Comparison of thresholding schemes for visible light communication using mobile-phone image sensor

Yang Liu,[1] Chi-Wai Chow,[2,*] Kevin Liang,[2] Hung-Yu Chen,[2] Chin-Wei Hsu,[2] Chung-Yen Chen,[2] and Shih-Hao Chen[2]

[1]Philips Electronics Ltd., Shatin, Hong Kong

[2]Department of Photonics and Institute of Electro-Optical Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan

[*]cwchow@faculty.nctu.edu.tw

Abstract: Based on the rolling shutter effect of the complementary metal-oxide-semiconductor (CMOS) image sensor, bright and dark fringes can be observed in each received frame. By demodulating the bright and dark fringes, the visible light communication (VLC) data logic can be retrieved. However, demodulating the bright and dark fringes is challenging as there is a high data fluctuation and large extinction ratio (ER) variation in each frame due. Hence proper thresholding scheme is needed. In this work, we propose and compare experimentally three thresholding schemes; including third-order polynomial curve fitting, iterative scheme and quick adaptive scheme. The evaluation of these three thresholding schemes is performed.

OCIS codes: (230.3670) Light-emitting diodes; (060.4510) Optical communications; (060.4080) Modulation.

References and links

1. H. H. Lu, Y. P. Lin, P. Y. Wu, C. Y. Chen, M. C. Chen, and T. W. Jhang, "A multiple-input-multiple-output visible light communication system based on VCSELs and spatial light modulators," Opt. Express **22**(3), 3468–3474 (2014).
2. C. H. Chang, C. Y. Li, H. H. Lu, C. Y. Lin, J. H. Chen, Z. W. Wan, and C. J. Cheng, "A 100-Gb/s multiple-input multiple-output visible laser light communication system," J. Lightwave Technol. **32**(24), 4723–4729 (2014).
3. B. Janjua, H. M. Oubei, J. R. Durán Retamal, T. K. Ng, C. T. Tsai, H. Y. Wang, Y. C. Chi, H. C. Kuo, G. R. Lin, J. H. He, and B. S. Ooi, "Going beyond 4 Gbps data rate by employing RGB laser diodes for visible light communication," Opt. Express **23**(14), 18746–18753 (2015).
4. H. Le-Minh, D. O'Brien, G. Faulkner, L. Zeng, K. Lee, D. Jung, and Y. Oh, "High-speed visible light communications using multiple-resonant equalization," IEEE Photonics Technol. Lett. **20**(14), 1243–1245 (2008).
5. C. H. Yeh, Y. L. Liu, and C. W. Chow, "Real-time white-light phosphor-LED visible light communication (VLC) with compact size," Opt. Express **21**(22), 26192–26197 (2013).
6. Z. Wang, C. Yu, W. D. Zhong, J. Chen, and W. Chen, "Performance of a novel LED lamp arrangement to reduce SNR fluctuation for multi-user visible light communication systems," Opt. Express **20**(4), 4564–4573 (2012).
7. S. Wu, H. Wang, and C. H. Youn, "Visible light communications for 5G wireless networking systems: from fixed to mobile communications," IEEE Netw. **28**(6), 41–45 (2014).
8. I. Takai, S. Ito, K. Yasutomi, K. Kagawa, M. Andoh, and S. Kawahito, "LED and CMOS image sensor based optical wireless communication system for automotive applications," IEEE Photonics J. **5**(5), 6801418 (2013).
9. P. Ji, H. M. Tsai, C. Wang, and F. Liu, "Vehicular visible light communications with LED taillight and rolling shutter camera," Proc. IEEE Vehicular Tech. Conf., 2014.
10. 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 **23**(20), 26080–26085 (2015).
11. C. W. Chow, C. Y. Chen, and S. H. Chen, "Enhancement of signal performance in LED visible light communications using mobile phone camera," IEEE Photonics J. **7**(5), 7903607 (2015).
12. R. C. Gonzalez and R. E. Woods, *Digital Image Processing* (Pearson, 2002).
13. P. D. Wellner, "Adaptive thresholding for the digital desk," Tech. Rep. EPC-93–110, 1993.
14. E. E. Richman, "Requirements for lighting levels," (https://www.wbdg.org/pdfs/usace_lightinglevels.pdf).

#252816 Received 28 Oct 2015; accepted 25 Dec 2015; published 25 Jan 2016

Corrected: 28 January 2016

1. Introduction

Visible light communication (VLC) [1–3] based on LED has been an active research for the last decades; and it has already brought VLC technology close to commercialization [4–6]. VLC could potentially play an important role in future 5G systems [7]. As there are already exiting LED light sources and displays in many different places in our daily life, this can facilitate the deployment of VLC using LED as transmitter (Tx). However, most of the proposed VLC systems use PIN photodiode (PD) as receiver (Rx), and this is not flexible to embed PIN PD in personal electronic devices. Hence, it is desirable if these VLC systems can be implemented by using the embedded complementary metal-oxide-semiconductor (CMOS) image sensor to provide low-cost VLC. However, using the CMOS image sensor as VLC Rx faces many confrontations since the typical frame rate is low (~30 Hz). A tailor-made CMOS image sensor with specific high speed pixels for VLC and low speed pixels for imaging is proposed [8]; however, this sensor needs complicated fabrication process. By using the rolling shutter effect of CMOS image sensor, the transmission data rate can be potentially higher than the frame rate [9–11].

Owing to the rolling shutter effect of the CMOS image sensor, bright and dark fringes can be observed in each received frame. By demodulating the bright and dark fringes, the VLC data logic can be retrieved. Hence the VLC link can be operated faster than the CMOS frame rate. However, demodulating the bright and dark fringes is challenging as there is a high data fluctuation and large extinction ratio (ER) variation in each frame. The high data fluctuation and large ER variation seldom occur in traditional optical fiber communication; hence proper thresholding scheme to define the logic 1 and 0 is needed. Here, we propose and compare experimentally three thresholding schemes using image processing techniques. These thresholding schemes include third-order polynomial curve fitting, iterative scheme [12] and quick adaptive scheme [13]. At [10] it was reported the preliminary work of using rolling shutter effect for VLC detection. In the selection of column matrix of pixels, manual selection is used in order to avoid blooming effect. This manual selection is slow and inefficient. When compared with [10, 11], here we propose and demonstrate two new thresholding schemes (iterative thresholding and quick adaptive thresholding). The significance of these two thresholding schemes is that there is no need to have the ER enhancement process reported before. When compared with [11], the processing time of the proposed iterative thresholding and quick adaptive thresholding can be significantly reduced by 97% and 6% respectively.

2. Algorithms of proposed threshold schemes and experiment

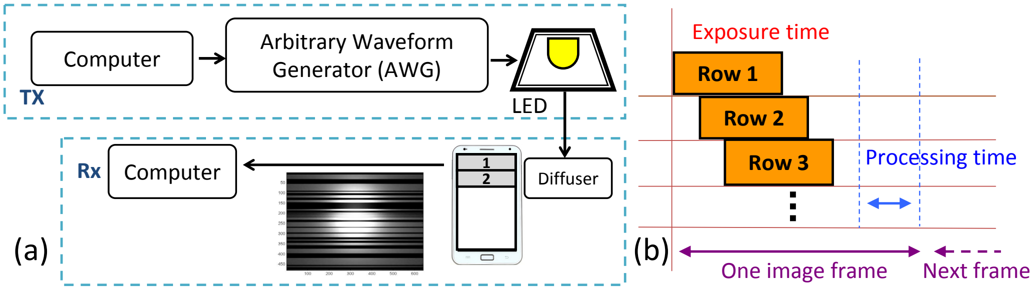

Fig. 1. (a) A proof-of-concept experimental setup of the VLC using mobile-phone CMOS image sensor, inset: an image frame captured by the CMOS image sensor; (b) the CMOS rolling shutter effect showing the overlap of pixel row and the processing time.

Figure 1(a) shows the proof-of-concept experimental setup of the VLC using mobile-phone CMOS image sensor. The data is generated in a computer Matlab program, which is then stored in an arbitrary waveform generator (AWG, Tektronix, AFG 3252C) with 2 GSample/s sampling rate and 240 MHz bandwidth for digit-to-analog conversion (DAC). The data then drives a single white-light LED (Cree XLamp XR-E) with modulation bandwidth of about 1 MHz. The VLC signal is then received by a mobile-phone CMOS image sensor (Samsung

#252816 Received 28 Oct 2015; accepted 25 Dec 2015; published 25 Jan 2016

GT-S7500), with 480 x 640 pixels resolution and 28 frame/second frame rate. Then the movies are transferred to a computer for demodulation.

When the CMOS image sensor is exposed to light, each row of pixels is activated sequentially. This is called the rolling shutter effect, which can be used to increase the VLC data rate faster than the frame rate; and bright and dark fringes can be observed in each frame. By demodulating these fringes, the VLC data logic can be retrieved. During the rolling shutter operation, the activation of each pixel row will overlap with the neighbor pixel row, as shown in Fig. 1(b); hence each pixel row cannot represent one logic bit. Besides, after the exposure time, there is a processing time; which is the time period needed for merging different pixels into a single image frame. During this time, the CMOS sensor cannot sense any signal even though it is under the light exposure. The measured processing time in our mobile-phone is 14.29 ms (~40% of an image frame); hence, each data packet should be transmitted 3 times successively to ensure each image frame captured by the mobile-phone contains a complete data packet. In our experiment, each packet consists of a 4-bit header (in Manchester coding format), 32-bit payload and 1-bit trailer (in on-off keying (OOK) format). In order to ensure each image frame contains a complete data packet, it cannot be too long; and we select 32-bit for the payload. Besides, we should keep the header short to carry more payload information. The selection of 4-bit header is enough for the synchronization in our program. The header is for clock recovery, while the trailer is always at logic 0 in order to distinguish different packets. By removing the duplicated 3 successively transmitted data packets and the header, the net data rate is 0.896 kbit/s.

In the synchronization and demodulation processes, the raw movie file is converted into the format readable by the Matlab program. Each image in the movie is converted into grayscale format, in which 255 represents completely bright level while 0 represents completely dark level. Then a column matrix (480 x 1) of grayscale levels is selected in each image to represent the logic bits as shown in Fig. 2(a) and 2(d). The detail description of column selection for decoding in order to avoid blooming effect has been discussed in [11]. As the header is in Manchester coding format, much narrower bring and dark fringes can be easily distinguished from the OOK data. Finally, proper threshold scheme to define the logic 1 and 0 is needed.

Here we describe the algorithms of the thresholding schemes. The first thresholding scheme is the third-order polynomial curve fitting. The resolution of the CMOS sensor we used is 480 x 640 pixels. Assume each element in a selected column matrix of grayscale is (x_i, y_i), where x_i is the position of the pixel, and y_i is the grayscale value of that pixel, $i = 1, 2, \dots 480$. Then the third-order polynomial fitting curve $f(x_i)$ is shown in Eq. (1),

$$f(x_i; a_0, a_1, a_2, a_3) = a_0 + a_1 x_i + a_2 x_i^2 + a_3 x_i^3 \tag{1}$$

then the square deviation is shown in Eq. (2),

$$[y_i - f(x_i)]^2 \tag{2}$$

and the total square deviation E can be represented in Eq. (3),

$$E(a_0, a_1, a_2, a_3) = \sum_{i=1}^{480} [y_i - f(x_i)]^2 = \sum_{i=1}^{480} [y_i - (a_0 + a_1 x_i + a_2 x_i^2 + a_3 x_i^3)]^2 \tag{3}$$

By setting $\frac{\partial E}{\partial a_0}, \frac{\partial E}{\partial a_1}, \frac{\partial E}{\partial a_2}, \frac{\partial E}{\partial a_3} = 0$, we can obtain four simultaneous equations; hence we can solve these equations and obtain the values of a_0, a_1, a_2, a_3. After finding the values of a_0, a_1, a_2, a_3 in Eq. (1), a third order polynomial curve can be constructed as shown in Fig. 2(a). This curve will be the threshold. At each pixel position, if the grayscale value above the third order polynomial curve (threshold), logic one is recorded; if the grayscale value below the curve, logic zero is recorded.

#252816 Received 28 Oct 2015; accepted 25 Dec 2015; published 25 Jan 2016

Another thresholding scheme we apply is the iterative thresholding, in which the whole data packet including the header and payload will be divided into several sections. For example, if 20 pixels are grouped in each section, there are 24 sections. 480 should be dividable by the number of pixels. Then in each section, we apply iterative operation. Assume y_i is the grayscale value of that pixel, $i = 1, 2, \ldots 20$. Then the initial average grayscale value T in this section is represented in Eq. (4),

$$T = \sum_{i=1}^{20} \frac{y_i}{20} \tag{4}$$

Then, two subsets of grayscale values R_1 and R_2 can be defined as shown in Eq. (5) based on the initial average grayscale value T.

$$\begin{cases} y_i \in R_1, & y_i \geq T \\ y_i \in R_2, & y_i < T \end{cases} \tag{5}$$

After this, we calculate the average grayscale values of the two subsets R_1 and R_2, and obtain U_1 and U_2 respectively. Then, the two average grayscale values U_1 and U_2 are added together and divided by 2 to obtain the new grayscale value T_k in the iterative process, as shown in Eq. (6).

$$T_k = \frac{U_1 + U_2}{2} \tag{6}$$

The new T_k will replace the initial T in Eq. (5), and the process described in Eq. (6) is repeated until $T_k = T$. The ultimate grayscale value will be the threshold obtained in the iterative thresholding process.

The last proposed thresholding scheme is the quick adaptive thresholding [13]; basically its operation is to calculate a moving average of grayscale values of pixels. This algorithm can be easily implemented in hardware [13]. Let y_i be the grayscale value of a pixel at point i, assume $f_s(i)$ be the sum of the values of the last s pixels at point i; hence it can be represented in Eq. (7),

$$f_s(i) = \sum_{n=0}^{s-1} y_{i-n} \tag{7}$$

A much faster way to calculate the weighted moving average is to subtract *1/s* part of it and add the value of only the latest pixel instead of all s pixels. This is like emphasizing the grayscale values closer to the target value. Then the threshold in the quick adaptive thresholding is represented by Eq. (8), where r is the adjustment ratio.

$$T = r\sum_{n=0}^{s-1} (1-\frac{1}{s})^n y_{i-n} / \sum_{n=0}^{s-1} (1-\frac{1}{s})^n \tag{8}$$

3. Results and discussion

We then compare experimentally the three thresholding schemes. Figure 2(a) shows the grayscale values and the threshold (red curve) obtained by the third-order polynomial curve fitting. Figures 2(b) and 2(c) show the grayscale values and the threshold (orange curve) by using the iterative scheme with different number of sections. By dividing too few sections (i.e. more pixels included in each section), the threshold curve cannot properly locate at the middle of the grayscale pattern. By dividing too many sections (i.e. less pixels included in each section), the threshold curve will approach the maximum grayscale values of the pattern, as shown in Fig. 2(c). Obviously, there is an optimum section number. Figure 2(d) shows the grayscale values and the threshold (pink curve) obtained by using quick adaptive thresholding. It is worth to mention that the data pattern shown in Fig. 2 is only one column

#252816 Received 28 Oct 2015; accepted 25 Dec 2015; published 25 Jan 2016

matrix of grayscale values selected in one frame. For each measurement, we will have > 1000 image frames. The position of the header and the payload in each frame will be different; hence proper thresholding schemes reported here are crucial and highly required.

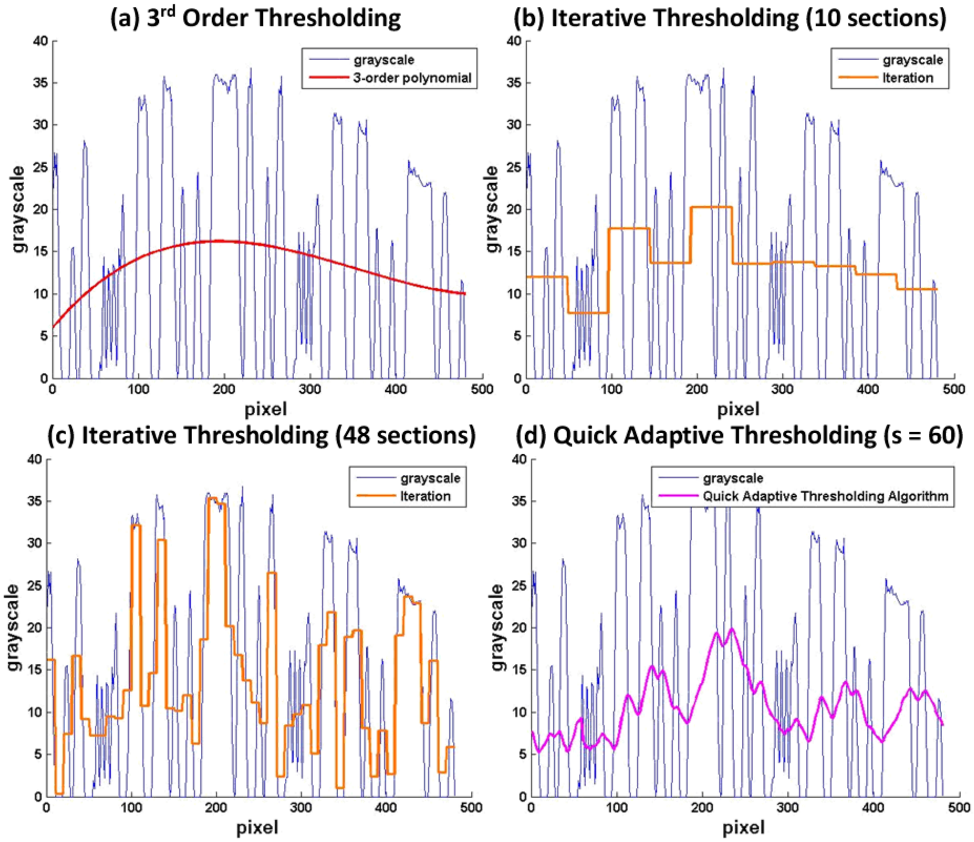

Fig. 2. Experimental grayscale values with (a) third order polynomial thresholding, (b), (c) iterative thresholding with 10 and 48 sections, and (d) quick adaptive thresholding (s = 60) for comparison.

Finally, we evaluate the BER performances of the three thresholding schemes. Figure 3(a) show the BER performance of using iterative scheme with different number of sections. As discussed in last paragraph, there is an optimum section number. We can observe that the optimum section number is 8. Figure 3(b) shows using different number of pixels *s* in the quick adaptive scheme; and the optimum *s* is around 60. Then, the BER performances of the third-order polynomial scheme, the iterative scheme (using 8 sections), the quick adaptive scheme (*s* = 60), and the scheme reported in [11] are compared and shown in Fig. 3(c). As the third-order polynomial threshold scheme is not actuate enough for the fast changing data; hence higher BER is observed; while the quick adaptive scheme outperforms the other two schemes reported in this work. As discussed before [11], requires ER enhancement process (histogram equalization + Sobel filter) together with third order thresholding scheme, while the proposed iterative and quick adaptive thresholding schemes do not. In our experiment, when the illuminance is > 5,000 lux, the saturation effect (merging of rows) is significantly. For the next level research, we can also apply different extinction ratio (ER) enhancement schemes to increase the contrast ratio of bright and dark fringes, it is expected that better signal fidelity can be achieved. In the experiment, illuminance of 750 lux is needed to meet the forward error correction (FEC). According to requirements for lighting levels [14], illuminance of 500 lux is recommended for office involving computer tasks; while 750-1000

#252816 Received 28 Oct 2015; accepted 25 Dec 2015; published 25 Jan 2016

lux is recommended for office involving paper-based reading tasks. Illuminance of 750 lux is also used in supermarkets, mechanical workshops, etc, and 1000 lux is used in drawing workshops, detailed mechanical workshops and operation theatres, etc.

We have also evaluated the processing time of different schemes. The computer with an Intel i5-4210U processor @ 1.7 GHz, 4 GB RAM; Matlab 2007 is used. The processing times required for the second order polynomial [10], ER enhancement process (histogram equalization + Sobel filter) together with third order thresholding [11], third order polynomial, iterative, and quick adaptive schemes are 0.95 ms, 8.2 ms, 1.2 ms, 0.2 ms, 7.7 ms, respectively. We can observe that [11] needs the longest processing time and iterative scheme needs the least processing time.

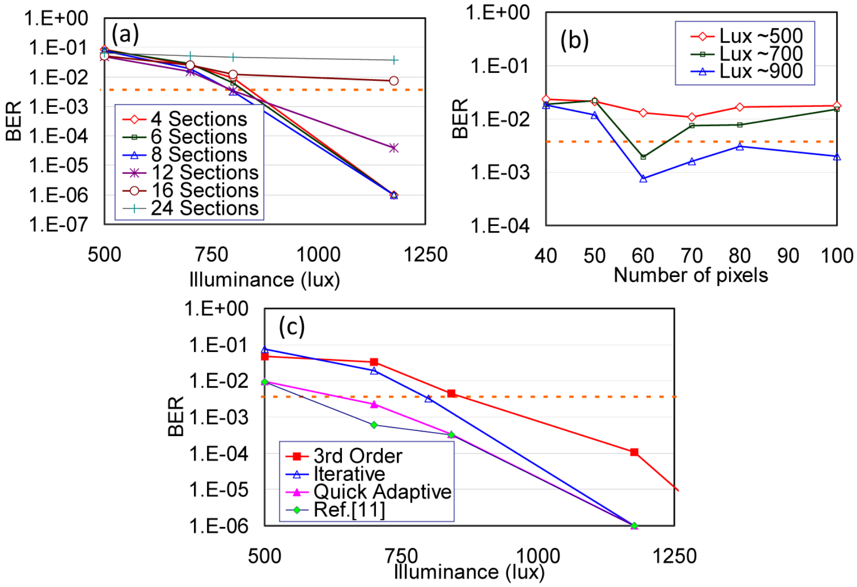

Fig. 3. (a) BER of iterative thresholding using different number of sections, showing the optimum is 8 sections, (b) BER of quick adaptive thresholding using different number of pixels s, showing the optimum is s = 60; and (c) BER of the third-order polynomial thresholding, iterative scheme (using 8 sections), quick adaptive scheme (s = 60), and the scheme reported in [11] are compared.

4. Conclusion

By demodulating the bright and dark fringes of the image received by the CMOS image sensor, the VLC data logic can be retrieved. However, demodulating the bright and dark fringes is challenging as there is a high data fluctuation and large ER variation in each frame. Here we proposed and compared experimentally three thresholding schemes with detail implementing algorithms. These thresholding schemes included third-order polynomial curve fitting, iterative scheme and quick adaptive scheme. In iterative thresholding scheme, using 8 sections was the optimum. In the quick adaptive thresholding scheme, using s = 60 pixels was the optimum. As the third-order polynomial threshold scheme was not actuate enough for the fast changing data; hence higher BER was observed; while the quick adaptive scheme outperformed the other two schemes reported in this work. Although quick adaptive thresholding scheme performed the best, it needed longer processing time.

Acknowledgments

This work was supported by Ministry of Science and Technology, Taiwan, MOST- 104-2628-E-009-011-MY3, MOST-103-2221-E-009-030-MY3, Aim for the Top University Plan, Taiwan, and Ministry of Education, Taiwan.

#252816 Received 28 Oct 2015; accepted 25 Dec 2015; published 25 Jan 2016