Delft University of Technology
Master of Science Thesis in Embedded Systems Group

LED-to-Camera Communication in Mobile Scenarios

Runyu Feng



TU Delft Delft University of Technology

LED-to-Camera Communication in Mobile Scenarios

Master of Science Thesis in Embedded Systems Group

Embedded Systems Group
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Van Mourik Broekmanweg 6, 2628 XE Delft, The Netherlands

Runyu Feng

8th July 2024

Author
Runyu Feng
Title
LED-to-Camera Communication in Mobile Scenarios
MSc Presentation Date

15th July 2024

Graduation Committee

dr. Qing Wang (Chair)	Delft University of Technology
dr. Zaid Al-Ars	Delft University of Technology

Abstract

There are increasing applications of Visible Light Communication, and LED-to-Camera communication is one of the promising research directions. In this paper, we explore LED-to-Camera communication in mobile scenarios based on the rolling shutter effect to get higher data transmission rate. Movement causes the deformation of the overall shape of the LED in the captured image. And stripes close to the LED edges are too dark to be detected. Both affect the data detection by utilizing the rolling shutter effect. A method is proposed that can increase the transmission rate, which solves the problems caused by movement and darkness of stripes that close to the LED cover edges. After processing, the region of interest is recouped, i.e., the processing range used to detect the rolling shutter stripes becomes larger, which contains more complete data. An approach to detect stripe edges by converting pixel information into binary signal is proposed, which is used to solve the problem of undetectable stripe edges due to high frequency. The proposed method is evaluated through experiments under different conditions. The results demonstrate that the method is effective for addressing the challenges of LED-to-camera communication in mobile scenarios.

"There ain't no such thing as a free lunch"

Preface

The program I studied during my Bachelor's degree was Digital Media Technology, which is completely different from my Master's degree in TU Delft and largely irrelevant to this thesis project. But during my Bachelor's degree, I was fortunate to participate in an international exchange program organized by Nanyang Technological University (NTU), in which I was exposed to and learned about visible light communication for the first time, which I thought was very novel and interesting. Therefore, towards the end of the first year of my master's degree I was very happy to learn that dr. Qing Wang from the embedded group, i.e. my thesis supervisor, is working on research including visible light communication. Then in the first quarter of my second year I took the course CESE4110 Visible Light Communication and Sensing, taught by dr. Marco Zuniga and dr. Qing Wang. In this course, I learned a lot about VLC, which prepared me well for my thesis.

After many times of discussions with Qing as well as gradual mapping and research, my thesis topic was finally finalized. Time flies, I spent nearly 9 months on this topic, and I feel very fulfilled and accomplished to see the final success. During the thesis, with Qing's help, I had a poster presentation of part of my work at SITB 2024 and a presentation at the 2024 EuCNC & 6G Summit, which were amazing experience!

During my thesis, I would like to thank dr. Qing Wang, who is a very patient and sincere teacher who wants his students to learn real skills. In every discussion with me, Qing was very patient and guided me, which helped me a lot in my thesis! He also often pointed me in the right direction in many times. I am very lucky to have such a thesis supervisor! Also I'm very grateful to Hanting Ye who helped me with the equipment. My friends in the Netherlands and in China provided me with a lot of emotional support during my thesis, which made the process of doing thesis much more enjoyable. My family and boyfriend also provided me with a lot of emotional value, cracking me up when I was anxious and being very tolerant of my sometimes cranky moods. I'm very grateful to them.

In addition, dr. Zaid Al-Ars is a very important person for my master studies. During my first quarter at TU Delft, I took CESE4010 Advanced Computing Systems taught by Zaid. Since what I studied for my bachelor's degree was very different from what I studied for my master's degree, I didn't have a solid foundation in coding before that. After Zaid found out my problem, he was very patient and positive in guiding me and enlightening me, giving me extra time to do the assignments. I ended up completing the course successfully and gained confidence through it, which was a huge help in my master's studies!

Thank you very much!

Runyu Feng

Delft, The Netherlands
8th July 2024

Contents

Preface **vii**

1 Introduction **1**
1.1 Research Objective . . . 3
1.2 Problem Statement . . . 3
1.3 Contributions . . . 4
1.4 Thesis Content . . . 4

2 Background **7**
2.1 Rolling Shutter & Global Shutter . . . 7
2.2 Related Work . . . 8
2.2.1 Modulation Schemes . . . 9
2.2.2 LED-to-Camera Communication . . . 9

3 Design **13**
3.1 System Overview . . . 13
3.2 The Shrunk Region of Interest . . . 14
3.2.1 The State-of-the-art Method . . . 14
3.2.2 Drawbacks of the State-of-the-art Method . . . 15
3.3 Recouping the Region of Interest . . . 16
3.3.1 Challenges . . . 17
3.3.2 Determine Midpoints of Stripes within Initial Range . . . 21
3.3.3 Linear Fit for the Stripe Midpoints . . . 22
3.3.4 Find Critical Points on the Linear Fit Line . . . 23
3.3.5 Get New Processing Range . . . 23
3.4 Decoding the Data . . . 24
3.4.1 Challenge . . . 24
3.4.2 Convert Pixel Data to Binary Signal . . . 25

4 Evaluation **27**
4.1 Testbed . . . 27
4.1.1 Overview . . . 27
4.1.2 Hardware . . . 28
4.1.3 Software . . . 32
4.2 Testbed Evaluation . . . 34
4.2.1 Different LED Blinking Frequencies . . . 34
4.2.2 Different LED-to-Camera Distances . . . 36
4.2.3 Different Ambient Light Conditions . . . 38

4.2.4 Different Moving Directions 39
4.2.5 Different Moving Speeds 42

5 Conclusion **45**
5.1 Future Work 46

Chapter 1

Introduction

With the rapid development of the Internet, the demand for high-speed and reliable wireless communication continues to grow. However, have you ever considered that a very common element in our daily lives could be used for wireless communication? The answer is visible light! Visible light communication (VLC) is an emerging technology that utilizes the visible light spectrum for data transmission. Unlike traditional wireless communication methods that use radio frequencies, VLC uses visible light to transmit data, offering several unique advantages. These include higher security, faster transmission speeds, and lower interference levels. Furthermore, VLC is energy-efficient and can complement existing communication technologies. As a result, VLC is a promising solution for the future of wireless communication.

One valuable approach of VLC is LED-to-Camera communication, which is based on light-emitting diodes (LEDs) as transmitters and cameras as receivers, as shown in Figure 1.1. In this approach, data is captured by taking photos or recording videos with the camera of a mobile device. This LED-to-Camera communication takes advantage of the widespread availability of LEDs in various devices, as well as the prevalence of cameras in modern smartphones and other mobile devices. The rolling shutter effect is the core of LED-to-Camera communication, commonly found in modern digital cameras, which is a more efficient way to realize data transmission. The rolling shutter effect occurs because the shutter scans the image sensor line by line as the camera captures an image. The prerequisite for getting the rolling shutter effect is that the shutter speed is high enough and the ISO (International Organization for Standardization) is low enough. When the LED blinks at a certain frequency, the image captured with the rolling shutter will display light and dark stripes. The arrangement and width of these stripes can be used to demodulate LED signals to transmit data. This technology has practical applications, such as indoor positioning [12, 16, 27].

However, with the increase in application scenarios, LED-to-Camera Communication faces new challenges. Currently, relevant applications are based on stationary scenarios. Examples include indoor localization, indoor guided tours, intelligent lighting and communication systems, etc. The existing method for LED-to-Camera communication involves detecting the contours of all bright stripes and selecting the stripe with the largest contour. The center and radius of the smallest circumscribed circle around this contour are used to determine

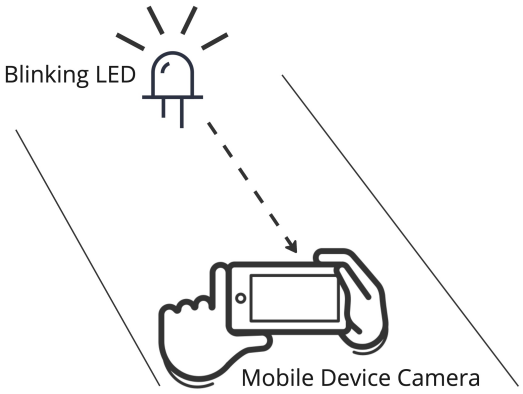


Figure 1.1: **LED-to-Camera communication**

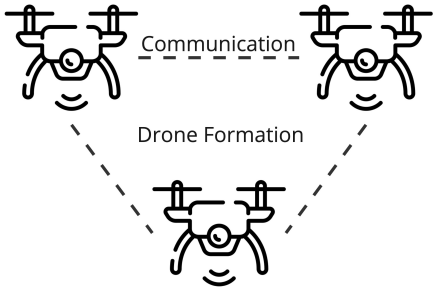


Figure 1.2: **Communication between drones**

the center and radius of the LED in the image. Then the smallest circumscribed square around the LED is defined as the processing area, detailed in Section 3.2. However, this approach encounters challenges in mobile scenarios, such as communication between drones in a formation (see Figure 1.2) and autonomous vehicles receiving signals from traffic lights (see Figure 1.3). In these cases, the original signal processing method has problems. Due to the relative motion between the transmitter and receiver, and the rolling shutter effect, the LED outline in the image will be deformed and no longer appears as a circle. This deformation leads to difficulties in signal decoding and reduction of data transmission rate. Based on existing methods applied in stationary scenes, it's limited detect all stripes, and only the centrally located stripes can be detected, described in Section 3.2.2.

In order to solve the problem caused by movement and improve the data transmission rate of LED-to-Camera communication, this research proposes a novel method. This method focuses on the center of the stripes rather than the entire outline of the LED. Because the shutter speed is very fast and the time of taking a picture is very short, the movement of the LED during this time can be approximated as uniform linear motion. First, we identify the center of each stripe detectable using the existing method. The coordinates of these centers are then linearly fitted. The other stripes are then detected along this linearly fitted line.

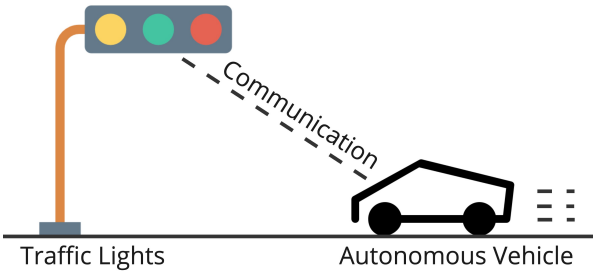


Figure 1.3: **Autonomous vehicle receives information from traffic lights**

At the same time, other problem has arisen in the application of this method. When using a large LED with a large cover, the stripes located at the center of the cover are easily detected. In contrast, the stripes located at the edges of the cover are difficult to be detected using the existing method, due to their low brightness. In order to solve this problem, we design a threshold that can be adapted to the overall and partial brightness of the image. Stripes close to the cover edge are detected by identifying points with pixel values larger than the threshold along the linear fit line of the center points of the stripes.

1.1 Research Objective

The objective of this thesis is to implement a comprehensive processing methodology based on LED-to-Camera communication in mobile scenarios, aimed at solving the problem of degradation of data transfer rate caused by movement. This thesis should also address the problem of stripes near the edge not being detected due to large covers encountered in the implementation. The proposed solution seeks to provide a practical approach to applying LED-to-Camera communication effectively in mobile scenarios.

1.2 Problem Statement

The research problems of this thesis are formulated as follows:

- *Solve the problem of frequency limitation of stripe detection.*

  The existing method obtains the stripe information by detecting the contours of bright stripes and then confirming the stripe boundaries. When the LED frequency is very high, the stripe width will be very small, which means that it is difficult to obtain the stripe boundaries with the existing method, resulting in the inability to obtain accurate stripe information. So we need a new method to solve this problem.

- *Solve the problem of not being able to detect all the rolling shutter stripes due to the movement.*

  The existing method utilizes the outer circle of the largest bright stripe to determine the processing range. But for mobile scenarios, the overall shape

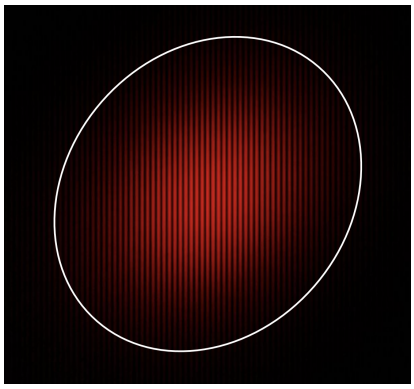


Figure 1.4: **Overall shape of LED in image with movement**

of the LED in the image is no longer circular. As shown in Figure 1.4, the white line is the overall shape of the LED in image. This means that the existing method is no longer applicable. A new approach applicable for mobile scenarios should be developed.

- *Solve the problem of not being able to detect all the rolling shutter stripes caused by the large LED cover.*

  The existing method is to detect the contours of the bright stripes to determine the width and position of the stripes. However, for LEDs with a large cover, the brightness of the stripes near the edge of the cover is very low, as shown in Figure 1.4. So that it is impossible to detect the contours of the darker stripes. A solution that can detect all the stripes of LEDs with large cover needs to be proposed.

1.3 Contributions

This thesis proposes a methodology for realizing LED-to-Camera communication in mobile scenarios. The research challenges presented in Section 1.2 can be effectively solved by this method. The contributions of this thesis are summarized as follows:

- A method is proposed for LED-to-Camera communication in mobile scenarios. This method enable to solve the problem caused by the deformation of the LED outline due to the movement, through enlarging the processing range that includes complete information transmitted by LEDs.
- An improvement is applied to the demodulation of LED signals based on the rolling shutter effect. This enhancement addresses the issue of stripes near the edges being too dark to detect due to large LED covers, thereby extending the processing range to obtain complete information.

1.4 Thesis Content

The goal of this thesis is to present a complete overview of the design, implementation and evaluation of the demodulation methodology applied for LED-

to-Camera communication in mobile scenarios. Additionally, a review of the related work in LED-to-Camera communication research is conducted.

1. *Introduction*

   Chapter 1 describes the motivation, brief background and main challenges of this thesis.

2. *Background*

   Chapter 2 provides a detailed description of the research background and reviews previous research on LED-to-Camera communication.

3. *Design*

   Chapter 3 presents our methods to solve the problems arising from LED-to-Camera communication in mobile scenarios, and the problems that arise during the application process.

4. *Evaluation*

   In Chapter 4, details of the proposed method implementation in a testbed are given first. Then, the evaluation of the proposed method is provided from several aspects based on experiments. Experiments are conducted from a number of different perspectives, including different LED cover sizes, different LED-to-Camera distances, different movement directions and speeds, different ambient lights, etc. The final results are obtained based on the analysis of these experiments.

5. *Conclusion*

   In Chapter 5, conclusions and future work of this thesis are summarized.

Chapter 2

Background

This chapter begins with an overview of LED-to-Camera communication, providing essential background information on the topic. It then explores various modulation and demodulation schemes used in LED-to-Camera communication. A comparative summary of these schemes is presented to show their strengths and limitations. Finally, based on the previous discussions, this chapter outlines the specific requirements for effective LED-to-Camera communication in mobile scenarios, considering the unique conditions.

2.1 Rolling Shutter & Global Shutter

Global Shutter and Rolling Shutter are two common image sensor shutter technologies that are primarily used in digital cameras, each offering distinct mechanisms and performance characteristics. The different ways these shutters operate result in different performance when capturing fast-moving objects or shooting specific scenes.

Camera with global shutter exposes all pixels simultaneously. At the moment of capture, all pixels throughout the image sensor begin and end their exposure at the same time. Because all pixels are exposed at the same time, the global shutter avoids image distortions that might be caused by fast moving objects or rapid camera movement. In contrast, for camera with rolling shutter, it exposes the pixels of the image sensor sequentially, row by row. This means that different rows of pixels start and end their exposure at different times during the shooting process, typically progressing either from top to bottom or left to right.

As shown in Figure 2.1, a fan rotating at an even speed is photographed with the rolling shutter and global shutter exposure modes, respectively. There is distortion of the fan in photos taken with the rolling shutter exposure mode. For global shutter, the fan shape is normal, avoiding spatial distortion. As shown in Figure 2.2, an LED that blinks at a fixed frequency is photographed with two exposure modes: (a) is the result of rolling shutter mode shooting, showing bright and dark stripes. And different frequencies will result in different stripes width; (b) is the result of a global shutter mode shot, which is consistent with the LED seen by the naked eye.

In this thesis, the LED acts as a data transmitter and the mobile phone camera acts as a receiver, which is LED-to-Camera communication. Based on

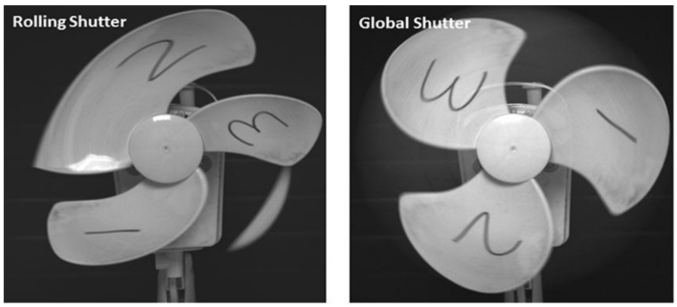


Figure 2.1: **Images of a moving fan, acquired with Neo sCMOS camera with rolling and global shutter exposure modes, same exposure time. The spatial distortion associated with the 'rolling shutter effect' is apparent in the left image. Global shutter is a 'snapshot' acquisition mode and avoids spatial distortion. [6]**

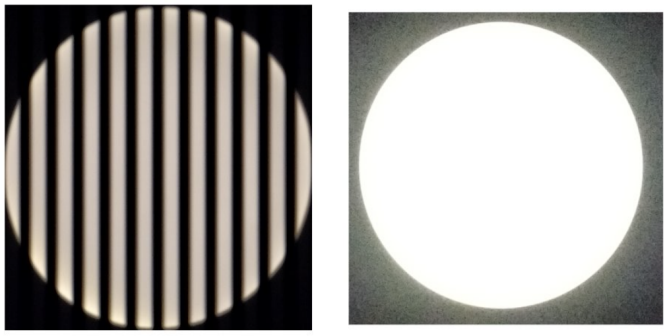


Figure 2.2: **Images of a flashing LED captured by camera with rolling(a) and global(b) shutter exposure modes, same exposure time.**

the rolling shutter effect, the information is modulated by flashing LEDs. The LED stripes obtained by capturing with the rolling shutter mode can be used to demodulate the information. Different stripe widths and stripe brightnesses contain different information. If we incorporate encoding methods, such as Manchester encoding, we can transmit data.

2.2 Related Work

For this topic, the ultimate goal is to increase the data transmission rate of the LED-to-Camera communication, which is highly dependent on the modulation approach. In addition to modulation based on the rolling shutter effect, there are many other modulation schemes. In this section, several different modulation schemes are presented, along with their corresponding extensions.

In the research process for this thesis, there are two main focuses. The first one is LED-to-Camera communication and the second one is the mobile scenarios.

There are some researches related to these aspects, which are introduced in this section.

2.2.1 Modulation Schemes

OOK Modulation

On-off keying (OOK) is an amplitude keying (ASK) method that represents binary data by the presence or absence of a carrier signal (in VLC, an optical signal). In OOK, the LEDs are turned on and off to transmit binary information, where LED on indicates binary 1 and LED off indicates binary 0. The main challenges to visible light communication are LED non-linearity and a narrow modulation bandwidth. Multi-LED phase-shifted OOK (MP-OOK) modulation can improve the bandwidth efficiency [30]. [13] evaluates the performance of VLC system with OOK-RZ (Return-to-Zero On Off Keying).

CSK Modulation

Color Shift Keying (CSK) is a modulation technique that uses changes in the color of light to represent binary data. In visible light communication, CSK transmits data by controlling the intensity of red (R), green (G) and blue (B) LEDs to produce different color combinations. These color combinations correspond to different symbols, thus enabling data transmission. A color-shift-keying embedded direct-current-optical-orthogonal-frequency-division-multiplexing (CSK-DCO-OFDM) scheme for VLC systems has been demonstrated [9]. In addition to providing a non-flickering CSK signal, this VLC system adds a high data rate OFDM signal to provide an extra modulation dimension.

PWM Modulation

Pulse Width Modulation (PWM) is a technique for transmitting information by adjusting the pulse width of a signal. PWM encodes data by varying the duty cycle of the pulsed signal, i.e., the ratio of the pulse's high level duration to its period. In VLC, the switching states of LEDs (on and off) can be used to represent binary data, and the pulse width is used to transmit specific information. A scheme of data transmission at the base of the pulse-width modulation (PWM) using visible light is proposed [1], whose maximum frequency of the transmission is 87 KHz.

2.2.2 LED-to-Camera Communication

LED-to-Camera communication is a technology based on Visible Light Communication (VLC), in which data is transmitted using an LED as the transmitter and a camera as the receiver. This technology utilizes the high-speed switching characteristics of LEDs and the camera's ability to take pictures, allowing data to be transmitted via optical signals and decoded by the camera capturing the optical signals. LED-to-camera communication has been the focus of many studies.

In [26], authors demonstrates how bits can be successfully encoded using frequency shift ON-OFF keying at frequencies that prevent flicker and successfully

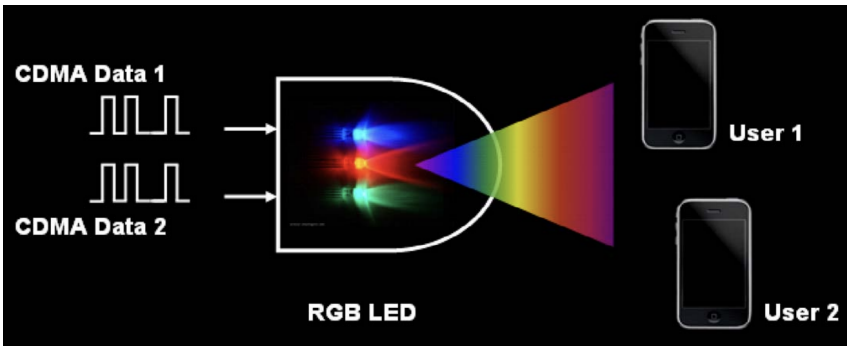


Figure 2.3: **CSK-CDMA VLC system [3]**

decoded by a low frame rate camera utilizing subsampling based on LED-to-Camera communication, by proposing the use of undersampled OOK. By utilizing the rolling shutter effect of CMOS sensors to capture continuous changes in the state of light (high speed OOK), which is not visible to the human eye, a data rate much higher than the camera's frame rate is achieved [7].

A LED-to-Camera communication system that uses code-division multiple-access (CDMA) and color-shift-keying (CSK) modulation simultaneously is proposed and demonstrated [3]. CDMA permits numerous users to access the network, whereas CSK is utilized to reduce single color light interference and increase the capacity of the VLC system. The suggested VLC scenario utilizing CSK and CDMA is displayed in Figure 2.3. The CSK signal is created when the CDMA Data1 and CDMA Data2 join. An array of RGB-LEDs transmits the CSK signal. Mobile phone cameras 1 and 2, provide the CSK signal to users 1 and 2, respectively. This is a novel approach to improve the throughput of VLC. [11] introduced ColorBars, an LED-to-camera communication system that uses the color-modulating properties of tri-LEDs to modify information that rolling-shutter cameras are to receive.

Localization landmarks can be encoded with data by modulating the PWM signal that is often used to operate LED lighting systems [25]. The usage of FSK was suggested by the authors in [1], which also addresses a number of other problems such inter-frame gap loss and synchronization. FSK is also used in [25] for NLOS (Non-Line of Sight) communications. NLOS communication, where the signal transmission does not require a direct line of sight (direct path of vision), is a method that utilizes the reflection or scattering of light to transmit signals, resulting in a more flexible method of communication, especially in environments where the line of sight is obstructed. [24] proposed a method that combines NLOS and LED-to-Camera communications.

The rolling shutter effect of the CMOS image sensor is applied by [14], [29], [15] and [2] to increase the data rate and get around the frame rate constraint on the CMOS image sensor. And [24] is also based on rolling shutter effect. In [14], a line-of-sight light-to-camera communication system called RollingLight is designed, implemented, and evaluated. It allows a light to communicate with various commercially available rolling shutter cameras. For RGB VLC transmission, [15] uses the multi-input multi-output (MIMO) approach to reduce inter-channel interference. [19] proposed a method that avoids signal interference

between two LEDs and mitigates performance degradation caused by receiver movement when using multiple LEDs for simultaneous illumination and communication. In [29], authors proposed and demonstrated a packet-reconstruction scheme, called beacon jointed rolling shutter pattern scheme. This scheme helps recover the original data packet, which is affected by the rolling shutter mode operation's processing time gap and the CMOS image sensor frame rate. It could be able to capture data at 10.32 Kbps. [2] demonstrates a unique thresholding approach (called EVA) to demodulate and synchronize the VLC based on the rolling shutter scheme. In [5], authors proposed a method of LED-to-Camera communication to improve the data rate of VLC. Packet loss may occur when passing data with LED-to-Camera communication, [10] preseted loTorch, which can efficiently prevent the packet losses for a fast and reliable LED-to-Camera communication. [8] proposed that LED-to-Camera communication based on rolling shutter can be used for interactive tours of cities and museums in addition to indoor and outdoor localization. The authors investigated the use of light diffusers to enhance images emitted from a rolled shutter mode. [19] introduced the Rolling Shutter MIMO OFDM technique based on the Rolling Shutter effect. In [20], asynchronous optical camera communication (asynchronous OCC) was introduced as a wireless communication method employing a rolling shutter camera and an LED, in which the performance of the system was evaluated.

There are some studies in indoor localization that also apply the LED-to-Camera communication [12, 16, 27]. In [12], the LEDs in different positions are set to flicker at different frequencies, which can be detected using the rolling shutter effect. Based on the known position of the LEDs and the position of the camera as well as the position of the LEDs in the image, a corner-of-arrival localization algorithm can be utilized to calculate the position and orientation of the smartphone relative to the LEDs. [16] presents an indoor VLC and localization system based on an orthogonal frequency division multiple access (OFDMA) scheme. [27] determines the distance to the light source by detecting the intensity of the received visible light signal lamp, thus realizing indoor positioning. In [22], authors suggested a technique based on visible light communication (VLC) and artificial intelligence (AI) technologies for precisely locating a user in an indoor setting by applying a weighted k-nearest neighbor (WkNN). [28] offers an LED-based indoor location method paired with hybrid OFDM (HOFDM) to reduce the influence of multipath distortion. Based on LED-to-Camera communication indoor positioning, in [23], a hybrid method that combines approaches for angle of arrival and received signal strength is used to determine the position of the remote receiver. In [18], authors presented an indoor locating technique with higher precision using visible light communication employing white-light LEDs and a stereo image sensor. [17] improved the precision of indoor positioning based on VLC further.

The above researches are based on stationary scenarios, which are easier scenarios for LED-to-Camera communication. [4] evaluated the bit error rate of data transmission between drones using LED-to-Camera communication in mobile scenarios, which concluded that different mobility conditions have different effects on the quality of data transmission. [21] proposed a visible light communication-based 3D indoor positioning and indoor navigation system that is enabled by UAVs.

In this thesis, the data transfer rate degradation caused by mobility will be improved, and evaluations of the proposed methodology will be conducted.

Chapter 3

Design

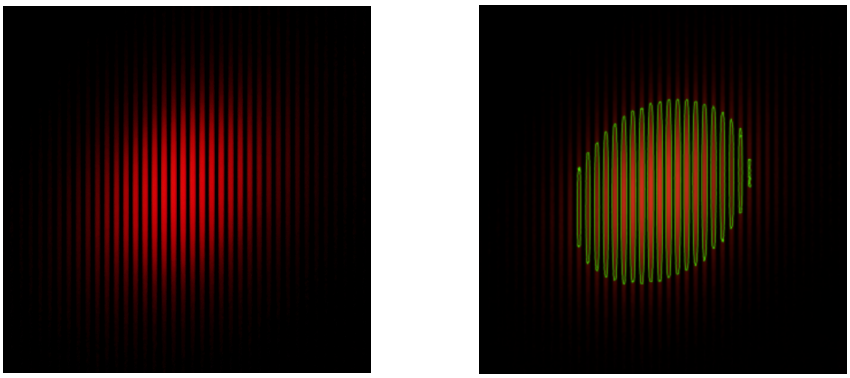
This chapter describes the methodology proposed by this thesis to solve the problems mentioned in Section 1.2. According to the previous section, we can learn that the LED-to-Camera communication based on the rolling shutter effect transmits data using the bright and dark stripes of the LEDs in pictures. It is easy to conclude that the main focus of the method design is the detection of rolling shutter stripes. Since the LED only occupies a part of the image, if we process the whole image it will lead to an increase in the amount of data and reduce the rate of processing. So we need to first determine the processing range, i.e., the region of interest, and then detect the stripes. Therefore, we introduce the current state-of-the-art method for determining processing range and detecting stripes, analyze its drawbacks, and then present our methodology and the design details.

3.1 System Overview

In this thesis, an LED-to-camera system is explored, in which the LED acts as a transmitter (TX) and the camera acts as a receiver (RX). The LED is turned on/off at a certain speed to modulate the data to be transmitted. Under rolling shutter mode, the camera captures images of the LED, resulting in bright and dark stripes that can be decoded into the transmitted information. Depending on the modulation used at the LED transmitter, different stripe widths and stripe brightness can contain different information.

State-of-the-art studies mainly focus on LED-to-Camera communication in stationary scenarios [12]. However, in this thesis, we consider mobile scenarios where either the LED transmitter or the camera receiver is in motion during communication, or both are hosted by moving devices. This movement introduces significant challenges, as the LED image captured by the rolling shutter camera becomes deformed, leading to potential information loss and decoding errors.

A novel methodology is proposed to address the unique challenges posed by LED-to-Camera communication in mobile scenarios. Our approach aims to compensate for the deformations in the captured LED image caused by movement, ensuring robust data transmission and accurate information retrieval. By enhancing the processing range (i.e., the region of interest) and employing ad-



(a) (b)

Figure 3.1: **(a) Original image of LED with movement; (b) Brief contours of the stripes**

aptive thresholding techniques, our method can effectively handle the variability and distortions introduced by motion.

The proposed methodology involves several key steps:

- **Determining the Region of Interest (RoI)**
  Identifying the appropriate processing range is crucial for capturing the full extent of the transmitted data. This step involves analyzing the captured images to define a range that includes all relevant information while minimizing background noise.

- **Stripe Midpoint Identification**
  The midpoints of the stripes are the key to decode the transmitted data. They are determined by locating the brightest points along each stripe. This step is essential for accurately determining the motion and enlarging the processing range.

- **Adaptive Thresholding**
  Considering the variations in brightness and stripe widths due to movement, an adaptive thresholding technique is applied. This method dynamically adjusts the threshold based on the observed pixel values, improving the detection accuracy of stripe edges and LED edges.

3.2 The Shrunk Region of Interest

3.2.1 The State-of-the-art Method

The first crucial step in efficiently parsing signals from blinking LEDs is determining the processing range, which dictates the amount of pixels to process and ensures all data is included.

For this, first, we need to detect the contours of the stripes in the image captured by the rolling shutter camera as shown in Figure 3.1 (a). Then, the image is binarized to obtain a binary image. Then all contours are retrieved and a complete hierarchical tree is constructed. Then the redundant points in

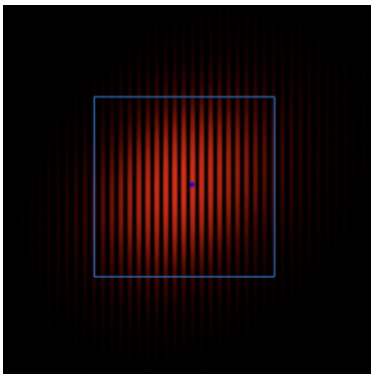


Figure 3.2: **Initial processing range, i.e., the Region of Interest (RoI)**

horizontal, vertical and diagonal directions are compressed and only the key points are retained. In this way, the contours are effectively compressed to obtain the main contours in the figure, which are the contours of the bright stripes. As shown in Figure 3.1 (b), the red lines surround the bright stripes. Third, we select the largest of the contours and obtain the outer circle of this contour as well as the circumscribed square of the outer circle. Then, the center of the outer circle is considered as the center and the circumscribed square as the processing range. As Figure 3.2 shows, the red square is the obtained processing range. Finally, within the processing range obtained in the previous step, we average the positions of the contour points in the vertical direction as the edges of the stripes[12].

Since the camera scans a row of pixels using a fixed amount of time, different frequencies will produce different stripe widths. Based on the previous four steps, the position of the stripes is obtained. Combined with the encoding method, we can transmit the complete data.

The above steps are the state-of-the-art method for determining the processing range. If this method is applied to a mobile scenario with high power LED and large cover, it can be seen in Figure 3.2 that the resulting processing range contains only the brighter part in the center, excluding the darker parts close to the edges of the LED and the parts that are distorted due to movement. Therefore, region of interest has shrunk, which means that information loss happens.

3.2.2 Drawbacks of the State-of-the-art Method

Detect stripe contours

In Figure 3.1 (b), we first binarize the image and then determine the stripe contours by finding the points in the binarized image where the pixel values jump. This means that the pixel value difference between the bright and dark stripes should be relatively large for the boundary to be detected. However, for LEDs with a large cover, the bright stripes near the edge of the cover are darker and the difference in pixel value from the dark stripes is not large enough, which means that the method in this step has problem in detecting these less bright stripes. Eventually, this will lead to the loss of information.

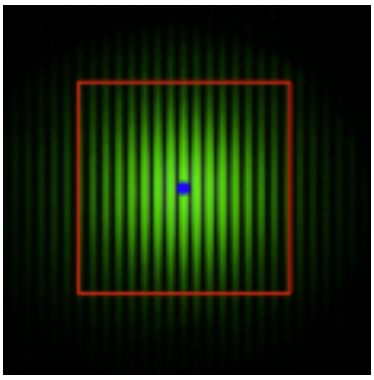


Figure 3.3: **Processing range using the state-of-the-art method with large LED cover in stationary scenario**

Determine processing range

In Figure 3.2, the largest contour is chosen, which is the longest stripe, and the outer square of its circumcircle is considered as the processing range. However, in mobile scenarios, the overall shape of the LED is no longer a circle, which will lead to some stripes beyond this processing range, meaning that we will miss some information. As shown in Figure 3.2, the red square is the processing range by using this method in mobile scenarios, which doesn't include all the stripes because of the movement.

Darker stripes close to the LED edges are not able to be detected using this state-of-the-art method. To view the stripe loss exactly, the processing range by using this method with large LED cover in stationary scenario is shown in Figure 3.3, which also doesn't include all the stripes, because of the low brightness of the stripes close to the LED edges.

Determine stripe edges

To determine the stripe edges, the averages of the points on the vertical contour lines need to be calculated, which serves as the edges of the stripes. In order to get the average, a range should be set. Then we calculate the mean of the points' positions within this range around the contour as the edges. Therefore, this range is crucial for detecting stripe edges. However, if the flickering frequency of the LED is very high, it will cause this range to be difficult to determine, because of the smaller stripe width. This means that accurate stripe edges will not be able to be obtained.

According to these three drawbacks, a new methodology that can be applied in mobile scenarios, solve the large-cover problem and detect high-frequency signals needs to be proposed, which should recoup the region of interest.

3.3 Recouping the Region of Interest

To address the drawbacks of detecting stripe contours and determining processing range when applying the state-of-the-art method to mobile scenarios as mentioned in Section 3.2.2, method of recouping the region of interest should

be proposed, which enlarges the processing range. The main challenges of this method are shown in this section. Each step of the method is explained and the reason of processing is described.

As previously discussed, our primary objective is to extend the processing range to include a greater number of stripes and detect signals with higher frequencies. To achieve this, it is essential to first establish a rough preliminary processing range that will allow us to apply the method we propose effectively. Initially, we adopt the first three steps of the state-of-the-art method as illustrated in Section 3.2.1. These foundational steps are crucial for setting the stage for our enhanced processing technique. After executing these three initial steps, we obtain a processing range that contains the majority of the stripes, particularly those that are centrally positioned and bright enough for stripe contour detection. This initial range is highlighted by the red square in Figure 3.2.

The subsequent steps build upon this initial processing range, extending and making it more accurate to achieve our goal of capturing more stripes and higher frequency signals. The detailed procedure for these subsequent steps is as follows.

3.3.1 Challenges

Challenge 1: How to determine processing range in mobile scenarios?

The most important first step in efficiently parsing the signals transmitted through the blinking LEDs is to determine the processing range, which decides the amount of pixels in the image will be processed and whether all the stripes are included. However, it is difficult to confirm the processing range due to the deformation of the LED contour generated by the movement and the darkness of the stripes close to the LED edges. So we need a method to get the accurate processing range.

Since the LED is located in the center of the cover, in theory, the brightness of the stripes in the vertical direction should diminish from the middle to the two ends and be symmetrical. Therefore, we only need to find the midpoint of each stripe and perform a linear fit for the midpoints. This obtained linear fit line will pass through the midpoint of all the stripes. Then we set a pixel value threshold for the midpoints on the line to determine the LED edges in the image, detailed in Challenge 3. Based on the LED edges and considering the middle of the two ends as the processing center, we can obtain the processing radius and then determine the processing range in mobile scenarios.

Challenge 2: How to get the midpoint of each stripe?

The midpoint of the stripe is the brightest point on this stripe. First, we can get the two ends of the stripe based on pixel values along the vertical line through this stripe. The middle of the line connecting the two ends can be considered as the midpoint of the stripe. In order to find the ends of the stripe, a pixel value threshold also needs to be set, which is the same as the threshold to determine the LED edges. The method to set adaptive threshold is described in Challenge 3. As shown in Figure 3.4, the green points are the middles of the bright stripes.

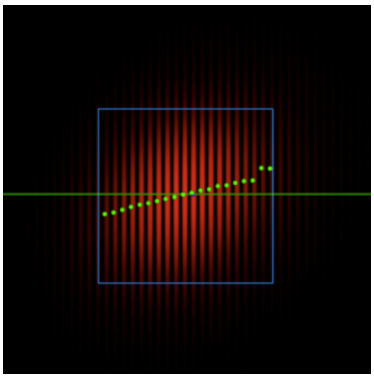


Figure 3.4: **Midpoints within processing range using existing method in mobile scenario**

Challenge 3: How to set the adaptive threshold of pixel value for determining the LED edges?

Different LEDs, covers, blinking frequencies, brightness, distance of the LED from the camera, the ambient light, etc. will affect the threshold that needs to be set. This is because these conditions have effects on the percentage of LED in the image, the brightness of the stripes, and the edge brightness of the LED. So different thresholds should be set for different situations, which means we need to find an adaptive approach to obtain the threshold.

Regardless of the stripe width, distance between LED to camera and brightness of the LED, the distribution of pixel values in the region of the image containing the stripes should be similar, because the centers of both the stripes and the entire LED are brightest, with brightness gradually dimming around them. In order to find the distribution of the pixel values, we obtain the pixel values on a horizontal line through the center of the processing range in the image, and plot the Cumulative Distribution Function (CDF) of the pixel values. As shown in Figure 3.5, these plots are CDF of pixel values of images in different scenarios (different distances, movement, etc.). The red dots are the CDF of each pixel value and the blue curve is the fitted curve of CDF.

In Figure 3.5 (a), (b), (c) and (d), these plots share common trends. The cumulative probability increases rapidly in the range of lower pixel values, then the growth rate decreases, followed by a gradual tendency to grow at a steady rate.

This indicates that the ambient light and dark stripe portions comprise the majority of the processing range, so the increase rate of cumulative probability is rapid within the range of low pixel values in the CDF. The cumulative probability increases nearly uniformly in the brighter portion, which is the part of the bright stripes. The pixel value for determining the edges of the LED in the image is the pixel value for the boundaries of the light and dark stripes, which is the x-value when the cumulative probability growth rate starts to stabilize. To acquire this value, we first obtain the derivative function of the CDF fitted curve and plot it, as shown in Figure 3.6.

In Figure 3.6, the derivatives of CDF fitted curve of these four plots all decrease first, and then there is a turnaround and they start to increase. This

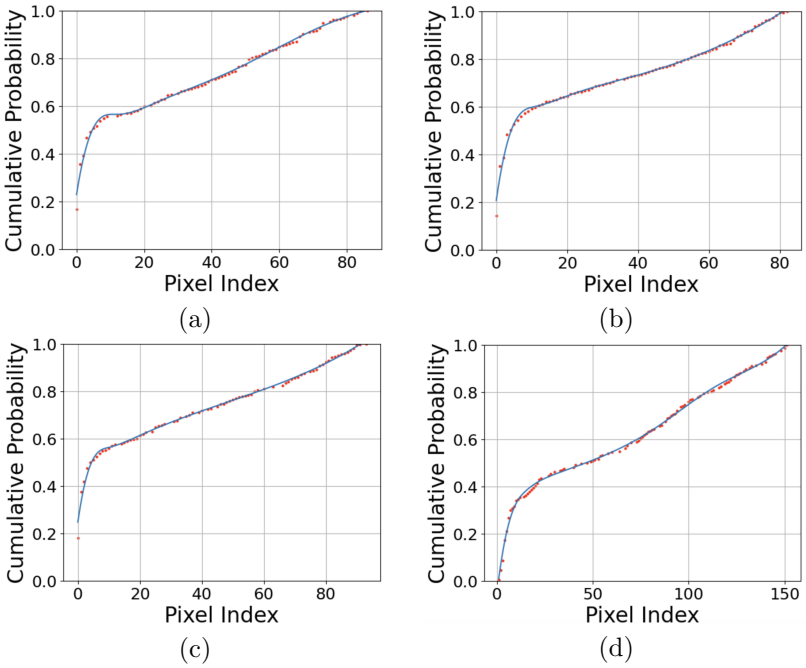


Figure 3.5: **CDF of pixel values within the processing range of four different images**

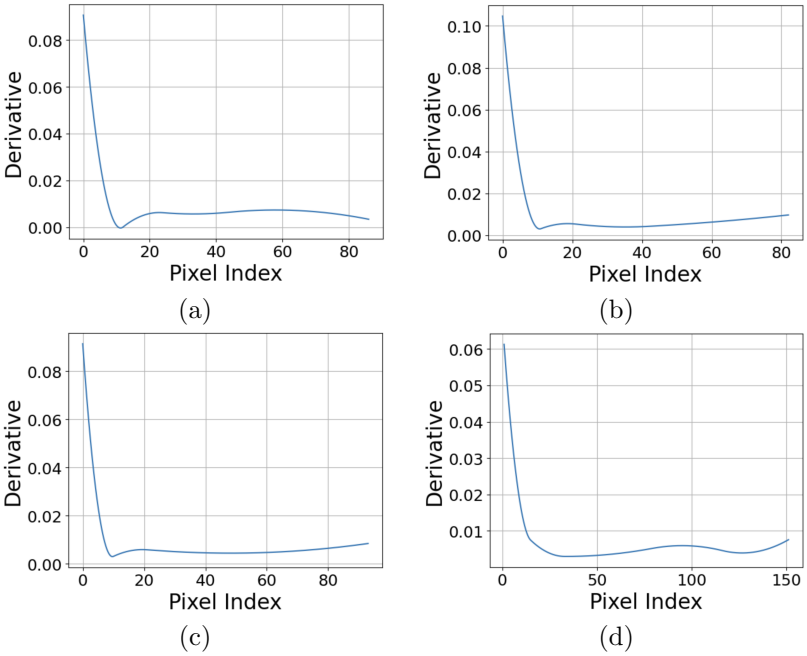


Figure 3.6: **Derivative of CDF fitted curve of four different images**

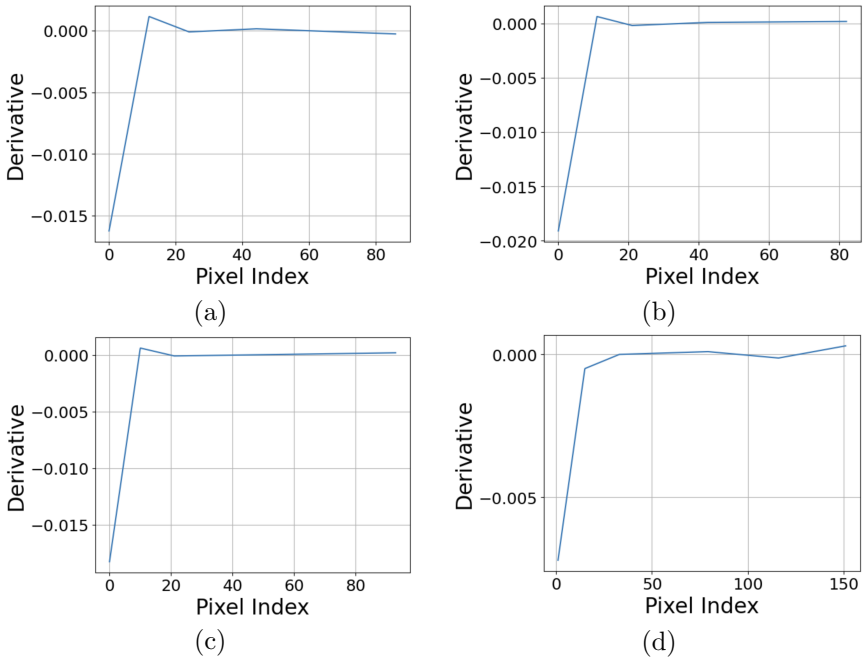


Figure 3.7: **Second Derivative of CDF Fitted Curve of four different images**

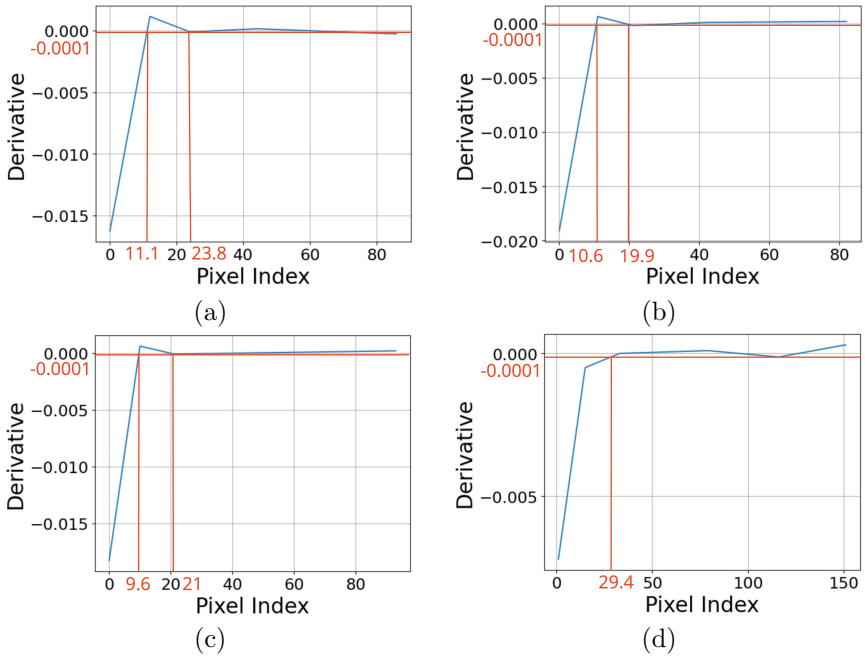


Figure 3.8: **x values when y=-0.0001 of four different images**

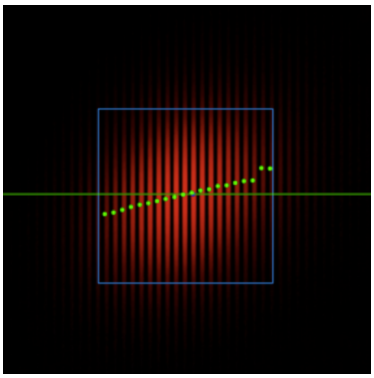


Figure 3.9: **Determine midpoints of stripes within initial range**

x-value (pixel value) of this turnaround point is the pixel value for determining the LED edges. To obtain this value, the second derivative function of CDF fitted curve is made and plotted, as shown in Figure 3.7.

After experimentation, we use the x-value with the second order derivative equals to -0.0001 as the pixel value to be found. As shown in Figure 3.8, for some plots, there are more than one x-value. As a rule of thumb, if a pixel value is below 15, we approximate it as black. So we choose the x-value greater than 15 and smaller than 35 as the threshold. After applying this threshold, we can get the new processing range include more complete data.

3.3.2 Determine Midpoints of Stripes within Initial Range

In mobile scenarios, the overall shape of the LED in the captured images is irregular due to the movement of the mobile phone camera or the LED itself. This irregularity causes a challenge when attempting to obtain an accurate processing range based on the outer circle of the largest stripe contour. However, given that the exposure time for capturing an image is typically very short, we can reasonably assume that any relative motion during this period can be approximated as uniform linear motion. This assumption allows us to simplify the problem: despite the overall outline is irregular, the midpoints of these stripes should align a straight line, as these points represent the center of each stripe.

As discussed in Challenge 2 of Section 3.3.1, the midpoint of each stripe corresponds to the brightest point of the stripe. Therefore, in this step, we focus on accurately identifying these midpoints within the initial processing range outlined by the red square in step 3. After applying the solution of Challenge 2 of Section 3.3.1, which is setting threshold to obtain the top and bottom ends of each stripe and regarding the middle of the two ends as the midpoint, the midpoints are as shown by the green dots in Figure 3.9.

By focusing on the midpoints, we are working with the consistent feature of the stripes, regardless of the overall irregularity in the shape of the LED. This step is crucial, because it makes the foundation for more precise and robust signal detection and processing in the following steps.

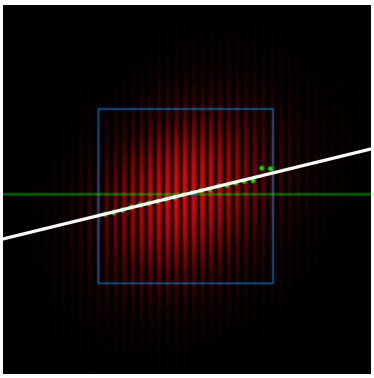


Figure 3.10: **Linear fit for the stripe midpoints**

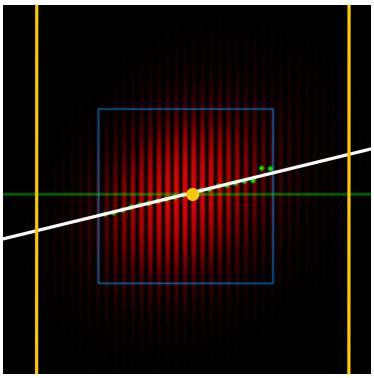


Figure 3.11: **Find critical points on the linear fit line**

3.3.3 Linear Fit for the Stripe Midpoints

Given the assumption that the movement during image capture can be approximated as uniform linear motion, the midpoints of all stripes should ideally lie along a straight line. This forms the basis for this step: performing a linear fit for the stripe midpoints within the previous processing range.

First, the coordinates of the midpoints of each stripe are collected, which are identified and marked in the previous step. These midpoints, represented by green dots in Figure 3.9, serve as the data points for the linear fit. Second, a linear regression algorithm is applied to these midpoint coordinates. The relationship between the horizontal and vertical coordinates of the stripe midpoints is modeled. The linear regression algorithm calculates the best-fit line, which represents the optimal linear approximation of the stripe midpoints' positions. The resulting best-fit line is shown as the white line in Figure 3.10.

This line helps to confirm that the midpoints indeed align along a straight line, validating the uniform linear motion assumption. Therefore, we can consider that all the stripe middles are approximately on this fitting line, which means the focus for the subsequent steps is on this line.

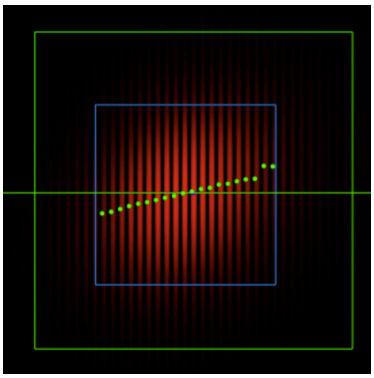


Figure 3.12: **Get new processing range**

3.3.4 Find Critical Points on the Linear Fit Line

In this step, the concentration is on detecting the pixel values along the linear fit line that was established in the previous step. The goal of this step is to determine the boundary of the LED in the image by identifying the brightness of the pixels near the edge of the LED.

First, the pixel values of all the pixels that lie on the linear fit line are extracted. To identify the LED's boundary, a threshold for the pixel values needs to be established. This threshold corresponds to the brightness level that differentiates the LED from the background. The pixels near the edge of the LED typically can be used as a critical point. Since the brightness conditions can vary significantly in different images and scenarios, a fixed threshold value is not suitable. Instead, an adaptive approach, that can determine the appropriate threshold dynamically for each image, is necessary. This adaptive thresholding method is detailed in Challenge 3 of Section 3.3.1.

After the adaptive threshold is applied, the pixels that meet this threshold value are identified as the critical points of the linear fit line. These points represent the boundary of the LED where the brightness changes from the LED to the background. By drawing vertical lines through these critical points, the two ends of the LED can be marked. These vertical lines, shown as yellow lines in Figure 3.11, are the LED's boundaries in the image.

By following this detailed process, the LED's boundaries are determined effectively, which can precisely obtain the processing range of LED. This step is crucial for further processing, as it defines the Region of Interest (ROI) for subsequent analysis and signal detection.

3.3.5 Get New Processing Range

In this step, a new and more accurate processing range is determined, based on the previously identified LED boundaries. This precise processing range ensures that more data transmitted by the LED can be captured and improves the accuracy of subsequent analysis.

After identifying the two ends of the LED with the vertical yellow lines in the previous step, the midpoint between these two ends is calculated. This midpoint represents the geometric center of the LED in the image, which is found by

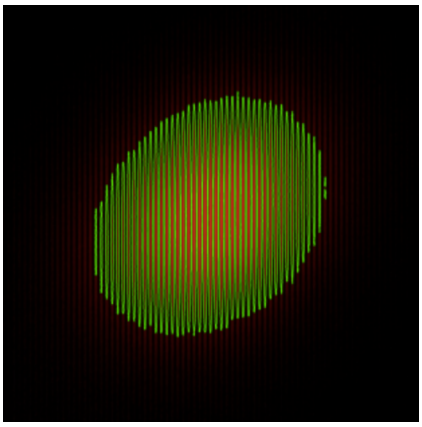


Figure 3.13: **Contours of Bright Stripes with High Frequency**

averaging the horizontal coordinates of the two end points. This point is marked as the yellow dot in Figure 3.12. The side length of the new processing range is determined by the horizontal distance between the two end points of the LED. This distance is used as the length of each side of the square processing range, ensuring that the entire LED is encompassed and the unnecessary background areas are excluded. The new processing range is represented by the green square in Figure 3.12.

The new processing range includes more data transmitted by the LED. It's crucial for improving data transmission rate in mobile scenarios, because it ensures that all the information is captured. By expanding the processing range, the robustness and reliability of the data processing pipeline are improved.

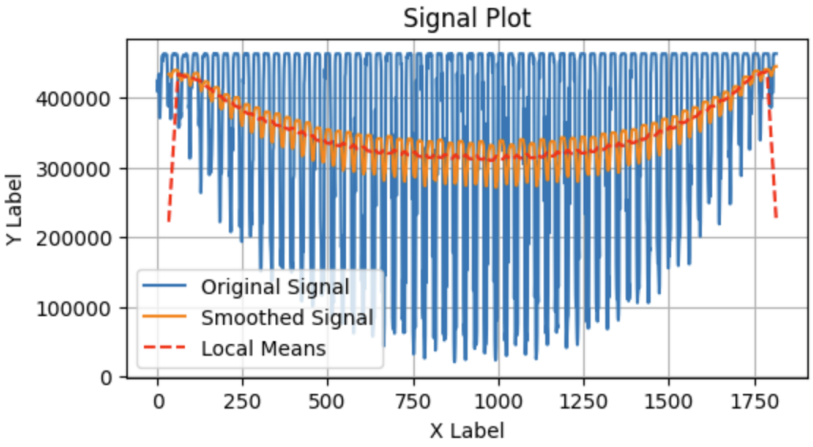
3.4 Decoding the Data

3.4.1 Challenge

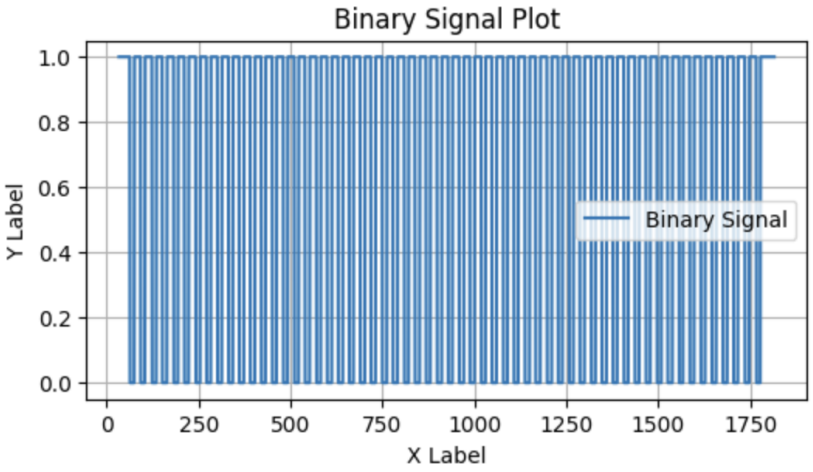
How to detect the stripe edges?

The method in Section 3.2 is to get the contours of the bright stripes first, and then the averages of the vertical positions of the points on the left and right outlines of the stripe are used as the edges of the stripe. For stripes with large widths, this method is feasible. However, for smaller widths, i.e., LEDs that blink at high frequency, this method will make it difficult to distinguish between the different stripes. As shown in Figure 3.13, the outlines of the stripes are so blended that it is not possible to obtain accurate stripe boundaries by calculating the average vertical position. According to the problem of this situation, a new method to detect stripe edges needs to be proposed.

It's not feasible to get the outline directly, so we think differently and decide to convert the pixel values into signals. After converting the signals to binary, the lengths of 0 and 1 are the widths of dark and bright stripes. After using the solution in Challenge 1 to obtain a more accurate processing range, we sum the pixel values in each column within the new processing range as the signal of that column. As shown in Figure 4.10 (a), the blue curve is the signal plot. Then we



(a)



(b)

Figure 3.14: **(a) Signal plot of the sum of pixel values; (b) Binary signal.**

smooth this signal. Sliding window smoothing is applied and the window size is set to 35. The smoothed signal are shown by the yellow curve in Figure 4.10 (a). Because the pixel signals are different in different columns, we can't set a fixed threshold for converting the signal to binary. According to this problem, we compute the local means of the smoothed signal as thresholds, which is the red dotted curve in Figure 4.10 (a). By setting signals greater than local means to 1 and signals less than local means to 0, we can successfully convert signals to binary, as shown in Figure 4.10 (b).

3.4.2 Convert Pixel Data to Binary Signal

This is the final step. The objective is to detect the edges of each stripe within the new and more accurate processing range. This process transforms the raw pixel data into a clear binary signal, enabling precise stripe edge detection.

First, within the new green processing range, the sum of pixel values for

each column is calculated, which represent the total brightness for each column. This plot shows the intensity variations across the LED stripes, with peaks corresponding to bright stripes and troughs to dark stripes. Next, to convert the brightness into binary signals, we set local thresholds that distinguish between bright and dark regions, which are determined adaptively. The process for setting these local thresholds is detailed in Challenge of Section 3.4.1.

Using the local thresholds, the brightness is converted into binary signals. In the brightness plot, regions above the threshold are assigned a value of 1 (representing bright stripes), while regions below the threshold are assigned a value of 0 (representing dark stripes). This binary representation simplifies the detection of stripe edges. With the binary signals, the lengths of continuous *0*s and *1*s. These lengths correspond to the widths of the dark and bright stripes, respectively. The transitions between 0 and 1 in the binary signals indicate the edges of the stripes. By identifying these transition points, we can accurately detect the stripe edges within the processing range.

By following this detailed process, we transform raw pixel data into binary signals, which is essential for accurately captured the data transmitted by the blinking LED.

Chapter 4

Evaluation

This chapter presents the implementation and performance evaluation of the proposed method for LED-to-Camera communication in mobile scenarios. It begins with a comprehensive introduction to the testbed used for the method, providing an overview and both the hardware and software implementations. Following this, the chapter details the experimental evaluation, including the methodology, process, and results.

4.1 Testbed

The purpose of this testbed is to implement and evaluate the proposed method detailed in Section 3, including both hardware and software components. The hardware components are the transmitter, which is a LED, and the receiver, which is a mobile phone camera, as detailed in Section 4.1.2. The software aspect includes the control of the hardware and the upper layer processing on the acquired rolling shutter effect images, described in Section 4.1.2.

4.1.1 Overview

The main objective of the testbed is to create a controlled environment for the implementation and evaluation of the proposed method. By simulating real-world conditions, the testbed ensures that the communication between the LED transmitter and the mobile phone camera receiver is accurately assessed in mobile scenarios.

The testbed consists of two main components, hardware and software. For LED-to-Camera communication, the LED serves as the signal transmitter, different LED types, cover sizes and configurations can be used to test various characteristics and conditions of LED transmission. In this testbed, a low power LED with small cover and a high power LED with large cover are used, and different blinking frequencies are tested, detailed in Section 4.1.2. The mobile phone camera acts as a receiver, which uses rolling shutter mode when taking pictures and recording videos, also described in Section 4.1.2.

The software comprises of hardware control and upper layer processing, detailed in Section 4.1.3. The hardware control module manages the operation of the LED. After capturing the images, the processing module processes the

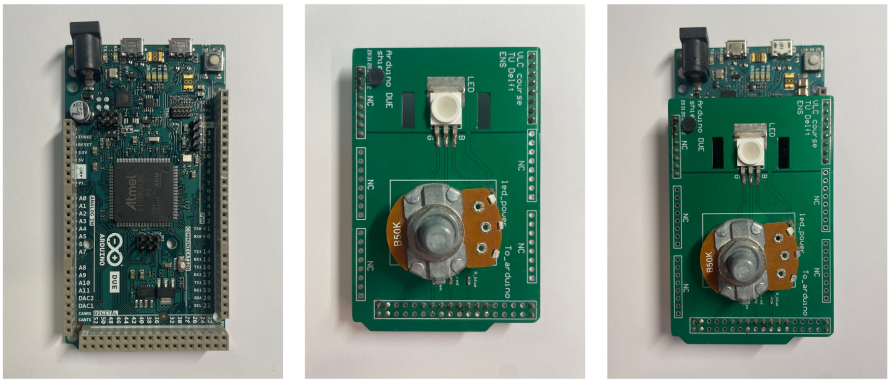


Figure 4.1: **The transmitter: (a) Arduino DUE Board; (b) Low Power LED Cape; (c) Plugging LED Cape to Arduino DUE Board. (The transmitter was borrowed from the Visible Light Communication & Sensing course of TU Delft.**

rolling shutter effect images, which detects the stripes of the LED and convert the stripe information to binary signals.

Based on the hardware and software implementations, the testbed is prepared to effectively evaluate the proposed method.

4.1.2 Hardware

In this subsection, the details of hardware implementation is presented, including the transmitter and the receiver.

Transmitter(LED)

For LED-to-Camera communication, the LED is used as the transmitter. In order to simulate a variety of scenarios, create a more comprehensive evaluation environment and test the robustness of the proposed method, two different LEDs are used, a low power LED with a small cover and a high power LED with a large cover. By applying both of the LEDs, the aim is to completely test the proposed method's performance across a wide range of conditions and help to adjust to a variety of real world conditions.

- **Low Power LED**

  As shown in Figure 4.1, the setup includes two primary components: (a) is Arduino DUE board, (b) is the low power LED cape. The Arduino DUE board serves as the control unit. After plugging the LED cape to Arduino DUE board, the LED can be precisely controlled to adjust its blinking frequency.

  For example, in Figure 4.2, images (a) and (b) were captured by the mobile phone camera, as detailed in the Receiver section of Section 4.1.2. Image (a) shows the LED blinking at a frequency of 10,000 Hz, while image (b)

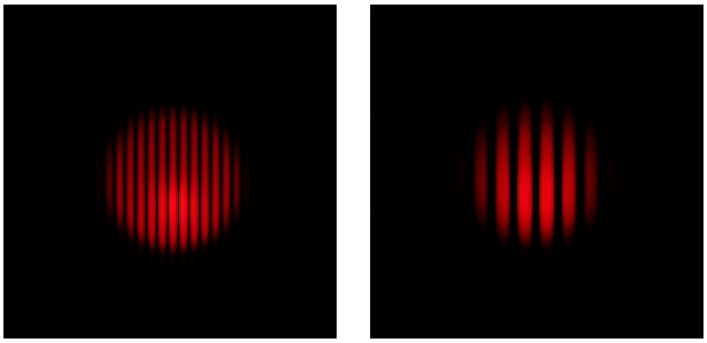


Figure 4.2: **(a) Blinking Low Power LED with frequency = 10 kHz; (b) Blinking Low Power LED with frequency = 5 kHz**

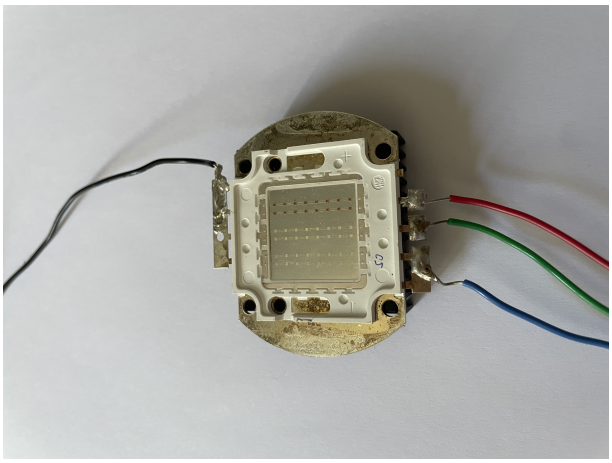


Figure 4.3: **The used High Power LED in the evaluation.**

depicts the LED blinking at a frequency of 5,000 Hz. Both images were taken with an LED-to-Camera distance of 20 cm.

The low power LED with small cover simulates scenarios with limited light output and closer LED-to-Camera distances, enabling to evaluate performance of the proposed method under these conditions. By changing the blinking frequency of the LED through Arduino DUE board, different transmission rates can be tested. The robustness of the proposed method in handling varying signal frequencies under these scenarios can be accessed.

- **High Power LED**

  The LED in Figure 4.3 is the high power LED, which is brighter than the low power one and covered by the large LED cover in Figure 4.5. The model number is Warm Natural Cold White RGB Bulb Lamp DIY Matrix Integrated 3000K, RGB with 50W of brand CHANZON. This LED is also

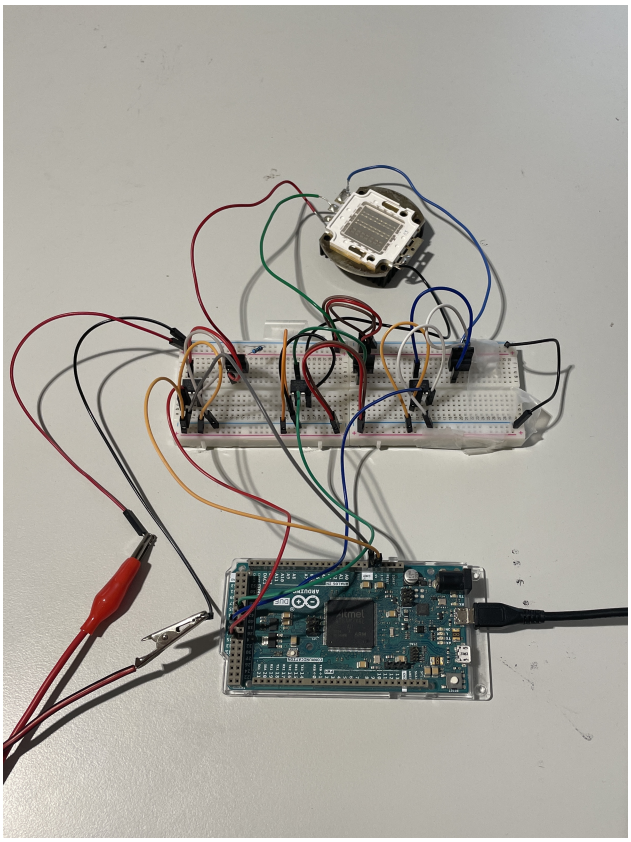


Figure 4.4: **The Control Circuit for High Power LED.**

controlled by Arduino DUE board. As shown in Figure 4.4, the control circuit allows for precise adjustment of the LED's blinking frequency.

For example, the images in Figure 4.6 are captured by the mobile phone camera, with frequency of 10 kHz and 5 kHz, which contain more stripes than the low power LED with small cover in Figure 4.2. The LED-to-Camera distance for both frequencies is 2 m.

The high power LED with large cover is designed to simulate scenarios with greater light intensity and increased LED-to-Camera distances. The brightness and coverage area of the LED is crucial for maintaining effective signal transmission in enriched environments. By using high power LED with large cover, the proposed method's performance in environments that require strong illumination and long-range communication, ensuring the robustness and adaptive of the proposed approach.

Receiver: Mobile Phone Camera

The model number of the mobile phone served as the receiver of this testbed is HUAWEI EML-AL00. When the image is taken, the camera's ISO is set to 50, and the exposure duration is 1/10,000 s, which is the amount of time it takes



Figure 4.5: **LED Cover for the High Power LED.**

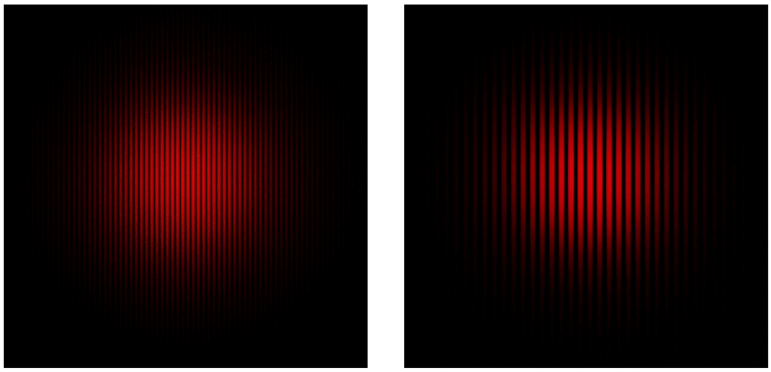
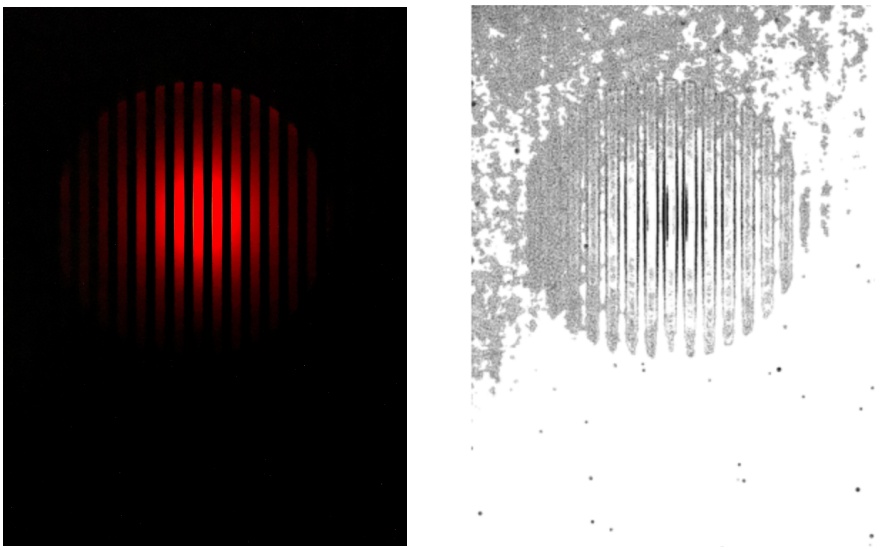


Figure 4.6: **(a) Blinking High Power LED with frequency = 10 kHz; (b) Blinking Hight Power LED with frequency = 5 kHz**

to scan a row of image. These settings are the lowest ISO and the smallest exposure duration available on this device.

Low ISO ensures sharper images are captured with less noise interference, which improves the accuracy of signal decoding. LED flashes are usually stronger, and using a low ISO will avoid overexposing the image and retain more detail. This is crucial for accurately detecting and decoding the information transmitted by the LED. Figure 4.7 (a) is the original image of blinking LED with ISO = 400. Figure 4.7 (b) is the binarized version of this image. This binarized version has much ambient light noise interference, complicating the accurate detection of the LED stripes. Under these conditions, it's hard to distinguish between the LED's signals and the background noise, making it difficult to effectively decode the transmitted data. Therefore, low ISO improves the precision of the stripe detection and the reliability of this system.



(a) (b)

Figure 4.7: **(a) Original image of LED with ISO = 400; (b) Binarized image.**

Scene Construction

The scene construction of the implementation is shown in Figure 4.8. One LED controlled by the Arduino DUE board is covered with a large cover and held in place, which is the transmitter. The receiver is the camera of a mobile phone, which is horizontal to the LED and at a certain distance from the LED. The relative movement between the LED and the camera is realized by moving the phone up and down, left and right. Rolling shutter images are captured and converted to binary signal by processing.

4.1.3 Software

Hardware Control

The three channels of the RGB LED (red channel R, green channel G, and blue channel B) can be controlled individually to achieve different colors of light output. To control these channels, we connected each channel separately to specific GPIO (General Purpose Input Output) pins on the Arduino Due. The R channel is connected to GPIO pin 35 for high power LED and 38 for low power LED, the G channel is connected to GPIO pin 42 for both high power and low power LEDs, and the B channel is connected to GPIO pin 39 for high power LED and 34 for low power LED.

Since our proposed method focuses on recouping the region of interest to improve the data transfer rate in moving scenes, the effect of different colors of light on streak detection is not taken into account. In this testbed, we use red light. An Arduino program is used to control the LED, which makes it blink continuously at a certain frequency. The LED is controlled to blink at different frequencies for the evaluation.

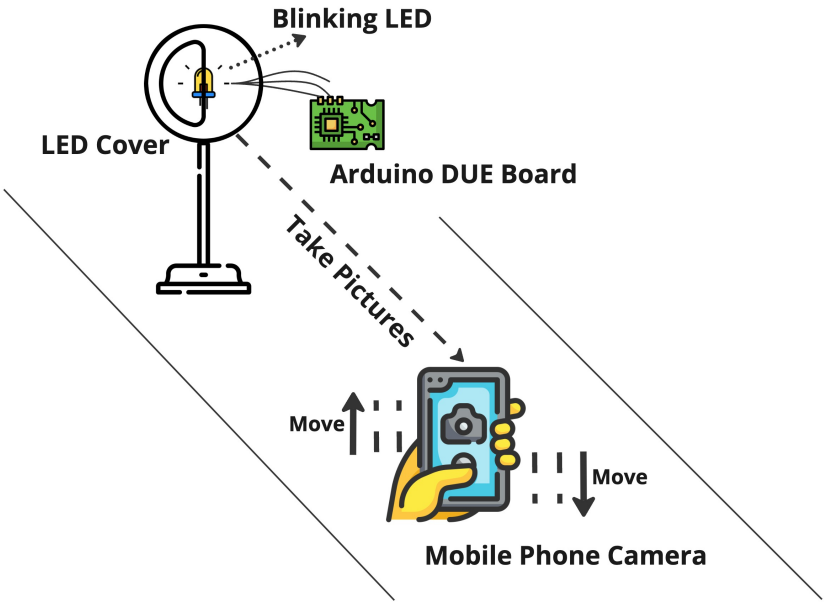


Figure 4.8: **Illustration of the evaluation setup.**

Upper Layer Processing

The processing of the captured rolling shutter images consists of the following functions.

Function of getting adaptive threshold. Within the processing range obtained from the outer square of the outer circle of the maximal stripe contour, the Cumulative Distribution Function (CDF) is computed, and the second derivative of its smoothed curve is obtained. A pixel value greater than 15 and letting the value of the second derivative be -0.0001 is served as the adaptive threshold.

Function of finding midpoints of stripes. Based on the adaptive threshold obtained in the function of getting adaptive threshold, this function obtains the upper and lower ends of a stripe, and takes the midpoint of the two ends as the midpoint of the stripe.

Function of Linear fitting for the midpoints. This function linearly fits the midpoints of all the stripes obtained in the previous function. Then it finds the points on the linear fit line whose value is the previously calculated adaptive threshold, to get the edges of the LED in the image.

Function of getting processing radius and center. In this function, the midpoint of the two points on the linear fit line with the value of adaptive threshold obtained in the function of Linear fitting for the midpoints is considered as the center of the new processing range. Half of the horizontal distance between these two points is used as the radius of the new processing range.

Function of converting pixel values to binary signal In this function, the sum of the pixel values in each column is calculated within the new processing range. According to local means, the sum of the pixel values in each column is converted to binary signal, and the lengths of 0s and 1s are the widths of the dark and bright stripes.

Based on these several main functions, the number of detectable rolling shutter stripes is obtained.

4.2 Testbed Evaluation

This section presents a comprehensive evaluation of the proposed method under different conditions, including for different LED blinking frequencies, LED-to-Camera distances, ambient lights, moving directions, and moving speeds. Based on the testbed of Section 4.1, the rolling shutter images under various conditions are captured by the mobile phone camera, which are processed by the proposed method to detect the stripes and then converted into binary signals.

Since this proposed method mainly addresses the loss of stripe detection caused by mobile scenarios and low stripe brightness at the edges of LEDs with large covers, which is more apparent using high power with large cover, for more visible processing results, the evaluation is mainly based on the high power LED with large cover of Section 4.1.2.

4.2.1 Different LED Blinking Frequencies

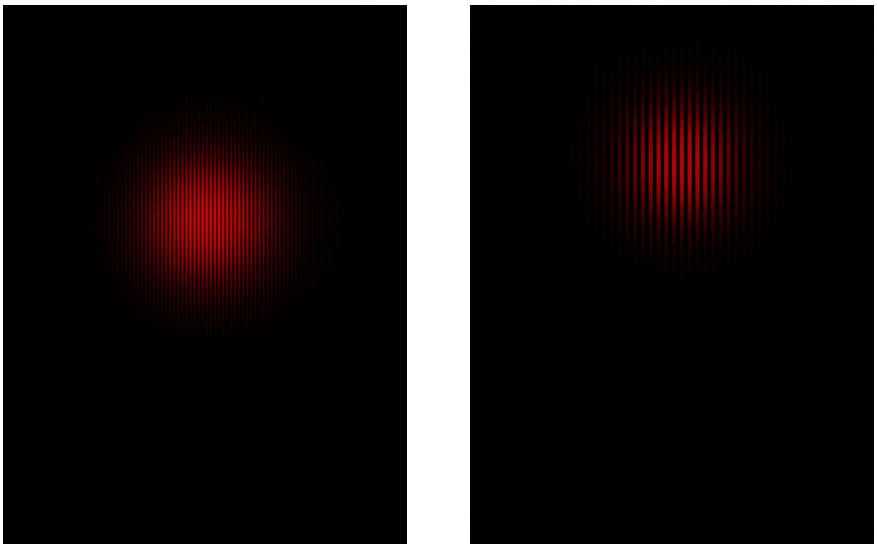
As described in Section 3.4.1, for high frequency rolling shutter images, such as 10 kHz, the stripe edges cannot be accurately detected with the state-of-the-art method. This problem can be solved by utilizing the approach of converting pixel information into binary signal in Section 3.4, which indicates that the results of this approach of decoding the data is remarkable.

In order to evaluate our proposed method of recouping region of interest of Section 3.3 effectively and clearly, we process the results before and after applying this method using the approach of Section 3.4 to convert pixel information within the processing range into binary signals for stripe detection.

In this section, different LED blinking frequencies are tested, including frequencies of 10 kHz, 6667 Hz and 5 kHz. All of these tests for different frequencies are conducted under the following conditions, a stationary scenario, a distance of 0.5 meters from the LED to the Camera, and totally dark ambient light. In Figure 4.9, (a) is the original image of frequency of 10 kHz, (b) is the original image of frequency of 5 kHz. The width of the rolling shutter stripes in the image with a frequency of 10 kHz is narrower, theoretically one-half the width of 5 kHz.

Figure 4.10 (a) is the signal plot of the sum of pixel values before recouping the region of interest, where the blue line is the original signal, the orange line is the smoothed signal and the red dotted line is the local means. Based on the smoothed signal and local means, we can get the binary signal of the rolling shutter stripes using the approach of Section 3.4, shown in Figure 4.10 (b). The lengths of continuous *0*s and *1*s are the stripe width, and the points where the signal jumps are the edge of the stripes. According to the binary signal plots, for frequency of 10 kHz, 53 stripes can be detected. The number of stripes that can be detected is 33 with frequency of 6667 Hz. And for frequency of 5 kHz, the stripe number is 23.

After applying the proposed method of recouping the region of interest, the signal plot is shown as Figure 4.11 (a), and (b) is the binary signal converted from it. For 10 kHz, 81 stripes can be detected. The stripe number is 51



(a) (b)

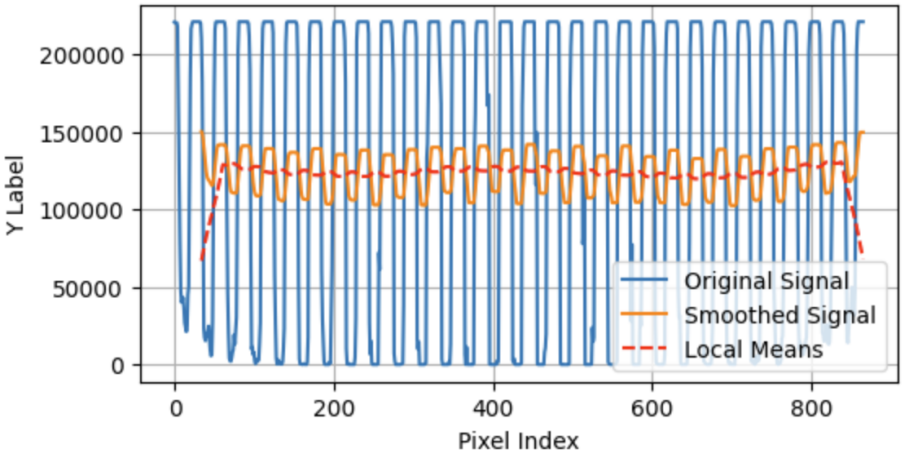
Figure 4.9: **Captured images at different TX frequencies: (a) Original image of frequency=10 kHz; (b) Original image of frequency=5 kHz.**

Table 4.1: **Detected Stripe Number of High Power LED with Different Frequencies. State: stationary; Distance from LED to Camera: 0.5 meters; Ambient Light: totally dark environment.**

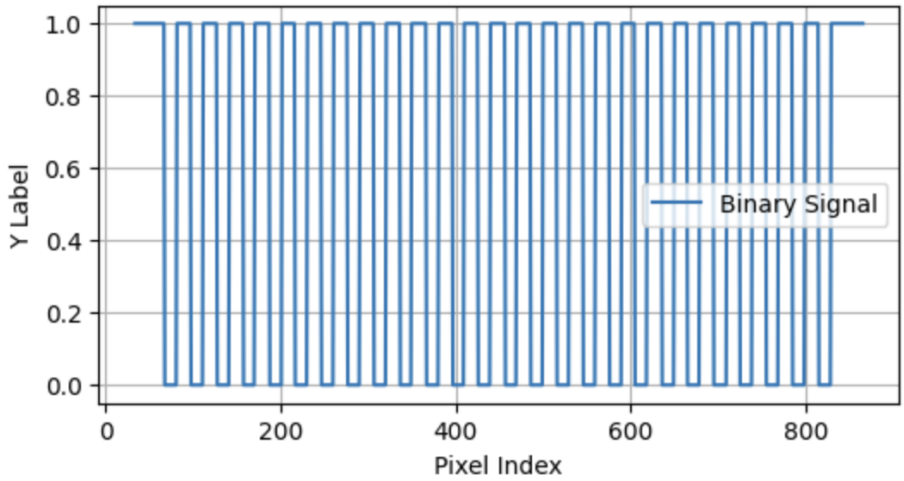
Conditions	Frequency	10 kHz	6667 Hz	5 kHz
Detected Stripe Numbers	Original	53	33	25
	Our method	85	51	37
	Gain	1.60×	1.55×	1.48×

with 6667 Hz, which is 35 for 5 kHz. The conditions and results are shown in Table 4.1, the gains are 1.53×, 1.55×, and 1.52×, respectively.

For different frequencies, the application of our proposed method has significant results, and the gains are basically close. Since the width of the stripes is inversely proportional to the frequency, the number of stripes detected in the same range is proportional, in theory. However, for wide stripes, the luminance range at the left and right ends of the bright stripes spans a wide range, especially near the edges of the LEDs. In the proposed method, the way of recouping the region of interest, i.e., expanding the processing range, is to determine the boundaries of the processing range by setting an adaptive threshold. This processing for wide stripes may make the boundary is just through the bright stripes and fail to detect the stripes located on the boundary, resulting in a reduced gain. That's why the gain of low frequency is not as good as the high frequency's, in Table 4.1.



(a)



(b)

Figure 4.10: **(a) Signal plot of the sum of pixel values before recouping the region of interest; (b) Binary signal.**

4.2.2 Different LED-to-Camera Distances

For different LED-to-Camera distances the effectiveness of our proposed method is also evaluated, including distances of 0.5 meters, 1 meter, 5 meters. Other conditions for this test are as follows: frequency of 5 kHz, stationary scenario and totally dark ambient light. Figure 4.15 (b) is the original image of the distance of 0.5 meters, Figure 4.12 (a) and (b) represent the images taken at distances of 1 meter and 5 meters, respectively.

It is obvious that the number of stripes gradually decreases as the distance gets longer. For the rolling shutter image captured at the distance of 5 meters, as shown in Figure 4.12 (b), the stripes almost disappear, this is due to the LEDs being too small on the image as well as being too low in brightness due to the distance. In conditions of different distances, we compare the results before and after applying our proposed method of recouping the region of interest. The

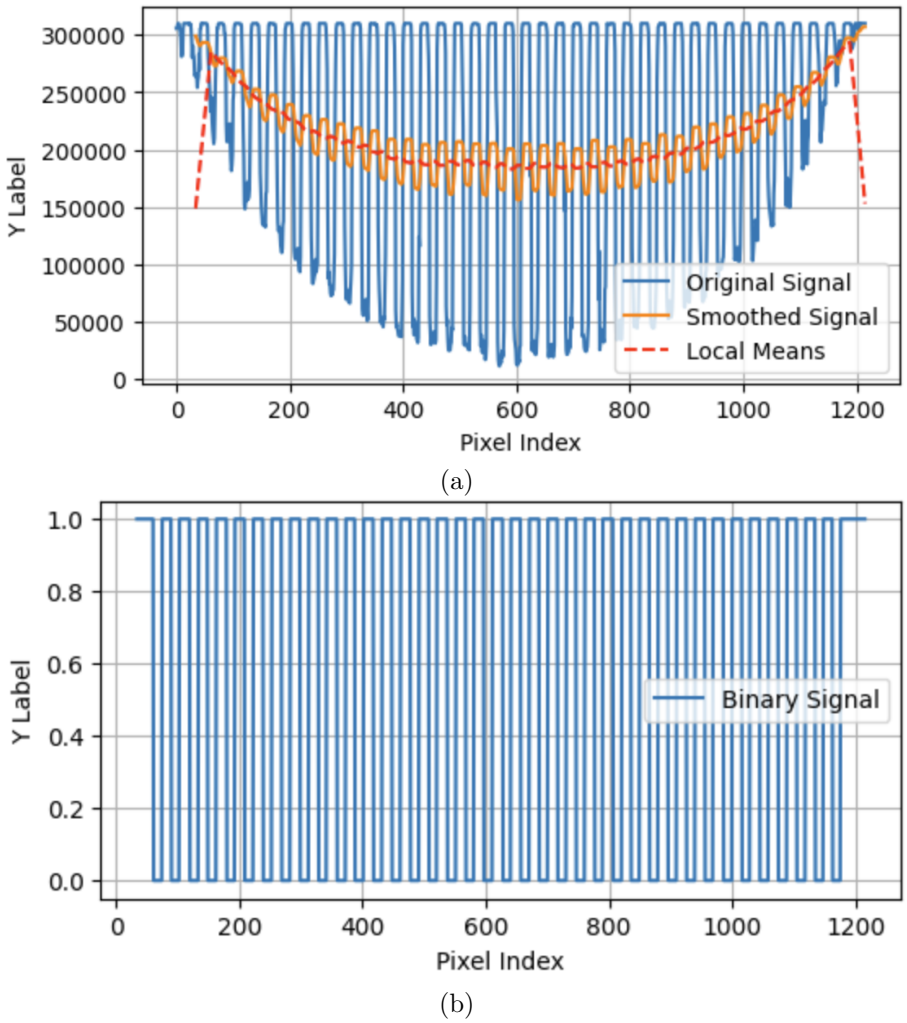
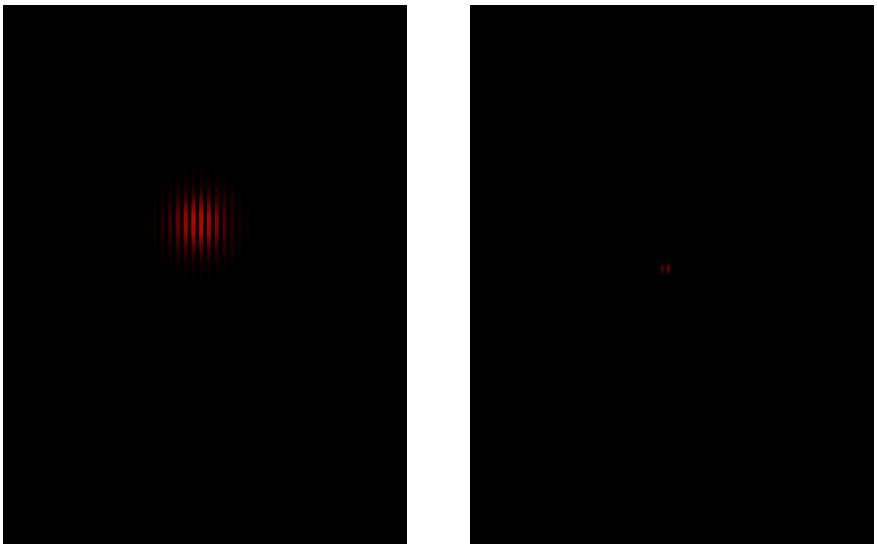


Figure 4.11: **(a) Signal plot of the sum of pixel values after recouping the region of interest; (b) Binary signal.**

same method in Section 3.4 is used to obtain the number of stripes before and after application.

The processing results are shown in Table 4.2. The gains are 1.48× and 1.24× for distances of 0.5 meters and 1 meter. For the scenario at a distance of 5 meters, neither the state-of-the-art method nor our proposed method can detect the stripes. It can be concluded that the effect of our proposed method is weaker for scenarios where the LED-to-Camera is farther away. The smaller LEDs in the image and the reduced brightness due to the further distance lead to this processing effect.



(a) (b)

Figure 4.12: **Evaluation at different TX-RX distance: (a) Original image of distance = 0.5 m; (b) Original image of distance = 1 m.**

Table 4.2: **Detected Stripe Number of High Power LED with Different Distances. Frequency: 5 kHz; State: stationary; Ambient Light: totally dark environment.**

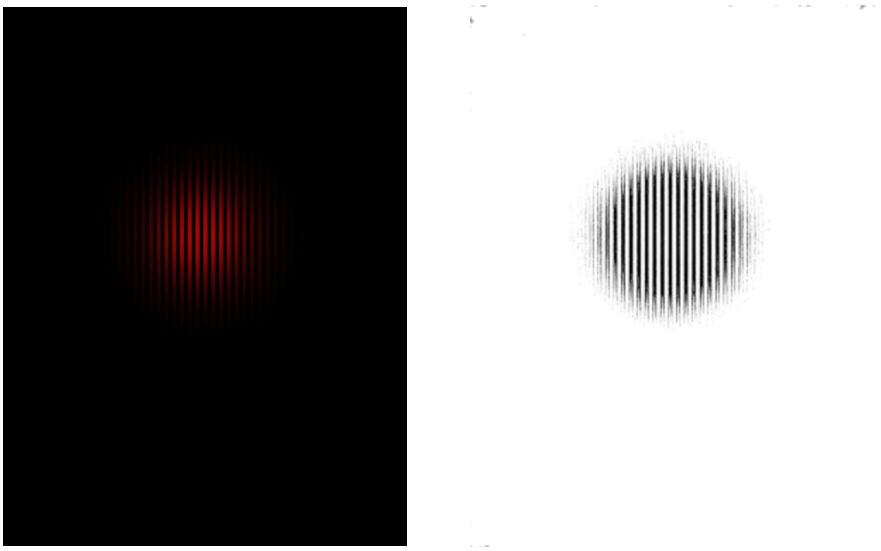
Conditions	Distance	0.5 m	1 m	5 m
Detected Stripe Numbers	Original	25	17	None
	Our method	38	21	None
	Gain	1.48×	1.24×	None

4.2.3 Different Ambient Light Conditions

For different application scenarios, the brightness of ambient light is also an important factor to be considered. Therefore, our proposed method is also evaluated under different ambient lights. We test the proposed method under fully-dark environment, indoor light and strong sunlight. The other conditions are frequency of 5 kHz, stationary scenario, distance of 0.5 meters. Figure 4.13 (a) is the original image taken in the totally dark environment. This image is converted to a binary image in order to facilitate subsequent processing and to see the effects of different ambient lights more clearly, as shown in Figure 4.13 (b). Figure 4.14 and Figure 4.15 are the images of indoor light and strong sunlight.

It is evident that the image captured in complete dark environment contains very little noise, while the image taken under indoor lighting shows some noise, and the image taken in bright sunlight appears with significant noise.

We also used the approach in Section 3.4 for the detection of the stripes before and after the application. The processing results are as shown in Figure 4.3. The gains under totally dark environment and indoor light are basically the same.



(a) (b)

Figure 4.13: **Evaluation in a dark environment: (a) Original image of totally dark environment; (b) Binarized image.**

Table 4.3: **Detected Stripe Number of High Power LED with Different Ambient Light. Frequency: 5 kHz; State: stationary; Distance: 0.5 meters.**

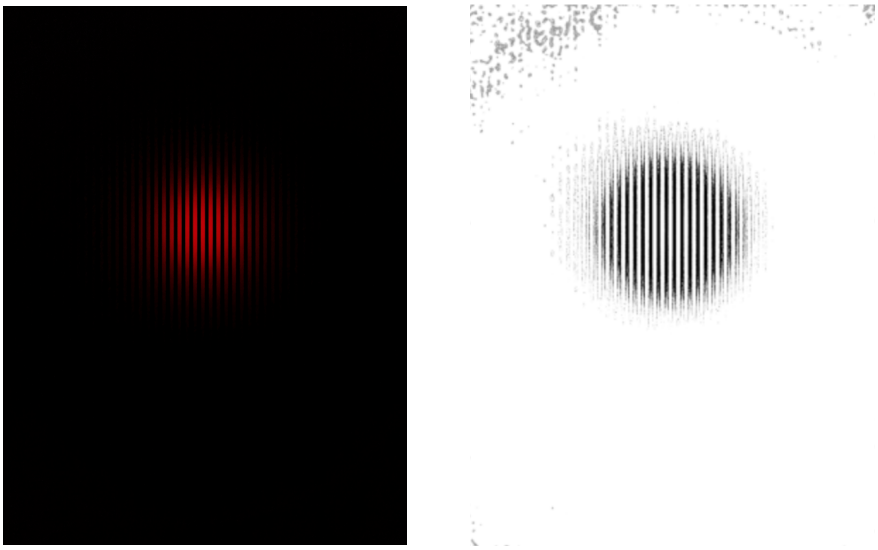
Conditions	Ambient Light	Dark	Dim	Bright
Detected Stripe Numbers	Original	23	23	None
	Our method	35	35	None
	Gain	1.52×	1.52×	None

For images taken under strong sunlight, the situation is similar to the results taken with high ISO in Figure 4.7, and neither the state-of-the-art method nor our proposed method can detect the rolling shutter stripes properly. This is due to excessive noise in the images. Overall, for totally dark environments as well as ambient light with low brightness, our method can perfectly recoup the region of interest, i.e., extend the processing range, as well as detect rolling shutter stripes. However, the proposed method does not work for very bright and intense sunlight conditions.

4.2.4 Different Moving Directions

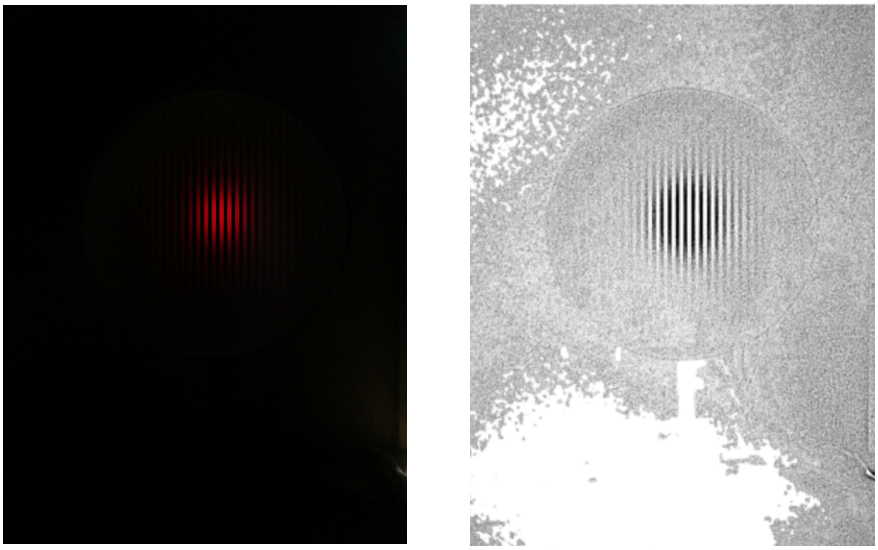
Considering that the application scenarios may contain different motion directions, the processing effect of our proposed method for different motion directions is also evaluated. In this case, we test in four scenarios of upward, downward, leftward, and rightward motions. The other conditions are frequency of 5 kHz, distance of 0.5 meters, totally dark environment and moving speed of 8-12 m/s.

The camera of the device used in Testbed scans from left to right, line by line,



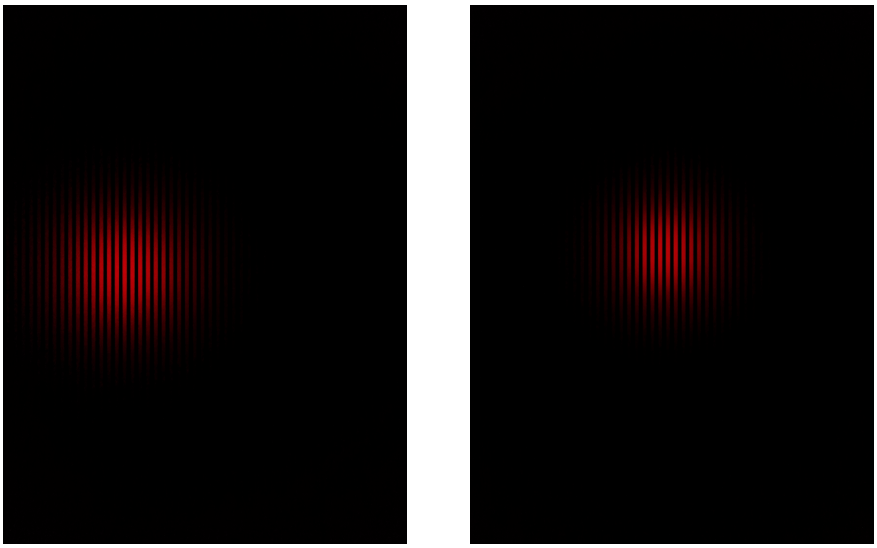
(a) (b)

Figure 4.14: **Evaluation in an indoor environment: (a) Original image of indoor light; (b) Binarized image.**



(a) (b)

Figure 4.15: **Evaluation under strong ambient light: (a) Original image of strong sunlight; (b) Binarized image.**

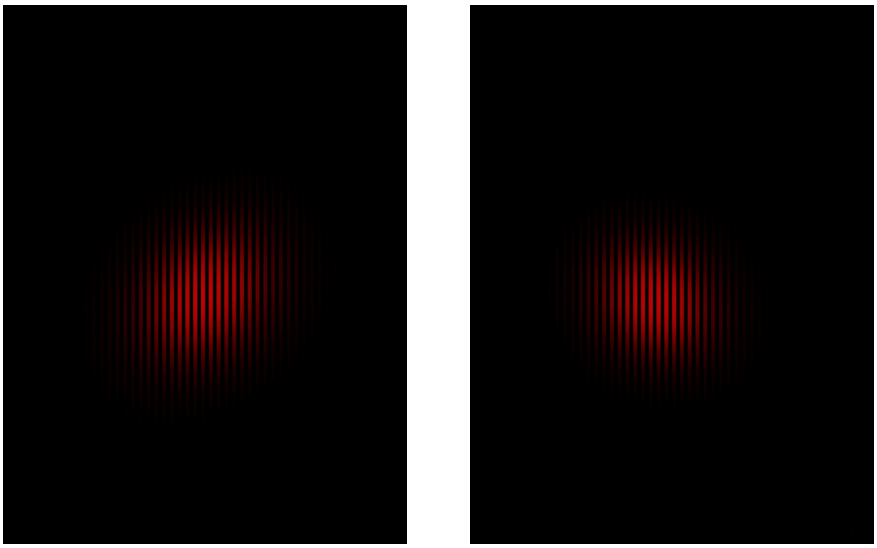


(a) (b)

Figure 4.16: **Evaluation under the left/right movement of the RX: (a) Original image of moving left; (b) Original image of moving right.**

to get the image. Therefore, if the camera moves to the left relative to the LED, its progressive scanning results in the overall profile of the LED being stretched horizontally. As shown in Figure 4.16 (a), it's the original image captured by moving the camera to the left, which contains more stripes than that of stationary scenarios. Conversely, if the camera moves to the right relative to the LED, progressive scanning results in the overall profile of the LED being compressed horizontally. Figure 4.16 (b) is the original image of moving to the right, including less stripes than that of stationary scenarios. For upward and downward motion, the width of the LED in the horizontal direction of the captured image does not change, but is deformed in the vertical direction. Figure 4.18 (a) and (b) are the original images captured when the camera moves down and up, respectively. For downward movement, the midpoint of the stripe gradually moves upward, for upward movement it is the opposite.

The detected stripe numbers before and after applying our proposed method are shown in Table 4.4. Under the condition of moving to the right, both the numbers of stripes using the state-of-the-art method and the proposed method are less than in stationary scenarios, and the gain is smaller. For moving to the right, the stripe numbers of using two methods are both more than that of stationary scenarios, which gains more with the application of the proposed method. In the state-of-the-art method, the largest stripe contour is selected, and the outer square of its circumcircle is treated as the processing range. For scenarios that move quickly up and down, this method may result in the stripes not being included, which are close to the left and right ends of the LED in the figure and away from the horizontal line through the LED center, i.e., beyond the upper or lower processing range. Therefore, for mobile scenarios of quickly moving up and down, the numbers of detected stripes are smaller than in stationary scenarios. After applying the proposed method in Section 3.3,



(a) (b)

Figure 4.17: **Evaluation under the up/down movement of the RX: (a) Original image of moving down; (b) Original image of moving up.**

Table 4.4: **Detected Stripe Number of High Power LED with Different Moving Directions. Frequency: 5 kHz; State: mobile; Distance: 0.5 meters; Ambient Light: totally dark environment.**

Conditions	Moving Direction	Right	Left	Down	Up
	Moving Speed	8-12 m/s	8-12 m/s	8-12 m/s	8-12 m/s
Detected Stripe Numbers	Original	23	25	21	21
	Our method	33	39	35	33
	Gain	1.43×	1.56×	1.67×	1.57×

more stripes can be detected, which are basically the same as that of stationary scenarios, and the gains are larger. In conclusion, the proposed method can successfully solve the problem caused by the LED deformation in mobile scenarios.

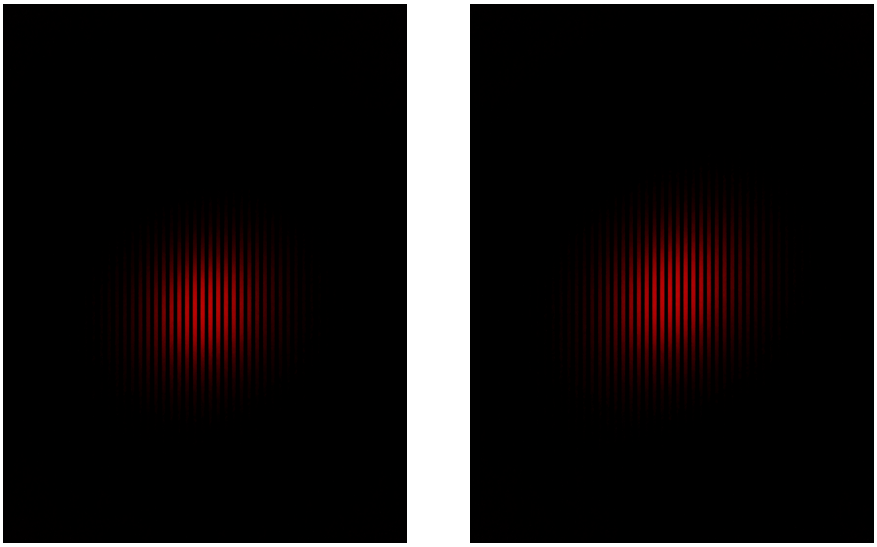
4.2.5 Different Moving Speeds

Different movement speeds are involved in the application scenarios. Therefore, the application of the proposed method to different movement speeds scenarios is evaluated. We tested three speed ranges: stationary (0 m/s), moderate motion (2-5 m/s), and fast motion (8-12 m/s), with frequency of 5 kHz, distance of 0.5 meters, totally dark environment and moving direction of upward.

The results are shown in Table 4.5. In slow-moving scenarios, i.e., speeds in the range of 2-5 m/s, the processing results of the state-of-the-art method are consistent with that of stationary scenarios. This is because due to the slow movement, the stripes near the edges are not beyond the upper and lower edges of the processing range obtained based on the largest stripe contour. After the

Table 4.5: **Detected Stripe Number of High Power LED with Different Moving Speeds. Frequency: 5 kHz; Distance: 0.5 meters; Ambient Light: totally dark environment.**

Conditions	State	Stationary	Mobile	Mobile
	Moving Direction	Down	Down	Down
	Moving Speed	0	2 - 5 m/s	8 - 12 m/s
Detected Stripe Numbers	Original	23	23	21
	Our method	35	35	35
	Gain	1.52×	1.52×	1.67×



(a) (b)

Figure 4.18: **Evaluation under moving down the RX at different speed: (a) Original image of moving down at speed of 2-5 m/s; (b) Original image of moving down at speed of 8-12 m/s.**

application of the proposed method in Section 3.3, the effect of enlarging the processing range and detecting more stripes is the same as that of stationary scenarios. For fast motion, the number of stripes detected by using the state-of-the-art method is smaller than that of slower motion. However, with the application of the proposed method, same number of stripes can be detected as that under slower motions. Therefore, it can be concluded that our proposed method is more effective in fast mobile scenarios.

Chapter 5

Conclusion

In this thesis, a method that can improve the signal transmission rate of LED-to-Camera communication in moving scenarios has been proposed. It has addressed two main problems met in mobile scenarios. First, the stripe contours near the edge of the cover are too dark to detect due to the LED cover being too large. Second, it has solved the problem of missing part of the stripe detection due to the deformation of the overall LED shape in the image caused by motion.

Throughout the procedure, we find a suitable solution for how to determine the processing range accurately in mobile scenarios. In this solution, first, the midpoint of each stripe is determined by identifying the brightest point on the stripe. Because the shooting time is very short, the relative motion of the LED-to-Camera during this period can be regarded as a uniform linear motion. The midpoints of the stripes that can be detected are then linearly fitted, which is considered to pass through the midpoints of all stripes. Next, after various attempts, the method of setting the adaptive threshold is finalized. Based on the threshold, the edges of the expanded processing range are obtained, which recoups the region of interest. We also find a new more efficient and accurate way to determine the edges of the stripes by converting pixel values to binary signal and considering the length of 0 and 1 as the widths if dark and bright stripes.

Based on the testbed, the proposed methodology is evaluated from a variety of perspectives. First, by processing the rolling shutter images captured with high-frequency flickering LED, we have demonstrated the applicability of our proposed method for detecting stripe boundaries to such scenarios. Then we have evaluated the proposed method of recouping the region of interest under the conditions of different LED blinking frequencies, different LED-to-Camera distances, different ambient lights, different moving directions, and different moving speeds. It can be concluded from the tests that our proposed method has great processing results for the above different scenarios. And it is more effective for the scenarios with motion perpendicular to and opposite to the scanning direction, very fast motion, and high flickering frequency.

Overall, the proposed method has been proven to be effective, which is instrumental in obtaining more complete information transmitted by blinking LED in mobile scenarios.

5.1 Future Work

Under Stronger Ambient Light

In the evaluation, it can be seen that our proposed method is not applicable under strong ambient light, which is due to the fact that the strong light leads to a large amount of noise in the captured image. For the future work, scenarios with strong ambient light will be solved, by finding a suitable and effective way to filter images.

Complete Packet of Data Transmission

The proposed method can successfully detect the rolling shutter stripe information, but it does not form a complete packet used to transmit data, including preamble, protocol, etc. For different frequencies, different distances from the LED to the camera, etc., the data that can be conveyed in an image is different. Therefore, a complete packet is necessary to transmit complete data and to ensure its accuracy. In future work, the complete and adaptive packet for our proposed method will be designed.

Data Transmission by Video

LED-to-Camera communication can transmit data using video in addition to image, where each frame is a separate packet. But for a camera recording video, there is a certain amount of time gap between each frame, meaning that the camera is not capturing during that time. Therefore, for transmitting data via video, there is no continuity between each frame. For each frame of data to be concatenated together, on the basis of the complete packet, some relevant mechanisms need to be designed in the future.

Bibliography

[1] Abdelmoujoud Assabir, Jamal Elmhamdi, Ahmed Hammouch, and Abdellatif Akherraz. Application of li-fi technology in the transmission of the sound at the base of the pwm. In *2016 International Conference on Electrical and Information Technologies (ICEIT)*, pages 260–265, 2016.

[2] Chia-Wei Chen, Chi-Wai Chow, Yang Liu, and Chien-Hung Yeh. Efficient demodulation scheme for rolling-shutter-patterning of cmos image sensor based visible light communications. *Opt. Express*, 25(20):24362–24367, Oct 2017.

[3] Shih-Hao Chen and Chi-Wai Chow. Color-shift keying and code-division multiple-access transmission for rgb-led visible light communications using mobile phone camera. *IEEE Photonics Journal*, 6(6):1–6, 2014.

[4] Bhawana Chhaglani, Abhay Sheel Anand, Nakul Garg, and Ashwin Ashok. Evaluating led-camera communication for drones. In *Proceedings of the Workshop on Light Up the IoT*, LIOT '20, page 18–23, New York, NY, USA, 2020. Association for Computing Machinery.

[5] Chi-Wai Chow, Chung-Yen Chen, and Shih-Hao Chen. Enhancement of signal performance in led visible light communications using mobile phone camera. *IEEE Photonics Journal*, 7(5):1–7, 2015.

[6] Dr Colin Coates and Dr Ines Juvan-Beaulieu. Rolling shutter vs global shutter scmos camera mode, 2020-12.

[7] Christos Danakis, Mostafa Afgani, Gordon Povey, Ian Underwood, and Harald Haas. Using a cmos camera sensor for visible light communication. In *2012 IEEE Globecom Workshops*, pages 1244–1248, 2012.

[8] Veronique Georlette, Florian Piras, Cristo Jurado–Verdu, Sebastien Bette, Nicolas Point, and Veronique Moevaert. Content triggering system using tricolor led strips and optical camera communication in rolling shutter mode. In *2021 Third South American Colloquium on Visible Light Communications (SACVLC)*, pages 01–06, 2021.

[9] Wahyu Hendra Gunawan, Yang Liu, Chien-Hung Yeh, and Chi-Wai Chow. Color-shift-keying embedded direct-current optical-orthogonal-frequency-division-multiplexing (csk-dco-ofdm) for visible light communications (vlc). *IEEE Photonics Journal*, 12(5):1–5, 2020.

[10] Y. Hokazono, A. Koizuka, G. Zhu, M. Suzuki, Y. Narusue, and H. Morikawa. Iotorch: Reliable led-to-camera communication against inter-frame gaps and frame drops. *IEEE Transactions on Mobile Computing*, 20(02):550–564, feb 2021.

[11] Pengfei Hu, Parth H. Pathak, Huanle Zhang, Zhicheng Yang, and Prasant Mohapatra. High speed led-to-camera communication using color shift keying with flicker mitigation. *IEEE Transactions on Mobile Computing*, 19(7):1603–1617, 2020.

[12] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta. Luxapose: indoor positioning with mobile phones and visible light. In *Proceedings of the ACM MobiCom*, page 447–458, 2014.

[13] Pradipta Kurniawan, Kris Sujatmoko, and Brian Pamukti. Performance of ook-rz and nrz modulation techniques in various receiver positions for li-fi. In *2019 IEEE International Conference on Signals and Systems (ICSigSys)*, pages 173–177, 2019.

[14] Hui-Yu Lee, Hao-Min Lin, Yu-Lin Wei, Hsin-I Wu, Hsin-Mu Tsai, and Kate Ching-Ju Lin. Rollinglight: Enabling line-of-sight light-to-camera communications. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '15, page 167–180, New York, NY, USA, 2015. Association for Computing Machinery.

[15] Kevin Liang, Chi-Wai Chow, and Yang Liu. Rgb visible light communication using mobile-phone camera and multi-input multi-output. *Opt. Express*, 24(9):9383–9388, May 2016.

[16] Bangjiang Lin, Xuan Tang, Zabih Ghassemlooy, Chun Lin, and Yiwei Li. Experimental demonstration of an indoor vlc positioning system based on ofdma. *IEEE Photonics Journal*, 9(2):1–9, 2017.

[17] Huichao Lv, Lihui Feng, Aiying Yang, Peng Guo, Heqing Huang, and Shufen Chen. High accuracy vlc indoor positioning system with differential detection. *IEEE Photonics Journal*, 9(3):1–13, 2017.

[18] Myoung-geun Moon and Su-il Choi. Indoor position estimation using image sensor based on vlc. In *2014 International Conference on Advanced Technologies for Communications (ATC 2014)*, pages 11–14, 2014.

[19] Huy Nguyen and Yeong Min Jang. Design and implementation of rolling shutter mimo-ofdm scheme for optical camera communication system. In *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 798–800, 2021.

[20] Trang Nguyen, Chang Hyun Hong, Nam Tuan Le, and Yeong Min Jang. High-speed asynchronous optical camera communication using led and rolling shutter camera. In *2015 Seventh International Conference on Ubiquitous and Future Networks*, pages 214–219, 2015.

[21] Guanchong Niu, Junpeng Zhang, Sheng Guo, Man-On Pun, and Chung Shue Chen. Uav-enabled 3d indoor positioning and navigation based on vlc. In *ICC 2021 - IEEE International Conference on Communications*, pages 1–6, 2021.

[22] Sung Hyun Oh and Jeong Gon Kim. Vlc positioning by dnn via wknn in indoor environment. In *2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 450–453, 2022.

[23] R. Othman, A. Gaafar, L. Muaaz, and M.H. Elsayed. A hybrid rss+aoa indoor positioning algorithm based on visible light communication. In *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pages 1–5, 2021.

[24] Christoforos Papaioannou, Marina E. Plissiti, Yiorgos Sfikas, Georgios Papatheodorou, Simon-Ilias Poulis, Aristides Efthymiou, and Yiorgos Tsiatouhas. Signal decoding in an nlos vlc system with the presence of anti-reflective obstacles. In *2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 303–309, 2022.

[25] Niranjini Rajagopal, Patrick Lazik, and Anthony Rowe. Visual light landmarks for mobile devices. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, pages 249–260, 2014.

[26] Richard Roberts. Undersampled frequency shift on-off keying (ufsook) for camera communications (camcom). pages 645–648, 05 2013.

[27] Hamid Sharifi, Akshaya Kumar, Fakhrul Alam, and Khalid Mahmood Arif. Indoor localization of mobile robot with visible light communication. In *2016 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pages 1–6, 2016.

[28] Jin Wang, Haoxu Li, Xiaofeng Zhang, and Rangzhong Wu. Vlc-based indoor positioning algorithm combined with ofdm and particle filter. *China Communications*, 16(1):86–96, 2019.

[29] Wei-Chung Wang, Chi-Wai Chow, Chia-Wei Chen, Hsiang-Chain Hsieh, and Yen-Ting Chen. Beacon jointed packet reconstruction scheme for mobile-phone based visible light communications using rolling shutter. *IEEE Photonics Journal*, 9(6):1–6, 2017.

[30] Dong-Fang Zhang, Yi-Jun Zhu, and Yan-Yu Zhang. Multi-led phase-shifted ook modulation based visible light communication systems. *IEEE Photonics Technology Letters*, 25(23):2251–2254, 2013.