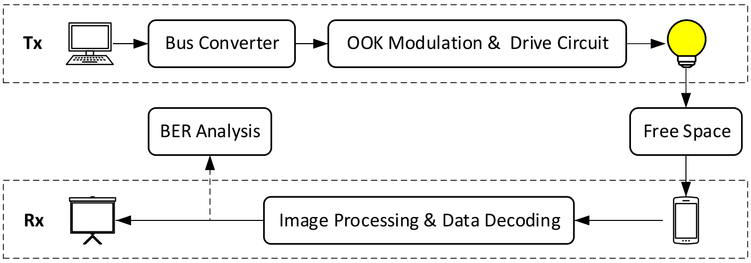
IEEE Photonics Journal
An IEEE Photonics Society Publication

Open Access

HyperSight: A Precise Decoding Algorithm for VLC With Mobile-Phone Camera

Volume 12, Number 4, August 2020

Yuanmei Meng
Xiongbin Chen
Tianhao Pan
Tong Shen
Hongda Chen


Tx
Bus Converter
OOK Modulation & Drive Circuit
BER Analysis
Free Space
Rx
Image Processing & Data Decoding

DOI: 10.1109/JPHOT.2020.3010716

HyperSight: A Precise Decoding Algorithm for VLC With Mobile-Phone Camera

Yuanmei Meng,[1] **Xiongbin Chen**,[1,2,3] **Tianhao Pan**,[1,4] **Tong Shen**,[5] **and Hongda Chen**[1]

[1]State Key Laboratory on Integrated Optoelectronics, IOS, CAS, Beijing 100083, China
[2]Center of Materials Science and Optoelectronics Engineering, UCAS,
Beijing 100049, China
[3]School of Electronic, Electrical and Communication Engineering, UCAS,
Beijing 101408, China
[4]School of Micro-Electronics, UCAS, Beijing 101408, China
[5]Huawei Technologies Co., Ltd, Beijing 518129, China

DOI:10.1109/JPHOT.2020.3010716

Manuscript received February 26, 2020; revised July 2, 2020; accepted July 16, 2020. Date of publication July 21, 2020; date of current version July 29, 2020. This work was supported in part by the National Key R&D Program of China under Grant 2017YFB0403605, in part by the National Natural Science Foundation of China under Grant 61875183, and in part by the High Tech Youth Talent Program of the Chinese Academy of Sciences under Grant GQRC-19-03. Corresponding author: Yuanmei Meng (e-mail: mengyuanmei@semi.ac.cn).

Abstract: In this paper, we propose and experimentally demonstrate HyperSight, which is a precise image decoding algorithm for the mobile-phone camera based visible light communication (VLC) system. The conventional algorithms have to select an appropriate column matrix, and then decode the image according to the fitting curve. The selection of the column matrix and the fitting of the function are not only time-consuming but also error-prone, leading to low decoding efficiency and high bit error rate (BER). While HyperSight calculates characteristic grayscale matrix to avoid column matrix selection and uses critical grayscale to replace the function fitting to decode the image. As expected, our algorithm not only reduces the BER, but also improves the processing performance. The experimental results show that, the BER can be reduced by two orders of magnitude on average compared with the conventional scheme. For the 4.8 kbps data rate, our algorithm can achieve the transmission distance of 50 cm under the 7% forward error correction (FEC) threshold with the low illuminance of 702 lux.

Index Terms: VLC, mobile-phone camera, blooming effect, critical grayscale.

1. Introduction

With the arrival of the 5G and the expectation of 6G, VLC has received considerable attention in the academic and commercial fields [1], [2]. As a wireless communication technology, VLC has attracted wide interest because of its abundant spectrum resources, fast transmission speed, and no electromagnetic radiation. At present, VLC has begun to play an important role in near field communication. Nowadays, there is an urgent need for new technologies with low power consumption and wide spectrum in the era of data explosion in mobile-phone terminals and IoT devices [3]. As we know, VLC not only has a huge unregulated bandwidth (375～780 nm) and outstanding security, but also integrates green lighting and communication functions. Thus, it receives more and more attention. Due to the widespread popularity of mobile-phones, the use of mobile-phone camera as the receiver of visible light communication becomes more and more popular [4]–[6]. However, the traditional frame rate of the mobile-phone camera image sensor with global shutter is pretty low as 25～60 frames per second. It cannot capture all the data sent from LEDs. Therefore, the rolling shutter effect of the complementary metal-oxide-semiconductor (CMOS) [7]–[9] image sensor is used to increase the data rate [10]. With the rolling shutter effect, the optical signal is recorded as bright and dark stripes which represents '1' and '0'.

However, the development of VLC encounters a bottleneck. Although the data rate is increased, two problems exist in rolling shutter pattern of the CMOS sensor. One problem is that camera is 'blind' in a frame-to-frame processing time gap [11]. So, the data between two frames cannot be captured. Accordingly, the original data is usually repeated three times to ensure that each frame of image captured by the mobile-phone contains a complete packet, which is convenient for original data restoration. Another problem is the blooming effect, which is caused by the overflow of charge from the saturated pixels into the neighboring pixels. As a result, the width of these fringes is distorted in the blooming area, and it severely causes the deteriorated performance of VLC system. The conventional solution is to select an ideal column matrix to avoid the blooming effect. Through this selection, the main blooming area can be avoided, while the contrast ratio (CR) between bright and dark stripes is still low. In the process of signal demodulation, the grayscale values of each stripe needs to be binarized by a threshold. This is a very important step because the binarization accuracy is directly related to the BER and system performance. The conventional thresholding schemes are based on multi-order polynomial fitting [11]–[14], such as 3 rd order polynomial fitting (3rd-OPF). However, the function fitting based thresholding scheme has limited system performance since it cannot resist the grayscale fluctuation. Additionally, it has high computational complexity. It limits the widespread application of VLC systems based on mobile-phone camera. To address these problems, many schemes have been proposed. A novel column matrix selection scheme based on energy diffusion of LED is proposed and experimentally demonstrated [15]. Weight moving average based thresholding schemes for VLC system, including equal weight moving average (EWMA) based thresholding scheme, fractional weight moving average (FWMA) based thresholding scheme and equal ratio weight moving average (ERWMA) based thresholding scheme, are presented to lower computational complexity [16], [17]. The normalization and the second-order polynomial CR enhancement are proposed as simple CR enhancement schemes [18]. Moreover, a novel thresholding scheme based on the boundary pixel based piecewise linear function thresholding (BPBPLFT) is first used to improve the accuracy of the data without using curve fitting thresholds [19]. However, these methods are inefficient and have limited usage scenarios. Hence, additional studies of high scalability are needed, and a more accurate and efficient technique is necessary.

In this work, we propose and experimentally demonstrate the HyperSight, which is a precise image decoding algorithm for the mobile-phone camera based VLC system. HyperSight calculates characteristic grayscale matrix to avoid column matrix selection and uses critical grayscale to replace the function fitting to decode the image. We implement a VLC test system as shown in Fig. 1, where the on-off-keying (OOK) modulator is used as the modulation scheme. The experimental results show that the scheme can not only avoid the main blooming area, but also can mitigate the high-intensity fluctuation of grayscale which caused by uneven light exposure of the mobile-phone image sensor. As expected, compared with the state-of-the-art algorithms, our algorithm not only reduces the BER, but also improves the processing performance.

The remainder paper is organized as follows. Section 2 provides the theoretical algorithm. Our experimental setup and results are described in Section 3. Section 4 demonstrates the advantages and potential of our proposed algorithm. Finally, we summarizes our work in Section 5.

2. Principle and Algorithm

As a precise and high-performance approach, HyperSight is proposed to decode the image captured by mobile-phone, which can be applied to various scenes with different light source distance,

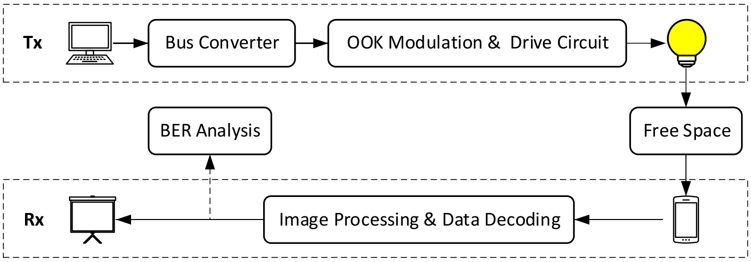
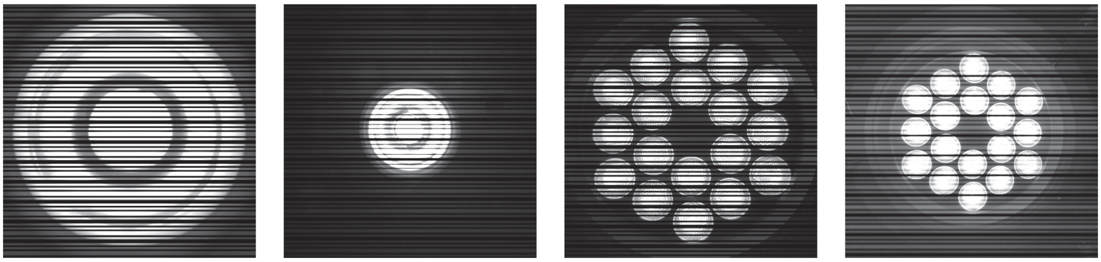

Tx
Bus Converter
OOK Modulation & Drive Circuit
Free Space
BER Analysis
Rx
Image Processing & Data Decoding

Fig. 1. Block diagram of mobile-phone camera based VLC system.



(a) 1 LED, 10 cm (b) 1 LED, 30 cm (c) 18 LEDs, 10 cm (d) 18 LEDs, 30 cm

Fig. 2. Grayscale images captured by mobile-phone camera under different conditions.

different number of LEDs, and different LED shapes. HyperSight has three main processes: 1) calculating the characteristic grayscale value of each row of image matrix, 2) deriving the grayscale value curve to determine the position of each critical grayscale, and 3) binarizing the grayscale value and decoding the data. We will describe them in detail below.

2.1 Characteristic Grayscale

As we know, the number, shape, and arrangement of LEDs used by various VLC systems are different. Besides, the distance between the mobile-phone camera and the light source is also different. Therefore, the boom and luminance of the captured images vary greatly (Fig. 2), which is a formidable challenge to guarantee the BER.

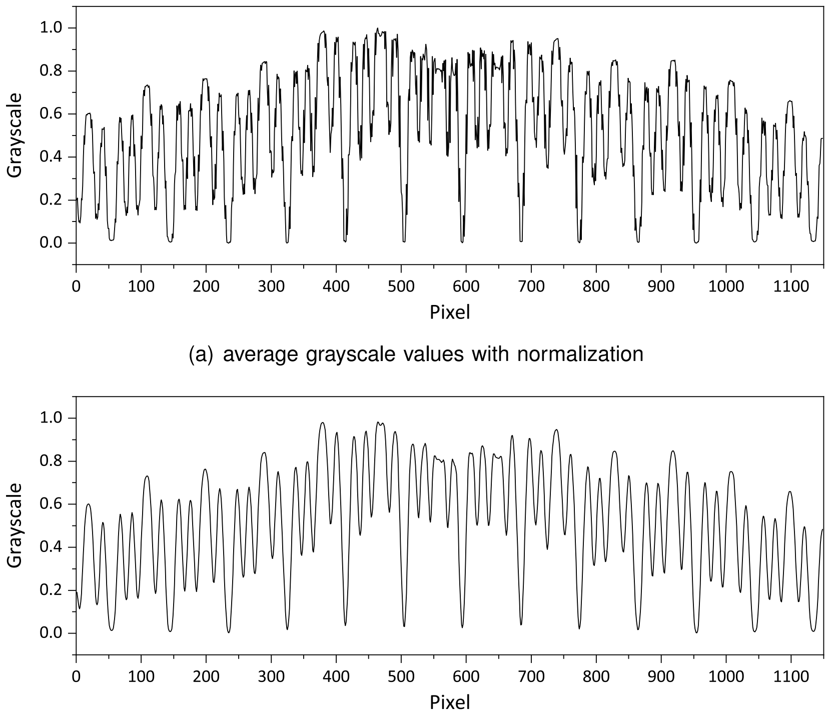
In order to prevent the bit error caused by the booming effect, the conventional methods are to select a proper column matrix that avoids boom as the data-bearing matrix according to the characteristics of the image [15], [17], [19]. These methods are complicated, and only support specific scenes. Poor scalability results in not finding such a column matrix in the image sometimes.

Therefore, we try to compute an ideal column matrix as the data-bearing matrix named as characteristic grayscale matrix.

Given an $i \times j$ pixel image

$$M_{i\times j} = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,j} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ m_{i,1} & m_{i,2} & \cdots & m_{i,j} \end{bmatrix}, \quad (1)$$

where $m_{i\times j}$ indicates the grayscale value of the pixel in row i and column j, we first calculate the average grayscale value of each row of image matrix, and the results constitute a preliminary



(a) average grayscale values with normalization

(b) average grayscale values with smoothing

Fig. 3. Characteristic grayscale matrix.

characteristic grayscale matrix

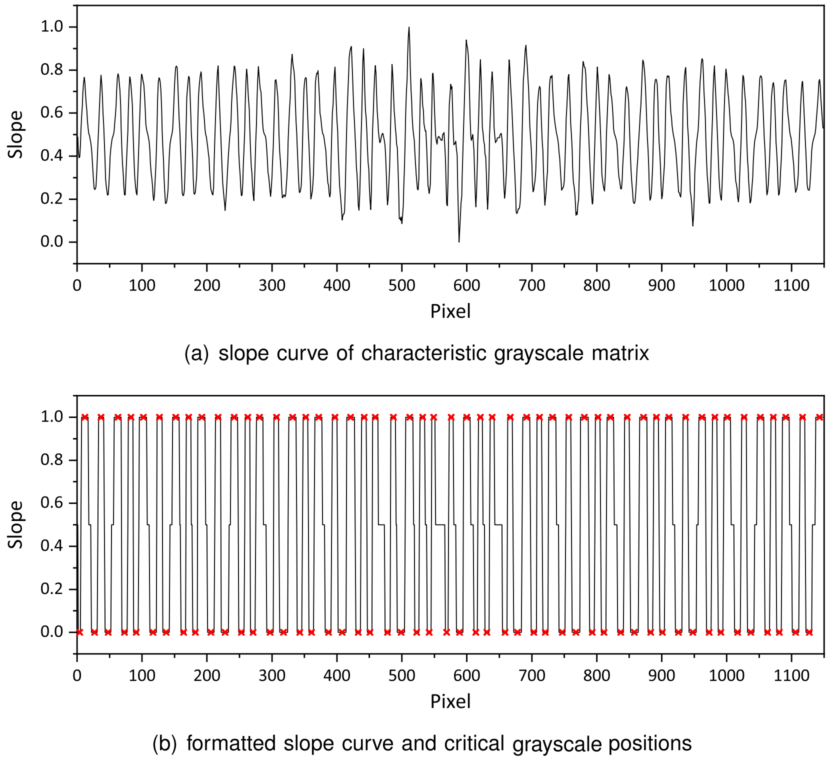
$$G = \begin{bmatrix} \bar{m}_1 \\ \bar{m}_2 \\ \vdots \\ \bar{m}_i \end{bmatrix}, \tag{2}$$

where $\bar{m}_k$, $k \in \{1, 2, \ldots, i\}$ indicates the average grayscale value of the k-th row of image matrix of $M_{i\times j}$. To reduce the booming effect, we can remove some large grayscale values before averaging by observing the relationship between the grayscale values and the number of pixels. It should be noted that removing the boom is not required, but optional. Because grayscale averaging can reduce the booming effect greatly, besides, experiments in Section 3 show that boom has little effect on the BER of our algorithm. Taking Fig. 2(d) as an example, Fig. 3(a) shows its normalized characteristic grayscale matrix without boom pixels.

To mitigate the fluctuation and glich of grayscale curve in Fig. 3(a), we smooth it by averaging the surrounding grayscale values as shown in Formula (3).

$$\bar{m}'_k = \frac{\sum_{p\in[k-\theta,k+\theta]} \bar{m}_p}{2\theta + 1}. \tag{3}$$

In Formula (3), θ is the smoothing coefficient which specifies the range of the neighborhood that contains all elements in a distance less than or equal to θ. And $\bar{m}'_k$ means the average value of the



(a) slope curve of characteristic grayscale matrix

(b) formatted slope curve and critical grayscale positions

Fig. 4. Find the critical grayscales by analysing the slope of grayscale values.

θ-neighborhood of $\bar{m}_k$. Finally we obtain an ideal characteristic grayscale matrix

$$G' = \begin{bmatrix} \bar{m}'_1 \\ \bar{m}'_2 \\ \vdots \\ \bar{m}'_i \end{bmatrix}. \tag{4}$$

So, we smooth the characteristic grayscale matrix in Fig. 3(a) by local averaging, and get the curve shown in Fig. 3(b).

2.2 Critical Grayscale

When we find the data-bearing matrix (characteristic grayscale matrix in our algorithm), the conventional approach, in general, is to find a fitting function (e.g. 3rd-OPF, FWMA [17] and BPBPLFT [19]) to decode the grayscale value into '0' and '1'. However, no matter what the fitting function is, it needs to undergo complicated calculations, which brings extra overhead and leads to the performance degradation of the decoding algorithm. Moreover, a specific fitting curve is only suitable for individual grayscale value curves.

As a result, we abandon this conventional method. We propose the concept of critical grayscale, which can directly binarize the values of characteristic grayscale matrix into '0' and '1'. As we discuss in Section 3, our method is not only more efficient and versatile, but also more precise.

We still take the Fig. 2(d) and its characteristic grayscale matrix curve (Fig. 3(b)) as an example to demonstrate this process. We take the first-order derivative of the characteristic grayscale matrix curve to get its corresponding slope, and show its normalized curve in Fig. 4(a). We can get the

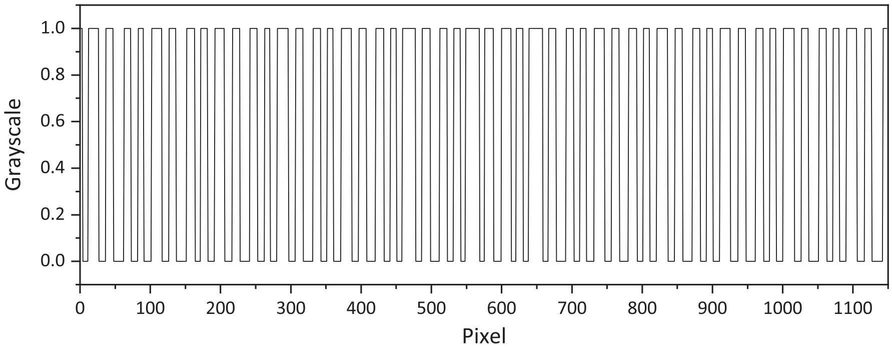


Fig. 5. Column matrix of grayscale values after binarization.

grayscale slope matrix S from G' through Equation (5).

$$S = \nabla G' = \begin{bmatrix} \bar{m}'_2 - \bar{m}'_1 \\ \bar{m}'_3 - \bar{m}'_2 \\ \vdots \\ \bar{m}'_i - \bar{m}'_{i-1} \end{bmatrix}. \tag{5}$$

The so-called critical grayscale refers to the adjacent grayscale between bright and dark stripes. One of the characteristics of the critical grayscale is that the difference between the critical grayscale value and the adjacent grayscale value reaches the maximum in its neighborhood. Thus, we can directly binarize the characteristic grayscale matrix if we can locate all the critical grayscales accurately. Therefore, we have to find all the critical grayscale positions, which actually correspond to the local maximum and minimum values of the slope curve in Fig. 4(a). However, the slope curve may have glitches at the extremes, making it difficult to find the exact location of the critical grayscale. To address this problem, we find a reasonable approximate solution by formatting the slope curve as follow. After normalization, we use 0.5 to indicate that the grayscale value has no change, while closer to 1 indicates that the grayscale value is extremely increased, and closer to 0 indicates that the grayscale value decreases sharply as shown in Fig. 4(a). We set all slope values greater than 0.55 to 1 which means that their corresponding grayscale values change from small to big (dark stripes to bright stripes), and set all slope values less than 0.45 to 0 which means that their corresponding grayscale values change from big to small (bright stripes to dark stripes). All the other remaining slope values are reset to 0.5. So we get a new slope curve

$$S' = \begin{bmatrix} f(\bar{m}'_2 - \bar{m}'_1) \\ f(\bar{m}'_3 - \bar{m}'_2) \\ \vdots \\ f(\bar{m}'_i - \bar{m}'_{i-1}) \end{bmatrix}, \quad f(s) = \begin{cases} 1, & s > 0.55 \\ 0.5, & 0.45 \leqslant s \leqslant 0.55, \\ 0, & s < 0.45 \end{cases} \tag{6}$$

which is shown in Fig. 4(b). As we can see, we mark the midpoints of all the short line segments as the positions of the corresponding critical grayscale values.

2.3 Binary Matrix

After we get these critical grayscales, we can directly figure out the binary matrix (Fig. 5) of the transmitted data. Because it must be a dark stripe if it is between the critical grayscale with slope '0' (at left side) and the adjacent critical grayscale with slope '1' (at right side), and a bright stripe if it is between the critical grayscale with slope '1' (at left side) and the adjacent critical grayscale with slope '0' (at right side).

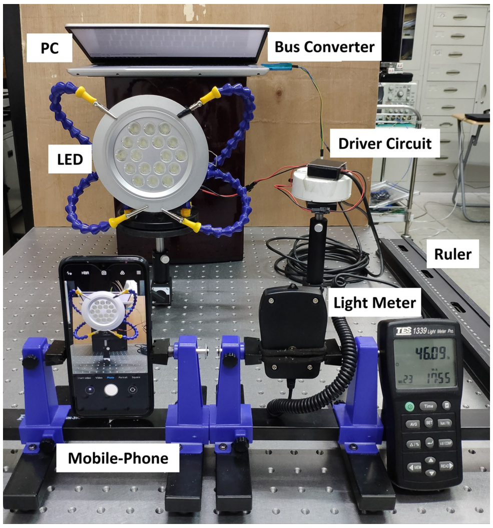


Fig. 6. Experimental setup of VLC system.

On the one hand, our decoding algorithm has no complicated computing, and it has higher performance than the fitting-based algorithms in theory. On the other hand, our algorithm is not sensitive to the shape of the characteristic grayscale matrix (i.e., the selected column matrix). So, it is more scalable than those fitting-based algorithms.

3. Experiments and Results

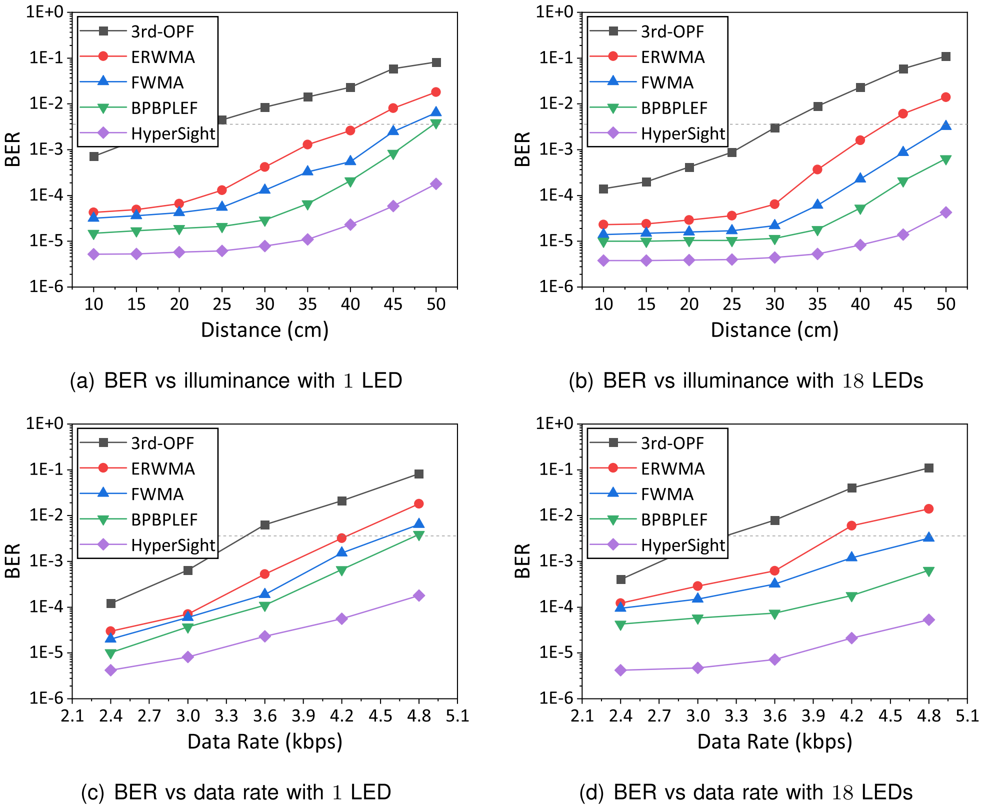
We compared the advantages and disadvantages of our algorithm HyperSight and the state-of-the-art algorithms under different conditions. In this section, we will describe our experimental setup and results.

3.1 Setup

Generally, the data transmission in a VLC system is shown in Fig. 1. The data is generated from a terminal device (e.g., PC, field programmable gate array, arbitrary waveform generator), and is passed through bus converter, OOK modulator and drive circuit in turn, and finally is sent out in the form of visible light through LEDs. Then, the mobile-phone camera captures the blinking of the LED and records it as a stream of images through the rolling shutter. We utilize a data decoding algorithm (e.g., HyperSight) to parse the image into the original data sent and show it on the display device (e.g., a screen, a projector).

Following this block diagram, we set up our experimental platform as shown in Fig. 6. On the transmitter side (Tx), we generate data online through a PC (RedmiBook XMA1901-AA). We use a USB-RS485 converter (Model 003) as a bus converter to send data to a drive circuit that can control the LED module. We use two kind of LED modules for testing. One is composed of a LED array with 18 LEDs and an LED lens module, and another is composed of only 1 LED (1 watt per LED). On the receiver side (Rx), we use a mobile-phone (Redmi 7) camera to capture the optical signal. The raw image resolution is 3000×3000 pixels. In particular, we use a ruler to measure the distance between the camera and the light source, and an illuminometer to measure the illuminance.

We set the transmission rate in our experiments as 2.4～4.8 kbps, which are enough to transmit the key messages for indoor positioning, navigation and smart home system [20]. We use the same randomly generated 8-bit sequence as the test data in each group of control experiments, which guarantees the fairness and universality of the experiments.



(a) BER vs illuminance with 1 LED

(b) BER vs illuminance with 18 LEDs

(c) BER vs data rate with 1 LED

(d) BER vs data rate with 18 LEDs

Fig. 7. BER under different conditions.

3.2 Comparison of Bit Error Rate

Bit error rate is an important indicator for evaluating the accuracy of any communication algorithm and system. As a type of near field communication, the BER of the VLC system is closely related to the distance of the light source and the data rate. We tested the BER of HyperSight and the state-of-the-art algorithms under different distances of illuminances and different data rates, and the result is shown in Fig. 7.

In our experiments, we use two demo VLC systems, one of which has a single 1-watt LED (Fig. 2(c) and (b)) and the other of which has an LED array with 18 1-watt LEDs (Fig. 2(c) and (d)), to compare the BER of various decoding algorithms respectively.

As we know, the BER is affected by many factors. So we cut in from two aspects: 1) different distances between the mobile-phone camera and the light source (corresponding to different illuminances); and 2) different data rates (corresponding to different baud rates). For each demo system, 5000 images were taken at different distance to the light source and different data rate. Each image carries 150 bits of random data. So, under any particular condition, we calculate the BER of 750 thousand bits to get the result.

We first fixed the data rate to 4.8 kbps and tested the BER of each algorithm within a range of 10 to 50 cm from the light source. The illuminances corresponding to these distances are 880, 433, 295, 217, 181, 156, 132, 117 and 103 lux with the 1-LED demo system, and 12823, 5569, 3412, 2304, 1744, 1393, 1066, 863 and 702 lux with the 18-LED demo systems, respectively. The results are shown in Fig. 7(a) and (b). Over the distances of 10 to 50 cm, HyperSight achieves BERs of 5.2×10^{-6} to 1.8×10^{-4} in the 1-LED demo system, and achieves BERs of 3.8×10^{-6} to 5.3×10^{-5} in the 18-LED demo system, respectively. The average BER of HyperSight in the 1-LED demo system is 3.4×10^{-5}, which is 0.22%, 4.65%, 8.31% and 21.38% of that of 3rd-OPF,

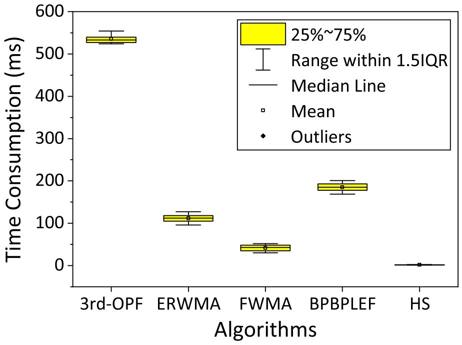


Fig. 8. Performance of different algorithms.

ERWMA, FWMA and BPBPLEF on average. And in the 18-LED system, HyperSight has a average BER of 1.2×10^{-5}, which is 0.70%, 7.40%, 15.28% and 28.47% of that of 3rd-OPF, ERWMA, FWMA and BPBPLEF on average. We then fixed the distance between the mobile-phone camera and the light source to 50 cm, and tested the BER of each algorithm within a range from 2.4 to 4.8 kbps of data rate. The results are shown in Fig. 7(c) and (d). Over the data rate of 2.4 to 4.8 kbps, the BER of HyperSight is 4.2×10^{-6} to 1.8×10^{-4} in the 1-LED demo system, and that in the 18-LED demo system is 4.2×10^{-6} to 5.3×10^{-5}, respectively. The average BER of HyperSight in the 1-LED demo system is 5.4×10^{-5}, which is 1.12%, 6.56%, 10.65% and 19.47% of that of 3rd-OPF, ERWMA, FWMA and BPBPLEF on average. And in the 18-LED system, HyperSight has a average BER of 1.8×10^{-5}, which is 0.29%, 1.39%, 2.64% and 9.51% of that of 3rd-OPF, ERWMA, FWMA and BPBPLEF on average. The dashed lines in the result figures represent the 7% FEC (equal to the BER of 3.8×10^{-3}) limit.

From the results we can see that, on the one hand, with the increasing of the distance (i.e., the decreasing of the illuminance), the BER of all algorithms gradually increases. It is worth noting that the BER of HyperSight is always the smallest of all comparison algorithms. Especially in the 1-LED demo system, when the distance reaches 50 cm, only HyperSight meets the FEC requirement of 7%. On the other hand, it can be observed that as the data rate increases, the BER increases. Compared with other algorithms, HyperSight always has the lowest BER at different data rates.

When the distance is greater than 30 cm, the captured image starts to have boom, thus the BER is significantly increased. Unlike conventional algorithms, where the selection of the column matrices is affected by the boom, HyperSight uses the characteristic grayscale matrix to alleviate the booming effect. Therefore, when boom occurs, HyperSight has a lower BER. Moreover, under the same illuminance condition, the BER of HyperSight in the special-shaped LED array demo system is slightly higher than that of the classic 1-LED system, while the BERs of the other algorithms have grown significantly. Therefore, we can conclude that HyperSight has better adaptability and compatibility to special-shaped LEDs or LED arrays.

3.3 Comparison of Time Consumption

In VLC systems, the time consumption of the decoding algorithm is directly related to the data rate that can be processed. We compared the average decoding performance of each algorithm, and the result is shown in Fig. 8.

In this experiment, we used 1-LED demo system and fixed the distance between the mobile-phone camera and the light source to 50 cm, and always sent data at a baud rate of 4.8 kbps. We tested each algorithm 32 times for a total of 160 performance tests. The average decoding time consumption of 3rd-OPF, ERWMA, FWMA, and BPBPLEF is 535.6, 111.6, 41.5 and 185.2 ms respectively, while HyperSight only takes only 1.7 ms to decode an image on average. It can be said that the system performance of HyperSight is nearly 315×, 65×, 24× and 109× that of the classic 3rd-OPF, ERWMA, FWMA, and BPBPLEF algorithms. This is due to the huge advantage of our proposed algorithm in terms of computational complexity.

4. Discussion

Our experiment did not test the BER when the mobile-phone camera was more than 50 cm away from the light source. This is because we designed the HyperSight for NFC scenarios, and 50 cm is enough to meet the distance requirements of NFC. Besides, in our experiments, the maximum data rate we tested was 4.8 kbps. This does not mean that HyperSight cannot support larger data rates. In fact, the maximum data rate supported by our algorithm can be much larger than 4.8 kbps. But due to the scanning frequency limitation of the mobile-phone camera and the pixel size of the images, we can only capture the data sent at the maximum rate of 4.8 kbps. However, as we claimed earlier, 4.8 kbps is enough to transmit all data in scenarios such as indoor positioning, navigation and smart home system.

With the arrival of the 5G and the expectation of 6G, more and more attentions are paid to VLC worldwide. Predictably, VLC will be widely applied in IoT and our daily life in future.

5. Conclusion

In this paper, we propose and experimentally demonstrate HyperSight for VLC with the mobile-phone camera. It can improve the performance of VLC system. Compared to the conventional column matrix selection scheme, our proposed decoding algorithm not only mitigates the booming effect, but also resists the high-intensity fluctuation to improve the contrast ratio between bright and dark stripes. Experiment results prove that the BER of HyperSight is lower than that of the conventional decoding algorithms which are based on column matrix selection and curve fitting. In addition, HyperSight can reduce the complexity of the binary judgment compared to those conventional algorithms. At the data rate of 4.8 kbps, HyperSight achieves a low BER of less than 10^{-5} on average, which is two orders of magnitude lower than that of conventional algorithms.

References

[1] P. A. Haigh and I. Darwazeh, “Real-time experimental demonstration of multi-band CAP modulation in a VLC system with off-the-shelf LEDs,” in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2019, pp. 1001–1002.

[2] I. Siddique, M. Z. Awan, M. Y. Khan, and A. Mazhar, “Li-Fi the next generation of wireless communication through visible light communication (VLC) technology,” *Int. J. Scientific Res. Comput. Sci., Eng. Inf. Technol.*, vol. 5, no. 1, pp. 30–37, 2019.

[3] I. Demirkol, D. Camps-Mur, J. Paradells, M. Combalia, W. Popoola, and H. Haas, “Powering the Internet of Things through light communication,” *IEEE Commun. Mag.*, vol. 57, no. 6, pp. 107–113, Jun. 2019.

[4] S.-H. Chen and C.-W. Chow, “Color-filter-free spatial visible light communication using RGB-LED and mobile-phone camera,” *Opt. Express*, vol. 22, no. 25, pp. 30713–30718, 2014.

[5] B. Zhang, K. Ren, G. Xing, X. Fu, and C. Wang, “SBVLC: Secure barcode-based visible light communication for smartphones,” *IEEE Trans. Mobile Comput.*, vol. 15, no. 2, pp. 432–446, Feb. 2016.

[6] A. K. Gupta and J. G. Andrews, “Comments on “coverage analysis of multiuser visible light communication networks”,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4605–4606, Sep. 2019.

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

[8] T.-H. Do and M. Yoo, “Performance analysis of visible light communication system using rolling shutter CMOS sensor,” *J. Korean Inst. Commun. Inf. Sci.*, vol. 40, no. 10, pp. 2065–2067, 2015.

[9] C.-W. Chow *et al.*, “Using advertisement light-panel and CMOS image sensor with frequency-shift-keying for visible light communication,” *Opt. Express*, vol. 26, no. 10, pp. 12530–12535, 2018.

[10] R. Deng, J. He, Y. Hong, J. Shi, and L. Chen, “2.38 kbits/frame WDM transmission over a CVLC system with sampling reconstruction for SFO mitigation,” *Opt. Express*, vol. 25, no. 24, pp. 30575–30581, 2017.

[11] C.-W. Chow, C.-Y. Chen, and S.-H. Chen, “Visible light communication using mobile-phone camera with data rate higher than frame rate,” *Opt. Express*, vol. 23, no. 20, pp. 26080–26085, 2015.

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

[13] K. Liang, C.-W. Chow, and Y. Liu, “Mobile-phone based visible light communication using region-grow light source tracking for unstable light source,” *Opt. Express*, vol. 24, no. 15, pp. 17505–17510, 2016.

[14] C.-W. Chow, C.-Y. Chen, and S.-H. Chen, “Enhancement of signal performance in LED visible light communications using mobile phone camera,” *IEEE Photon. J.*, vol. 7, no. 5, Oct. 2015, Art. no. 7903607.

[15] J. He, Z. Jiang, J. Shi, and Y. Zhou, “A novel column matrix selection scheme for VLC system with mobile phone camera,” *IEEE Photon. Technol. Lett.*, vol. 31, no. 2, pp. 149–152, Jan. 2019.

[16] Z. Zhang, T. Zhang, J. Zhou, Y. Qiao, A. Yang, and Y. Lu, “Performance enhancement scheme for mobile-phone based VLC using moving exponent average algorithm,” *IEEE Photon. J.*, vol. 9, no. 2, Apr. 2017, Art. no. 7903207.

[17] Z. Zhang, Y. Qiao, T. Zhang, and Y. Lu, “Fractional weight moving average based thresholding scheme for VLC with mobile-phone camera,” *IEEE Photon. J.*, vol. 11, no. 1, Feb. 2019, Art. no. 7901308.

[18] Y. Liu, “Decoding mobile-phone image sensor rolling shutter effect for visible light communications,” *Opt. Eng.*, vol. 55, no. 1, 2016, Art. no. 016103.

[19] Z. Zhang, T. Zhang, J. Zhou, Y. Lu, and Y. Qiao, “Thresholding scheme based on boundary pixels of stripes for visible light communication with mobile-phone camera,” *IEEE Access*, vol. 6, pp. 53053–53061, 2018.

[20] Y. Li, Z. Ghassemlooy, X. Tang, B. Lin, and Y. Zhang, “A VLC smartphone camera based indoor positioning system,” *IEEE Photon. Technol. Lett.*, vol. 30, no. 13, pp. 1171–1174, Jul. 2018.