



Exploring the feasibility of short-range VLC schemes in MIMO systems
Otsu Thresholding and Sliding Window Protocols

Alexandru Nicolae Lolea¹

Supervisor(s): Qing Wang¹, Amund Kiste¹

¹EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Alexandru Nicolae Lolea
Final project course: CSE3000 Research Project
Thesis committee: Qing Wang, Shun Zhuge, Bo Yang, Mark Neerinx

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

With radio communications bandwidth becoming increasingly scarcer and expensive, researchers have turned towards the light medium, namely the field of Visible Light Communication (VLC). Although the field of Visible Light Communication (VLC) was pioneered in the late 1800's, it faced criticism from the scientists of that age, radio communications being preferred instead. VLC has regained its opportunity to shine by complementing existing radio communication methods. This research paper is focused on exploring different short-range multiple-input multiple-output (MIMO) screen-to-camera VLC schemes operating solely on the red optical channel. The transmitting screen is a 4x6 LED grid on a prototype board, while the receiver is an off-the-shelf smartphone back camera. The chosen modulation technique is On-Off Keying (OOK) with Manchester Encoding (ME), while demodulation is performed using three different strategies, with the first two using Otsu thresholding and the last using sliding window. Our experiments showed that, while the modulation schemes present a transmission rate of 6 symbols per LED every frame, which can amount to 144 symbols per frame, and a Bit Error Rate (BER) of less than 10^{-1} , the limited resolution and frame rate hardly allow the presence of important data frame header fields such as the Sequence Number.

1 Introduction

"I have heard articulate speech by sunlight! I have heard a ray of the sun laugh and cough and sing!". This quote belongs to the man who first developed the photophone [19]. While the telephone converts the speaker's voice to radio waves, the photophone converts the same voice into a beam of visible light. It is this invention that made the world remember him as the main driving force behind the field of Visible Light Communication (VLC). His name was Alexander Graham Bell.

Bell's design and use of visible light were put aside, both because of social resistance and the need for a design that is both easier to scale and more useful for the military [13].

However, as we all know, technology evolution is a synonym to larger amounts of data being transferred between communicating parties. This naturally results in bandwidth becoming scarcer and more expensive, which means that engineers have to look at other methods of transmitting data. One of these methods is by exploiting VLC.

In the field of VLC there are multiple different communication pairs, such as LED-to-LED communication [10], LED-to-camera where an off-the-shelf camera is used to detect the information, replacing the photo diode that was previously used [21], and even screen-to-camera communication, where a smartphone screen or monitor is used as the transmitter instead of a dedicated LED [23].

Throughout this paper, we concern ourselves with screen-to-camera communication. On one side of state of the art

research in screen to camera communication you have many different definitions for what is considered a 'screen'. For example, a screen can be anything from a smartphone screen [14], [2], [22], to the backlight of a mini-LED LCD [24], and even the taillight of a car for Vehicle-to-Vehicle communication (V2V) [12]. All of these papers use screens that are made up of at least 100 modulated transmitters and experimental frameworks that bring about unnecessary overhead for low-level prototyping. On the other side, you have research in VLC performed on prototypes, such as [9]. However, in this experiment, the team used one LED.

This outlines the lack of an "in-between" option. It seems that if you want to research the field of VLC, you need to either choose to prototype on a single LED, or a screen comprised of hundreds. One downside of this option sparsity is resource waste in scenarios where the desired data transmission rate can be achieved with a smaller number of LEDs, but a conventional "screen" is used.

This is precisely the research gap that this project will attempt to fill. Our aim is to evaluate the OOK + ME modulation method, together with the localization pipeline described in Section 2, and three demodulation techniques on a prototype board containing a 4x6 LED grid. This is supposed to prove that intermediate LED count in a screen to camera communication scheme ensures efficient resource usage, as well as a reliable data transfer rate.

The success of this communication scheme development and evaluation brings about a new possibility for research teams that are currently limited by the amount of financial resources, providing them a simple and cost-efficient method of creating prototypes for different projects involving VLC.

2 Preliminaries

2.1 Modulated Signals

Modulated signals are at the center of wireless communication. These signals are comprised of two main components:

1. Carrier wave
2. Modulating wave

Let's consider a scenario with two parties, *Alice* and *Bob* that wish to communicate. For example, *Alice* wants to send *Bob* the following sequence of bits $message = 01$. Lacking a guided medium (e.g. wire), the two decide to employ communication via radio signals. In our current scenario, the radio wave is the carrier wave, while the digital signal represented by $message$ is the modulating wave. The carrier wave is the "how" data transfer occurs, while the modulating wave is "what". In order to add information to the carrier wave, *Alice* must choose which property of the radio wave to modify according to the desired $message$. These properties include the *amplitude, phase or frequency* of the wave. One example of a popular technology employing signal modulation are the Wi-Fi protocols 802.11g/b, which act on the first two of the aforementioned wave properties [18].

2.2 Light Modulation Techniques

Within the broader framework of signal processing, the specific domain of VLC presents itself with the following four main domains of light modulation, namely:

1. Pulse-based
2. Continuous time
3. Color
4. Frequency

Throughout the rest of the research paper, the focus will be mainly on a pulse-based modulation technique, namely OOK. In order to modulate a signal using OOK, you must define what are the amplitudes for both the "On" and "Off" Key. For example, an amplitude of 0 represents the light being turned off completely, which can be associated with either a logical 1 or 0, depending on the communication scheme. A visual representation of this modulation scheme can be observed in Figure 1.

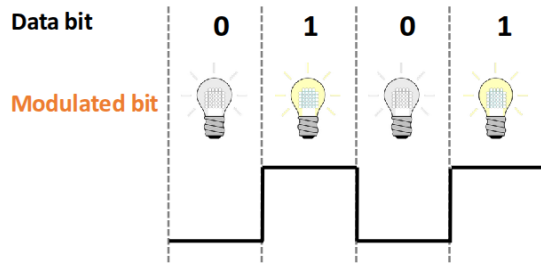


Figure 1: On-Off Keying. Source: [16]

Despite its simplicity, OOK is vulnerable to the flickering effect, even if we use the minimum modulation frequency invisible to the human eye, namely $200Hz$ according to [1]. If the modulation frequency is not above. In the context of screen-to-camera communication, this is one of the main factors we want to avoid while choosing a modulation scheme, since it can have negative effects on human health as well as affecting the display capabilities of the screen [15].

In order to solve this, a technique called Manchester Encoding is employed. ME avoids the flicker effect by encoding each logical bit into a pair of opposite symbols. A visual representation can be observed in Figure 2. The only case that we need to take into consideration is when a logical 1 is followed by a logical 0, which would cause two consecutive OFF symbols to be captured by the signal receiver. Thus, we must ensure that the frequency at which we are able to switch between symbols is twice the flicker-free frequency, which would amount to at least $400Hz$.

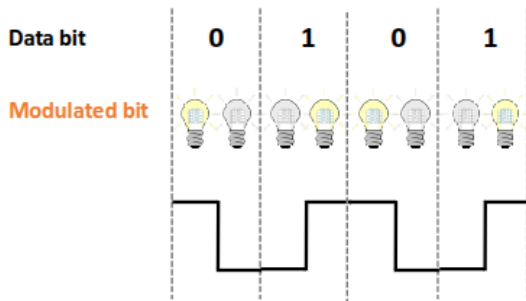


Figure 2: On-Off Keying with Manchester Encoding. Source: [16]

3 Our Contribution

The research question that we will attempt to answer can be expressed as: "What is a feasible method to implement a communication scheme for a custom LED-screen platform, focusing on the red-to-red optical link?". For the sake of clarity, we decided to split the aforementioned research questions into several sub-questions:

1. *Q1*: How does OOK+ME perform when demodulated using the baseline demodulation method?
2. *Q2*: How does OOK+ME perform when demodulated using the guarded Otsu thresholding demodulation method?
3. *Q3*: How does OOK+ME perform when demodulated using the sliding window demodulation method?

In the following section, under subsection 4.4, the reader can find the evaluation metrics used to measure the performance of each method.

4 Methodology

This section will describe the research framework we have employed. We will begin by explaining the setup used for completing the experiments, namely the demodulator smartphone application and the physical components that ensure that hand-held jitter while holding the camera or prototype board is not interfering with the experiments. Next, we will present the chosen experiments and conclude with a formal problem description, through which our results will be interpreted.

4.1 Demodulator Application

The demodulator was implemented on a ZTE Axon 30 5G. The user interface of the smartphone application was developed in Kotlin. Regarding the image processing engine, there are two versions written, both of which can be found in the appendix. The two versions were written in:

1. Kotlin: this was the first version of the engine. It was mainly used in order to achieve a working end-to-end scenario. The method through which this version demodulates the received light signal is by first calculating for each row, an average of the red channel pixel values. Then, based on how many rows of the screen should correspond to a whole data bit (let's call it n), an average of every group i of n adjacent values avg_i was computed. Finally, with respect to a threshold $\gamma = 0.5$, the engine made the decision of whether or not the bit is a logical 1 (for $avg_i \geq \gamma$) or a 0 otherwise.
2. C++: this was the final version of the engine. We used C++ because of its OpenCV library meant for more complex computer vision tasks. One of the features we can implement using this library is the LED localization pipeline which can be visualized in Figure 7. This version of the image processing engine has three methods of demodulation, referred throughout this work as 'guarded', 'unguarded', and 'sliding window'. The 'guarded' and 'unguarded' methods both use Otsu Thresholding on each individual LED in order to find

a ‘local‘ threshold. The guarded method takes into account the fact that symbol borders are usually noisy, so it chooses a percentage of ‘guard‘. Afterwards, that percentage of the total number of pixel columns for a symbol gets removed from the left and right sides of the symbol columns currently analyzed, thus removing noise. The final method is a sliding window algorithm inspired from [3].

Figures 3 and 4 show examples of the demodulator app’s user interface. The first image shows two screens, namely the Home Screen And Frame Upload Screen. The second image shows the two main steps a user takes when demodulating a captured frame, namely providing a reference image for the app to know where are the LED bounds, followed by providing a frame containing the modulating wave.

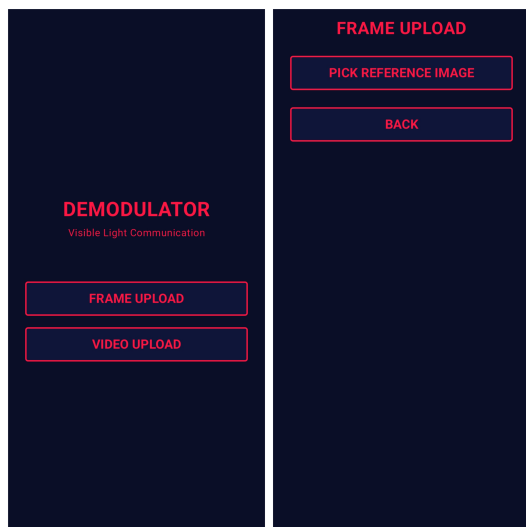


Figure 3: Demodulator Home Screen and Frame Upload Screen

4.2 Camera Settings

In order to capture videos using the smartphone, we have chosen the OpenCamera Android application [6]. This was decided due to the application’s high flexibility in terms of camera settings. The most important settings were the following:

1. Exposure: Set to $\frac{1}{5000}s$, in order to make sure that even in an environment with ambient lighting, only the red light of the LED was captured by the video
2. Video Resolution: 1920x1080. The higher the resolution, the more accurate the demodulator will be. The general rule of thumb is that regardless of what resolution you pick, the demodulator should always be aware of the choice made, lest you lose the available data provided by a supposedly higher resolution or give the engine ‘phantom confidence’ in a specific decision for a smaller resolution
3. Framerate: 30 frames per second. As mentioned in the introduction, the purpose of this research is to see what can be achieved with limited resources.

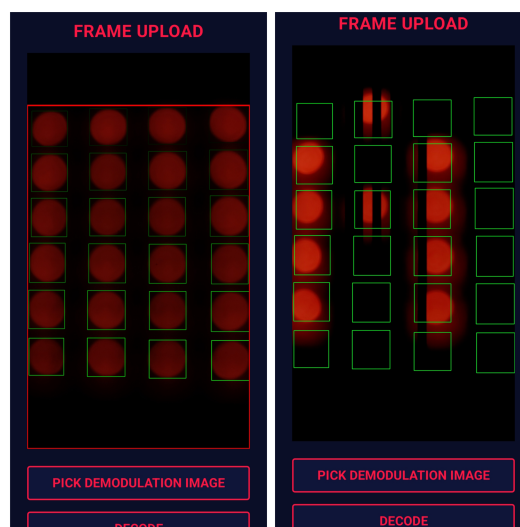


Figure 4: Reference Frame and Demodulation Frame

4. Video Format: MPEG4 H264

5. Video Bitrate: 200Mbps. We decided to pick the best available bitrate, due to the fact that preserving frame details is very important when it comes to demodulating the received signals

The frames have been extracted from the video using ffmpeg [5] in the beginning of the project. While extracting frames from the video we picked the PNG format due to it being lossless.

In order to ensure that the LEDs are positioned in the exact same position with respect to the picture frame, we used a simple tripod to hold the phone, thus reducing the camera shake. At the same, a frame has been designed in order to hold the board in a stable position. It is important to mention that due to the exposure being set at a value as small as $1/5000$ of a second, ambient light does not have an impact throughout the experiment.

4.3 LED dimming

As mentioned in the Introduction, this research is oriented towards efficient resource usage. Thus, we wanted to test how do different LED brightness values affect the error rate. The chosen PWM values for the LED grid are: 5, 10, 15, 20, 30, and 50. Based on our observations, anything above a PWM of 50 will bring discomfort to the human eye, which means that the display capabilities of the LED grid are affected. Thus, we have decided to place our upper threshold at 50 when screen brightness is concerned.

4.4 Evaluation Metrics

The two main evaluation metrics used are the following:

1. Data Transfer Rate (DTR)
2. Bit Error Rate (BER)

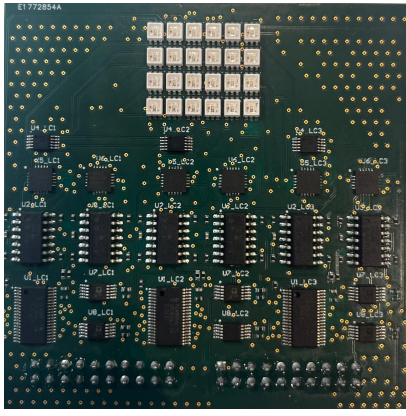


Figure 5: The prototype board. Creator: Amund Kiste, Technological University of Delft

4.5 Physical Setup

In Figure 5 the reader can see the prototype board used throughout the experiments. This prototype board is connected to an Arduino R1 Giga. Communication is established through the I^2C protocol. The board is divided into three 2×4 cells, each of them independently controlled by an I^2C driver, which explains the need for 3 different SDA/SCL connections, which the Arduino board is capable of providing. However, the SDA2/SCL2 pins, unlike pins 1 and 3, don't have a built-in pull-up resistor. In Figure 6, while observing the complete physical side of the experimental framework, the user can observe the breadboard that we used to build such a pull-up resistor.

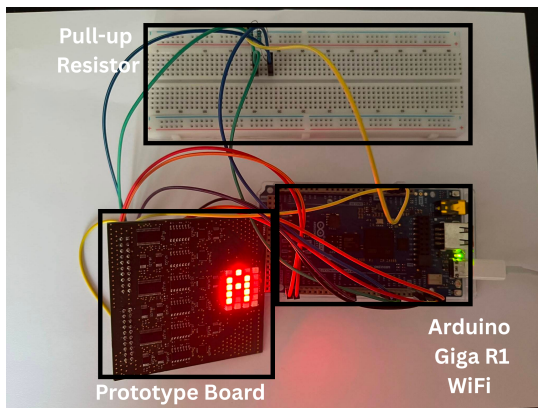


Figure 6: The complete physical setup

4.6 LED Localization Methods

In order to localize the LED grid and subsequently the individual LED bounds, we implemented the pipeline described in [4], which inspired the visual representation of this pipeline in Figure 7

In step (1), we simply send the bitmap of a reference frame to the C++ image processing engine of the native demodulator app. By reference frame we mean a frame in which all

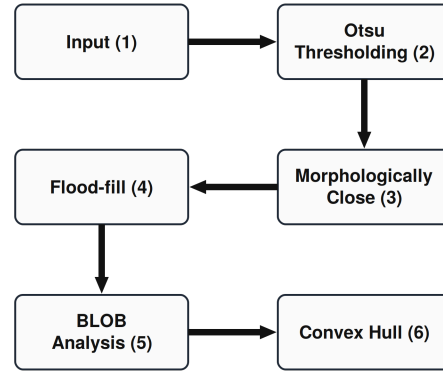


Figure 7: Panel Detection Pipeline

LEDs are turned on. This will allow us to compute the coordinates of all 24 boxes that each contain an LED. Before further processing, the input bitmap must be converted from the RGBA format to grayscale using the `cv::cvtColor()` method of OpenCV with the `code` parameter set to `cv::COLOR_RGBA2GRAY`.

Step (2) uses Otsu thresholding [11] in order to separate the pixels into two classes: foreground and background. In our case, the background pixels have value 0, while the foreground ones have value 255. For this step, the `cv::threshold()` method was used, with the `type` parameter set to `cv::THRESH_BINARY | cv::THRESH_OTSU`. At this point, the LEDs have been placed in the foreground. Since there is considerable distance between them, as can be seen in Figure 8, we can execute step (5) in order to obtain the bounding box coordinates.

Step (3) uses the previously binarized bitmap in order to unite foreground elements that are morphologically close. This process is executed by creating a structuring element using OpenCV's `cv::getStructuringElement()` and applying a closing operation in order to make sure that the LEDs are connected.

Step (4) then takes the connected components and fills the gaps between them, after which step (5) performs the BLOB analysis using `cv::connectedComponentsWithStats()`. After this operation, there should be a single BLOB detected, namely the panel itself, which brings us to step (6) where we can extract the LED grid bounding box coordinates.

4.7 Experiments

Our first experiment was based on plain OOK on a single modulated LED transmitting a logical 0 and 1 interleaved.

After obtaining the set of results, we decided to begin using the whole LED screen and split the following experiments into two subclasses, each corresponding to one of the evaluated metrics:

1. Distance experiments, corresponding to DTR
2. Brightness and demodulation experiments, corresponding to BER

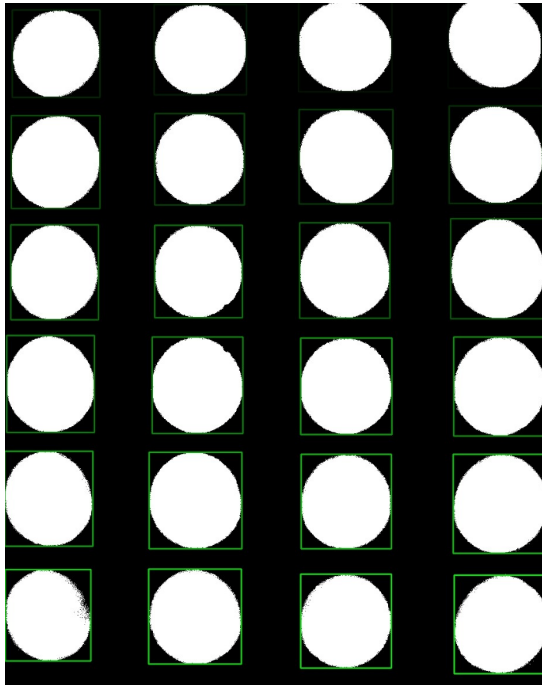


Figure 8: Binarized Bitmap

The plaintext transmitted throughout the experiments is 01010000. In order to ensure our simulations reflect real-world communication scenarios, we prepended the Manchester encoded plaintext data frames with a ‘Special Frame Delimiter (SFD)’ and encoded ‘Length’ field, as suggested in [17]. The SFD field has a length of $1B$, while the Length field has $6b$. The maximum size for a payload is $4b$, which translates to two logical bits in the OOK + ME modulation scheme.

4.8 Problem Formulation

For this type of experimental work, the methodology is centered around an empirical evaluation of trade-offs rather than the development of a new theoretical communication scheme. The focus is on analyzing how different design choices affect the error rate, information transfer rate and communication link strength under realistic deployment constraints.

The problem is formalized as a multi-objective evaluation problem. Instead of optimizing a single objective, the study investigates how different modulation techniques and LED localization techniques operate together.

More precisely, the following objectives are considered:

- maximizing symbol rate,
- minimizing error rate,
- minimizing the resources used by the smartphone demodulator
- minimizing the resources used by the prototype board

Rather than deriving a closed-form solution, the framework evaluates these trade-offs empirically through controlled experiments. The methodology is therefore based on systematic comparison of clearly defined configurations.

5 Results

We begin this section by presenting the results of our first experiment, focused on a single LED, modulated using OOK, transmitting a digital 0, followed by a digital 1, with this message being repeated throughout the entirety of the transmission period. All code related to the LED grid modulation can be found at [8].

5.1 Bit Error Rate

In Figure 9 we can see an example of a frame taken of a single LED toggling state at 1kHz. We can clearly observe the rolling shutter effect [20]. This was observed on a camera with a resolution of 1920×1080 and a framerate of 30fps. Empirically, we arrived at the conclusion that somewhere between 550 and 570 rows pass the actual LED. This in turn means that every solid band of red or black is represented by around 34 rows, which means that we will have a data rate of about 16 bits/frame.

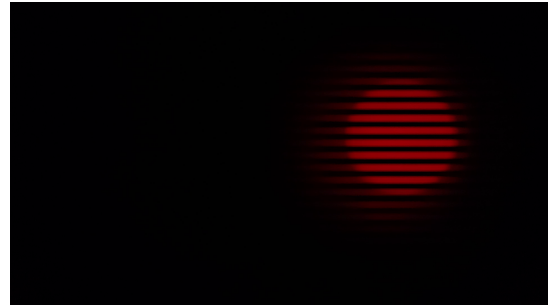


Figure 9: OOK generated frame

At the moment of running this first experiment, the image processing engine was written in Kotlin. The exact configurations of the Kotlin-based demodulator can be found in the *DecoderConfig.kt* file in [7]. At the moment of writing this, with a threshold of 0.3 for the average red value of the pixel, we achieved an bit error rate of 0.5.

The results of the subsequent experiments related to the bit error rate can be observed in Figures 10 to 12. We can notice that the guarded and unguarded methods are both yielding similar results, showing greater data reliability at PWM values 15 and 50. The aforementioned demodulation methods also present themselves with a higher standard deviation than the sliding window method.

5.2 Data Transmission Rate

By carefully positioning the LED screen as close as possible to the camera (2 cm), we managed to cover 1043 of the 1080 pixel columns defined by the resolution. The gaps between the four LED columns are approximately $81px$ each, which leads to a total modulation area $area = 1043 - 243 = 800px$. Dividing by the number of LED columns of the screen 4, we arrive at $200px$ per LED.

With a frame rate of $30fps$, the smartphone camera can capture a frame in $33ms$. Factoring in the resolution, we arrive at a ‘row rate’ of $30\mu s$. Using a modulation frequency

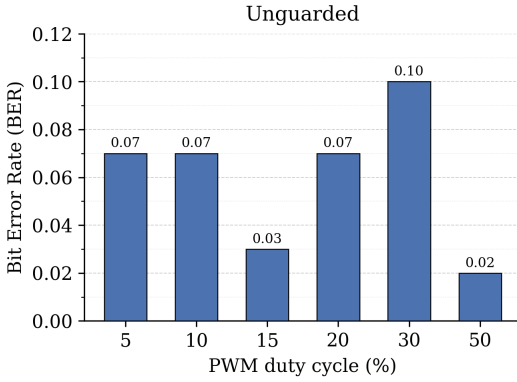


Figure 10: Average BER obtained with the unguarded demodulation method over 100 frames

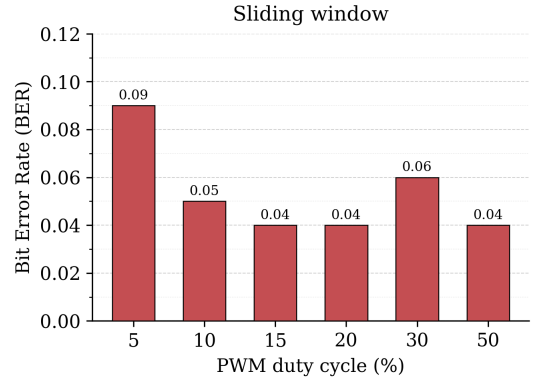


Figure 12: Average BER obtained with the sliding window demodulation method over 100 frames

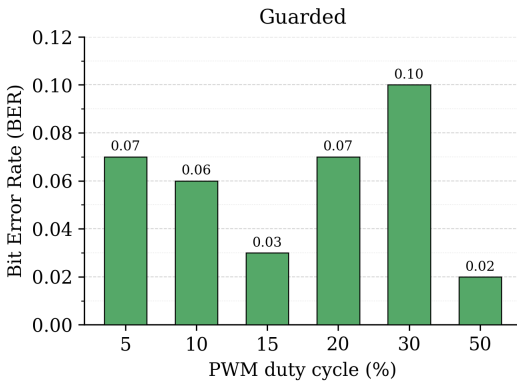


Figure 11: Average BER obtained with the guarded demodulation method over 100 frames

of 500Hz , we arrive at 2ms to transmit two alternate symbols. Thus, we switch the symbol every $1\text{ms} = 1000\mu\text{s}$. This results in every symbol requiring $\frac{1000}{30} \approx 33px$ rows. With the modulation area of an LED being defined previously at $200px$, we can therefore achieve a transmission rate of $\frac{200}{33} \approx 6$ bits per modulated LED.

6 Responsible Research

6.1 Generative A.I.

Generative A.I. platforms, such as Anthropic’s Opus 4.8 and Gemini’s Pro 3.1, have been used in the following ways:

1. Writing assistance: primary usage, mainly in contexts where we needed to rephrase a paragraph or certain section, as well as correcting any potential grammatical or spelling errors
2. Code generation: under specific guidance, the agents were employed to build specific parts of the code base, in order to save time.
3. Literature search: mainly used in order to explore the field of VLC under in a more controlled manner, by describing a specific number of ‘next steps’. Used mainly in the first weeks of starting this work.

Prompt examples can be found in the appendix.

6.2 Ethical considerations

Research into the field of Visible Light Communication directly transfers into researching methods of alternative methods of communicating some of the information we are currently using radio waves for. Progress in this subject will result in less radio spectrum congestion. Some of the benefits include less lag and higher data rates, which are critical in a society that is increasingly more connected every day. Besides this, there are important considerations to be made with regards to health, which can be affected by light flickers, as mentioned in [15], which further emphasized the need for the development of flicker-free VLC communication schemes.

7 Discussion

An analysis of the results shows that these methods are suitable for VLC screen-to-camera communication schemes implemented using off-the-shelf cameras, due to an average BER over 100 frames of less than 10^{-1} . However, we must take into consideration the fact that the symbol rate is limited to approximately 6 bits per LED, for each frame. This leads to a substantial lack of space for important header fields such as the Sequence Number, which sits at the core of parallel communication. Thus, while the three communication schemes present themselves as promising, they should be used in setups where both the modulation frequency and frame rate achieve much higher values than the ones used throughout our experiments.

7.1 Limitations

The first notable limitation of this research project is the need to have the board aligned with the camera, in order for the Otsu thresholding to work. The larger the angle of a supposed rotation, the bigger the bounding box of an LED will be. At the same time, if the rotation angle is large enough, the modulated columns will be interpreted as rows by the image processing engine.

A second limitation is caused by having all four columns of the LED matrix transmit the exact same light signal. In order to create a MIMO system, each transmitted data frame

would need a ‘Sequence Number (SN)’ field prior to the payload. This would further increase our data frames to 22 bits, considering the four data frames we are transmitting, which increases the system’s infeasibility.

The final important limitation is related to the synchronization between the camera and the modulated screen. This research paper doesn’t take into account the moment in the LED modulation at which video recording started. In order to detect the bit error rate, we keep a *truth* message on the receiver side. Upon receiving a string of l bits *message*, the demodulator loops through the aforementioned *truth* and seeks a sequence of l bits in *truth* that minimizes the Hamming Distance between it and *message*. This method is primarily susceptible to finding wrong minimal BERs in the presence of long runs of the same symbol. Such a run is represented by our SFD comprised of eight consecutive ON symbols.

8 Conclusion

This work introduced three short-range screen-to-camera VLC communication methods, relying on the red optical link, together with OOK modulation and Manchester Encoding, namely guarded and unguarded demodulation using a per-LED computed Otsu Threshold, as well as an adaptive sliding window protocol. Our analysis proved that the BER computed after using the aforementioned demodulation techniques lies just below 10^{-1} , at a symbol rate of 144 symbols per frame. However, our findings also showed that these methods are not fit for MIMO communication, since the limited frame rate and resolution don’t allow for important frame header fields such as the Sequence Number, as well as the Length and Special Frame Delimiter.

9 Future Work

Throughout this research paper we introduced plenty of components that can be explored in greater detail, from modulation techniques to error detection algorithms. However, we recommend prioritizing the demodulator application first, especially adding the possibility of taking a picture from within the smartphone application. This presents itself with one main advantage, which is gaining direct access to the camera, allowing a greater control over properties such as focus, frame rate, ISO, and exposure.

Following this, a second area of concern is represented by the class of error correction algorithm. We mainly focused on the BER throughout this work in order to prove the robustness of the communication channel. However, it would add plenty of value to the communication scheme if the original transmitted signal could be recovered.

A LLM Prompt examples

A.1 Literature Search

- “are there any research papers documenting a low-budget VLC setup? Screen to camera?”
- “give me 3 other options (*here referring to modulation techniques*) and also give me the link to a research paper implementing it. We can assume that there will be close to no movement of the board or phone”

A.2 Writing

- “‘Something lyies down.’ Is it gramatically correct?”
- “How can I rephrase ‘In order to bring our simulations closer to reality, we used the following methods...’”

A.3 Software Development

- “Okay, I want the following: a function that takes a decoded string `std::string`. It also takes the truth string. Now, this will use the min Hamming distance. So take all offsets from 0 to `length_of_truth - 1` and the result is the min Hamming distance”
- “I want the following, a JNI function that takes in a jobject Bitmap and a jobjectArray leds. I want to draw the LED bounds on the bitmap. It doesn’t need to return anything. Same architecture (*Here referring to the architecture I designed for another pair of functions*). A C++ function is doing the drawing, while the JNI is simply converting the leds array into `std::vector<cv::Rect<u>`”

References

- [1] IEEE recommended practices for modulating current in high-brightness LEDs for mitigating health risks to viewers, 2015.
- [2] Amanda Rühlemann Barreira, Hypolito José Kalinowski, Gonçalo Figueiredo, and Paulo Sérgio De Brito André. Screen-to-camera visible light communication on smartphones using csk modulation and cell-based encoding. pages 763 – 767, 2025. Cited by: 0.
- [3] Derek Bradley and Gerhard Roth. Adaptive thresholding using the integral image. *J. Graphics Tools*, 12:13–21, 01 2007.
- [4] Trong-Hop Do and Myungsik Yoo. A simple led panel dection algoritum for optical camera communication systems. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 747–749, 2019.
- [5] FFmpeg Developers. FFmpeg: A complete, cross-platform solution to record, convert and stream audio and video, 2026.
- [6] Mark Harman. Open Camera. Android application, Google Play Store, 2025. Last updated 18 August 2025.
- [7] Alexandru Lolea. demodulator: Vlc demodulator for the tudelft research project. <https://github.com/lolea-alexandru/demodulator>, 2026. [Online; accessed 9-June-2026].
- [8] Alexandru Lolea. Vlc on a budget. <https://gitlab.com/lolea-alexandru/vlc-on-a-budget>, 2026. Accessed: 2026-06-21.
- [9] Alina E. Marcu, Robert A. Dobre, and Marian Vlădescu. Flicker free visible light communication using low frame rate camera. *2018 International Symposium on Fundamentals of Electrical Engineering (IS-FEE)*, November 2018.

- [10] Muhammad Sarmad Mir, Behnaz Majleseini, Borja Genoves Guzman, Julio Rufo, and Domenico Giustiniano. LED-to-LED based VLC systems: Developments and open problems. In *Proceedings of the 2nd International Workshop on Internet of Lights (IoL)*, pages 1–6. ACM, 2021. Discusses challenges and open problems when using LEDs as receivers in Layer 1 VLC.
- [11] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.
- [12] Michael Plattner and Gerald Ostermayer. A camera-based vehicular visible light communication system using modulated taillights in public road scenarios. *Vehicular Communications*, 43, 2023.
- [13] RF Safe. From Bell’s Photophone to the Light Age: How wireless took a wrong turn — and How we correct it. <https://www.rfsafe.com/articles/cell-phone-radiation/from-bells-photophone-to-the-light-age-how-wireless-took-a-wrong-turn-and-how-we-correct-it>, 2024. Discusses the historical trajectory of wireless technology starting from Bell’s photophone, advocating for a shift toward light-based communication (such as Li-Fi) over traditional radio frequencies due to health and safety concerns. [Online; accessed 30-April-2026].
- [14] Vijen Sewpersadh, Narushan Pillay, and Hongjun Xu. Streaming qr codes-a survey. *IEEE Access*, 12:134144 – 134166, 2024. Cited by: 8; All Open Access; Gold Open Access; Green Open Access.
- [15] Baozhu Wang, Wenjing Sun, Jian Chen, and Changyuan Yu. Frequency domain anti-flickering in visible light communication system. *Optics Communications*, 516:128203, 2022.
- [16] Qing Wang. Lecture 2: Physical layer: Pulse-based modulation, September 2025. Unpublished presentation slides, Delft University of Technology, Course code:CESE4110.
- [17] Qing Wang, Domenico Giustiniano, and Daniele Puccinelli. An open-source research platform for embedded visible light networking. *arXiv*, 2014.
- [18] Wikipedia contributors. IEEE 802.11 — Wikipedia, the free encyclopedia, 2026. [Online; accessed 20-June-2026].
- [19] Wikipedia contributors. Photophone. <https://en.wikipedia.org/wiki/Photophone>, 2026. Discusses the history and operation of the photophone, an early wireless telecommunications device invented by Alexander Graham Bell that transmits speech on a beam of light. [Online; accessed 30-April-2026].
- [20] Wikipedia contributors. Rolling shutter — Wikipedia, the free encyclopedia, 2026. [Online; accessed 9-June-2026].
- [21] Yanbing Yang and Jun Luo. Boosting the throughput of LED-camera VLC via composite light emission. In *IEEE International Conference on Communications (ICC)*, pages 1–9. IEEE, 2018. Addresses Layer 2 Optical Camera Communication by exploring composite light emission to improve rolling-shutter throughput.
- [22] Hanting Ye, Jie Xiong, and Qing Wang. When vlc meets under-screen camera. pages 343 – 355, 2023. Cited by: 9; All Open Access; Gold Open Access; Green Open Access.
- [23] Vaigai Nayaki Yokar, Hoa Le Minh, Zabih Ghassemloooy, and Wai Lok Woo. Channel characterization in screen-to-camera based optical camera communication. *arXiv preprint arXiv:2506.23005*, 2025. Provides recent channel characterization for Layer 3 Screen-to-Camera (S2SVLC) links.
- [24] Zhiqing Zhao, Yuqing Qiu, Guowei Zou, Yi Liu, Jiefeng Weng, Bo-Ru Yang, and Zong Qin. High-capacity mimo visible light communication integrated into mini-led lcds. *Optics Express*, 32(8):14876 – 14891, 2024. Cited by: 6; All Open Access; Gold Open Access.