SmartVLC

When Smart Lighting Meets VLC

Wu, Hongjia ; Wang, Qing; Xiong, Jie; Zuñiga Zamalloa, M.A.

DOI
10.1145/3143361.3149824

Publication date
2017

Published in
CoNEXT'17 Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies

SmartVLC: When Smart Lighting Meets VLC

Hongjia Wu
Delft University of Technology
the Netherlands
h.wu-7@student.tudelft.nl

Qing Wang*
Delft University of Technology & KU Leuven
the Netherlands & Belgium
qing.wang@imdea.org

Jie Xiong
Singapore Management University
Singapore
jxiong@smu.edu.sg

Marco Zuniga
Delft University of Technology
the Netherlands
m.a.zunigazamalloa@tudelft.nl

ABSTRACT

Visible Light Communication (VLC) based on LEDs has been a hot topic investigated for over a decade. However, most of the research efforts in this area assume the intensity of the light emitted from LEDs is constant. This is *not* true any more when *Smart Lighting* is introduced to VLC in recent years, which requires the LEDs to adapt their brightness according to the intensity of the natural ambient light. Smart lighting saves power consumption and improves user comfort. However, intensity adaptation severely affects the throughput performance of the 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 new adaptive multiple pulse position modulation scheme is proposed to support fine-grained dimming levels to avoid flickering and at the same time, maximize the throughput under each dimming level. SmartVLC is implemented on low-cost commodity hardware and several real-life challenges in both hardware and software are addressed to make SmartVLC a robust realtime system. Comprehensive experiments are carried out to evaluate the performance of SmartVLC under multifaceted scenarios. The 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% on average, respectively, without bringing any flickering to users.

CCS CONCEPTS

• **Networks → Cyber-physical networks**;

KEYWORDS

Smart lighting, Visible light communication, flickering-free, AMPPM, System design, Implementation, Evaluation

*Corresponding author.

CoNEXT '17, December 12–15, 2017, Incheon, Republic of Korea

ACM ISBN 978-1-4503-5422-6/17/12…$15.00
https://doi.org/10.1145/3143361.3149824

ACM Reference Format:
Hongjia Wu, Qing Wang, Jie Xiong, and Marco Zuniga. 2017. SmartVLC: When Smart Lighting Meets VLC. In *CoNEXT '17: The 13th International Conference on emerging Networking EXperiments and Technologies, December 12–15, 2017, Incheon, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3143361.3149824

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 [15]. These systems adjust the illuminance of artificial lights (usually, LED lights) based on the contribution of ambient natural sunlight in our environments [9]. Smart lighting systems are expected to maintain a constant illumination within the area of interest, improving energy-saving and 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 to maintain a constant illumination when the natural light changes and also to make sure the intensity of artificial lights is changed in a graceful manner, without causing flickering to users.

Smart lighting systems are good 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 attentions from both industry and academia. In VLC, data is transmitted by modulating optical sources such as standard LED luminaries [11, 18]. 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 data rates 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) [36] and accurate indoor localization [19, 23]. *Most VLC systems, however, assume that the intensity of the light emitted from LEDs is constant [13, 20, 32], but this assumption is not true 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 the purpose of data communication;

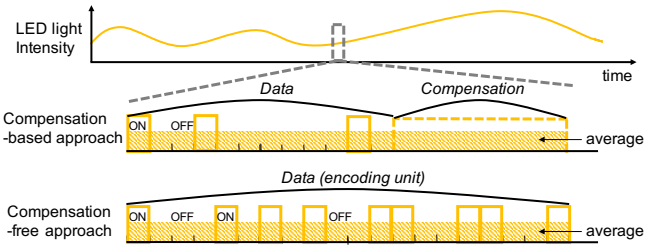


Figure 1: Illustration of state-of-the-art approaches for joint designed smarting lighting and VLC systems

with visible light, the carrier is modulated to achieve the dual goal of providing controlled illumination for users and high data rates 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 data 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 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 [1, 12, 29], and (ii) *compensation-free*, which favor high throughput [8, 21, 33]. 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 exclusively a function of the encoding scheme. 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 set 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-design Smart lighting (illumination) and VLC (communication) for visible light networks. The concrete contributions of our work are three-fold:

- *Method [Sections 3 and 4].* 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 multiplex individual symbols to achieve a dimming granularity comparable to *compensation-based* methods and data rates higher than both *compensation-free* and *compensation-based* methods. We efficiently utilize the property of human eye's non-linear reaction to light intensities to reduce the number of brightness adjustments which is critical for hardware's lifespan and at the same time, guarantee flickering-free 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 5].* To validate the generality of our approach, we implement SmartVLC with cheap off-the-shelf commodity hardware (BeagleBone Black), instead of using more advanced and costly software-defined radio platforms such as USRP [6] and WARP [7]. 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 packet 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 6].* We present a multifaceted evaluation. First, we compare the performance of SmartVLC with the state-of-the-art approaches. Our results show that our throughput is up to 170% and 30% 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.

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 dimming level control of LEDs. They are divided into two categories: *analog dimming* and *digital dimming* [12, 43]. The former increases/decreases LEDs' brightness by adjusting the forward current through LEDs which is simple to implement. However, it causes color shift [1, 43]. The latter is based on Pulse Width Modulation (PWM), where dimming is achieved by adjusting LEDs' duty cycles. Thus, it does not cause color shift. Next, we introduce

two popular digital dimming methods that are closely related to our work.

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 [1, 12, 29], as illustrated in Fig. 1 with label *'compensation-based approach'*. The *advantage* of OOK-CT 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.

MPPM (Multiple Pulse Position Modulation): In MPPM [21, 22, 33, 34], data is modulated by the positions of ONs in a symbol, as illustrated 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 higher data rate than OOK-CT. The *disadvantage* of MPPM is that the supported dimming levels are restricted.

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.

Type-I flickering: The first type of flickering is caused by slow change of ON and OFF states at the LED. If the frequency of the ON/OFF changes is high enough, people's eyes will not perceive the change and there will be no flickering.

Type-II flickering: On the other hand, even if the frequency intensity change is low, as long as the intensity change is small enough, people will also not perceive the change. 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.

3 SYSTEM OVERVIEW

The architecture overview of SmartVLC is shown in Fig. 2. 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 – 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 and LED light to be constant.
(2) The proposed AMPPM scheme then selects the best parameters to modulate the upper layer data to maximize the data rate under the current required dimming level.
(3) When transmitting the modulated data, the LED's brightness is adjusted to the required dimming level. Besides this, the header of each MAC layer frame is adjusted accordingly based on the targeted dimming level.
(4) The frames are transmitted by modulating the LED light.

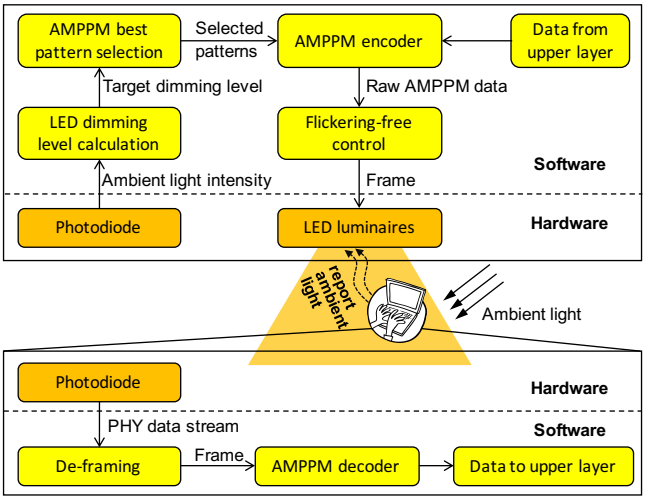


Figure 2: System architecture of SmartVLC (Top: transmitter; Bottom: receiver)

(5) The ACK frames and the real-time ambient light conditions sensed by receivers are sent from the receivers to the transmitter via off-the-shelf 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 the AMPPM (which is adaptive based on ambient light). Then the receiver extracts the data carried by the frames and passes them to upper layers.

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. 3):

- **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 $T = N \cdot t_{\text{slot}}$. The value of N depends on the modulation and is a ***variable***.
- **Dimming level**: denoted as l. It indicates the brightness of the LED. Mathematically, l can be expressed as

$$l = \frac{\text{number of state "ON" in a symbol}}{N} \quad (1)$$

  the range of l is[0,1]. For example, l=0.5 means that the brightness of the LED is 50% 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.

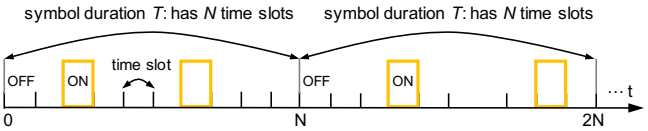


Figure 3: Illustration of the definitions of the symbol, time slot, pulse width, and so on. In this example, we have N=10, l=0.2, and the symbol pattern is $S(10, 0.2)$.

4 SYSTEM DESIGN

In this section, we present the detail 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 data rate R is given by

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

where P_{SER} is 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 higher P_{SER}.* Let's model the detection behavior of the photodiode as photon counting process following Poisson distribution [34]. The probabilities of decoding an OFF and an ON incorrectly are denoted as P_1 and P_2, respectively. To decode a whole symbol correctly in MPPM, all the ONs and OFFs need to be detected correctly. Thus, the symbol error rate P_{SER} can be written as

$$P_{\text{SER}} = 1 - (1 - P_1)^{N-K}(1 - P_2)^K \tag{3}$$

Eq. (3) implies the existence of trade-off between N (higher dimming level resolution can be achieved with larger N) and P_{SER}. Fig. 4 illustrates this relationship, where P_1 and P_2 are set to 9×10^{-5} and 8×10^{-5}, respectively, which are measured in our experiments. We can observe that a larger N leads to higher P_{SER}. We thus 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.* Therefore, we should not simply use a large N for fine-grained dimming levels.

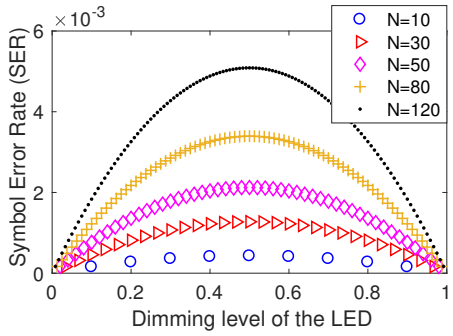


Figure 4: P_{SER} as a function of dimming level in MPPM.

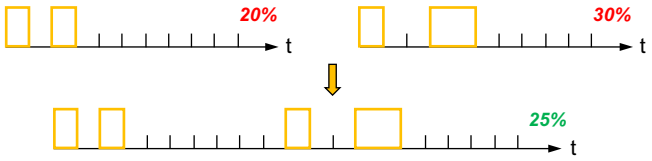
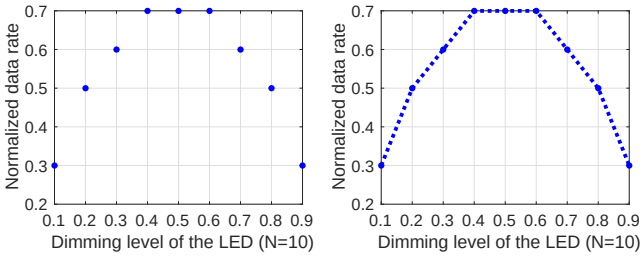


Figure 5: Using multiplexing to achieve fine-grained dimming levels (in this example, N=10)



(a) Before multiplexing (b) After multiplexing

Figure 6: Supported dimming levels before/after applying the proposed multiplexing approach

4.1.2 Increase dimming levels via multiplexing. To provide fine-grained dimming levels without sacrificing the system's performance, we propose a novel method to use *multiplexing* of the symbols. The intuition behind the multiplexing is straightforward: if we combine the symbol pattern $S(N_1, l_1)$ with symbol pattern $S(N_2, l_2)$ equally, then we successfully obtain a *super-symbol* where the new dimming level is $(N_1 l_1 + N_2 l_2)/(N_1 + N_2)$.

For example, 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 multiplexing, 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. 5. The number of slots in the super-symbol is now 20. Furthermore, this multiplexing process *does not* increase the symbol error rate because each symbol in the super-symbol will be decoded separately. Note that the resolution after multiplexing now becomes 0.05 instead of 0.1. We can have even more fine-grained resolution by multiplexing 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 multiplexing is visually presented in Fig. 6, where x-axis is the dimming level and y-axis is the normalized data rate. In Fig. 6(a), only nine discrete dimming levels are available. After applying the proposed multiplexing approach, the dimming levels become 'semi-continuous', as shown in Fig. 6(b).

4.2 Adaptive MPPM

With the multiplexing method proposed in Sec. 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. 6,

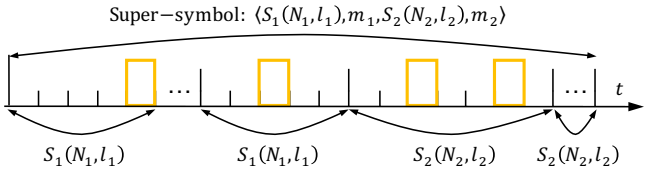


Figure 7: Illustration of a super-symbol

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). This 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 maintain 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 multiplexings bring us different throughput performance. Thus, the *key idea* in AMPPM is to *choose the best multiplexing to compose a super-symbol for maximum throughput*. The super-symbol satisfies the desired resolution of dimming level and at the same time achieves highest throughput at each dimming level.

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. 7. Let S_{super} denote a super-symbol represented by a tuple

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

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}

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

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 multiplexing symbol patterns that achieve the highest data rate, as presented later in Step 3. Luckily, we can address the problem by restricting the length of a super-frame to make sure no flickering occurs. Recall from Sec. 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

$$N_{\text{max}} = \frac{f_{\text{tx}}}{f_{\text{th}}} \tag{4}$$

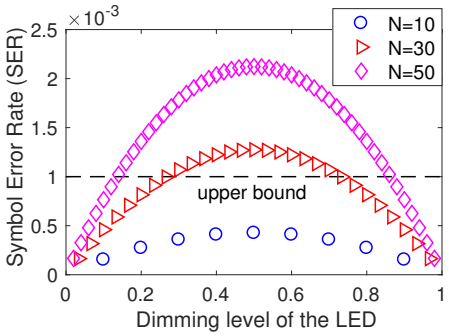


Figure 8: Available patterns: below the upper bound

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. For example, Fig. 8 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 of the P_{SER} are abandoned, e.g., $S(50, 0.3)$ and $S(30, 0.4)$ in Fig. 8. Then the data rates 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 pick up the best symbol patterns which achieve the highest data rate when multiplexed to form the super-symbol. Before multiplexing, we only have symbol patterns with discrete dimming levels. If we simply pick the pattern which generates highest data rate under each discrete dimming level, the dimming level resolution is not changed, as shown by the red-dash line in Fig. 9. We further show that even the achieved "highest data rate" is not optimum. In this work, we propose a new method to improve the data rate as well.

Through the analysis presented in Sec. 4.1, we know that we can achieve more fine-grained dimming levels through multiplexing. 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 using Fig. 9. The key idea is to find the *envelop*, namely, the blue-solid line in Fig. 9. To achieve this, the system works as follows:

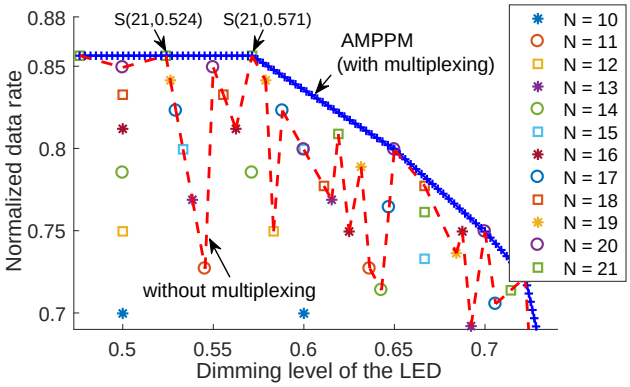


Figure 9: Best pattern selection based on slope

(1) For the available symbol patterns whose dimming level is around 0.5, find the symbol pattern that gives the highest normalized data rate.[1] In Fig. 9, 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 $S(21, 0.524)$ in Fig. 9, we identify the next symbol pattern which satisfies this requirement: **when connecting this symbol pattern with $S(21, 0.524)$, the resulted slope is the smallest.** Mapped to Fig. 9, 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. 9.
(4) For each pair of neighbouring two symbol patterns that are found through the above steps, e.g., $S(21, 0.524)$ and $S(21, 0.571)$, we multiplex 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.*

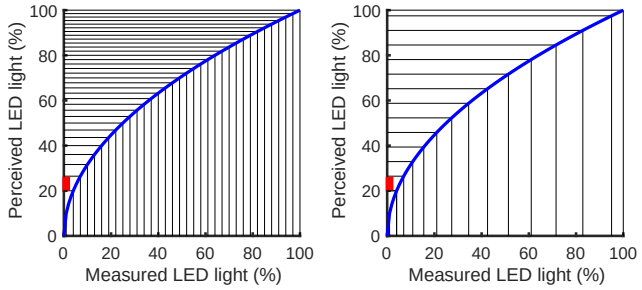
Through the above steps, AMPPM can provide fine-grained dimming levels (as demonstrated by the marker '+' on the blue-solid line in Fig. 9), at the same time can optimize the throughput under each supported dimming level.

4.3 Adaptation to various ambient light

In the above subsections, we present how to achieve fine-grained dimming levels and how to select the best pattern of a super-symbol to achieve the highest data rate 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. There are 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.

[1] 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 data rate among all the symbols with the same symbol duration.



(a) Adaptation in the measured domain (b) Adaptation in the perceived domain (*proposed in this work*)

Figure 10: Adaptation to dynamic ambient light

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, and thus expand the lifespan of hardware.

Solution to achieve **Goal 1** is straightforward. Let 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:

$$I^1_{\text{led}} + I^1_{\text{amb}} = I^2_{\text{led}} + I^2_{\text{amb}} \rightarrow \triangle I_{\text{led}} = I^2_{\text{led}} - I^1_{\text{led}} = I^1_{\text{amb}} - I^2_{\text{amb}}. \quad (5)$$

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 the τ is a *constant* and its maximum value is constrained by the second type of flickering. This adaptation is illustrated in Fig. 10(a).

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 [30]: $I_{\text{p}} = 100 \times \sqrt{I_{\text{m}}/100}$, which is also shown by the blue line in Fig. 10(a) and (b).

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

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

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

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

$$\triangle I_{\text{led-p}} = I^2_{\text{led-p}} - I^1_{\text{led-p}} \quad (6)$$

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 = IP;   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: 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
```

(3) Increase $I^1_{\text{led-p}}$ *gradually* by a step of τ_p, until it reaches to $I^2_{\text{led-p}}$. 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. 10(b).

4.4 Encoding/decoding

To transmit the data received from upper layers, the transmitter must encode the original data into codewords. To do this, classical methods based on pulse position can be categorized as two main groups: tabulation [31] and constellation [26]. The former records the mapping between 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.

Table 1: The frame format in SmartVLC

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

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 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 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.

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 subsection.

4.5 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. These details are used by the receiver to decode the corresponding frame.

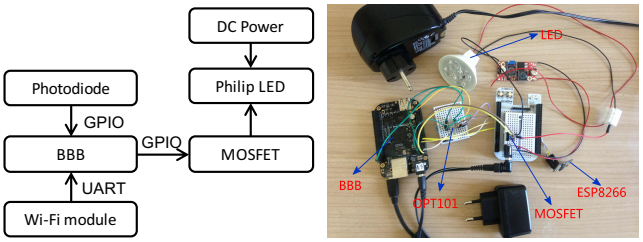
The *Compensation* and *Sync* fields are used to avoid intra-frame flickering. To align the brightness of the frame header with that of the payload, *compensation time* must be appended to the frame header, as shown in Table 1. The compensation can be consecutive ONs or OFFs, depending on the brightness 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 *CRC* (Cyclic Redundancy Check) are placed at the end of each frame.

5 IMPLEMENTATION

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

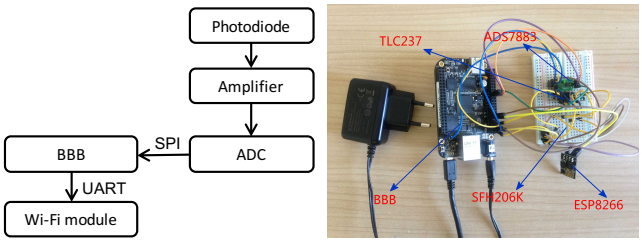
5.1 Hardware

Transmitter. The block diagram of the transmitter is given in Fig. 11(a). There are mainly four components: the BeagleBlack board (BBB) which costs around $60, transistor (ON MOSFET 20N06L), LED (Philips 4.7W), and the photodiode (TI OPT101). The GPIO of



(a) Block diagram (b) Prototype

Figure 11: Implementation of the transmitter



(a) Block diagram (b) Prototype

Figure 12: Implementation of the receiver

BBB triggers the transistor to modulate the LED light. The LED is powered by a 24V DC voltage. The LED is also disassembled by removing the AC-DC converter that can slow down the transition speed between ON and OFF states. Sensing the ambient light at the transmitter is done by sampling the photodiode driven directly by the on-board 3.3V DC voltage from the BBB's GPIO. Uplink from the receiver to the transmitter is implemented via the Farnell ESP8266 Wi-Fi module to transmit ACKs (to acknowledge the received frames)[2] and the intensity information of ambient light sensed by the receiver. The prototype of the transmitter is shown in Fig. 11(b).

Receiver. Its block diagram is shown in Fig. 12(a). There are also mainly four components: BBB, ADC (TI ADS7883), amplifier (TLC237), and photodiode (OSRAM 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 the photodiode SFH206K at the receiver instead of the OPT101 (as used at the transmitter) 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. 12(b).

5.2 Software

The BBB used in SmartVLC is very cheap ($60), much cheaper than high-end platforms such as the widely used USRP and WARP ($5000). The latter two expensive software-defined radio platforms are FPGA-based and support a much higher ADC&DAC sampling rate. Since we select the low-cost BBB in our SmartVLC system, the *challenge* in the software implementation is two-fold: (1) how to modulate 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 low-cost BBB platform.

The BBB runs the Linux OS. To modulate the LED and perform the sampling, Linux offers an easy way to write/read the GPIO pins through the files linked to GPIOs. Besides, the GPIOs are assigned with physical memory addresses, which can be used to control GPIOs at a much faster speed (around 10x in our test) than writing/reading GPIO associated files. These methods are not fast enough to support the required high enough sampling rate, mainly because of the non-realtime Linux OS. This issue can be mitigated via patching the Linux kernel with Xenomai that supports real-time operations, which can achieve a sampling rate of up to 50 KHz [38]. However, this speed is still far away from our target because the ADC we use can support a sampling rate of 3.0 MHz.

In this work, we exploit the *existing PRUs (micro-controllers) of the BBB* to address the above challenge, without introducing new cost on additional hardware, such as using an additional FPGA. The implementation complexity with FPGAs is also higher than that with PRUs. 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 the requirement of our system with off-the-shelf low cost hardware.

6 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.

6.1 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[3], i.e., four times of f_{tx}.

[2]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.

[3]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 [16]. 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.

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

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

Figure 13: Control ambient light using window blind

Maximal super-symbol duration. According to the IEEE 802.15.7 standard [29], 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. (4), we have $N_{\max} = f_{\text{tx}}/f_{\text{th}} = 125000/250 = 500$, which means that we can have up to 500 consecutive slots for a super-symbol (Sec. 4.2) without causing flickering.

Detecting errors of `ON`s and `OFF`s. 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 the P_1 and P_2, as presented in Eq. (3). Based on our empirical experiment of an extreme case (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 put up to the top), P_1 and P_2 are measured to be 0.00009 and 0.00008. 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.

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. 13. We fix the window blind to a position to provide a static ambient light condition, as shown in Fig. 13(a); 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. 13(b).

Deriving the super-symbols. The transmitter adopts the three steps presented in Sec. 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.

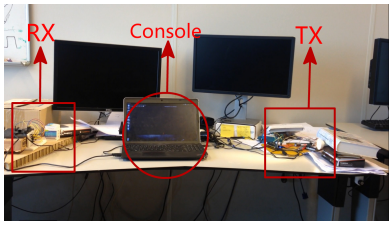


Figure 14: Placement of the transmitter and receiver

6.2 Static 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[4] 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. 14.

Comparison with MPPM and OOK-CT[5]: 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, 17 discrete dimming levels are considered, ranging from 0.1 to 0.9. The receiver is placed at a distance of 3m from the receiver. For MPPM, if the value of N is too large, the symbol error rate will be higher than the upper bound. To make sure the symbol error rate is below the upper bound, an appropriate value of N is selected as 20. For OOK-CT, the brightness is adjusted by compensating the data in the payload.

The evaluation results are shown in Fig. 15. 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 Kbps, 21.7 Kbps, and 44.3 Kbps, respectively. AMPPM improves the performance of OOK-CT by up to 170% when l is either low or high, and on average by 40%. For MPPM, its throughput performance can be improved by up to 30% (when l=0.9) with AMPPM, and on average by 12%. This is because AMPPM always selects the best symbol pattern under each dimming level. Moreover, recall from Fig. 9, 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)

[4]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.

[5]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.

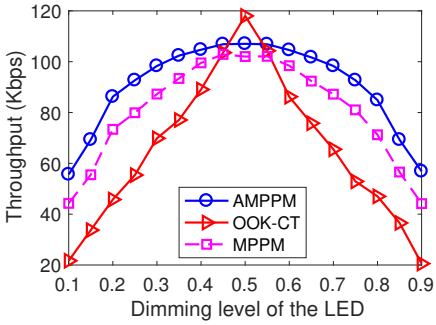


Figure 15: Comparison with OOK-CT and MPPM

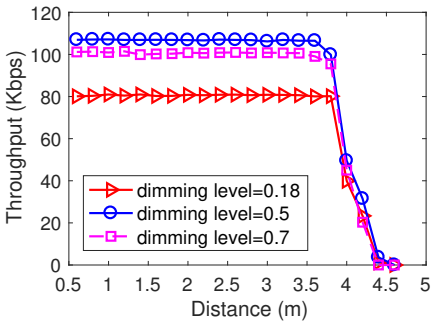


Figure 16: Throughput vs. communication distance

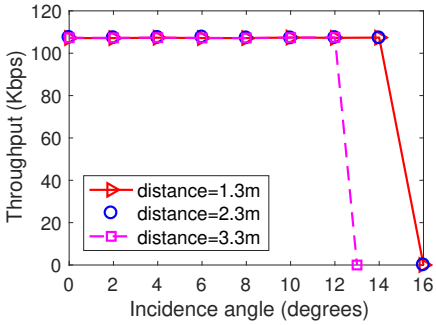


Figure 17: Throughput vs. incidence angle

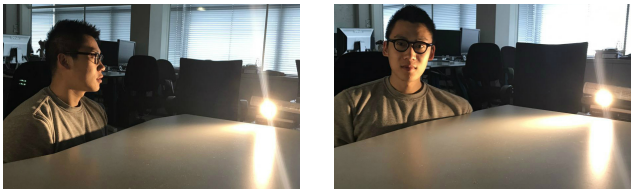
AMPPM introduces a small overhead on deriving the optimal symbol patterns to compose the super-symbol (by performing the Steps 1 to 3 as described in Sec. 4.1), as well as the four additional bytes (field "Pattern") in the frame header to describe the patterns. However, OOK-CT loses the advantage quickly 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. 16, 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 ceiling is usually around 2.5–3m in a typical office, the communication distance of 3.6m 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.

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 (1.3m, 2.3m, and 3.3m), and the results are shown in Fig. 17. We can observe that SmartVLC can almost maintain its performance with the LED's Field of View (FoV). Another observation is that longer distance has 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.

6.3 Dynamic scenario

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



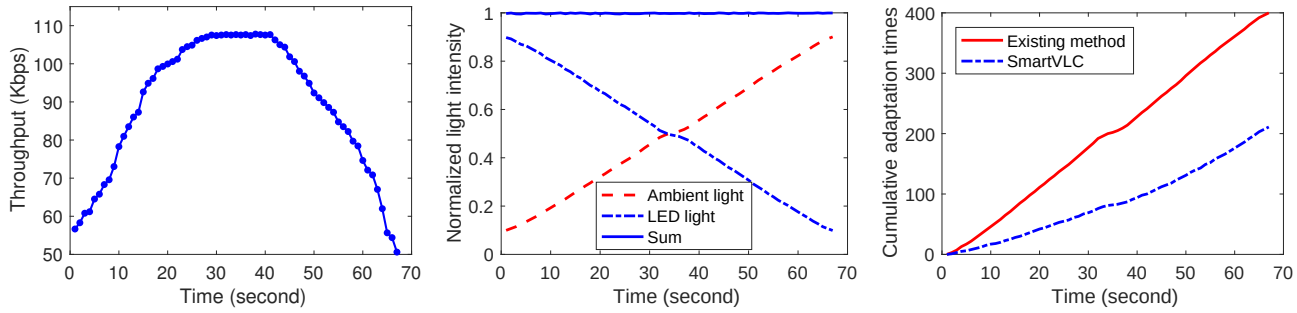
(a) Direct viewing (b) Indirect viewing

Figure 18: Users' perception of light changes

Minimum dimming level resolution for non-flickering adaptation: 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. 18(a); *(ii) indirect viewing*, where they judge the flickering/non-flickering based on the reflection of LED light, as illustrated in Fig. 18(b). 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 to LED's flickering under the direct viewing. The resolution of dimming light level 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 the scenarios. *In the following experiments, we set τ_p=0.003 (refer to Sec. 4.3) when adapting the LED's brightness to ambient light changes.*

Dynamic throughput: In this scenario, we place the transmitter and the receiver at a distance of 3m. We change the intensity of the ambient light by pulling up the window blind from bottom to top at a low constant speed (the process takes 67 seconds). The system reports the average throughput every second, as plotted in Fig. 19(a). The shape of the throughput is nearly symmetrical and matches the static results shown in Fig. 15 well, which implies that the proposed AMPPM scheme can optimize the throughput



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

Figure 19: Experimental results in the dynamic scenarios

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%

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 when the weather changes fast during the daytime. For example, in the Netherlands, the weather changes super fast and for most of the time, there are 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. 19(b). 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. 19(a) is that the throughput does not change as smoothly as the static results presented in Fig. 15. This is because the ambient light does not changed *perfectly* linearly with the blind's position in real life, resulting in non-smooth change of the achieved throughput. Furthermore, the achieved throughput in Fig. 19(a) 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.

6.4 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 and increase the lifespan of the hardware. 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. 19(c) 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. 10(b)) 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%, which helps significantly in reducing the computational load on the low-cost hardware, and increasing the hardware lifespan at the same time.

7 RELATED WORK

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 [1, 12, 29]; and *compensation-free*, which favors higher throughput [8, 21, 33]. In Sec. 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) [1], Overlapping Pulse Position Modulation (OPPM) [35], 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 [17, 32, 37] employ low-cost hardware, e.g., BeagleBone [4], Raspberry Pi [5], or Arduino [3]. 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 data rates. The high-end testbeds are suitable for data-rate intensive applications where advanced modulation schemes can be adopted, such as ACO-OFDM (Asymmetrically Clipped Optical Orthogonal Frequency Division Multiplexing) [10, 25]. The representative high-end platform includes Mango WARP [7] and NI USRP [6]. The disadvantage is that the high cost of hardware, usually in the order of thousands of dollars, limits their real-life applications.

VLC Applications. VLC has been exploited in many applications including wireless communication [13, 28], indoor localization [19, 23, 40], human sensing [24, 44], screen-to-camera communication [14, 27], vehicle-to-vehicle communication [39, 41, 42], and so on. The above mentioned systems can employ SmartVLC to save power consumption and improve user experiences. SmartVLC is orthogonal to Darklight [35] 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.

8 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'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 low-cost VLC solutions, for example, it has been merged into OpenVLC1.2 (an open-source project for low-end VLC).

ACKNOWLEDGEMENT

Our sincere thanks to the reviewers and our shepherd Wenjun Hu for all of their great comments and constructive feedbacks.

REFERENCES

[1] 2011. IEEE Standard for Local and Metropolitan Area Networks–Part 15.7: Short-Range Wireless Optical Communication Using Visible Light. *IEEE Std 802.15.7-2011* (September 2011), 1–309.
[2] 2017. *http://www.earth-policy.org/data_highlights/2011/highlights15* (2017).
[3] 2017. Arduino. *https://www.arduino.cc/* (2017).
[4] 2017. BeagleBone Black. *http://beagleboard.org/Products/BeagleBone+Black* (2017).
[5] 2017. Raspberry Pi. *https://www.raspberrypi.org/* (2017).
[6] 2017. Universal Software Radio Peripheral. *https://www.ettus.com/* (2017).
[7] 2017. WARP Project. *http://warpproject.org* (2017).
[8] Bo Bai, Zhengyuan Xu, and Yangyu Fan. 2010. Joint LED dimming and high capacity visible light communication by overlapping PPM. In *Wireless and Optical Communications Conference (WOCC), 2010 19th Annual*. IEEE, 1–5.
[9] Craig DiLouie. 2006. *Advanced lighting controls: energy savings, productivity, technology and applications*. The Fairmont Press, Inc.
[10] Svilen Dimitrov and Harald Haas. 2010. On the clipping noise in an ACO-OFDM optical wireless communication system. In *GLOBECOM 2010*. 1–5.
[11] Svilen Dimitrov and Harald Haas. 2015. *Principles of LED Light Communications: Towards Networked Li-Fi*. Cambridge University Press.
[12] John Gancarz, Hany Elgala, and Thomas Little. 2013. Impact of lighting requirements on VLC systems. *IEEE Communications Magazine* (2013), 34–41.
[13] Liane Grobe, Anagnostis Paraskevopoulos, Jonas Hilt, Dominic Schulz, Friedrich Lassak, Florian Hartlieb, Christoph Kottke, Volker Jungnickel, and Klaus-Dieter Langer. 2013. High-speed visible light communication systems. *IEEE Communications Magazine* 51, 12 (2013), 60–66.
[14] Tian Hao, Ruogu Zhou, and Guoliang Xing. 2012. COBRA: color barcode streaming for smartphone systems. In *Proceedings of the ACM MobiSys*. ACM, 85–98.
[15] Global Industry Analysts Inc. 2015. Smart Lighting: A Global Strategic Business Report. (2015).
[16] Mohamed Sufyan Islim, Ricardo X Ferreira, Xiangyu He, Enyuan Xie, Stefan Videv, Shaun Viola, Scott Watson, Nikolaos Bamiedakis, Richard V Penty, Ian H White, et al. 2017. Towards 10 Gb/s orthogonal frequency division multiplexing-based visible light communication using a GaN violet micro-LED. *Photonics Research* 5, 2 (2017), A35–A43.
[17] Lennart Klaver and Marco Zuniga. 2015. Shine: A step towards distributed multi-hop visible light communication. In *Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on*. IEEE, 235–243.
[18] Toshihiko Komine and Masao Nakagawa. 2004. Fundamental analysis for visible-light communication system using LED lights. *IEEE transactions on Consumer Electronics* 50, 1 (2004), 100–107.
[19] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta. 2014. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of the ACM MobiCom*. ACM, 447–458.
[20] Hoa Le Minh, Dominic O'Brien, Grahame Faulkner, Lubin Zeng, Kyungwoo Lee, Daekwang Jung, and YunJe Oh. 2008. High-speed visible light communications using multiple-resonant equalization. *IEEE Photonics Technology Letters* 20, 14 (2008), 1243–1245.
[21] Kwonhyung Lee and Hyuncheol Park. 2011. Modulations for visible light communications with dimming control. *IEEE photonics technology letters* (2011).
[22] Rimhwan Lee, Kyungsu Yun, Jong-Ho Yoo, Sung-Yoon Jung, and Jae Kyun Kwon. 2013. Performance analysis of M-ary PPM in dimmable visible light communications. In *Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on*. IEEE, 380–383.
[23] Liqun Li, Pan Hu, Chunyi Peng, Guobin Shen, and Feng Zhao. 2014. Epsilon: A Visible Light Based Positioning System. In *NSDI*. 331–343.
[24] Tianxing Li, Chuankai An, Zhao Tian, Andrew T Campbell, and Xia Zhou. 2015. Human sensing using visible light communication. In *Proceedings of the ACM MobiCom*. ACM, 331–344.
[25] Xia Li, Jelena Vucic, Volker Jungnickel, and Jean Armstrong. 2012. On the capacity of intensity-modulated direct-detection systems and the information rate of ACO-OFDM for indoor optical wireless applications. *IEEE Transactions on Communications* 60, 3 (2012), 799–809.
[26] Hyuncheol Park and John R Barry. 2004. Trellis-coded multiple-pulse-position modulation for wireless infrared communications. *IEEE transactions on communications* (2004).
[27] Samuel David Perli, Nabeel Ahmed, and Dina Katabi. 2010. Pixnet: interference-free wireless links using lcd-camera pairs. In *Proceedings of the ACM MobiCom*. ACM, 137–148.
[28] M Rahaim, A Miravakili, T Borogovac, TDC Little, and V Joyner. 2011. Demonstration of a software defined visible light communication system. In *Proceedings of the ACM MobiCom*.
[29] Sridhar Rajagopal, Richard D Roberts, and Sang-Kyu Lim. 2012. IEEE 802.15. 7 visible light communication: modulation schemes and dimming support. *IEEE Communications Magazine* 50, 3 (2012).
[30] Mark S Rea. 2000. Illuminating Engineering Society of North America. *The IESNA lighting handbook* (2000).
[31] K. Sato, T. Ohtsuki, I. Sasase, and S. Mori. 1993. Performance analysis of (m, 2) MPPM with imperfect slot synchronization. In *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*. 765–768 vol.2.
[32] Stefan Schmid, Giorgio Corbellini, Stefan Mangold, and Thomas R Gross. 2013. LED-to-LED visible light communication networks. In *Proceedings of the ACM MobiHoc*. ACM, 1–10.
[33] Abu Bakar Siddique and Muhammad Tahir. 2013. Joint rate-brightness control using variable rate MPPM for LED based visible light communication systems. *IEEE Transactions on Wireless Communications* 12, 9 (2013), 4604–4611.
[34] Hisayoshi Sugiyama and Kiyoshi Nosu. 1989. MPPM: A method for improving the band-utilization efficiency in optical PPM. *Journal of Lightwave Technology* 7, 3 (1989), 465–472.
[35] Zhao Tian, Kevin Wright, and Xia Zhou. 2016. The DarkLight rises: Visible light communication in the dark. In *Proceedings of the ACM MobiCom*. 2–15.
[36] Dobroslav Tsonev, Stefan Videv, and Harald Haas. 2013. Light fidelity (Li-Fi): towards all-optical networking. In *SPIE*.
[37] Qing Wang, Domenico Giustiniano, and Daniele Puccinelli. 2014. Openvlc: Software-defined visible light embedded networks. In *Proceedings of the 1st ACM MobiCom workshop on Visible light communication systems*. ACM, 15–20.
[38] Qing Wang, Domenico Giustiniano, and Daniele Puccinelli. 2015. An Open-Source Research Platform for Embedded Visible Light Networking. *IEEE Wireless Communication* (2015).
[39] Li-Che Wu and Hsin-Mu Tsai. 2013. Modeling vehicle-to-vehicle visible light communication link duration with empirical data. In *IEEE Globecom Workshops*. 1103–1109.
[40] Bo Xie, Guang Tan, and Tian He. 2015. Spinlight: A high accuracy and robust light positioning system for indoor applications. In *Proceedings of the SenSys*. ACM, 211–223.
[41] Shun-Hsiang You, Shih-Hao Chang, Hao-Min Lin, and Hsin-Mu Tsai. 2013. Visible light communications for scooter safety. In *Proceeding of the ACM MobiSys*.
[42] Shun-Hsiang Yu, Oliver Shih, Hsin-Mu Tsai, Nawaporn Wisitpongphan, and Richard D Roberts. 2013. Smart automotive lighting for vehicle safety. *IEEE Communications Magazine* 51, 12 (2013), 50–59.
[43] Fahad Zafar, Dilukshan Karunatilaka, and Rajendran Parthiban. 2015. Dimming schemes for visible light communication: the state of research. *IEEE Wireless Communications* 22, 2 (2015), 29–35.
[44] Chi Zhang, Josh Tabor, Jialiang Zhang, and Xinyu Zhang. 2015. Extending mobile interaction through near-field visible light sensing. In *Proceedings of the ACM MobiCom*. ACM, 345–357.