Receiving data through light with low-end smartphones

by

Shaan Hossain

Supervisors:

Koen Langendoen, Marco Zuniga

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 19, 2022

Abstract

Although the demand for wireless communication continuous to grow, the number of frequencies available in traditional data transmission is limited. Visible Light Communication (VLC) is a promising alternative communication method proven to work on high-end devices. However, large-scale adoption requires VLC to also work on everyday machines. Yet, no published research has proven effectiveness on low-end devices. This work describes how a communication system using light can be implement on a low-end smart phone, using a LED as a transmitter. Furthermore, it compares two methods of (de)modulation, namely On-Off keying and Differential Manchester Code. Evaluation results demonstrate a working system with a maximum bit-rate of approximately 3000bps.

1 Introduction

The rise of wireless communication is at an all-time high and climbing in popularity. Think of wireless headphones, paying wirelessly and even operating home appliances wirelessly. The electromagnetic wave-spectrum these devices consume is getting more crowded and researchers are looking into an alternative form of communication, that is, with light [1]. Recent studies on VLC have proven that bit-rates around 40bps at long ranges are possible with modern devices [2]. However, what if one does not have the latest and fastest smartphone? Using a device that is slower in computation and equipped with a slightly worse camera might hamper an application that relies on fast and consistent data transmission through VLC. Older smartphones are limited in their computational power, so possible applications for these devices must be considered.

In this paper we research how we can demodulate an incoming optical signal with low-end smartphones and see how it affects latency and throughput. The following research question aim to encapsulate the main topic:

> *How can we demodulate an optical signal with low-end smartphones that are limited in computational resources affecting latency and throughput?*

By answering this question we determine the lower limit in efficiency for VLC. Furthermore, we conclude what strategies can be applied best for demodulation on low-end smartphones.

This paper is organised as follows. Section 2 mentions related work on this topic. Section 3 describes how the potential dangers of this research and how these should be handled. In section 4, the process cycle of VLC is mentioned as well as a description of the transmitter and the receiver used in this research. Section 5 gives a detailed description of what experiments were carried out and in section 6 results of these experiments are analyzed. Finally, conclusions and recommendation for future work are given in section 7.

2 Related Work

Work related to using smartphone cameras for VLC has shown that this communication method does have its limits. We briefly review previous research in the categories: data format, detection/demodulation and device limits.

2.1 Data format

The format of the data is almost identical in all experiments that have been done in previous research. This format consists of- a header, a payload and some sort of trailer. The trailer can be used for error detection [3] or to detect the end of the message [4]. Furthermore, it can be concluded that making use of the rolling shutter effect is one of the most optimal methods of data transfer [5]. The rolling shutter effect occurs when a camera captures an object that changes in light intensity at a high frequency. This effect can be used to encode bits into dark and bright lines, as seen in figure 2. At last, On-Off Keying and Manchester Code are the dominating (de)modulation methods in the field of VLC. Using Manchester Code adds the benefit of being able to detect errors relatively easily. Unfortunately, this does come at the cost of throughput as will be explained in section 5.2.

2.2 Detection/demodulation

A range of approaches have been used to detect, track and demodulate an incoming optical signal. A possible approach is removing color data from the image, leaving only black and white values [5; 6]. This is followed by applying a filter on the black and white image to properly distinguish on and off signals. This approach is considered outdated since the algorithms used are fast and simple but rather inconsistent and not very reliable[5]. The simplicity of the algorithms was particularly convenient for the first generation of smartphones that did not have as much computation power as present-day smartphones. Algorithms found in more recent research rely on the use of blurring images and comparing different buffered frames. These frames are stored sequentially and are compared to each other in order to identify the position of a LED in the image. This adds considerately more computational overhead, but is no issue, however, since more recent smartphones have relatively more computation power. A commonality that both generations of research in this field share is the use of thresholding during image analyzation. A certain average is calculated from the brightest and the darkest values from the image. This average acts as threshold to distinguish a '1' (bright line) and a '0' (dark line). An unique method for optical signal detection is by alternating between illumination and communication periods [7]. These periods are fixed length and allow receivers to synchronize and transfer data alternately.

2.3 Device limits

Some key features for using smartphone cameras as receivers are the option to change the exposure time and ISO values of the camera. By adjusting these values we can enhance the receiver, a feature that was lacking on older generation (Android) smartphones and their APIs [5; 8]. Research done with older devices has shown a bit-rate between 800bps and

1400bps with distances ranging between 3cm and 1m between the transmitter and receiver [6]. Note that the smartphones used in these experiments were composed of 20fps cameras with a resolution of 640 by 480 pixels. More recent studies have shown that greater distance can be achieved with similar transfer speeds on modern smartphones [3; 4]. From this we can conclude that the bit-rate mainly depends on the quality of the camera and not just the processing speed of the images.

3 Responsible Research

When working with low-level hardware and LEDs certain concerns must be addressed. Flickering light can be experienced as disruptive to some (and if not handled properly by all) human eyes. A flickering frequency between 50Hz and 90Hz must be avoided at all times as this may trigger epileptic seizures [9]. Furthermore, looking directly into the LED should also be avoided as this might damage the eyes of the user. These risks can be largely mitigated by dimming the LED to a minimal brightness or turning off the device completely when not in use.

4 Background and Methodology

In VLC, demodulation is one of the very last steps in data transfer with light. Several steps need to be taken care of before we can actually decode light into ones and zeros. At first the transmitter (in this case a LED) must be detected. When the distance between the transmitter and receiver is too large the camera might struggle to detect the LED. This might also be the case when the ambient light intensity is too high and the LED 'blends in' in its surroundings. Secondly, once the LED is detected it must be tracked. Due to incoming noise and other influences, such as camera shake, the LED might not be in one particular spot while it is being detected. At last, the signal from the LED can be demodulated. Finding the optimal scheme for demodulation can be challenging, especially for low-end phones. A very extensive and sophisticated algorithm might result in low Bit Error Rates but comes at the cost of relatively more computation. A balance in complexity and throughput must be found to accommodate the hardware restrictions on low-end phones.

4.1 Methodology

Transmitter

The hardware device that is in charge of transmitting the optical signal is an Arduino Due with a LED. The LED on the Arduino is turned on and off at a frequency of 8kHz. This type of flickering is invisible to the human eye but can be detected by cameras. The transmitter, as seen in figure 1, does also contain a potentiometer which acts as a dimmer for the LED. The transmitter is powered by a USB-cable connected to a computer on which input is entered for the LED.

Receiver

The receiver for this experiment is a low-end budget Android smartphone. Present-day market allows to purchase an Android smartphone with a 240 fps camera for approximately

Figure 1: Arduino Due with LED

€150. In this research a Samsung Galaxy S8 running on Android 9 was used. The smartphone will run an image analyzer app created by professor Koen Langendoen with Android Studio. This app detects, tracks and demodulates the incoming optical signal. The app allows to adjust the ISO values of the camera in order to enhance the rolling shutter effect. This is particularly useful in settings where there intensity of ambient lighting is high and might influence the detection of the LED from the camera.

According to professor Langendoen a bit error rate lower than 0.1 is desired. For the receiver to demodulate consistently and reach this goal the following trade-offs must be considered:

1. The number of bits in the payload: having more bits as payload can result in higher error rates. Having fewer bits as payload is less likely to cause errors but will result with a lower throughput.
2. The distance between receiver and transmitter: a larger distance between the two devices is desired. However a distance too large might result in demodulating a message incorrectly or perhaps even no demodulation at all.

When the camera is pointed to a LED, images will be processed constantly in streamlike fashion. Such an image can be seen in figure 2. The app processes the image and translates the bars created by the rolling shutter effect into a stream of symbols. Once the pre-defined header, is detected the data is demodulated and outputted on the smartphone screen.

5 Experimental Work

Setting up the experiment environment consists of determining how the data is formatted, what (de)modulation schemes are used and what additional features are be added to the application.

5.1 Data format

The complete message that is modulated consists of: a header, a payload and (if desired) an error detection code. The length of the message we can modulate depends on the frequency of the clock of the transmitter, the frame rate of the camera and the distance between the transmitter and receiver. A higher clock frequency allows us to fit more symbols in a single image since the LED turns on and off faster. However, if the clock frequency is too high the smartphone camera will not be able to detect the symbols properly. Figure 2 illustrates how the rolling shutter effect is used to demodulate the image into symbols. The symbol marked with a circle (○) is the first symbol of the message. Note that the payload in this figure is still encoded with DMC and must be decoded first in order to obtain the data.

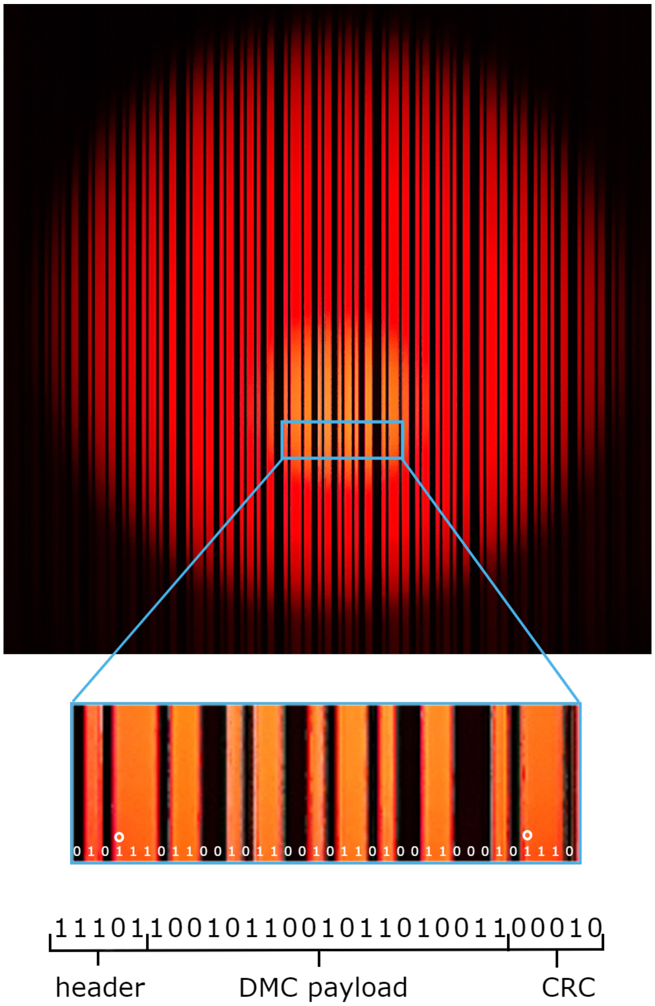

Figure 2: Rolling shutter modulating a message in DMC with CRC

5.2 (De)modulation schemes

Firstly, we used OOK as a (de)modulation scheme. This is the simplest form of (de)modulating a signal since the bits are not encoded and sent as such. The drawback of this simplicity is that any combination of bits is possible making it more susceptible to errors. This drawback is often eliminated in alternative (de)modulation schemes. The benefit of using OOK is that all bits in the message are used as the payload, thus resulting in a relatively high throughput.

Secondly, we used Differential Manchester Code (DMC). DMC ensures that every bit in the payload is encoded by either '10' or '01'. Because of this property it is impossible to have 3 or more identical symbols consecutively. With this property we can use '11101' as the header of the message. Since this sequence of symbols can not occur in the payload, the header is easily detectable and cannot be mistaken as part of the payload. This also means that the length of the payload is twice as long. Thus, the throughput for DMC is theoretically half as that of OOK.

5.3 Application features

In order to detect whether messages are received correctly we used error detection codes. In particular, we opted for cyclic redundancy check (CRC). Other options such as parity check or checksum we considered not to be reliable enough to detect errors. This is because the CRC's probability to detect an error in the message is much higher than that of the parity bit or the checksum. Alongside CRC, the option to change the exposure time of the camera was also added as a feature to the app. As mentioned in section 4, high intensity ambient lighting might influence the detection of a LED. By adjusting the exposure time of the camera we can make the LED easier to detect in a variety of ambient light scenarios as well as improve detection at larger distances. Also, a logging feature was added to the application. This feature logs the processing time and the demodulated message over a user-defined period of time of every image that is analyzed. Lastly, the detection and tracking algorithm present in the app that is used in this research makes use of bounding boxes and histograms. The bounding boxes are used to indicate where the LED is detected. Once the image is detected it is spliced into sections that fit into a histogram. Then, a thresholding filter is applied to distinguish bright and dark lines. At last, the histogram translates its bars into ones and zeros after which demodulation of the translated sequence takes place.

6 Results

The experiment is conducted over multiple distances so the degradation in performance of the smartphone can be captured. For the distances of 3cm, 6cm, 10cm and 20cm all combinations of DMC, OOK both with and without CRC are tested. Different scenarios with ambient lighting proved to make no significant difference in results. This is mainly due to the ability to change the exposure time and the ISO value of the camera.

6.1 Effective throughput, false positive rate and processing time

As mentioned in section 4.1 the transmitter clock is running at a frequency of 8kHz. Theoretically this means that 8000 symbols are being transmitted per second. However, only a portion of the transferred message contains actual data. The amount of bits of actual data sent per second is what we call effective throughput. Table 1 illustrates how such a message is structured, the efficiency of data symbols per message (symbol efficiency) and its theoretical throughput. The theoretical throughput is the throughput that would be achieved if all messages from the transmitter are read correctly by the receiver. Combining this percentage of correctly read messages with the theoretical throughput we get the effective throughput, which is illustrated in figure 3. This figure shows that over all distances CRC does not seem to have much of an impact on the effective throughput. More messages arrive correctly with CRC but this is at the cost of lower symbol efficiency and thus resulting in a similar throughput per (de)modulation scheme. OOK performs approximately a factor 8 worse than DMC at close distances. This is because the receiver accomplishes to detect the header correctly. However, due to the possibility of long sequences of identical symbols it fails to distinguish the length of this sequence. For instance, a sequence of '111' could be mistaken for '1111' by the receiver. Other researchers managed to achieve a throughput of 1200bps in near similar conditions for OOK [7]. The achieved effective throughput of at most 350bps implies that the tracking and detection algorithm used in this experimental setup is inefficient for OOK demodulation. Furthermore, the effective throughput decreases near exponential as the distance grows for DMC. At distances larger than 6cm the camera produces an inaccurate image from the rolling shutter effect and therefore leads to incorrect results. The tracking and detection algorithm is thus not effective at long distances either.

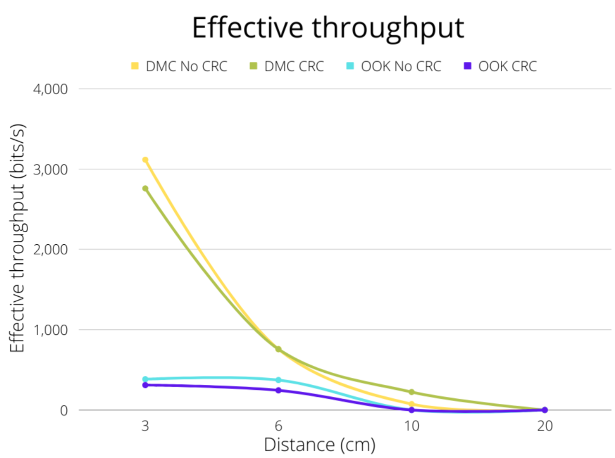

Figure 3: Effective throughput over different distances

A message is marked as a false positive when anything but the correct data is outputted. This can occur when there are errors present in the payload or when the CRC fails. The percentage of messages that are demodulated into false positives in figure 4 reflect similarly to the effective throughput. OOK performs significantly worse than DMC, approximately a factor 20 worse. The false positive rate is at its highest at a distance of approximately 10cm. At this distance the receiver is close enough to detect the header but too far away to correctly demodulate the remainder of the message. At a large enough distance the receiver fails to detect the header at all. The consequence of this is that no message is demodulated at all and thus no error is detected. This figure also shows that the addition of CRC reduces the false positive rate by a factor 9 on average for DMC and by a factor of 2 on average for OOK. The computational overhead and the additional bits for CRC are overshadowed by the lower false positive rate and higher effective throughput. Figure 5 shows

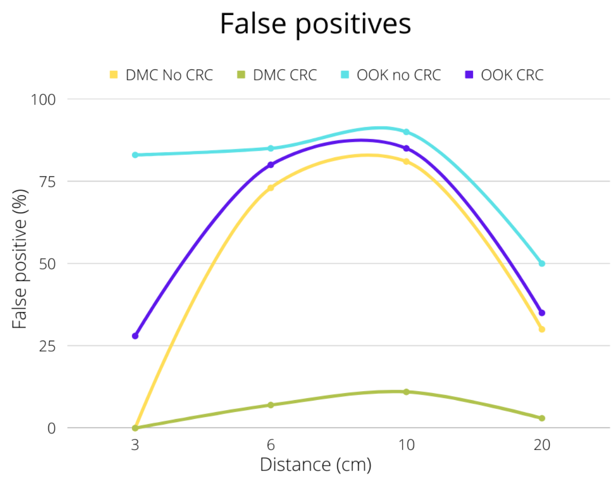

Figure 4: False positives over different distances

that there is no significant difference in processing time for different (de)modulation schemes. It would be expected that OOK has a lower processing time since it has less symbols to process than DMC. However, this figure shows the contrary. The longer processing time can be derived from the fact that an OOK message occurs more often than a DMC message. As mentioned in section 5.2 an OOK message is practically halve the size of a DMC message and thus will occur in the image twice as much. In other words, twice as many messages are transmitted in the same image thus twice as much processing needs to be done. However, the processing time for OOK in figure 5 is not twice as high either. This is because as soon as an incorrect payload is detected, the whole message is discarded. Ultimately, a large fraction of all the messages with OOK will be discarded during the processing of the image, resulting in a slightly longer processing time.

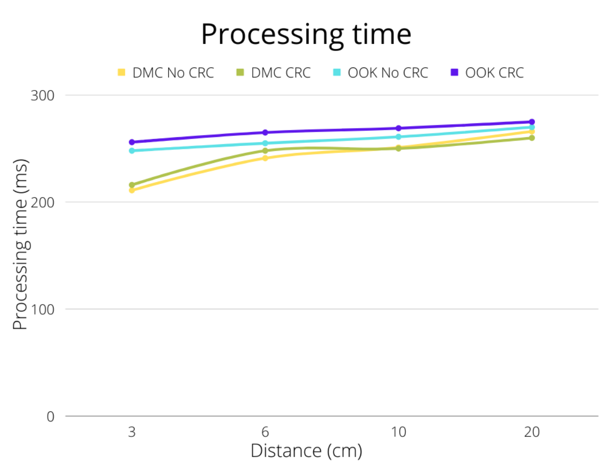

Figure 5: Processing time over different distances

(De)modulation scheme	Data format (number of symbols)				Symbol efficiency	Theoretical throughput (bps)
	Header	Encoded payload	CRC	Total		
OOK	5	10	-	15	0.66	5333
OOK with CRC	5	10	4	19	0.53	4211
DMC	5	20	-	25	0.40	3200
DMC with CRC	5	20	4	29	0.34	2759

Table 1: Data format of messages

6.2 Possible applications

The effective distance of this system and the achieved effective throughput of at most approximately 3kbps limits the possible applications for this system. However, several use cases still come to mind. A possible use case would be using LEDs to send information about the schedules and opening times of certain rooms around campus. The LED would be placed next to the entrance of a room and transmit information regarding its availability. Another possibility would be to let the LED serve as some sort of key for a smart device. This key would allow the smartphone to connect to a server and operate this device remotely as seen in a similar research [2]. Use cases from other experiments that relied on higher throughput and better performing receivers are not feasible with current lower end smartphones. That is, with the setup used in this research.

7 Conclusions and Future Work

This paper analyzes the performance of different demodulation schemes in VLC using a low-end Android smartphone. The smartphone makes use of an app that demodulates the incoming optical signal. Following the setup in this research a higher effective throughput is obtained by using DMC as (de)modulation scheme. The message that is transmitted is 25 symbols long and serves an effective throughput of approximately 3000bps. The detection and tracking algorithm present in this setup is not very efficient for distances longer than 6cm compared to algorithms from other researches. For future work it is encouraged to experiment with different clock speeds as well as different lengths of the message. Using different low-end smartphones with similar camera features may still give different results as the computational power might differ. In addition to CRC, error correction codes (ECC) can be used to correct faulty symbols and improve the current bit-rate. At last, changing the detection and tracking algorithm to a more generic algorithm mentioned in section 2 might improve the maximum effective throughput of this setup.

8 Acknowledgement

This research was aided and guided by Professor Langendoen and Professor Zuniga. The Android app used in this research was created by professor Langendoen. Arduino Code for modulating the LED on the transmitter was partly provided by PhD candidate Keyarash Ghiasi and co-student Eric Kemmeren.

References

[1] M. Kavehrad, "Optical wireless applications: a solution to ease the wireless airwaves spectrum crunch," in *Broadband Access Communication Technologies VII* (B. B. Dingel, R. Jain, and K. Tsukamoto, eds.), vol. 8645, pp. 109 – 115, International Society for Optics and Photonics, SPIE, 2013.

[2] J. J. Yang and J. A. Landay, "Infoled: Augmenting led indicator lights for device positioning and communication," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, (New York, NY, USA), p. 175–187, Association for Computing Machinery, 2019.

[3] R. A. Sharma, A. Dongare, J. Miller, N. Wilkerson, D. Cohen, V. Sekar, P. Dutta, and A. Rowe, "All that glitters: Low-power spoof-resilient optical markers for augmented reality," in *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 289–300, 2020.

[4] K. Ahuja, S. Pareddy, R. Xiao, M. Goel, and C. Harrison, "Lightanchors: Appropriating point lights for spatially-anchored augmented reality interfaces," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, (New York, NY, USA), p. 189–196, Association for Computing Machinery, 2019.

[5] 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*, pp. 1244–1248, 2012.

[6] A. Duque, R. Stanica, H. Rivano, and A. Desportes, "Unleashing the power of led-to-camera communications for iot devices," p. 4, 10 2016.

[7] S. Schmid, L. Arquint, and T. 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, (New York, NY, USA), p. 61–66, Association for Computing Machinery, 2016.

[8] J. Ferrandiz-Lahuerta, D. Camps-Mur, and J. Paradells-Aspas, “A reliable asynchronous protocol for vlc communications based on the rolling shutter effect,” in *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2015.

[9] “Ieee recommended practices for modulating current in high-brightness leds for mitigating health risks to viewers,” *IEEE Std 1789-2015*, pp. 1–80, June 2015.