

Delft University of Technology
Master's Thesis in MSC: Embedded Systems

Analysing the Performance and Stability of LED-to-Camera Links

Shashwat Verma



Analysing the Performance and Stability of LED-to-Camera Links

Master's Thesis in MSC: Embedded Systems

Embedded Software Section
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

Shashwat Verma
s.verma@student.tudelft.nl

19th December 2017

Author

Shashwat Verma (s.verma@student.tudelft.nl)

Title

Analysing the Performance and Stability of LED-to-Camera Links

MSc presentation

21st December 2017

Graduation Committee

Prof. Dr. K.G. Langendoen (chair) Delft University of Technology

Dr. M. A. Zuniga Zamalloa Delft University of Technology

Dr. C. Lofi Delft University of Technology

Abstract

Recently, Visible light communication systems have been gaining the interest of the research community due to their great potential for creating high-speed data links and providing an alternative to the crowded radio spectrum. As smartphones have become an integral part of life, it would be great if visible light communication could provide additional benefits like line-of-sight data transfer, via the smartphone. Recent studies have demonstrated the feasibility of using the light-to-camera communication to take advantage of the current VLC infrastructure.

The current systems present different approaches to establish the light-to-camera communication link. The performance of these links depends on many variables like the camera frame-rate, operational frequency, exposure time, etc., thus, their direct comparison is not possible. There is no common ground on which these state-of-the-art schemes can be evaluated. This thesis presents a model where these schemes are compared on a common platform.

The current systems are also very inflexible in terms of adjusting their performance based on the distance between the transmitter and the receiver. Creating a flexible light-to-camera communication system requires some form of synchronisation between the transmitter and the receiver. In this thesis, we also present a Mathematical model to optimise the reliability of the communication link with varying distances between the transmitter and the receiver. This model can also be used for calculating the performance of the communication link based on the aforementioned variables (i.e. camera frame-rate, operational frequency, read-out duration).

Preface

This thesis is submitted for the degree of Master of Science in Embedded Systems at the Delft University of Technology. The learning and outcome of this project is of great value for me, where I learnt a range of concepts, although to various depths. The past 10 months since I started this project have been like a roller coaster ride. The completion of this project is approaching. Now it is the time to express my thanks for those who have helped and guided me through the progress of this project.

Foremost, I would like to express my gratitude to my supervisor Marco Zuniga, for introducing me to the field of visible light communication and smartphones. This work couldn't have been possible without his generous support and motivation. His guidance was crucial for this thesis, and his knowledge and experience have helped me greatly with my writing and presentation skills. I would also like to thank Eric and the members of VLC group who were always available for discussions.

I would also like to thank all my friends in The Netherlands. We had a lot of good times together. Lastly, I would like to thank my parents, who have been with me through all the times, and given me their unconditional love and support.

Shashwat Verma

Delft, The Netherlands

31st December 2017

Contents

Preface	v
1 Introduction	1
1.1 Problem statement	2
1.2 Contribution	3
1.3 Thesis organisation	3
2 Background	5
2.1 Transmitter-Receiver of VLC	5
2.2 Rolling Shutter	6
2.3 Camera-based VLC	8
2.3.1 Summary	10
3 Overview	13
3.1 System Overview	13
3.2 Basic Functionalities	14
4 Transmitter	17
4.1 Flickering Effect	17
4.2 Encoding schemes	18
4.3 Hardware Implementation	21
4.4 Duty-Cycle Adjustment	22
5 Receiver	25
5.1 Light Sensor as Receiver	25
5.2 Camera as Receiver	26
5.2.1 Camera Parameters	26
5.2.2 Image Processing	28
5.2.3 Data Decoding	30
5.3 Implementation Details	33
5.3.1 Parameter Values	33
5.4 Modulation Frequencies	34
6 Mathematical Model	37

6.1	Distance Model	37
6.1.1	Distance calculation	37
6.1.2	Symbol Capacity Calculation	39
6.2	Packet Optimisation	40
7	Comparison of Encoding Methods	43
7.1	Manchester Encoding	43
7.2	FSK Encoding	44
7.3	Comparison	45
8	Evaluation and Results	47
8.1	Flickering Compensation	47
8.2	Distance model evaluation	49
8.2.1	Distance calculation	50
8.2.2	Symbol Capacity Calculation	51
8.2.3	Adapting to Variable Distance	53
8.3	Comparison of State of the Art schemes	55
9	Conclusions	57
9.1	Conclusions	57

Chapter 1

Introduction

Wireless communication is the fastest growing segment in the communication industry. The last two decades have witnessed significant advances in the field of wireless communication systems and they are rapidly becoming an integral part in everyday life. The most common example are cellular networks (GPRS) and Wi-Fi, whose popularity and usage has grown exponentially. Wireless communication uses radio waves for transmitting information across devices without utilising cables or any form of conductors. Radio waves have a pivotal role in wireless systems due to their ability to transmit information over short distances like Wi-Fi as well as long distances like satellite television.

One of the subsets of wireless communication is *OWC* (Optical wireless communication), which uses light (visible or infra-red) for information transmission in free space. Communication through light is not a new concept. Mankind has been using some form of light communication for centuries. In early days, light from big bonfires were used to signal events to distant observers. The British Royal Navy used a similar concept in mid 19th century. They used big sign lamps to transmit Morse code between navel vessels. Later in 1880, Alexander Bell and Charles Tainter developed *Photophone* (a wireless telephone), which could transmit audio signals in form of modulated light beams up to 200 meters.

In the last decade or so *Visible Light Communication* (VLC), a subset of optical wireless communications technology, has been gaining interest in the research community. VLC provides extra communication bandwidth without using the crowded *Radio Frequency* (RF) spectrum. VLC uses a light source (e.g. Light Emitting Diode (LED) or LED lamp) as a transmitter and a suitable detector (e.g. photodiode, reverse-biased LED or camera) as a receiver to interpret the light signal. LEDs are attractive as they are more energy efficient than incandescent lamps and are inexpensive. Existing

research recognises the critical role of visible light in communication as it provides plenty of unlicensed bandwidth (430 to 770 THz).

Data transmission in VLC systems can be achieved by performing a number of transitions between on (light emission) and off (no light emission) states of light source. The light sensors can interpret these transition patterns, which is used to create high-speed VLC links. Most of the VLC systems focus on turning an LED “ON” and “OFF” at high frequencies (above 120 Hz) with the light source maintaining a constant average intensity. This is done so that the modulation is not perceivable by humans and the light source appears to be switched “ON” continuously. Thus, VLC can be summarised as illumination plus communication.

Recent improvements in smartphones have led to its increased usage as wireless data receivers. Data from several studies [5], [8], [9] indicates that smartphones can also be used as VLC receivers by using its camera as depicted in Figure 1.1. This light-to-camera link can be utilised either for localisation or data transfer.

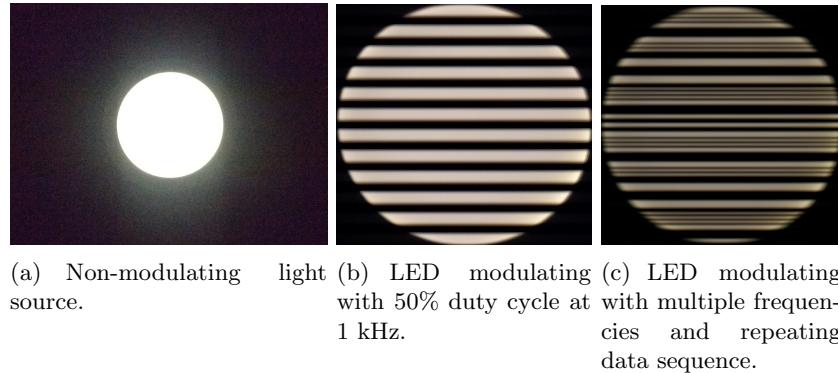


Figure 1.1: Image of light captured with the different modulation patterns. The camera is 50cm from the light source and exposure time is set to $161\mu s$.

1.1 Problem statement

Modern day smartphones have different hardware which dictates its capabilities. More specifically, they have variations in camera sensors (tuned for capturing image under various conditions), processing power and operating systems. These factors effect the flexibility and reliability of the communication link. Most of the current light-to-camera systems try to overcome this wide diversity by utilising high-end smartphones, as they tend to have similar hardware configurations.

Recent works like [5], [8], [9] present different approaches to establish the light-to-camera communication link. They use different modulation schemes, capture frame rates, decoding methods and hardware, which result in different throughputs generated by the system.

Moreover, these systems are designed to operate up to a certain distance using a fixed packet. This inflexibility results in a suboptimal throughput at different distances between the transmitter and the receiver. One work [11] tries to understand how distance affects the channel capacity. This highlights the need of a system that can optimise the data-rate automatically with respect to the distance.

In this thesis, we will try to overcome two problems:

1. In different works, what factor contributes more towards the throughput? Is it the smartphone or the data encoding method?
2. How to overcome the fixed distance limitation for light-to-camera link?

The goal of this project is to create a mathematical framework to analyse different encoding schemes and create a distance-adaptive light-to-camera link.

1.2 Contribution

This work presents a line-of-sight light-to-camera based VLC system and puts forward a design for a distance-adaptive system. This work gives rise to the following contributions:

- A channel optimisation technique based on the distance between the transmitter and the receiver in order to maximise the data-rate.
- A mathematical framework for analysing different encoding schemes based on the overhead and the operational frequency.
- Implementation of the existing encoding schemes and their comparison on the basis of data overhead and flickering.

1.3 Thesis organisation

Chapter 2 presents the possible transmitters and receivers for VLC and analyse different camera-based VLC systems. The following Chapter 3 presents a basic overview of the system. A comparison of the encoding schemes along with the transmitter's hardware implementation is presented in Chapter 4. The design of the receiver and data decoding pipeline

can be found in Chapter 5. Chapter 6 describes the proposed models for distance calculation and achieving an adaptable range. Further comparison of two modulation schemes using the mathematical model is presented in Chapter 7. Experimental results of the adaptable system is presented in Chapter 8. Lastly, the conclusion is presented in Chapter 9.

Chapter 2

Background

This chapter describes the basic building blocks and concepts required for understanding the work done in this thesis. Section 2.1 provides an overview of the possible transmitters and receivers in VLC systems and the quality of links between them. Section 2.2 explains the inner working of the rolling-shutter mechanism and its usage for receiving modulated signals. Section 2.3 presents the background work utilising light-to-camera link for the communication.

2.1 Transmitter-Receiver of VLC

Communication systems consists of three components; transmitter, receiver and a channel. VLC systems can use various light sources and light sensors according to the requirements of the application's environment.

Light sources which can switch states at high rate (i.e. transition between on and off) can be used as transmitters for VLC systems. Similarly, LED screens can also be used as transmitters as suggested by [7].

Any electronic device which can detect the presence or absence of visible light can be utilised as a VLC receiver. Most of the work is focused towards using photodiodes as receivers because of their fast response and high bandwidth. Works like [10] shows that regular LEDs can operate as receiver in reverse bias mode, but they have limited sensitivity as compared to photodiodes. Smartphone cameras can also be used to detect high frequency light patterns as discussed by [1].

The majority of the VLC research is focused towards creating efficient and high-speed data links. LED-photodiode links are used to achieve this

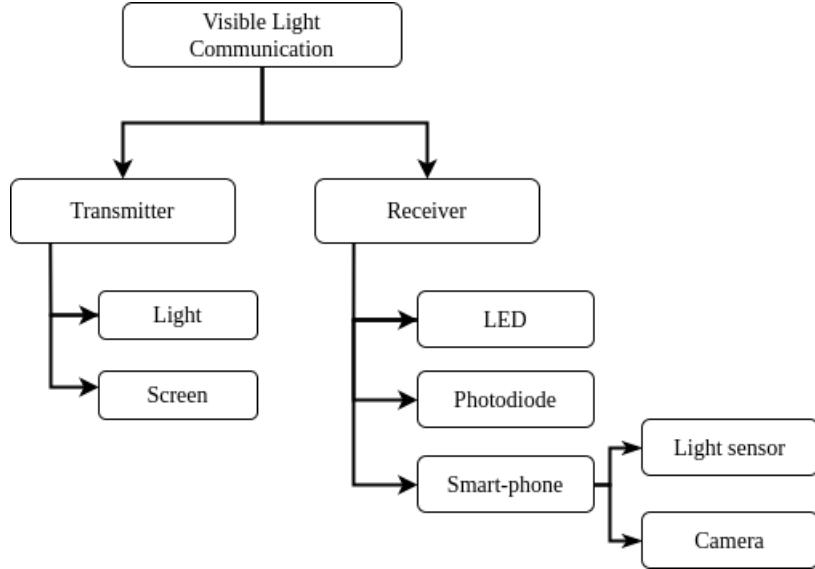


Figure 2.1: Receivers and transmitters for VLC.

as photodiodes provide fast-response and high bandwidth, that can support complex spectrally-efficient modulation schemes.

Prior works like [2], [3], [7] utilise LED screens for transmitting information to cameras. These techniques use barcodes or QR codes displayed on screens to encode information. HiLight [7] is a new form of a screen-camera link, which encodes data into pixels' translucency. This technique can efficiently transmit information without being obstructive to displayed image or video.

Smartphones can also be used as receiver in VLC systems. Danakis et al. [1], illustrate that the cameras of smartphones with the so called rolling shutter mechanisms can capture patterns from fast modulating light source. Utilising this mechanism, a camera can be used as a VLC receiver for data transfer and localisation. In this thesis, *we mainly focus on the camera of smartphones as VLC receivers*.

2.2 Rolling Shutter

Modern day cameras widely use two types of image sensors, Charge Coupled Devices (CCD) and Complementary Metal Oxide Semiconductors (CMOS). These two technologies have a number of similarities but one major distinction is the way each sensor exposes its pixels to light. CCD sensors often use the *Global Shutter* readout mode, where every pixel is exposed simultan-

uously at the same instant in time and then each pixel is read sequentially as shown in Figure 2.2(a). This mechanism helps in capturing a still image of a moving object. Figure 2.2(b) shows that, a moving fan appears still when the image is captured using the Global Shutter mechanism. On the other hand, CMOS sensors use the *Rolling Shutter* readout mode, where each individual row is exposed in a row-sequential way with fixed time delay. In Figure 2.2(c), it can be seen that the time delay is the same as the read-out time (red box in each row). This read-out time is synchronised such that only one row is available to be read at any instance. Due to this mechanism, there is significant time difference between the beginning of the exposure of the first and the last row, making them no longer simultaneous. This leads to a distortion in the captured image of a fast moving object as depicted in Figure 2.2(d).

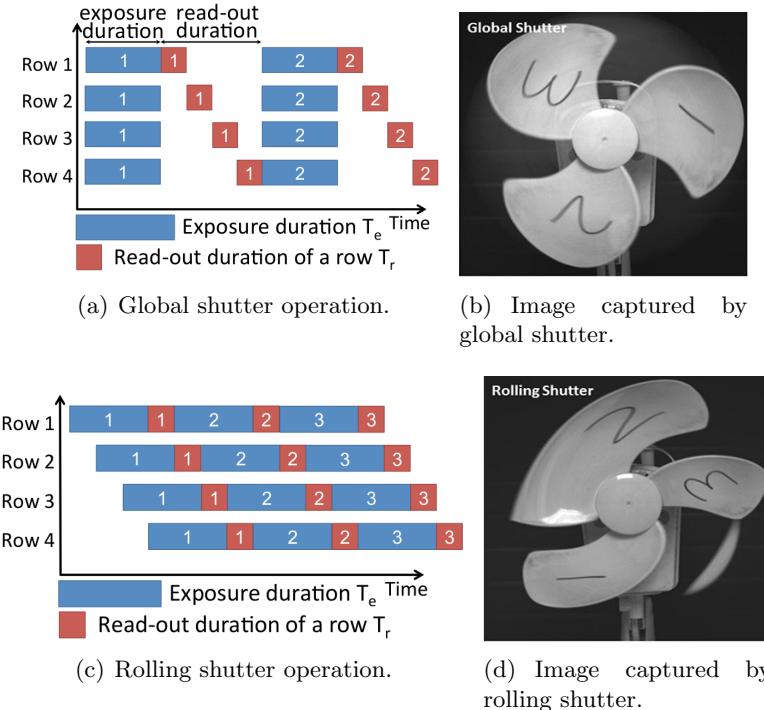


Figure 2.2: Comparison between rolling shutter and global shutter.
In rolling-shutter, each row is exposed right before the readout, and not all at once, as in a global shutter. This shift in the beginning of the exposure time, as shown in 2.2(c), results in a distorted image of the fan, as shown in 2.2(d) Figure (a) and (c) taken from [5]

Since the exposure of each row does not start together in rolling shutter mode, it causes distortions when an object moves a significant distance dur-

ing the time period of one frame. Figure 2.2(d) shows the distortion when using rolling shutter mode.

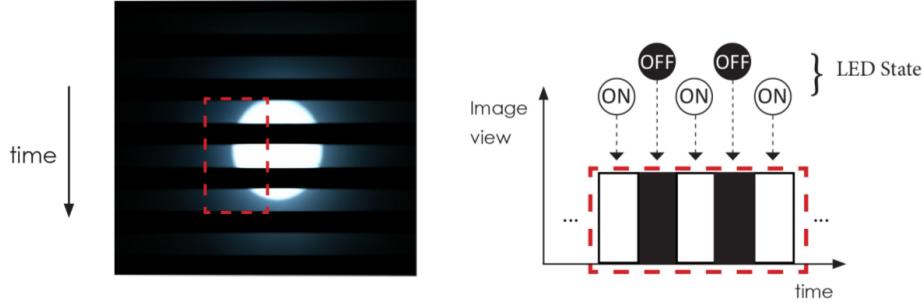


Figure 2.3: **Strip formation and LED state.** The width of each strip, corresponds to the duration of the LED state (on or off).

When an LED is modulated at a frequency higher than the rolling shutter speed, strips (distortions) of different light intensity are captured in the image as shown in Figure 2.3. A row appears white, when the LED was ‘on’ during the row’s exposure time (t_e). On the other hand, a row appears black, when the LED was ‘off’ during the exposure time. The intensity and width of the strip depends on the transmitter’s modulation frequency and the camera’s properties. However, if the transmitter’s modulation frequency is higher than the exposure time of a row, no distinct patterns can be observed, this will be explained in Section 5.4.

A key camera property in determining the rolling shutter speed is the exposure time. Section 5.2.1 will further explain the importance of the exposure time along with other important camera properties in light-to-camera systems.

Modern day smartphones’ camera are equipped with CMOS image sensors due to their low power consumption and low cost. Mid range image sensors can capture a video of 25 to 30 frames per second in full resolution mode. These cameras emphasise more towards picture quality and sharpness which leads to an unstable frame rate. Like in the case of the Motorola G4 , the frame rate fluctuates from 24 to 29 per second under full resolution mode.

2.3 Camera-based VLC

As described in Section 2.2, a camera can capture information from a modulating light source. This allows for a communication channel capable of

sending information continuously. Recent works have investigated this light-to-camera communication link focusing on a smartphone implementation.

The authors of [1] employed OOK and Manchester encoding schemes and achieved a data-rate of 125-375 bytes/sec at a close proximity of 9cm by using the frame-rate of 20fps. Rajagopal et al [8] utilise a BFSK scheme with iOS devices as a receiver, which resulted in the throughput of 1.25 bytes/sec up to a distance of 1m. RollingLight [5] presents an FSK scheme and using iPhone 4 as the receiver, they reported a throughput of 11.32 bytes/sec up to a distance of 1.6m using an light source of 60cm x 60cm. DynaLight [11] uses a smaller light source and achieved a distance of 1.2m with a low-cost Android device. The maximum reported throughput was around 10 bytes/sec with OOK and Manchester encoding. A recent work presented by Schmid et al [9] achieved the throughput of more than 150 bytes/sec with a PPM encoding scheme. They implemented their receiver on iPhone 6s with a frame-rate of 240fps and reported a maximum distance of 2.75m.

All of these works uses different encoding schemes and utilises different smartphones which results in different throughput generated by the system. Comparing these works only on the basis of throughput will not be fair, as two components are involved here, i.e. the data encoding method and the smartphone camera. It can be observed in Figure 2.4, that works [5], [11] and [8] implemented different methods and had medium camera capabilities and managed to get similar throughput values.

Schmid et al [9] uses a smartphone with a high-end camera and achieved a high throughput. Whereas, [1] uses a low-end smartphone camera and achieved the highest throughput (among the discussed works). *This raises the question of which parameter plays a vital role in determining the throughput of the system.* In this thesis, we focused on developing a common ground for the comparison of above-discussed methods by analysing as many variables as possible.

The authors in RollingLight [5] extensively studied the rolling shutter effect and found an unpredictable and varying idle time gap (*jitter*) between two consecutive frames. This idle gap leads to loss of the data packets in the captured image, which is depicted in Figure 2.5. This packet loss was observed in the case when frame duration was slightly greater than the packet duration. This means that one frame can capture at most one packet, but the varying idle time gap can result in random packet loss and frames without a complete packet. In order to overcome this situation, DynaLight [11] proposed a way to ensure the reception of at least one packet per frame.

Moreover, except for [11], all the works were designed to work with a fixed

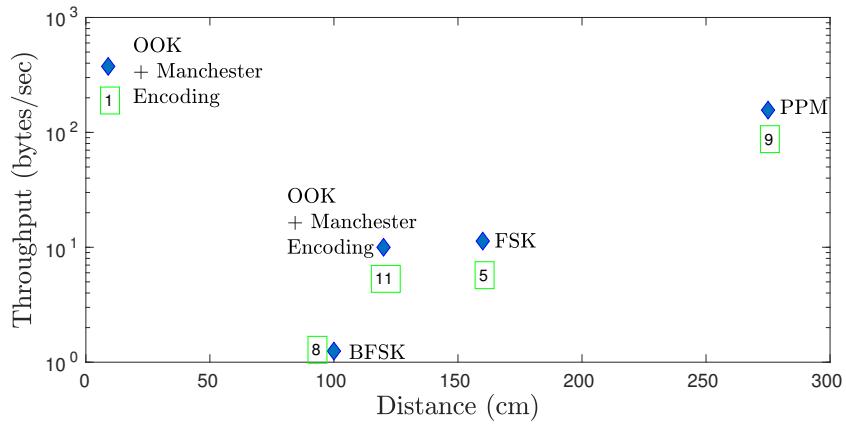


Figure 2.4: **State-of-the-art comparison.** Throughput vs maximum distance of the previous works which uses light-to-camera link for communication.

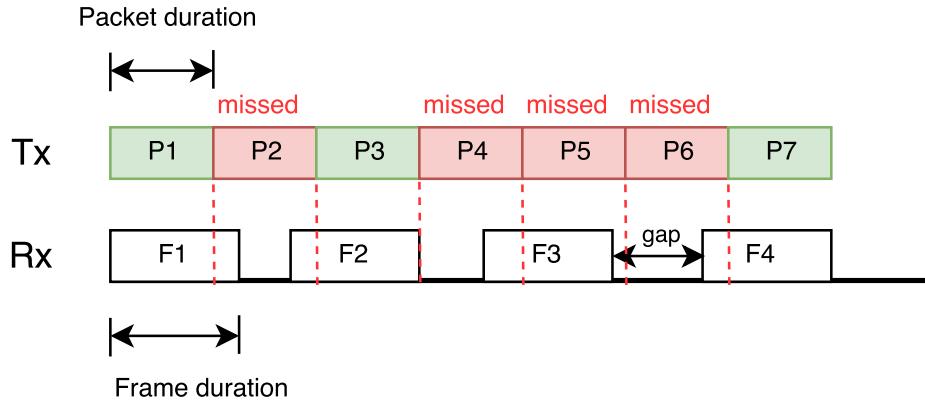


Figure 2.5: Missed packets when the frame duration is slightly greater than the Packet duration.

packet size and hence does not utilising the full channel capacity. This results in a suboptimal throughput at different distances between the transmitter and the receiver. In order to overcome this issue, we designed a distance adaptive light-to-camera communication link.

2.3.1 Summary

Table 2.1 give the summarised overview of the camera-based communication systems which were discussed above. It can be observed from the table that all the systems have different camera control which differs in functionality.

It must be mentioned that every system used a different frame rate and the size of light source used was also different, which affected the stated throughput.

Project	Frame rate (fps)/ Camera Control	Encoding Scheme	Throughput (bytes/sec)	Maximum Distance (cm)
CMOS for VLC [1]	≈ 20 / low	OOK + Manchester	375	9
Visible Light Landmarks [8]	30 / low	BFSK	1.25	100
RollingLight [5]	≈ 30 / medium	FSK	11.32	160
DynaLight [11]	≈ 30 / low	OOK + Manchester	≈ 10	120
Smartphones as Continuous Receiver [9]	240 / high	PPM	156.25	275

Table 2.1: An overview of State-of-the-art discussed in 2.3.

Chapter 3

Overview

This chapter gives a short overlook of the system. Section 3.1 presents the main components of the system, i.e. transmitter, receiver and mathematical model. Section 3.2 briefly discusses the basic functionalities of the system.

3.1 System Overview

This VLC system is designed for Line-of-Sight (LOS) data communication. Hence, it consists of a receiver (smart-phone) directly facing a transmitter (LED lamp).

The transmitter consists of an LED lamp, light driving circuitry and a microcontroller. The microcontroller controls the encoding of information and the modulation of light. Modulated signals are adjusted in such a way that there is no noticeable flickering.

On the other side, a smartphone is used as the receiver. As discussed in Chapter 2, we can either use a camera by taking advantage of the rolling shutter capturing mechanism or utilise the in-built light sensor. In order to be able to use the camera for detecting the light signal, we need to tune rolling shutter parameters so that “ON” and “OFF” signals can be differentiated, which is explained in Section 5.3.1. By capturing video and processing each frame separately, we are able to, identify the light source in the frame and decode the received symbols.

This system has to be adaptable with respect to distance to maintain reliable communication link and optimise throughput. To achieve this, we develop a model by which a receiver can calculate its distance from transmitter and provide an optimal packet size.

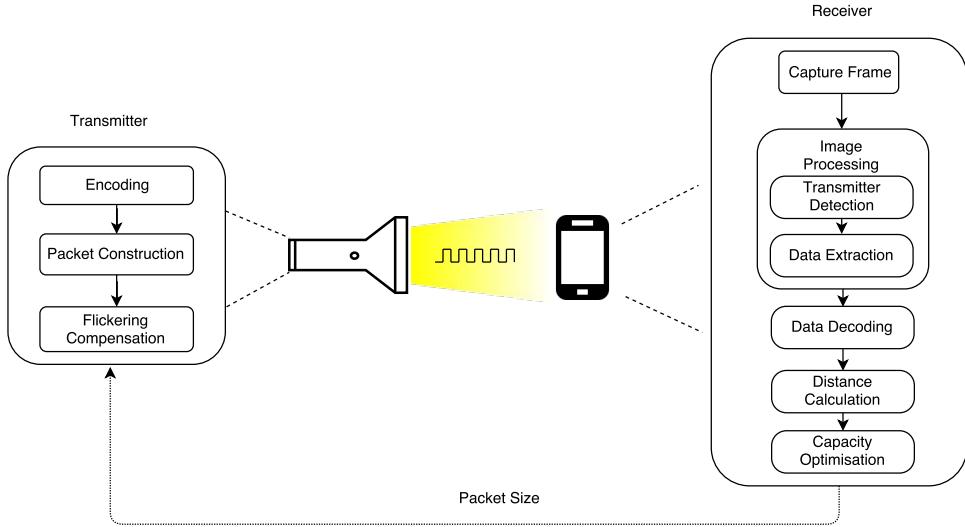


Figure 3.1: **System Overview.** The system consists of a transmitters and a receivers that communicate via modulating light.

3.2 Basic Functionalities

This section describes the main operation of each system component. The transmitter controls the modulation of light by performing data encoding, packet construction and flickering compensation. The receiver’s basic functionalities includes image processing, data decoding, distance estimation and capacity optimisation. The calculated distance also facilitate in determination of the symbol capacity.

Encoding Scheme: Encoding schemes are the methods for mapping data bits (digital “0” and “1”) to symbols (light OFF and light ON). In order to create an efficient and flicker free system, four encoding schemes were implemented and compared based on different parameters which will be further explained in Section 4.2.

Image Processing: Image processing refers to all operations performed on the captured image to extract information. It covers the methods for detecting the position of decodable information in the captured frame (transmitter detection) and extracting data patterns from it. These methods are based on the one discussed in DynaLight [11], the difference being the way transmitter is detected. Further description of the adopted methods with challenges are provided in Section 5.2.2.

Data Decoding Methods: Data decoding methods are the algorithms we use to reconstruct the transmitted information based on patterns extracted during the image processing. Different algorithms are used for different encoding methods which will be further explained in Section 5.2.3.

Distance Calculation: The receiver can calculate its distance from the transmitter. The distance calculation is based on the detected transmitter size and camera's hardware specification, with these parameters, a receiver can calculate its distance with an error upto 5%. More details will be presented in Section 6.1.

Capacity/Packet Optimisation: Packet Optimisation refers to the adjusting of packet size based on calculated distance. This helps in achieving longer range and overcome the issue of synchronisation as depicted in Figure 2.5. This is further discussed in Section 6.2.

Chapter 4

Transmitter

The main functionality of the transmitter (Tx) in a VLC system is to encode the data and control the modulation of light source, while maintaining a constant light intensity and avoid any flickering effect. This chapter presents an analysis of different data encoding schemes based on overhead and compensation for flickering along with the transmitter's hardware implementation.

4.1 Flickering Effect

A primary requirement in any VLC system is to maintain a constant average light intensity and eliminate any observable flickering in the transmitted light signal. Flickering in a modulating light source can be observed in two scenarios. First, when the frequency of modulating light is within human eye's perception range (direct flickering), which is approximately 100 Hz. Second, when the average light intensity is not constant over time (indirect flickering). In order to overcome direct flickering, modulation frequencies above 1 KHz were used.

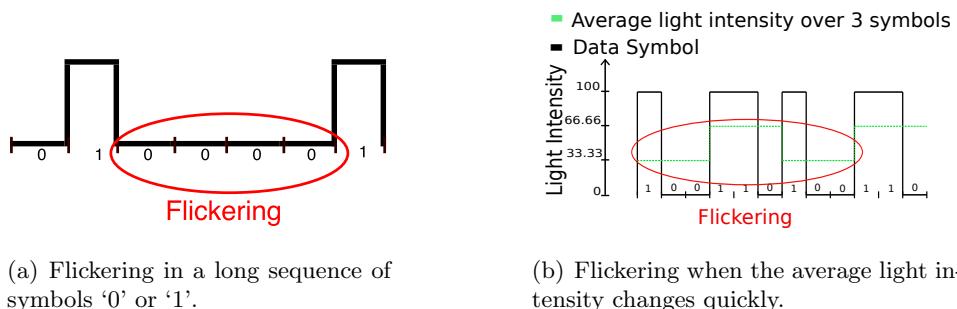


Figure 4.1: Cause for indirect flickering in the modulating signal.

Indirect flickering is observed when the transmitted signal has long sequences of ‘0’ or ‘1’ symbols (*atomic unit for data transmission*) or when the average light intensity changes over the interval of $\approx 10ms$ (the period where humans can sense change in light intensity), as it can be seen in Figure 4.1(a). Figure 4.1(b) shows a modulating light signal (black) where the single symbol duration is $3ms$ and the average light intensity (green) over three symbols i.e. $9ms$. As the average is changing every $9ms$, it can be perceived by humans and this change results in flickering.

In order to remove any indirect flickering, the average light intensity should remain constant over time. If this flickering is observed in VLC, *additional data symbols need to be added in between the signal* (compensation symbols) to create a constant value average. A detailed discussion regarding the addition of compensation symbols is presented in Section 8.1.

4.2 Encoding schemes

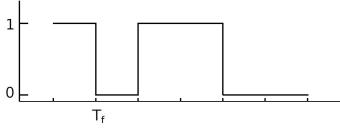
In order to transmit data, the system modulates the LED in such a way that symbols can be decoded accurately by the receiver and at the same time not to generate any noticeable flickering.

As discussed in Chapter 2, all the previous studies use different encoding schemes such as *OOK+Manchester*, *PWM*, *FSK* etc. This section presents a comparison and analysis of these individual encoding schemes considering the overhead required to overcome flickering effects.

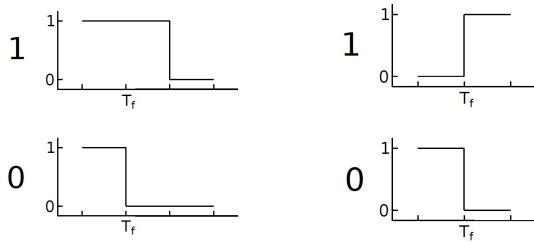
On-Off Keying (OOK) is the most basic approach, where digital data “1” and “0” are represented by turning the LED ON and OFF respectively. In *Pulse Width Modulation* (PWM) encoding, data is represented in form of variable ON period of the light source. *Manchester encodes* data using two symbols, i.e. symbol sequence ‘01’ for data bits ‘1’ and ‘10’ for data bit ‘0’. It provides an added benefit of always yielding 50% light intensity. These schemes are illustrated in Figure 4.2.

These encoding schemes are based on fixed modulation frequencies of the transmitter. Whereas *Frequency-shift Keying* (FSK) rely on modulating a single or several source(s) at multiple frequencies. These frequencies can either carry encoded binary data or act as code words.

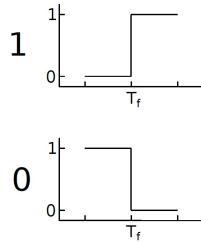
In order to compare these schemes, they are analysed using the same packet structure and the same amount of data payload.



(a) OOK encoding for the bitstream 101100.



(b) Pulse width modulation encoding.



(c) Manchester encoding.

Figure 4.2: Encoding schemes used in previous work.

Packet Structure

The packet consists of a preamble, payload, parity check and compensation for flickering. As discussed in Section 4.1, compensation symbols should be added in between the data to remove flickering. The packet header is a unique sequence of symbols to make sure it is distinguishable from other symbol sequences. The parity (XOR) check is added to detect errors in the received signal, but not for error correction.

$$\text{PacketSize} = M + N * n_{symbols} + P + C \quad (4.1)$$

The total size of a packet(in symbols) carrying “N” bits of data is given by the Equation 4.1, where ‘M’ is the fixed preamble width in symbols, ‘ $n_{symbols}$ ’ is the *symbol to bit ratio* of a encoding scheme, ‘P’ is parity and ‘C’ is the number of compensation symbols required.

On-Off Keying (OOK): Indirect flickering was observed in OOK, due to the difference in average ‘ON’ period of the packet with encoded data. Flickering was also observed while sending the same packet repeatedly because of the presence of long sequences of ‘0’ or ‘1’ symbols. In order to overcome the flickering, one compensation symbol was inserted after every two symbols (i.e $N/2$ compensation symbols). Even after adding compensation

symbols, flickering was still present. The cause will be further explained in Section 8.1. Due to the flickering, OOK was not implemented for decoding.

OOK can have repetitive sequences of the same symbol, the preamble should be bigger than data and parity combined i.e. it should be at least $N+1$ symbols long. With this set of values, the packet size for ‘N’ data bits for *OOK* will be $5/2 \cdot N + 3$.

Pulse Width Modulation (PWM): The PWM used in this thesis consists of 3 symbols. Data bits ‘1’ and ‘0’ are represented by symbol sequences ‘110’ and ‘100’ respectively. Due to the changing duty cycle for a data bit, indirect flickering was observed, as depicted in Figure 4.1(b). In order to remove the flickering, one compensation symbol was added after every 3 symbols (i.e. $N+1$ compensation symbols). It will be further explained in Section 8.1.

Considering the bit sequence after compensation, symbol ‘0’ can’t be repeated thrice in any condition (‘1’ can be repeated thrice, once from a compensation symbol and twice from the first two symbols of a data ‘1’). As a result, a preamble sequence ‘10001’ was chosen. The additional ‘1’ on the boundaries help in decoding the signal and prevents the formation of longer symbol sequence of ‘0’s, which would introduce flickering. With this set of values, the packet size for ‘N’ data bits for *PWM* will be $4 \cdot N + 9$.

Manchester Encoding: Due to the data encoding pattern, it is not possible to have more than two consecutive matching symbols. As a result no flickering was observed in this scheme. The preamble is set as symbol sequence of ‘10001’. Similar to PWM, ‘1’s on the boundaries help in decoding the signal correctly. With these values, the packet size for ‘N’ data bits for *Manchester* will be $2 \cdot N + 6$.

Frequency Shift Keying (FSK): This encoding scheme is the method proposed by RollingLight [5]. The authors implemented frequency modulation as code words i.e. each modulating frequency is considered as one symbol. A high-frequency splitter was inserted between transmission of each data frequency to minimise effect of jitter in rolling-shutter capturing mechanism.

$$\lfloor \log_2 |F| \rfloor \quad (4.2)$$

If F denotes the total number of frequencies used, then the number of data bits represented by each symbol can be given by Equation 4.2. A total of 48 modulation frequencies from 320 Hz to 5.6 KHz were used, and hence each symbol represents $\lfloor \log_2(48) \rfloor = 5$ data bits.

Comparison of Encoding Schemes

The motivation to implement these encoding schemes was to analyse their pros and cons and to find out which of these schemes either individually or combined will be suited for a light-to-phone link. OOK's major advantage was its low encoding complexity (smallest symbol to bit ratio), but the requirement of ' $N/2$ ' compensation symbols to reduce flickering, increases the packet overhead significantly. PWM encoding inherently had a low data density, which was further reduced by adding one compensation symbol after every 3 symbols (i.e. $N+1$ extra symbols), which in turn resulted in an increased packet overhead. To transmit the same amount of data, PWM uses packets that are almost twice the size of Manchester's.

Table 4.1 summarises the comparison of OOK, PWM and Manchester encoding schemes on basis of compensation for flickering, packet size and overhead.

Encoding scheme	Symbol to bit ratio	Preamble size	Compensation for flickering	Packet size	Overhead (for $N = 8$)
OOK	1	$N + 2$	$N/2^*$	$\frac{5}{2} \cdot N + 3$	76%
PWM	3	5	$N + 1$	$4 \cdot N + 9$	81%
Manchester	2	5	-	$2 \cdot N + 6$	65%

Table 4.1: Comparison of the overhead of different encoding schemes(* denotes that flickering was not completely removed).

It can be concluded that the Manchester encoding is better than OOK and PWM encoding methods in terms of overhead (for the same data size) and flickering.

Comparing Manchester and FSK encoding will not be feasible on the basis of overhead as their data transfer methods are different. An in-depth comparison of these two methods is discussed in Chapter 7, with the help of the *Mathematical model* developed in Chapter 6.

4.3 Hardware Implementation

The transmitter in this system consists of a microcontroller (arduino *AT-MEGA328p*), LED driving circuit and a round LED light with a diameter of 23cm and a power of 15W.

The encoding of data and LED modulation is performed by the microcontroller. In order to have precise control over the modulation frequency, we use the atmega328p's timer interrupts with a prescale factor of 256. The

data to be transmitted was pre-coded on the microcontroller during experiments, but it can also be dynamically sent to the microcontroller via a serial connection from the PC.

This LED panel consists of multiple LEDs with a diffuser screen to provide even light output. The maximum driving current for the panel is 1.5A which results in 2000 lm at 15W. In order to drive the LED panel, a constant current driver circuit is used with an output of 1.5A, which is shown in Figure 4.3.

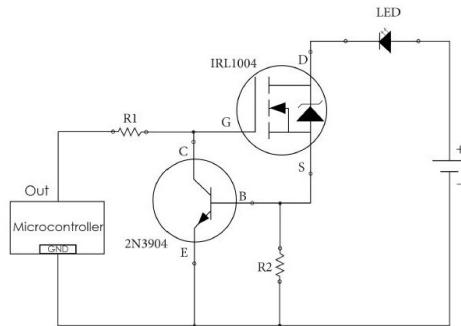
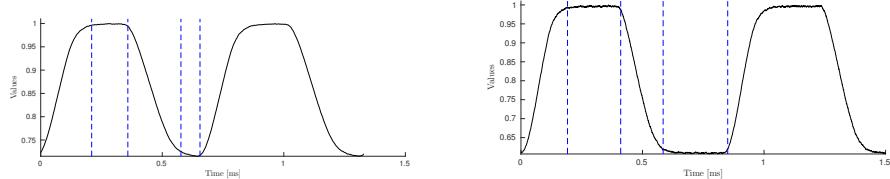


Figure 4.3: LED driving circuit.

4.4 Duty-Cycle Adjustment

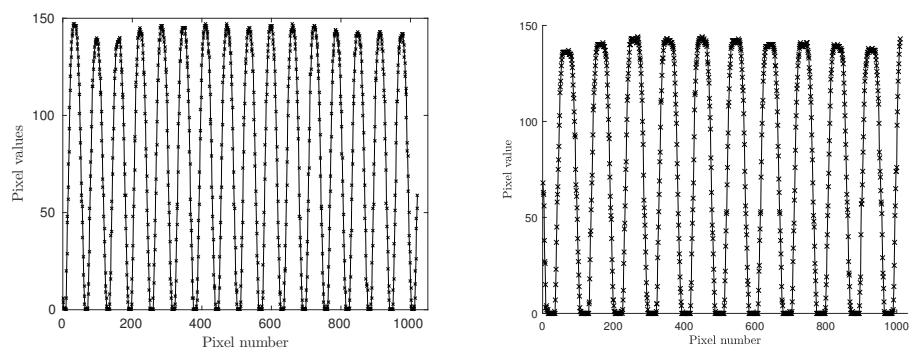
When modulating the above mentioned LED with a 50% duty cycle at high frequency (more than 1KHz), it was noticed that the average ON and OFF times of the transmitted light signal are not equal. It can be seen in Figure 4.4(a) that very few samples were acquired when the light was at the OFF state. This was because, the charging (0.217ms) and discharging (0.22ms) of the LED took around 66% of the whole period and the actual OFF time (0.079ms) was reduced. This causes signals received by a smartphone to be very irregular, as shown in Figure 4.5(a).

In order to correctly decode the signal, the average ON and OFF times of light should be approximately equal. The OFF time of the modulating signal is increased by changing the duty cycle to 40%. After this adjustment the signal received by the photodiode shows that the actual ON and OFF time are almost equal (0.141ms and 0.152ms respectively) as shown with blue lines in Figure 4.4. Width of the ON and OFF pulses in the received signal after adjusting the duty cycle, the width are approximately equal as seen in Figure 4.5(b).



(a) Transmitted signal with 50% duty cycle at 1.5KHz. (b) Transmitted signal with 40% duty cycle at 1.5KHz.

Figure 4.4: Transmitted signal with different duty cycle as captured by a photodiode.



(a) Transmitted signal with 50% duty cycle. (b) Transmitted signal with 40% duty cycle.

Figure 4.5: Transmitted signal with different duty cycle as received by a smartphone.

Chapter 5

Receiver

This chapter discusses the possible receivers to establish a Light-to-Phone link. This link can be established by using either of these three receivers: inbuilt camera, on-board light sensor or external add-on devices [6]. Add-ons are not analysed as the main focus of this thesis is to use off-the-shelf smartphone without any modification.

5.1 Light Sensor as Receiver

Nowadays, almost all major smartphones include a light sensor which is used for adjusting screen brightness and contrast according to ambient light intensity. The light sensors consist of an array of photodiodes whose sampling rate is controlled by the operating system and the user has no control over them.

An application was developed using the Android sensor API to acquire values from the light sensor as fast as possible and store them in a file. Figure 5.1 shows the number of samples recorded by the phone's light sensor in a one-second interval for different transmitter frequencies. It can be seen that below 60Hz, the number of samples acquired were not enough to reconstruct the transmitted signal. This means that in order to communicate information to a phone's light sensor, the LED would need to be modulated with frequencies above 60Hz, which would cause flickering effects as explained in Section 4.1. Unfortunately, smartphone ambient light sensors are optimised for dynamic range and typically do not have the required frequency response to decode high-speed data.

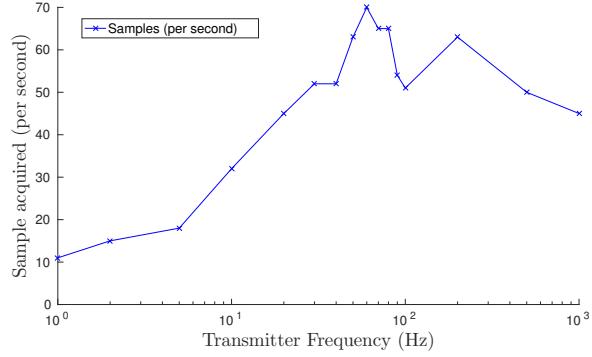


Figure 5.1: Samples acquired (per second) vs Modulation frequency.

5.2 Camera as Receiver

This section describes the development of a camera system as the receiver (Rx). It discusses the tuning of camera parameters to receive the transmitted signal, the description of the image processing pipeline and the data decoding methods.

5.2.1 Camera Parameters

As described in Section 2.2, smartphone cameras employ a rolling shutter mechanism to capture images. In order to get the dark and light bands in the captured frame, some of the camera parameters have to be tuned appropriately.

Exposure Time: The exposure time is one of the most important parameter when capturing the image of a modulated light. It is the amount of time taken by each pixel to collect light. In practice, shorter exposure times increases the ability to distinguish the boundaries between light and dark bands in an image. From Figure 5.2, it can be observed that as the exposure time increases, adjacent bands blend into each other.

As depicted in Figure 2.2, the exposure of each subsequent row is off-set with the read-out time (T_r). With the help of this read-out time, the width of bands (in pixels, W_s) can be estimated using Equation 5.1 (given by RollingLight [5]). This equation presents a relation between the modulation frequency of the light source (f_l) and the read-out duration of the camera.

$$W_s = \frac{1}{2 \cdot f_l \cdot T_r} \quad (5.1)$$

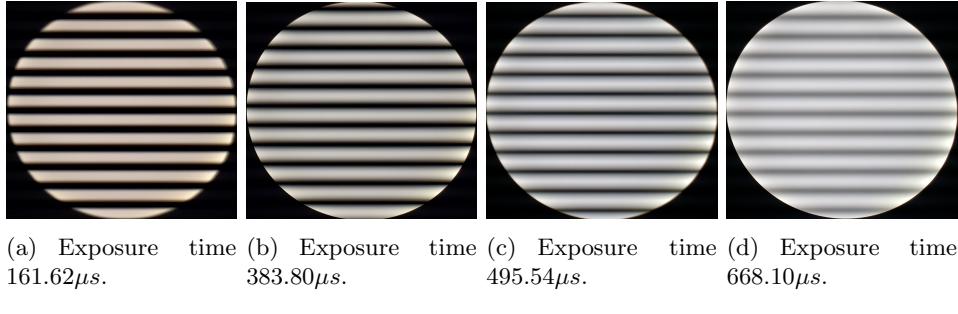


Figure 5.2: **Effect of exposure rate.** Image showing captured frames at 1KHz. As the exposure time increases, bands mix into each other.

Frame rate: The frame rate of a camera is defined as the number of frames captured in a second. This parameter is significantly important when it comes to specifying the data-rate of a VLC link. It depends on the time needed to capture a single row and the number of rows in one frame (camera resolution).

Each “pixel-row” is first exposed and then read. This is known as *scan-rate* of a camera. According to the rolling shutter effect, the time to scan a single frame is given from Equation 5.2, where T_e is the exposure time, T_r is the read-out time and N is number of rows.

$$T_{frame} = T_e + N \cdot T_r \quad (5.2)$$

Combining Equations 5.1 and 5.2, gives Equation 5.3, which relates the width of each band with the most important camera parameters, i.e. the exposure time, the frame rate and the number of rows (resolution).

$$W_s = \frac{1}{2 \cdot f_l \left(\frac{T_{frame} - T_e}{N} \right)} \quad (5.3)$$

Equation 5.3 infers that the *width of bands does not depends on the camera orientation, the size of the light or the distance between the light and the smartphone.*

Other Camera Functions: Some of other camera parameters which affect the captured frames are ‘Auto-focus’, ‘White-balance’ and ‘Antibanding’. These parameters don’t play a major role in our system, but they should have a fixed value in order to keep their effects minimal. Therefore, in this system auto-focus, white-balancing and antibanding were disabled.

Limitations

In practice, the Android OS limits the users' ability to adjust or modify the above mentioned parameters. The scan rate and the read-out time are hardware parameters and thus, can't be changed. Control over the exposure time and frame rate is very limited and varies from phone to phone. Moreover, the exposure time is also limited by the image sensors due to their hardware capabilities. More information regarding the selected values for each parameter is discussed in Section 5.3.1.

5.2.2 Image Processing

After adjusting the camera parameters, a frame with distinguishable white and dark bands was obtained. This section explains the processing required on the captured image in order to extract out useful data bits from bands. Methods provided by the *OpenCV image processing library* were utilised in our system's image processing pipeline.

Transmitter Detection

Detection of the transmitter is an important step while extracting information from the captured frame. As this system can operate at multiple distances, the transmitter will not be spread out in the whole frame as it can be observed in Figure 6.1 and there is no guarantee that the transmitter will be in the centre of a frame. So, this transmitter detection mechanism will identify, which part of the image contains decodable information.

The transmitter detection method employed in this thesis is based on the one that is presented in DynaLight [11].

Existing Method: The sub-steps used to locate the transmitter are: 1) Blurring the captured frame using a 100×100 kernel to create a "shadowy" image. 2) Applying an OTSU filter to create a binary image. 3) Find contours to detect the area where the transmitter's light is diffused. 4) Find a minimum enclosing circle for each contour to cover the area and get the size of the image.

Figure 5.3 shows the above-mentioned steps being applied to a captured frame. While using this detection method, it was found that it considers each detected contour as a separate transmitter, which results in multiple detections of a single transmitter, as shown in Figure 5.4(b). This was caused because OpenCV's "findContours" method detects parts of the image which are joined together. Figure 5.4(c), shows that three contours were detected

in the image because of the particularly ‘wide’ gaps between some bands. These gaps led to irregularities in data extracted from the frame. As a result, we had to investigate a method to merge all these different contours into a single one.

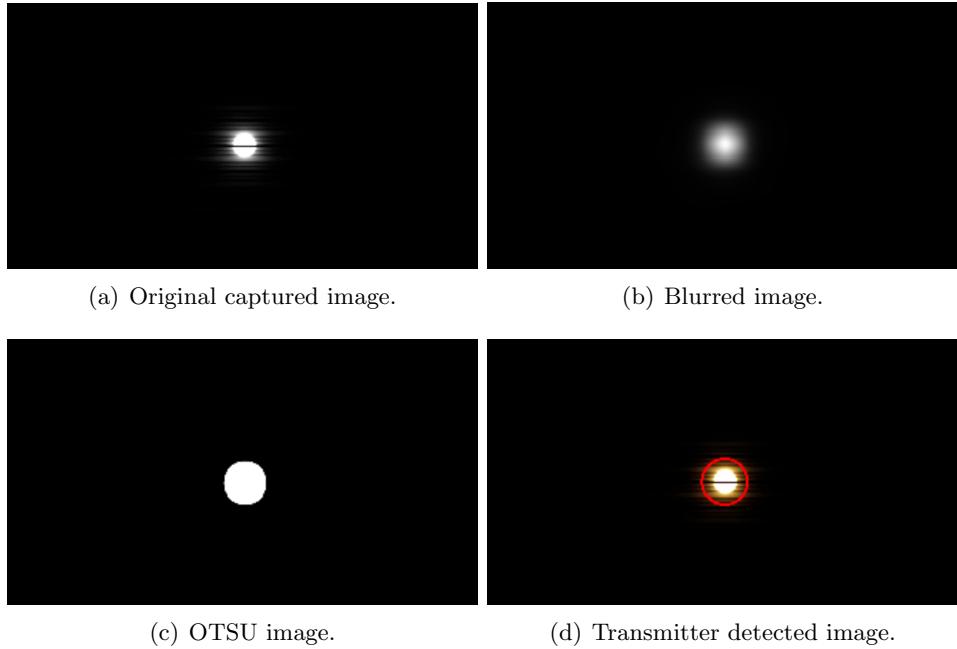


Figure 5.3: **Transmitter detection pipeline of DynaLight.** First the image is blurred (b), then passed through the OTSU filter (c) and finally, minimum enclosing circle (red circle) is calculated (d).

Contour normalisation: In order to get a single contour for a transmitter, OpenCV’s “moments” function was used. This function helps in calculating the area and centre of the contour. Normalising and joining the centres of detected contours based on weighted average, a single circle can be plotted to cover the transmitter in the image, as shown in Figure 5.4(d). The diameter of this final circle is called the “*image size in pixels*”, which will be used in later analysis.

Data Extraction

The encoded data in the detected sub-region of the image must be converted into an acceptable single dimension waveform (samples) in order to be decoded. DynaLight [11] and [1], suggest different methods to extract data from the captured frame. The authors in [1], employed a 3rd degree

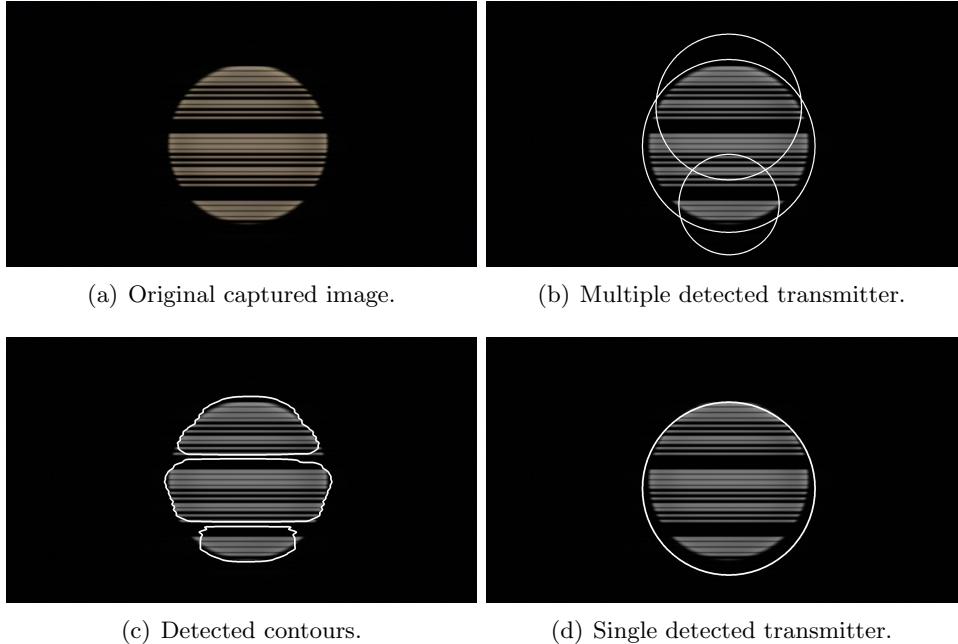


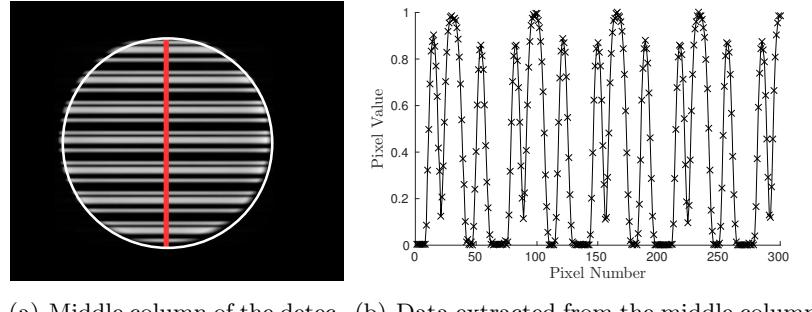
Figure 5.4: Detection of Multiple transmitters. Multiple detection of transmitters from an old algorithm (b) and single detected transmitter after contour normalisation(d).

polynomial regression to normalise and extract the raw values from the non-homogeneous bands due to the low granularity (resolution) of the camera. Whereas DynaLight suggests extracting data from an offset from the centre of detected region to avoid over-exposed region. By utilising different hardware and camera parameters, we were able to overcome the limitations these works encountered.

In this thesis, pixel intensities were extracted from the vertical diameter of the detected transmitter, as shown in Figure 5.5(a). As there was no over-exposure effects and the bands were homogeneous, resultant single dimension waveform had similar reference level for similar values. Afterwards, the whole signal is normalised by bringing it down to the same reference level. More specifically, by dividing the data stream with the maximum pixel value, as shown in Figure 5.5(b).

5.2.3 Data Decoding

After getting the waveform from a frame, the next step is to decode these samples to symbols ‘1’ or ‘0’. A threshold is applied to the normalised data sequence in order to distinguish the two symbols of the transmitted signal.



(a) Middle column of the detected transmitter. (b) Data extracted from the middle column.

Figure 5.5: The red vertical line represents the middle column (a) and the extracted single dimension waveform (b).

Since, normalisation brings the mean to 0.5, every value that is above the mean is considered as ‘1’ and every value below the mean is considered as ‘0’ . After the threshold, values are down-sampled in order to have a single symbol for every band and thus a continuous symbol sequence.

In order to down-sample the threshold signal, the camera’s read-out time (T_r) should be known. With the help of it, the expected width of the symbol can be calculated using Equation 5.1. Usually the read-out time of a device is an unknown parameter. A square wave of 1KHz was used as a training signal to estimate the read-out duration. This calibration step needs to be performed for new devices once, after that the calculated read-out time can be used directly. After knowing the read-out duration, it can be used to estimate either transmitter’s frequency depending on width of strip or vice-versa.

For PWM and Manchester encoding a constant modulation frequency of 5KHz was used (the choice of this frequency will be discussed in Section 5.4). Since the preamble for PWM and Manchester encoding is ‘10001’, a sliding window of 57 zeros i.e $3 \cdot W_s$ (calculated strip width at 5 KHz) was applied to detect the preamble. The detection becomes more reliable by using the ones in the preamble, as they prevent merging of preamble’s zeros with data symbols. After detecting the preamble, the data sequence proceeds further to the next stage.

Pulse Width Modulation: In the case of PWM encoding, symbol ‘1’ can repeat a maximum of three times (after adjusting for compensation), so another sliding window of 19 ones (W_s) was used to make hard decision between symbol sequence of 1, 11 and 111. Symbol ‘0’ can repeat a maximum of two times in the data sequence(even after adjusting for compens-

ation). In order to make hard decisions between symbol sequence of *0 and 00*, sliding window of 18 *zeros* (W_s) was used. This was due to the fact that, the widths of symbols ‘1’ and ‘0’ are slightly different (as described in Section 4.4). Any other case is considered as an erroneous reception and the received data is discarded.

After down-sampling the data stream, every fourth symbol is removed as it was inserted to overcome flickering. This sequence is then converted to the original bit-sequence by following the same method for encoding in reversed order (i.e. sequence of ‘100’ maps to data bit ‘0’ and sequence of ‘110’ maps to data bit ‘1’). Parity is calculated on the decoded bits and compared with the received value. If the parity check fails, then the received data is classified as erroneous reception.

Manchester Encoding: In Manchester encoding, a similar down-sampling method as that of PWM was employed to extract a symbol sequence. As none of the symbols can appear more than twice consecutively, a single window of 10 *ones* (W_s) was used to differentiate between single and double symbols (i.e. ‘1/0’ or ‘11/00’).

After down-sampling the data stream, Manchester encoding was performed in reverse (i.e. sequence of ‘10’ maps to data bit ‘0’ and sequence of ‘01’ maps to data bit ‘1’). Parity is calculated on the decoded data bits and compared with the received value. If parity check fails, then the received data is classified as erroneous reception.

Frequency Shift Keying: The decoding of FSK encoding scheme does not require down-sampling, as it is based on estimation of frequency from the acquired strip width. RollingLight [5] suggests a *YIN-based* method for strip width estimation, which estimates the frequency depending on the difference in light intensity (pixel values) of at least 6 consecutive bands. They limit their system to detect only 1 frequency from the captured frame.

A similar method was used in our system, but we can detect more than 1 frequency in the signal waveform. This method first calculates a running average over 6 bands, if the values of all 6 stripes are in range of ± 1 (because, the system uses frequencies at strip spacing of 2, which is explained in Section 5.4) then the frequency is estimated using Equation 5.1. As this method does not average over the whole signal at once, it can detect multiple frequencies in the signal (given that a frequency has at least 6 bands).

5.3 Implementation Details

The receiver in this thesis were the back facing cameras of the Motorola G4, OnePlus 2 and Samsung Galaxy S5 smartphones. The Motorola G4 and OnePlus 2 have a 13MP camera whereas the Galaxy S5 have a 16MP back camera. All of these smartphones operate on *Android 6.0.1 (Marshmallow)* operating system.

An Android application was developed to operate based on the parameter values discussed in Section 5.3.1. This application requires Android 6.0 or higher to run, as older APIs for camera control have been deprecated by Android. It should be noted that, this application takes around 6 to 7 seconds to capture a frame and decode information from it. So the developed application was used to serve as a common ground to capture 1080p videos with maximum of 30fps and then Matlab and C++ was used for the post processing of data.

In order to overcome the limitations of Java with respect to memory management and limited image processing capabilities, the image processing pipeline was developed in the Android native interface, which is C/C++. Another benefit of implementing the processing pipeline in the native interface, is to take advantage of the OpenCV image processing methods in C++.

5.3.1 Parameter Values

As described in Section 5.2.1, the exposure time and scan rate are the most important parameters of the camera to calculate the expected width of each band. Table 5.1 summarises the calculated or measured parameters used in this thesis.

Parameters	Value
Scan Rate	$\approx 49,090$ rows/sec
Frame Rate	≈ 40 frames/sec
Exposure Time	$1/6,188s (161.6\mu s)$
Resolution	1920x1080

Table 5.1: Camera parameters uses in this work.

In order to determine the camera's scan rate, the receiver was set at a fixed distance from the transmitter such that the light occupies the complete image. After fixing the distance, a square wave of 1KHz frequency was transmitted, which formed 22 complete bands in the image. By multiplying the number of bands with the period of the transmitted signal (1ms), it was concluded that, the camera takes around 22ms to scan 1080rows. As

a result, the calculated scan rate was approximately 49090 rows per second (or approximately 40 frames per second).

The Android operating system is not very flexible when it comes to adjusting the exposure time of the camera. In new versions of camera APIs, Android does provide a method to disable auto adjustment features and to set the exposure time to a predefined value. All the previous works suggest to set the exposure time as low as possible. So, the exposure time was fixed at $1/6188\text{sec}$ (i.e. $161.6\mu\text{s}$). It was also observed that the exposure value can reach below the specified value with the help of auto-exposure but in that case, the camera can adjust it any time depending on lighting conditions.

The parameters like antibanding, white balance and autofocus were disabled, to remove their effect from the system.

5.4 Modulation Frequencies

In order to maximise the amount of information that can be sent from the transmitter, we need to identify the maximum frequency that the camera can distinguish. Assuming that each received band is one pixel wide, a theoretical upper bound of half of the camera's scan rate can be derived using Nyquist criterion. However, decoding one pixel wide band is impractical. In order to decode the image correctly, the band should be a few pixels wide. Experiments with different frequencies were performed in order to find the upper bound for the modulation frequency.

As can be observed in Figure 5.6, as the modulation frequency increases above 5KHz, the differentiation between different bands is not possible as the width of individual bands is too narrow. From Equation 5.1, it was calculated that for the frequency of 6KHz around 9 pixels per band were acquired, this small width makes the band almost invisible. Whereas, at 5KHz an average of 11 pixels per band were acquired, which makes the width sufficient enough for decoding.

During the processing of the signal, it was observed that, each band should be at least 18 pixels wide to be decoded correctly. In our system the maximum modulation frequency of 5 KHz was used, as it fulfils the requirements of distinguishable bands and not causing flickering.

In order to have higher accuracy in estimating the frequencies, they should have a spacing of at least 2 pixels. The minimum frequency was fixed at 1 KHz, which gives the widest strip width of 95 pixels and the maximum detectable frequency was fixed at 5 KHz, which gives the narrowest strip width of 19 pixels. With a spacing of 2 pixels, a set of 39 available frequencies were used for FSK modulation.

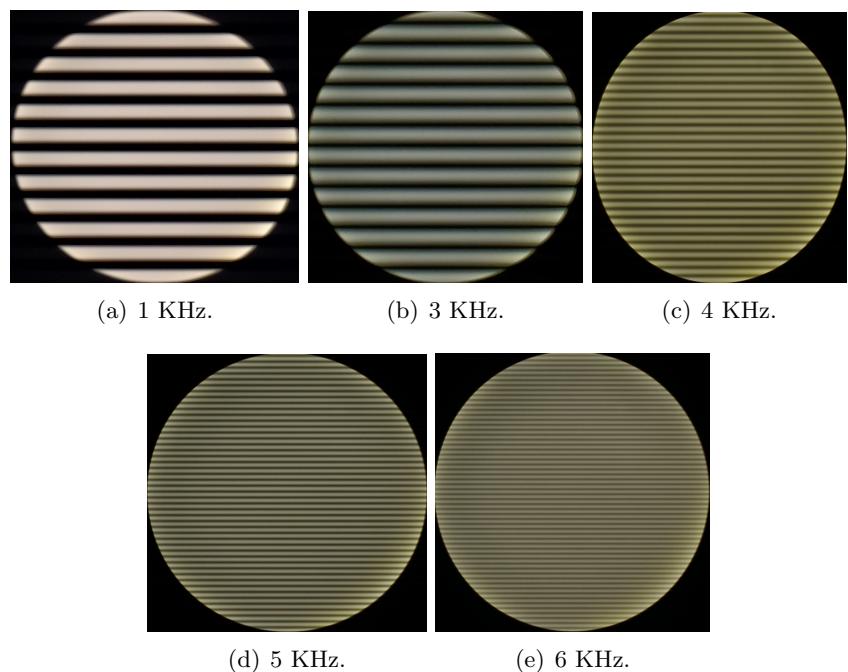


Figure 5.6: Frequency and pixels per band. With an increase in the frequency of the transmitted signal, the white and dark bands mix into each other and cannot be distinguished by software.

Chapter 6

Mathematical Model

This chapter discusses the proposed Mathematical models in order to achieve dynamic channel capacity. Section 6.1 presents a method to relate the number of symbols in a captured frame with the distance between a transmitter and a receiver. Based on the number of symbols in a frame, the packet size can be adjusted to maximise the range, which is discussed in Section 6.2.

6.1 Distance Model

As mentioned earlier, one of the goals of this project is to create a light-to-camera system, which can maximise its data capacity based on the distance between the receiver and the transmitter. First, the relation between the channel capacity and the distance has to be identified. The shorter the distance between the receiver and the transmitter, the bigger the light source will appear as it can be observed in Figure 6.1. This model proposes a method to calculate the distance between a transmitter and a receiver based on the camera's internal parameters. It also relates this distance to the number of symbols present in a captured frame based on the image size. This model will be used in calculating the packet size to optimise the channel.

6.1.1 Distance calculation

The distance between a transmitter and a receiver can be calculated using the detected transmitter's size, where the lens magnification ratio is a necessary parameter. As the distance increases, the projected transmitter area on the image sensor decreases, provided the lens magnification remains constant.

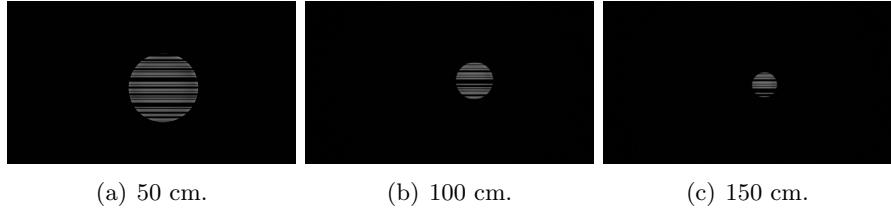


Figure 6.1: **Distance and image size.** The larger the distance, the smaller the image.

Few assumptions are made in this method: 1) a full image of the light is captured in a frame and 2) complex lens systems of cameras can be simplified as a single lens system with varying focal length.

Fundamental concepts of optics are involved in the distance calculation (i.e. focal length of lens and image magnification) in combination with physical parameters of the camera. Figure 6.2 shows the schematic diagram and variables used in the formula.

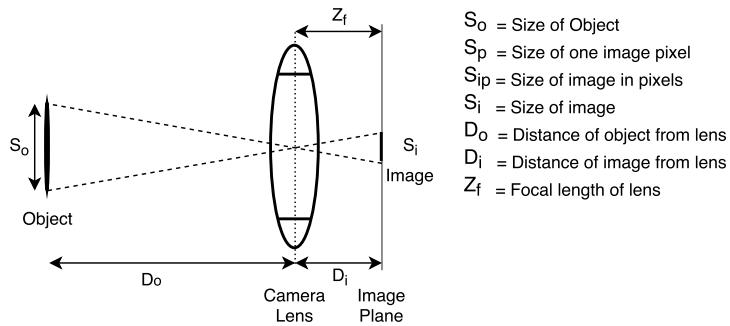


Figure 6.2: Image showing the parameters involved in the distance formula.

The size of image is a product of the individual pixel size and number the of pixels covered by the image (S_{ip} , which is calculated in Section 5.2.2).

$$S_i = S_{ip} \cdot S_p \quad (6.1)$$

Magnification (m) of an image is a ratio of the image size and the object size (S_o), given by Equation 6.2a. Using this, Equation 6.2b gives a relation between distance and the size of image.

$$m = \frac{S_i}{S_o} = \frac{D_i}{D_o} \quad (6.2a)$$

$$\frac{D_o}{D_i} = \frac{S_o}{S_{ip} \cdot S_p} \quad (6.2b)$$

The relation between the focal length of lens (Z_f) and the lens's distance from the object (D_o) and the image (D_i) is given by Equation 6.3.

$$\frac{1}{Z_f} = \frac{1}{D_i} + \frac{1}{D_o} \quad (6.3)$$

Now rearranging Equation 6.3 for object distance and substituting the image distance from Equation 6.2b, we get Equation 6.4.

$$D_o = \left(\frac{S_o}{S_{ip} \cdot S_p} + 1 \right) \cdot Z_f \quad (6.4)$$

Using Equation 6.4, the distance between the transmitter and the receiver can be calculated based on the size of the image formed in pixels (given that the pixel size and focal length of the image sensor are known). This formula is verified using four different smartphones and the calculated values match very closely to actual values. More information about the experiment can be found in Section 8.2.

6.1.2 Symbol Capacity Calculation

Figure 6.1 shows that, as the image of transmitter becomes smaller in size, the amount of information encoded in it decreases. By calculating the number of symbols in the captured image, it can be related to distance and the result can be used to adjust the packet size accordingly.

The formula presented by RollingLight (Equation 5.1), to approximate the width of one band can be extended in order to calculate a lower bound on the number of symbols present in detected transmitter.

$$N_s = \lfloor S_{ip} \cdot f_l \cdot T_r \rfloor \quad (6.5)$$

Equation 6.5 provides a relation between the size of the detected transmitter, the modulation frequency and the exposure time of the camera. The ‘Floor’ function helps in approximation, as it removes the partial symbols present in an image which are undecodable.

After rearranging Equation 6.4 for the size of the transmitter we get:

$$S_{ip} = \left(\frac{S_o \cdot Z_f}{S_p} \right) \cdot \left(\frac{1}{D_o - Z_f} \right) \quad (6.6)$$

Substituting Equation 6.6 in Equation 6.5 gives a relation between the number of symbols in the captured frame at a given distance.

$$N_s = \left\lfloor \left(\frac{S_o \cdot Z_f}{S_p} \right) \cdot \left(\frac{f_l \cdot T_r}{D_o - Z_f} \right) \right\rfloor \quad (6.7)$$

In Equation 6.7, an approximation can be done $D_o - Z_f \approx D_o$ because the value of focal length of camera (Z_f) usually varies from 2mm to 4mm. This gives the final relation as:

$$N_s = \left\lfloor C \cdot \left(\frac{f_l \cdot T_r}{D_o} \right) \right\rfloor \text{ where } C = \left(\frac{S_o \cdot Z_f}{S_p} \right) \quad (6.8)$$

Where ‘C’ is a constant for a given smartphone.

Equation 6.8 quantitatively defines how distance affects the received number of symbols in the light-to-camera communication link. This equation implies that, if the physical parameters are constant, the number of symbols increases with the modulation frequency and decreases when either the distance is increased or the exposure time is increased. The evaluation of this method is presented in Section 8.2.2

6.2 Packet Optimisation

As discussed in Chapter 2, a packet can be received in more than one frame or missed completely. This is due to the fluctuation in reception rate of the camera and the distance between transmitter and receiver. So, packet size is a very important characteristic of a light-to-camera communication system. Having a big packet may result in the reception of a single packet in more than one frame i.e. overestimating the channel capacity. Whereas, having a small packet may result in receiving multiple packets in a single frame i.e. underestimating the channel capacity. It is difficult to estimate which part of the packet will be lost due to the discontinuous capturing of the frames.

Our system provides a flexible method to adapt its channel capacity at different distances. As discussed in the previous section, the number of symbols in the detected transmitter area can be calculated depending on the distance. In order to overcome the no packet received situation, it has to be ensured that at least one whole packet fits within the detected transmitter region. This will ensure that, in every captured frame the receiver can get one complete packet and the communication link will remain stable.

DynaLight proposed a method to tackle this problem by adjusting the packet size such that, the detected transmitter area can always fit two complete packets. This will make sure that, in the case where only a small portion of the first packet is lost, the second packet will always be received completely. This can be achieved by having a variable packet size.

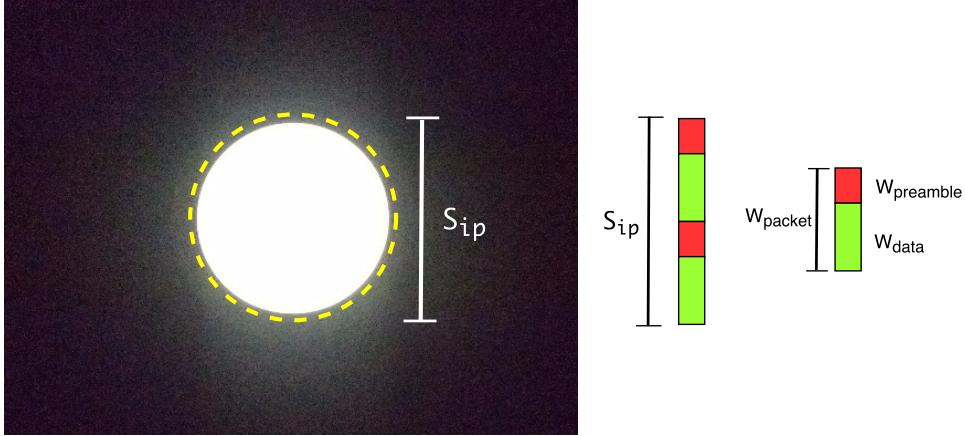


Figure 6.3: **Detected transmitter area and packet size.** Two complete packets should fit in detected area.

In order to be able to fit at least two whole packets in the detected transmitter area as shown in Figure 6.3, the size of the packet (in pixels W_p) should follow Equation 6.9a. According to Equation 6.9a, the number of symbols (N_s) should be at least twice than that of the size of packet (in symbols W_{packet}), as shown in Equation 6.9b.

$$S_{ip} \geq 2 \cdot W_p \quad (6.9a)$$

$$N_s \geq 2 \cdot W_{packet} \quad (6.9b)$$

Following from the analysis of Manchester encoding in Section 4.2, the packet size (for N data symbols) and the number of symbols (from Equation 6.8) in an image can be related together with Equation 6.9b.

$$N_s = \left\lfloor C \cdot \left(\frac{f_l \cdot T_r}{D_o} \right) \right\rfloor \geq 2 \cdot (2 \cdot N + 6) \quad (6.10)$$

Rearranging Equation 6.10 for the Number of data symbols (N) will give:

$$N \leq \left\lfloor C \cdot \left(\frac{f_l \cdot T_r}{4 \cdot D_o} \right) - 3 \right\rfloor \quad (6.11)$$

Equation 6.11 can also be used to calculate the theoretical maximum throughput using Manchester encoding at any specific distance.

Adhering to the minimum criterion of Equation 6.11 ensures that, there is at least one packet received and at the same time there is an optimal throughput. The evaluation of this method is presented in Section 8.2.3.

Chapter 7

Comparison of Encoding Methods

This chapter presents a comparison between the Manchester and FSK encoding schemes described in Section 4.2. After comparing OOK, PWM and Manchester encodings, it was concluded that Manchester was the better option among these schemes. During this comparison, FSK encoding was not included since it requires a relationship between the image size and the distance between transmitter and receiver. As this relation is now established in Chapter 6, comparison on the basis of distance and throughput per frame can be performed for FSK and Manchester encoding schemes.

7.1 Manchester Encoding

The image should contain a minimum number of pixels to ensure that the symbols can fit a complete packet. The size of the image depends on the distance and the relationship between them is given by Equation 6.6. As described in Section 6.2, this image size should be at least twice that of the packet size (in pixels). In order to fulfil this criterion, the packet size should be adjusted (only the data, as the preamble is of fixed length) in order to ensure the reception of at least one packet. This adjustment helps in increasing the maximum distance for communication in light-to-camera systems.

From Figure 7.1, it can be seen that after a certain point (D_1), the packet size cannot be further decreased, as it would not be able to carry any data. At this distance D_1 , the packet size will only contain two symbols (one bit of data). After the point D_{max} (maximum communication distance), the

image size will be so small that it can not even fit one complete packet, and at this distance, the communication link will break.

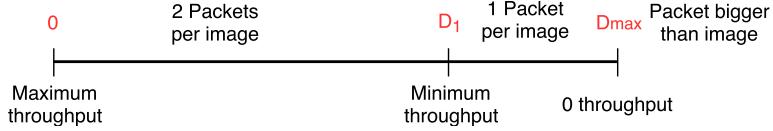


Figure 7.1: Division of the distance into regions based on throughput and packets per frame. The values of these distance points will depend on the size of the light source and the camera.

These distance points can be calculated using the relationship from the Equation 6.11 for a specific frequency and size of the given light source.

7.2 FSK Encoding

In this encoding scheme, the restriction is not on the reception of packets, but on the minimum number of symbols per image. As described in Section 5.2.3, at least 6 bands should be received in the image to decode the frequency information from the captured frame. In our system, we set the narrowest strip width to 19 pixels (5KHz frequency) and the widest strip width to 95 pixels (1KHz frequency), with a spacing of 2 pixels, which corresponds to a set of 39 available frequencies. The spacing of these frequencies is not linearly distributed because the width of a band and its corresponding frequency follow an inversely proportional relationship which is given by Equation 5.1.

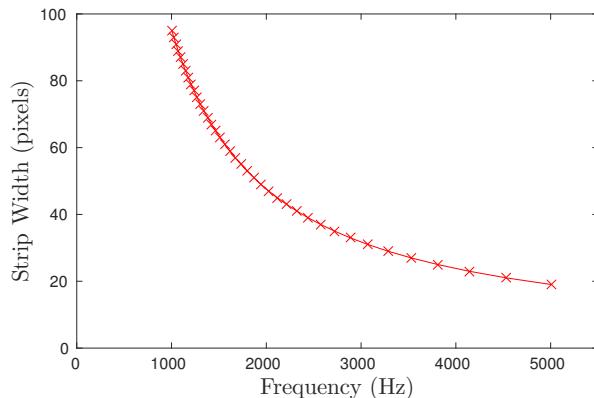


Figure 7.2: Frequency corresponding to the strip width with the spacing of 2 pixels. The distribution of frequencies is more sparse towards the lower end.

The reception of at least 6 bands can be related to the image size and the distance, which is given by the Equation 6.7. The maximum distance will be the point where the image size cannot accommodate at least 6 of the widest bands (lowest frequency).

In the implementation of RollingLight, the authors were only processing the frames which contain only one frequency and frames with multiple frequencies were rejected. In this thesis, we can detect multiple frequencies in a single frame using the method described in Section 5.2.3. This will double if two frequencies were detected in the frame. It is one of the major additions to the existing system.

7.3 Comparison between Manchester and FSK encoding

In FSK encoding, we used 39 different frequencies from 1 KHz to 5 KHz, and hence each frequency represents $\lfloor \log_2(39) \rfloor = 5$ data bits. This gives a throughput of 5 bits per frame. In order to match this throughput, the Manchester encoding should have 10 data bits in the packet (considering two packets in a frame). As the packet size is related to the distance, the point where the throughput of these encoding schemes becomes equal can be calculated using Equation 6.11.

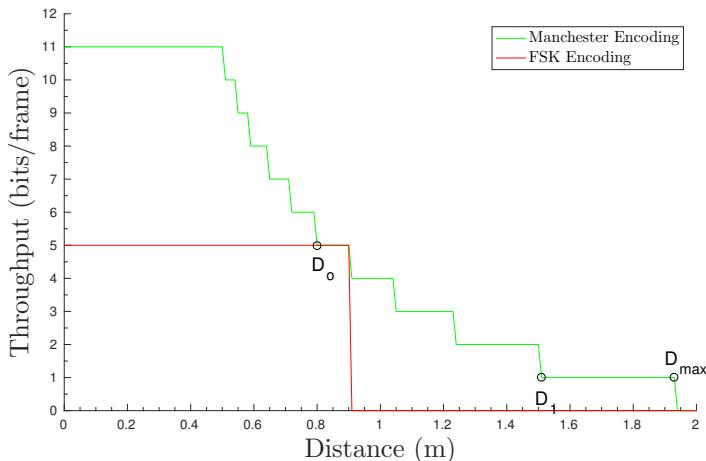


Figure 7.3: Throughput per frame of the Manchester and FSK encoding Schemes with the size of light as 15.5cm. The point D_0 represents the distance at which both of the encoding schemes will have same throughput, the point D_1 represents the distance where its no longer possible to fit 2 packets in a single frame for the Manchester encoding.

From the Figure 7.3, it can be concluded that: at short range, Manchester encoding will provide a better throughput per frame (until D_0). At a certain point (break-even point, D_0), the throughput of both the encoding schemes will become equal, this point will depend on the size of light source used. After this point, it will not be possible to fit 6 bands of the lowest frequency for the FSK encoding, it will mark the maximum distance for the FSK encoding scheme. Whereas, Manchester encoding will provide a stable link until the distance D_1 (at a lower throughput, but at least 1 packet per frame will be received). After the point D_1 , the image size will not be big enough to fit 2 complete packets. This would significantly reduce the stability of the communication link.

Chapter 8

Evaluation and Results

This chapter presents the results and evaluation of the developed system. Section 8.1 presents the experiments that refer to ways of removing the flickering as discussed in Section 4.1. Section 8.2 evaluates the performance of the distance model and examines the impact of improvements made by changing the channel capacity. The comparison of state of the art schemes based on the model is presented in Section 8.3.

For the evaluation of the system, we used the application discussed in Section 5.3. The application was used to capture images and store them for further evaluation as this application cannot perform real-time decoding of the captured image. In order to process the images offline, we created an OpenCV C++ program which is the same as that of the image processing pipeline in the application. With this method, multiple frames were captured and processed in order to evaluate and improve the system implementation offline.

8.1 Flickering Compensation

Due to the changing light intensity of the light source to transmit data, flickering is observed in some VLC systems. As discussed in Section 4.1, the indirect flickering was observed in OOK and PWM encoding. To observe the cause of flickering, the transmitted light signal is captured using a photodiode.

Setup and Procedure

The light source was connected to a microcontroller, which encodes the data with the given encoding scheme and then modulates the light accordingly.

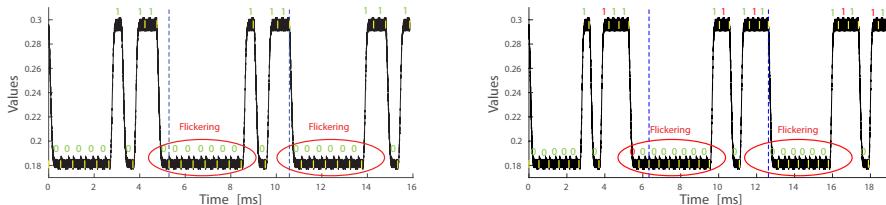
This setup was placed on a table horizontally, emitting light towards a white wall. The photodiode was kept stationary in direct line-of-sight of the light, which helped in getting a strong signal. The voltage levels from the photodiode were saved for analysis. Additionally, 5 volunteers were invited to perform an experiment to assess the users' perception of flickering. The transmitter was set to send a sequence of data repeatedly for each of the encoding schemes.

Three encoding schemes were tested under two conditions, first without any modifications and second after adding the compensation symbols for flickering. In the first case, the packet sequence consisted of the preamble, 4-bit data and parity (the number of symbols vary with the corresponding encoding scheme). For the second case, the same data was sent with the addition of compensation symbols in the packet.

Results

The flickering in the system was observed in OOK and PWM encoding but not in the Manchester encoding scheme. It was caused mainly due to either the presence of a long sequence of the same symbol ('0' or '1') or unequal average light intensity over a period of 10ms.

OOK: Without the use of compensation, the flickering was very dominant and could have been seen directly. This was due to the sudden change in light intensity after the preamble bits. In order to validate that the flickering was due to the preamble, data was chosen as such to have 3 HIGH (1) bit in the packet, but flicking was still observed as depicted in Figure 8.1(a).



(a) OOK encoding without compensation. (b) OOK encoding with 1 compensation symbol after every 2 symbols.

Figure 8.1: **Flickering in OOK:** Light-intensity of the OOK signal with and without compensation.

In order to minimise the effect of flickering, 1 compensation symbol after every 2 symbols were added. These compensation symbols were determined

based on the two previous symbol trying to keep the same average light intensity. Even after adding the compensation, the flickering was clearly observable by all the volunteers due to the preamble as shown in Figure 8.1(b). Adding more symbols for compensation will mean that each data bit will have 2 symbols, which will convert OOK encoding to Manchester encoding. Hence, we were not able to remove flickering from the OOK encoding and it was not implemented for decoding.

PWM: In case of PWM encoding, the flickering was observed due to the change in average ON periods in between packets. This can be seen in Figure 8.2(a), vertical blue lines indicate the end of a packet and the horizontal red line represents the average intensity of that packet. As the average is changing in every packet, the flickering was reported by 4 of the volunteer.

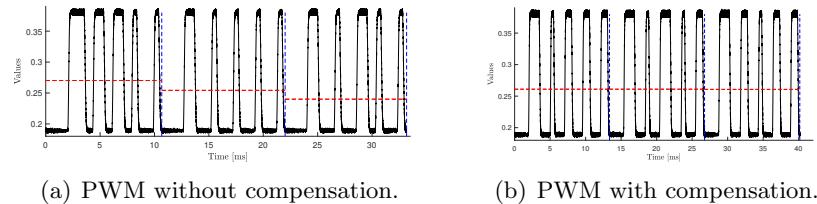


Figure 8.2: **Flickering in PWM:** Comparing the average-light intensity of the PWM signal with and without the use of compensation symbols.

In order to remove flickering, 1 compensation symbol was added to every 3 symbols (1 data bit). This compensation symbol will bring the average to 50%, as in every 4 symbols, 2 symbols will be HIGH (1) and other will be LOW (0). After implementing this, the flickering was removed as reported by all the volunteers. This change can be seen in Figure 8.2(b), as now the average (horizontal red line) remains constant.

Manchester Encoding: No flickering was observed in the Manchester encoding by any volunteer as the average remain constant with every data bit. And all the averages of the light intensity (horizontal red line) were the same. This can be seen in Figure 8.3.

8.2 Distance model evaluation

This section presents the evaluation of the developed mathematical model in Chapter 6. It discusses the performance of the distance calculation method,

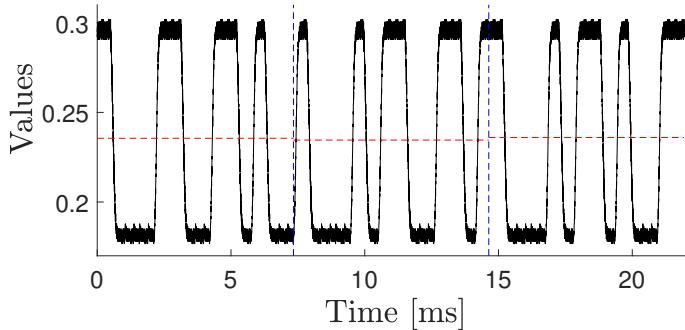


Figure 8.3: No flickering was observed in the Manchester Encoding as the average light intensity remains constant in the whole packet.

along with the symbol capacity calculation method. Moreover, the impact of having an adaptable packet size is discussed.

8.2.1 Distance calculation

Methodology

As described earlier, the image size depends on the object size and the phone parameters. The image was captured when the light was constantly on (i.e. no modulation), this increases the accuracy of the circle detection algorithm of OpenCV. The image was captured at various distances between the transmitter (light source) and the receiver (smartphone), varying from 30 cm to 3 m. 4 different phones were used to verify the method as each phone had different focal length and image sensor.

Results

The distance between the transmitter and the receiver was calculated using Equation 6.4 with the camera parameters (i.e. pixel size and focal length). The distance was calculated for each captured image of the light source and the error of the estimated value was obtained. The results of the experiment are presented in Figure 8.4.

All the smartphones have a small RMS error ranging from 2.92cm to 2.15cm except iPhone 7 (it might be due to the difference in actual and stated pixel size value), which is shown in Table 8.1. This error is acceptable as it will have a small effect on the symbol capacity calculation based on distance. The error in the calculated values could be due to a slight movement of

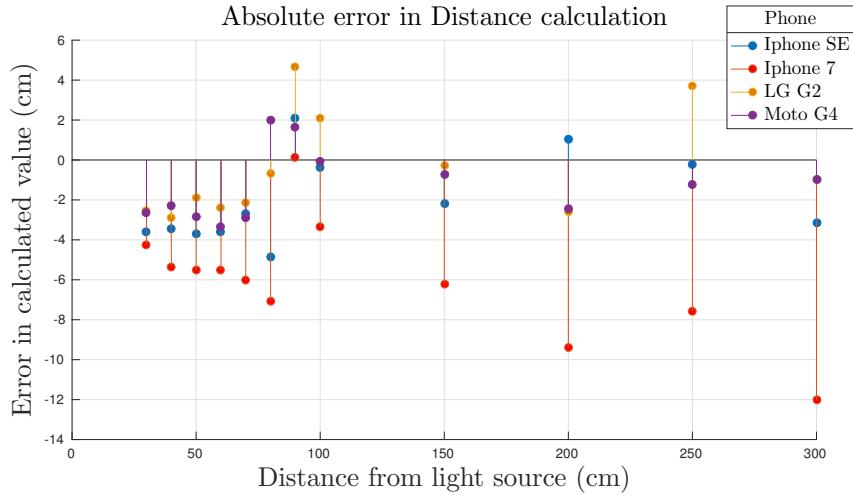


Figure 8.4: **Actual distance vs Calculated distance.** Error in distance calculation performed using 4 different smartphones.

the camera while capturing the image or due to a quantisation error while measuring the image size in pixels.

	iPhone SE	iPhone 7	LG G2	Moto G4
Maximum absolute error (cm)	4.85	12.3	4.6	3.35
RMS error (cm)	2.92	6.67	2.53	2.15

Table 8.1: **Actual distance vs Calculated distance.** Maximum error and RMS error values per phone. Except for iPhone 7, all the RMS error values are in range of 2.92cm to 2.15 cm, which is an acceptable value.

8.2.2 Symbol Capacity Calculation

Methodology

After showing that we can calculate the distance between the light and the smartphone, we can continue with the evaluation of the number of symbols that can fit in the image size based on Equation 6.8.

The images captured during the previous experiment using the Moto G4 phone were used to calculate the number of symbols the image can have. Using this calculated image size and the Equation 6.5, the possible number of symbols were predicted. For the evaluation, 5 different frequencies were used (1, 2, 3, 4, 5 KHz) at a distance varying from 30 cm to 3 m. The images of

these modulating lights were calculated and the number of (whole) symbols present in them was counted to compare with the values obtained from the formula. In this experiment, the size of the light source is 15.5cm.

Results

The number of symbols in the captured image follows the measured values closely. The increase in the number of symbols is more noticeable at close distances where the image size is bigger. The positive observation is that even at a distance of 2 meters, 13 symbols were obtained which is enough to transmit 1 packet.

A comparison between the measured values and the calculated values is presented in Table 8.2. It can be observed that for most points, the measured and calculated values are the same. The difference in values is due to the fact that, the calculation technique always considers that the first symbol starts at the initial point of the image, which can lead to round-off error in some cases.

Frequency (Hz) / Distance (cm)	1000 Hz	2000 Hz	3000 Hz	4000 Hz	5000 Hz
10	11/11	22/22	34/34	44/45	56/56
20	11/11	21/22	34/34	44/45	55/56
30	10/11	21/22	34/34	44/45	56/56
40	11/11	22/22	33/34	44/45	55/56
50	10/10	21/21	32/32	43/43	54/54
60	8/9	18/18	26/27	35/36	43/45
70	6/7	15/15	23/23	31/31	36/38
80	6/6	13/13	20/20	27/27	33/34
90	6/6	12/12	18/18	24/24	30/30
100	5/5	9/10	16/16	21/21	27/27
150	3/3	6/7	10/10	13/14	18/18
200	2/2	5/5	8/8	10/10	13/13
250	1/2	4/4	6/6	8/8	10/10
300	1/1	3/3	5/5	7/7	9/9

Table 8.2: **Number of symbols vs Distance.** The number of symbols were measured for 5 different frequencies and 14 different data points. Each cell in the graph represents (Measured value/Calculated value), cells in bold represents mismatch between the values.

Figure 8.5 shows the pattern of the measured number of symbols at different frequencies. It is noticeable that the number of symbols does not decrease linearly after a distance of 100cm. This is mainly caused by the

fact that the detected image size decreases but is still large enough to include the same number of symbols.

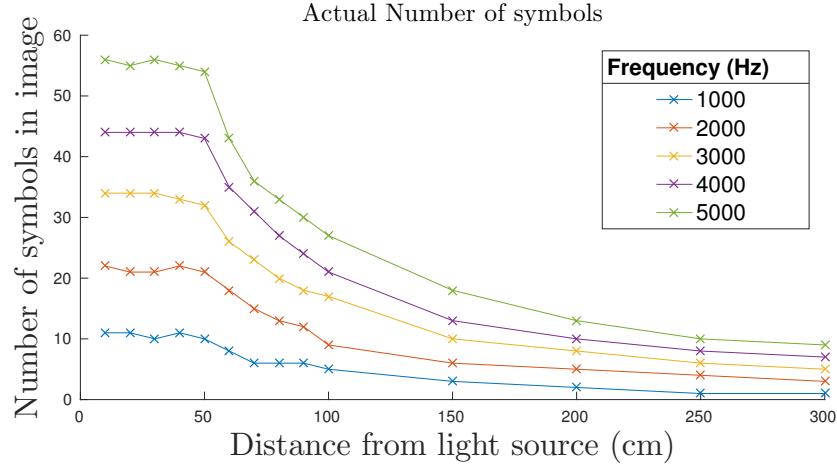


Figure 8.5: **Number of symbols vs Distance.** Measured values for 5 different frequencies, Due to change in the size of image, there is rapid decrease in the number of symbols in a frame.

8.2.3 Adapting to Variable Distance

This section evaluates the impact of the proposed packet optimisation method based on the calculated distance and the image size that was presented in Section 6.2. In the state-of-art systems, the packet size is fixed or the frequency set does not change. Having an adaptable packet size helps in increasing the flexibility of the system in terms of the distance.

Methodology

The goal of this experiment was to verify the proposed adaptable packet size with the fixed packet size. In this experiment, the fixed data size for the Manchester encoding was evaluated using data sizes of 4 and 8 bits (8 and 16 symbols), with a constant preamble. The distances between the transmitter and the receiver started at 30 cm and increased in steps of 10 cm until 1 m and later on with steps of 20 cm until 2.4m. For each distance 50 frames were processed and the average throughput per frame was taken for all the three setups.

In the first two cases, a fixed packet size was used following the state-of-art method with no feedback to the transmitter. Whereas in the third case, an

adaptable packet size method was used which can change packet size based on the distance. As the developed application cannot work in real-time, it captures 50 images and saves them for offline processing. A C++ program was used to calculate the distance based on the size of the first captured image (non-modulating light). This calculated distance is then converted to the optimal packet size. The microcontroller is reprogrammed with this packet's size in order to achieve an optimal channel capacity.

Result

The result of the average throughput of 50 captured frames vs the distance is shown in Figure 8.6. The throughput of fixed packet sizes decreases gradually because not all 50 frames contain the same number of packets. It resulted in different throughput per frame and hence the average is different at each distance.

As it can be observed in Figure 8.6, a small packet size underestimates the channel capacity but ensures that the whole packet is received at a longer distance. It was seen that until 90 cm at least 2 complete packets were received with 4-bit information. This gave a reliable communication link as each frame contained a minimum of 1 packet (keeping the average throughput above 4bits/frame). After this point, there is no guarantee of receiving at least 1 packet, which decreased the throughput for a distance greater than 90 cm. As a consequence, 4 bits per packet may offer reliability but is not the most efficient for all distances.

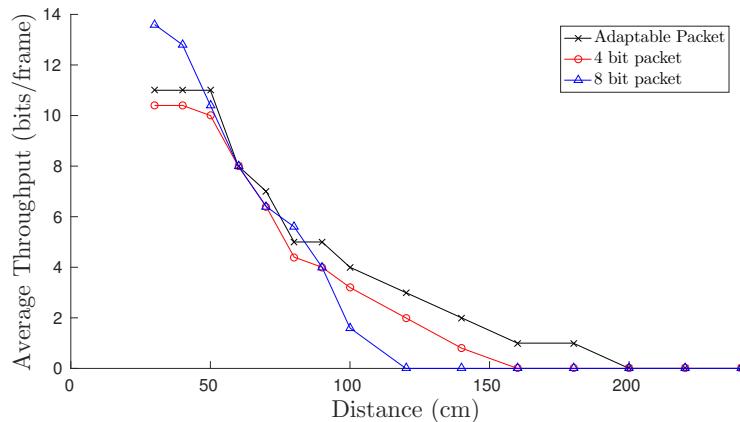


Figure 8.6: **Throughput vs packet size.** Average throughput for 50 captured frames, the adaptable packet size improves the channel capacity.

Whereas the biggest packet of 8-bit information gave the highest throughput initially, but as the distance was increased, its throughput went down

quickly. For example, one whole packet cannot fit at a distance greater than 60cm. After this point, not all frames contained a complete packet, which decreased the throughput rapidly. From 1 m onwards, the image size was not big enough to fit a whole packet, so the throughput became zero. As a consequence, an 8-bit per packet may offer higher throughput at close range, but with the increase in distance, throughput reduces quickly.

From Figure 8.6, it is clear that an adaptable packet does not give the best throughput in close range because packet size is adjusted such that at least two whole packets fit in the image size (throughput is always $N/2$). But this method keeps up with the distance by reducing its packet size and ensures that communication link is stable to the longest distance possible. This method compromise on the throughput at the closer distance to increase reliability at longer distances.

8.3 Comparison of State of the Art schemes

As described in Chapter 2, each of the previous works uses different encoding schemes, different sizes for the light source and different capture frame rates of the camera, which results in the stated throughput. A comparison of these works based only on the throughput would not be fair as there are too many variables (i.e the size of light source, distance, operational frequency, frame rate, and the exposure time).

Throughput per frame

In order to fix as many variables as possible, it was considered that all the system uses the same smartphone with a fixed exposure time of $161.6\mu s$ ($T_r = 10.5\mu s$) and the light image occupies the whole frame of 1080 rows. These values are a particular case and can be generalised for any set of values. With the help of these values, the number of symbols received in one captured frame was calculated based on the reported operational frequency by using the developed mathematical model. This allows for analysing the performance only on the basis of the operational frequency and the code density of the encoding scheme.

The final results of the evaluation are presented in Table 8.3. During the analysis, any optimisation present in a particular work is applied and the best case scenario is considered, as indicated by *.

After the evaluation it was found out that camera properties play a significant role in determining the throughput of the system. and out of the state of the art the best method in term of throughput per frame was CMOS for

VLC [1], but in real scenario it had a drawback of short distance. Manchester encoding alone was found out to be best due to its low overhead. The distance and link reliability was increased with the help of packet optimisation.

Project	Encoding Scheme	Reported Throughput	Operational Frequency	Number of symbols per frame	Calculated Throughput (bits/frame)
CMOS for VLC [1]	OOK + Manchester	125 and 375 bytes/sec	2 KHz	22	16 or 72 bits/frame
Visible Light Landmarks [8]	BFSK	1.25 bytes/sec	2 KHz	22*	8.5* bits/frame
RollingLight [5]	FSK	11.32 bytes/sec	-	1	5 bits/frame
DynaLight [11]	OOK + Manchester	-	3 KHz	32*	10* bits/frame
Smartphone as receivers [9]	PPM	156.25 bytes/sec	0.5 ms [#]	2 bits/ms	11.32 bits/frame
This project	Manchester	-	5 KHz	23*	11* bits/frame

Table 8.3: **Comparison of he State-of-the-art:** on throughput/frame. * presents the best case scenario. # 1 packet was transmitted every 0.5ms.

Chapter 9

Conclusions

9.1 Conclusions

To conclude, through the completion of this thesis a number of contributions were made to the field of the light-to-camera communication. First, a thorough analysis of the basic encoding schemes (OOK, PWM, Manchester) was carried out and it was found that Manchester encoding was the best amongst these schemes, as it has a low data overhead and can maintain constant light intensity while avoiding flickering effects.

Secondly, a significant portion of this thesis focuses on developing a mathematical model which can calculate the distance between the transmitter and the receiver based on the size of the light source and the camera parameters. Our distance estimation method results in a small RMS error value of 2.9cm. The calculated distance was used in determining the optimal packet size to ensure a stable communication link. This adjustable packet size helped in balancing the trade-off between throughput and link reliability. It meant that at close range, throughput would not be highest, but this method helped in keeping up with the varying distance by reducing its packet size to ensure that the communication link was stable up to the longest distance possible.

Finally, the state of the art light-to-camera systems were evaluated on a common ground using the developed mathematical model and it was found that the camera properties play a significant role in determining the throughput of the system.

Bibliography

- [1] C. Danakis, M. Afgani, G. Povey, I. Underwood, and H. Haas. Using a cmos camera sensor for visible light communication. In *2012 IEEE Globecom Workshops*, pages 1244–1248, Dec 2012.
- [2] Tian Hao, Ruogu Zhou, and Guoliang Xing. Cobra: Color barcode streaming for smartphone systems. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys ’12, pages 85–98, New York, NY, USA, 2012. ACM.
- [3] Wenjun Hu, Jingshu Mao, Zihui Huang, Yiqing Xue, Junfeng She, Kaigui Bian, and Guobin Shen. Strata: Layered coding for scalable visual communication. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom ’14, pages 79–90, New York, NY, USA, 2014. ACM.
- [4] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom ’14, pages 447–458, New York, NY, USA, 2014. ACM.
- [5] 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, pages 167–180, New York, NY, USA, 2015. ACM.
- [6] Liqun Li, Pan Hu, Chunyi Peng, Guobin Shen, and Feng Zhao. Epsilon: A visible light based positioning system. In *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, NSDI’14, pages 331–343, Berkeley, CA, USA, 2014. USENIX Association.
- [7] Tianxing Li, Chuankai An, Xinran Xiao, Andrew T. Campbell, and Xia Zhou. Real-time screen-camera communication behind any scene. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys ’15, pages 197–211, New York, NY, USA, 2015. ACM.
- [8] Niranjini Rajagopal, Patrick Lazik, and Anthony Rowe. Visual light landmarks for mobile devices. In *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, IPSN ’14, pages 249–260, Piscataway, NJ, USA, 2014. IEEE Press.
- [9] Stefan Schmid, Linard Arquint, and Thomas R. Gross. Using smartphones as continuous receivers in a visible light communication system. In *Proceedings of*

- the 3rd Workshop on Visible Light Communication Systems*, VLCS '16, pages 61–66, New York, NY, USA, 2016. ACM.
- [10] S. Schmid, G. Corbellini, S. Mangold, and T. R. Gross. An led-to-led visible light communication system with software-based synchronization. In *2012 IEEE Globecom Workshops*, pages 1264–1268, Dec 2012.
 - [11] Michail Vasilakis. Dynalight: A dynamic visible light communication link for smartphones. Msc thesis, TU Delft, December 2015.