Rolling-shutter-based asynchronous optical camera communication by a cycle pattern of received symbols using smartphones

Kazuki Ohshima¹, Takuji Naramoto¹, Keisuke Yamaguchi¹, and Wataru Chujo¹(2)

¹Department of Electrical and Electronic Engineering, Meijo University
1-501 Shiogamaguchi, Tempaku-ku, Nagoya 468-8502, Japan

(2) wchujo@meijo-u.ac.jp

Abstract:
We propose rolling-shutter (RS)-based asynchronous optical camera communication (OCC) with a variable symbol rate based on a cycle pattern of the receiver’s symbols which is uniquely determined by the ratio of the line interval of RS to the transmitter’s symbol length. RS-based asynchronous OCC has been experimentally demonstrated using various Android smartphones’ built-in cameras with a variable symbol rate when the ratio of the line interval of RS to the symbol length is expressed as a simple integer ratio. Error-free transmission has been achieved from 4 to 14 kilo-symbols per second (symbols/s) using smartphones’ 30-frame-per-second (fps) image sensors.

Keywords: visible light communication, optical camera communication, image sensor, rolling shutter, smartphone, asynchronous

Classification: Wireless communication technologies

References


1 Introduction

RS-based OCC has been studied to increase the symbol rate to 100 or 1000 times faster than the frame rate of the image sensor [1, 2]. However, since the line interval and timing of RS are fixed and not adjustable, the symbol rate is limited to almost the same as the line rate or integer divisions of the line rate to avoid the difficulty of symbol synchronization with variable symbol rates.

In this study, we find a solution to this issue by asynchronous communication with variable symbol rates. When the transmitter repeats symbols, “1” and “0,” alternatively by on-off keying (OOK) in the preamble, and the transmitter’s symbol length and the line interval of RS are different from each other, a cycle pattern of the receiver’s symbols is uniquely determined by the ratio of the line interval to the symbol length.

If the transmitter’s symbol length is unknown at the receiver, the cycle is used to estimate the symbol length [3, 4]. On the contrary, if the symbol length is known at the receiver, it becomes possible to distinguish correct symbols from wrong ones by the cycle. We propose and experimentally demonstrate RS-based OCC with variable symbol rates based on the cycle when the ratio of the line interval to the symbol length is expressed as a simple integer ratio.

2 Principle

In Fig. 1(a), the line interval, $T_l$, is a time delay between adjacent lines of RS. The transmitter’s symbol length, $T_s$, is assumed to be known at the receiver and limited to $T_l < T_s < 2T_l$ [3]. When the transmitter repeats symbols, “1” and “0,” alternatively by OOK in the preamble, a cycle pattern of the receiver’s symbols is given by

$$\frac{x}{y} = \frac{\frac{2T_l}{T_s} - 1}{1 - \frac{T_l}{T_s}},$$

where $x$ and $y$ are the numbers of the receiver’s symbols with a length of $T_l$ and $2T_l$ in the cycle, respectively. Both $x$ and $y$ are integer numbers and indicate the receiver’s symbol sequence. The cycle is determined with $x$ and $y$ as follows.

Fig. 1(b) shows an example of a cycle, where the ratio of the line interval to the symbol length, $T_l/T_s=3/4$. Since $x=2$ and $y=1$ are calculated from Eq. (1), the receiver’s cycle consists of two symbols with a length of $T_l$ and one symbol with a length of $2T_l$. When the transmitter’s symbol sequence is “010101,” the receiver’s symbol sequence is “01001011.” It turns out that the 4th and 8th receiver’s symbols are the same as the 3rd and 7th symbols, respectively.

As the number of symbols in a cycle increases, the cycle becomes more
complex. A cycle consists of short and long cycles. The symbol length is assumed to be $T_l < T_s < 1.5T_l$ [3]. The short cycle consists of $n$ symbols with a length of $T_l$ and one symbol with a length of $2T_l$, and the long cycle consists of $n+1$ symbols with a length of $T_l$ and one symbol with a length of $2T_l$. The cycle is given by

$$u = \frac{(n+2)-(n+2)}{(n+2)} \frac{T_l}{T_s},$$

where $u$ and $v$ are the numbers of short and long cycles, respectively in the cycle. In Eq. (2), $u$, $v$ and $n$ are integer numbers, and $n$ is derived from $x$ and $y$. The cycle is uniquely determined with $u$, $v$ and $n$ as follows.

Fig. 1(c) shows an example of a cycle, where $T_l/T_s=9/11$. Since $x=7$ and $y=2$ are calculated from Eq. (1), $n=3$ is derived. Next, $u=v=1$ is calculated from Eq. (2). Since $u=v=1$, the cycle consists of a short cycle and a long cycle. Since $n=3$, the short cycle consists of three symbols with a length of $T_l$ and one symbol with a length of $2T_l$, and the long cycle consists of four symbols with a length of $T_l$ and one symbol with a length of $2T_l$. When the transmitter’s symbol sequence is “101010101,” the receiver’s symbol sequence is “101010101.” It turns out that the 5th and 11th receiver’s symbols are the same as the 4th and 10th symbols, respectively.

(a) System configuration

(b) $T_l/T_s=3/4$, repetition of “1” and “0”

(c) $T_l/T_s=9/11$, repetition of “1” and “0” in the preamble

(d) $T_l/T_s=9/11$, random data sequence

Fig. 1. Examples of the transmitter’s symbols and cycle pattern of the receiver’s symbols.
Since the symbol sequence in the cycle is clearly determined in the preamble, correct symbols can be restored from the subsequent random data symbols. Fig. 1(d) shows an example of a cycle when the random data sequence is transmitted, where $T_l/T_s=9/11$. When the transmitted symbol sequence is “110010010,” the received symbol sequence is “1100100100.” Since it turns out in the preamble that the 5th and 11th received symbols are the same as the 4th and 10th symbols, respectively, the correct symbol sequence, “110010010,” is restored.

### 3 Experimental demonstration

To verify asynchronous communication by a cycle, RS-based asynchronous OCC is experimentally demonstrated using Android smartphones’ built-in cameras. In the transmitter, white LEDs with 850 and 7200 lumens are used. Converted luminance of both LEDs is approximately 1 Mcd/m$^2$. Field programmable gate array is used to provide accurate and stable symbol length, $T_s$. In the receiver, five vendors’ Android smartphones with a built-in camera were used. The frame rate of the image sensor is 30 fps. Measured image resolutions are QVGA (320x240 pixels) and VGA (640x480 pixels). Camera2 application programming interface is adapted for asynchronous OCC [5].

First, RS-based OCC is experimentally demonstrated when $T_l/T_s=1$. Figs. 2(a) and (b) show LED images captured with a built-in camera of AQUOS SH-M04 and Zenfone3 ZE520KL, respectively. Approximately 1-Mcd/m$^2$, 850-lumen LED
is used, image resolution of the built-in camera is QVGA, and measured distance is 5 centimeters, where $T_l/T_s=1$. In both figures, a threshold was set to each line using 8-bit luminance value, $Y$, in YUV color space to make symbol decision. Receiver’s symbols, “1” and “0,” are repeated regularly without error when the transmitter repeats symbols, “1” and “0,” alternatively by OOK. Only SH-M04 and ZE520KL achieved error-free transmission among five vendors’ smartphones when $T_l/T_s=1$. Their image sensor’s shutter speed or exposure time is considered to be fast or short. Measured symbol rates, 7397.5 and 7526.7 symbols/s are slightly different from each other because the line rates of both image sensors are slightly different.

Although the line rate is approximately given by multiplication of the frame rate and number of lines of RS, 30x240=7200 lines per second, it depends on each image sensor. Fig 2(c) shows measured bit error rate (BER) versus symbol rate around the line rate. When the symbol rate is 7397.5 symbols/s, the symbol rate coincides with the line rate, and error-free transmission is achieved.

![Graphs showing bit error rates and a captured image versus variable symbol rates.](image)

Fig. 3. Bit error rates and a captured image versus variable symbol rates.
Next, RS-based asynchronous OCC is experimentally demonstrated with variable symbol rates. Figs. 3(a), (b), and (c) show measured BERs versus variable symbol rates when $T_l/T_s$ is expressed as a simple integer ratio. Approximately 1-Mcd/m², 850-lumen LED is used. Transmitter’s symbols in each image frame consist of preamble and data symbols.

In the preamble, transmitter’s symbols repeat “1” and “0” alternatively by OOK. Since the transmitter’s symbol length is known at the image-sensor receiver, correct symbol sequence can be found by the cycle pattern.

In the data symbols, even when pseudo-random binary sequence is transmitted, correct data sequence can be restored by the cycle (see Fig. 1(d)). However, in the data symbols, repetition of “1” and “0” was transmitted instead of random data sequence to receive the preamble symbols, “1” and “0,” at the beginning of a line of the captured LED image in every image frame.

Figs. 3(a) and (b) show measured BERs versus variable symbol rates at 5 centimeters in QVGA. Since the maximum value of the denominator of $T_l/T_s$ is 13, the number of symbols required for the preamble becomes 13. The remaining symbols are used for the data symbols in every image frame. Error-free transmission was achieved from 4 to 7.5 kilo-symbols/s with variable symbol rates when $T_l/T_s$ is from 6/11 to 1.

In addition, Fig. 3(c) shows measured BERs versus variable symbol rates at 5 centimeters in VGA. Symbol rate in VGA increases two times more than that in QVGA. Since the maximum denominator of $T_l/T_s$ is 21, the number of symbols required for the preamble becomes 21. Error-free transmission was achieved from 10 to 14 kilo-symbols/s with variable symbol rates when $T_l/T_s$ is from 2/3 to 20/21. High-resolution, such as full HD (1920x1080 pixels) will adapt to more complex integer ratio because number of symbols required for the preamble can increase.

Moreover, Fig. 3(d) shows an example of captured LED image in QVGA when $T_l/T_s$ is 10/11. Approximately 1-Mcd/m², 7200-lumen LED is used, and measured distance is 5 centimeters. Since $x=9$ and $y=1$ are calculated from Eq. (1), it turns out that 9 symbols with a length of $T_l$ and one symbol with a length of $2T_l$ are repeated without error.

4 Conclusion

Asynchronous communication method for OCC by a cycle pattern of the receiver’s symbols was proposed and adapted for RS-based OCC. RS-based asynchronous OCC was achieved with variable symbol rates when the ratio of the line interval to the transmitter’s symbol length is expressed as a simple integer ratio. RS-based asynchronous OCC was experimentally demonstrated using Android smartphones’ built-in cameras. Error-free transmission of BER<10^-4 has been achieved from 4 to 14 kilo-symbols/s in QVGA and VGA.