Maximum likelihood decoding based on pseudo-captured image templates for image sensor communication

Shintaro Arai1a), Haruna Matsushita2, Yuki Ohira3, Tomohiro Yendo3, Di He4, and Takaya Yamazato5

1 Okayama University of Science, 1-1 Ridai-cho, Kita-ku, Okayama 700–0005, Japan
2 Kagawa University, 2217-20 Hayashi-cho, Takamatsu 761–0396, Japan
3 Nagaoka University of Technology, 1603–1 Kamitomioka, Nagaoka 940–2137, Japan
4 Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China
5 Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464–8603, Japan

a) arai@ee.ous.ac.jp

Received July 27, 2018; Revised November 20, 2018; Published April 1, 2019

Abstract: This paper focuses on an image sensor communication system that uses an LED as the transmitter and a high-speed image sensor (camera) as the receiver. Communication in this scheme depends on the quality of images transmitted from the LED to the sensor. If the image becomes unfocused on the way to the receiver, the LED luminance that make up the signal cannot be detected, so the receiver cannot demodulate the signal data. To overcome this problem, this study proposes a novel demodulation scheme to recover data from a degraded image, based on a maximum likelihood decoding (MLD) algorithm. The proposed method creates template images that imitate all possible blinking patterns produced by the LED transmitter, and then calculates the Euclidean distances between pixels in the captured image and the pseudo images for all possible blinking patterns. Finally, the algorithm chooses the image template with the smallest Euclidian distance from the received signal as the recovered data. Though an exhaustive set of image templates must be prepared for the proposed MLD, the number of templates depends on the number of LEDs on the transmitter. Thus, the computational complexity of this method increases as the number of transmitter LEDs increases. To reduce the computational complexity of the proposed MLD algorithm, the binary differential evolution (BDE) algorithm is used, which is a swarm intelligence technique. Computer simulations are used to evaluate the BDE algorithm’s usefulness for reducing computational complexity and improving the BER of the communication system.

Key Words: LED, visible light communication (VLC), image sensor communication (ISC), maximum likelihood decoding (MLD), swarm intelligence, binary differential evolution (BDE)
1. Introduction

Visible light communication (VLC) has become an important trend in the field of optical wireless communication systems as light-emitting diodes (LED) have become more popular and inexpensive [1–5]. The brightness of LEDs can be easily controlled at high speeds, which makes them useful for VLC. In addition, LEDs have the advantages of long life, energy efficiency, ecological friendliness, low voltage, and high visibility. Herein, a VLC system that communicates from infrastructure-to-vehicles using an LED traffic light to transmit and an in-vehicle high-speed image sensor to receive is considered [6–12], as shown in Fig. 1. This VLC system is referred to as an image sensor communication (ISC) system [13–16]. This system allows the exchange of safety information between roadway infrastructure and vehicles. The advantages of using the high-speed-image sensor include easy recognition of the LED traffic light and an estimation of its position. The image sensor simultaneously captures multiple LED transmitters in a single image, and this captured image is used to distinguish the LED luminance values of each transmitter. The image sensor can also recognize whether the signals came from the transmitter or noise sources, such as the sun and other ambient lights.

In an ISC system, the image-processing modules at the receiver end is vital for precise data recovery. ISC is limited by the focus of the transmitted image over distance, as shown in Fig. 1. Defocusing blurs the captured images and prevents the image-processor from distinguishing between different patterns of LED illumination. Here, we consider on-off keying (OOK) as the modulation scheme used in the transmitter. To perform parallel data transmissions, we allocate a bit to each LED on the transmitter array and send “1” or “0” with the LED turned “on” or “off”. The receiver recovers data by determining whether each has LED blinked. If defocusing occurs, the received LED pixel values will be conflated with neighboring LED bits. Consequently, if the captured image appears blurred in

![Image Sensor Communication for Infrastructure-to-Vehicle Communication for ITS](image)

*Fig. 1. Image sensor communication for infrastructure-to-vehicle communication for ITS. The image captured by the image sensor degrades in quality with increasing communication distance. When the captured image appears blurred in the sensor, the receiver can no longer detect the correct luminance values to demodulate the data.*
the image sensor, the receiver cannot detect the correct luminance values to demodulate the signal.

To overcome this problem, the previous studies have proposed a variety of coding and decoding schemes: hierarchical coding [17, 18], overlay coding [19], Alamouti-type coding [20], layered space-time coding [21, 22]. Hierarchical coding and the overlay coding focus on the spatial-frequency components of the captured images and use their characteristics for modulation and demodulation. Although both coding schemes achieve the successful data transmission at the appropriate distance, they sacrifice the transmission data rate to do so. Alamouti-type coding and the layered space-time coding techniques modulate the data with the LED blinking in the time domain, and then recover the signal using successively captured images. Both methods achieve a longer communication range with limited pixel resolution of the image sensor, though the receiver needs to know the corresponding positions of LEDs on the transmitter.

This present study proposes a novel demodulation scheme that recovers data from degraded visual signals without a reduction in the transmission data rate. Our method is based on the maximum likelihood decoding (MLD) algorithm. As described above, each luminance value in the received image is expressed as a sum of itself and the diffused luminances from neighboring LEDs, indicating that the convolution operation is relevant. If the diffuse light from all the LEDs can be approximated as an optical channel matrix and the transmitter consists of an $M \times N$ LED matrix, each LED’s pixel value can be expressed as an equation. Figure 2 shows a concept of our proposed MLD algorithm. The proposed method computationally prepares templates of captured images that imitate all possible blinking patterns of the LED transmitter. We refer to this image template as a pseudo-captured image in this study. After creating pseudo-captured images, the demodulation algorithm calculates the Euclidean distances between pixels in a captured image and all of the pseudo-captured images. The pseudo-captured image that is most like the captured image is chosen as the most-likely pattern of data transmitted by the LED array.

The original MLD algorithm exhibits a lower bit error rate (BER) than other demodulation meth-

![Fig. 2. Concept of proposed MLD algorithm. The proposed method creates pseudo-captured images that imitate all possible blinking patterns of the LED transmitter calculates the Euclidean distances between pixels in a captured image and all of the pseudo-captured images. The pseudo-captured image that is most like the captured image is chosen as the most-likely pattern of data transmitted by the LED array.](image-url)
ods; however, the computational complexity of this method is very high because it depends on the number of candidate solutions. In the case of this study, the number of candidates is equivalent to the number of LEDs included in the transmitter. To reduce the computational complexity of our MLD algorithm, this paper also proposes a novel MLD process that uses the binary differential evolution (BDE) algorithm, which is a swarm intelligence technique [23–26]. Computer simulations are then used to evaluate the BER reduction gained with this step.

The rest of this paper is organized as follows. The system model for ISC is described in Section 2. The proposed MLD based on pseudo-captured images is presented in Section 3, and the ability to reduce computational complexity using BDE is explained in Section 4. The simulation results are discussed in Section 5, and Section 6 summarized the study’s findings.

2. System model of image sensor communication

2.1 Transmitter

Figure 3 shows a block diagram of ISC system. The transmitter comprises an encoder, a mapper, and an LED array arranged in a matrix of $M \times N$ LEDs. Parallel data transmission is performed by modulating different transmitted data via on-off-keying (OOK: On = 1, Off = 0) with each LED. The transmitter generates nonnegative pulses with bit duration $T_b$, and in this case, the data rate of each LED is $R_b = 1/T_b$. Thus, the total transmission rate becomes $(M \times N)R_b$. Let LED$_{m,n}$ be the LED in row $m$ and column $n$. When the LED transmitter blinks according to the $i$-th blinking pattern, the luminance of LED$_{m,n}$ at time $t$ is expressed as

$$S_{m,n}(t) = \sum_i s_{m,n,i} \cdot A \cdot g_p(t - (i - 1)T_b), \quad (1)$$

where $s_{m,n,i}$ is the coefficient for determining the luminance of LED$_{m,n}$, $A$ is the peak luminance, and $g_p(t)$ is the following rectangular pulse function:

$$g_p(t) = \begin{cases} 1 & (0 \leq t < T_b), \\ 0 & \text{otherwise}. \end{cases} \quad (2)$$

2.2 Receiver

The receiver is composed of a high-speed image sensor, an image processing unit, and a decoder. The transmitted optical signals pass through the spatial optical channel before they arrive at the receiver. The receiver uses the image sensor to capture these optical signals at a high frame rate and converts them to electrical signals. The image-processing unit detects the transmitter position from the captured image and extracts each LED luminance value as a pixel of an image. The extracted luminance values are then used by the receiver to recover the data.

Next, the extracted LED luminance value for each pixel is defined. Defocusing diffuses the light transmitted from each LED. If the blur can be approximated as an optical channel matrix $G$, and the transmitter comprises an $M \times N$ LED matrix, each received luminance value $R'_{m,n}(t)$ at time $t$, corresponding to LED$_{m,n}$ on the captured image can be expressed as
where $n_{m,m}(t)$ is an ambient light noise, which is added to the luminance value of $\text{LED}_{m,n}$, at time $t$, $G_{p,q}$ is the convolution kernel of an optical channel matrix, and $p$ and $q$ are the vertical and horizontal components, respectively, of the distance from the origin. The optical channel matrix based on the 2D Gaussian filter with 5 x 5 kernel $(x = y = 2)$ is defined as

$$G_{p,q} = \frac{1}{2\pi\sigma_g^2} \exp\left(-\frac{p^2 + q^2}{2\sigma_g^2}\right),$$

(4)

where $\sigma_g^2(>0)$ is the variance of the filter. This study approximates the channel matrix by 2D Gaussian filter [17]. As described in Sec. 1, ISC is limited by the focus of the transmitted image over distance. If defocusing occurs, the received LED pixel values will be conflated with neighboring LED bits. This phenomenon is restated as a loss of high frequency component of the spatial frequency of images. We consider that it is possible to model the channel characteristics of ISC as a low pass filter (LPF) that a cutoff frequency varies depending on the distance. In [17], the channel characteristic was approximated by 2D Gaussian filter that is one of 2D LPF. Thus, we can easily express the cutoff frequency of the channel by parameterizing the variance of 2D Gaussian filter.

Here, let us assume that the image sensor achieves a complete synchronization between its exposure time and an LED blinking duration of the transmitter. The $i$-th extracted luminance value $R_{m,n,i}$, corresponding to LED$_{m,n}$ is expressed as

$$R_{m,n,i} = c \int_{(i-1)T_b}^{iT_b} R_m'_{m,n}(t) \cdot f_e(t) \, dt,$$

(5)

where $c$ is the photoelectric conversion efficiency, $f_e(t)$ is a function of the image sensor’s exposure time:

$$f_e(t) = \sum_i g_p(t - (i-1)T_b).$$

### 3. MLD based on pseudo-captured image templates

This section presents the proposed MLD algorithm. As described above, each LED pixel value on the transmitter is expressed as a sum of itself and the diffused light from neighboring LEDs (Eq. (3)). Based on Eq. (3), we create pseudo-captured images that are templates of blinking patterns of the LED transmitter in the following equation.

$$Z_k = \left[ Z_{1,1,k} \cdots Z_{m,n,k} \cdots Z_{M,N,k} \right]^T = GS,$$

(7)

where $Z_k$ is the pixel-value vector of the $k$-th pseudo captured image ($k: 1, 2, \cdots, 2^{M\times N}$), $Z_{m,n,k}$ expresses the luminance value of LED$_{m,n}$ of the $k$-th pseudo-captured image, and $S$ is the $[S_{(1,1)} \cdots S_{M,N}]^T$ transmitted symbol vector. In this system, the blinking pattern of the $M \times N$ LED-array transmitter is determined by the $2^{M\times N}$ bit sequence. Thus, the transmitted LED blinking pattern must be included in the created $2^{M\times N}$ patterns, even if the captured image blur is significant. This fact allows us to find the appropriate data pattern from the blurred images by considering which data pattern is most similar to the blinking pattern. Thus, we create $2^{M\times N}$ pseudo-captured images to recover data correctly.

After creating pseudo-captured images, the Euclidean distance is calculated as follows:

$$D_k = \sqrt{\sum_{m=1}^{M} \sum_{n=1}^{N} (R_{m,n,i} - Z_{m,n,k})^2},$$

(8)

where $D_k$ is the Euclidean distance corresponding to the $k$-th pseudo captured image. When the data pattern of the $k$-th pseudo-captured image is similar to that of the transmitted image, the calculated
value becomes the smallest value among all the calculations. Therefore, the algorithm finds the smallest \( D_k \) among all the calculated distances and chooses the \( k \)-th blinking pattern corresponding to the smallest \( D_k \) as the recovered data.

To create pseudo-captured images, the receiver has to estimate the optical channel matrix \( G \). Here, we assume that the parameters of \( G \) can be estimated in the receiver. To estimate \( G \) in the actual ISC, in our previous study we independently blinked each LED on the transmitter as a preamble of data [9]. Since each LED luminance value is captured by several pixels on the image sensor, therefore each of their values can be extracted in the receiver and they can be used as \( G \).

4. Computational complexity reduction using BDE for MLD

As described above, the proposed MLD creates \( 2^{M \times N} \) pseudo-captured images to recover from a blurry image reaching the image-sensor receiver. The number of pseudo-captured images depends on the number of LEDs on the transmitter array. With increasing the number of LEDs, the number of template images increases exponentially. This indicates that the computational complexity of the proposed MLD is determined by the number of LEDs. To overcome this problem, this section proposes a novel algorithm to reduce computational complexity using BDE.

Figure 4 shows a flowchart for the proposed computational complexity reduction algorithm. BDE, a modified version of DE [27], is a population-based algorithm that operates in binary search spaces [23–26]. There are two main studies [23, 24] in literature, which modify the DE for application to binary optimization problems. As a result of the experiments performed in [25, 26], the BDE in [24] has been used in this study. BDE includes four main steps: initialization, mutation, recombination, and selection. Each individual in the population moves through these steps, and its solution vector (or target vector) is updated until an acceptable solution is found or until a predefined number of maximum iterations are reached.

The target vector for each individual in the population is expressed as

\[
X_{j,g} = (x_{j1}, x_{j2}, \ldots, x_{jn_B}, \ldots, x_{j(N_B)}),
\]

where \( j \) is the index of individual \( (j = 1, 2, \ldots, J_B) \), \( J_B \) is the number of individuals, \( g \) is the index of the generation, and \( N_B \) is the number of dimensions in the vector. In this study, \( N_B \) is equal to the total number of LEDs (i.e., \( N_B = M \times N \)). In BDE, each element of \( X_{j,g} \), \( x_{jn_B} \) is either 1 or 0. Namely, \( x_{jn_B} \) corresponds to the binary state of each LED. Since the position of the LEDs is

---

![Fig. 4. Flowchart of proposed computational complexity reduction algorithm using BDE.](image-url)
expressed in a matrix, Eq. (9) can be rearranged as
\[
X_{j,g} = (x_{j(1,1)}, x_{j(1,2)}, \cdots, x_{j(m.n)}, \cdots, x_{j(M,N)}).
\] (10)

The present BDE algorithm calculates \( Z'_j' \), which is the pixel-value vector of the \( j \)-th pseudo-captured image, and evaluates \( j \)-th target vector using the following objective function,
\[
f(X_{j,g}) = \sqrt{\sum_{m=1}^{M} \sum_{n=1}^{N} (R_{m,n,i} - Z'_{m,n,j})^2},
\] (11)
where \( Z'_{m,n,j}' \), which is an element of \( Z'_{j}' \), expresses the luminance value of LED of the \( j \)-th pseudo-captured image. The Euclidean distance between the pixels in the original and the pseudo-captured images is calculated from \( X_{j,g} \). Thus, the objective of the proposed method is to minimize \( f(X_{j,g}) \).

The proposed method is based on the following steps (see Fig. 4 and Appendix A). Note that all the randomness used in the proposed algorithm are generated based on uniform random numbers.

**Step 1 (Initialization):** This step randomly chooses initial binary values for each \( X_{j,g} \) in the population and calculates each initial \( f(X_{j,g}) \).

**Step 2 (Mutation):** Three different individuals \( (X_{e1,g}, X_{e2,g}, X_{e3,g}) \) are chosen from the population to update \( X_{j,g} \) (\( j \neq e1 \neq e2 \neq e3 \)) and the number of dimensions for mutating \( X_{e1,g} \), \( N_{\text{mut}} \) are determined based on the Hamming distance \( d_h \) between \( X_{e2,g} \) and \( X_{e3,g} \) as follows.
\[
N_{\text{mut}} = \begin{cases} (\text{int})d' + 1 & \text{if } \text{rand}_1 < d' - (\text{int})d', \\ (\text{int})d' & \text{otherwise,} \end{cases}
\] (12)
where \( d' = F_w \times d_h \), \( F_w \) is the scaling parameter \((0 < F_w \leq 1)\), and \( \text{rand}_1 \) is a random floating-point number uniformly distributed in \([0, 1]\). Finally \( N_{\text{mut}} \) elements of \( X_{e1,g} \) are chosen at random to create a mutant vector \( V_{j,g} (= (v_{j(1,1)}, \cdots, v_{j(M,N)}) \) by reversing the chosen elements.

**Step 3 (Recombination):** A new vector, called the trial vector, \( U_{j,g} (= (u_{j(1,1)}, \cdots, u_{j(M,N)}) \), is created according to \( X_{j,g} \) and \( V_{j,g} \) as follows.
\[
u_{j(m,n)} = \begin{cases} v_{j(m,n)} & \text{if } \text{rand}_2 < Cr \text{ or } \\
x_{j(m,n)} & \text{else,} \end{cases}
\] (13)
where \( \text{rand}_2 \) is a random floating-point number uniformly distributed in \([0, 1]\), and \( (m_{\text{rand}}, n_{\text{rand}}) \) is a row and column index randomly chosen from the matrix of \( M \times N \) LEDs to mutate at least one of the elements. Each element of \( U_{j,g} \) is taken from \( X_{j,g} \) if \( \text{rand}_2 \) is greater than a predefined parameter \( Cr \) \((0 < Cr \leq 1)\); otherwise, the element is removed from \( V_{j,g} \).

**Step 4 (Selection):** Fig. 5 displays a process example of selection step. \( U_{j,g} \) is evaluated and \( X_{j,g+1} \) for the next generation is created using the following criterion.
\[
X_{j,g+1} = \begin{cases} U_{j,g} & \text{if } f(U_{j,g}) < f(X_{j,g}), \\ X_{j,g} & \text{otherwise.} \end{cases}
\] (14)
In addition, the counter variable \( C_B \) is increased when \( f(X_{j,g+1}) \) is equal to \( X_{\text{best}} \), which is the best solution in the process of update.

**Step 5: Steps 2–4** are iterated for all \( X_{j,g} \) in the population.

**Step 6:** \( g \) is increased \((g = g+1)\). To avoid being trapped by a local optimum, a reinitialization step is included. If \( C_B \) is larger than \( J_B \times J_{\text{rate}} \), the iterations are stopped and the algorithm returns to Step.
for the reinitialization of all individuals. $J_{\text{rate}}$ is an arbitrary weight coefficient ($0.0 < J_{\text{rate}} \leq 1.0$). Otherwise, the counter variable $n_I$ is incremented, and the iterations are continued until $n_I$ reaches 100. If this condition is satisfied (i.e. $n_I = 100$), the iterations are stopped and the algorithm returns to Step 1. When the iterations are stopped, the counter variable $ReI$ is incremented. Note that $X_{\text{best}}$ is stored even if the population is reinitialized.

**Step 7 (Local Search):** When $ReI$ reaches $ReI_{\text{max}}$, a local search for $X_{\text{best}}$ is performed to further avoid local optima. Figure 6 shows a process example of local search step. When $x_{j(m,n)}$ of $X_{\text{best}}$ is not coincident with the data transmitted from LED$_{m,n}$, anticoincidence between the three nearest elements ($x_{j(m+1,n)}$, $x_{j(m,n+1)}$, $x_{j(m+1,n+1)}$) and the transmitted elements frequently occurs. Thus, herein, $\tilde{X}_{\text{best}} (= (\tilde{x}_{j(1,1)}, \cdots, \tilde{x}_{j(M,N)}))$ from $X_{\text{best}}$ based on Eq. (15) is first created, and $f(\tilde{X}_{\text{best}})$ is calculated.

$$\tilde{x}_{j(m,n)} = \begin{cases} l_1 & \text{if } (m,n) = (m_l,n_l), \\ l_2 & \text{if } (m,n) = (m_l + 1,n_l), \\ l_3 & \text{if } (m,n) = (m_l,n_l + 1), \\ l_4 & \text{if } (m,n) = (m_l + 1,n_l + 1), \\ x_{j(m,n)} & \text{otherwise,} \end{cases}$$

(15)

where $l_1$, $l_2$, $l_3$, and $l_4$ are binary values, and $(m_l,n_l)$ is a row and column chosen from the $M \times N$ LEDs for the local search. If $f(\tilde{X}_{\text{best}}) < f(X_{\text{best}})$, elements of $\tilde{X}_{\text{best}}$ are adopted as the new $X_{\text{best}}$. This search tests all the possible patterns. In this case, the number of trials is $2^4 = 16$. This step is performed for all elements of $X_{\text{best}}$ in the ranges of ($1 \leq m_l \leq M - 1$) and ($1 \leq n_l \leq N - 1$).
Therefore, the total number of searches $N_L$ becomes $(M - 1)(N - 1) \times 2^4$.

**Step 8 (Demodulation):** The elements of $X_{\text{best}}$ are extracted as the decoded signal data.

### 5. Simulation results

This section explains the computer simulations we used to test this algorithm. In the test transmitter, the LED array is composed of $4 \times 4$ LEDs (i.e., $M = N = 4$). We assume that all the LEDs are uniform in their maximum luminance. No gaps appear between the neighboring LEDs in the image, as shown in Fig. 7. We use a 2D Gaussian filter with $5 \times 5$ kernel to model the optical-channel matrix, and additive white Gaussian noise (AWGN) to model ambient light noise [28]. Figure 7 also shows the example of captured image defocused by the 2D Gaussian filter when $\sigma_g^2 = 0.7$. Moreover, the simulations assume that the parameters of the optical channel matrix can be estimated in the receiver. The receiver can create pseudo-captured images using the same channel matrix as the original captured images. After the filtering process, we add AWGN with a power-spectrum density of $N_0/2$ to each image pixel. The BDE parameters are summarized in Table I. In this study, the parameters of $F_w$ and $Cr$ were chosen based on [24].

We conduct three simulations to test the algorithms discussed above. First, we vary $\sigma_g^2$ of 2D Gaussian filter and calculate the BER performance with $E_b/N_0$ held constant. Second, we replace $\sigma_g^2$ in the first simulation with $E_b/N_0$ to calculate the BER performance with fixed $\sigma_g^2$ as $E_b/N_0$ is varied. Third, we investigate the robustness of the proposed MLD with/without BDE when the estimation error of the channel matrix occurs in the receiver. We purposely give a mismatch of $\sigma_g^2$ between the actual channel matrix and estimated one, and calculate the BER performance with fixed $E_b/N_0$.

All simulations are iterated 10,000 times for every value of $\sigma_g^2$ or $E_b/N_0$ to measure the BER. In addition, we also count the total number ($N_E$) of calculations of Euclidean distance performed in the signal-processing unit to evaluate the BDE algorithm’s capability for complexity reduction.

First, we compare the performance of the proposed MLD with a simple threshold-processing algorithm. Figure 8 shows the BERs of the proposed and conventional methods plotted against $\sigma_g^2$ when $E_b/N_0 = 5$ and 10 dB. The BER performance of the proposed MLD algorithm is significantly improved compared with the conventional method. We can confirm that the proposed MLD achieves error-free transmission until $\sigma_g^2 = 0.6$ ($E_b/N_0 = 5$ dB) or 0.8 ($E_b/N_0 = 10$ dB). Conversely, the performance of the conventional method rapidly degrades when $\sigma_g^2$ exceeds 0.2. The large value of $\sigma_g^2$ indicates that the image quality is seriously degraded by the 2D Gaussian filter. As described above, each LED luminance value can be expressed as a convolution operation. In the case of large $\sigma_g^2$, the pixels corresponding to the position of LED are heavily influenced by diffused light neighboring LEDs. Thus, since the conventional method uses a simple threshold value to recover data modulated by OOK, the determination of which LEDs are lit becomes difficult. On the other hand, since the proposed MLD creates pseudo-captured images that are templates of the expected distortions in the blinking patterns of the LED transmitter, the proposed method accounts for the influence of diffused lights and demodulates the data effectively.

Second, we compare the performance of the proposed MLD with BDE for the reduction of computational complexity to the MLD algorithm without BDE. Figures 9(a) and (b) show the BER performances of the proposed MLD with/without BDE plotted against $\sigma_g^2$ when $E_b/N_0 = 10$ or 5 dB, respectively. Figure 10 plots the BER performance of both methods against $E_b/N_0$ when $\sigma_g^2 = 0.7$.

<table>
<thead>
<tr>
<th>Table I. BDE Parameters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Individuals ($J_B$)</td>
</tr>
<tr>
<td>Number of Dimensions ($N_B$)</td>
</tr>
<tr>
<td>Scaling Parameter $F_w$</td>
</tr>
<tr>
<td>Predefined Parameter $Cr$</td>
</tr>
<tr>
<td>Weight coefficient $J_{rate}$</td>
</tr>
<tr>
<td>$ReI_{max}$</td>
</tr>
</tbody>
</table>

181
Fig. 7. Example of captured image. The $4 \times 4$ LED array and the corresponding captured image are shown. There is no gap between the neighboring LEDs in the captured image.

Fig. 8. BERs of the proposed MLD and a simple threshold-processing algorithm (conventional method) plotted against $\sigma_g^2$ when $E_b/N_0 = 5$ and 10 dB. It can be observed that the BER performance of the proposed MLD algorithm is significantly improved compared with the conventional method.

Note that the BER performance of the proposed MLD without BDE returns the lowest BER. As one can see, the performance of the proposed MLD with BDE increases as the number of individuals ($J_B$) increases. The number of individuals indicates the number of solution candidates in the search space. In the case of fewer individuals ($J_B = 16$), the difference between the curve of $J_B = 16$ and that without BDE increases with increasing $\sigma_g^2$ or decreasing $E_b/N_0$, since the probability that the proposed MLD with BDE falls into a local optimum is high due to the low number of solution candidates. Conversely, in the case of many individuals ($J_B = 64$), the curve is almost coincident with that returned without BDE. This result indicates that the proposed MLD with BDE reduces the risk that the optimization process falls into a local optimum because of the large number of solution candidates.
Fig. 9. The BER performances of the proposed MLD with/without BDE plotted against $\sigma_g^2$: (a) $E_b/N_0 = 10$ dB, (b) $E_b/N_0 = 5$ dB. In the case of $J_B = 64$, the curve is almost coincident with that returned without BDE.
Fig. 10. The BER performance of both methods against $E_b/N_0 (\sigma_g^2 = 0.7)$.

Meanwhile, we can also confirm that a few plots of the proposed MLD with BDE is better than that without BDE when $E_b/N_0$ is low (Figs. 9(b) and 10). When a state includes a large noise or blurring is great, the global optimum solution detected by MLD may be irreconcilable with the true solution. Conversely, since BDE is not guaranteed to be optimally global, the local optimum solution detected by BDE may coincide with the true solution.

Third, we discuss the robustness of the proposed methods against the mismatch between the actual $\sigma_g^2$ and estimated one. Figures 11(a) and (b) show the BERs of the proposed MLD with/without BDE plotted against the degree of the mismatch when an actual $\sigma_g^2 = 0.4$ or 0.7, respectively. Note that the center (0,0) of horizontal axis indicates no mismatch between the actual and estimated ones. As seen, both performances deteriorate depending on increasing the mismatch of $\sigma_g^2$ regardless of the use of BDE. This is because the receiver cannot create template images that completely imitate all possible blinking patterns of the LED transmitter due to the mismatch. Meanwhile, we can also find that the BER performance deteriorate when the mismatch of $\sigma_g^2$ exceeds at least $\pm0.1$. This result indicates that the proposed MLD with/without BDE can recover correctly data even if the estimated $\sigma_g^2$ does not completely coincide with the actual one. Therefore, the proposed methods have the robustness against a certain level of estimation error of channel matrix.

Finally, we evaluate how well the BDE algorithm reduces complexity. The present study uses the total number of calculations of the Euclidean distance $N_E$ for decoding the data transmitted at time $t$ as an indicator of the reduction in computational complexity. In the case of the proposed MLD without BDE, $N_E$ is $2^M \times N$. In contrast, the $N_E$ of that with BDE is determined according to the number of calculations of $f(X_{j,g})$ in Steps 4 and 5 ($J_B \times g'$) and the number of searches $N_L$ $(= (M - 1)(N - 1) \times 2^4)$ in Step 7, where $g'$ is a value of $g$ when $ReI$ reaches $ReI_{max}$. In other words, we can calculate the $N_E$ of the algorithm with BDE as the sum of $(J_B \times g')$ and $N_L$. Note that the value of $g'$ depends on the variance $(\sigma_g^2)$ of the 2D Gaussian filter and the intensity of the noise ($E_b/N_0$). We performed computer simulations of the proposed MLD with BDE and calculate the average value of $g'$ with fixed $E_b/N_0$ for every value of $\sigma_g^2$. Based on the calculated $N_E$, the rate of complexity reduction $CR_{rate}$ [%] is calculated as follows.

\[ CR_{rate} = \left( \frac{N_E - N_{E_{BDE}}}{N_E} \right) \times 100 \]
Fig. 11. BER versus degree of mismatch of $\sigma_g^2$ when $E_b/N_0 = 10$ dB. Note that the center (0.0) of horizontal axis indicates no mismatch between the actual and estimated ones.

(a) Actual $\sigma_g^2 = 0.4$.

(b) Actual $\sigma_g^2 = 0.7$. 

185
The proposed MLD with BDE effectively reduces computational complexity, although the effect drops off as $\sigma_g^2$ increases. In this case, the proposed method with $J_B = 64$ achieves a complexity reduction of about 90% when $\sigma_g^2$ is low. Moreover, even as the number of $N_E$ increases owing to the large number of solution candidates, the proposed method achieves at least 50% complexity reduction in all cases.

$$CR_{rate} = \left(1 - \frac{N_E \text{ with BDE}}{N_E \text{ without BDE}}\right) \times 100.$$  \hspace{1cm} (16)

Figure 12 plots the rate of complexity reduction versus $\sigma_g^2$ when $J_B = 16, 32, 64$. The proposed MLD with BDE effectively reduces computational complexity, although the effect drops off as $\sigma_g^2$ increases. Let us focus on the curve for $J_B = 64$, which indicates BER performance nearly equal to the proposed MLD without BDE. In this case, the proposed method with $J_B = 64$ achieves a complexity reduction of about 90% when $\sigma_g^2$ is low. Moreover, even as the number of $N_E$ increases owing to the large number of solution candidates, the proposed method achieves at least 50% complexity reduction in all cases.

6. Conclusions
This present paper proposed a novel MLD method that uses pseudo-captured images that anticipate the blinking patterns of the LED receiver at a certain distance. The BDE algorithm was used to reduce the computational complexity needed for this technique. The results confirm that the MLD algorithm is effective. In addition, it has been observed that the demodulation performance of proposed MLD with BDE is almost equivalent to that without BDE when $J_B = 64$. Using BDE significantly decreases the computational complexity of the MLD calculations.

Acknowledgments
This work was supported in part by “JSPS KAKENHI Grant Number JP17K18282” and “the joint research program of the Institute of Materials and Systems for Sustainability, Nagoya University”.

Appendix
A. Updating process of BDE individuals
This section explains a BDE mutation and recombination process [24] (Step 2–4 in Sec. 4) using an
example shown in Fig. A-1.

Let us consider the LED array arranged in a matrix of $4 \times 4$ LEDs. Therefore, each vector of BDE has 16-dimensional elements, and Fig. A-1(a) shows each index $(m, n)$. To update a target vector $X_{j,g}$ shown in Fig. A-1(g) of an individual $j$ at an iteration $g$, three different individuals $e_1$, $e_2$, and $e_3$ are chosen, and their target vectors are shown in Figs. A-1(e), (b) and (c), respectively. According to **Step 2**, the Hamming distance $d_h$ between $X_{e_2,g}$ and $X_{e_3,g}$ is calculated as shown in Fig. A-1(d), thus $d_h = 6$. Since $d' = F_w \times d_h = 0.2 \times 6 = 1.2$, the $d' - (\text{int})d'$ should be 0.2. Assuming that $rand_1$ in Eq. (12) is smaller than 0.2, the number of changed elements for the mutation should be $N_{\text{mut}} = 2$.

Applying any two mutually exclusive changes in the base vector $X_{e_1,g}$ shown in Fig. A-1(e) produces a possible mutant vector $V_{j,g}$ shown in Fig. A-1(f). As shown in Fig. A-1(e), the two elements highlighted in black are randomly selected to change to produce the mutant vector. According to **Step 3**, a trial vector $U_{j,g}$ is generated with the current vector $X_{j,g}$ and the mutant vector $V_{j,g}$. Figure A-1(h) is the random numbers uniformly generated between 0 and 1. When $Cr = 0.3$ and $(m_{\text{rand}}, n_{\text{rand}}) = (3,1)$ denoted by * in Fig. A-1(i), the trial vector $U_{j,g}$ should be Fig. A-1(i). Finally, according to **Step 4**, $f(X_{j,g})$ and $f(U_{j,g})$ are evaluated and the better vector is stored as the updated vector $X_{j,g+1}$.

**Fig. A-1.** Mutation and recombination process of BDE (Step 2–4). $F_w = 0.2$, $Cr = 0.3$ and $(m_{\text{rand}}, n_{\text{rand}}) = (3,1)$.

<table>
<thead>
<tr>
<th>$(m, n)$</th>
<th>$X_{e_2,g}$</th>
<th>$X_{e_3,g}$</th>
<th>Distance $d_h = 6$</th>
<th>$X_{e_1,g}$</th>
<th>$V_{j,g}$</th>
<th>$X_{j,g}$</th>
<th>rand$_2$ in Eq.(13)</th>
<th>$U_{j,g}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.465</td>
<td>1</td>
</tr>
<tr>
<td>(1,2)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.814</td>
<td>1</td>
</tr>
<tr>
<td>(1,3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.218</td>
<td>0</td>
</tr>
<tr>
<td>(1,4)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.951</td>
<td>1</td>
</tr>
<tr>
<td>(2,1)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.546</td>
<td>1</td>
</tr>
<tr>
<td>(2,2)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.958</td>
<td>1</td>
</tr>
<tr>
<td>(2,3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.800</td>
<td>0</td>
</tr>
<tr>
<td>(2,4)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.063</td>
<td>1</td>
</tr>
<tr>
<td>(3,1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.632</td>
<td>0</td>
</tr>
<tr>
<td>(3,2)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.485</td>
<td>0</td>
</tr>
<tr>
<td>(3,3)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.119</td>
<td>1</td>
</tr>
<tr>
<td>(3,4)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.351</td>
<td>0</td>
</tr>
<tr>
<td>(4,1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.795</td>
<td>0</td>
</tr>
<tr>
<td>(4,2)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.154</td>
<td>1</td>
</tr>
<tr>
<td>(4,3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.872</td>
<td>0</td>
</tr>
<tr>
<td>(4,4)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.543</td>
<td>0</td>
</tr>
</tbody>
</table>

**References**


