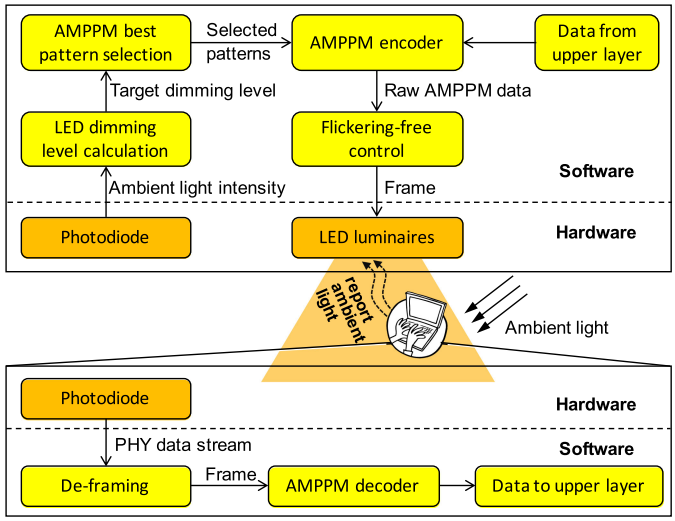


Fig. 4. SmartVLC system architecture (top: Transmitter; bottom: Receiver).

Type-II Flickering ("Intensity" Domain). On the other hand, even if the frequency change is low, as long as the intensity change is small enough, i.e., below a threshold τ_{th} as illustrated in Fig. 3, people will also not perceive the change of light. So the second type of flickering is caused by a slow "big" change of the LED's light intensity, i.e., the LED does not adjust its light intensity smoothly. The study of τ_{th} is conducted in Section 7.3.

3 System Overview

The architecture overview of SmartVLC is shown in Fig. 4. As in a typical VLC system, it is composed of a transmitter and receivers. The transmitter works as follows:

1) Upon receiving the data from upper layers, the transmitter first updates its knowledge about the environment – the real-time intensity of ambient light (at both the transmitter and receivers). Based on the illumination requirement of the area-of-interest, the transmitter calculates the required dimming level of the LED, i.e., the brightness of the light that should be emitted from the LED to maintain the *sum* of ambient light and LED light to be constant.
2) The proposed scheme then selects the best parameters to modulate upper layer data to maximize the throughput under the current required dimming level.
3) When sending modulated data, the LED's brightness is adjusted to the required dimming level. The header of each MAC layer frame is also adjusted accordingly based on the aimed dimming level.
4) The frames are sent by modulating the LED light.
5) The ACK frames and the real-time ambient light conditions sensed by receivers are sent from the receivers to the transmitter via a Wi-Fi module.

At the receiver, light signals are detected by a photodiode. The receiver first decodes the header of each frame to obtain the current parameters of AMPPM (which is adaptive based on ambient light). Then the receiver extracts the data carried by the frames and passes them to upper layers.

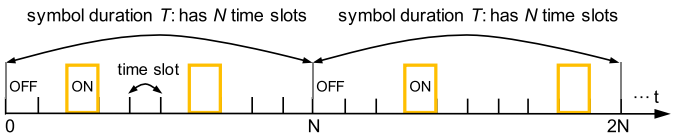


Fig. 5. Illustration of the definitions of the symbol, time slot, pulse width, etc. In this example, we have N=10, l=0.2, and the symbol pattern is $S(10, 0.2)$.

Before presenting the design details of SmartVLC, we first give the key definitions that are used throughout this work (some of them are illustrated in Fig. 5):

- *Time slot*: denoted as t_{slot}. For a specific hardware, t_{slot} is *fixed*. The system can turn on/off the LED at a maximum rate of $1/t_{\text{slot}}$.
- *ON/OFF state*: *meta* states of an LED, achieved via turning on/off the LED for a time slot t_{slot}.
- *Symbol*: a *group* of N time slots composed of ON and OFF states that *together* represent one or several *data bits*.
- *Symbol duration*: denoted as T, and $T = Nt_{\text{slot}}$. Its value depends on the modulation and is a *variable*.
- *Dimming level*: denoted as l. It indicates the brightness of the LED. Mathematically, l can be written as

$$l = \frac{\text{number of state "ON" in a symbol}}{N}. \tag{1}$$

  The range of l is [0,1]. For example, l=0.5 means that the LED's brightness is 50 percent of the maximum value.
- *Resolution of dimming level*: the difference between two consecutive dimming levels. For instance, for a set of dimming levels $[0.1, 0.2, \ldots, 1.0]$, the resolution is 0.1.
- *Symbol pattern*: denoted as $S(N, l)$. N is the number of time slots and l is the dimming level of that symbol. Note that in this work, a symbol pattern *does not* refer to the specific distribution of ON and OFF in the symbol.

4 System Design

In this section, we present the detailed design of SmartVLC.

4.1 Support Fine-Grained Dimming Levels

4.1.1 Why MPPM does not Work?

In MPPM, if a symbol consists of N time slots, then the resolution of dimming level is $1/N$. Moreover, if K slots are ONs among the N slots, then in theory the achievable throughput R is given by [18]

$$R = \frac{\lfloor \log_2 \binom{N}{K} \rfloor}{N \cdot t_{\text{slot}}}(1 - P_{\text{SER}}) \qquad \text{bit/s}, \tag{2}$$

where P_{SER} denotes the Symbol Error Rate (SER). *To provide fine-grained dimming levels, a straightforward solution is to increase the N. However, a larger N brings in a higher P_{SER}.* Let us assume the probabilities of decoding an OFF and an ON incorrectly are denoted as $P_1(N, K)$ and $P_2(N, K)$, respectively. In reality, $P_1(N, K)$ and $P_2(N, K)$ are *variables* that heavily depend on the dimming levels. [18]

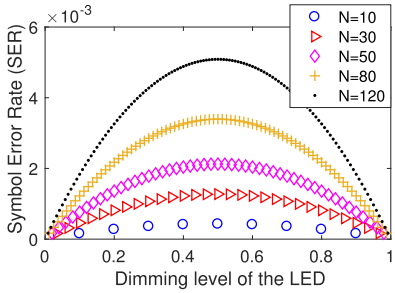


Fig. 6. P_{SER} as a function of dimming level in MPPM.

Empirically, they can be expressed as below:

$$\begin{cases} P_1(N, K) = \frac{K}{N}\overline{P_1} \\ P_2(N, K) = \left(1 - \frac{K}{N}\right)\overline{P_2} \end{cases}, \tag{3}$$

where the $\overline{P_1}$ and $\overline{P_2}$ are constant state-detection-error probabilities. [18] To decode a whole symbol correctly in MPPM, all the ON and OFF states need to be detected correctly[22].[2] Therefore, the symbol error rate P_{SER} can be written as

$$\begin{aligned} P_{\text{SER}} =& 1 - \left(1 - P_1(N, K)\right)^{N-K}\left(1 - P_2(N, K)\right)^{K} \\ =& 1 - \left(1 - \frac{K}{N}\overline{P_1}\right)^{N-K}\left(1 - \left(1 - \frac{K}{N}\right)\overline{P_2}\right)^{K}. \end{aligned} \tag{4}$$

Eq. (4) implies the existence of trade-off between N (higher dimming level resolution can be achieved with larger N) and P_{SER}. Fig. 6 illustrates this relationship, where $\overline{P_1}$ and $\overline{P_2}$ are set to 9×10^{-5} and 8×10^{-5}, respectively, which are measured in our experiments (the measurement details will be presented in Section 7.1). We can observe that a larger N leads to higher P_{SER}. Therefore, we obtain the following conclusion: *MPPM can provide fine-grained dimming level by simply increasing the N value, but this reduces the system's other performance greatly, e.g., symbol error rate.* Furthermore, based on the experimental results we will present in Section 7, the resolution of fine-grained dimming levels should be at least 0.003 (the light intensity is between 0 and 1). Thus, the N of MPPM should be at least larger than 333, which could bring a huge symbol error rate. Therefore, we should not simply use a large N for fine-grained dimming levels.

4.1.2 Increase Dimming Levels via Dithering

To provide fine-grained dimming levels without sacrificing the system's performance, we propose a novel method to use *dithering* of the symbols. The intuition behind the dithering is straightforward: if we combine the symbol pattern $S(N_1, l_1)$ with symbol pattern $S(N_2, l_2)$ equally, then we obtain a *super-symbol* where the new dimming level l_{new} can be written as

2. We assume that there is no error correcting code, such as Reed Solomon, within each MPPM symbol. Applying error correcting code to each MPPM symbol brings high overhead when N is large and may not be useful. For example, as we will show in Section 7, the resolution of fine-grauined dimming levels for a non-flickering system should be at least 0.003. This means that N, the number of slots in a APPM symbol, should be at least larger than 333. Then, the maximal symbol error rate (when the dimming level is 0.5) could reach as high as 1.4 percent. Thus, a long error correcting code will be needed and it may not help achieve a stable system at all.

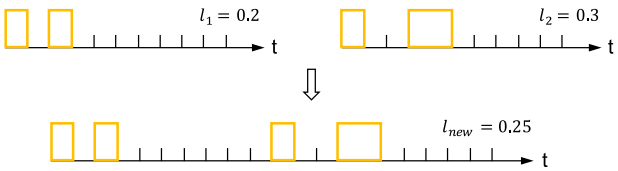
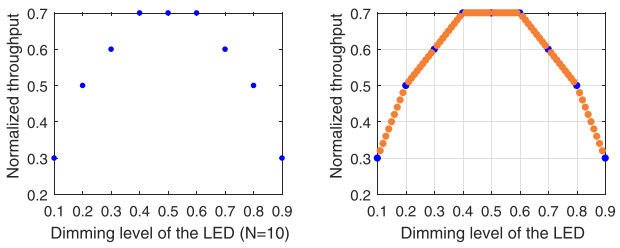


Fig. 7. Using dithering to achieve fine-grained dimming levels (in this example, N=10).



(a) Before dithering (b) After dithering

Fig. 8. Supported dimming levels before/after applying the proposed dithering approach (the data points highlighted in orange color are newly created dimming levels through the proposed dithering).

$$l_{\text{new}} = \frac{N_1 l_1 + N_2 l_2}{N_1 + N_2} \ . \tag{5}$$

For instance, when $N_1 = N_2 = 10$, the number of dimming levels supported by the system is nine, i.e., $0.1, 0.2, \ldots, 0.8, 0.9$ and the resolution is 0.1. Via dithering, we can append a symbol with pattern (10,0.2) to a symbol with pattern (10,0.1), which generates a super-symbol that has a dimming level of 0.15, as illustrated in Fig. 7. The number of slots in the super-symbol is now 20. Furthermore, this dithering process *does not* increase the symbol error rate because each symbol in the super-symbol will be decoded separately. Note that the resolution after dithering now becomes 0.05 instead of 0.1. We can have an even more fine-grained resolution by dithering more than two symbols. For example, for a dimming level of 0.175, we can append three symbols with pattern (10,0.2) to a symbol with pattern (10,0.1). Then the new resolution becomes 0.025.

The achieved fine-grained resolution after dithering is visually presented in Fig. 8, where the x-axis is the dimming level, and the y-axis is the normalized throughput. In Fig. 8a, only nine discrete dimming levels are available. After applying the proposed dithering approach, the dimming levels become 'semi-continuous', as shown in Fig. 8b.

4.2 Adaptive MPPM

With the dithering method proposed in Section 4.1, we successfully achieve fine-grained dimming levels without increasing the symbol error rate. However, *it is still not enough*. If we look back at Fig. 8, we can easily observe that *the throughput drops dramatically* from when l is 0.5 to when l is either low (e.g., 0.1) or high (e.g., 0.9), which is not what we want. To alleviate this problem, we propose a novel modulation scheme –*Adaptive Multiple Pulse Position Modulation (AMPPM)* to maximize the throughput performance while still maintaining the high dimming level resolution.

We employ a simple example to illustrate the key idea of AMPPM. We can append one symbol with pattern (10,0.1) to a symbol with pattern (10,0.2) to achieve a new dimming level 0.15. We can also append three symbols with pattern (10,0.1) to a symbol with pattern (10,0.3) to achieve the same dimming level 0.15. Though same dimming level resolution, these two ditherings bring us different throughput performance. Thus, the *key idea* in AMPPM is to *choose the best dithering to compose a super-symbol for maximum throughput*. The super-symbol satisfies the desired resolution of dimming level and at the same time achieves red highest throughput at each dimming level.

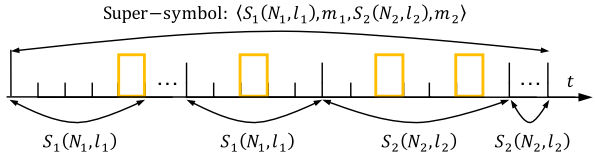


Fig. 9. Illustration of a super-symbol.

In this work, a super-symbol is formally defined as a composition of *two* symbol patterns $S_1(N_1, l_1)$ and $S_2(N_2, l_2)$. A super-symbol is generated by concatenating a number of $S_1(N_1, l_1)$ and $S_2(N_2, l_2)$, as illustrated in Fig. 9. Let S_{super} denote a super-symbol represented by the following tuple:

$$S_{\text{super}}: \ < S_1(N_1, l_1),\ m_1,\ S_2(N_2, l_2),\ m_2 > \ , \tag{6}$$

where m_i is the number of the symbol $S_i(N_i, l_i)$ in S_{super}, $i \in \{1, 2\}$. The detail composition of a super-symbol (i.e., the above tuple) is referred as the *pattern* of the super-symbol. Let N_{super} denote the number of slots in the super-symbol S_{super}, then we have $N_{\text{super}} = m_1 \cdot N_1 + m_2 \cdot N_2$. Let l_{super} be the new dimming level supported by S_{super}, then

$$l_{\text{super}} = \frac{l_1 \cdot m_1 \cdot N_1 + l_2 \cdot m_2 \cdot N_2}{N_{\text{super}}} \ . \tag{7}$$

In AMPPM, the *challenge* is to find the best super-symbol pattern at any given dimming level, i.e., choose the best N_1, l_1, m_1, and N_2, l_2, m_2 that maximize the throughput for a required dimming level l_{super}. To obtain the best super-symbol pattern, we adopt the following four steps:

Step 1: Calculate the upper bound of N_{super}

As presented above, a super-symbol consists of symbols with two different types of pattern $S_1(N_1, l_1)$ and $S_2(N_2, l_2)$. In most of the cases, l_1 and l_2 are not the same and the difference exceeds the *Type-II* flickering threshold. This difference brings in flickering, if not handled well. We are not controlling this difference when we search the best dithering symbol patterns that achieve the highest throughput, as presented later in Step 3. Luckily, we can address the problem by restricting the length of a super-symbol to make sure no flickering occurs. Recall from Section 2 that if the brightness changes quickly enough, flickering will not occur, *i.e., let the occurrence frequency of super-symbols be higher than the threshold* f_{th}. Assume that the transmitter can operate the LED at a maximum frequency of f_{tx} (i.e., $t_{\text{slot}} = 1/f_{\text{tx}}$). Let $N_{\max}$ denote the maximal duration of a super-symbol will never cause flickering, then $N_{\max}$ can be calculated as

$$N_{\max} = \frac{f_{\text{tx}}}{f_{\text{th}}} \ . \tag{8}$$

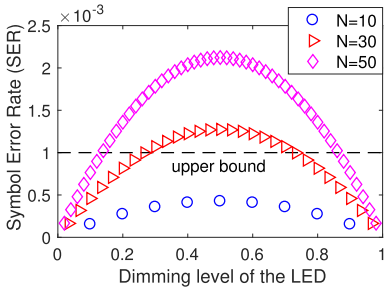


Fig. 10. Available patterns: Below the upper bound.

Step 2: Calculate the available Ns and dimming levels under the constraint of symbol error rate

As mentioned in previous sections, a VLC system suffers from higher symbol error rate with a larger N. To meet the requirement of a reliable communication system, the P_{SER} must be kept below a threshold. Therefore, not all Ns satisfying the requirement in Step 1 can be taken for further selection. The range of N must be further constrained and those lead to higher P_{SER} are abandoned. In general, for any $N \leq N_{\max}$, if the P_{SER} under the symbol pattern $S(N,l)$ is below the specified upper bound, then the symbol pattern $S(N,l)$ will be considered as a candidate for composing the super-symbol that will be detailed in Step 3. Otherwise, the symbol pattern $S(N,l)$ is discarded. For example, Fig. 10 plots the P_{SER} as a function of the dimming level for some of the Ns obtained in Step 1. The symbol patterns $S(N,l)$ above the upper bound (e.g., 1×10^{-3}, a typical upper bound in communication systems) of the P_{SER} are abandoned, e.g., $S(50, 0.3)$ and $S(30, 0.4)$ in Fig. 10. Then the throughputs with the remaining valid Ns are calculated according to Eq. (2).

Step 3: Obtain the best pattern of super-symbol

The last step is to dither symbol patterns to form super-symbol, in order to achieve the highest throughput. Before dithering, we only have symbol patterns with discrete dimming levels. If we pick the pattern which leads to the highest throughput under each discrete dimming level, the dimming level resolution is not changed, as shown by the red-dash line in Fig. 11. Further, even the achieved "highest throughput" is actually not optimal. In SmartVLC, we propose a new method to improve the throughput as well.

Through the analysis presented in Section 4.1, we know that we can achieve more fine-grained dimming levels through dithering. For a fixed dimming level, we have different pattern combinations. The proposed AMPPM scheme exploits these combinations, so not only *fine-grained dimming level resolution is achieved* but also *the throughput under each dimming level is maximized*. We describe AMPPM below in details, assisted by Fig. 11. The key idea is to find the *envelop*, namely, the blue-solid line as shown in Fig. 11. To achieve this target, the system works as follows:

1) For available symbol patterns whose dimming level is around 0.5, we find the symbol pattern that gives the highest normalized throughput.[3] In Fig. 11, the found symbol pattern is $S(21, 0.524)$ that is located at the position (0.524, 0.855).
2) On the right side of $S(21, 0.524)$, we identify the next symbol pattern which satisfies the following requirement: *when connecting this symbol pattern with $S(21, 0.524)$, the resulted slope is the smallest*. Mapped to Fig. 11, this newly found symbol pattern is $S(21, 0.571)$ that is located at the position (0.571, 0.855).
3) Repeat the previous step until reaching the maximal dimming level. Connect all the found symbol patterns in sequence, and now we obtain the *'envelop'* – the blue-solid line in Fig. 11.
4) For each pair of neighboring symbol patterns that are found through the above steps, e.g., $S(21, 0.524)$ and $S(21, 0.571)$, we dither over them to generate the super-symbols for any required dimming level in between, i.e., between the range (0.524,0.571). *Note that at most two different symbol patterns are required to compose a super-symbol.*

3. This is based on the fact that for a given symbol duration $N \cdot t_{\text{slot}}$, the symbol with pattern $S(N, \lfloor N/2 \rfloor)$ is most likely to have the highest throughput among all the symbols with the same symbol duration.

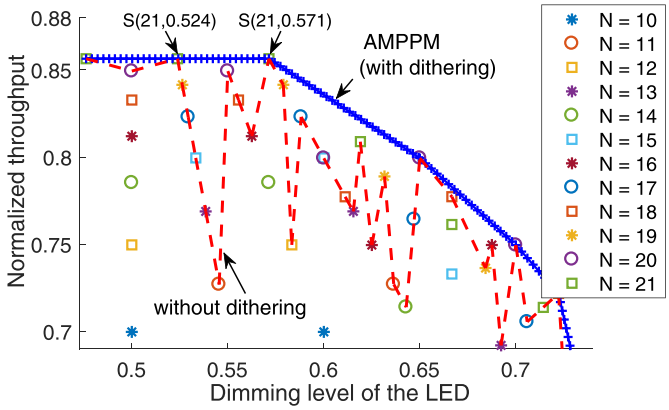


Fig. 11. Best pattern selection based on slope.

Through the above steps, AMPPM can provide fine-grained dimming levels (as demonstrated by the marker '+' on the blue-solid line in Fig. 11), at the same time can optimize the throughput under each supported dimming level.

4.3 Encoding/Decoding

To transmit the data received from the upper layers, the transmitter must encode the original data into codewords. To do this, classical methods based on pulse position can be categorized into two main groups: tabulation [23] and constellation. [24] The former records the mapping between the data stream and codeword in tabulation while the latter in constellation graphs. However, both of them are based on exhaustion search, and all the items are recorded in the memory space. In our system, with the increment of N, the number of mappings increases exponentially, which brings in extremely high computational load and large memory usage. For example, when $N = 50$ and $K = 25$, the number of mappings is $\binom{N}{K} = \binom{50}{25} \approx 1.26\text{e}+14$. If each mapping item occupies 4 bytes, a total of 126 TB memory is required, and the search space is also huge.

To solve this problem, we propose *heuristic algorithms* based on *combinatorial dichotomy* for the coding and encoding in SmartVLC, which enables direct mapping between the data stream and the codeword without exhaustion searching. The

encoding algorithm based on combinatorial dichotomy is given in Algorithm 1. It generates the codeword from the Least Significant Bit (LSB) to the Most Significant Bit (MSB). If the LSB is set to the value "1", the remaining bits in the codeword can represent $\binom{N-1}{K-1}$ different types of binary inputs. Therefore, if the value of the binary input is smaller than the value of $\binom{N-1}{K-1}$, the LSB is set as "1". Otherwise, the LSB is set as "0" and the binary input is subtracted by $\binom{N-1}{K-1}$. Next, the algorithm calculates the value of the second LSB following the same process. This iteration stops either when it is processed N times, or the K times of ONs or $(N-K)$ times of OFFs are all filled. After that, the remaining slots are set to either OFFs or ONs.

Algorithm 1. Encoding

Input: 1) N and K: number of slots and ONs in one symbol, respectively; 2) data_s: original $\lfloor \log_2 \binom{N}{K} \rfloor$-bit data stream.
Output: code_w: the generated N-bit codeword.

```
val = data_s;  i_N = 1;  i_K = 1;
while i_N <= N && i_K <= K && i_N - i_K <= N - K
do
  if val >= C(N-i_N, K-i_K) then
    code_w[i_N] = OFF; val = val - C(N-i_N, K-i_K); i_N = i_N + 1
  else
    code_w[i_N] = ON;  i_K = i_K + 1;  i_N = i_N + 1
  end if
end while
if i_K > K then
  code_w[(i_N + 1) : N] = OFF
else if i_N - i_K > N - K then
  code_w[(i_N + 1) : N] = ON
end if
```

Algorithm 2. Decoding

Input: 1) N and K: number of slots and ONs in one symbol, respectively; 2) code_w: the N-bit codeword.
Output: data_s: the decoded $\lfloor \log_2 \binom{N}{K} \rfloor$-bit data stream.

```
data_s = 0;  i_N = 1;  i_K = 1;
while i_N <= N && i_K <= K && i_N - i_K <= N - K
do
  if code_w[i_N] == OFF then
    data_s = data_s + C(N-i_N, K-i_K)
  else
    i_K = i_K + 1
  end if
  i_N = i_N + 1
end while
```

The decoding algorithm is given in Algorithm 2. It runs exactly in the opposite way of the encoding. Note that the receiver knows the symbol patterns (i.e., the values of N and K) from the header of the frames before decoding, which is presented in details in the following section. For both our encoding algorithm and the decoding algorithm, the time complexity is $O(N^2)$, and the required space complexity is $O(1)$. Our algorithms could reduce the time complexity to $O(N)$ if the results of $\binom{N}{K}$ are pre-stored in a matrix with memory size proportional to $O(N^2)$. As a comparison, the required memory size of the exhaustive-search based algorithm is proportional to $O(N!)$, and the time complexity is $O(1)$ if the mappings of raw data and codeword are saved in the memory.

TABLE 1
The Frame Format in SmartVLC

Preamble	Length	Pattern	Compensation	Sync	Payload	CRC
3 Bytes	2B	4B	X bits	1 bit	0-MAX B	2B

4.4 Frame Format

To enable communication between transmitter and receiver, a frame format must be properly designed. In SmartVLC, we design it as shown in Table 1. Each frame starts with a three-byte *Preamble* (consisting of an alternate sequence of ON and OFF) indicating the beginning of a new frame. The frame *header* comes after the preamble, and it includes the *Length* and *Pattern* fields. The *Length* field indicates the number of bytes in the payload. The *Pattern* filed occupies four bytes and carries the details about the super-symbol with the tuple $< N_1, m_1, N_2, m_2 >$.[4] These details are used by the receiver to decode the corresponding frame.

The *Compensation* and *Sync* fields are used to avoid intra-frame flickering, caused by brightness discrepancies between the header part (i.e., Preamble, Length, Pattern, and Sync) and the data part (i.e., Payload and CRC). To align the brightness of the frame header with that of the payload, *compensation time* must be appended to the frame header, as shown in Table 1. The compensation can be consecutive ONs or OFFs, depending on the brightness of the payload. After the compensation field, a *Sync* bit, that is a rising or falling edge, is appended to achieve synchronization of the frame header after the compensation. The *Payload* and a two-byte *Cyclic Redundancy Check (CRC)* are placed at the end of each frame.

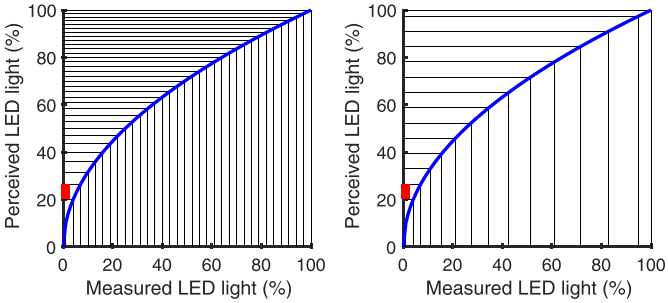
5 Adaptation to Various Ambient Light

In the above section, we present how to achieve fine-grained dimming levels and how to select the best pattern of a super-symbol to achieve the highest throughput for each dimming level. Recall that in smart lighting systems, the brightness of LED changes with the available ambient light. Therefore, an adaptation algorithm must be designed to efficiently and smoothly reach the required dimming levels. Two goals must be achieved:

Goal 1: the sum of the *ambient* and *LED* light intensity should be constant within the area of interest, i.e., $I_{\text{sum}} = I_{\text{led}} + I_{\text{amb}}$, where I_{sum} is the targeted constant light intensity that depends on users' preference. I_{led} and I_{amb} are the intensities of LED and ambient light, respectively.

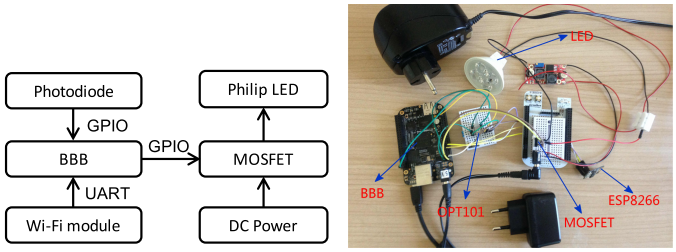
Goal 2: the adaptation process of LED light should not be observed by users, namely, flickering-free to users. Meanwhile, the number of adaptation times should be minimized to reduce the overhead of finding the optimal patterns of super-symbols.

4. This tuple is very similar to the one for S_{super} that is given in Eq. (6). The dimming levels are omitted in this tuple (i.e., not carried in the frame header) because the receiver does not need them to decode the super symbols. When needed, the receiver can calculate them from the received super symbols.



(a) Adaptation in the measurement domain (b) Adaptation in the perception domain (*this work*)

Fig. 12. Adaptation to dynamic ambient light.



(a) Block diagram (b) Prototype

Fig. 13. Implementation of the transmitter *(the ON/OFF state of the LED is controlled by the BBB's GPIO signals that are applied to the MOSFET)*.

Solution to achieve *Goal 1* is straightforward. Let us assume at time t_1, the intensities of LED light and ambient light are I^1_{led} and I^1_{amb}, respectively. At time t_2, the intensity of ambient light is decreased to I^2_{amb}. Then we just need to increase the brightness of the LED by $\triangle I_{\text{led}}$ that is given as below:

$$I^1_{\text{led}} + I^1_{\text{amb}} = I^2_{\text{led}} + I^2_{\text{amb}} \ , \tag{9}$$

$$\Longrightarrow \ \triangle I_{\text{led}} = I^2_{\text{led}} - I^1_{\text{led}} = I^1_{\text{amb}} - I^2_{\text{amb}} \ . \tag{10}$$

To achieve *Goal 2*, the brightness of LED should not be changed too much in one step as it will be perceived by the users, causing flickering. So one solution is to adjust I^1_{led} *gradually and evenly* at a step of τ which can not be perceived by the human's eyes, until I^1_{led} reaches $I^1_{\text{led}} + \triangle I_{\text{led}}$. The number of steps taken is $\lceil \triangle I_{\text{led}}/\tau \rceil$. Note that τ is a *constant* and its maximum value is constrained by the second type of flickering. This adaptation is illustrated in Fig. 12a.

In SmartVLC, we propose an even better method to achieve *Goal 2*. The main idea is to adopt a *variable* τ that can avoid the flickering and at the same time, minimize the number of steps taken to reach the target change $\triangle I_{\text{led}}$. The motivation behind this idea is that *the response of human's eyes to visible light changes is nonlinear*. In dark environment, people enlarge their eye opening and therefore induce more light coming in. In this work, we use *measurement domain* and *perception domain* to represent the brightness measured by light meters and perceived by humans, respectively. Specifically, the relationship between the perceived brightness I_p and the measured brightness I_m is [25]: $I_{\text{p}} = 100 \times \sqrt{\frac{I_{\text{m}}}{100}}$, which is also shown by the blue line in Figs. 12a and 12b.

The proposed method to achieve *Goal 2* works as below:

1) Convert the brightness I^1_{led} and I^2_{led} in *measurement domain* to $I^1_{\text{led}-\text{p}}$ and $I^2_{\text{led}-\text{p}}$ in *perception domain*

$$I^i_{\text{led}-\text{p}} = 100 \times \sqrt{\frac{I^i_{\text{led}}}{100}}, \quad i \in \{1, 2\}.$$

2) Calculate the difference between $I^1_{\text{led}-\text{p}}$ and $I^2_{\text{led}-\text{p}}$

$$\triangle I_{\text{led}-\text{p}} = I^2_{\text{led}-\text{p}} - I^1_{\text{led}-\text{p}}. \tag{11}$$

3) Increase $I^1_{\text{led}-\text{p}}$ *gradually* by a step of τ_p, until it reaches to $I^2_{\text{led}-\text{p}}$. We make sure a change of τ_p will not cause flickering. Note that with a *constant* step of τ_p in perception domain, the step change τ in measurement domain is actually a *variable*, which is illustrated in Fig. 12b.

6 Implementation

This section presents the implementation of the SmartVLC, including both the hardware and software implementations.

6.1 Hardware

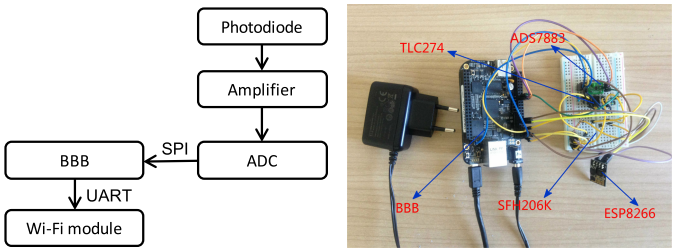
Transmitter. The block diagram of the transmitter is given in Fig. 13a. There are mainly four components: the BeagleBlack board (BBB) which costs around $60, transistor (ON MOSFET 20N06L), LED (Philips 4.7W), and the photodiode (OPT101). The GPIO of BBB triggers the transistor to modulate the LED, powered by a 24V DC voltage. Sensing the ambient light at the transmitter is done by sampling the OPT101 driven by the BBB on-board 3.3V DC voltage. Uplink from the receiver to the transmitter is implemented via the Farnell ESP8266 Wi-Fi module to transmit ACKs (to acknowledge the received frames)[5] and the intensity information of ambient light sensed by the receiver. The latter is used by the transmitter on how to adjust the dimming level of the LED in order to keep the sum of the ambient light and illumination light constant at the receiver. The prototype of the transmitter is shown in Fig. 13b.

Receiver. Its block diagram is shown in Fig. 14a. There are also mainly four components: BBB, ADC (ADS7883), amplifier (TLC274), and photodiode (SFH206K). Incoming light signals are first sensed by the photodiode and then amplified by the amplifier. Analog signals from the amplifier are converted to digital signals by the ADC and then sampled by the BBB for further process. Note that here we use SFH206K at the receiver instead of OPT101 because SFH206K can meet both the requirements of low response time and high sensitivity. The receiver can be fully powered by the BBB, which enables the possibility of unifying the receiver to the BBB for convenient testing. The prototype of the receiver is shown in Fig. 14b.

6.2 Software

Following the choice in OpenVLC, [26] we use the low-end BBB in our SmartVLC system. The *challenge* in the software implementation is two-fold: (1) how to modulate

5. We use WiFi for the ACKs only because of the fact that in practice, the field-of-view of LEDs residing at the mobile nodes are not powerful enough to support the required communication coverage. Therefore, we choose to use WiFi for the ACKs in our experiments. We can use VLC for both uplink and downlink in the future when more advanced LEDs are available for mobile nodes.



(a) Block diagram (b) Prototype

Fig. 14. Implementation of the receiver.

the LED light at as high speed as possible at the transmitter; and (2) how to sample the incoming signal as fast as possible at the receiver, with the off-the-shelf BBB platform. In, [26] Linux was patched with Xenomai to support real-time operations, which can achieve a sampling rate of up to 50 KHz. However, this speed is far away from our target because the ADC we use can support a sampling rate of 3.0 MHz.

In SmartVLC, we exploit the *PRUs (micro-controllers) of the BBB* to address the above challenge, without introducing new cost on additional hardware, such as FPGA. We cooperate the ARM processor with the PRUs via shared memories. The PRU controls GPIOs to modulate the LED light and perform the sampling at both the transmitter and the receiver. The ARM processor deals with upper layer processing, such as encoding/decoding and framing/de-framing. Through this novel implementation, we can modulate the LED light and perform sampling at speeds in the order of Mbps, satisfying our system's requirement of off-the-shelf low-cost hardware. *Note that our software implementation has been partially merged into the latest version of OpenVLC. [27]*

7 Evaluation

We evaluate the performance of SmartVLC through comprehensive experiments in real environments. We first determinate some key parameters of SmartVLC, then introduce the setups, followed by the evaluations.

7.1 Setup

Time Slot and Sampling Rate. Generally speaking, the value of t_{slot} can be restricted by both the hardware (the LED's properties) and software (how fast the system can operate the LED). In the current implementation of SmartVLC, t_{slot} is restricted by the hardware: the slow rising/falling speed of the off-the-shelf Philips LED when it is turned on/off. Therefore, in our experiments, we set t_{slot} as 8 μs (the minimal time slot the LED supports, under which the transmitted signals are not distorted too much), i.e., f_{tx}=125 KHz. At the receiver, we set the sampling rate to 500 KHz,[6] i.e., four times of f_{tx}.

Maximal Super-Symbol Duration. According to the IEEE 802.15.7 standard, [16] the minimum frequency f_{th} to turn on/off an LED is 200 Hz, under which people will not observe flickering. However, according to our findings in the experiments, f_{th} varies with different type of LEDs and with different people. To find the proper f_{th} in SmartVLC, we invite 20 volunteers (10 male and 10 female) between 19 to 41 years old to participate in the experiments, and find that setting the f_{th} to 250 Hz (slightly higher than the specification in the standard) is a safe threshold which will not cause flickering with the LED used in our system for all the volunteers. Therefore, according to Eq. (8), we have

$$N_{\max} = \frac{f_{\text{tx}}}{f_{\text{th}}} = \frac{125000}{250} = 500 \ ,$$

which means that we can have up to 500 consecutive slots for a super-symbol (Section 4.2) without causing flickering.

Synchronization. Two separate clock oscillators (or clock generators) can never be perfectly synchronized because of the oscillators' frequency deviation. The factors which may impact the deviation degree include the internal structure of clock oscillators and thermal tolerance of the oscillator circuit,[29], [30] etc. Due to this hardware artifact, the transmitter and the receiver, used for modulating the LED light and for sampling incoming signals, are not perfectly synchronized, and the decoding error can occur at the receiver. To solve this problem, we adjust at the receiver the sampling interval at rising/falling edges, which is commonly used in the literature, e.g.,. [16], [31] In our proposed AMPPM, the pattern of consecutive ONs or OFFs does exist. SmartVLC may thus lose synchronization before encountering a rising/falling edge. Therefore, it is necessary to figure out the positions of the first 'out-of-synchronization' slots, i.e., to find the maximum running length of consecutive ONs or OFFs before the system loses synchronization. Since the position depends on the hardware artifact, we identify its value through experiments. We let the transmitter send a continuous 50 percent duty cycle square waveform so that the 'out-of-synchronization' slot can be identified immediately based on the repetitive rising/falling edges. The result is shown in Fig. 15. Our experimental results show that under the worst case, our system starts to lose synchronization after about successive 337 slots. This means that our system should not have more than 337 successive slots of ONs or OFFs in one symbol (not a super-symbol) or two successive symbols. However, this threshold is far from reachable in SmartVLC because the number of slots N in one symbol is below 100 from our empirical results. Therefore, adjusting the sampling interval at rising/falling edges to synchronize the transmitter and receiver is enough for the current implemented SmartVLC system.

Detecting Errors of ONs and OFFs. The performance of the proposed AMPPM scheme depends on the surrounding noise level which affects the symbol error rate P_{SER}. To obtain the optimal patterns for the super-symbol under different dimming levels, we need to determine $\overline{P_1}$ and $\overline{P_2}$. The values of $\overline{P_1}$ and $\overline{P_2}$ can be obtained via fitting into the experimental results of P_{SER} within different N. We conduct an empirical experiment in an extreme case where the receiver is placed at a distance of 3.6 m from the transmitter and the ambient noise level is high with ceiling lights on and the window blind placed to the top. From the results

6. Note that in our software implementation, we can sample the incoming signal at the maximal rate supported by the ADC, i.e., 3 Mbps. But with the bottleneck at the LED, a sampling rate of 500 KHz is enough. The bottleneck can be addressed with a more advanced LED, for example, Micro LEDs. [28] Then, the system's throughput will be limited by the PRUs' clocks at the transmitter and the receiver, where they could be hardly perfectly synchronized due to the hardware artifact.

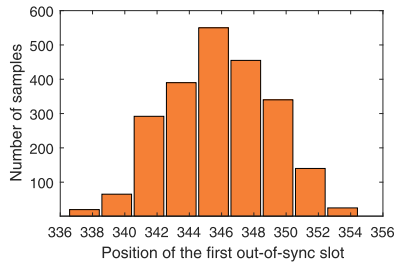
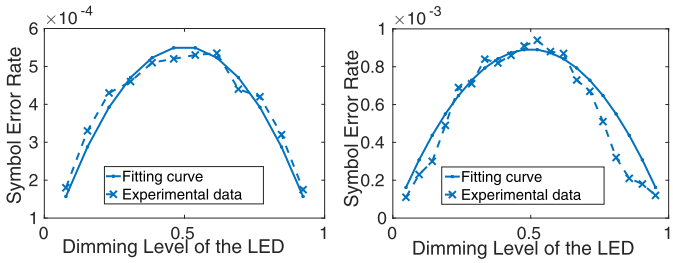


Fig. 15. Statistics of the position of the first 'out-of-synchronization' slot.



(a) N=13 (b) N=21

Fig. 16. Experimental evaluation of symbol error rate.

(a) The position of the window blind is fixed to the top

(b) The position of the window blind is increased constantly

Fig. 17. Control ambient light using window blind.

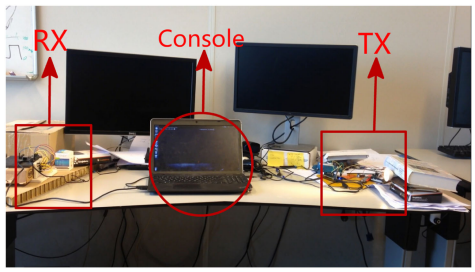


Fig. 18. Placement of the transmitter and receiver.

shown in Fig. 16 ($N = 13$ and $N = 21$), $\overline{P_1}$ and $\overline{P_2}$ are determined to be 0.00009 and 0.00008, respectively. Besides, the upper bound of the P_{SER} is set to 0.001 in our calculations to obtain the best patterns of the super-symbols. In the experiments, if the receiver detects an error through CRC check when it decodes a frame (due to symbol loss), it will drop that frame and will not send an ACK to the transmitter.

Throughput Calculation. The throughput is calculated as the number of bits (in the payload) correctly received per second at the receiver when the transmitter continuously sends frames with a dummy payload. As presented before, the frame format is shown in Table 1. The CRC in the frame is used by the receiver to check whether the frames are received correctly.

Ambient Light Control. In the experiments, we control the intensity of the ambient light in the office using the window blind, as shown in Fig. 17. We fix the window blind to a position to provide a static ambient light condition, as shown in Fig. 17a; and pull it down/up at a constant speed (this is supported by the blind in our building as it is electronically controlled) to provide a dynamic ambient light condition, as illustrated in Fig. 17b.

Deriving the Super-Symbols. The transmitter adopts the three steps presented in Section 4.2 to derive the optimal symbol patterns for the super-symbol, to satisfy the required dimming level. Note that in AMPPM, a super-symbol is only consisted of up to two different symbol patterns.

Frame Size. The payload is fixed to 128 bytes in all the experiments. The gain of AMPPM will decrease if the payload is too small. This is due to the overhead in the frame header. Note that for the same reason, the performance of all other schemes (such as OOK-CT and MPPM) will also degrade when the payload is small. Besides, in our evaluation, the probabilities of bit 0 and bit 1 in the payload are assumed to be equal, without loss of generality.

7.2 Static Lighting Scenario

We first evaluate the performance of SmarVLC in a *static* scenario and compare it with existing solutions. The static scenario is achieved by fixing the window blind to a position to provide a fixed[7] amount of ambient light. We also turn on the ceiling illumination lights in our office. The placement of the transmitter and receiver is shown in Fig. 18.

Comparison with MPPM and OOK-CT.[8] We first compare the performance of proposed AMPPM with the state-of-the-art compensation-based OOK-CT and compensation-free MPPM schemes. In the experiments, the dimming range is set from 0.1 to 0.9. The receiver is placed at a distance of 3 m from the transmitter. For MPPM, if the value of N is too large, the symbol error rate will be higher than the upper bound. To make sure that the symbol error rate in MPPM is below the upper bound (we set it to 1×10^{-3} that is a typical upper bound in communication systems, and we also use it as the upper bound for AMPPM in Fig. 10) while gaining as many dimming levels as possible within the dimming range, we set $N = 20$. We do not use $N = m_1N_1 + m_2N_2$ for MPPM in the comparison because it will lead to a large value of N, e.g., above 100, which brings very high symbol error rate as indicated in Fig. 6, and therefore, results in even lower system throughput. For OOK-CT, the brightness is adjusted by compensating the data in the payload.

The evaluation results are shown in Fig. 19. First, it is easy to observe that the proposed AMPPM outperforms MPPM under all dimming levels, and outperforms OOK-CT under 16 out of the 17 dimming levels. When the dimming level is $l = 0.1$ or $l = 0.9$, the achieved throughputs under AMPPM, OOK-CT, and MPPM are 55.6, 21.7, and 44.3 Kbps, respectively. AMPPM improves the performance of OOK-CT by up

7. The intensity of the ambient light changes with time. In a very short period of time, the change is very small, and we assume the intensity is constant.

8. Note that the performance of VPPM in terms of achievable throughput is worse than that of MPPM in theory. Therefore, we choose not to compare AMPPM with VPPM in the experiments.

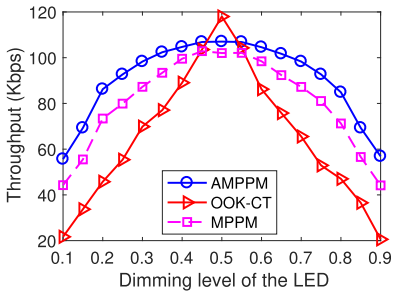


Fig. 19. Comparison with OOK-CT and MPPM.

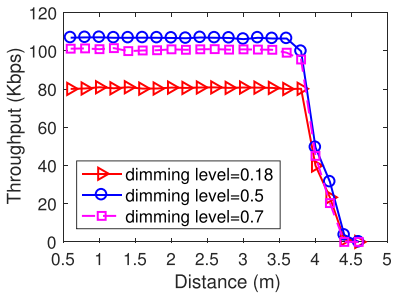


Fig. 20. Throughput versus communication distance.

to 170 percent when l is either low or high, and on average by 40 percent. For MPPM, its throughput performance can be improved by up to 30 percent (when $l = 0.9$) with AMPPM, and on average by 12 percent. This is because AMPPM always selects the best symbol pattern under each dimming level. Moreover, recall from Fig. 11, AMPPM does not just improve the throughput performance, *but also provides much more fine-grained dimming levels than MPPM* which is essential for smart-lighting systems.

Another interesting observation is that in a narrow dimming level range between 0.47 to 0.53, OOK-CT performs slightly better than AMPPM. That is because: (1) OOK-CT has very little overhead (close to 0 compensation time) when l is near to 0.5, meaning that almost all time slots are efficiently used to represent data bits; (2) AMPPM introduces a small overhead on deriving the optimal symbol patterns to compose the super-symbol (by performing Steps 1 to 3 as described in Section 4.1), as well as the four additional bytes (field "Pattern") in the frame header to describe the patterns. However, OOK-CT loses the advantage quickly under other dimming levels, and the throughput is much lower than that of AMPPM.

Throughput versus Distance. The throughput of SmartVLC is also evaluated by varying the distance between the transmitter and receiver. We test three dimming levels (0.18, 0.5, and 0.7) in the experiments. The results are plotted in Fig. 20, where each marker represents an experiment lasting for 30 seconds. We can observe that SmartVLC maintains its peak throughput at each dimming level at distances *up to 3.6m*. After this distance, the throughput drops dramatically because the received signal strengths are not large enough for the receiver to decode the signal. Considering that the height of the ceiling is usually around 2.5–3 m in a typical office, the communication distance of 3.6 m supported by SmartVLC is sufficient in reality. Besides, we observe that the dimming level of the LED does not affect the communication distance of SmartVLC. This is because the brightness of the LED is varied via duty cycles instead of the amplitudes.

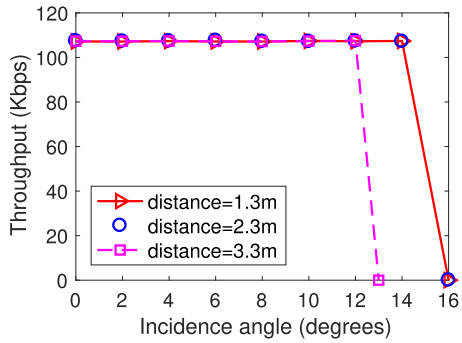


Fig. 21. Throughput versus incidence angle.

Throughput versus Incidence Angle. The performance of SmartVLC is further evaluated under different incidence angles. In the experiments, the distance between the transmitter and the receiver is kept constant when varying the incidence angles. We conduct experiments with different distances between the transmitter and receiver (1.3, 2.3, and 3.3 m), and the results are shown in Fig. 21. We can observe that SmartVLC can almost main tain its performance with the LED's Field of View (FoV). Another observation is that longer distance has a shorter cut-off incidence angle. This is because the system already reaches the upper bound of the communication distance before the incidence angle starts affecting the system.

Impact of Ambient Light. We study the impact of ambient light on the performance of SmartVLC. The ambient light condition affects the maximal communication distance of our system. To study this impact, we place the window blind to two extreme positions: (1) *at the top* (more ambient light); (2) *at the bottom* (less ambient light). The ceiling light is kept on in both the above scenarios. The dimming level of the LED is fixed to 0.5. The distance between the transmitter and the receiver is varied, and the average throughput at each spot is measured, as shown in Fig. 22. We can observe that SmartVLC reaches a longer communication distance when the blind is at the bottom, i.e., less ambient light. The reason behind this is that a higher SNR is achieved at the receiver when there is less ambient light.

Throughput Under Different Natural Light Conditions. We also study the impact of different natural light conditions on the throughput of SmartVLC. Three different natural light conditions are considered: *(1)* cloudy day (with 320–2130 lux variation); *(2)* sunny afternoon (with 450–2990 lux variation); *(3)* sunny morning (with 660–9780 lux variation. The testing room is faced to east, thus this condition can provide the strongest ambient light). Different weather conditions contribute to the various range of ambient light. Under each of the weather condition, in order to evaluate SmartVLC's performance under different dimming levels of the LED, we also place the window blind at nine different positions to provide various ambient light levels (1–9, from the weakest to the strongest). In the experimental setup, we calibrate the dimming level of the LED to the lowest when the window blind is at the top and to the highest when the window blind is at the bottom. The distance between the transmitter and the receiver is fixed to 3 m. The relationship between the position of the window blind (ambient light level) and the achieved throughput is shown in Fig. 23.

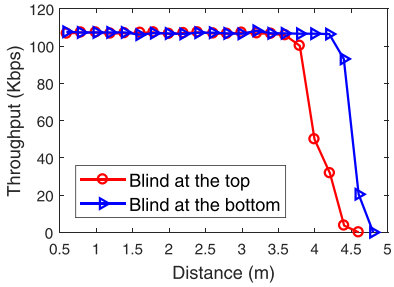


Fig. 22. Impact of ambient light on the communication distance.

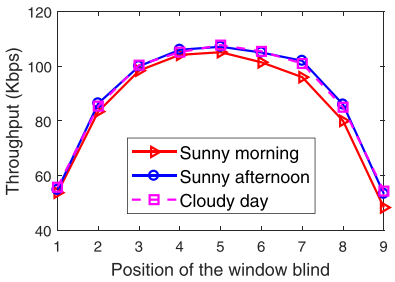


Fig. 23. Throughput under different natural light conditions.

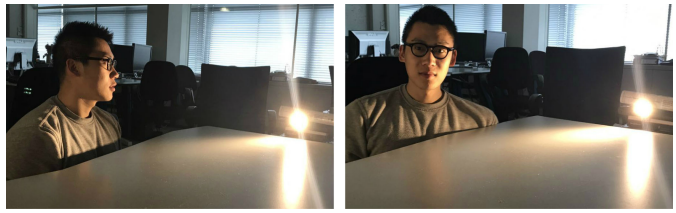
First, we observe that the shapes of the three curves are basically subject to the 'theoretical' shape of the throughput of AMPPM as presented in Fig. 19. Furthermore, we can observe that the throughput under the weather 'Sunny morning' is lower than the other two, and its shape is also not symmetric. The main reason is that the average ambient light intensity in 'Sunny morning' is stronger which thus results in lower SNRs at the receiver. Besides, since the ambient light range is larger in 'Sunny morning', the gap of the received SNRs at symmetrical blind positions is thus larger, bringing in a higher difference between the throughput under the symmetrical positions of the window blind.

7.3 Empirical Minimum Resolution of the Dimming Level

In this section, we conduct experiments to study empirically the minimum resolution of the dimming levels in perception domain that will not cause flickering.

We invite 20 volunteers, and they are instructed to observe the intensity change of the LED light in two different manners: *(i) direct viewing* , where they look at the LED directly, as illustrated in Fig. 24a; *(ii) indirect viewing* , where they judge the flickering/non-flickering based on the reflection of LED light, as illustrated in Fig. 24b. The volunteers are not aware of the resolution of dimming levels employed, and they are asked to judge whether they can perceive the flickering or not. The experiments are conducted in three different ambient light conditions: *(L1.)* a sunny day with indoor ceiling light on (8900–9760 lux); *(L2.)* a sunny day with ceiling light off (7960–8200 lux); and *(L3.)* the window blind is pulled down to the bottom, and ceiling light is off (12–21 lux).

The results are shown in Table 2. We observe that under the same viewing manner, weaker ambient light (L3) makes users more sensitive to the flickering of LED light. This is because human beings tend to enlarge their pupils in dark environments. On the other hand, users are more sensitive



(a) Direct viewing (b) Indirect viewing

Fig. 24. Users' perception of light changes.

TABLE 2
Users' Perception of Flickering (*Res.: Resolution; the Percentage Denotes the Percentage of Volunteers Perceiving Flickering)*

(a) Under indirect viewing			
Res.	L1	L2	L3
0.04	0%	0%	0%
0.05	0%	15%	20%
0.06	30%	50%	90%
0.07	100%	100%	100%
0.08	100%	100%	100%

(b) Under direct viewing			
Res.	L1	L2	L3
0.003	0%	0%	0%
0.004	0%	0%	15%
0.005	5%	30%	50%
0.006	40%	75%	100%
0.007	100%	100%	100%

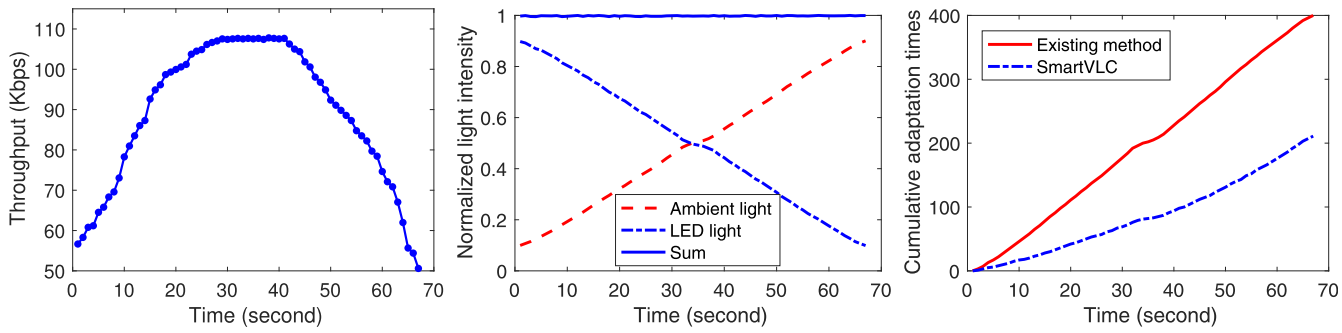
to LED's flickering under direct viewing. The resolution of dimming light level in perception domain thus needs to be equal to or smaller than 0.003 (the maximum intensity is 1) for people not to observe any flickering in all scenarios. *In the following experiments, we set τ_p=0.003 (refer to Section 5) when adapting the LED's brightness to ambient light changes.*

7.4 Variable Lighting Scenario

We continue to evaluate SmartVLC's performance in a dynamic scenario where we change the ambient light continuously by pulling up the window blind at a constant speed.

Dynamic Throughput. In this scenario, we place the transmitter and the receiver at a distance of 3 m. We change the intensity of the ambient light by pulling up the window blind from bottom to top at a constant low speed (the process takes 67 seconds). We carry out the experiments on a sunny morning. Based on the position of the window blind, the ambient light measured at the receiver is with the range of 630–9580 lux.[9] This is nearly identical with the lux range of the sunny morning scenario shown in Fig. 23. The system reports the average throughput every second, as plotted in

9. This range mimics the ambient light range in a day. Both the upper and lower boundary are higher than the other VLC works [31], [32] which consider ambient light. The high ambient light level can decrease SNR and aid us to verify the communication robustness of the system. Based on our historical records and the orientation of the window in the lab, this is the most extensive range that we can capture in a month.



(a) Throughput under AMPPM (b) Recording of the light intensity (c) Cumulative adaptation times

Fig. 25. Experimental results in the dynamic scenarios.

Fig. 25a. The shape of the throughput is nearly symmetrical and matches the static results shown in Fig. 19 well, which implies that the proposed AMPPM scheme can optimize the throughput performance under a large range of ambient light variations automatically, outperforming the state-of-the-art solutions. Therefore, SmartVLC is very useful for those scenarios where the weather is erratic during the daytime. For example, in the Netherlands, the weather condition varies quite fast most of the time, because of heavy and moving clouds. In these areas, SmartVLC works well due to its quick adaptation to ambient light changes.

During the experiment, we also record the instantaneous intensities of the ambient and LED light shown in Fig. 25b. We can observe that SmartVLC can adapt the LED's light intensity in a fine-grained manner when ambient light changes. The sum of the LED and ambient light is maintained nearly constant. We also do not perceive any flickering in the whole process.

Another interesting observation in Fig. 25a is that the throughput does not change as smoothly as the static results presented in Fig. 19. This is because the ambient light does not change *perfectly* linearly with the blind's position in real life, resulting in a non-smooth change of the achieved throughput. Furthermore, the achieved throughput in Fig. 25a is not *perfectly* symmetrical and the throughputs on the right side of the figure are slightly lower. This is because when the blind is pulled to near the top, the system suffers higher interference from ambient light.

7.5 Number of Adaptation Adjustment

It is critical to minimize the number of intensity adjustments to reduce the computational overhead on determining the optimal pattern of the super-symbol. We evaluate the number of adaptation adjustments in SmartVLC (*denoted as "SmartVLC"*) and compare it with the method that does not consider the non-linear reaction of the human's eyes to light intensity (*denoted as "existing method"*). Fig. 25c shows the cumulative number of adjustments in brightness adaptation in the above experiment. Although we use a *fix* adaptation step τ_p=0.003 in the *perception domain* , the τ (refer to Fig. 12b) is actually a *variable* in the *measurement domain* , i.e., a larger τ is chosen when the LED's light intensity is high, and vice versa. With our proposed adaptation method in SmartVLC, we successfully reduce the number of adaptation adjustments by 50 percent, which helps significantly in reducing the computational load on the low-cost hardware.

8 Discussion

VLC Modulation Schemes with Dimming Support. To support smart lighting in VLC system, several modulation schemes have been proposed. They can be categorized as *compensation-based*, which favors fine-grained dimming levels [14], [15], [16]; and *compensation-free*, which favors higher throughput.[17], [18], [19] In Section 2, we have introduced the *compensation-based* OOK-CT and *compensation-free* MPPM schemes that are tightly related to the proposed AMPPM method in SmartVLC. Besides, other schemes are also proposed by researchers, such as Variable Pulse Position Modulation (VPPM), [14] Single Edge Position Modulation (SEPM), [33] Overlapping Pulse Position Modulation (OPPM),[31] etc. Compared to AMPPM, none of them can provide fine-grained dimming levels and optimize the throughput concurrently.

VLC Platforms. Previous VLC platforms can be categorized as *1) low-end* with commodity hardware and *2) high-end* with software defined radios. The low-end testbeds [13], [27], [34], [35] employ low-cost hardware, e.g., BeagleBone, [36] Raspberry Pi, [37] or Arduino. [38] Their performance is limited by the computational capabilities of the hardware (processor, sampling rate of ADC, etc.). They are more suitable for applications that do not require high throughput. The high-end testbeds are suitable for throughput intensive applications where advanced multi-carrier pulse based modulation schemes can be adopted, such as ACO-OFDM (Asymmetrically Clipped Optical Orthogonal Frequency Division Multiplexing). [39], [40] The representative high-end platform includes Mango WARP [41] and NI USRP. [42] Our proposed SmartVLC is implemented on the low-end testbed BeagleBone Black. Nevertheless, we believe that our design principles in SmartVLC, such as the support of fine-grained dimming levels with dithering, the fast adaptation to different ambient light, etc., can benefit future designs of new multi-carrier pulse based modulation schemes that jointly consider smart lighting and VLC.

VLC Applications. VLC has been exploited in many applications including wireless communication, [11], [43] indoor localization, [9], [10], [44] human sensing, [45], [46] screen-to-camera communication, [47], [48] vehicle-to-vehicle communication, [49], [50], [51] and so on. The above mentioned systems can employ SmartVLC to save power consumption and improve user experiences. SmartVLC is orthogonal to Darklight [31] and can be combined with it for better performance. When illumination is required, SmartVLC can be applied and when illumination is not required (e.g., at night), DarkLight can then be applied instead.

9 Conclusion

The co-design of smart lighting and visible light communication is an important topic but still in its immature stage. We propose SmartVLC to achieve fine-grained dimming levels so flickering will not occur, and at the same time, throughput is maximized at each dimming level. SmartVLC has been implemented with cheap off-the-shelf devices, and extensive experiments demonstrate the superior performance over the state-of-the-art solutions. SmartVLC is particularly suitable for the places with erratic weather during the daytime, e.g., London, Amsterdam, etc. SmartVLC's design principles could intrigue more system-level research efforts on co-designing smart lighting and communication. The implementation of SmartVLC using the PRUs benefits other VLC solutions, for example, it has been partially merged into OpenVLC1.2 (an open-source project for low-end VLC). [27] The performance of SmartVLC, in terms of its throughput and distance, can be further improved if it is implemented with better hardware such as USRP.

Acknowledgments

This work is supported in part by the Research Foundation Flanders (FWO) postdoctoral fellowship under grant number 12Y0919N and the Netherlands Organisation for Scientific Research (NWO) Top Grant 612.001.854. This work was partially performed when H. Wu and Q. Wang were affiliated with the Delft University of Technology, the Netherlands. This work was partially presented at the ACM International Conference on Emerging Networking EXperiments and Technologies (CoNEXT'17) [1].

References

[1] H. Wu, Q. Wang, J. Xiong, and M. Zuniga, "SmartVLC: When smart lighting meets VLC," in *Proc. ACM Int. Conf. Emerging Netw. Experiments Technol.*, 2017, pp. 212–223.
[2] 2017. [Online]. Available: http://www.earth-policy.org/data_highlights/2011/highlights15
[3] Research and markets. Smart Lighting - Global Strategic Business Report. 2015. [Online]. Available: https://www.researchandmarkets.com/reports/3502815
[4] C. DiLouie, *Advanced Lighting Controls: Energy Savings, Productivity, Technology and Applications*. Lilburn, GA, USA: Fairmont Press, Inc., 2006.
[5] ISO 8995–1:2002 - Lighting of work places – Part 1: Indoor. [Online]. Available: https://www.iso.org/standard/28857.html
[6] S. Dimitrov and H. Haas, *Principles of LED Light Communications: Towards Networked Li-Fi*. Cambridge, U.K.: Cambridge Univ. Press, 2015.
[7] T. Komine and M. Nakagawa, "Fundamental analysis for visible-light communication system using led lights," *IEEE Trans. Consum. Electron.*, vol. 50, no. 1, pp. 100–107, Feb. 2004.
[8] D. Tsonev, S. Videv, and H. Haas, "Light fidelity (Li-Fi): Towards all-optical networking," in *Proc. SPIE*, 2013, Art. no. 900702.
[9] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A visible light based positioning system," in *Proc. USENIX Conf. Netw. Syst. Des. Implementation*, 2014, pp. 331–343.
[10] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor positioning with mobile phones and visible light," in *Proc. ACM Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 447–458.
[11] L. Grobe, A. Paraskevopoulos, J. Hilt, D. Schulz, F. Lassak, F. Hartlieb, C. Kottke, V. Jungnickel, and K.-D. Langer, "High-speed visible light communication systems," *IEEE Commun. Mag.*, vol. 51, no. 12, pp. 60–66, Dec. 2013.
[12] H. Le Minh, D. O'Brien, G. Faulkner, L. Zeng, K. Lee, D. Jung, and Y. Oh, "High-speed visible light communications using multiple-resonant equalization," *IEEE Photon. Technol. Lett.*, vol. 20, no. 14, pp. 1243–1245, Jul. 2008.
[13] S. Schmid, G. Corbellini, S. Mangold, and T. R. Gross, "LED-to-LED visible light communication networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2013, pp. 1–10.
[14] *IEEE Standard for Local and Metropolitan Area Networks–Part 15.7: Short-Range Wireless Optical Communication Using Visible Light*, IEEE Standard 802.15.7–2011, pp. 1–309, Sep. 2011.
[15] J. Gancarz, H. Elgala, and T. Little, "Impact of lighting requirements on VLC systems," *IEEE Commun. Mag.*, vol. 51, no. 12, pp. 34–41, Dec. 2013.
[16] S. Rajagopal, R. D. Roberts, and S.-K. Lim, "IEEE 802.15.7 visible light communication: Modulation schemes and dimming support," *IEEE Commun. Mag.*, vol. 50, no. 3, pp. 72–82, Mar. 2012.
[17] B. Bai, Z. Xu, and Y. Fan, "Joint LED dimming and high capacity visible light communication by overlapping PPM," in *Proc. IEEE Conf. Wireless Opt. Commun.*, 2010, pp. 1–5.
[18] A. B. Siddique and M. Tahir, "Joint rate-brightness control using variable rate MPPM for LED based visible light communication systems," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4604–4611, Sep. 2013.
[19] K. Lee and H. Park, "Modulations for visible light communications with dimming control," *IEEE Photon. Technol. Lett.*, vol. 23, no. 16, pp. 1136–1138, Aug. 2011.
[20] F. Zafar, D. Karunatilaka, and R. Parthiban, "Dimming schemes for visible light communication: The state of research," *IEEE Wireless Commun.*, vol. 22, no. 2, pp. 29–35, Apr. 2015.
[21] R. Lee, K. Yun, J.-H. Yoo, S.-Y. Jung, and J. K. Kwon, "Performance analysis of M-ary PPM in dimmable visible light communications," in *Proc. IEEE Conf. Ubiquitous Future Netw.*, 2013, pp. 380–383.
[22] H. Sugiyama and K. Nosu, "MPPM: A method for improving the band-utilization efficiency in optical PPM," *J. Lightw. Technol.*, vol. 7, no. 3, pp. 465–472, 1989.
[23] K. Sato, T. Ohtsuki, I. Sasase, and S. Mori, "Performance analysis of (m, 2) MPPM with imperfect slot synchronization," in *Proc. IEEE Pacific Rim Conf. Commun. Comput. Signal Process.*, 1993, pp. 765–768.
[24] H. Park and J. R. Barry, "Trellis-coded multiple-pulse-position modulation for wireless infrared communications," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 643–651, Apr. 2004.
[25] M. S. Rea, *The IESNA Lighting Handbook: Reference & Application*. New York, NY, USA: Illuminating Eng. Soc. North America, 2000.
[26] Q. Wang, D. Giustiniano, and D. Puccinelli, "An open-source research platform for embedded visible light networking," *IEEE Wireless Commun.*, vol. 22, no. 2, pp. 94–100, Apr. 2015.
[27] A. Galisteo, D. Juara, Q. Wang, and D. Giustiniano, "OpenVLC1.2: Achieving higher throughput in low-end visible light communication networks," in *Proc. IEEE/IFIP Wireless On-Demand Netw. Syst. Serv. Conf.*, 2018, pp. 1–4.
[28] M. S. Islim, R. X. Ferreira, X. He, E. Xie, S. Videv, S. Viola, S. Watson, N. Bamiedakis, R. V. Penty, I. H. White, et al., "Towards 10 Gb/s orthogonal frequency division multiplexing-based visible light communication using a GaN violet micro-LED," *Photon. Res.*, vol. 5, no. 2, pp. A35–A43, 2017.
[29] S.-M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits*. New York, NY, USA: McGraw-Hill Education, 2003.
[30] E. Rubiola, *Phase Noise and Frequency Stability in Oscillators*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[31] Z. Tian, K. Wright, and X. Zhou, "The darklight rises: Visible light communication in the dark," in *Proc. ACM Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 2–15.
[32] F. Tan, D. Caicedo, A. Pandharipande, and M. Zuniga, "Sensor-driven, human-in-the-loop lighting control," *Lighting Res. Technol.*, vol. 50, no. 5, pp. 660–680, 2018.
[33] N. Stevens and L. D. Strycker, "Single edge position modulation as a dimming technique for visible light communications," *J. Lightw. Technol.*, vol. 34, no. 23, pp. 5554–5560, 2016.
[34] L. Klaver and M. Zuniga, "Shine: A step towards distributed multi-hop visible light communication," in *Proc. IEEE Conf. Mobile Ad Hoc Sensor Syst.*, 2015, pp. 235–243.
[35] Q. Wang, D. Giustiniano, and D. Puccinelli, "OpenVLC: Software-defined visible light embedded networks," in *Proc. 1st ACM MobiCom Workshop Visible Light Commun. Syst.*, 2014, pp. 15–20.
[36] BeagleBone Black, 2017. [Online]. Available: http://beagleboard.org/Products/BeagleBone+Black
[37] Raspberry pi, 2017. [Online]. Available: https://www.raspberrypi.org/
[38] Arduino, 2017. [Online]. Available: https://www.arduino.cc/

[39] X. Li, J. Vucic, V. Jungnickel, and J. Armstrong, "On the capacity of intensity-modulated direct-detection systems and the information rate of ACO-OFDM for indoor optical wireless applications," *IEEE Trans. Commun.*, vol. 60, no. 3, pp. 799–809, Mar. 2012.
[40] S. Dimitrov and H. Haas, "On the clipping noise in an ACO-OFDM optical wireless communication system," in *Proc. IEEE Global Telecommun. Conf.*, 2010, pp. 1–5.
[41] WARP project, 2017. [Online]. Available: http://warpproject.org
[42] Universal software radio peripheral, 2017. [Online]. Available: https://www.ettus.com/
[43] M. Rahaim, A. Miravakili, T. Borogovac, T. Little, and V. Joyner, "Demonstration of a software defined visible light communication system," in *Demo and Exhibits of Proc. ACM Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 1–4.
[44] B. Xie, G. Tan, and T. He, "SpinLight: A high accuracy and robust light positioning system for indoor applications," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, 2015, pp. 211–223.
[45] T. Li, C. An, Z. Tian, A. T. Campbell, and X. Zhou, "Human sensing using visible light communication," in *Proc. ACM Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 331–344.
[46] C. Zhang, J. Tabor, J. Zhang, and X. Zhang, "Extending mobile interaction through near-field visible light sensing," in *Proc. ACM Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 345–357.
[47] T. Hao, R. Zhou, and G. Xing, "COBRA: Color barcode streaming for smartphone systems," in *Proc. ACM Int. Conf. Mobile Syst. Appl. Services*, 2012, pp. 85–98.
[48] S. D. Perli, N. Ahmed, and D. Katabi, "PixNet: Interference-free wireless links using LCD-camera pairs," in *Proc. ACM Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 137–148.
[49] L.-C. Wu and H.-M. Tsai, "Modeling vehicle-to-vehicle visible light communication link duration with empirical data," in *IEEE Globecom Workshops*, 2013, pp. 1103–1109.
[50] S.-H. You, S.-H. Chang, H.-M. Lin, and H.-M. Tsai, "Visible light communications for scooter safety," in *Proc. ACM Annu. Int. Conf. Mobile Syst. Appl. Services*, 2013, pp. 509–510.
[51] S.-H. Yu, O. Shih, H.-M. Tsai, N. Wisitpongphan, and R. D. Roberts, "Smart automotive lighting for vehicle safety," *IEEE Commun. Mag.*, vol. 51, no. 12, pp. 50–59, Dec. 2013.

Hongjia Wu received the MS degree from the Delft University of Technology, the Netherlands. He is currently working toward the PhD degree in the Simula Research Laboratory, Norway. His research interests include visible light communication and multipath transport protocol design.

Qing Wang received the BE and MS degrees from the University of Electronic Science and Technology of China, in 2008 and 2011, respectively, and the MS and PhD degrees from the University Carlos III of Madrid and the IMDEA Networks Institute, in 2012 and 2016, respectively. He is currently the FWO postdoc associate with KU Leuven, Belgium. Prior to that, he was a postdoc with the Delft University of Technology, the Netherlands. His research interests include visible light communication, Internet of things, and device-to-device communication. He is a member of the IEEE.

Jie Xiong received the BEng degree from Nanyang Technological University, Singapore, in 2005, the MSc degree from Duke University, Durham, North Carolina, in 2009, and the PhD degree in computer science from University College London, London, United Kingdom, in 2015. He is an assistant professor with the College of Information and Computer Sciences, University of Massachusetts Amherst. His research interests include building practical wireless and mobile systems that bridge the gaps between theory and reality. His recent work appears at MobiCom, NSDI, CoNEXT, Ubicomp, and INFOCOM. He was the recipient of the prestigious Google European Doctoral Fellowship in Wireless Networking for his doctoral studies. His PhD thesis was the 2016 British Computer Society Distinguished Dissertation Award runner-up. He is a member of the IEEE.

Marco Zuniga received the BSc degree in electronics engineering from the Pontificia Universidad Catolica del Peru, and the MSc and PhD degrees in electrical engineering from the University of Southern California. He is an assistant professor with the Department of Computer Science, Delft University of Technology, the Netherlands. Before this appointment, he was a member of the research staff with Xerox Research Labs; an IRCSET fellow with the National University of Ireland, Galway; and a senior researcher with the University of Duisburg-Essen, Germany. His research interests include on the areas of cyber-physical systems and mobile computing. He is a member of the IEEE.