Delft University of Technology
Master of Science Thesis in Computer & Embedded Systems Engineering

Optimizing Simultaneous LoS and NLoS Visible Light Communication with Event Cameras

Robert van Dijk

TU Delft Delft University of Technology

Optimizing Simultaneous LoS and NLoS Visible Light Communication with Event Cameras

Master of Science Thesis in Computer & Embedded Systems Engineering

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

Robert van Dijk

9th April 2026

Author
Robert van Dijk

Title
Optimizing Simultaneous LoS and NLoS Visible Light Communication with Event Cameras

MSc Presentation Date
17th April 2026

Graduation Committee

Dr. Marco Zuñiga Zamalloa	Delft University of Technology
Dr. Qing Wang	Delft University of Technology

Abstract

The radio-frequency spectrum is increasingly congested and costly to license, which motivates the use of complementary wireless links in other parts of the electromagnetic spectrum. Visible Light Communication (VLC) transmits data by modulating visible light. Among the receiver types used in this field, event cameras are attracting increasing attention due to their significantly higher rates than conventional cameras. Recent work has studied event-camera VLC in either Line-of-Sight (LoS) or Non-Line-of-Sight (NLoS) settings, but has not combined both links in a single transmitter. In applications such as infrastructure-to-vehicle communication, receivers may operate under both LoS and NLoS conditions, making it desirable to support both link types simultaneously.

This thesis presents a single LED-matrix transmitter that supports both a high-data-rate (high-fidelity) LoS stream and a low-data-rate (low-fidelity) NLoS stream simultaneously. To this end, we introduce Dual On-Off Keying (DOOK), a multi-fidelity modulation scheme that encodes high-fidelity data in the *spatial* and *temporal* dimensions, while encoding low-fidelity data in the *temporal* dimension only. We further combine DOOK with state-of-the-art modulation schemes and design flicker-free variants. We evaluate the resulting trade-offs between throughput, Bit Error Rate, and flicker.

Using DOOK, we achieve 366 kbps on the LoS link and 2,9 kbps on the NLoS link with a Bit Error Rate (BER) below 10^{-3}. DOOK with Manchester encoding halves the throughput and produces the least flicker among the evaluated schemes. Compared with prior work, our NLoS throughput is 1,7× higher, while our LoS throughput is 1,8× higher per channel. More importantly, our system combines both links in a single transmitter.

Contents

Acronyms

API Application Programming Interface

BER Bit Error Rate

DMD Digital Micromirror Device

EM Expectation-Maximisation

IC Integrated Circuit

LED Light Emitting Diode

LoS Line-of-Sight

MCU Microcontroller Unit

MIMO Multiple-Input Multiple-Output

NLoS Non-Line-of-Sight

OOK On-Off Keying

PCB Printed Circuit Board

PWM Pulse Width Modulation

RF Radio-Frequency

ROI Region of Interest

SDK Software Development Kit

SFD Start Frame Delimiter

SISO Single-Input Single-Output

SoA State of the Art

VLC Visible Light Communication

Chapter 1

Introduction

The Radio-Frequency spectrum used by technologies such as Wi-Fi and 5G is becoming increasingly crowded [1]. Visible Light Communication (VLC) offers an alternative by transmitting data through a different, unregulated medium: the visible light spectrum. Imagine turning a flashlight on and off to send a message, like Morse code, but at rates of thousands of times per second. To the human eye, the light appears steady, while a sensor can decode the information embedded in the light beam. In VLC, the light source is modulated to encode a message [2]. This modulation can be generated using LEDs, screens, or other light sources. The modulated light is then detected by a photodiode or camera, after which the message can be decoded [3]. In this thesis, we focus on communication between an LED array and a camera. An example is shown in Figure 1.1, where an LED traffic sign transmits data to cars with cameras.

Rather than using a conventional camera, such as the type commonly found in smartphones, we use a specialised sensor known as an event camera. A conventional camera records frames that capture the intensity of every pixel. By contrast, an event camera generates events only when the intensity at a pixel changes [4]. Figure 1.2 illustrates this difference. Because an event camera

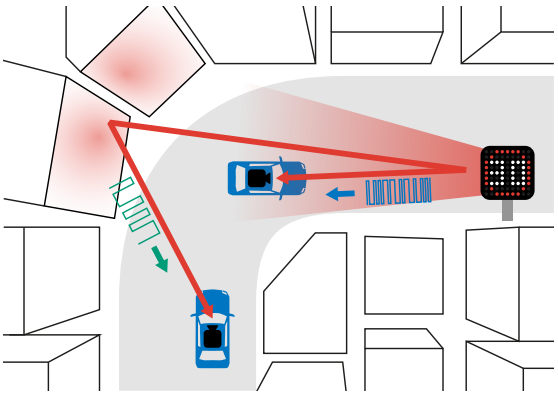
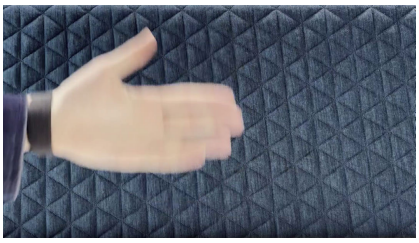
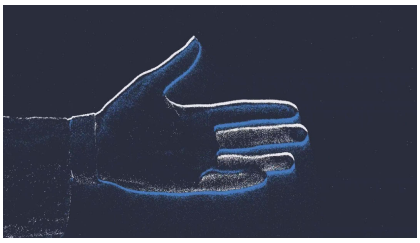


Figure 1.1: **Visible Light Communication using a Light Emitting Diode (LED) traffic sign as transmitter and cars equipped with event cameras as receivers.**



(a) **A conventional camera captures the intensity of each pixel.**



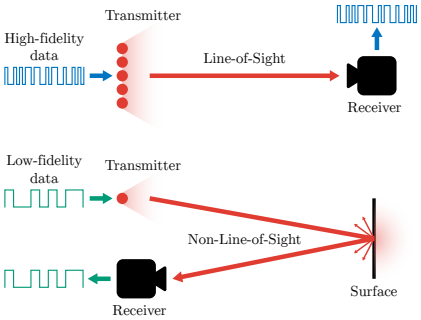
(b) **An event camera captures which pixels have changed in intensity.**

Figure 1.2: **The difference between a conventional camera and an event camera, illustrated by a waving hand.**

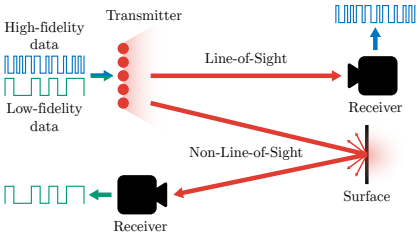
produces output only for pixels that change, it can operate at a much higher speed, with an equivalent rate of up to 1.000.000 fps [5, 6]. This characteristic makes event cameras particularly attractive for high-speed VLC channels.

Event cameras are already used in applications such as robotics and virtual reality [7]. When LED arrays such as traffic lights or LED road signs are combined with event cameras in self-driving vehicles, these light sources can support communication in addition to illumination, without disturbing users [8, 9].

VLC links can be divided into Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) links. In LoS communication, an unobstructed path exists between transmitter and receiver. In NLoS communication, the light is reflected by at least one surface before reaching the receiver [2, 10]. Several studies have investigated LoS VLC with event cameras [6, 9, 11–13], whereas only a smaller set has investigated NLoS VLC with event cameras [14, 15] (Figure 1.3a). To the best of our knowledge, however, no prior work has combined LoS and NLoS communication within a single transmitter design (Figure 1.3b).



(a) **Other studies focusing on LoS high-fidelity and NLoS low-fidelity LED array to event camera communication separately.**



(b) **Contribution of this thesis: combining high-fidelity LoS and low-fidelity NLoS communication in one transmitter.**

Figure 1.3: **Comparing the state of the art to the contribution of this thesis.**

A LoS link can achieve higher throughput than a NLoS link when multiple LEDs are used as transmitters, and each LED blinks independently. In the NLoS case, the emitted light is reflected and mixed before it reaches the camera. As a result, the receiver observes a blended signal rather than the contribution of each individual LED. When the camera observes the LEDs directly, by contrast, it can distinguish the light emitted by each LED separately [6, 15]. This gives rise to the central problem addressed in this thesis: Can we design a modulation scheme that supports both LoS and NLoS links while maximising throughput in both cases? More specifically, we seek to encode high-fidelity data, such as detailed traffic information or video, on the LoS link, while simultaneously encoding low-fidelity data, such as information about the current road or a text message, on the NLoS link.

This thesis addresses this challenge through the following contributions:

- We build a hardware platform in the form of an LED matrix that can update 256 LEDs at 20 kHz. The platform operates without the unwanted flicker artefacts found in commercial LED matrices. We describe this design in Chapter 3.
- We introduce Dual On-Off Keying (DOOK). DOOK encodes a high-fidelity message in both the temporal (time) and spatial (pixel (x, y)) dimensions, while simultaneously encoding a low-fidelity message in the temporal dimension only. It achieves this by controlling the average brightness of the LEDs in each spatial frame. We describe the modulation scheme in Chapter 4 and the decoding pipeline in Chapter 5.
- We implement both basic and state-of-the-art modulation schemes for the NLoS low-fidelity stream in Chapter 4. We also design and implement flicker-free variants. We evaluate these schemes in terms of throughput, error, and flicker. Although the State of the Art relies on a complex modulation scheme for the NLoS link, our experiments show that the basic modulation schemes, namely DOOK and DOOK with Manchester encoding, perform best. We present these results in Chapter 6.

Chapter 2

Background and Related Work

This chapter provides the background needed for the remainder of the thesis and situates our work within the existing literature. We begin in Section 2.1 by introducing the main concepts used by the system studied in this thesis, namely Visible Light Communication (VLC), Line-of-Sight (LoS)/Non-Line-of-Sight (NLoS) links, and event cameras. We then turn to prior research on event-camera VLC and multi-fidelity communication schemes in Section 2.2. Taken together, these sections establish the basis for the modulation and decoding approaches developed later in Chapters 4 and 5.

2.1 Background

This section introduces the technical concepts used throughout the thesis. We first outline the main principles of VLC, including transmitter and receiver types, modulation schemes, and the distinction between LoS and NLoS links. We then introduce event cameras and describe the event-based sensing model on which our receiver design relies. These concepts provide the framework for the system design presented in the following chapters.

2.1.1 Visible Light Communication (VLC)

VLC is an optical wireless technology that uses visible light (380–780 nm) to transmit data [16]. The visible spectrum is unregulated and free to use [16]. This makes VLC attractive as an addition to Radio-Frequency (RF) communication, where the spectrum is congested and costly to license [1, 2, 10].

In VLC systems, Light Emitting Diodes (LEDs) typically serve a dual role by providing both illumination and communication. This can reduce hardware cost and energy consumption. In addition, it allows existing lighting infrastructure to be reused [10].

At a high level, a VLC system consists of four parts: a transmitter, a modulation scheme, a propagation path, and a receiver. In this thesis, we use an LED array as the transmitter and an event camera as the receiver. The transmitter sends data by changing the emitted light over time, following some modulation

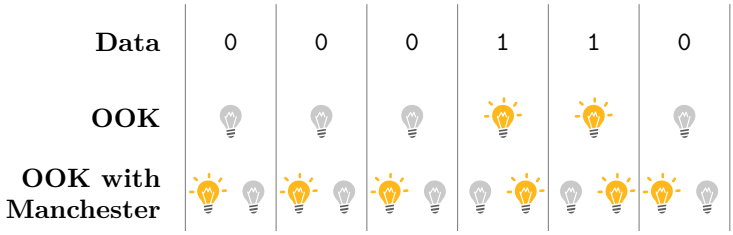


Figure 2.1: **An example of data encoded with On-Off Keying (OOK), and OOK with Manchester encoding. Manchester encoding is flicker-free.**

scheme. That light can reach the receiver directly or after reflecting from surrounding surfaces, which gives rise to Line-of-Sight and Non-Line-of-Sight links. In addition, a system may use a single channel or multiple channels in parallel, the latter being referred to as Multiple-Input Multiple-Output. The following sections explain these concepts in more detail.

Modulation schemes and flicker

A VLC transmitter sends data by modulating the intensity of the emitted light. The simplest modulation scheme is On-Off Keying (OOK). In OOK, a `0` bit is represented by the light being off, whereas a `1` bit is represented by the light being on [17]. Figure 2.1 shows an example.

Because the same LEDs often provide both communication and illumination, straightforward intensity modulation can introduce visible or invisible flicker [18]. For example, long runs of `0`s or `1`s may cause the light output to vary in a way that appears as blinking, even when the refresh rate is high. Moreover, even imperceptible flicker can have negative effects on users [19]. These effects include headaches, eyestrain, and, in extreme cases, epileptic seizures [20]. A practical VLC system must therefore reduce flicker while still sending data.

One common way to reduce flicker is to combine OOK with Manchester encoding, which balances the time for which the light is on and off [21]. Following the IEEE 802.3 convention, a `0` is encoded as a transition from on to off, while a `1` is encoded as a transition from off to on [22]. This is illustrated in Figure 2.1. Under this encoding, the light does not remain in the same state for more than two consecutive slots, regardless of the input data. As a result, the transmitted signal maintains a 50% duty cycle over time, thereby reducing flicker.

A modulation scheme operates on symbols. A symbol is a group of N time slots, denoted as t_{slot}, consisting of on and off states that together represent one or more bits. For example, OOK with Manchester encoding uses the patterns on-off and off-on as symbols for `0` and `1`, respectively. Since both symbols span two slots, the symbol duration is $2 \cdot t_{slot}$.

Receiver types

Different types of receivers can be used in VLC, including photodiodes and cameras [23, 24]. Photodiodes provide high temporal resolution (high sample rate) and therefore support high data rates. Cameras, by contrast, provide

spatial resolution, which helps separate signals and suppress background interference [24]. However, conventional cameras typically operate at a lower frame rate/refresh rate than photodiodes, resulting in lower data rates. In this thesis, we use an event camera, which combines the spatial resolution of cameras with much higher temporal resolution. We discuss this receiver type in Section 2.1.2.

Single-Input Single-Output and Multiple-Input Multiple-Output

Most basic VLC links are Single-Input Single-Output (SISO) systems, meaning they transmit a single data stream over a single transmitter-receiver pair. Communication between one LED and one photodiode is a typical example.

Multiple-Input Multiple-Output (MIMO) extends this principle by using multiple transmitters and multiple receiving elements [23]. In VLC, this may involve an LED matrix at the transmitter and the pixel array of an image sensor at the receiver. Such a system can transmit several independent data streams simultaneously or introduce redundancy to improve reliability [10]. In camera-based systems, the optical lens maps light from different LEDs onto different regions of the image plane. This spatial separation makes it possible to decode the contribution of each LED individually [23]. We use this MIMO principle throughout this thesis.

Line-of-Sight and Non-Line-of-Sight links

VLC links are commonly classified as Line-of-Sight (LoS) or Non-Line-of-Sight (NLoS) [2]. In a LoS link, the receiver has a direct and unobstructed path to the light source [10]. In a NLoS link, by contrast, the light reaches the receiver only after reflection from one or more surfaces [2]. In indoor environments, this may involve diffuse reflections from walls or other objects [10].

This distinction becomes especially important in multi-channel systems such as an LED array observed by a camera. In a LoS link, the receiver can observe the contribution of individual LEDs. In a NLoS link, or in an out-of-focus LoS link, light from neighbouring LEDs may spread and blend together before detection [23]. This leads to spatial interference between channels. Such interference is commonly modelled as a spatial filter, for example a two-dimensional Gaussian filter, that combines neighbouring channels mathematically [24, 25]. In this thesis, we establish both a LoS and a NLoS link using a single transmitter. This dual setting is central to the multi-fidelity communication problem studied in the remainder of the thesis.

2.1.2 Event Cameras

As discussed above, camera-based VLC offers an important advantage over photodiode-based VLC because it preserves spatial information. Multiple transmitters can be separated by grouping pixels, and background interference can be reduced by ignoring irrelevant image regions. However, conventional cameras are limited by their frame rate, which constrains the achievable throughput. This limitation motivates the use of event cameras.

Unlike a conventional camera, which records full image frames at fixed time intervals, an event camera reports only changes in brightness [4]. More specifically, a conventional camera measures the intensity of every pixel in each

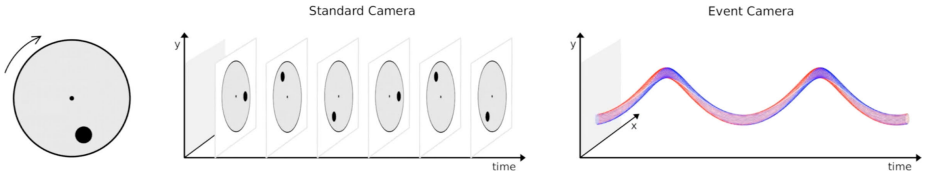


Figure 2.2: **A conventional camera captures frames of the spinning disk at a fixed rate, while an event camera generates a stream of events that captures only which pixels have changed [26].**

frame, whereas an event camera generates an event only when the intensity at a pixel changes. Figure 2.2 illustrates this difference. Because the sensor outputs changes rather than full frames, it can achieve an equivalent rate of up to 1.000.000 fps [5, 6].

Event cameras are also referred to as neuromorphic cameras or dynamic vision sensors. Their sensing principle is inspired by biological vision in the sense that pixels respond independently to changes in light intensity [4]. Each pixel operates asynchronously and generates events without waiting for a global frame clock [7]. These events are transmitted to a computer as a stream of address-events. An event is represented as (x_k, y_k, t_k, p_k), where x_k and y_k denote the pixel coordinates, t_k is the timestamp, and p_k is the polarity indicating whether the brightness has increased or decreased (binary) [7]. The sensor, therefore, does not measure absolute brightness directly. Instead, it responds to changes in brightness, and larger changes produce more events.

This sensing model gives event cameras several properties that are attractive for communication systems. These include high temporal resolution, high dynamic range, and low latency [7]. As a result, event cameras perform well in scenarios where conventional cameras struggle. Applications that leverage these properties include mobile robotics, augmented and virtual reality, gesture recognition, and Simultaneous Localisation and Mapping [7]. In the context of VLC, the same microsecond time resolution also supports the decoding of high-frequency optical signals. This opens the way to joint sensing and communication [15]. A further practical benefit is that the same receiver can support multiple functions, which may reduce overall system cost [6].

For VLC reception, event cameras occupy a useful middle ground between photodiodes and conventional cameras. Photodiodes offer excellent temporal resolution, but they lack spatial resolution. This makes it difficult to separate multiple light sources and to suppress ambient interference [27]. Conventional cameras, by contrast, provide spatial resolution but are limited by relatively low frame rates [11, 28]. Event cameras combine spatial resolution with microsecond temporal resolution. This combination makes them particularly well suited to VLC systems that use multiple LEDs at the transmitter.

2.2 Related Work

This section reviews the prior work that is most relevant to this thesis. We first discuss event-camera VLC in settings that are closest to LoS communication in Section 2.2.1. We then turn to NLoS VLC with event cameras in Section 2.2.2.

Finally, Section 2.2.3 reviews multi-fidelity modulation schemes. Together, these works provide the context for the design choices made later in Chapters 4 and 5.

2.2.1 Event cameras for VLC

While earlier work on VLC between LEDs and conventional cameras reports throughputs of only a few kbps [29, 30], the work by Shen, Chen and Tsai is the earliest work we are aware of that studies VLC with an event camera. They achieve a throughput of 16 kbps by designing a custom waveform [11]. A key point in their work is that the modulation scheme should minimise transitions, because event cameras have a limited event rate [11]. However, they use only a single-channel LED transmitter. As a result, they primarily use the event camera's spatial resolution to remove noise rather than to increase throughput across multiple spatial channels [11].

The work by Wang et al., called *Selene*, extends this idea by leveraging the event camera's spatial resolution to increase throughput using multiple channels. They use a Digital Micromirror Device (DMD) to project 1995 squares onto a wall, where each square acts as a separate VLC channel [6]. Although this setup is technically NLoS, it is closer to LoS systems in practice because the light is focused on the wall and the channels remain spatially separated rather than blurred together. They achieve a throughput of 1,61 Mbps [6]. This high throughput is achieved in part by their novel techniques: dual refresh rates and event-relative decoding. With dual refresh rates, squares near the edges of the frame refresh more slowly than squares near the centre, because the camera is less sensitive at the edges. With event-relative decoding, the system decodes symbols from the time between events rather than from the absolute timestamps of events [6].

This work is the closest prior work to our LoS link, as it also uses both the spatial and temporal dimensions to encode data. For that reason, we use it as a baseline for the LoS performance of our system. However, we do not use dual refresh rates or event-relative decoding in this thesis. These techniques assume that each spatial channel can be decoded independently. In our system, by contrast, all channels are combined to form a single message, which is necessary for our combined LoS/NLoS setup.

Another relevant development is the N-Pulse modulation scheme introduced by Aranda et al. Their goal is to enable computationally efficient demodulation for event-camera VLC. N-Pulse encodes two bits in one symbol by pulsing the light one, two, three, or four times within that symbol. Because each pulse generates events, demodulation mainly reduces to counting the number of events per symbol [12]. This is different from the argument made by Shen, Chen and Tsai, who state that modulation should minimise transitions in order to reduce the number of events [11]. As a result, N-Pulse reduces demodulation complexity, but it also reduces the achievable throughput.

2.2.2 Non-Line-of-Sight VLC with event cameras

Conventional cameras can be used for NLoS VLC by exploiting the rolling-shutter effect [31]. In these systems, the light is modulated faster than the camera frame rate and then reflected around the room. Because a rolling-shutter camera exposes one row at a time, this modulation appears as intensity

bands in the image, which can then be decoded. Many works in this area focus on reducing the error rate using novel algorithms [32, 33].

However, these approaches do not apply to event cameras. An event camera does not have a rolling shutter. Instead, its pixels operate asynchronously. Methods that rely on rolling-shutter patterns, therefore, cannot be used in our setting.

Only a small number of studies have examined NLoS VLC with event cameras. The work by Nishar, Paul and Ashok shows that event cameras can detect reflected optical signals. They modulate light from an LED transmitter at different frequencies and show that the reflected signal can be detected [14]. They also show that this detection becomes vulnerable to ambient-light interference when the ambient-light flicker frequency is close to the modulation frequency [14]. However, they do not transmit data in this work. As a result, they do not evaluate throughput or bit error rate.

Later, Nishar, Marefat and Ashok show that NLoS VLC with event cameras can be used for data transmission. They achieve a throughput of 1702,13 bps with a Bit Error Rate (BER) of 0,000402 [15]. For this, they use a variant of N-Pulse modulation. They also evaluate a scheme in which one pulse maps to `1` and no pulse maps to `0`, which is a variation of OOK. They state that the BER for OOK is too high to be usable for sending non-repeating data, namely above 10^{-3} [15]. Since this is, to the best of our knowledge, the only prior work that sends data over an NLoS link to an event camera, we use it as a baseline for the NLoS performance of our system. In this thesis, we also implement both N-Pulse and OOK for the NLoS link. Whereas Nishar, Marefat and Ashok report that OOK is not usable in their setting, our results show that OOK outperforms N-Pulse in both throughput and BER.

2.2.3 Multi-fidelity modulation schemes

Several works combine multiple data streams with different fidelities into one transmitter. A multi-fidelity data stream contains multiple streams with different bit rates. High-fidelity data has a high bit rate, while low-fidelity data has a low bit rate.

The work by Arai et al. combines two data streams into a single signal by using two carrier frequencies [34]. The work by Nishimoto et al. encodes low-fidelity data in large blocks on an LED matrix and layers high-fidelity data in smaller blocks on top using a transparency value [35]. Both works use an LED matrix as transmitter and a conventional camera as receiver.

These approaches are relevant because they demonstrate how multiple streams can be combined into a single optical transmitter. However, they rely on the receiving camera being able to distinguish between shades of grey. An event camera cannot observe shades of grey. It reports only whether a pixel has become brighter or dimmer. For that reason, these approaches cannot be applied directly to our problem. In addition, these systems are designed for a LoS link and a partially blurred LoS link, whereas our system is designed for a LoS link and a fully blurred NLoS link.

Taken together, these areas of work address important parts of our problem, but not the full combination considered in this thesis. Prior work on event-camera VLC shows that event cameras can support high-speed communication in settings close to LoS, and more recent work shows that they can also support lower speed NLoS communication. Related work on multi-fidelity modulation shows how multiple streams can be combined into a single transmitter, but only for conventional camera receivers. In this thesis, we build on prior MIMO event-camera VLC for the LoS link and on prior NLoS event-camera modulation schemes for the NLoS link. However, the literature does not provide a system that combines a high-fidelity LoS stream with a low-fidelity NLoS stream within a single event-camera-compatible modulation scheme.

Flicker is another point on which the existing literature remains limited. To the best of our knowledge, prior work neither designs flicker-free modulation schemes nor evaluates flicker alongside throughput and error rates. In this thesis, we address this gap by introducing flicker-free variants and by explicitly evaluating the trade-off between throughput, error, and flicker.

Chapter 3

Platform

As discussed in Chapter 2, no prior work combines LoS and NLoS VLC with event cameras in a single system. For this reason, we develop our own transmitter platform in Section 3.1. We first examine the requirements of our setup and explain why the transmitters used in prior work do not meet them. We then consider commercially available transmitters and show why they are not suitable for our application. Finally, we present the design of our custom transmitter, shown in Figure 3.1.

After introducing the transmitter, we describe the receiver used in our experiments in Section 3.2. In particular, we discuss the hardware platform and the provided API, which forms the basis of our decoding pipeline.

3.1 Transmitter

To transmit data in both the spatial and temporal dimensions, we require a matrix of transmitters. This transmitter must satisfy three main requirements.

- High refresh rate (20 kHz) to match the modulation speed and throughput reported in related work [15].
- A bright light source to ensure sufficient light for the NLoS link.
- No additional sources of modulation apart from our own. In particular, the matrix driver must not introduce unwanted modulation.

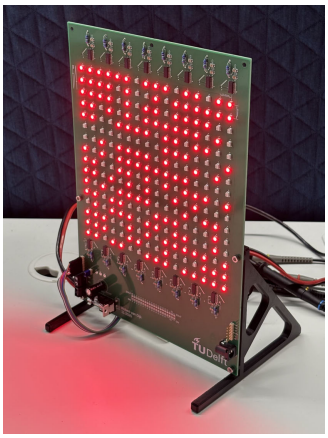


Figure 3.1: **Transmitter: High-Speed 16×16 LED matrix.**

3.1.1 State of the Art transmitters

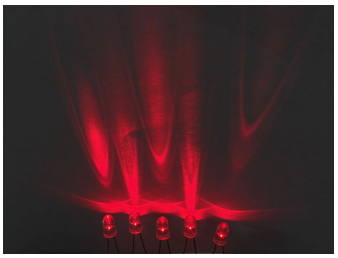
Given these requirements, the transmitters used in the State of the Art (SoA) are not suitable for our setup. This is for the following reasons:

- **Not fast enough**. The matrices used in earlier multi-fidelity systems, discussed in Section 2.2.3, are designed for use with conventional cameras [34–36]. As a result, their refresh rates are too low for event-camera reception.
- **Not bright enough**. The system proposed by Wang et al. uses a DMD to project light onto a wall. As discussed in Section 2.2.1, this first reflection is closest to our LoS link. In our setting, however, the NLoS link would require at least one further reflection. This would likely scatter the light too much, reducing signal quality.
- **No technical documentation**. The work by Soga et al. uses a transmitter that comes close to matching our requirements. Their system uses 96 LEDs with a refresh rate of 10 kHz. However, they do not provide sufficient technical detail about the transmitter design [9].

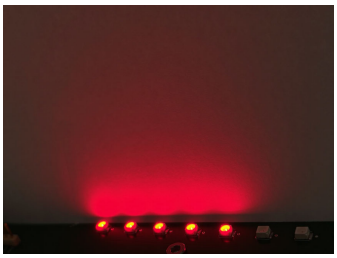
3.1.2 Off-the-shelf transmitters

Commercially available LED matrices are designed to display graphics to human observers. For this reason, they are not well-suited to our application. The main limitations are as follows:

- **Not fast enough**. LED matrices with 256 addressable LEDs update in 7,73 ms [37]. This corresponds to a refresh rate of 129 Hz, which is orders of magnitude below our target.
- **Extra modulation due to multiplexing**. Many off-the-shelf LED matrices use multiplexing to drive the display. In such systems, the matrix activates one row at a time. When this happens sufficiently quickly, the display appears steady to the human eye. However, an event camera is fast enough to detect this row-wise switching. This adds unwanted modulation on top of our own signal. It can therefore make the transmitted data harder to decode. In addition, these events consume part of the event-rate budget without carrying useful information.
- **Extra modulation due to dimming and colour control**. Commercial matrices often implement digital dimming and adjustable colour using Pulse Width Modulation (PWM). PWM controls brightness by switching the LEDs on and off with a varying duty cycle [38]. This introduces additional on–off transitions on top of the intended communication signal. As in the case of multiplexing, this makes decoding more difficult and uses part of the available event-rate budget.



(a) **LEDs with 8° viewing angles.**



(b) **LEDs with 120° viewing angles.**

Figure 3.2: **LEDs with different lenses.**

3.1.3 Design of the custom transmitter

To meet these requirements, we design and build a custom transmitter. The design aims to provide a high refresh rate, sufficient brightness, and full control over the emitted optical signal.

We implement the matrix using shift registers. A shift register allows us to send a long sequence of bits over a single data line and store one bit for each LED. While a new frame is shifted into the registers, the matrix continues to display the previous frame. Once the full frame has been loaded, we toggle the latch signal and update all outputs simultaneously. This ensures that the new frame appears simultaneously across the matrix. As a result, we avoid partial updates that would otherwise create unsynchronised events. Appendix B shows the full circuit diagram.

For the shift registers, we use the Texas Instruments TLC59282 [39]. This Integrated Circuit (IC) integrates a shift register and a latch into a single component. In addition, it includes circuitry to drive the LEDs directly, which simplifies the hardware design. We daisy-chain 16 of these chips to drive 256 LEDs.

We select red LEDs with a wide viewing angle of 120°. Although the colour itself does not affect the communication principle, we choose red to resemble traffic lights or brake lights, in line with earlier work [11, 34]. The viewing angle is more important for the system's optical behaviour. Many LEDs use a lens that narrows the beam, which can make reflections less diffuse over short distances. Figure 3.2 compares LEDs with viewing angles of 8° and 120°. We choose the 120° LED shown in Figure 3.2b to increase the effective viewing angle in the LoS case and to avoid uneven reflections in the NLoS case. Specifically, we use the Broadcom HSMC-A100-Q00J1 [40].

We design and manufacture a custom Printed Circuit Board (PCB) to hold the shift registers and LEDs. The PCB also includes the required power-management circuitry. Appendix C contains the full PCB design.

To control the matrix, we use a Teensy 4.1. We chose this platform because it combines high speed with ease of use. The Teensy drives the matrix over the Serial Peripheral Interface (SPI) at 20 MHz. This setup provides a stable 20 kHz refresh rate for all LEDs, including the required frame computation.

During normal operation, in which the LEDs have an average duty cycle of 50%, the transmitter consumes 13,4 W.

Figure 3.3: **Prophesee EVK4HD Event Camera.**

3.2 Receiver

We use the Prophesee EVK4HD as the receiver. This camera uses the Sony IMX636 sensor. The sensor provides a spatial resolution of 1280×720 pixels [5], which is sufficient to cover each transmitter LED with multiple pixels. It also provides a temporal resolution of 1 µs [5], which is orders of magnitude higher than the refresh rate of the matrix.

The camera manufacturer provides a Software Development Kit (SDK), namely the Metavision SDK. This SDK publishes camera events in real time via an Application Programming Interface (API).

We use this API to implement the decoding algorithms described in Chapter 5. In particular, we build on the provided multithreaded pipeline architecture to process events and decode them in near real time.

Chapter 4

Modulation

This chapter describes the modulation methods developed in this thesis. We aim to design a modulation scheme that can be decoded over both the LoS and NLoS links while maintaining high throughput on both.

The LoS link can support a higher throughput than the NLoS link because it can use both the spatial and temporal dimensions to encode data [6, 15]. By contrast, in the NLoS setting, reflected light blurs neighbouring channels together. This effect is commonly modelled as a Gaussian filter [23–25]. As a result, the NLoS link can encode data only in the temporal dimension.

As discussed in Chapter 2, no prior work considers both links at the same time. Existing work that encodes data in the spatial dimension does not examine how this modulation appears in an NLoS setting [6, 9, 13]. Conversely, work on NLoS event-camera communication does not encode data spatially at the transmitter [14, 15].

To address this gap, we design a new modulation scheme that transmits a high-fidelity stream in the spatial and temporal dimensions, and a low-fidelity stream in the temporal dimension only. Figure 4.1 illustrates the main idea. In the LoS setting, we transmit multiple bits per frame. In the NLoS setting, we transmit one bit per frame. Since the transmitter also provides illumination, a further design objective is to reduce flicker.

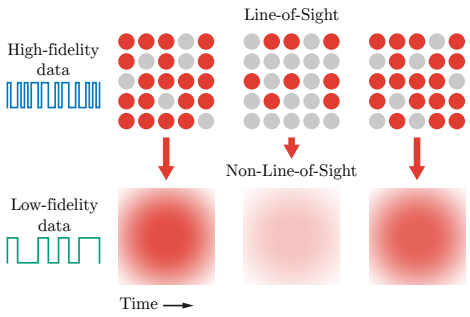


Figure 4.1: **High-fidelity data is encoded in the matrix's pixels and over time (the spatial and temporal dimensions). Low-fidelity data is encoded in the average brightness of all LEDs, so only in time (temporal dimension).**

This chapter is organised as follows. We begin in Section 4.1 by introducing our modulation method, Dual On-Off Keying (DOOK). We then examine how DOOK can be combined with other modulation schemes in Section 4.2 to reduce error and flicker. Finally, Section 4.3 explains how we group frames into packets.

4.1 Dual On-Off Keying (DOOK)

This section introduces the main modulation method proposed in this thesis: Dual On-Off Keying (DOOK). DOOK is a multi-fidelity modulation scheme. It encodes high-fidelity data in each spatial frame and low-fidelity data in the average brightness of a sequence of frames. In this way, the high-fidelity stream uses both the spatial and temporal dimensions, whereas the low-fidelity stream uses only the temporal dimension.

We first explain how DOOK combines a high-fidelity stream with a low-fidelity stream. We then discuss an ambiguity that arises when DOOK is received with an event camera. Finally, we introduce the measures we use to improve decodability and reduce flicker.

4.1.1 Combining two bitstreams

DOOK takes two input bitstreams: a high-fidelity bitstream and a low-fidelity bitstream. We encode these streams as follows. Figure 4.2 shows an example.

1. We split the high-fidelity message into chunks. We choose the chunk size such that, after encoding, one chunk fits exactly on the LED matrix.
2. For each chunk, DOOK provides two encodings: a dark one and a bright one. If the original chunk contains more `0` bits than `1` bits, or an equal number of `0` and `1` bits, we treat the original as the dark version and the inverted chunk, in which all bits are flipped, as the bright version. If the original chunk contains more `1` bits than `0` bits, we treat the original as the bright version and the inverted chunk as the dark version.
3. We append an 'inverted flag' to indicate which version is transmitted. We append a `0` after a non-inverted chunk and a `1` after an inverted chunk.
4. We sequence frames in time. For each chunk, we select either the dark frame or the bright frame based on the low-fidelity bit. For `0`, we select the dark frame, and for `1`, we select the bright frame.

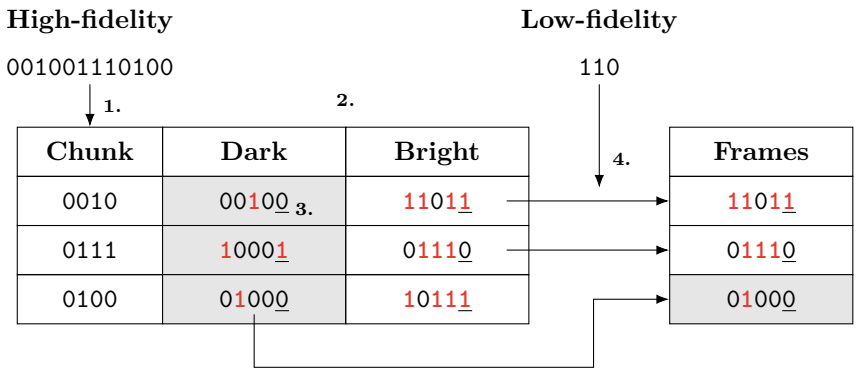


Figure 4.2: **Example of encoding two data streams with DOOK. 1. The high-fidelity message is split into chunks. 2. Dark and bright frames are generated for each chunk. 3. The inverted flag is underlined. 4. The frames are selected based on the low-fidelity data.**

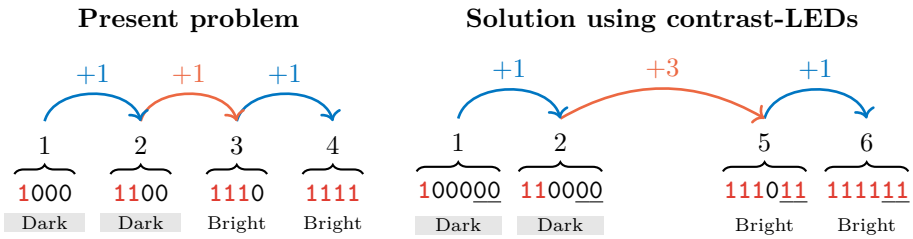


Figure 4.3: **Example of the effect of contrast-LEDs. The bottom rows show the bit string and whether it is a dark or bright frame. The number above it indicates how many LEDs are on. The number above the arrow indicates the difference between two frames, as observed by an event camera. Without contrast-LEDs, the change in brightness is the same when transmitting the same low-fidelity bit (blue arrow) and when going from a dark frame to a bright frame (orange arrow). With contrast-LEDs (underlined), the change in brightness between a dark frame and a bright frame is increased (orange arrow).**

4.1.2 Ambiguity for event cameras

To decode the low-fidelity data over the NLoS link, we must determine whether a frame is dark or bright. In an NLoS setting, the receiver cannot resolve individual LEDs. Instead, it observes only their average brightness.

This presents an extra challenge when using event cameras. A conventional camera can measure absolute brightness and can therefore observe whether a frame is dark or bright. An event camera cannot do this. It reports only whether pixels become brighter or darker between successive frames. For this reason, we cannot detect the transmitted low-fidelity bit from the absolute brightness of a single frame. Instead, we must infer it from the number of events generated during the transition between two frames. The larger the brightness change, the more events are generated.

This leads to an ambiguity, illustrated in Figure 4.3. The example contains four frames: `1000`, `1100`, `1110`, and `1111`. Here, `0` means that an LED is off and `1` means that it is on. The first two frames are dark because they contain more `0`s than `1`s. The last two frames are bright because they contain more `1`s than `0`s. In the NLoS setting, the light from individual LEDs blurs together, so the total brightness can be described by the number of LEDs that are on, namely 1, 2, 3, and 4.

A conventional camera can detect these brightness levels and therefore distinguish a dark frame from a bright one. An event camera cannot observe these levels directly. It only observes changes between frames, which in this example are always +1. Consequently, the camera cannot distinguish a dark→dark or bright→bright transition (blue arrow) from a dark→bright or bright→dark transition (orange arrow). In both cases, the observed brightness change is the same (+1).

4.1.3 Contrast-LEDs

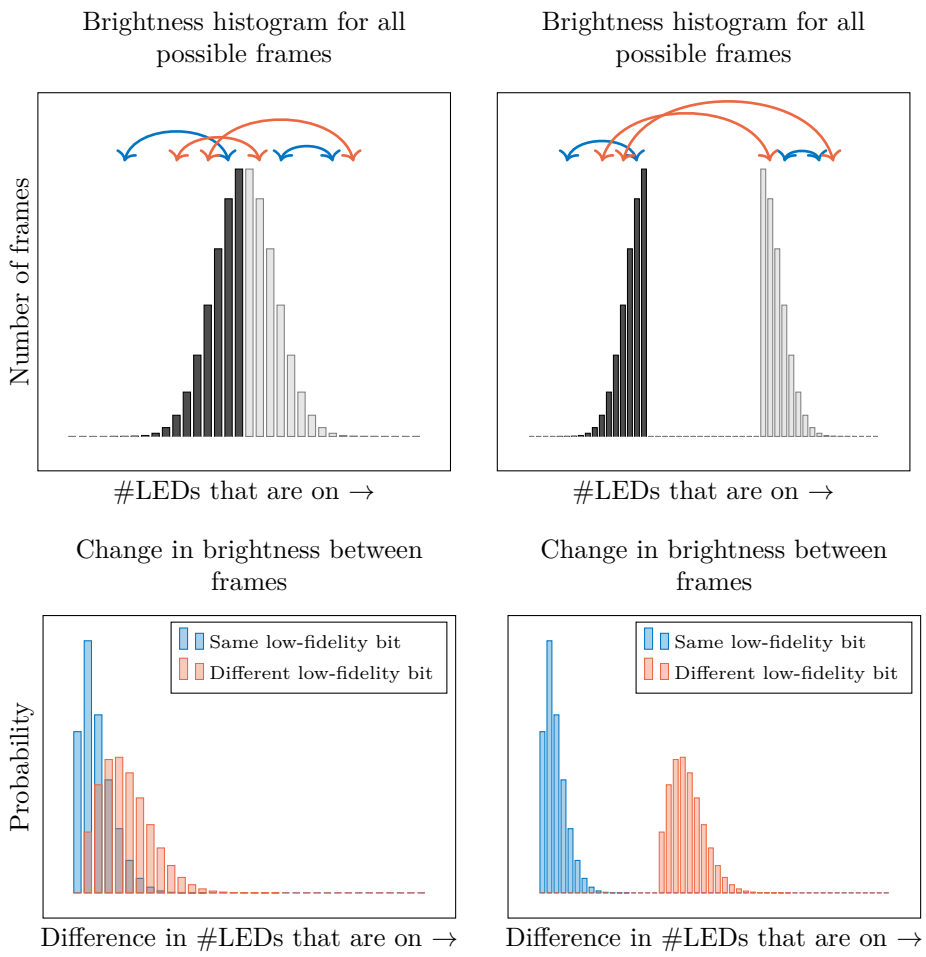
We address this ambiguity by adding 'contrast-LEDs'. In Figure 4.3, two contrast-LEDs are added and are shown underlined. These contrast-LEDs always take the value of the low-fidelity bit. They are off in a dark frame and on in a bright frame. As a result, they increase the difference in average brightness between dark and bright frames.

In the example, the brightness change between a dark frame and a bright frame increases from +1 to +3 (orange arrow). This enables the event camera to distinguish dark→dark and bright→bright transitions (blue arrow) from dark→bright and bright→dark transitions (orange arrow). The contrast-LEDs therefore make the low-fidelity transitions easier to separate.

Figure 4.4 on the following page shows a more general example of this effect. The top-left plot in Figure 4.4a shows a histogram of all possible frames with 32 data-LEDs and their brightness, made using a simulation. The black bars represent dark frames, and the grey bars represent bright frames. The bottom-left plot shows the distribution of brightness changes for all possible transitions. It separates dark→dark and bright→bright transitions, shown in blue, from dark→bright and bright→dark transitions, shown in orange. Without contrast-LEDs, these two groups overlap. As a result, some transitions cannot be classified reliably with an event camera.

The right-hand plots in Figure 4.4b show the effect of adding 16 contrast-LEDs. Bright frames become brighter, and the grey bars therefore shift to the right. At the same time, the brightness change for both dark→bright and bright→dark transitions increases, whereas the brightness change for dark→dark and bright→bright transitions remains the same. In the bottom plot, the orange group therefore shifts to the right and no longer overlaps with the blue group. This makes the transitions distinguishable.

Adding contrast-LEDs thus improves low-fidelity decoding. However, the matrix contains a fixed number of LEDs. Each contrast-LED reduces the number of data-LEDs that remain available for the high-fidelity stream. Consequently, using more contrast-LEDs improves NLoS decoding but lowers the LoS throughput.



(a) **32 data-LEDs and 0 contrast-LEDs. For some changes in brightness, the transition could be either in the blue group or in the orange group.**

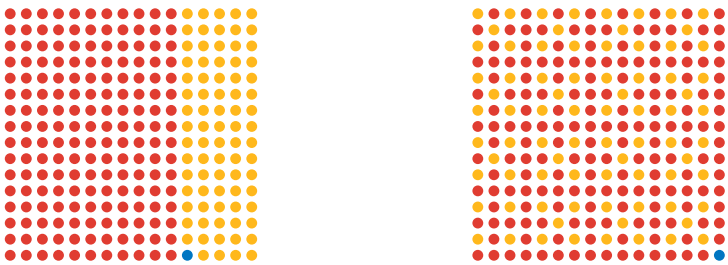
(b) **32 data-LEDs and 16 contrast-LEDs. The blue and orange groups are separated.**

Figure 4.4: **Top: A histogram of the brightness of all possible frames with 32 data-LEDs. Blue arrows indicate transitioning between frames with the same low-fidelity data (dark→dark or bright→bright), and orange arrows indicate transitioning between frames with different low-fidelity data (dark→bright or bright→dark). Bottom: The change in brightness when making a transition between frames. The blue group is the change in brightness when transitioning between frames with the same low-fidelity data. The orange group is the change in brightness when transitioning between frames with different low-fidelity data.**

4.1.4 Distributing contrast-LEDs

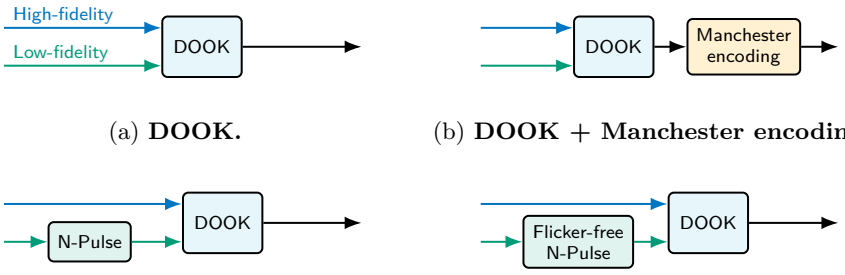
Contrast-LEDs can be distributed across the matrix in different ways. One option is to cluster them, as shown in Figure 4.5a. This creates a region of the matrix that visibly modulates differently from the rest of the matrix. Another option is to distribute them more evenly so that the emitted illumination remains more homogeneous. In this thesis, we use Bayer's threshold map for this distribution [41], as shown in Figure 4.5b.

Taken together, the resulting scheme applies an OOK-like principle on both links, but at different levels. On the LoS link, the high-fidelity data is encoded by the state of individual LEDs, so that an LED that is on represents a `1` and an LED that is off represents a `0`, or the inverse when the inverted flag is set. On the NLoS link, the low-fidelity data is encoded by the brightness of whole frames, so that a dark frame represents a `0` and a bright frame represents a `1`. Hence the name Dual On-Off Keying.



(a) **All contrast-LEDs clustered.** (b) **Contrast-LEDs following Bayer's pattern.**

Figure 4.5: **Different distributions of 76 contrast-LEDs on a 16×16 matrix. Red indicates data-LEDs, blue indicates the inverted flag, and yellow indicates contrast-LEDs.**



(a) **DOOK.** (b) **DOOK + Manchester encoding.**

(c) **DOOK + N-Pulse.** (d) **DOOK + flicker-free N-Pulse.**

Figure 4.6: **Overview of four modulation variants layered on top of DOOK. The high-fidelity input stream is shown in blue and the low-fidelity input stream in green. Our contributions are DOOK and flicker-free N-Pulse, while N-Pulse is part of the SoA.**

4.2 Combining DOOK with other modulation schemes

The previous section introduced DOOK as the basic multi-fidelity modulation method used in this thesis. We now consider how DOOK can be combined with other modulation schemes. The aim of this section is to modify DOOK to reduce flicker or improve robustness on the low-fidelity NLoS link.

We consider four variants, shown in Figure 4.6. We begin with unmodified DOOK in Section 4.2.1. We then introduce DOOK with Manchester encoding in Section 4.2.2. After that, we discuss DOOK combined with N-Pulse modulation in Section 4.2.3. Finally, we present our novel flicker-free variant of N-Pulse modulation in Section 4.2.4 and compare all four variants in Section 4.2.5.

4.2.1 DOOK

We begin with the unmodified DOOK pipeline. In this case, the low-fidelity NLoS link is comparable to OOK, as introduced in the previous section. Within DOOK, this means that a dark frame represents `0`, whereas a bright frame represents `1`. Under this scheme, one low-fidelity bit is transmitted per frame.

4.2.2 DOOK + Manchester encoding

Another important aspect of the design is flicker reduction. To make both the high-fidelity and low-fidelity streams flicker-free, we apply Manchester encoding. Manchester encoding balances the time for which the light is on and off by representing each bit as an off→on or on→off transition, as shown in Figure 2.1 on page 6.

We apply Manchester encoding to the output of the unmodified DOOK pipeline. This is illustrated in Figure 4.6b. In practice, we implement this by duplicating and inverting every frame. As a result, every second frame is the inverse of the previous one.

The main advantage of this approach is that the average brightness remains 0,5 in both the LoS and NLoS links, independent of the transmitted data. This

reduces flicker on both links. However, this benefit comes at the cost of throughput. Because each original frame is now represented by two frames, the throughput on both links is reduced by a factor of two.

4.2.3 DOOK + N-Pulse

We now turn to N-Pulse modulation. The work by Nishar, Marefat and Ashok, which also studies NLoS VLC with event cameras, compares OOK with N-Pulse modulation and reports better performance for N-Pulse [15]. For this reason, we also implement N-Pulse on the low-fidelity NLoS link and compare it with the previous two variants.

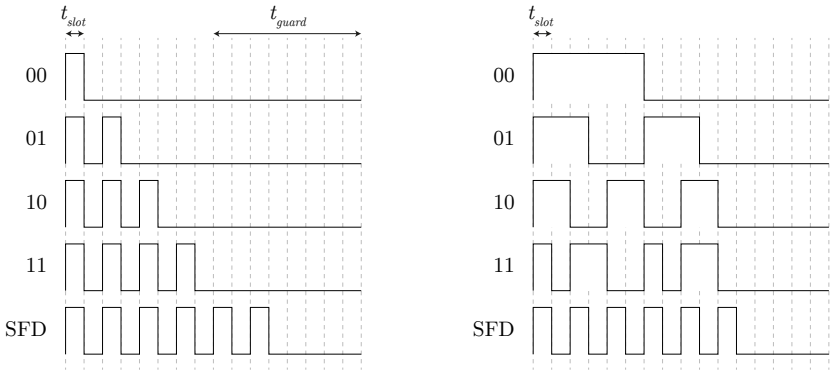
N-Pulse is designed for event-camera VLC and has low demodulation complexity [12]. It encodes two bits per symbol using pulses. One pulse represents `00`, two pulses represent `01`, and so forth, as shown in Figure 4.7a. A guard time t_{guard} is added to create a detectable separation between symbols. We also use a Start Frame Delimiter (SFD) as a preamble to mark the start and end of packets.

We combine N-Pulse with DOOK by applying N-Pulse modulation to the low-fidelity bitstream, as shown in Figure 4.6c. Every 2 bits from the low-fidelity stream are first encoded into 16 bits, after which DOOK is applied. Since each symbol spans 16 frames and encodes 2 bits, N-Pulse has one eighth of the low-fidelity throughput of regular DOOK.

We decode N-Pulse by counting the number of events in each symbol. A symbol with one pulse generates the fewest events, while symbols with more pulses generate more events. In this way, demodulation reduces to distinguishing symbols by their event count.

4.2.4 DOOK + flicker-free N-Pulse

To reduce flicker while retaining the basic N-Pulse structure, we design a flicker-free variant of N-Pulse in which every symbol has the same duty cycle. Figure 4.7b shows this encoding. The benefit of this design is that the average



(a) **Original N-Pulse encoding [12].** (b) **Flicker-free variant of N-Pulse encoding.**

Figure 4.7: **N-Pulse encoding with 2 bits per symbol.**

signal value remains independent of the transmitted data. At the same time, it preserves the same throughput as standard N-Pulse modulation.

When we combine this scheme with DOOK, as shown in Figure 4.6d, the high-fidelity stream can still reintroduce flicker. This is because the flicker-free N-Pulse scheme is applied only to the low-fidelity stream. Thus, every two low-fidelity bits are always encoded using 6 bright frames and 10 dark frames. However, bright frames and dark frames do not always have exactly the same brightness, because that brightness still depends on the high-fidelity data. As a result, the combined scheme is not fully flicker-free.

4.2.5 Comparison of variants

Table 4.1 compares the four variants introduced above against the baseline DOOK pipeline. The throughput values are given relative to this baseline.

Adding Manchester encoding halves the throughput on both links because each original frame is replaced by two frames. Its main advantage is that the duty cycle of each LED is forced to 50%, which reduces flicker.

N-Pulse is applied only to the low-fidelity stream. The LoS throughput, therefore, remains unchanged. However, each 2-bit low-fidelity symbol is represented by 16 frames. As a result, the NLoS throughput is reduced to one eighth of the unmodified DOOK pipeline.

Flicker-free N-Pulse has the same throughput as standard N-Pulse. Its advantage is that every two low-fidelity bits are represented by the same ratio of bright and dark frames, reducing flicker. However, it is not fully flicker-free because not every bright frame or every dark frame has exactly the same brightness.

The effect of these variants on flicker and error is quantified and analysed in Chapter 6.

Table 4.1: **Comparison of the four modulation variants layered on top of DOOK. Throughput values are relative.**

Variant	LoS throughput	NLoS throughput	Flicker-free
DOOK	1	1	×
+ Manchester encoding	$\frac{1}{2}$	$\frac{1}{2}$	✓
+ N-Pulse	1	$\frac{1}{8}$	×
+ flicker-free N-Pulse	1	$\frac{1}{8}$	∼

4.3 Packets

The final part of this chapter describes how we group frames into packets. To simplify data processing and measurement, we group frames into packets consisting of a preamble followed by 512 payload frames. The choice of preamble depends on the modulation scheme. In all cases, the preamble is transmitted using frames in which either all LEDs are on, or all LEDs are off.

For DOOK and DOOK + Manchester encoding, we use `00001111` as the preamble. This pattern has a 50% duty cycle and therefore does not introduce flicker when Manchester encoding is used. For regular DOOK, if `00001111` appears in the payload, the encoder must escape it. Otherwise, in the NLoS setting, this sequence could be misclassified as the preamble.

For N-Pulse and flicker-free N-Pulse, we use the SFD shown in Figure 4.7a and Figure 4.7b as the preamble.

Chapter 5

Decoding

This chapter describes the decoding methods used for the modulation schemes introduced in Chapter 4. We implement each decoder as a three-stage pipeline consisting of Detect, Sync, and Decode. In the Detect stage, we use image processing to locate the LEDs or the Region of Interest used in the later stages. In the Sync stage, we estimate the phase offset between transmitter and receiver so that the receiver samples each frame at the intended moment within the slot. For the NLoS pipelines, this stage also estimates the thresholds used to distinguish symbols. In the Decode stage, we demodulate the signal and assemble packets.

The chapter is organised as follows. We begin in Section 5.1 with the LoS DOOK decoding pipeline, which reconstructs the high-fidelity stream. We then turn in Section 5.2 to the NLoS DOOK decoding pipeline for the low-fidelity stream. Finally, Section 5.3 presents the NLoS N-Pulse decoding pipeline, specifically for the DOOK + N-Pulse and DOOK + flicker-free N-Pulse variants. Taken together, these sections describe the receiver-side processing for all modulation variants considered in this thesis.

5.1 LoS DOOK decoding pipeline

This section describes the decoding pipeline for the high-fidelity LoS stream in DOOK. The pipeline consists of the same three stages introduced above, namely Detect, Sync, and Decode. The Detect stage estimates the LED coordinates and passes them to the Sync and Decode stages. The Sync stage estimates the phase offset and passes it to the Decode stage. Figure 5.1 shows the complete pipeline. This pipeline decodes the LoS component for all four DOOK variants introduced in Section 4.2.

Unlike several state-of-the-art systems, which process the event data offline after acquisition, we implement this pipeline online, meaning that events are processed as they arrive. However, the current LoS implementation does not run in real time. The main reason is the high event rate generated by the LoS link. With more processing power, the same implementation could, in principle, run in real time.

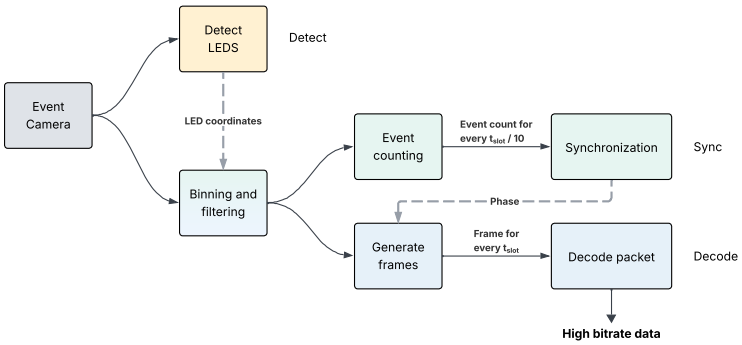


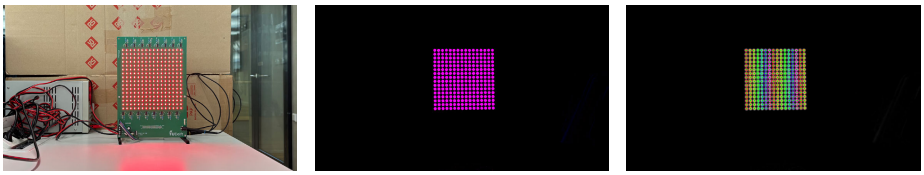
Figure 5.1: **The pipeline for decoding the high-fidelity LoS DOOK data. Dashed lines represent the results of one stage being used by a later stage. Solid lines represent the events emitted by the event camera as the different stages process them.**

5.1.1 Detect

The Detect stage determines the pixel location of each LED.

To do this, we first construct a histo frame from the event stream, as shown in Figure 5.2b. A histo frame counts events over a time window and forms a bitmap in which red indicates on-events and blue indicates off-events [42]. When both on-events and off-events occur frequently at the same pixel, the resulting colour becomes magenta. This representation is similar to the heat map used by Wang et al. to detect channels [6].

We then convert the histo frame to greyscale and apply a blur to suppress noise. Next, we use OpenCV's `SimpleBlobDetector` and `findCirclesGrid` to detect the LED locations. In Figure 5.2c, each coloured circle indicates one detected LED. We pass the resulting pixel coordinates to the Sync and Decode stages.



(a) **Scene that the event camera is looking at.** (b) **Histo frame built using the generated events.** (c) **Detected LED positions using OpenCV.**

Figure 5.2: **Detect stage in the LoS pipeline.**

5.1.2 Sync

The purpose of the Sync stage is to determine when the matrix should be sampled. To do this reliably, we first reduce the number of irrelevant events, then estimate the phase offset between the transmitter and receiver. This phase offset indicates the point within each slot at which the matrix state can be sampled most reliably.

Filtering and binning

Before estimating the phase offset, we first filter irrelevant events. We do this by enabling a trail filter. This filter removes sequential events with the same polarity at each pixel [42]. For example, if a pixel generates multiple consecutive off-events, we process only the first one. Only after that pixel generates an on-event again do we process it. We can do this because these repeated events are not used in the later decoding stages. The work by Wang et al. (Selene) also uses the trail filter successfully [6].

We apply this trail filter in the camera rather than in our software. As a result, the filtered events are not transmitted to the computer. This makes more efficient use of the camera link and avoids unnecessary processing on the host computer. In turn, this reduces the overall computational load.

After trail filtering, we apply binning. Instead of processing only the centre pixel of each detected LED, we also include events within a distance of 2 pixels from that centre. The work by Shen, Chen and Tsai shows that binning leads to a significantly lower error rate [11]. Finally, we discard all remaining events outside these selected pixels.

Need for synchronization

The need for synchronisation follows from the way DOOK is decoded on the LoS link. The work by Wang et al. (Selene) decodes each channel independently and therefore avoids tight synchronisation by using event-relative decoding [6]. In that approach, the transmitted symbol is inferred from the time between events. However, this assumes that each LED carries an independent data stream [6].

In DOOK, all LEDs together form a single message. For this reason, we must sample the full matrix at one common point in time for each frame. However, we cannot do this at an arbitrary moment within the slot.

The main difficulty is that a high event rate can delay events. An event camera does not assign a timestamp at the exact moment the light intensity changes at a pixel. Instead, the timestamp is assigned later when the event is processed by the camera's Microcontroller Unit (MCU) [6, 43]. When many events are generated simultaneously, a queue forms, delaying the timestamping of some events.

This effect is illustrated in Figure 5.3. A new frame is displayed on the matrix at each red arrow. Because many events are generated at once, the internal queue grows. As a result, timestamps are assigned later than the physical event occurrence, and these delayed timestamps are then sent to the computer. This period corresponds to the transient state. Once the event rate decreases again, the system enters the stable state.

Our aim is therefore to sample the state of each LED during the stable state, after the events for the current frame have arrived and before the next burst of

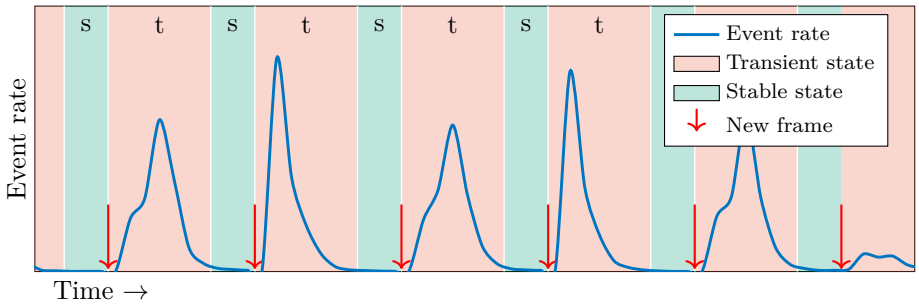


Figure 5.3: **After a new frame is displayed on the matrix, events are generated. The events do not appear instantly because of the large number. During the transient state, events are still arriving, whereas in the stable state, all events for that frame have arrived.**

events begins. To achieve this, we detect the start of the next transient state and sample immediately before it. The next section explains how we estimate this sampling point.

Detecting when to sample

To determine the correct sampling point, we estimate the phase offset between transmitter and receiver by subsampling each slot. The resulting phase offset is then used in the Decode stage to sample the matrix at the end of the stable state, just before the next transient state begins.

The key idea is that the best sampling point is the point within a slot where the event rate is lowest. This point lies in the stable state, after the events caused by the current frame have arrived and before the next frame generates a new burst of events. The work by Tang, Yamazato and Arai uses subsampling with a factor of 2 to avoid sampling during the transient state [44]. We follow the same principle, but use a higher subsample rate to estimate the phase more precisely.

More specifically, we estimate the phase offset as follows, as illustrated in Figure 5.4:

1. We divide each slot of duration t_{slot} into 10 equally spaced subsamples, each with duration $t_{slot}/10$.

2. For each candidate phase $p \in \{0, 1, \ldots, 9\}$, we count the events that fall into the p-th subsample of every slot. These candidate phases are shown in different colours in Figure 5.4.

3. For each phase p, we sum the event counts over all slots.

4. We select the phase with the fewest events and use it as the phase offset.

The selected phase corresponds to the point in the slot where the event rate is minimal. This is in the stable state, which is exactly when we want to sample the LED matrix. We therefore pass this phase offset to the Decode stage.

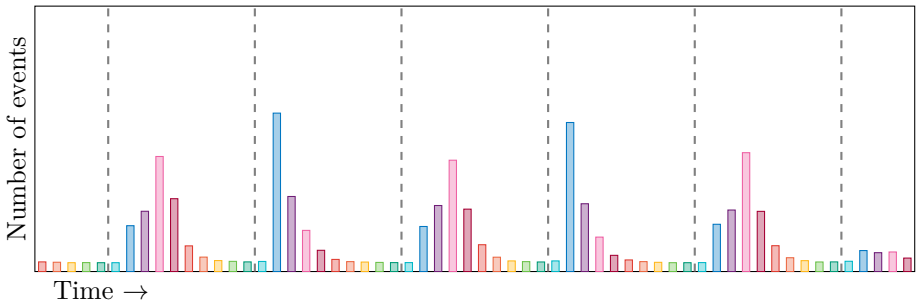


Figure 5.4: **Sync stage in the LoS pipeline. Each bar indicates the number of events per subsample. Each colour represents a group of subsamples. The group with the fewest events is selected (dark green). This group marks the end of the stable state (dashed line).**

5.1.3 Decode

In the Decode stage, we generate one frame every t_{slot} using the phase offset estimated in the Sync stage. Because the phase offset places the sample at the end of each stable state, the generated frame represents the latest reliable state of the matrix for that slot. For each LED, the frame stores `0` or `1` based on the polarity of the most recent event at the corresponding pixel. This yields 256 values, one for each LED.

We then apply the inverse of the DOOK encoding described in Section 4.1. First, we ignore the contrast-LEDs. Next, we invert the data bits if the inverted flag is set. If the transmitted signal uses the DOOK + Manchester variant from Section 4.2.2, we also decode the Manchester encoding at this stage. Finally, we reassemble the chunks to reconstruct the original message. This yields the original high-fidelity data stream.

5.2 NLoS DOOK decoding pipeline

This section describes the decoding pipeline for the low-fidelity NLoS stream in DOOK. At first sight, this stream resembles conventional OOK. In regular OOK, a light off represents a `0` and a light on represents a `1`, whereas in DOOK, a dark frame represents a `0` and a bright frame represents a `1`. However, as explained below, this similarity does not mean that existing OOK decoding methods can be applied directly.

Prior work on event-camera OOK typically decodes the transmitted bit from the polarity of the most recent event at the transmitter location [6, 11]. The polarity indicates whether a pixel has become brighter or darker. Accordingly, a negative polarity, meaning that the pixel has become darker, is interpreted as `0`, whereas a positive polarity, meaning that the pixel has become brighter, is interpreted as `1` [6, 11].

This approach works only when repeated bits generate no events. In DOOK, however, not every dark frame has exactly the same brightness, nor does every bright frame. As a result, repeated low-fidelity bits can still produce events. This can lead to incorrect symbol decisions. For example, a `0`→`0` transition, which corresponds to a dark→dark transition in DOOK, can generate positive events if the second dark frame is slightly brighter than the first. A polarity-based decoder would then classify the second bit as `1`, even though the transmitted bit is still `0`. For this reason, we cannot determine the transmitted bit solely from the polarity of the most recent event.

We therefore use an event-count-based decoder for the NLoS DOOK stream. The key observation is that `0`→`0` (dark→dark) and `1`→`1` (bright→bright) transitions generate relatively few events, whereas `0`→`1` (dark→bright) and `1`→`0` (bright→dark) transitions generate many events. In other words, the event count indicates whether the low-fidelity bit remains unchanged or flips. This observation forms the basis of the decoding pipeline introduced in this section.

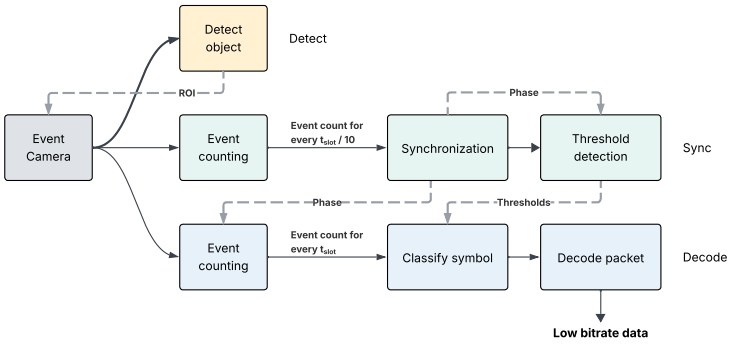


Figure 5.5: **The pipeline for decoding the low-fidelity NLoS DOOK data. Dashed lines represent the results of one stage being used by a later stage. Solid lines represent the events emitted by the event camera as the different stages process them.**

As in the LoS DOOK pipeline, the NLoS DOOK pipeline consists of three stages: Detect, Sync, and Decode. Figure 5.5 shows the full pipeline. As with the LoS pipeline, we implement it online, meaning that the events are processed as they arrive. Unlike the LoS pipeline, however, this pipeline runs in real time because the NLoS link generates a much lower event rate. This pipeline is used to decode the NLoS link of the DOOK and DOOK + Manchester variants introduced in Sections 4.2.1 and 4.2.2.

5.2.1 Detect

The Detect stage determines a Region of Interest (ROI). In the later stages, only events inside this ROI are processed. This suppresses noise from other light sources and from weaker secondary reflections.

The work by Nishar, Marefat and Ashok detects the ROI by constructing a binary image in which pixels with recent events are active, and all other pixels are inactive [15]. We instead use a histo frame, shown in Figure 5.7b. This allows us to use event counts as a measure of pixel brightness, which helps distinguish the main reflection from weaker reflections and background noise.

As a result, the ROI detection step can remain relatively simple. We blur the histo frame, locate the brightest pixel, and then cluster the surrounding pixels. We then draw a bounding box around this cluster.

Finally, we apply this bounding box as an ROI filter in the camera. Applying the filter in the camera rather than in software makes more efficient use of the camera's event-rate limit. Consequently, noise and weaker reflections do not consume part of this event budget, which supports higher data rates.

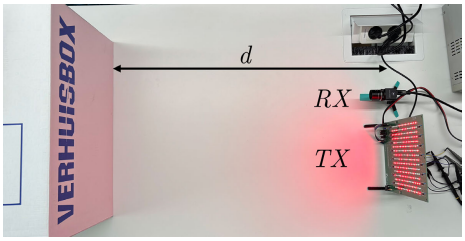
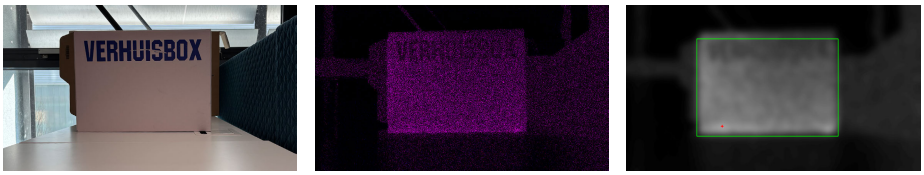


Figure 5.6: **NLoS setup.**



(a) **Scene that the event camera is looking at.**

(b) **Histo frame built using the generated events.**

(c) **Detected ROI using the brightest pixel and clustering.**

Figure 5.7: **Detect stage in the NLoS pipeline.**

5.2.2 Sync

The Sync stage serves two purposes. First, it estimates the phase offset between transmitter and receiver. Second, it determines the threshold for separating transitions with few events from those with many events.

We begin by estimating the phase offset in the same way as in the LoS DOOK pipeline. We divide each slot into 10 subsamples and select the phase with the fewest events, as illustrated earlier in Figure 5.4.

At this point, the objective differs from that of the LoS pipeline. For the LoS stream, we sample the stable state of the matrix. For the NLoS stream, by contrast, we are interested in the number of events generated during the transient state. This is because a larger brightness change generates more events. We therefore use the event count per frame to distinguish dark→dark and bright→bright transitions from dark→bright and bright→dark transitions.

To do this, we estimate a threshold from the observed event counts. The procedure is illustrated in Figure 5.8:

1. Using the detected phase offset, we count the number of events in each frame and build a histogram of these counts. The height of each bar indicates how many frames produce a given event count.

2. We fit two Gaussian distributions to this histogram using OpenCV's Expectation-Maximisation (EM) algorithm.

3. We compute the intersection of the two curves and use it as the threshold.

Frames with fewer events than this threshold indicate that the low-fidelity bit remains unchanged. Frames with more events than the threshold indicate that the low-fidelity bit flips.

In the current implementation, this threshold is obtained through calibration and may therefore need to be updated when the environment changes, for example, due to changes in ambient light. A positive aspect, however, is that this calibration does not require a dedicated calibration sequence from the transmitter. Instead, the receiver can estimate the required threshold from a regular data stream. We return to this point in Chapter 7.

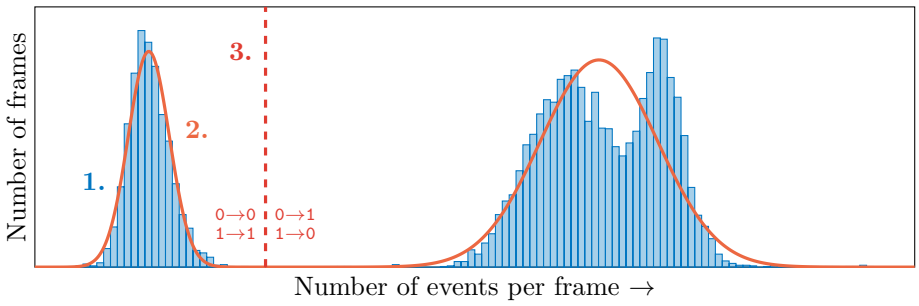


Figure 5.8: **Sync stage in the NLoS DOOK pipeline. 1. A histogram is created of all frames, where the bins are the number of events per frame. 2. Two Gaussian distributions are fitted. 3. A threshold is calculated at the intersection.**

5.2.3 Decode

In the Decode stage, we count the number of events in each frame using the phase offset estimated in the Sync stage. We then compare this count with the threshold to determine whether the low-fidelity bit remains unchanged or flips. When a flip is detected, we use the event polarity to determine whether the new bit is `1` or `0`.

Next, we detect preambles and assemble packets. If the transmitted signal uses the DOOK + Manchester variant, we decode the Manchester encoding in this final step. Finally, we reconstruct the original low-fidelity data stream.

5.3 NLoS N-Pulse decoding pipeline

This section describes the decoding pipeline used for the DOOK + N-Pulse and DOOK + flicker-free N-Pulse variants introduced in Sections 4.2.3 and 4.2.4. The pipeline is closely related to the NLoS DOOK pipeline from the previous section, but it is adapted to decode N-Pulse symbols rather than single-bit transitions.

The central idea behind N-Pulse decoding is straightforward. Because each symbol contains a different number of pulses, different symbols generate different numbers of events [12]. In principle, the symbols can therefore be decoded by counting events. However, these event counts still need to be mapped to the transmitted symbols. Existing work performs this mapping either using predefined thresholds or estimating them offline [12, 15].

In this thesis, we estimate these thresholds online. Figure 5.9 shows the full pipeline. Like the NLoS DOOK pipeline, this pipeline runs in real time.

5.3.1 Detect

The Detect stage is identical to that of the NLoS DOOK pipeline. We detect the same ROI and apply it in the camera. As before, this restricts subsequent processing to the main reflected signal, suppressing weaker reflections and background noise.

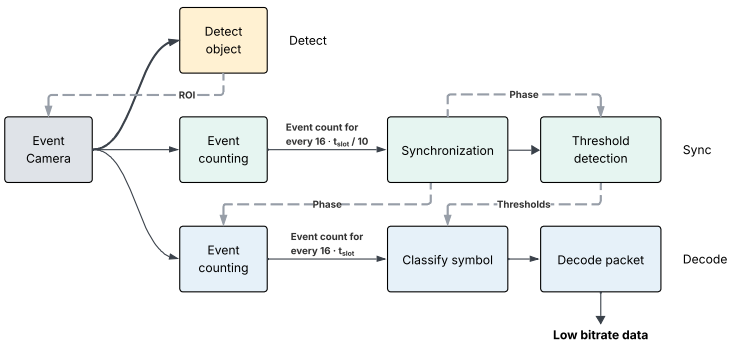


Figure 5.9: **The pipeline for decoding the low-fidelity NLoS N-Pulse data. Dashed lines represent the results of one stage being used by a later stage. Solid lines represent the events emitted by the event camera as the different stages process them.**

5.3.2 Sync

The Sync stage has two tasks. First, it determines the symbol boundaries. Second, it estimates the thresholds used to map event counts to N-Pulse symbols.

We begin by estimating the phase offset in the same spirit as in the DOOK pipelines. For N-Pulse, however, the relevant unit is no longer a single frame but a full symbol. Each N-Pulse symbol contains a guard time, t_{guard}, during which no pulse is transmitted. During this interval, few or no events are generated. This guard time, therefore, plays the same role as the stable state in the DOOK pipelines.

We exploit this property to locate the start and end of each symbol. As in the earlier Sync stages, we use subsampling and search for the phase with the fewest events. The difference is that we now apply this procedure over the duration of one N-Pulse symbol, which is $16 \cdot t_{slot}$, rather than over a single slot. This follows from the fact that one N-Pulse symbol spans 16 DOOK frames, as shown in Figure 4.7a. The phase with the fewest events must be during the guard time. We use this position to split the event stream into separate symbols.

Once the symbol boundaries are known, we estimate the thresholds needed for symbol classification. The procedure is illustrated in Figure 5.10:

1. Using the detected phase offset, we count the number of events in each symbol and build a histogram of these counts. The height of each bar indicates how many symbols produce a given event count.

2. We fit five Gaussian distributions to this histogram using OpenCV's EM algorithm, corresponding to the five possible symbols, namely `00`, `01`, `10`, `11`, and the SFD.

3. We compute the intersections between consecutive distributions and use them as thresholds for symbol classification.

As in the NLoS DOOK pipeline, these thresholds are currently obtained through calibration and may need to be recomputed when the environment changes.

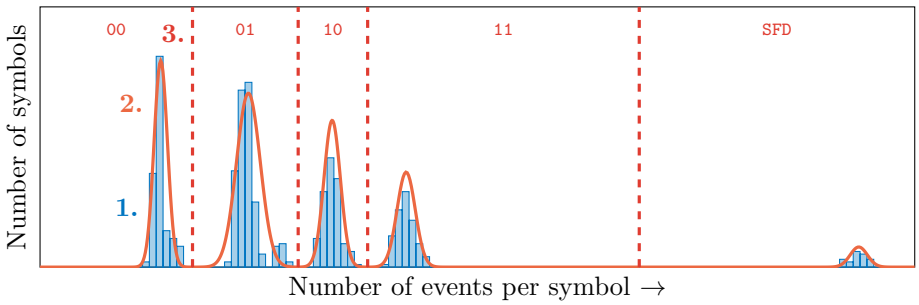


Figure 5.10: **Sync stage in the NLoS N-Pulse pipeline. 1. A histogram is created of all symbols, where the bins are the number of events per symbol. 2. Five Gaussian distributions are fitted. 3. The thresholds are calculated at the intersections of the distributions.**

5.3.3 Decode

In the Decode stage, we count the number of events per symbol and classify each symbol using the thresholds estimated in the Sync stage. We then detect the SFD and assemble packets. Finally, we reconstruct the original low-fidelity data stream.

For the DOOK + flicker-free N-Pulse variant from Section 4.2.4, we use exactly the same pipeline. This is possible because each symbol in the flicker-free variant contains the same number of transitions as in the original N-Pulse scheme.

Chapter 6

Evaluation

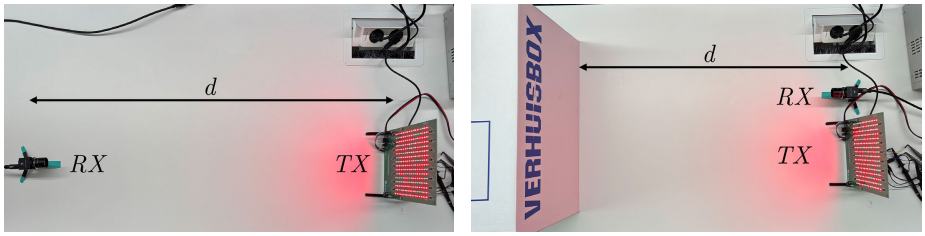
This chapter evaluates the system presented in the previous chapters. We examine communication performance in terms of Bit Error Rate and throughput. In addition, we assess the flicker produced by the different modulation schemes.

We begin in Section 6.1 by describing the experimental setup and the metrics used throughout the chapter. We then compare our results with the SoA in Section 6.2. Next, in Section 6.3, we analyse how key system parameters affect performance. Finally, Section 6.4 evaluates flicker and compares the flicker-free variants with their corresponding baselines.

6.1 Experimental Setup and Metrics

This section describes the experimental setups and defines the metrics used to evaluate the system. We use separate setups for the LoS and NLoS links, shown in Figure 6.1.

For the LoS experiments, shown in Figure 6.1a, we point the event camera directly at the transmitter at a distance d. For each distance, we manually adjust the focus and aperture to keep the image sharp while avoiding an excessive event rate.



(a) **Experimental setup for LoS experiments. Distance d is the distance between the transmitter and camera.**

(b) **Experimental setup for NLoS experiments. Distance d is the distance between the transmitter/camera and the reflective surface. The length of the optical path is $2 \cdot d$.**

Figure 6.1: **Experimental setups for LoS and NLoS experiments.**

Figure 6.2: **Objects used in the experiments: a diffuse white box (left) and a glossy white cup (right).**

For the NLoS experiments, shown in Figure 6.1b, we place the event camera next to the transmitter. Both are directed towards a reflective surface at a distance d. As reflective objects, we use either a diffuse white box or a glossy cup, shown in Figure 6.2. In this setup, the light travels a total path length of $2 \cdot d$.

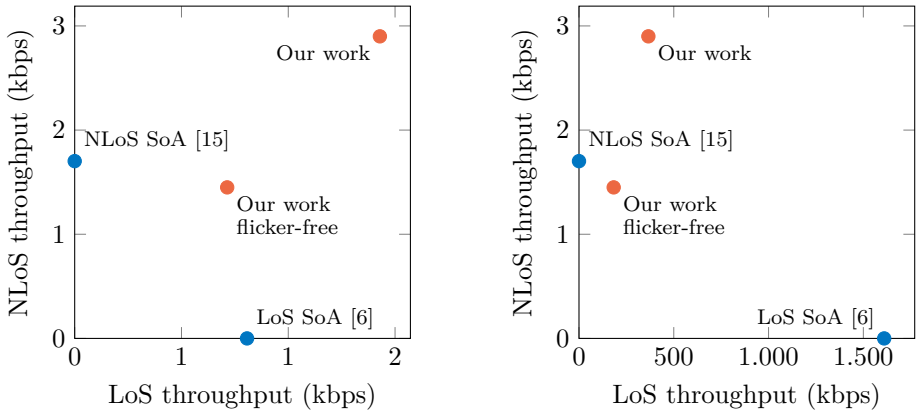
We evaluate both links using Bit Error Rate (BER) and throughput. The **Bit Error Rate** is the number of incorrectly received and demodulated bits divided by the total number of transmitted bits [12]. The **throughput** is the number of transmitted bits per second. We compute the throughput from the modulation scheme, t_{slot}, and, for the LoS link, the number of data-LEDs.

Unless stated otherwise, the LoS experiments use 10 packets, each containing 512 data frames. We place the camera at 100 cm from the transmitter. We use DOOK + Manchester encoding with a t_{slot} of 350 µs. In addition, we use 128 data-LEDs and 127 contrast-LEDs arranged according to the Bayer contrast pattern.

Unless stated otherwise, the NLoS experiments use 50 packets, each containing 512 data frames. We use a box positioned at 150 cm as the reflective surface. We use DOOK + Manchester encoding with a t_{slot} of 350 µs. As in the LoS setup, we use 128 data-LEDs and 127 contrast-LEDs with the Bayer contrast pattern.

6.2 Comparison with State of the Art

This section compares our results with the SoA. We first compare the maximum LoS and NLoS throughputs of our method with prior work in Section 6.2.1. We then compare the different DOOK variants introduced in Section 4.2 in terms of BER and throughput in Section 6.2.2.



(a) **Throughput per channel.** (b) **Total throughput.**

Figure 6.3: **Throughput for the LoS and NLoS links of our work, compared to the State of the Art.**

6.2.1 Highest achievable throughput

In this subsection, we compare the maximum throughput of our system with the SoA. For this comparison, we use DOOK and its flicker-free variant, DOOK + Manchester encoding. With the parameters defined in Section 6.1, DOOK achieves a total throughput of 366 kbps on the LoS link and 2,9 kbps on the NLoS link. On the LoS link, this corresponds to a per-channel throughput of 1,43 kbps. For the flicker-free variant, the throughput is exactly halved, resulting in 0,71 kbps per channel on the LoS link. In all of these cases, both links achieve a BER below 10^{-3}. Figure 6.3 compares these results with the work by Nishar, Marefat and Ashok for the NLoS link [15] and the work by Wang et al., called Selene, for the LoS link [6].

We simultaneously achieve higher per-channel throughputs on the LoS and NLoS links than the SoA. On the NLoS link, our throughput is 1,7× higher than that reported by Nishar, Marefat and Ashok. A likely reason is that DOOK minimises the number of events per bit, which is advantageous when the receiver is limited by its event rate [11]. By contrast, the N-Pulse modulation used by Nishar, Marefat and Ashok reaches this event-rate limit earlier. On the LoS link, our throughput per channel is 1,8× higher than that of Selene, as shown in Figure 6.3a. This difference is largely explained by the shorter t_{slot} used in our system, namely 350 µs, compared with Selene's dual refresh rates of 840 and 1680 µs. Selene uses a lower refresh rate near the edges of the frame because those channels update more slowly.

At the same time, Selene uses 1995 channels, whereas our system uses 256. As a result, Selene achieves a higher total LoS throughput, as shown in Figure 6.3b. Our design therefore outperforms the SoA on a per-channel basis, but not in total LoS throughput.

With flicker-free modulation, we achieve per-channel throughput comparable to flickering state-of-the-art systems. None of the SoA studies analyse or minimise flicker. By contrast, a key contribution of our work is that we both evaluate flicker and reduce it through the modulation design. Our flicker-free

variant achieves per-channel throughput comparable to flickering state-of-the-art systems, while also combining LoS and NLoS communication in a single transmitter.

In the current design, we cannot match Selene's total throughput because we encounter the same event-rate limitation. Increasing the number of channels would increase the event rate and eventually lead to dropped events. Scaling to more channels would therefore require fewer events per bit, for example, by reducing LED size or brightness. However, this would likely increase the NLoS error rate. In addition, the contrast-LEDs generate events in the LoS setup without contributing to the LoS throughput. This contrast-LEDs overhead remains even if Selene's transmitter or its other optimisations are adopted. For this reason, our current design cannot reach the same total LoS throughput. A more detailed comparison with Selene is provided in Appendix D.

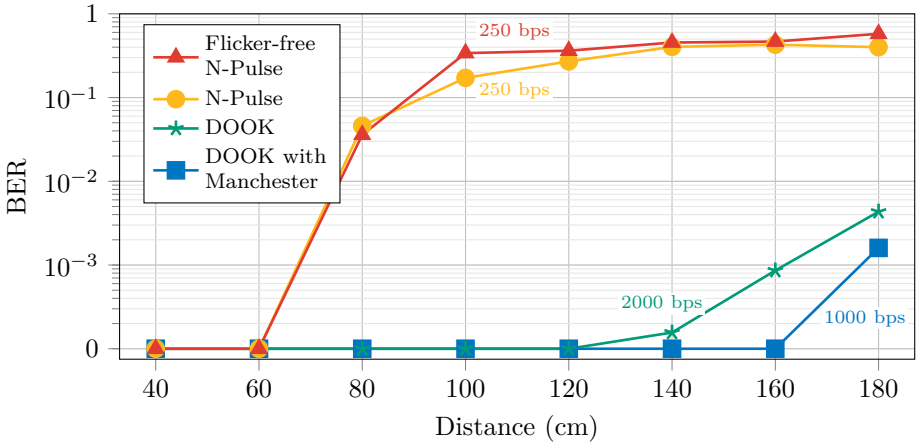


Figure 6.4: **The BER of the different modulation schemes at different distances. The modulation schemes use the same t_{slot}, so they differ in throughput, shown by the labels next to the lines.**

6.2.2 NLoS performance of different DOOK variants

We now compare the NLoS performance of the different DOOK variants, namely baseline DOOK, DOOK combined with state-of-the-art N-Pulse modulation, and their flicker-free variants. We compare these schemes in terms of BER over distance and throughput. For these experiments, we use a t_{slot} of 500 µs and the cup in Figure 6.2 as the reflective object at a distance of 100 cm. Figure 6.4 shows the results.

Basic DOOK outperforms DOOK + N-Pulse in both BER and throughput because DOOK symbols are shorter and easier to distinguish. As shown in Figure 6.4, the BER of N-Pulse increases rapidly with distance, whereas DOOK maintains a zero BER over a longer range. At larger distances, scattering and other multipath reflections increase, which makes it more difficult to distinguish the four N-Pulse symbols reliably. By contrast, DOOK uses only two symbols on the NLoS link. In addition, the event polarity changes between these symbols, which helps with decoding at larger distances. DOOK also achieves 8× higher throughput than N-Pulse, because N-Pulse encodes only 2 bits every 16 frames, whereas DOOK encodes 1 bit in every frame.

This contradicts the findings of Nishar, Marefat and Ashok, who conclude that N-Pulse performs better than OOK when transmitting regular data [15].

DOOK + Manchester encoding outperforms DOOK because every symbol contains a transition. DOOK + Manchester encoding achieves a lower BER than regular DOOK. The reason is that each symbol contains a transition, which is exactly the type of signal an event camera responds to. In regular DOOK, repeated bits generate fewer events, which leaves more room for noise to interfere with decoding. With Manchester encoding, every symbol produces a transition and therefore generates many events, which can be detected more reliably. This improvement comes at the cost of halving the throughput on both the LoS and NLoS links.

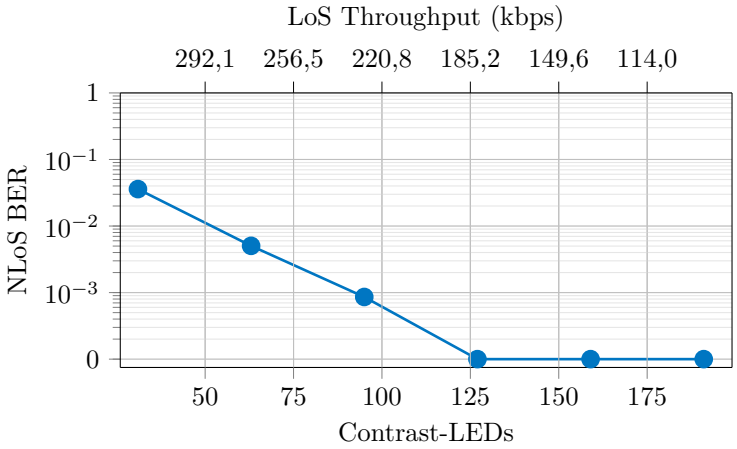


Figure 6.5: **The trade-off between the NLoS BER and the LoS throughput. Increasing the LoS throughput increases the NLoS BER.**

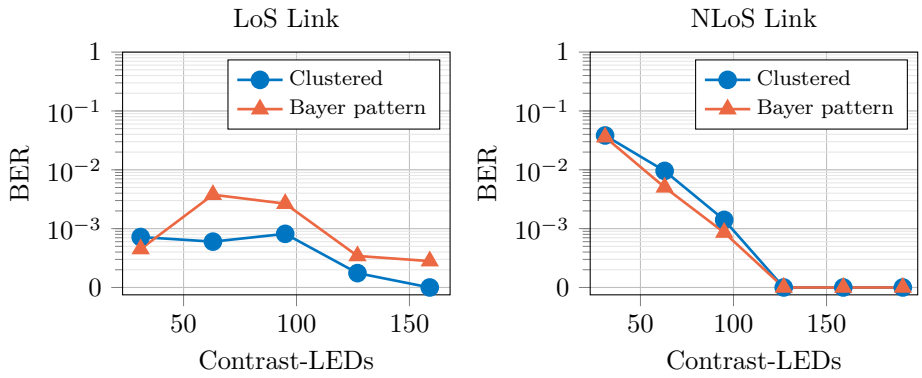
6.3 Parameter analysis

This section examines how key system parameters affect error and throughput. We first consider the trade-off between NLoS BER and LoS throughput in Section 6.3.1. We then compare clustered contrast-LEDs with the Bayer pattern in Section 6.3.2. Next, Section 6.3.3 evaluates the effect of the reflecting object in the NLoS setup. Finally, Section 6.3.4 examines the effect of t_{slot} on BER.

6.3.1 Trade-off between LoS throughput and NLoS error

As discussed in Chapter 4, increasing the number of contrast-LEDs should improve NLoS decoding. Figure 6.5 confirms this prediction. When we assign more LEDs as contrast-LEDs, the difference between dark and bright frames increases. This makes the NLoS transitions easier to detect and reduces the BER.

Decreasing the NLoS error directly decreases the LoS throughput. At the same time, Figure 6.5 shows that this improvement comes at the cost of LoS throughput. When more LEDs are assigned as contrast-LEDs, fewer remain available as data-LEDs. As a result, the LoS throughput decreases. Applying DOOK therefore requires a balance between low NLoS error and high LoS throughput.



(a) **The LoS link performs better when all contrast-LEDs are clustered together rather than distributed across the frame.**

(b) **There is no difference in BER between the contrast patterns on the NLoS link.**

Figure 6.6: **The effect of the contrast pattern and number of contrast LEDs on BER.**

6.3.2 Clustered contrast LEDs versus Bayer pattern

This subsection compares two ways of placing the contrast-LEDs: clustering them together and distributing them according to the Bayer pattern. The corresponding patterns are shown in Figure 4.5 in Chapter 4.

Distributing contrast-LEDs can trigger false events on neighbouring data-LEDs in the LoS link. Figure 6.6a shows that the LoS BER is higher when the contrast-LEDs are distributed using the Bayer pattern than when they are clustered. A likely reason is that light from a contrast-LED can trigger events in neighbouring data-LED pixels. For example, a data-LED may remain off for two consecutive frames, while nearby contrast-LEDs switch on in the second frame. This can generate events at the data-LED location, leading to decoding errors. Clustering the contrast-LEDs separates them from the data-LEDs and therefore reduces this effect.

By contrast, Figure 6.6b shows that the distribution pattern has little effect on the NLoS BER.

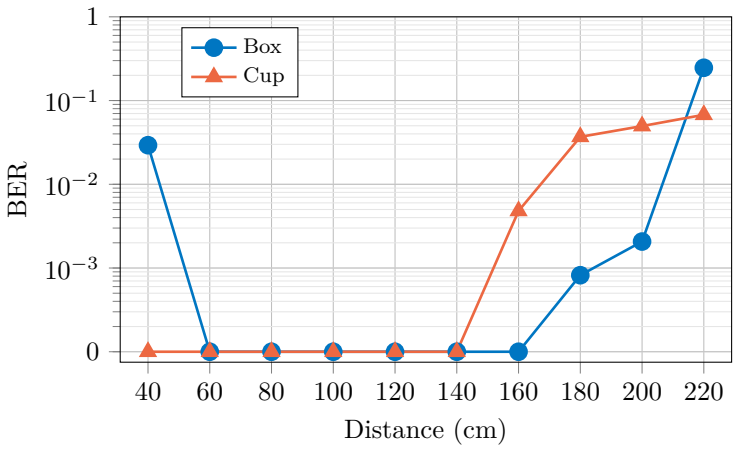


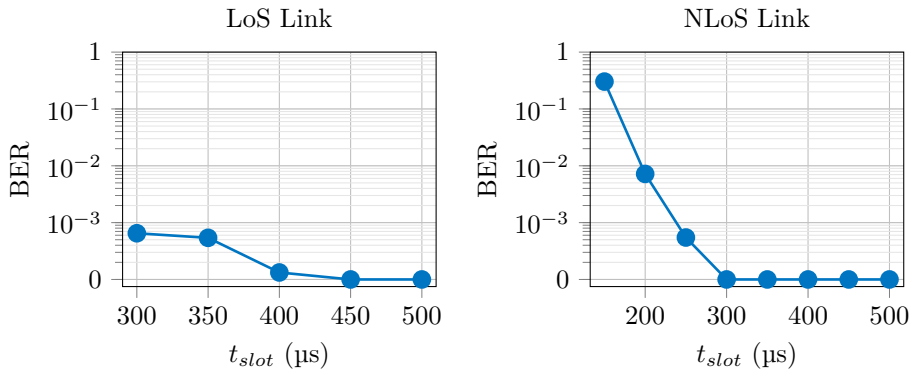
Figure 6.7: **The effect of different objects as a reflective surface at different distances.**

6.3.3 Effect of the material and size of the reflecting object

In this experiment, we compare the two reflecting objects shown in Figure 6.2 over multiple distances. We use BER as the performance metric. The box is larger and has a matte finish, whereas the cup is smaller and has a glossy finish. Figure 6.7 shows the results.

The box performs better than the cup over most distances. A likely reason is that its larger surface reflects more light towards the receiver. Although Nishar, Marefat and Ashok report that glossy objects can outperform matte objects because they introduce less multipath reflection [15], in our setup the effect of object size dominates this difference in material.

At the shortest distance, however, the box produces a high error rate. In this case, the box fills most of the camera's field of view and therefore generates a large number of events. This saturates the event-rate capacity, leading to dropped events, with 55% of events being lost. One way to mitigate this effect is to limit the size of the Region of Interest.



(a) **The LoS link performs better when each frame is displayed longer.** (b) **The effect of different refresh rates on BER.**

Figure 6.8: **Effect of frame duration on BER for NLoS and LoS links.**

6.3.4 Effect of t_{slot} on BER

The choice of t_{slot} affects the event rate and therefore the decoding error. As discussed in Chapter 5, an event camera does not assign timestamps at the exact moment the light intensity changes at a pixel. Instead, timestamps are assigned later when the camera MCU processes the events [6, 43]. When many events are generated in a short time, a queue forms and some timestamps are delayed.

Figure 6.8 shows that a shorter t_{slot} increases the BER on both links. A shorter slot means that more frames are transmitted per second and therefore more events are generated per second. This increases the queue length and, in turn, the timestamp delays. These delays can desynchronise the receiver and introduce decoding errors.

Selene addresses this problem using Event Relative Time decoding [6]. However, we cannot apply this method directly to DOOK, because Event Relative Time decoding assumes that the channels are independent. In DOOK, by contrast, all channels within one frame together represent a single message and therefore cannot be sampled independently.

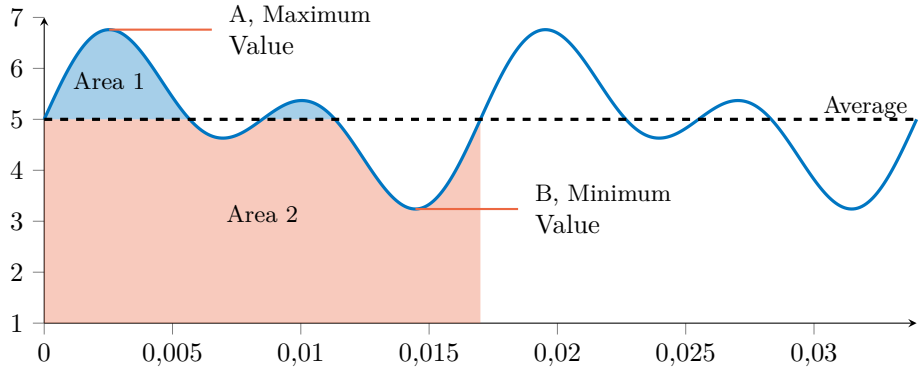


Figure 6.9: **Diagram for definition of Flicker Index and Percent Flicker [19].**

6.4 Flicker

This section evaluates flicker in our system. Reducing flicker is important when VLC provides both communication and illumination, because flicker may introduce health risks [19]. To the best of our knowledge, no prior work on event-camera VLC has analysed or evaluated flicker.

We begin in Section 6.4.1 by defining the metrics used in this section. We then describe the measurement device in Section 6.4.2. Next, Section 6.4.3 compares flicker across modulation schemes. Finally, Section 6.4.4 examines the effect of the number of contrast-LEDs.

6.4.1 Flicker metrics

To quantify flicker, we use two established metrics: Flicker Index and Percent Flicker.

The **Flicker Index** is defined as the area above the average-light line and below the waveform, divided by the total area below the waveform. Equation (6.1) defines this metric with reference to Figure 6.9 [19, 45, 46].

$$\text{Flicker Index} = \frac{\text{Area 1}}{\text{Area 1} + \text{Area 2}} \tag{6.1}$$

The **Percent Flicker**, also referred to as Peak-to-Peak Contrast, Michelson Contrast, or Modulation (%), depends on the minimum and maximum of the waveform. Equation (6.2) defines this metric with reference to Figure 6.9 [19, 46].

$$\text{Percent Flicker} = 100 \cdot \frac{\text{Max} - \text{Min}}{\text{Max} + \text{Min}} \tag{6.2}$$

The IEEE 1789-2015 standard provides a risk assessment for flicker in LED lighting and recommends limits on Percent Flicker as a function of frequency [19, 20]. These limits are represented in a modulation-versus-frequency diagram, shown in Figure 6.10, which marks different risk zones.

In the standard representation, each transmitter/modulation scheme produces a single point on the diagram by combining its dominant frequency with

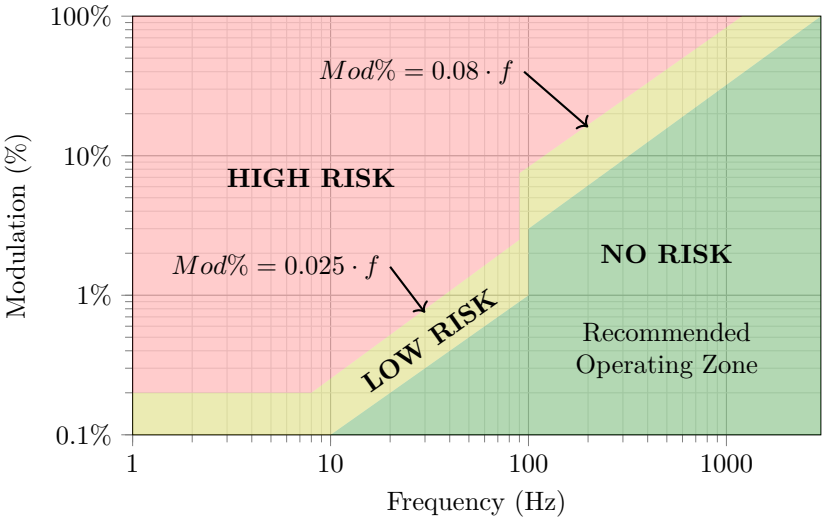


Figure 6.10: **IEEE 1789-2015 risk zones [19, 20].**

its Percent Flicker. We cannot use this representation directly to compare our modulation schemes. The reason is that our measured waveforms share the same minimum and maximum, so Equation (6.2) produces the same Percent Flicker for all of them. As a result, each experiment would map to the same point, even though the waveforms differ strongly in how flicker is distributed across frequencies.

We therefore introduce a variant that computes Percent Flicker in the frequency domain. Instead of analysing the waveform as a whole, we analyse it per frequency range. The procedure is shown in Figure 6.11. We decompose the waveform into 500 logarithmically distributed frequency bands using a Fast Fourier Transform and isolate the component in each band using an inverse Fast Fourier Transform. For each isolated component, we compute the Percent Flicker. Finally, we plot the resulting 500 points on the IEEE-style diagram and apply smoothing. This produces a curve that shows how strongly the waveform flickers across frequencies.

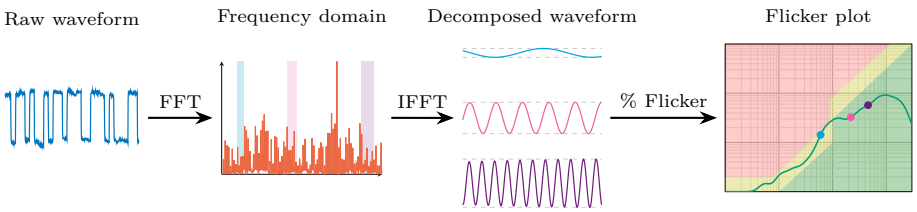


Figure 6.11: **Pipeline used to construct the IEEE-style flicker plot from the measured waveform.**

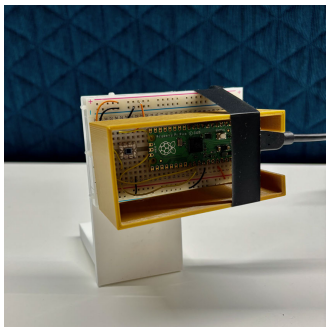


Figure 6.12: **Flicker measurement device using an OPT101.**

6.4.2 Flicker measurement device

To capture waveforms for flicker analysis, we build a low-cost measurement device. In doing so, we take inspiration from Perumal and Baharum, who use a low-cost sensor to analyse flicker with respect to the IEEE diagram [20].

Our device uses an OPT101 photodiode connected to a Raspberry Pi Pico. We mount the photodiode vertically and shield it from ambient light using a shade cover, as shown in Figure 6.12. The Pico samples the OPT101 at 50 kHz for 2 seconds. During the measurement, we place the sensor 100 cm from the transmitter and point it directly at the source. The transmitter sends regular data, and we record the resulting light intensity waveform. We then use this waveform in the analysis described above.

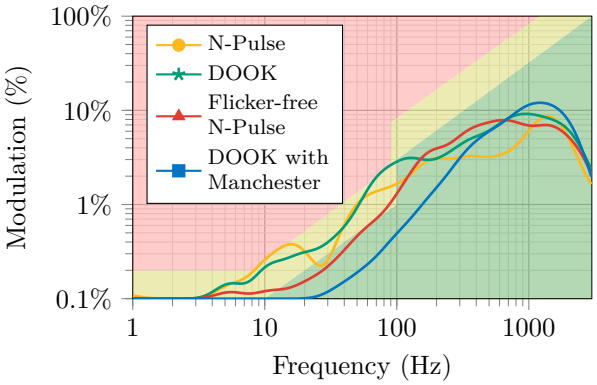


Figure 6.13: **Effect of different modulation schemes on Percent Flicker across frequency bands.**

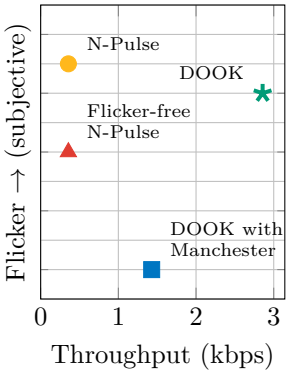


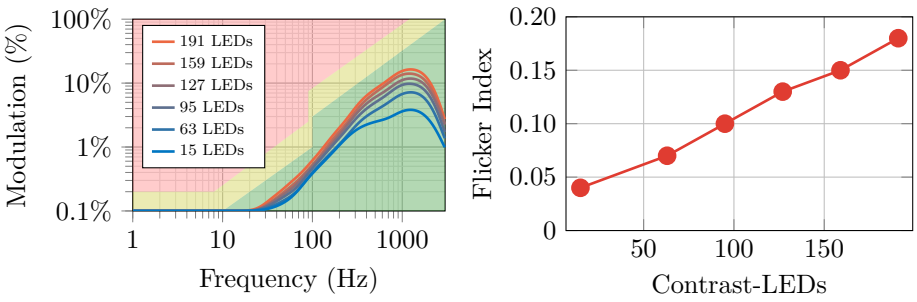
Figure 6.14: **The NLoS throughput versus flicker (subjective).**

6.4.3 Flicker of different modulation schemes

We now compare the flicker-free variants with their corresponding baselines. For each scheme, we record a waveform and analyse its flicker using the method described above. All measurements use a t_{slot} of 350 µs. Figure 6.13 shows the resulting Percent Flicker across frequency for the four DOOK variants introduced in Section 4.2: DOOK, DOOK + Manchester encoding, DOOK + N-Pulse, and DOOK + flicker-free N-Pulse.

The flicker-free variants reduce flicker, with DOOK + Manchester encoding performing best. Figure 6.13 shows that regular DOOK and DOOK + N-Pulse produce the strongest flicker. In both cases, the waveform can vary strongly depending on the transmitted data. DOOK + flicker-free N-Pulse reduces this effect by fixing the duty cycle per low-fidelity symbol, although flicker can still be reintroduced by the high-fidelity stream. DOOK + Manchester encoding produces the least flicker. In that scheme, each LED turns on and off exactly once every two frames, independent of both the high-fidelity and low-fidelity data.

Figure 6.14 summarises the trade-off between NLoS throughput and flicker. To obtain this overview, we assign each modulation scheme a subjective flicker score based on Figure 6.13 and plot it against throughput. DOOK + N-Pulse is dominated by the other schemes. DOOK + flicker-free N-Pulse preserves the throughput of DOOK + N-Pulse while reducing flicker. DOOK + Manchester encoding outperforms both N-Pulse variants in this trade-off. Taken together, regular DOOK and DOOK + Manchester encoding form the main trade-off between higher throughput and lower flicker.



(a) **Effect on Percent Flicker across frequency bands.**

(b) **Effect on Flicker Index.**

Figure 6.15: **Effect of the number of contrast-LEDs on flicker metrics.**

6.4.4 The effect of the contrast-LEDs on flicker

This subsection examines how the number of contrast-LEDs affects flicker. This question is important because the same contrast-LEDs that improve NLoS decoding also affect the amount of flicker.

Assigning more LEDs as contrast-LEDs decreases NLoS error, but increases flicker. As shown earlier in Section 6.3.1, adding contrast-LEDs reduces the NLoS error. However, Figure 6.15 shows that this comes at the cost of increased flicker. Both Percent Flicker, shown in Figure 6.15a, and Flicker Index, shown in Figure 6.15b, increase as more LEDs are assigned as contrast-LEDs. The reason is that additional contrast-LEDs increase the brightness difference between dark and bright frames, leading to stronger waveform fluctuations.

Chapter 7

Limitations and Future Work

This thesis takes a first step towards combined high-fidelity LoS and low-fidelity NLoS communication using an LED matrix and an event camera. Although the proposed system works in a controlled setup, several limitations should be noted. The most important issues concern robustness against loss of synchronisation, robustness to motion and transmitter tracking, and automated camera configuration. A further point concerns the interpretation of the flicker measurements and the extent to which these support safety claims.

7.1 Robustness against getting out of sync

A limitation of the current system is that the receiver can drift out of sync over time. Timing drift can shift the estimated phase offset, causing the receiver to sample the matrix during the transient state rather than during the stable state. Likewise, changes in ambient light can alter the number of events generated per symbol, thereby invalidating the thresholds used in the NLoS pipelines. Because the Sync stage currently runs only once before decoding starts, the link may fail when these conditions change.

A natural progression of this work is to run synchronisation continuously in parallel with decoding. This would allow the receiver to update the phase offset and thresholds whenever drift is detected. Such an approach is feasible because the receiver does not require a dedicated training signal from the transmitter. Instead, it can estimate these parameters from regular traffic, provided that each symbol appears sufficiently often. Under these conditions, the data stream should contain enough information to re-estimate the phase offsets and thresholds described in Chapter 5.

7.2 Robustness to motion and transmitter tracking

The current decoding pipelines assume that the transmitter remains fixed relative to the receiver after the Detect stage. In a dynamic setting, however, such as

vehicle VLC, motion can shift the LED positions or the NLoS reflection region [9]. This may cause the receiver to select the wrong events for decoding, even when synchronisation itself remains correct.

Further work is therefore needed to improve robustness to motion. Previous work has addressed this problem by integrating more trackable modulation schemes. For example, Soga et al. use Walsh–Hadamard codes to track and separate channels in mobile settings [9]. These codes have low cross-correlation and are therefore well suited to tracking under motion [9, 47]. An important next step is to determine how such trackable codes interact with DOOK, and how they affect the trade-off between the LoS and NLoS links.

7.3 Automated camera configuration and link-type selection

At present, switching between LoS and NLoS setups, or changing the camera position in the LoS setup, requires manual adjustment of the camera parameters. Some of these parameters can be set in software, such as the event threshold. Others, however, require physical adjustment, most notably focus and aperture. This is particularly important for the LoS link. If the image is out of focus, light from one LED may spill into neighbouring LEDs. If the aperture is too large, the event rate may become too high, leading to dropped or delayed events. If it is too small, relevant events may be missed.

Future work should therefore investigate automated camera configuration. A lens with programmatic control of focus and aperture would enable automatic tuning methods such as those described by Ge et al. [48]. In addition, the receiver should determine whether it is observing a LoS or NLoS link and select the appropriate parameter set accordingly. This would require a more advanced detection method than the one used in the current system.

7.4 Limits of the flicker evaluation and safety claims

The flicker evaluation in this thesis supports comparisons between modulation schemes and parameter settings. However, it does not directly establish whether the transmitter is safe to view. For example, even when DOOK with Manchester encoding remains in the *NO RISK* region of Figure 6.13, this cannot be taken as proof that no risk is present. One reason is that the evaluation uses a low-cost, uncalibrated sensor. In addition, the measured curve can be affected by the measurement conditions, such as distance and ambient light.

Further research is needed to relate system settings to human perception more directly. One possible next step is to conduct an empirical study similar to that of Wu et al. [49]. Participants could view the transmitter or its reflection and report the amount of flicker they perceived. Such a study would also need to account for the transmitter's high intensity and the associated safety constraints.

Chapter 8

Conclusions

This thesis set out to examine whether a single transmitter can simultaneously support high-fidelity LoS and low-fidelity NLoS communication using an event camera receiver. To address this question, we built a high-speed LED-matrix platform and designed an end-to-end communication system around it.

A central contribution of this thesis is the introduction of Dual On-Off Keying (DOOK), a multi-fidelity modulation scheme. DOOK encodes high-fidelity data in the spatial and temporal dimensions, and low-fidelity data in the temporal dimension via the average brightness of consecutive frames. In addition, this thesis evaluates both baseline low-fidelity modulation schemes against the SoA, as well as flicker-free variants.

The findings of this study show that reliable communication on both links is possible in the tested setup. With DOOK, we achieve 366 kbps on the LoS link and 2,9 kbps on the NLoS link with a BER below 10^{-3}. DOOK + Manchester encoding halves the throughput and produces the least flicker among the evaluated schemes. Another important finding is that, on the NLoS link, DOOK (similar to OOK) outperforms N-Pulse in both BER and throughput in our setup. This contrasts with the conclusion reported by Nishar, Marefat and Ashok [15].

Compared with the SoA, our NLoS throughput is 1,7× higher than that reported by Nishar, Marefat and Ashok [15]. For the LoS link, our throughput per channel is 1,8× higher than that of Selene [6]. However, Selene achieves a higher total throughput because it uses many more channels [6].

Taken together, these findings show that combined LoS/NLoS event-camera VLC with a single transmitter is feasible. They also highlight the central trade-offs between throughput, error, and flicker that determine practical system performance. Several questions remain, however, about how robust such a system can be in less controlled settings. Future work should therefore focus in particular on robustness to drift, motion, and changing camera conditions.

Bibliography

[1] Mohamed Ali El-Moghazi and Jason Whalley, 'World Radiocommunication Conference-19', en, in *The International Radio Regulations*, Cham: Springer International Publishing, 2021, pp. 181–203, DOI: 10.1007/978-3-030-88571-7_10.

[2] Fayzatul Ashmera Binti Merdan, Siva Priya Thiagarajah and Katrina Dambul, 'Non-line of sight visible light communications: A technical and application based survey', *Optik*, vol. 259, p. 168 982, Jun. 2022, DOI: 10.1016/j.ijleo.2022.168982.

[3] Minhao Cui, Yuda Feng, Qing Wang and Jie Xiong, 'Sniffing visible light communication through walls', in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '20, New York, NY, USA: Association for Computing Machinery, Sep. 2020, pp. 1–14, DOI: 10.1145/3372224.3419187.

[4] Patrick Lichtsteiner, Christoph Posch and Tobi Delbruck, 'A 128 x 128 120 dB 15 us Latency Asynchronous Temporal Contrast Vision Sensor', *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb. 2008, DOI: 10.1109/JSSC.2007.914337.

[5] Prophesee S.A., *EVK4-HD Prophesee Evaluation Kit: Camera Manual*, Version 1.1. Prophesee, Jan. 2023, [Online]. Available: https://www.prophesee.ai/wp-content/uploads/2023/01/EVK4-HD-Prophesee-Evaluation-Kit-Camera-Manual-1.1.pdf.

[6] Yanxiang Wang, Yiran Shen, Kenuo Xu, Mahbub Hassan, Guangrong Zhao, Chenren Xu and Wen Hu, 'Towards High-Speed Passive Visible Light Communication with Event Cameras and Digital Micro-Mirrors', in *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '24, New York, NY, USA: Association for Computing Machinery, Nov. 2024, pp. 704–717, DOI: 10.1145/3666025.3699368.

[7] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis and Davide Scaramuzza, 'Event-Based Vision: A Survey', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, Jan. 2022, DOI: 10.1109/TPAMI.2020.3008413.

[8] Takaya Yamazato, Isamu Takai, Hiraku Okada, Toshiaki Fujii, Tomohiro Yendo, Shintaro Arai, Michinori Andoh, Tomohisa Harada, Keita Yasutomi, Keiichiro Kagawa and Shoji Kawahito, 'Image-sensor-based visible light communication for automotive applications', *IEEE Communications Magazine*, vol. 52, no. 7, pp. 88–97, Jul. 2014, DOI: 10.1109/MCOM.2014.6852088.

[9] Ryota Soga, Shintaro Shiba, Quan Kong, Norimasa Kobori, Tsukasa Shimizu, Shan Lu and Takaya Yamazato, *Evaluation of Mobile Environment for Vehicular Visible Light Communication Using Multiple LEDs and Event Cameras*, arXiv:2505.15412 [cs], May 2025, DOI: 10.48550/arXiv.2505.15412.

[10] Svilen Dimitrov and Harald Haas, *Principles of LED Light Communications: Towards Networked Li-Fi*. Cambridge: Cambridge University Press, 2015, DOI: 10.1017/CBO9781107278929.

[11] Wen-Hsuan Shen, Po-Wen Chen and Hsin-Mu Tsai, 'Vehicular Visible Light Communication with Dynamic Vision Sensor: A Preliminary Study', in *2018 IEEE Vehicular Networking Conference (VNC)*, Dec. 2018, pp. 1–8, DOI: 10.1109/VNC.2018.8628425.

[12] Jaime Aranda, Victor Guerra, Jose Rabadan and Rafael Perez-Jimenez, 'Enhancing Computational Efficiency in Event-Based Optical Camera Communication Using N-Pulse Modulation', en, *Electronics*, vol. 13, no. 6, p. 1047, Jan. 2024, DOI: 10.3390/electronics13061047.

[13] Hang Su, Ling Gao, Tao Liu and Laurent Kneip, 'Motion-Aware Optical Camera Communication with Event Cameras', *IEEE Robotics and Automation Letters*, vol. 10, no. 2, pp. 1385–1392, Feb. 2025, arXiv:2412.00816 [cs], DOI: 10.1109/LRA.2024.3517292.

[14] Abbaas Alif Mohamed Nishar, Sonipriya Paul and Ashwin Ashok, 'Poster Abstract: Joint Optical Wireless Communication and Sensing using Neuromorphic Cameras', in *2024 23rd ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, May 2024, pp. 305–306, DOI: 10.1109/IPSN61024.2024.00053.

[15] Abbaas Alif Mohamed Nishar, Alireza Marefat and Ashwin Ashok, *Non Line-of-Sight Optical Wireless Communication using Neuromorphic Cameras*, arXiv:2503.11226 [cs], Mar. 2025, DOI: 10.48550/arXiv.2503.11226.

[16] Alin-Mihai Căilean and Mihai Dimian, 'Current Challenges for Visible Light Communications Usage in Vehicle Applications: A Survey', *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2681–2703, 2017, DOI: 10.1109/COMST.2017.2706940.

[17] T. Komine and M. Nakagawa, 'Fundamental analysis for visible-light communication system using LED lights', *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 100–107, Feb. 2004, DOI: 10.1109/TCE.2004.1277847.

[18] Carlos E. Mejia, Costas N. Georghiades, Mohamed M. Abdallah and Yazan H. Al-Badarneh, 'Code Design for Flicker Mitigation in Visible Light Communications Using Finite State Machines', *IEEE Transactions on Communications*, vol. 65, no. 5, pp. 2091–2100, May 2017, DOI: 10.1109/TCOMM.2017.2657518.

[19] 'IEEE Recommended Practices for Modulating Current in High-Brightness LEDs for Mitigating Health Risks to Viewers', *IEEE Std 1789-2015*, pp. 1–80, Jun. 2015, DOI: 10.1109/IEEESTD.2015.7118618.

[20] Sivachandran R. Perumal and Faizal Baharum, 'Measurement, Simulation, and Quantification of Lighting-Space Flicker Risk Levels Using Low-Cost TCS34725 Colour Sensor and IEEE 1789-2015 Standard', en, *Journal of Daylighting*, vol. 8, no. 2, pp. 239–254, Aug. 2021, DOI: 10.15627/jd.2021.19.

[21] Stefano Caputo, Lorenzo Mucchi, Regina Comparetto, Vittoria D'Antoni, Alessandro Farini, Valentina Orsi and Elisabetta Baldanzi, 'Exploring the Effects of LED-Based Visible Light Communication on Reading and Color Perception in Indoor Environments: An Experimental Study', *Sensors (Basel, Switzerland)*, vol. 23, no. 6, p. 2949, Mar. 2023, DOI: 10.3390/s23062949.

[22] 'IEEE Standards for Local Area Networks: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications', *ANSI/IEEE Std 802.3-1985*, 0_1–, 1985, DOI: 10.1109/IEEESTD.1985.82837.

[23] Koji Kamakura, 'Image Sensors Meet LEDs', *IEICE Transactions on Communications*, vol. E100.B, no. 6, pp. 917–925, 2017, DOI: 10.1587/transcom.2016LCI0001.

[24] Shintaro Arai, Zhengqiang Tang, Akinori Nakayama, Haruhiro Takata and Tomohiro Yendo, 'Rotary LED Transmitter for Improving Data Transmission Rate of Image Sensor Communication', *IEEE Photonics Journal*, vol. 13, no. 4, pp. 1–11, Aug. 2021, DOI: 10.1109/JPHOT.2021.3097772.

[25] Yuki Ohira, Tomohiro Yendo, Shintaro Arai and Takaya Yamazato, 'High performance demodulation method with less complexity for image-sensor communication', EN, *Optics Express*, vol. 27, no. 15, pp. 21 565–21 578, Jul. 2019, DOI: 10.1364/OE.27.021565.

[26] *Event-Based Camera vs Standard Camera*, Apr. 2016, [Online]. Available: https://youtu.be/kPCZESVfHoQ.

[27] Haruyuki Nakagawa, Yoshitaka Miyatani and Asako Kanezaki, 'Linking Vision and Multi-Agent Communication through Visible Light Communication using Event Cameras', in *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '24, Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2024, pp. 1436–1444, ISBN: 979-8-4007-0486-4.

[28] Parth H. Pathak, Xiaotao Feng, Pengfei Hu and Prasant Mohapatra, 'Visible Light Communication, Networking, and Sensing: A Survey, Potential and Challenges', *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2047–2077, 2015, DOI: 10.1109/COMST.2015.2476474.

[29] Christos Danakis, Mostafa Afgani, Gordon Povey, Ian Underwood and Harald Haas, 'Using a CMOS camera sensor for visible light communication', in *2012 IEEE Globecom Workshops*, Dec. 2012, pp. 1244–1248, DOI: 10.1109/GLOCOMW.2012.6477759.

[30] 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*, ser. MobiSys '15, New York, NY, USA: Association for Computing Machinery, 2015, pp. 167–180, DOI: 10.1145/2742647.2742651.

[31] Pinpin Zhang, Ziwei Liu, Xin Hu, Yimao Sun, Xiong Deng, Binbin Zhu and Yanbing Yang, 'Constraints and Recent Solutions of Optical Camera Communication for Practical Applications', en, *Photonics*, vol. 10, no. 6, p. 608, Jun. 2023, DOI: 10.3390/photonics10060608.

[32] Wei-Chung Wang, Chi-Wai Chow, Liang-Yu Wei, Yang Liu and Chien-Hung Yeh, 'Long distance non-line-of-sight (NLOS) visible light signal detection based on rolling-shutter-patterning of mobile-phone camera', EN, *Optics Express*, vol. 25, no. 9, pp. 10 103–10 108, May 2017, DOI: 10.1364/OE.25.010103.

[33] Jenn-Kaie Lain, Fu-Cheng Jhan and Zheng-Dao Yang, 'Non-Line-of-Sight Optical Camera Communication in a Heterogeneous Reflective Background', *IEEE Photonics Journal*, vol. 11, no. 1, pp. 1–8, Feb. 2019, DOI: 10.1109/JPHOT.2019.2892966.

[34] Shintaro Arai, Shohei Mase, Takaya Yamazato, Tomohiro Endo, Toshiaki Fujii, Masayuki Tanimoto, Kiyosumi Kidono, Yoshikatsu Kimura and Yoshiki Ninomiya, 'Experiment on Hierarchical Transmission Scheme for Visible Light Communication using LED Traffic Light and High-Speed Camera', in *2007 IEEE 66th Vehicular Technology Conference*, Sep. 2007, pp. 2174–2178, DOI: 10.1109/VETECF.2007.456.

[35] Sayaka Nishimoto, Takaya Yamazato, Hiraku Okada, Toshiaki Fujii, Tomohiro Yendo and Shintaro Arai, 'High-speed transmission of overlay coding for road-to-vehicle visible light communication using LED array and high-speed camera', in *2012 IEEE Globecom Workshops*, Dec. 2012, pp. 1234–1238, DOI: 10.1109/GLOCOMW.2012.6477757.

[36] Katsunori Ebihara, Koji Kamakura and Takaya Yamazato, 'Layered Transmission of Space-Time Coded Signals for Image-Sensor-Based Visible Light Communications', *Journal of Lightwave Technology*, vol. 33, no. 20, pp. 4193–4206, Oct. 2015, DOI: 10.1109/JLT.2015.2470091.

[37] Worldsemi, *WS2812B: Intelligent Control LED Integrated Light Source (Datasheet)*. Worldsemi, [Online]. Available: https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf.

[38] Prathyusha Narra and D.S. Zinger, 'An effective LED dimming approach', in *Conference Record of the 2004 IEEE Industry Applications Conference, 2004. 39th IAS Annual Meeting.*, ISSN: 0197-2618, vol. 3, Oct. 2004, 1671–1676 vol.3, DOI: 10.1109/IAS.2004.1348695.

[39] Texas Instruments, *TLC59282: 16-Channel, Constant-Current LED Driver with 4-Channel Grouped Delay (Datasheet, Rev. C)*. Texas Instruments, Oct. 2011, [Online]. Available: https://www.ti.com/lit/ds/symlink/tlc59282.pdf.

[40] Broadcom, *HSMx-A10x-xxxxx: PLCC-2, Surface-Mount LED Indicator (Data Sheet)*. Broadcom, Aug. 2022, [Online]. Available: https://docs.broadcom.com/doc/HSMx-A10x-xxxxx-PLCC-2-SMT-LED-Indicator-DS.

[41] Bryce E. Bayer, 'An Optimum Method for Two-Level Rendition of Continuous-Tone Pictures', in *1973 International Conference on Communications (Conference Record)*, vol. 1, IEEE, Jun. 1973, pp. 26.11–26.15.

[42] Prophesee S.A., *Metavision SDK Documentation (Stable)*, 2024, [Online]. Available: https://docs.prophesee.ai/stable/index.html.

[43] iniVation AG, 'Understanding the Performance of Neuromorphic Event-based Vision Sensors', iniVation AG, White paper, May 2020, [Online]. Available: https://inivation.com/wp-content/uploads/2020/05/White-Paper-May-2020.pdf.

[44] Zhengqiang Tang, Takaya Yamazato and Shintaro Arai, 'A Preliminary Investigation For Event Camera-Based Visible Light Communication Using The Propeller-type Rotary LED Transmitter', in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2022, pp. 646–650, DOI: 10.1109/ICCWorkshops53468.2022.9814467.

[45] Arthur A Eastman and John H Campbell, 'Stroboscopic and flicker effects from fluorescent lamps', *Illuminating Engineering*, vol. 47, no. 1, pp. 27–35, 1952.

[46] Brad Lehman, Arnold Wilkins, Sam Berman, Michael Poplawski and Naomi Johnson Miller, 'Proposing measures of flicker in the low frequencies for lighting applications', in *2011 IEEE Energy Conversion Congress and Exposition*, ISSN: 2329-3748, Sep. 2011, pp. 2865–2872, DOI: 10.1109/ECCE.2011.6064154.

[47] Daiki Ehara, Zhengqiang Tang, Masayuki Kinoshita, Takaya Yamazato, Hiraku Okada, Koji Kamakura, Shintaro Arai, Tomohiro Yendo and Toshiaki Fujii, 'Influence of Walsh-Hadamard Code Sequency in Visible Light Communication Using an Event Camera', ja, *IEICE Proceedings Series*, vol. 72, no. S9-5, Nov. 2022, ISSN: 2188-5079, [Online]. Available: https://www.ieice.org/publications/proceedings/summary.php?iconf=ICETC&session_num=S9&number=S9-5&year=2022.

[48] Zhou Ge, Haoyu Wei, Feng Xu, Yizhao Gao, Zhiqin Chu, Hayden K. -H. So and Edmund Y. Lam, 'Millisecond autofocusing microscopy using neuromorphic event sensing', *Optics and Lasers in Engineering*, vol. 160, p. 107 247, Jan. 2023, DOI: 10.1016/j.optlaseng.2022.107247.

[49] Hongjia Wu, Qing Wang, Jie Xiong and Marco Zuniga, 'SmartVLC: Co-Designing Smart Lighting and Communication for Visible Light Networks', *IEEE Transactions on Mobile Computing*, vol. 19, no. 8, pp. 1956–1970, Aug. 2020, DOI: 10.1109/TMC.2019.2915220.

Appendix A

AI statement

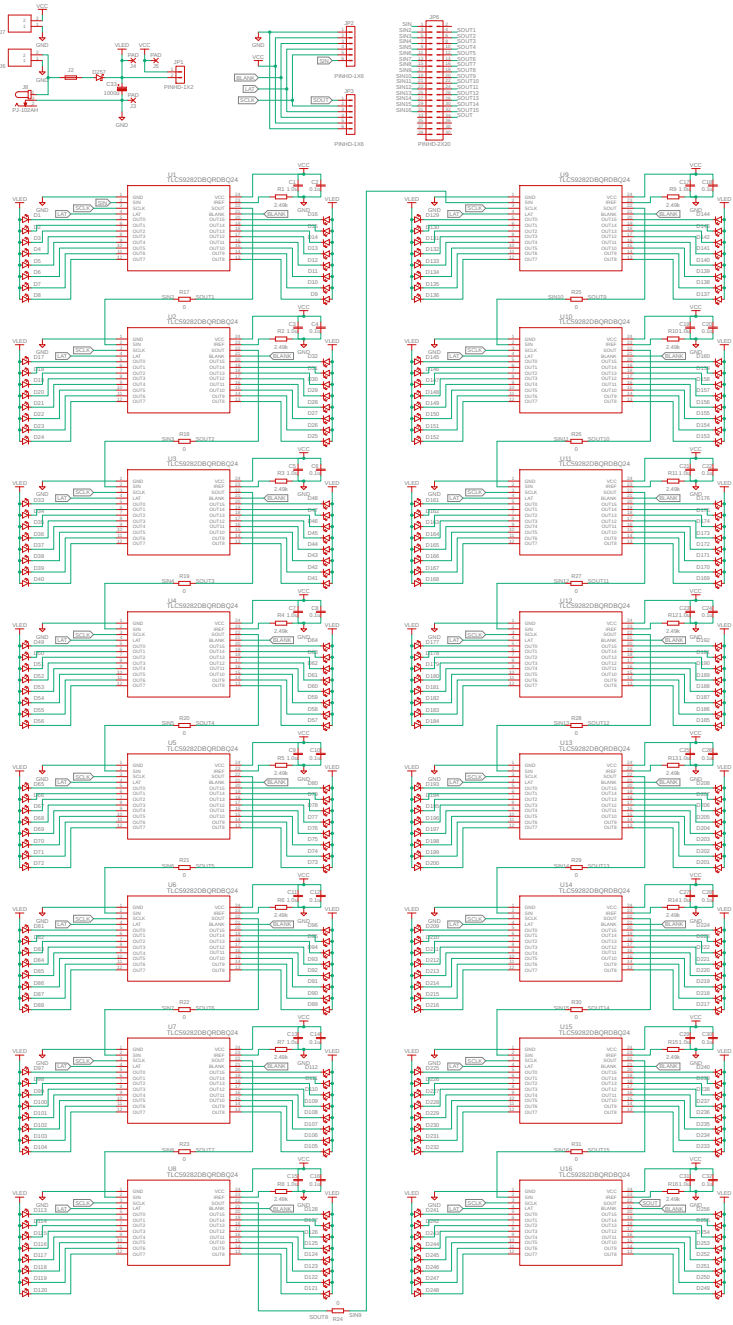
Generative AI tools were used in this thesis for two tasks: drafting code and improving writing clarity. ChatGPT and Codex were used to generate code from low-level prompts, where the intended algorithm and software structure were specified in advance. These tools were also used to help rewrite parts of the text for clarity, conciseness, and flow.

All technical ideas, system design choices, experiments, and conclusions in this thesis are original and were not generated using AI. All AI-generated code and text are verified and integrated only after review. Generative AI is not used to create research contributions, results, or interpretations.

Appendix B

Transmitter Circuit Diagram

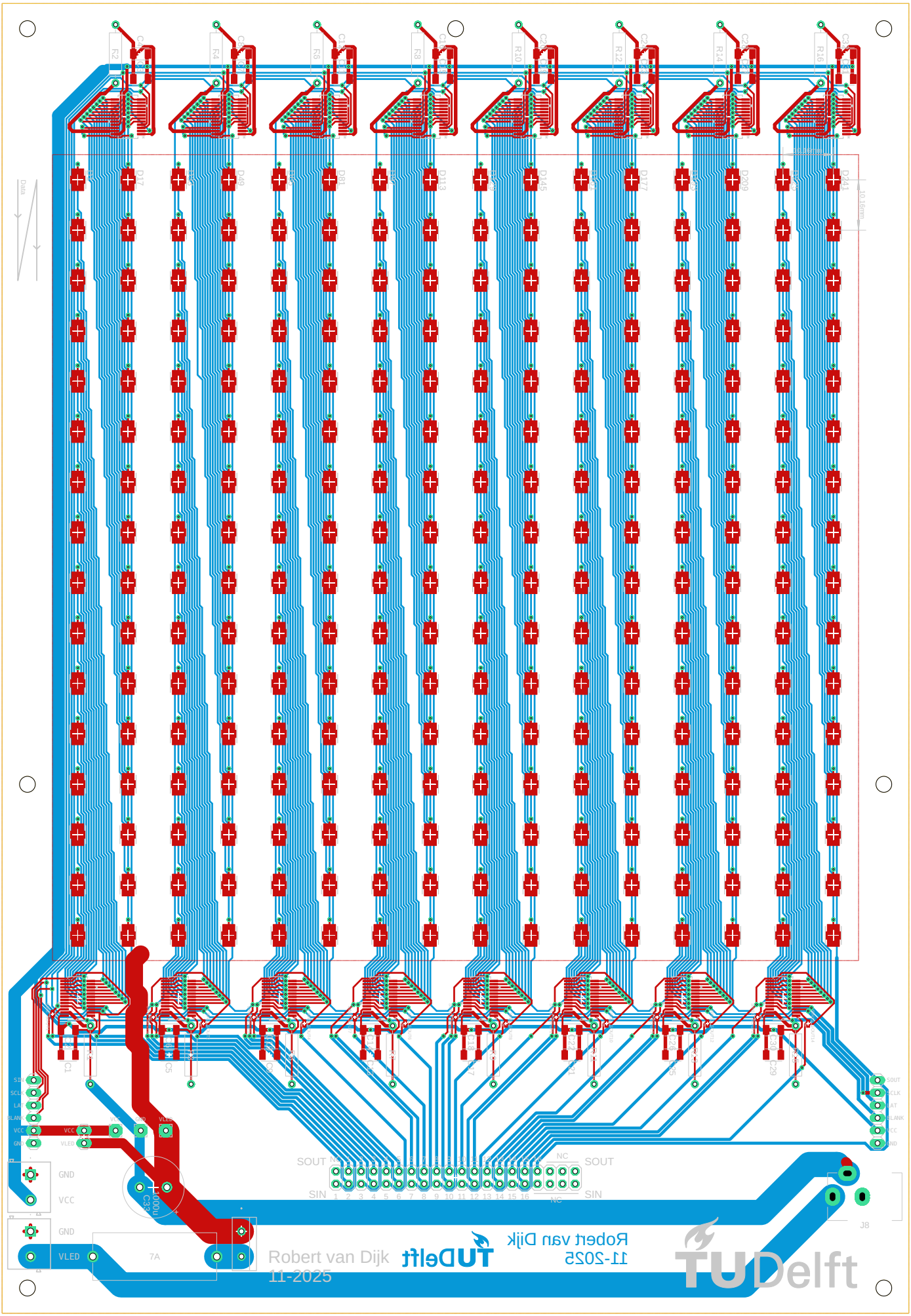
The next page contains the circuit diagram for the transmitter.



Appendix C

Transmitter PCB Design

The next page contains the PCB design for the transmitter at a 1:1 scale.


Data
GND
VCC
GND
VLED
7A
1000u
C33
SOUT
SIN
NC
Robert van Dijk
11-2025
TUDelft

Appendix D

Full comparison with Selene

This appendix presents the results of experiments on our system, compared with those reported in the work by Wang et al. (Selene) [6].

Table D.1 shows the parameters used by Selene and their throughput. It also presents our configurations and their corresponding LoS throughput. We use several configurations for the number of contrast-LEDs, which affects the LoS throughput.

Table D.1: **Comparison of Selene [6] and DOOK**

Implementation	**Refresh rate**	**Symbol period**	**Data channels**	**Throughput**	**BER**	**Throughput / channel**	**Throughput / LED**
Selene with single refresh rate policy	767 Hz	1304 µs	1995	1,53 Mbps	$1 \cdot 10^{-2}$	767 bps	—
Selene with dual refresh rate policy	1191 Hz & 595 Hz	840 µs & 1680 µs	709 & 1286	1,61 Mbps	$1 \cdot 10^{-2}$	807 bps	—
DOOK	2857 Hz	350 µs	64	183 kbps	$< 1 \cdot 10^{-3}$	2857 bps	714 bps
			96	274 kbps			1071 bps
			128	366 kbps			1429 bps
			160	457 kbps			1786 bps
			192	549 kbps			2143 bps
DOOK + Manchester	2857 Hz	700 µs	64	91 kbps	$< 1 \cdot 10^{-3}$	1429 bps	357 bps
			96	137 kbps			536 bps
			128	183 kbps			714 bps
			160	229 kbps			893 bps
			192	274 kbps			1071 bps