Decoding algorithm with multiple features based on optical camera communication system*

ZHANG Junming, SHI Wenxiao, WANG Qiang, LIU Anqi, and LIU Wei**

Department of Communications Engineering, Jilin University, Changchun 130012, China

(Received 21 June 2022; Revised 17 September 2022)

The performance of decoding algorithm is one of the important influential factors to determine the communication quality of optical camera communication (OCC) system. In this paper, we first propose a decoding algorithm with adaptive thresholding based on the captured pixel values under an ideal environment, and then we further propose a decoding algorithm with multiple features, which is more suitable under the existence of the interference of light sources. The algorithm firstly determines the light-emitting diode (LED) array profile information by removing the interfering light sources through geometric features, and then identifies the LED state by calculating two grayscale features, the average gray ratio (*AGR*) and the gradient radial inwardness (*GRI*) of the LEDs, and finally obtains the LED state matrix. The experimental results show that the bit error ratio (*BER*) of the decoding algorithm with multiple features decreases from 1×10^{-2} to 5×10^{-4} at 80 m.

Document code: A **Article ID:** 1673-1905(2023)02-0065-7

DOI https://doi.org/10.1007/s11801-023-2108-z

Optical camera communication (OCC) is a pragmatic version of visible light communication (VLC) technology[1] that applies image sensor (IS) as the receiver. OCC technology[2,3] has the advantages of no radio frequency (RF) interference, no need for spectrum authentication, and no electromagnetic radiation. Therefore, OCC technology has great potentialities for application in scenarios, such as intelligent transportation systems, indoor communication and positioning[4,5].

Due to the spatial imaging characteristics of the camera receiver, OCC systems can reduce the impact of ambient light aided by related image processing techniques[6,7]. Decoding algorithms have been studied by researchers worldwide from different perspectives. The study of decoding algorithms can be divided into global shutter mode decoding[8] and rolling shutter mode decoding[9] depending on the type of receiver. The key to the former decoding algorithm is to determine the light-emitting diode (LED) status by the distribution of grey values in the total LED image area, while the key to the latter is to determine the LED state by the grey value of the pixels in the respective striped image area. Besides, there are common decoding algorithms including traditional image decoding method[10,11] as well as deep learning decoding method[12,13]. The former method identifies the state of the LED light source by analyzing the distribution pattern of the gray value of the image, while the latter method identifies the state of the LED light source autonomously through the learned experience.

In the existing research, there are some performance limitation cases of decoding algorithms from two aspects. One is that existing studies have identified the state of LEDs by a fixed binarization threshold, without considering the effect of ambient light intensity changes on the overall gray value of the image, such as conventional directional projection method[14]. The other is that existing studies have analyzed the performance of decoding algorithms through simulation or in an ideal environment without noise, without considering the effect of interfering light sources, such as grayscale feature-based LED bit detection algorithm[15].

Based on the above-mentioned limitations, considering that different environments have different requirements for decoding algorithms, we have considered two different environments in order to achieve more targeted decoding. In an ideal environment, we introduce adaptive thresholding based on the directional projection method to further optimize the decoding performance to combat changing ambient light intensity and reduce the sensitivity of the binarization threshold. As for the environment with interfering light sources, Ref.[15] proposed a grayscale feature-based LED bit detection algorithm. However, the performance of the decoding algorithm is only tested on the simulated LED images and lacks the practical verification. Besides, the impact of ambient interference light sources on the decoding performance of the OCC system

* This work has been supported by the Department of Science and Technology of Jilin Province (No.20200401122GX).

** E-mail: swx@jlu.edu.cn

is also not considered. Therefore, we propose a ulti-feature-based LED decoding algorithm that has been validated in an experimentally setup OCC system. The algorithm first introduces a geometric feature-based LED array detection algorithm to eliminate the influence of interfering light sources, while obtaining the LED array contour simultaneously, and then compensates the LED array contour information to obtain accurate LED area information, and finally completes the decoding by grayscale feature-based LED bit detection algorithm.

The schematic diagram of the OCC system is shown in Fig.1.

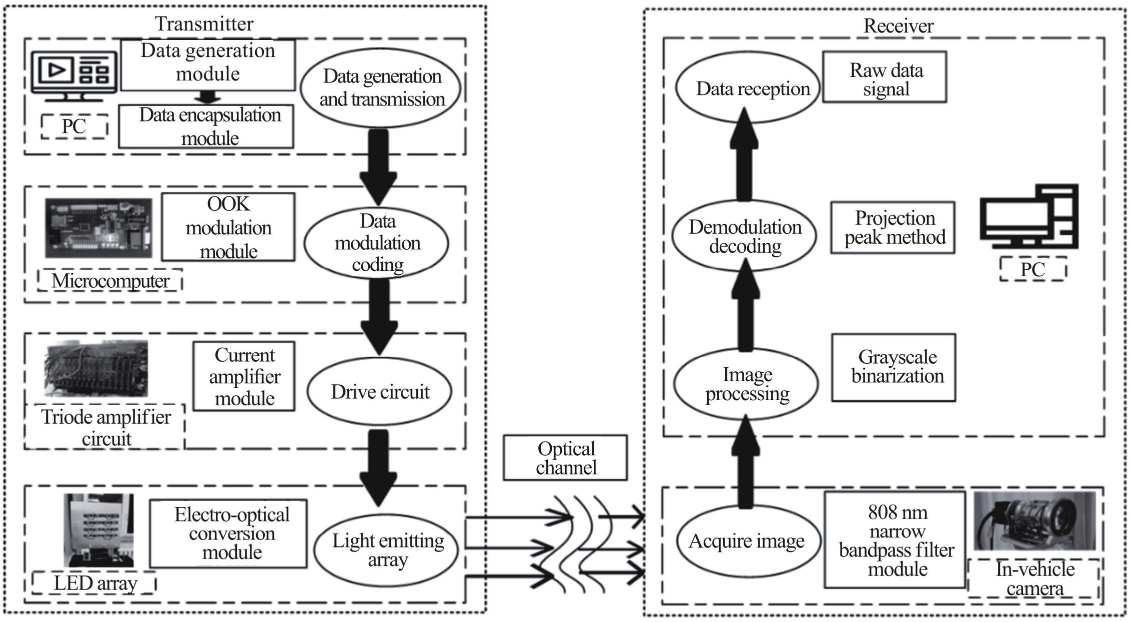


Fig.1 Schematic diagram of the OCC system

The OCC system transmitter uses a personal computer (PC) to randomly generate and encapsulate the raw data, and sends the encapsulated data to the microcontroller, which modulates the data with on-off keying (OOK) and transmits it to the driver circuit, and the circuit uses a triode to amplify the received signal to drive the LEDs. The LEDs in the array all use 808 nm wavelength near infrared light source to send light signals. The parameters of each part of the transmitter are shown in Tab.1.

Tab.1 Parameters of each part of the transmitter

Name of parameter	Value
PC version	Intel(R) Core(TM) i5-9300HCPU@2.40 GHz
MCU version	C8051F340
MCU output port	P2.0—P2.7, P3.0—P3.7
Amplifier circuit output current	75 mA
LED wavelength	808 nm
LED diameter	2 mm
LED forward voltage	1.5—2.4 V
LED maximum forward current	1 000 mA
Adjacent LED spacing	3 cm
LED arrangement	4×4 (16LEDs)

The system receiver uses a PC to complete image processing and demodulation, and the in-vehicle camera reduces the complexity of image processing by installing a narrowband filter centered at 808 nm wavelength in front of the camera lens to filter out stray light. The LED state matrix obtained by demodulation from the image is the raw data. The parameters of the receiver (camera) are shown in Tab.2.

Tab.2 Parameters of each part of the receiver

Name of parameter	Value
Pixel size	6.5 μm×6.5 μm
Resolution	2 048(H)×2 048(V)
Focal length	11—197 mm
Frame rate	50 fps
Spectral response interval	400—900 nm

Based on above OCC system, we use the decoding algorithm based on the directional projection method to obtain the state matrix in an interference-free light source environment. Using a fixed binarization threshold, we convert the original grayscale image containing the LEDs captured by the camera into the binary image, and then add up the pixel values of the pixel points in the vertical and horizontal directions of the binary image to obtain two projection histograms and LED central point coordinates, and finally calculate the coordinates of each LED state sampling point. The LED status is determined by the pixel value of each LED status sampling point on the binary image. If the pixel value of the sampling point is 255, the LED state is judged to be "1", otherwise, the LED state is judged to be "0".

As shown in Fig.2(a) and Fig.2(b), the algorithm identifies the LED state only by the image after binarization of the fixed threshold, which is highly dependent on the sensitivity of the binarization threshold, so the choice of different fixed thresholds can cause different degrees of error codes.

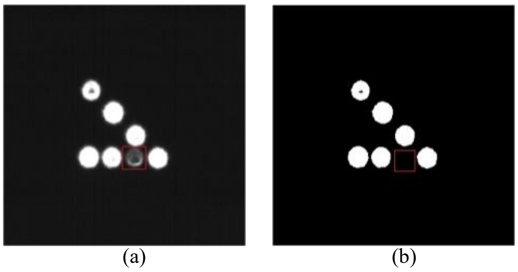


Fig.2 (a) Original image; (b) Binary image

To reduce the dependence of the decoding algorithm on the sensitivity of the binarization threshold, we use the Otsu threshold selection method[16]. The basic principles are as follows.

Let the total number of image pixels be N, the gray level be L, the number of pixels with gray value i (i=0, 1,..., $L-1$) be $h(i)$, the binarization threshold be t, and let the pixels with gray values in the range [0, t] constitute the background C_1, and the pixels with gray values in the range [t+1, $L-1$] constitute the background C_2.

Let p_1, p_2 be the ratios of C_1, C_2 to the total number of pixels respectively. Their values are given as

$$\begin{cases} p_1(t)=\dfrac{1}{N}\times\sum\limits_{i=0}^{t}h(i) \\ p_2(t)=\dfrac{1}{N}\times\sum\limits_{i=t+1}^{L-1}h(i) \end{cases}. \tag{1}$$

Let the mean grey value of the whole image be μ, and the value is given as

$$\mu=\frac{1}{N}\times\sum_{i=0}^{L-1}ih(i), \tag{2}$$

and the mean grey value of the pixels within C_1 and C_2 be μ_1 and μ_2 respectively. The values are given as

$$\begin{cases} \mu_1=\dfrac{1}{\sum\limits_{i=0}^{t}h(i)}\times\sum\limits_{i=0}^{t}ih(i) \\ \mu_2=\dfrac{1}{\sum\limits_{i=t+1}^{L-1}h(i)}\times\sum\limits_{i=t+1}^{L-1}ih(i) \end{cases}. \tag{3}$$

When the threshold is t, the inter-class variance of the image is shown as

$$\sigma^2=p_1(t)\times(\mu_1(t)-\mu)^2+p_2(t)\times(\mu_2(t)-\mu)^2. \tag{4}$$

The final threshold calculated to maximize the inter-class variance (σ^2) is the optimal threshold for binarization in an interference-free environment.

As for the environment with interfering light sources, based on the grayscale feature-based LED bit detection algorithm, we propose a multi-feature-based LED decoding algorithm, which introduces a geometric feature-based LED array detection algorithm to eliminate the influence of interfering light sources while obtaining the LED array profile, and then compensates the LED array profile information to obtain accurate LED area information, ultimately achieving more accurate decoding.

The specific steps are as follows.

Step 1. The camera captures the grayscale image, as shown in Fig.3(a). Otsu thresholding selection method is used to binary process the grayscale image to obtain the binary image, as shown in Fig.3(b).

Step 2. As shown in Fig.3(c), the binary image is dilated to obtain a candidate region containing the LED array and the interfering light source.

Step 3. The most values of x_{min}, x_{max}, y_{min}, y_{max} of the pixel point coordinates with pixel value of 255 in each candidate region are taken as the coordinates of the boundary of each candidate region. The red area in Fig.3(d) is the candidate region contour after dilation, and the green area is the candidate area contour after fitting. Considering the distortion of the imaging area resulting from the different luminous intensity of the LEDs, an aspect ratio of 0.8 to 1.2 was used instead of the original 1 as the determination condition 1. In addition, the LED array pixel width is inversely proportional to the communication distance, the closest (10 m) and farthest (80 m) communication distances corresponding to the width of the LED array are 192 pixels and 33 pixels, so the width range of 30—200 pixels is used as the determination condition 2. Candidate region 3 satisfies both condition 1 and condition 2, and is determined to be an LED array.

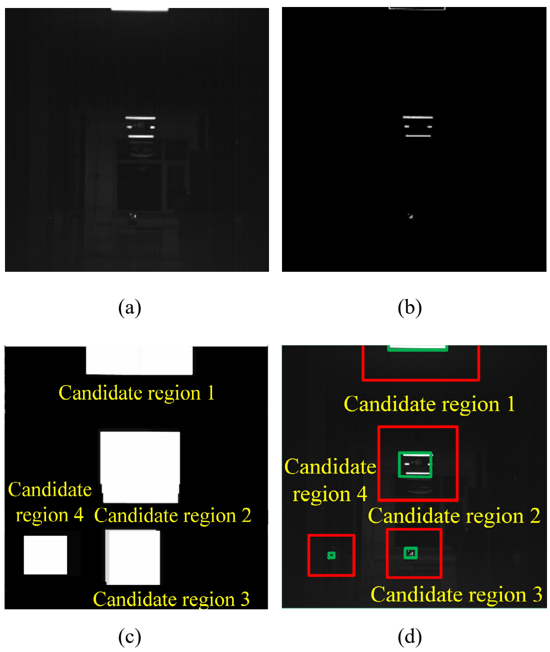
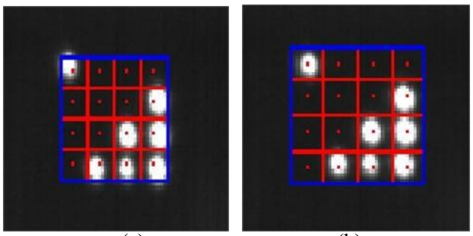


Fig.3 (a) Original image; (b) Binary image; (c) Dilated image; (d) LED contour confirmation

Step 4. However, during the binarization of the image, x_{max} and x_{min} may not be the true left and right LED array contour lines, but more likely the center point of the left-most (right-most) LED, so this will lead to over-fitting of the LED array contour lines, as shown in Fig.4(a). To ensure that the spot is in the center, the LED array contour needs to be compensated. Assuming that the LED spot width (height) and the horizontal (vertical) spacing between adjacent LED spots are the same, the LED array profile needs to be expanded by half the LED spot width (height) in each of the two vertical (horizontal) directions, as shown in Fig.4(b).



(a) (b)

Fig.4 (a) Without compensation; (b) With compensation

The compensation formula is shown as

$$\begin{cases} x_{\text{panel-min}} = x_{\min} - \dfrac{x_{\max} - x_{\min}}{(3+3)\times 2} \\ x_{\text{panel-max}} = x_{\max} + \dfrac{x_{\max} - x_{\min}}{(3+3)\times 2} \\ y_{\text{panel-min}} = y_{\min} - \dfrac{y_{\max} - y_{\min}}{(3+3)\times 2} \\ y_{\text{panel-max}} = y_{\max} + \dfrac{y_{\max} - y_{\min}}{(3+3)\times 2} \end{cases}, \tag{5}$$

where x_{min}, x_{max}, y_{min}, y_{max} denote the LED array contour coordinates, while $x_{panel\text{-}min}$, $x_{panel\text{-}max}$, $y_{panel\text{-}min}$, $y_{panel\text{-}max}$ denote the LED array contour coordinates after adding compensation.

Step 5. According to the compensated LED array contour coordinates, calculate each LED contour coordinates, and the calculation formula is shown as

$$\begin{cases} x_{\text{led, min},i} = x_{\text{panel-min}} + \left(\dfrac{x_{\text{panel-max}} - x_{\text{panel-min}}}{4}\right)\times(i-1) \\ x_{\text{led, max},i} = x_{\text{panel-min}} + \left(\dfrac{x_{\text{panel-max}} - x_{\text{panel-min}}}{4}\right)\times i \\ y_{\text{led, min},j} = y_{\text{panel-min}} + \left(\dfrac{y_{\text{panel-max}} - y_{\text{panel-min}}}{4}\right)\times(j-1) \\ y_{\text{led, max},j} = y_{\text{panel-min}} + \left(\dfrac{y_{\text{panel-max}} - y_{\text{panel-min}}}{4}\right)\times j \end{cases}, \tag{6}$$

where $x_{led,min,i}$, $x_{led,max,i}$, $y_{led,min,j}$ and $y_{led,max,j}$ are the contour coordinates of the LED area in row i ($1 \le i \le 4$) and column j ($1 \le j \le 4$). The coordinates of the center of the LED area in row i and column j ($x_{led\text{-}c,i}$, $y_{led\text{-}c,j}$) are shown as

$$\begin{cases} x_{\text{led-c},i} = \dfrac{x_{\text{led,min},i} + x_{\text{led,max},i}}{2} \\ y_{\text{led-c},j} = \dfrac{y_{\text{led,min},j} + y_{\text{led,max},j}}{2} \end{cases}. \tag{7}$$

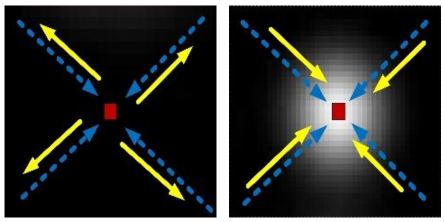
Step 6. After determining the contour coordinates and center coordinates of the LED area, two grayscale features, the *AGR* and the *GRI* are calculated for each LED area.

The *AGR* represents the ratio of the average grayscale value of the pixels in a single LED area to the average grayscale value of the pixels in the LED array area. The *AGR* of the *n*th LED region is shown as

$$AGR_n = \frac{G_{\text{led-}n}}{G_{\text{panel}}}, \tag{8}$$

where $G_{led\text{-}n}$ denotes the average gray value of the nth LED area, and G_{panel} denotes the average gray value of the LED array area.

The *GRI* represents the dot product of the direction vector of each pixel pointing to the center and the gradient vector of grayscale values in each LED region. The LED with state 0 has the gradient vector opposite to the direction vector and the *GRI* is negative, as shown in Fig.5(a). The opposite is true for the LED with state 1, as shown in Fig.5(b).



(a) (b)

Fig.5 (a) LED with status "0"; (b) LED with status "1"

For any point (x, y) in the LED region, its direction vector with respect to the center of the region is shown as

$$\boldsymbol{r}_n(x,y) = (x_{\text{led-c},i} - x, y_{\text{led-c},j} - y). \tag{9}$$

Let the pixel value at the point (x, y) be $f(x, y)$, then the gradient vector of the gray value change is shown as

$$\nabla f_n(x,y) = \begin{bmatrix} f(x+1) - f(x-1) \\ f(y+1) - f(y-1) \end{bmatrix}. \tag{10}$$

Therefore, the *GRI* of the *n*th LED region is shown as

$$GRI_n = \frac{\boldsymbol{r}_n(x,y) . \nabla f_n(x,y)}{|\boldsymbol{r}_n(x,y)| \times W_{\text{L}} \times H_{\text{L}}}, \tag{11}$$

where $|\boldsymbol{r}_n(x,y)|$ is the modal length of vector $\boldsymbol{r}_n(x,y)$, W_L is the width of the LED region, and H_L is the height of the LED region, n=1, 2,...,4×4.

Step 7. Fisher linear discrimination: The actual state of the LEDs in the image is captured by the receiver through pre-set transmission data from the sender. We acquire multiple images containing LED arrays and extract the *AGR* and *GRI* data features of all LEDs to construct a feature vector training set $\boldsymbol{S}=\{\boldsymbol{x}\,(x_{AGR}, x_{GRI})\}$. The target value corresponding to each object in the training set is divided into two classes of LEDs, S_0 (status=off), S_1 (status=on), respectively.

Taking an arbitrary two-dimensional vector $\boldsymbol{w}$ as shown in Fig.6, the projection of each object $\boldsymbol{x}$ in the training set is shown as

$$p(\boldsymbol{x}) = \boldsymbol{w} \bullet \boldsymbol{x}. \quad (12)$$

Let $\boldsymbol{m}_i$ be the mean of objects of class i (i=0, 1) and $\boldsymbol{S}_i$ be the set of all objects belonging to class i in the training set. The mean (μ_0, μ_1) and variance (σ_0, σ_1) of the projection of the two classes of objects on $\boldsymbol{w}$ are calculated separately, and the formula is given as

$$\begin{cases} \mu_i = \boldsymbol{w} \bullet \boldsymbol{m}_i \\ \sigma_i = \sum_{x \in S_i} (\boldsymbol{w} \bullet \boldsymbol{x} - \mu_i)^2 \end{cases}. \quad (13)$$

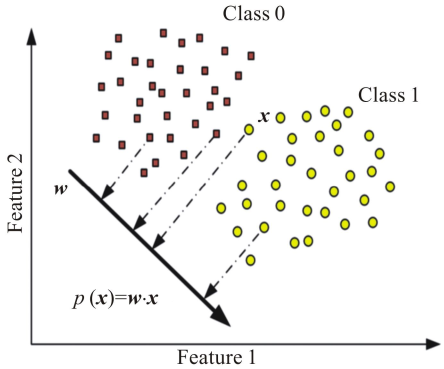


Fig.6 Projection of the object *x* onto the vector *w*

Fisher's linear discriminant method is used to find a discriminant vector $\boldsymbol{w}$ that maximizes the variance of object projections between classes and minimizes the variance of object projections within classes.

Then the discriminant vector is calculated as

$$G(w)_{\max} = \frac{(\mu_0 - \mu_1)^2}{\sigma_0 + \sigma_1}, \quad (14)$$

where the discriminant vector $\boldsymbol{w}$ is shown as

$$\boldsymbol{w} = \boldsymbol{S}_w^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_0), \quad (15)$$

where $\boldsymbol{S}_w$ is the intra-class covariance matrix, shown as

$$\boldsymbol{S}_w = \boldsymbol{C}_0 + \boldsymbol{C}_1, \quad (16)$$

where $\boldsymbol{C}_i$ is the covariance matrix of all LED feature vectors in the set $\boldsymbol{S}_i$, shown as

$$\boldsymbol{C}_i = \begin{bmatrix} \sigma_{x_{AGR}, x_{AGR}} & \sigma_{x_{AGR}, x_{GRI}} \\ \sigma_{x_{GRI}, x_{AGR}} & \sigma_{x_{GRI}, x_{GRI}} \end{bmatrix}, \quad (17)$$

where σ is the covariance between the two features of all LEDs in the $\boldsymbol{S}_i$ set.

Step 8. Once the discriminant vector $\boldsymbol{w}$ is found, the LED state S can be identified as

$$S = \begin{cases} 0 & \boldsymbol{w} \bullet \boldsymbol{\mu} < T_{\text{thres}} \\ 1 & \boldsymbol{w} \bullet \boldsymbol{\mu} \geq T_{\text{thres}} \end{cases}, \quad (18)$$

where $\boldsymbol{\mu}$ is the feature vector extracted from the input frame and T_{thres} is the average of all LED feature vectors, shown as

$$T_{\text{thres}} = \boldsymbol{w} \bullet \frac{\boldsymbol{m}_0 + \boldsymbol{m}_1}{2}. \quad (19)$$

The experimental distance is set from 10 m to 80 m, and 1 000 pictures are taken for the LED array at 10 m intervals, each picture contains 16 LEDs, so 128 000 LEDs need to be decoded for different communication distances. In this experiment, the decoding performance of the algorithm is analyzed by the number of error LEDs.

The experimental scenario is shown in Fig.7. The experimental environment is set up as two scenarios: an ideal environment without interference light sources (lights off) and a communication scene with interference light sources (lights on). The former is to test the performance of fixed threshold and adaptive threshold decoding algorithm, and the latter is to test the performance of multi-feature decoding algorithm.

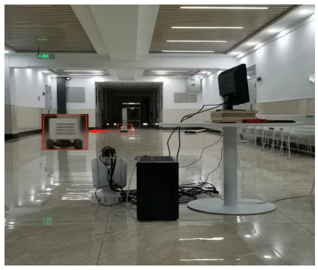


Fig.7 Experimental scenario

Case 1. As shown in Fig.8(a), with the adaptive threshold algorithm, the system decoding performance is improved at different distances, and the distances remain the same. The bit error ratios (*BERs*) of the adaptive thresholding algorithm are all lower than the fixed thresholds. As seen in Fig.8(b), the total *BER* (total *BER* for each fixed threshold) of the decoding algorithm is also reduced by the adaptive threshold.

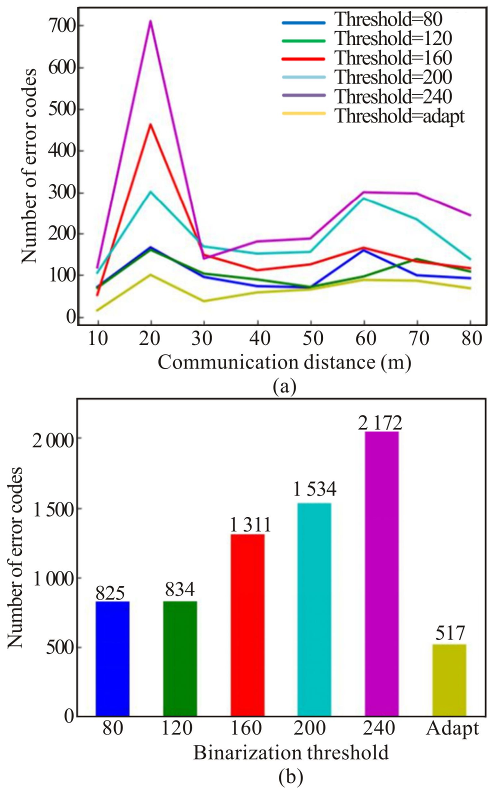


Fig.8 (a) *BER* for communication distance and binarization threshold; (b) *BER* for binarization threshold

Case 2. Using LED array detection algorithm to detect a total of captured 8 000 images, all can accurately obtain the LED array outline information, and the detection rate can reach 100% correctly.

As shown in Fig.9, the red color indicates the LED grayscale feature value with state "1" and the green color indicates the LED grayscale feature value with state "0". As shown in Fig.9(a) and Fig.9(b), the *AGR* or *GRI* grayscale features alone can differentiate the states of most LEDs. As seen in Fig.9(c), when the two grayscale features are combined, the LED states are better distinguished.

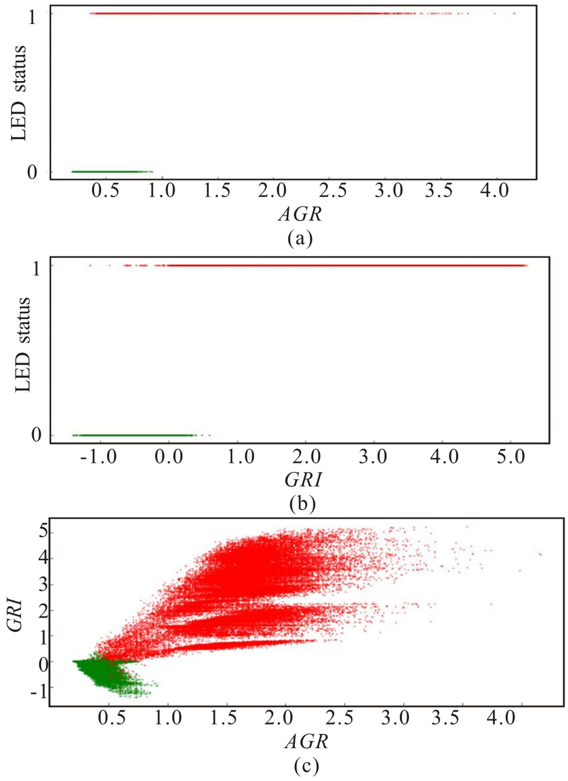


Fig.9 (a) LED *AGR* distribution; (b) LED *GRI* distribution; (c) *GRI* versus *AGR* distribution

As can be seen in Fig.10, the decoding performance of the system is further improved by the multi-feature decoding algorithm. The *BER* is reduced from 1 335.2 (average total *BER* for each fixed threshold) to 72, and the *BER* is reduced from 1×10^{-2} with the fixed threshold to 5×10^{-4}, which is a reduction of two orders of magnitude.

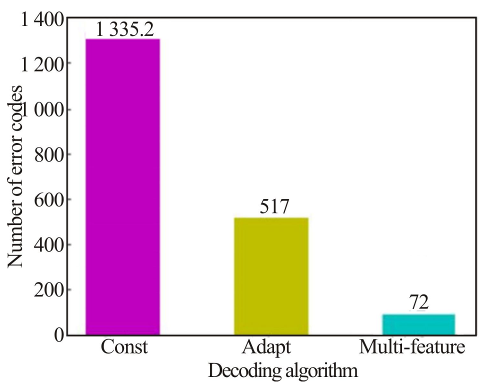


Fig.10 *BER* analysis

This paper analyzes the advantages and disadvantages of the decoding algorithm based on the directional projection method, then optimizes the algorithm by adaptive thresholding in an ideal environment to reduce the *BER* from 1×10^{-2} to 4×10^{-3}, and finally proposes a multi-feature decoding algorithm based on the optimized algorithm for the problems of low decoding performance in complex environments. The experimental results show that this algorithm reduces the *BER* from 4×10^{-3} to 5×10^{-4}, which improves the robustness of OCC system in complex environment.

Statements and Declarations

The authors declare that there are no conflicts of interest related to this article.

References

[1] SHAABAN K, SHAMIM M H M, ABDUR-ROUF K. Visible light communication for intelligent transportation systems：a review of the latest technologies[J]. Journal of traffic and transportation engineering-English edition, 2021, 8(4)：483-492.

[2] TAMANGL D, WOOK K B. Optical camera communication for vehicular applications：a survey[J]. IEEE transactions on smart processing & computing, 2021,10(2)：136-145.

[3] CAHYADI W A, CHUNG Y H, GHASSEMLOOY Z, et al. Optical camera communications ： principles, modulations, potential and challenges[J]. Electronics, 2020, 9(9).

[4] LIU A Q, SHI W X, OUYANG M, et al. Characterization of optical camera communication based on a comprehensive system model[J]. IEEE journal of lightwave technology, 2022, 40(18)：6087-6100.

[5] HASAN M K, ALI M O, RAHMAN M H, et al. Optical camera communication in vehicular applications：a review[J]. IEEE transactions on intelligent transportation systems, 2021, 23(7)：6260-6281.

[6] WU H N, CHEN Y C, XUE G T, et al. OnionCode：enabling multi-priority coding in LED-based optical camera communications[C]//2022 IEEE Conference on Computer Communications, May 2-5, 2022, London, UK. New York：IEEE, 2022：21817468.

[7] SHAHJALAL M, HASAN M K, ISLAM M M, et al. A two-stage power allocation-based NOMA architecture for optical camera communication[J]. IEEE systems journal, 2021,15(3)：4421-4430.

[8] ESFAHANI N M, GHOLAMI A, KORDAVANI N S, et al. The impact of camera parameters on the performance of V2V optical camera communications[C]//2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), July 20-22, 2020, Porto, Portugal. New York：IEEE, 2020：20165374.

[9] MATUS V, GUERRA V, JURADO-VERDU C, et al. Experimental evaluation of an analog gain optimization algorithm in optical camera communications[C]//2020

12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), July 20-22, 2020, Porto, Portugal. New York：IEEE, 2020：20165456.
[10] DO T H, YOO M. The necessity of LED to ambient light ratio optimization for vehicular optical camera communication[J]. Sensors, 2020, 20(1).
[11] NGUYEN D T, PARK Y. Performance enhancement of optical camera communication system using optical camera communication coding and region-of-interest detection[J]. IET optoelectronics, 2021,15(6)：255-263.
[12] CHOW C W, LIU Y, YEH C H, et al. Display light panel and rolling shutter image sensor based optical camera communication (OCC) using frame-averaging background removal and neural network[J]. Journal of lightwave technology, 2021, 39(13)：4360-4366.
[13] JEONG S, MIN J, PARK Y. Indoor positioning using deep-learning-based pedestrian dead reckoning and optical camera communication[J]. IEEE access, 2021, 9：133725-133734.
[14] SUN Y F, SHI W X, SHAO X R, et al. Implementation and decoding method of OCC system based on MIMO[J]. Optoelectronics letters, 2022, 18(1)：18-23.
[15] DO T H, YOO M. A multi-feature LED bit detection algorithm in vehicular optical camera communication[J]. IEEE access, 2019, 7：95797-95811.
[16] OTSU N. Threshold selection method from gray-level histograms[J]. IEEE transactions on systems man and cyberneticsjavascript:void(0), 1979, 9(1)：62-66.