Universidade de Aveiro
Departamento de Eletrónica, Telecomunicações e Informática
2019

Miguel Ângelo Roque Soares

Comunicações Ópticas por Câmera para sistemas de assistência à condução

Optical Camera Communication for driver's assistance systems

Universidade de Aveiro
Departamento de Eletrónica, Telecomunicações e Informática
2019

Miguel Ângelo Roque Soares

Comunicações Ópticas por Câmera para sistemas de assistência à condução

Optical Camera Communication for driver's assistance systems

"*Eu não tenho ídolos. Tenho admiração por trabalho, dedicação e competência.*"

— Ayrton Senna

Universidade de Aveiro
2019

Departamento de Eletrónica,
Telecomunicações e Informática

Miguel Ângelo Roque Soares

Comunicações Ópticas por Câmera para sistemas de assistência à condução

Optical Camera Communication for driver's assistance systems

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor Luis Filipe Mesquita Nero Moreira Alves, Professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Texto Apoio financeiro do POCTI no âmbito do III Quadro Comunitário de Apoio.

Texto Apoio financeiro da FCT e do FSE no âmbito do III Quadro Comunitário de Apoio.

Dedico este trabalho à minha familia e amigos pelo incansável apoio.

o júri / the jury

presidente / president

João Pedro Estima de Oliveira
Professor Doutor do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

vogais / examiners committee

Rafael Pérez-Jiménez
Professor Doutor da Universidad de Las Palmas de Gran Canária.

Luis Filipe Mesquita Nero Moreira Alves
Professor Doutor do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

agradecimentos / acknowledgements

Agradeço a ajuda dos meus colegas e companheiros do grupo de investigação de Circuitos Integrados do Instituto de Telecomunicações de Aveiro. À Exatronic, na pessoa do Engenheiro Nuno Gomes por todo o apoio prestado, tanto material como na discussão de soluções a implementar. E por fim, ao meu orientador, o Professor Doutor Luis Nero Alves, por todo o suporte dado desde o primeiro ao último momento da elaboração deste trabalho.

Palavras Chave

VLC, OCC, CNN, luz vísivel, câmera, automóvel, Driver, Processamento de Imagem

Resumo

As comunicações, qualquer que seja o seu tipo, mostram-se como um pilar fundamental para a sociedade. Especificamente as comunicações por luz visível, que apresentam inúmeras vantagens, desde a eficiência espectral e mais liberdade de regulamentação, até à energética pois alia duas caracteristicas distintas (iluminação e comunicação) numa só. Como tal, o mundo automóvel apresenta-se como um dos posíveis interessados na aplicação desta tecnologia, mais propriamente a aplicação como parte integrante do sistema inteligente de transportes (ITS). Este trabalho tem como objectivo o estudo e desenvolvimento de um demonstrador capaz de estabelecer um link de comunicação V2V (Vehicle to vehicle) por meio da modulação da luz visivel emitida pelas iluminárias LED já equipadas actualmente nos veículos. Sendo a implementação exterior um dos requerimentos deste sistema, a rececção através de OCC (Optical Camera Communication) mostra-se assim uma solução viável. Assim como o processamento do sinal recebido, que é efectuado por meio de CNNs (Convolutional Neural Networks), que mostram flexibilidade e resiliência, o que benefecia a capacidade do sistema de transmissão. Todo o projecto foi realizado em colaboração com o grupo de circuitos integrados do Instituto de Telecomunicações de Aveiro e a empresa Exatronic Lda, sediada em Aveiro, e especializada em inovação, investigação (I+I), engenharia e produção de eletrónica.

Keywords VLC, OCC, CNN, visible light, camera, car, Driver, Image Processing.

Abstract

Communications, whatever its type, is a pillar of our modern society. More specifically, communications by visible light, that show numerous advantages, from electromagnetic spectral efficiency and regulation freedom to energy saving (since it combine illumination and communication). As such, the automotive world is interested in this technology, in particularly, its application into the Intelligent Transport System (ITS). The objective of this work relies on the study and development of a demonstrator able to support VLC communication means in V2V (Vehicle to Vehicle) scenario, making use of the LED luminaries already implemented in nowadays cars. Since the outdoor implementation is one of the requirements, reception based in OCC (Optical Camera Communication) is a viable solution in this conditions. Also the signal processing/decoding is performed by a CNN (Convolutional Neural Network), this type of algorithm shows a huge decoding flexibility and resilience, which benefits the transmission system performance. All the project was done in collaboration with the integrated circuits systems group of Instituto de Telecomunicações de Aveiro and Exatronic Lda company, based in Aveiro and specialized in innovation and investigation (I+I), engineering and manufacturing of electronics.

Contents

List of Figures

List of Tables

Acronyms

IoT	Internet of Things
RF	Radio Frequency
IR	Infrared
BW	Bandwith
VLC	Visible Light Communication
OWC	Optical Wireless Communications
ITS	Intelligent Transportation System
V2V	Vehicle to Vehicle
OCC	Optical Camera Communication
LIDAR	Light Detection And Ranging
WLAN	Wireless Local Area Network
LTE	Long Term Evolution
LED	Light Emitting Diode
V2I	Vehicle to Infrastructure
CNN	Convulutional Neural Network
Li-Fi	Light Fidelity
Wi-Fi	Wireless Fidelity
HB-LED	High-Brightness LED
FOV	Field of View
MIMO	Multiple Input Multiple Output
PD	Photo Diode
CMOS	Complementary Metal-Oxide-Semiconductor
DSP	Digital Signal Processor
CPx	Communication Pixel
IPx	Image Pixel
LOS	Line of Sight
ROI	Region of Interest
SNR	Signal to Noise Ratio
PSD	Power Spectral Density
UNECE	United Nations Economic Commission for Europe
CCD	Charge-Coupled Devices
OOK	On-Off Keying
CFF	Critical Flicker-Fusion Frequency
FPS	Frames per Second
DSRC	Dedicated Short Range Communication
TAS	Travel Advisory Systems
TMC	Traffic Management Center
RTOS	Real-Time Operating System
CNN	Convolutional Neural Network
SSD	Single Shot multibox Detector
YOLO	You Only Look Once
SER	Symbol Error Ratio
SIR	Signal to Interference Ratio
UFSOOK	Undersampling Frequency Shift On-Off Keying
UPSOOK	Undersampling Phase Shift On-Off Keying
BLL	Beer-Lambert Law
COCO	Common Objects in Context

CHAPTER 1

Introduction

Today's world is increasingly dependent on telecommunications. The need to exchange information with the surrounding environment is fundamental for us to fulfill many of our daily tasks. Adding to the fact that we are in full implementation of the Internet of Things (IoT), where any device or machine is connected to the worldwide network, shows that telecommunications and connectivity are already "embedded in our DNA". As a result, data traffic and number of connections has increased dramatically, with Radio Frequency (RF) and Infrared (IR) wireless communications being the most prevalent choices, which makes the electromagnetic spectrum increasingly saturated in Bandwith (BW) used by these technologies. Therefore, the search for new solutions able to free the available spectrum and bring improvements in both spectral efficiency, energy and safety are welcome.

Thus appear Visible Light Communication (VLC), presented as a subset of Optical Wireless Communications (OWC), a technology that makes use of the visible light spectrum (400-700 nm), modulating it through variations of radiation intensity. It has a short range when compared to communications such as RF, however, has the advantage of making use of an already implemented infrastructure: lighting [1]. Another advantage of this technology is the fact that is free from telecommunication regulations and available for use at any moment.

Intelligent Transportation System (ITS) is one of the possible applications of VLC. It is a recent technology aimed at increasing road safety as well as the efficiency of car traffic by creating an intelligent network composed by all the traffic users (pedestrians, drivers, autonomous vehicles, among others). The interconnection between the different branches of the network has a fundamental importance, in this particular case, the connection between vehicles. With this kind of link is expected to exchange alerts, advises among other information, useful for the driver in way that helps him to make better choices or avoid dangers situations.

The objective of this work is to create a viable system of communication between cars, known as Vehicle to Vehicle (V2V), making use of homologated rear lights as transmitters and video cameras as receivers. This type of communication is denominated as Optical Camera Communication (OCC) in the literature. The system shall address requirements such as:

link distance, resistance to outdoor light interference and not affect the inherent signaling characteristics of the rear headlamps.

1.1 Motivation

Cars are increasingly evolved, intelligent and independent machines, a proof of this is the fact that any vehicle of this decade has greater computing power than the computers that took the first men to the Moon. Associated with this is the increasing number of vehicles on the roads. This has raised a fundamental need: to interconnect vehicles and enable important information to be exchanged in order to deliver the driver with the most up-to-date data and thus make more timely, safe and comfortable decisions.

Technologies such as Light Detection And Ranging (LIDAR) that make use of light for detecting and measuring obstacles in the environment surrounding the sensor, give the vehicle the ability to "see" and understand the space around it [2].

It is therefore essential to design a wireless communication network between them. Solutions based on RF technologies, such as Wireless Local Area Network (WLAN), Long Term Evolution (LTE), among others, have the advantage of already being properly structured and developed. But the need to find redundant systems that release the RF spectrum lead to the search for new solutions.

The fact that the modern car industry has recently invested in LED technology as a solution for road lighting and signaling has unlocked the way for the study and implementation of VLC/OCC as a communication link between independent vehicles.

Projects such as VIDAS [3], developed at the University of Aveiro, proved the possibility of practical application of this technology (in this case as Vehicle to Infrastructure (V2I)) as an aid to ITS systems [4][5].

1.2 Context

This project comes from the partnership between the Instituto de Telecomunicações de Aveiro and Exatronic Lda, with the objective of developing a real solution for an outdoor vehicle communication scenario using VLC/OCC technology. It is supported by the IT Integrated Circuits research group.

1.3 Methodology

The development process of this dissertation began with a study of the state of the art, projects and articles related to outdoor VLC and OCC. The second stage was based on Matlab simulations in order to better understand the channel behavior, radiation diagrams and other phenomena (reflection, absorption, among others) of light radiation. The third step was the development of the transmitter's electronic driver and the taillight adaptation. The fourth stage was the design of the entire receiving system, from Convolutional Neural Network training, image processing code and interconnections between Raspberry's and graphics

processing unit (PC). The last phase was the performance analysis and testing measurements of the final system.

1.4 Structure

This dissertation is composed of five chapters. Distributed as follows:

- **Chapter Two**: Discusses in more detail the different sectors that involve this technology as well as its historical evolution. It also presents all the simulation work developed for a better study of the communication channel and its behavior.
- **Chapter Three**: This chapter is reserved for the study and presentation of the entire practical system developed for the actual implementation of the VLC/OCC-based communication link. The different subsystems are properly detailed. In addition, some performance evaluations are exhibited and discussed.
- **Chapter Four**: Here the results of the various implementation test done are presented and analyzed. Small conclusions are drawn about the viability of the developed system.
- **Chapter Five**: Comprises the final conclusions and plans for future work.

1.5 Original contributions

This work contributed with research for one paper, that is at this moment in development phase. With the title "Detection of LED Traffic Light with Optical Camera Communication and Convolutional Neural Network" and is expected to be published at "12th IEEE/IET International Symposium on Communication Systems, Networks and Digital Signal Processing-CSNDSP".

CHAPTER 2

VLC systems based on image sensors

VLC is a technology that makes use of visible light Figure 2.1, modulating its intensity to transmit data through free space. Interest in VLC has grown rapidly with the evolution of high power light emitting diodes LED in the visible spectrum. These components have the advantage over light bulbs and fluorescent lamps of having higher bandwidth and lower power consumption for the same amount of lighting produced. In addition, LEDs have a much longer lifespan, and are now the main source of lighting, whether public, at home or in cars [1][6].

The first implementations of VLC technology were based on photodiode reception, depending on the system characteristics (driver bandwidth, LED, among others) high transfer rates can be achieved [7][8]. However, the susceptibility to destructive phenomena (interference from other light sources such as the sun or artificial lighting) means that the results shown are dependent on very restricted and unusual environment conditions. Thus, the study of new reception solutions with greater resilience to these problems motivated the interest in image sensors.

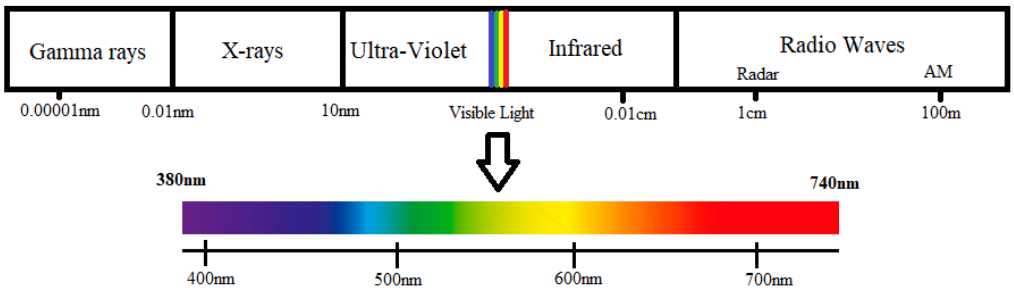


Figure 2.1: Electro Magnetic Spectrum

2.1 History

The earliest evidence of the use of visible light as a communication solution comes from Greco-Roman times, when reports of the use of smoke signals to communicate between watchtowers on Roman walls or the use of light sources such as bonfires, amplified by giant mirrors to communicate with vessels near the coast. This type of technology, although archaic, is still used today, as it is the case with lighthouses distributed around the coastal regions, or the use of smoke signals by the Conclave in Vatican for the announce of the new Pope of the Christian religion.

However the first invention with a greater degree of complexity was shown in 1880 by the hands of Alexander Graham Bell and his assistant Charles Tainter [9]. Named: Photophone, it combined the idea of sunlight modulation for voice signal transmission. The invention was successful and it was possible to transmit at a distance near 200m. The system modulated the light by varying the angle of reflection mirrors due to the vibration imposed by the air pressure differences of the human voice signal (figure 2.2). Despite being a remarkable discovery, it has been overlooked due to advances in cable communications.

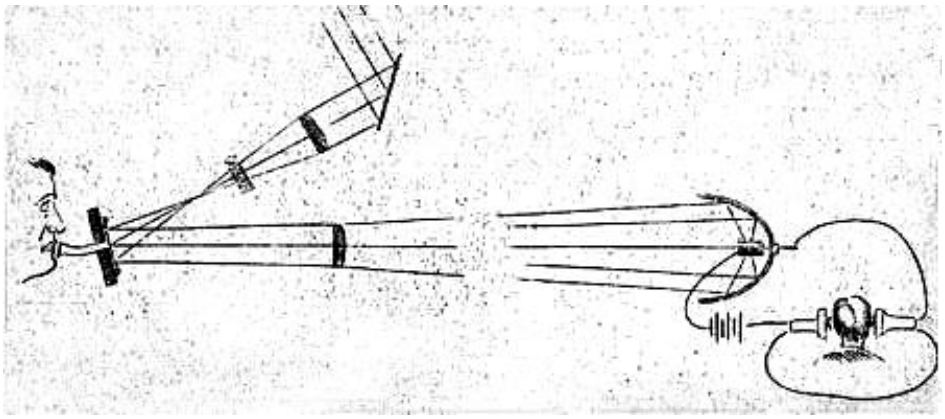


Figure 2.2: Diagram of the Photophone. [10]

In the 1920s, Oleg Vladimirovich Losev, was a laboratory technician in the formed Soviet Union. During a lab experiment, unexpectedly verified the emission of light from rectifier diodes made of zinc oxide and silicon crystals when directly polarized. This was the discovery of the first LED [11].

Nevertheless, it was just in 1962 that the first practical LED emerged. It was created by the hand of the General Electric Company engineer, Nick Holonyak Junior [12] .

The semiconductor built with Gallium Arsenide Phosphide (GaAsP) crystals emitted wavelength radiation in the 645-880nm range, giving it a red color. After that, several improvements were discovered, improving their efficiency and with emission in different spectrum ranges by combining different elements in the semiconductor junction.

Already in the new millennium, in 2003, at Keio University in Japan [13], a system for transmitting data using LED table lamps as transmitter was developed. The set consisted on a Power-line modem that modulated the data from the PC through the power line to the

table lamp. The LED table lamp, then transmitted the data to a photodiode-based receiver using VLC technology. Numerous research activities have been developed in recent years, most notably Light Fidelity (Li-Fi), designed by Harald Haas of the University of Edinbugrh in the United Kingdom [14], with the purpose of be a competitor to Wireless Fidelity (Wi-Fi).

In 2003, the Visible Light Communications Consortium (VLCC), composed by Japanese companies, was formed to share discoveries on VLC technology. This was a key step in affirming VLC in the world of OWC. Another notable achievement was the creation of the IEEE 802.15.7 task group with the purpose of develop compatibility standards for wireless personal networks.

2.2 Optical Camera Communication

Using LED and consequently visible light as a solution for communication is what defines VLC, however the importance of the reception system is significant for the performance of the all setup. Here we find two types: single Photo Diode (PD) receiver and systems that enable multiple signal reception and tracking using an image sensor. This type of VLC is known as OCC or Image Sensor Communication.

OCC was first introduced in 2001 [15], where it was proposed the use of an image sensor (camera) to achieve simultaneous reception from multiple point-of-light markers.

The key feature comes from the two dimensional PD matrix structure (Figure 2.3) [16]. An image sensor is composed of a large number of photo detectors (pixels). This allows the separation of different VLC sources and thus to distinguish noise sources [17] (sun, street lighting ...) from the signal sources. Focusing on a single VLC signal in an image plane, this signal will be sampled and processed as a time domain signal. Knowing the particularity of the two-dimensional image sensor plane, simultaneous processing for multiple VLC signals in the spatial domain becomes possible, thus creating a Multiple Input Multiple Output (MIMO) system.

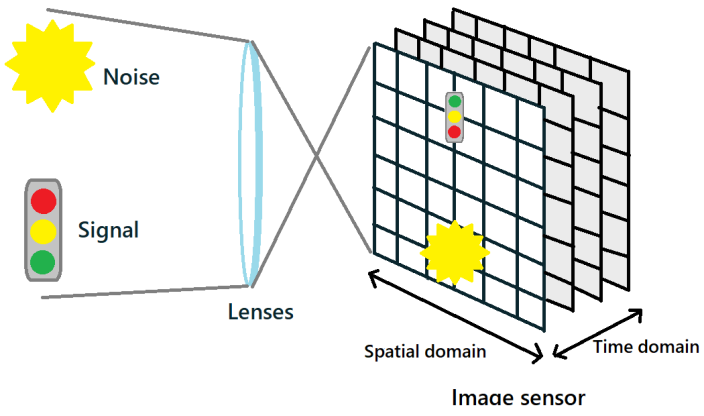


Figure 2.3: Image sensor capture.

Photo diode based reception is much simpler than in OCC because it comprises only a single (or an array) of photo elements, on the other hand, OCC involves a large amount of

photo elements spatially divided, this structural difference leads the image sensor features [18]:

- **Multi target tracking** - Due to the two-dimensional spacial property, it is possible with the same image sensor to receive multiple signals with different positions on the camera's FOV. This is possible without the need for any complex algorithm to identify different signals.
- **Long distance transmission** - The camera that is the fundamental element of this type of technology is composed of several elements, forming a system of some complexity. One of these components is the lens. Through proper lens choice, it is possible to increase, decrease or correct (through optical filters) the field of view, thereby increasing the communication range (Figure 2.4).
- **Post processing** - Generally the analog signals produced by the image sensor's photo sensitive cells are later sampled and converted to a digital scale. This makes it possible in collaboration with image processing algorithms, filtering, Region of Interest (ROI) detection, signal augmentation processes, to improve the signal quality (Figure 2.4).
- **Universal spread** - Cameras (specifically digital) are implemented on all current smartphones, personal computers, cars and other daily products. This means that the application of this type of technology does not depend on the diffusion or implementation of devices outside the range known to the majority of the population.

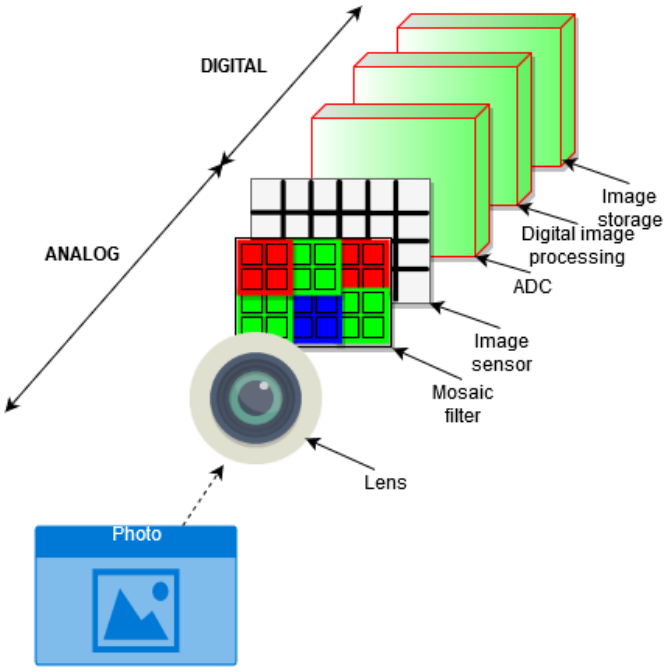


Figure 2.4: Digital camera embedded diagram.

Some works aiming at the application of camera receivers for real communication in V2V have already been done [19], using a special Complementary Metal-Oxide-Semiconductor (CMOS) image sensor, the Communication Pixel (CPx) has been reinvented in order to increase the response speed to variations of light intensity. An amplification lens has also been

added to the camera frame thus increasing the camera's FOV. Field measurements (indoor and outdoor) were performed, however it was found that camera intrinsic characteristics (frame-rate, exposure-time ...) varied according to the exposure environment (more or less light), thus showing that despite the several advantages of OCC, the high complexity of the entire camera system can present uncontrollable obstacles. Other results, this time using a High Speed Camera [20], tried to make also use of the distinct characteristics between those two different pixels. The CPx, are photosensitive cells responsible for detecting more abrupt variations of light intensity and then deliver a flag with the matrix positions of this point. This signal is subsequently measured by the Image Pixel (IPx) and processed by the image processor on an external Digital Signal Processor (DSP) (Figure 2.5). Communication pixels have a higher bandwidth compared to Image pixels, however they have a lower sensitivity, configuring the frame capture system to give priority to serving CPx can increase the system bandwidth, consequently impairing the image quality obtained at the output. The results compared to the previous work are more satisfactory, however its noticeable the complexity of the entire setup, as well as the hardware cost (High Speed Cameras).

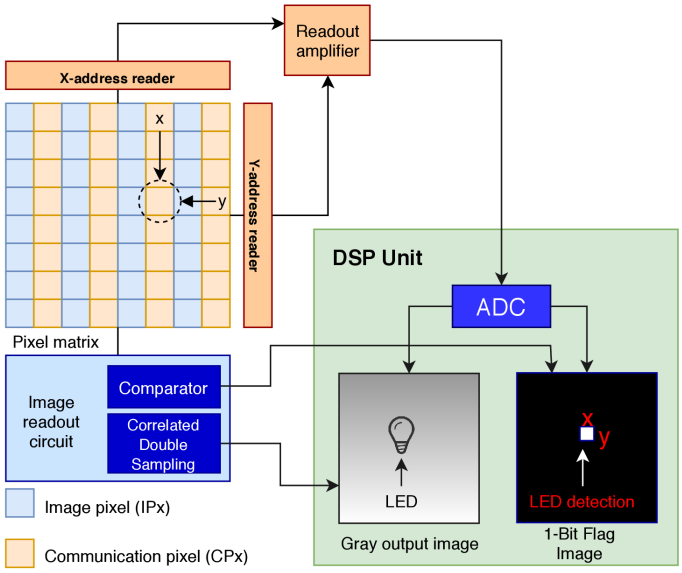


Figure 2.5: Overview of an image sensor.

2.3 System perspective

The communication system under study is divided into three main blocks: transmitter, channel and receiver (Figure 2.6). It is characterized by being one-way-link from transmitter (LED rear light) to the receiver (camera). From the perspective of the transmitter, it does not have any knowledge about the receiver, ie, has a behavior similar to broadcast. VLC communications have the particularity of being preferably directed Line of Sight (LOS), which means that optical power is concentrated in a narrower beam and directed to the position of the receiver. From the perspective of the receiver, consisting essentially of a camera (image sensor plus

lens), it has the task of find the ROI (in this case the car rear light), capture the frame and then decode it through computation/image processing algorithms. The transmission channel, an outdoor OWC environment, is characterized by a link range spanning from a few meters to a few hundred. The atmospheric conditions have impact in this type of communication link. Phenomena such as fog, snow and rain can lead to the fading, attenuation and dispersion of the transmitted signal. In addition, light reflection in other objects, and other noise lighting sources (Sun, street lights, cars headlights), also presents a limit to the quality of the transmission signal.

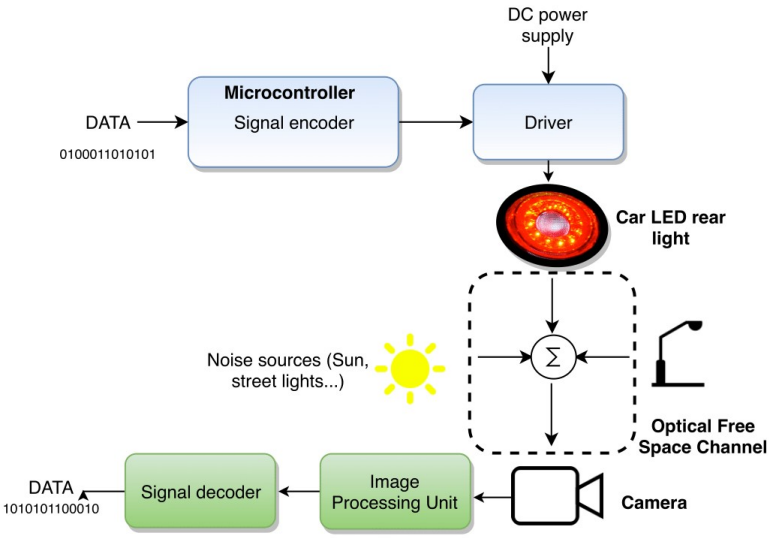


Figure 2.6: System diagram

2.3.1 The transmitter: LED

Today, LED lighting is everywhere, with different shapes, colors and applications ranging from indicator lights to solid-state lighting. Its energy efficiency [21], production cost, size and other features makes it the first choice for most lighting solutions, both indoor and outdoor. In addition, when compared the bandwidth of LEDs (with values close to 10MHz) to its predecessors, it can be said that it was the main invitation to apply these components to VLC communications. Most of the modulation techniques of this technology involve oscillating the light intensity level of the source at a high frequency (above the human eye perception), so a minimum bandwith capacity is needed. As mentioned earlier, it all started around 100 years ago when Oleg Losev [11] wrote:

> *"In this paper are further described on the phenomenon of the luminescence produced at the contact of a carborundum detector in the connection with the view on luminescence as the consequence of the process in the contact which is very similar to cold electronic discharge."*

Later, and as already presented in the section 2.1, Nick Holonyak Jr. created the first red LED. In subsequent years, companies like Hewlett Packard began marketing the first devices,

but their efficiency was rather low and costly to produce. In 1972, M. George Craford, who was a graduate student of Holonyak, by adding nitrogen to the semiconductor creates the yellow LED. Over the years and by investigating different bonds and compounds such as Aluminum Gallium Arsenide (AlGaAs), has been possible to improve the luminous efficiency. Shuji Nakamura of Nichia Corporation introduced the blue LED in 1992, giving rise to the High-Brightness LED (HB-LED).

2.3.1.1 P-N Junction

An LED is a semiconductor with a PN type junction [22][21]. When external forward-bias voltage is applied to the junction it emits light. Figure 2.7 illustrates the band energy model, with two available energy bands. W_g (band gap energy) is the energy width of the bandgap. In the upper energy band (conduction band) there are free electrons, this is the n-type semiconductor zone. In the lower zone (valence band), there are unbound "holes" in free movement, this is the zone of the p-type semiconductor. These holes exist due to electrons that have been released from the neutral atom, hence steady state equilibrium. When an electron attaches to a hole, energy in the form of photons is released and the atom returns to its neutral state. When the two semiconductors are brought into contact, the electrons and holes of the corresponding zones recombine in the interface region, however, a neutral region is also created which prevents the electrons and holes from combining because they do not have enough energy to overcome the barrier. However, when forward voltage bias is applied the barrier decreases and the potential energy of the electrons increases, so that, the electrons and holes can move to the barrier region. When an electron meets a hole, the electron "falls" into the valence band and recombines with a hole. During this process, energy is released in the form of a photon.

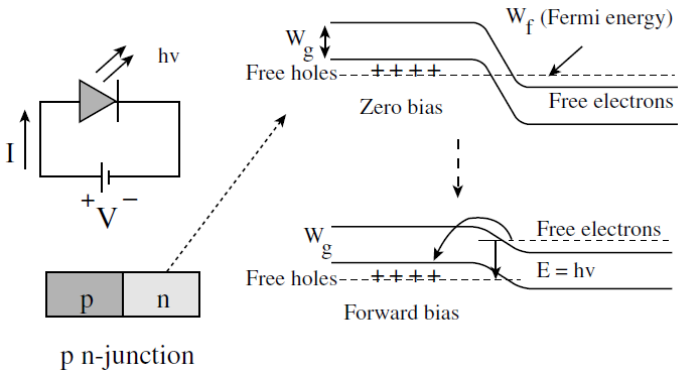


Figure 2.7: LED energy band gap model. [22]

Depending on the energy bandgap W_g, the emitted light will have different wavelengths, as shown in Equation 2.1:

$$\lambda = \frac{h \cdot c}{W_g} \tag{2.1}$$

Where h is the Planck constant and c is the speed of light. The energy of the bandgap is expressed in electron Volts.

Different semiconductor materials [23] lead to band gap energies of different values and consequently light emission with different wavelengths [22][21][24]. Table 2.1 represents these values.

Table 2.1: LED semiconductor compounds

LED Material	**Peak Wavelength or range (nm)**
A1GAN/GaN	230-350
InGaN/GaN	360-525
ZnTe/ZnSe	459
SiC	470
GaP	470
GaAs_{0.15}P_{0.85}	589
A1GaInP/GaAs	625-700
GaAs_{0.35}P_{0.65}/GaAs	632
GaAs_{0.6}P_{0.4}/GaAs	650
GaAsP/GaAs	700
GaAs	910-1020
InGaAsP/InP	600-1600

2.3.1.2 Bandwidth

The modulation bandwidth of the LED depends on several factors: injected current, junction capacitance and parasitic capacitance [24][1]. Capacitance effects can be reduced by maintaining a constant DC level, thereby modulating the signal by superimposing an AC level. The device frequency response is exposed in the Equation 2.2:

$$\frac{P(\omega)}{P_0} = \frac{1}{\sqrt{1+(\omega\tau)^2}} \tag{2.2}$$

P_0 represents the DC power, and $P(\omega)$ represents the relative power released by the LED at a frequency ω. From the equation it is possible to verify that the bandwidth is dependent of the carriers lifetime (τ), this means that increasing the lifetime of the carriers (by doping techniques for example) increases the modulation bandwidth [21].

The optical 3-dB bandwidth is obtained by setting the Equation 2.2 to 0.5. As is known the equivalent electrical bandwidth is given by:

$$eletrical\ bandwidth = \frac{1}{\sqrt{2}} \cdot optical\ bandwidth \tag{2.3}$$

And this is because the relation between optical and electrical signals follow a square-law process. The optical power (P_{opt}) varies linearly with the current: $P_{opt} = i \cdot R$, where R represents the LED responsivity. Instead, the electrical power is given by: $P_{elec} = i^2 \cdot R$, with R meaning the device resistance, thus the square-law relation. The relation is illustrated in Figure 2.8.

In Table 2.2 is represented some different types of LED's and corresponding bandwidth.

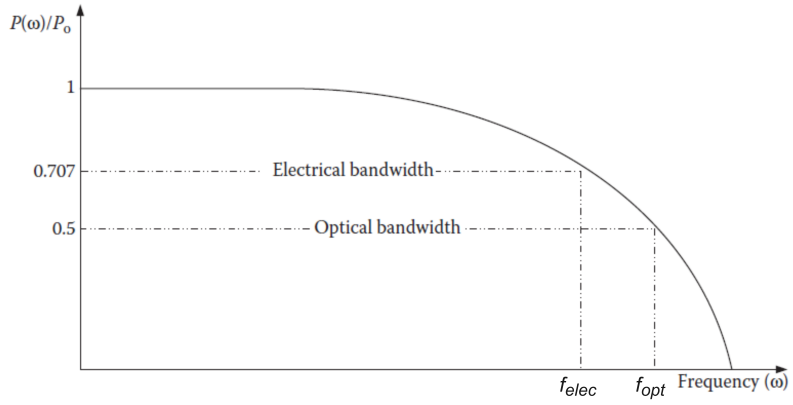


Figure 2.8: LED optical and electrical bandwidth. [24]

Table 2.2: Some different LED's types and corresponding bandwidth.

RGB / RYGB - LED	YB-LED	OLED
10-20 MHz	1-5 MHz	100 kHz

2.3.1.3 LED structure

LEDs have different structures [21][22]. Its structure has a high impact on both its luminous and energy efficiency. There are two distinct types of structures [24]. Planar structure is the simplest, it emits light all over its surface and can be approximated to the Lambertian emission pattern (further in the next section). The other type is known as the "LED dome", has an N-type hemispherical structure around a diffuse P-type zone (Figure 2.9).

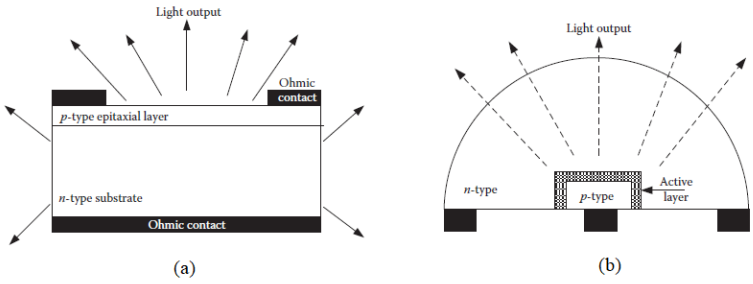


Figure 2.9: LED structures: (a) planar (b) dome. [24]

2.3.1.4 Radiation pattern

The radiation pattern of an LED defines the relative optical power strength of a light source. The radiation pattern emitted by every chip face will generally be Lambertian [6], although in some LEDs, this pattern is modified by special microstructures on the chip and by wave guide effects inside the chip. The final light pattern generated by an LED is the result of the sum of three terms: the light directly refracted by the encapsulating lens, the light internally reflected inside the lens, and the light reflected by the reflecting cup. To address all these

effects, Gaussian and cossine-power [25] functions are the solutions found, however, here we will only cover simpler emission patterns based only on Lambert equations.

As already mentioned, LEDs are designed to have a Lambertian beam distribution. This type of emission pattern is based on Lambert's cosine law [26][24].

$$I(\theta) = I_0 \cdot cos^{ml}(\theta) \tag{2.4}$$

According to the Equation 2.4, the radiation intensity of the source depends on the angle of viewing (θ). The maximum intensity (I_0) is observed for the perpendicular of the LED surface plane, ml represents the Lambertian emission order (Equation 2.5)[26].

$$ml = -\frac{ln(2))}{ln(cos(\theta_{1/2}))} \tag{2.5}$$

The $\theta_{1/2}$ is the half-intensity beam angle of an LED, an important characteristic of all LEDs. The coverage range and radiation pattern of single LED light source is affected by the half-intensity beam angle, such that narrower $\theta_{1/2}$ increases the illumination range and makes the pattern more directive, as could be verified in Figure 2.10, which represents different beam patterns depending on the $\theta_{1/2}$ value. This simulation was made on MATLAB®.

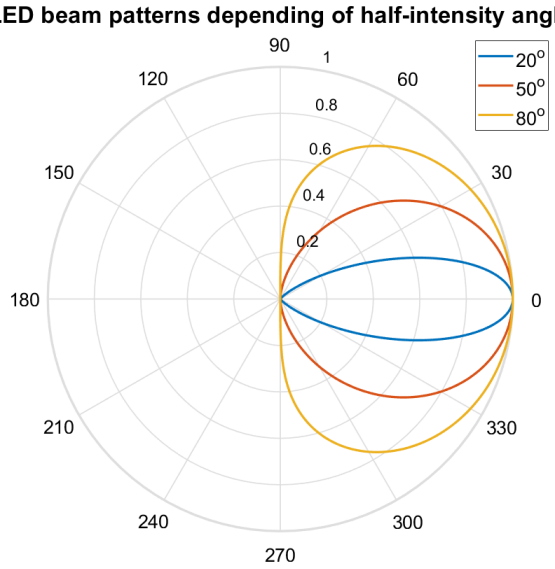


Figure 2.10: Different LED beam patterns depending on half-intensity angle.

2.3.2 Channel modeling

In order to obtain as much efficiency as possible from the communication channel, it is necessary to know its characteristics. The channel is characterized by its impulse response. Within OWC communications, there are two main classes: indoor and outdoor channels. In VLC, there is a considerable amount of work already published with analysis (experimental measurements and computer models) for the indoor channel. This is usually due to its isolation

from outside noise (sunlight, street light, advertising signs, among others). Indoor channels have a lower implementation complexity when compared to the outdoors case.

Directed LOS doesn't consider any reflections in its link configuration, and therefore path loss is easily calculated from the transmission beam pattern equations. However, when considered the reflections, a non-LOS configuration occurs. The impact of the environment variables in the outdoor channel have a great importance on the deterioration of the optical signal power. Also interference from other light sources (cars and streetlights) is a type of noise that impairs the Signal to Noise Ratio (SNR) balance of the VLC communication systems.

2.3.2.1 Propagation Model

LED's are the primary signal/light source for OWC communications, so the angular distribution of the radiation intensity pattern is modeled using a generalized Lambertian radiant equation, which has been previously analyzed in subsubsection 2.3.1.4.

Ignoring the effects of absorption and scattering imposed on the communication link and considering the LED as a Lambertian source, the DC gain for a receiver at a distance d with an angle θ is given by:

$$H_{los}(0) = \begin{cases} \frac{A_r(m_l+1)}{2\pi d^2} cos^{m_l}(\phi) T_s(\psi) g(\psi) cos(\psi) & 0 \leq \psi \leq \Psi_c \\ 0 & elsewhere \end{cases} \tag{2.6}$$

In Equation A.5 is taken into account the impact of the receiver: $g(\psi)$ represents the gain of the imaging concentrator, A_r the receiver area and $T_s(\psi)$ an optical band pass filter. For simplicity, the use of the reception filter $T_s(\psi)$ can be ignored by assigning it's value to 1. Moreover, and since this work relies on the use of OCC in signal reception, it makes no sense to take into account the impact of constraints imposed by the use of a photosensitive receivers, and thus variables such as A_r and $g(\psi)$ are also set aside. Thus, the considered DC gain becomes:

$$H_{los}(0) = \begin{cases} \frac{(m_l+1)}{2\pi d^2} cos^{m_l}(\phi) cos(\psi) & 0 \leq \psi \leq \frac{\pi}{2} \\ 0 & elsewhere \end{cases} \tag{2.7}$$

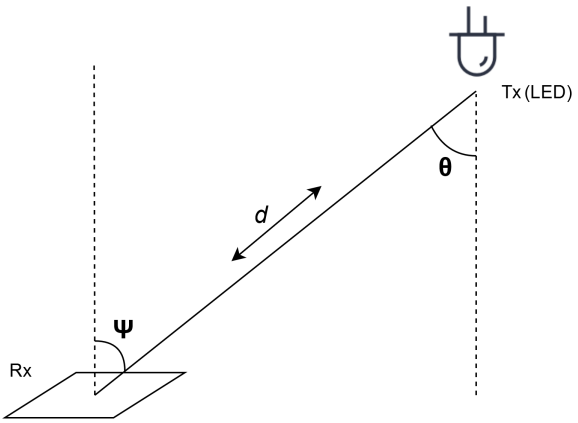


Figure 2.11: Geometry LOS propagation model.

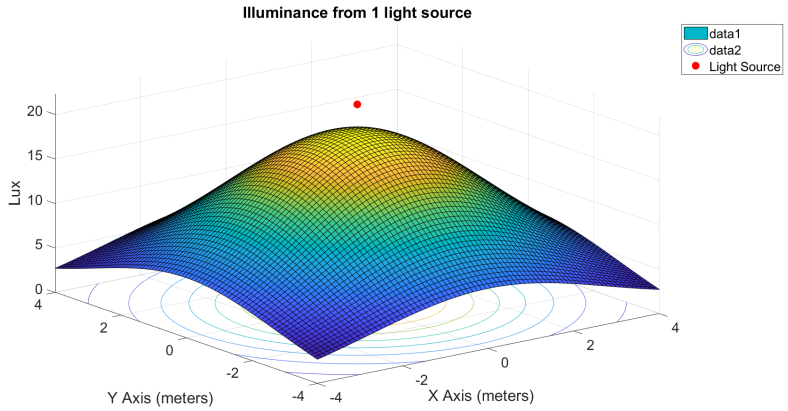
The received power therefore becomes:

$$P_{r-los} = H_{los}(0)P_t \tag{2.8}$$

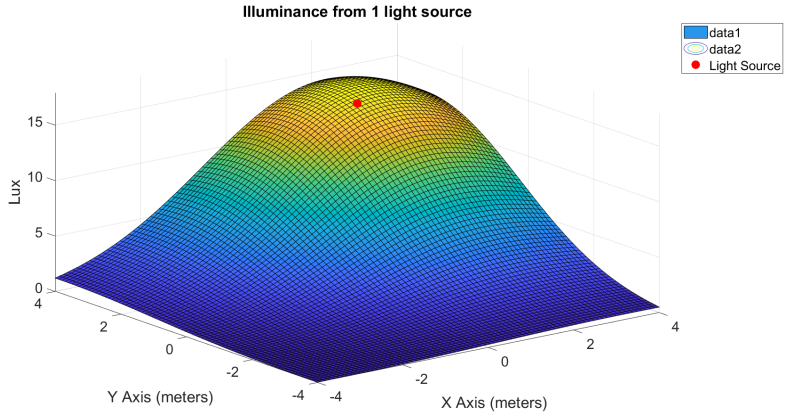
Figure 2.12a and Figure 2.12b presents the result of the simulation performed in MATLAB® for a simple LOS emission, with the following characteristics Table 2.3.

Table 2.3: LOS simulation properties

LED luminous flux:	1000 lm
Receiving surface dimension:	8x8 m
Distance:	10 m
LED half-power angle:	20º
Vector normal to reception plan:	[0 0 1]
Emission vector:	[0 0 -1], [3 4 -1]



(a) Emission vector [0 0 -1]



(b) Emission vector [3 4 -1]

Figure 2.12: LOS simulation for two different emission vectors.

It is possible to show that by changing the LED emission vector, the emission pattern also changes as expected. The vector of Figure 2.12b makes an angle of approximately 78° with

the normal vector. Comparatively, the vector of Figure 2.12a is parallel to the normal of the emission plane.

For nondirected LOS and diffuse links, the optical path loss is more complex to predict, since it depends on a third factor: the reflection surface. Characteristics such as reflection index, position and surface orientation among other factors influence signal propagation. The received power is then defined in Equation 2.9, $H_{refl}(0)$ represents the reflected paths.

$$P_{r-nlos} = (H_{los}(0) + \sum H_{refl}(0))P_t \tag{2.9}$$

Once again, the intrinsic characteristics of the receptor have been suppressed. Equation 2.10, shows the calculation of the optical path for a first degree reflection (only the first reflection is considered).

$$H_{refl}(0) = \begin{cases} \frac{m_l+1}{2\pi(d1d2)^2}\rho \cdot dA \cdot cos^{ml}(\alpha 1) \cdot cos(\alpha 2) \cdot cos(\alpha 3) \cdot cos(\alpha 4) & 0 \leq \alpha \leq \pi/2 \\ 0 & elsewhere \end{cases} \tag{2.10}$$

Light reflection can be classified into two types: diffuse or regular. Diffuse reflection (the type approached in this simulation), occurs when light falls on an uneven surface and reflects it. The reflected light rays propagate in many different directions. That is, each infinitesimal area of the reflecting surface behaves as a unique light source.

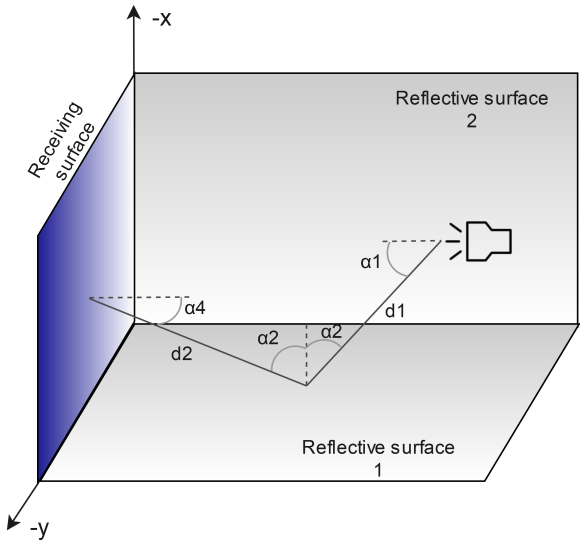
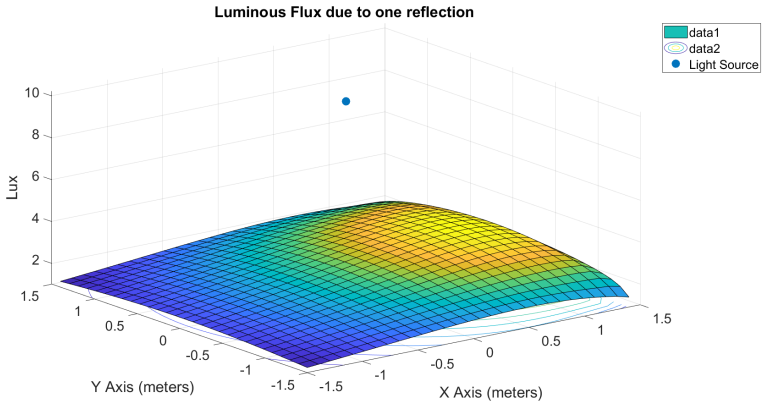


Figure 2.13: Geometry non-LOS propagation model.

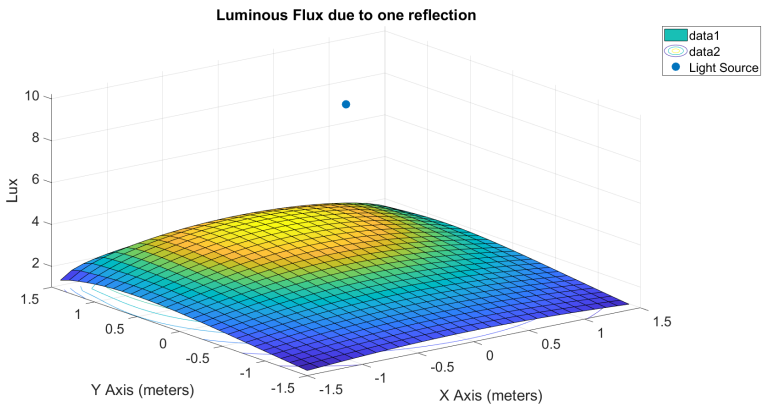
Table 2.4: Non LOS simulation properties

LED luminouns flux:	1000 lm
Receiving surface dimension:	3x3 m
Distance:	5 m
LED half-power angle:	40º
Vector normal to reception plan:	[0 0 1]
Emission vector:	[0 0 -1]
Reflection coef.:	1 (Both surfaces)

Some simulations were performed, based on the properties presented in Table 2.4 and respecting to the geometry of Figure 2.13 the results of Figure 2.14a and Figure 2.14b were obtained.



(a) Reflection due to surface 1



(b) Reflection due to surface 2

Figure 2.14: Non LOS simulation for two different reflective surfaces

2.3.2.2 Reflection coefficient for asphalt

Asphalt is the most commonly used material for paving roads. Since the purpose of this setup is to be equipped in an outdoor environment, the asphalt then becomes part of the transmission channel, and understanding its impact is important. The reflection coefficient, or

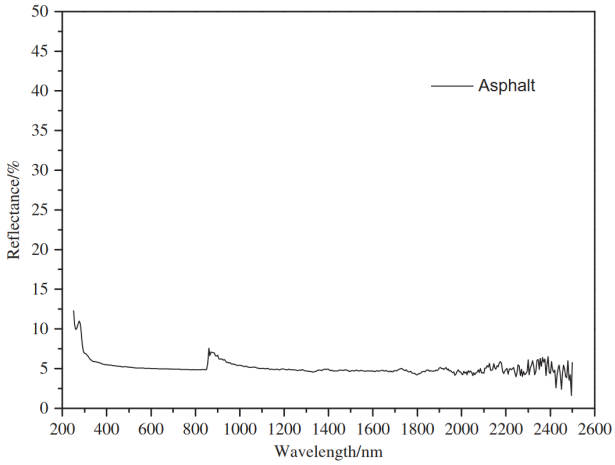
also known as Albedo, of a body highlights the portion of incident radiation that is reflected back to the medium.

It is in the interest of this work to know the value of the asphalt reflection coefficient for the visible light radiation spectrum. The reflectance is dependent on several factors: color, surface roughness, compounds, temperature and humidity.

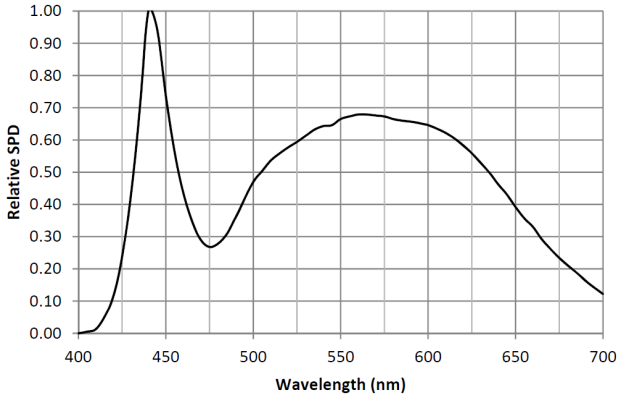
Visible light, as is well known, is not limited to a single wavelength, but to a range. Consequently, the coefficient of solar reflection along the visible light spectrum will also change regardless of the reflective body and ambient conditions. Another important factor is to understand that the power radiated by a light source (Power Spectral Density (PSD)) is not uniform for the whole wavelength, but variable according to the inherent characteristics of the light source (color temperature, light filters, among other factors).

In conclusion, the visible light reflection coefficient for asphalt is dependent on two factors: the reflectance of the asphalt and the PSD of the light source (Equation 2.11).

$$\rho_{asphalt} = \frac{1}{300} \int_{400nm}^{700nm} Reflectance(\lambda) \cdot PSD(\lambda) d\lambda \tag{2.11}$$



(a) Asphalt reflectance [27]



(b) LED with 5000K color temperature PSD. [28]

Figure 2.15: Asphalt reflection coefficient arguments

The graph of Figure 2.15a shows the reflectance of the asphalt according to the wavelength [27]. Important to note that it doesn't take into account the ambient conditions (humidity for example), so it could be said that is for a "dry condition". It is verified that for the visible spectrum range it has an almost uniform value around 5%. Figure 2.15b shows the relative PSD graph of an LED light source with a color temperature of 5000K [28].

The values of the graphs were both normalized and subsequently their equations drawn through interpolation. The reflection coefficient calculated for these circumstances was 8.20%.

2.3.2.3 Channel impulse response

Resuming the previous sections, the optical wireless channel is characterized by direct and reflective links, angles of incidence, surfaces areas, among other parameters. All this features will affect its response to an input signal (light signal for instance). When a signal is transmitted through a channel, and taking in consideration all the different surfaces that makes part of it, the impulse will travel through different paths, this is known as multipath. Which means that, looking for an infinitesimal light ray, its characteristics like received strength and propagation time will be different according with the communication link in question.

This is because all non-directed LOS signal is characterized by one (or more) reflections, resulting in a comparatively longer link distance, and consequently longer optical signal transmission period, thus resulting in a delay spread.

Considering the time-to-distance conversion formula: $v(m/s) = s(m)/t(s)$, based on the geometry of Figure 2.13 and taking into account the simulation characteristics of Table 2.5(reflections, direct LOS, position and radiation pattern of the light source), a channel impulse response analysis was made. For that, it was simulated the situation where an instantaneous light impulse was emitted in the channel from its central position. Then, all the independent light rays with different paths were received and the time that it took to travel from the light source to the receiver plane measured. The Figure 2.16 presents the results obtained. Important note, that, this is the impulse response for this particular channel, this means that, changing any characteristic will consequently generate a different response.

Table 2.5: Channel Impulse Response simulation properties

LED luminouns flux:	1000 lm
Receiving surface dimension:	3x3 m
Distance:	25 m
LED half-power angle:	40^{o}
Vector normal to reception plan:	[0 0 1]
Emission vector:	[0 0 -1]
Reflection coef.:	0.5 (Both surfaces)

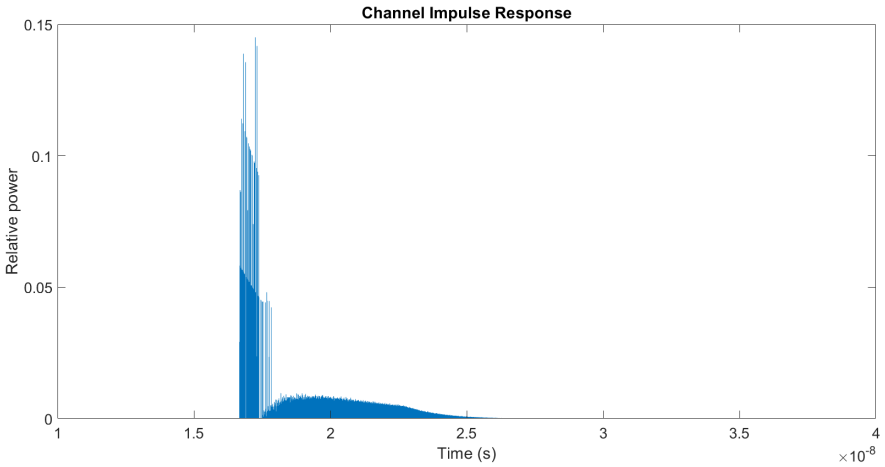


Figure 2.16: Channel impulse response graph.

Analyzing the results of the graph of Figure 2.16, it can be seen that the signal propagates through the channel for more than 10ns, the highest accumulated optical power is in the peak of the channel impulse response, due to LOS. Note that for this simulation only one reflection was taken into consideration.

2.3.2.4 Noise/attenuation sources

Any communication channel is affected by external and uncontrollable phenomena that impair the quality of the receiving signal. Considering the relation between the level of a desired signal with the level of background noise, also known as SNR, the objective of telecommunications is to keep this ratio as largest as possible. From this relation, it could be concluded that only two ways are possible to reduce the value of this ratio: by decreasing the desirable signal level, i.e., due to attenuation phenomena, or by increasing the noise level due to the impact of noise sources .

Focusing in OCC, more precising the image sensor element, the all process of capturing a frame can be resumed into three different operations: photon-to-charge operation, charge-to-voltage operation and voltage-to-digital operation [29]. All this phases are affected by noise sources. Focusing in the first stage, the pixel sensor behaves as photon-capturing element, the average number of photons I_{photon} collected by a single pixel during a unit time period is given by:

$$I_{photon} = round(\frac{I_{irrad} \cdot A_p}{E_p})$$

where I_{irrad} is the sensor's irradiance in W/m^2, A_p is the pixel sensor area, and E_p is the energy of a single photon at the wavelength λ. This process is affected by intrinsic and extrinsic noise sources. Looking firstly for the intrinsic sources, the shot noise I_{shot}, which is associated with the particle nature of light, shows a Poisson distribution and affects the photon counting. This means that the collected photons during a exposure period t, including the shot noise effect, that are converted to electrons I_{e^-} for each pixel is given by:

$$I_{e^-} = \eta_{QE} \cdot I_{photon} \cdot t$$

where η_{QE} is the quantum efficiency for a given wavelength and represents the capability of a semiconductor to produce electrons from incident photons. Another intrinsic noise source is the dark current I_{dc}, and is induced by the thermally generated electrons that discharge the pixel. Looking now for the extrinsic noise sources, it can be consider, that any light source present in the communication channel that doesn't produces relevant VLC data can be consider as noise source, the major example is the Sun. The photons produced by these noise sources will be captured by the pixel cell, that consequently induce an erroneous count offset. This means that the total number of electrons I_{total} is a result of light signal, noise light sources, inherent shot noise and dark current:

$$I_{total} = I_{dc} + \eta_{QE} \cdot (I_{photon} + I_{noise})$$

Resuming, a light source not recognized as signal transmitter will impose an offset in the total optical power received. In other words, it will saturate the image sensor, creating a "blind" effect.

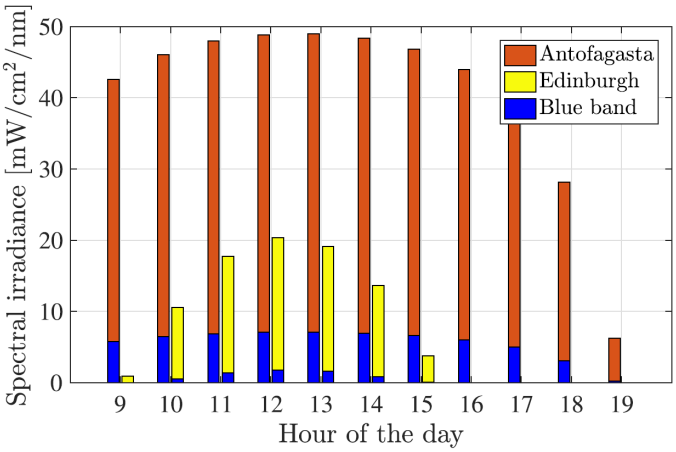
In the case of VLC, as the channel is unguided, it is more susceptible to disturbances, either attenuation or noise sources. In the next subsections some of these cases will be analyzed, with special attention to climatic phenomena.

2.3.2.4.1 ***Daylight.***

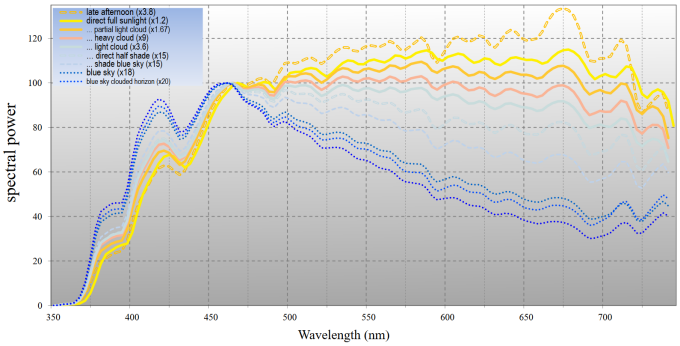
Solar radiation has the consequence of degrading the SNR of OWC by increasing the noise level. There are already several techniques to overcome this type of problem, from the use of more complex modulation techniques (Direct Current Optical Orthogonal Frequency Division Multiplexing or adaptive threshold), optical filters is another solution used, however, when changing the geometry of the system i.e., position of transmitter/receiver according to sun exposure, the efficiency varies abruptly. To overcome this constrain, digital cameras make use of digital filters, dynamic gain controllers, shutter speed or camera aperture.

According to the American Society of Testing and Materials (ASTM) [30], the solar constant flux density outside the Earth's atmosphere is around $1366.1 W/m^2$. The solar irradiance, as expected, is not limited to the visible light range, but from 250nm to 2500nm. The shortest distance between the Sun and the Earth's surface occurs when it is located at the zenith point, which reflects the moment of the highest radiation intensity received.

In addition, the solar irradiance varies by geographical position [30], seasonal and diurnal variations from the rotation of the Earth around the Sun, and the rotation of the Earth around its own axis. Also the radiation power through the visible spectrum is not uniform [31], this happens due to the different absorption and scattering effects of air molecules and aerosols in the atmosphere.



(a) Sun irradiance during the day for different geographic locations[30].



(b) Sun light PSD measured using a X-Rite i1Pro (118 sensor values between 350-740nm)[31]

Figure 2.17: Sun light radiation graphical analysis.

The results from Figure 4.7a were obtained using a *Hamamatsu S8664-50K* Photosensor, with a 19.6mm^2 area, from two different geographic locations: Edinburgh, UK and Antofagasta, Ch [30].

2.3.2.4.2 ***Atmospheric Attenuation.***

Optical signal propagation through a free space channel is very sensitive to the atmospheric conditions. Photons in the atmosphere are extinguished or scatered by the molecules. These is described by the Beer-Lambert Law (BLL). [24]. BLL describes the transmittance of optical field through the atmosphere as function of the propagation distance and wavelength.

The main reason for the decay of the received power, is the scattering effect due to diffraction by particles of similar size to the wavelength of light.

For an OWC link, the relationship between the transmitted and received optical power is given by:

$$P_r = P_t e^{-\tau} \tag{2.12}$$

The variable τ represents the optical depth. The optical depth is related to the atmospheric attenuation coefficient γ_t and the transmission path length $L(m)$ by the Beer-Lambert's law:

$$T(\lambda, L) = \frac{P_r}{P_t} = e^{-\tau} = exp[-\gamma_t(\lambda)L] \tag{2.13}$$

(NOTE: For a τ of 0.7, it corresponds to a loss of 3dB). Where γ_t means the attenuation coefficient of atmosphere for a determined wavelength, and $T(\lambda, L)$ represents the transmittance of the atmosphere at wavelength λ and distance L. The attenuation of the optical signal can generally be expressed as:

$$\gamma_t(\lambda) = \alpha(\lambda) + \beta(\lambda) \tag{2.14}$$

Where α corresponds to the absorption coefficient (due to molecules and aerosols dispersed in the atmosphere), and β is the scattering coefficient for the molecular and aerosol particles. Absorption and scattering of light is a complex process involving Mie scattering, nonselective scattering and Rayleigh scattering by smaller particles, which has not been studied in this work.

The atmospheric transmission coefficient depends on several factors as already presented. It is not the purpose of this dissertation to describe in detail all the parameters and inherent dynamics that may characterize the various possible environments. For this, and using researches already done [32], specialized in the study of the transmission coefficient of the atmosphere according to various conditions, it was chosen the data with more interest to the situation here presented.

From there it was extracted the Figure 2.18 and Figure 2.19 that represent the scattering and absorption coefficients respectively, based on Mie calculations . In the Mie scattering graph (Figure 2.18), considering the thin stratus cloud situation, the coefficient presents a constant value along the visible spectrum around $15Km^{-1}$. In the Mie absorption graph Figure 2.19, it presents a more complex behavior, the $0.0004/Km^{-1}$ average value was considered. Applying this values to Equation 2.14, it is obtained the $15.0004/Km^{-1}$ value for the γ_t variable.

Considering now a link distance with a range close to the ones implemented in VLC (0 to 100m), and the Equation 2.13, the corresponding transmittance can be calculated, and the results are exposed in Figure 2.20. The results show that in these conditions, for a link distance close to the 100m, waves in the range of the visible spectrum would face a 3dB attenuation.

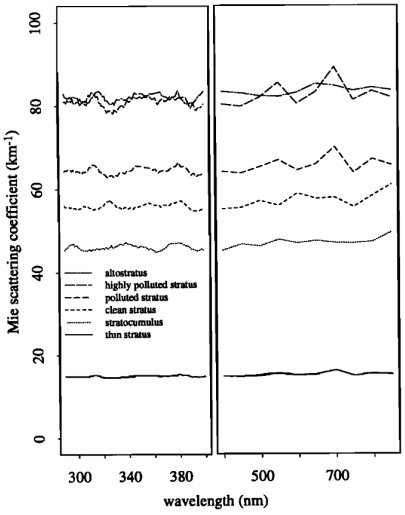


Figure 2.18: Mie scattering coeff. of pure water clouds [32].

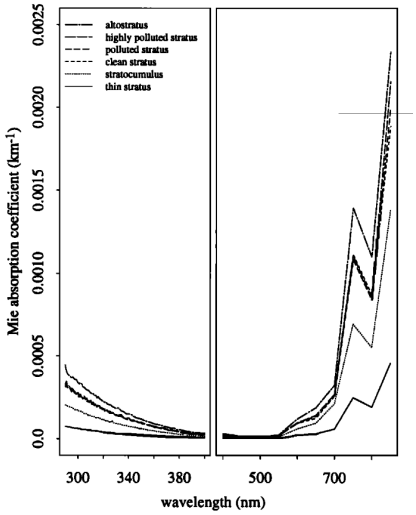


Figure 2.19: Mie absortion coeff. of pure water clouds [32].

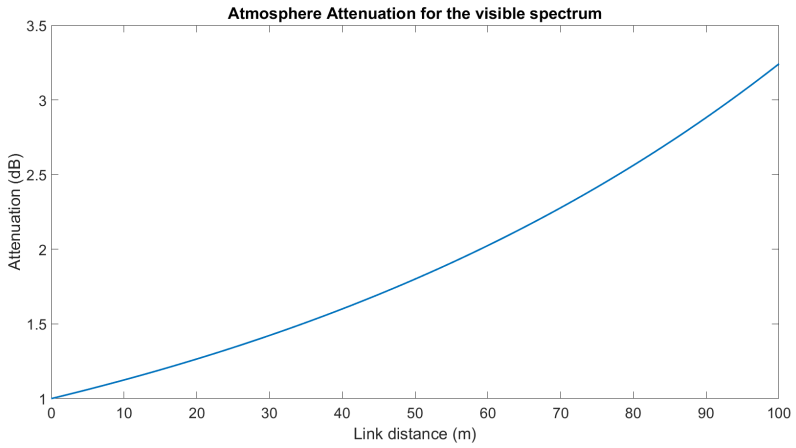


Figure 2.20: Atmospheric attenuation for visible spectrum in function of link distance and based on attenuation coefficient of Figure 2.18 and Figure 2.19

2.3.2.4.3 ***Rain.***

Rain has a distance-reducing impact on OWC[22], this is because the radius of raindrops $(100 - 200\mu m)$ are much larger than the visible light wavelength. Rainfall attenuation is generally modest for moderate precipitation values (<2.5cm/hour). However, for high precipitations (> 10cm/hour) the attenuation value may be high enough to disrupt the communication channel, however such storms occur rarely and for short periods of time.

As already mentioned,the raindrops have a significant size, this size is large enough to cause reflection and refraction. As a result these droplets cause wavelength independent scattering [24]:

$$\beta_{rain} = 1.25 \times 10^{-16} \frac{(\Delta x/\Delta t)}{a^2} (cm^{-1}) \tag{2.15}$$

The variable a characterizes the raindrop size distribution and $\Delta x/\Delta t$ is the rate of rainfall. The attenuation value for a rain rate of R $mm/hour$ is given by [33]:

$$\alpha_{rain} = 1.076R^{0.67}(dB/km) \tag{2.16}$$

According to the literature, moderate rain can be characterized by a rate of $7mm/hour$. Applying this to the Equation 2.16, and considering a link distance between 0 and 100m, the results of Figure 2.21 were computed.

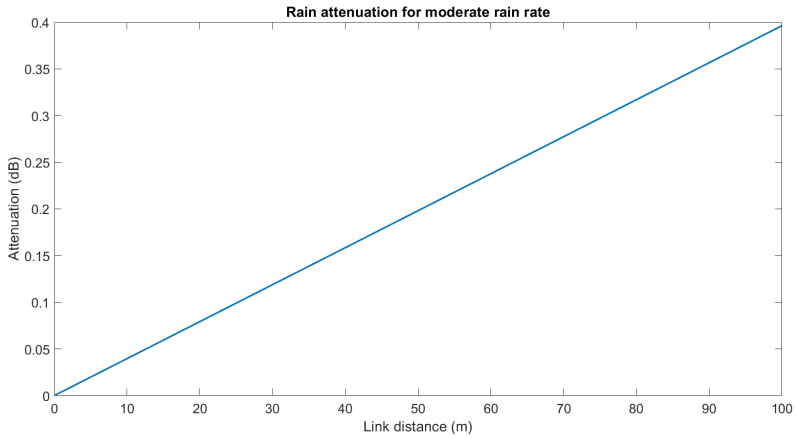


Figure 2.21: Rain attenuation considering a rain rate of 7mm/hour, in function of link distance.

From the results of Figure 2.21, and as expected, for link distances in the order of the used in VLC communications, the impact of rain attenuation is almost insignificant (<0.5dB).

2.3.2.4.4 ***Fog.***

Fog is the atmospheric phenomenon that imposes the most severe attenuation effect on OWC[22][24]. Fog consists of small water particles with a near-infrared radius size, which are formed near coastal areas and near the earth's crust. Since it is highly complex to characterize (it requires detailed information of parameters like particle size, refractive index, particle size distribution...), it is often described in terms of its visibility.

The models Kruse, Kim and Al Nabulsi, use this approach and predict specific attenuation using visibility [33]:

$$a_{spec} = \frac{10logV\%}{V(km)}(\frac{\lambda}{\lambda_0})^{-q}(dB/km) \tag{2.17}$$

In Equation 2.17, $V(km)$ means the visibility achieved by the user in Km, $V\%$ stands for transmission of air drops to percentage of clear sky. According to definitions [34], visibility is the distance for which transmission of air drops to 5% (13dB attenuation) of the clear sky value, so, $10logV\%$ is considered as $13dB$. The variable $\lambda(nm)$ represents the wavelength and

λ_0 is the visibility reference (550 nm).

$$q = \begin{cases} 1.6 & if \quad V > 50km \\ 1.3 & if \quad 6 > V > 50km \\ 0.16V + 0.34 & if \quad 1 < V < 6km \\ V - 0.5 & if \quad 0.5 < V < 1km \\ 0 & if \quad V < 0.5km \end{cases} \tag{2.18}$$

The only difference between the Kim and Kruse models is the calibration of the variable q. For Kruse, in any weather condition, less attenuation implies shorter wavelength. However, Kim's model rejects this assumption for low visibility fog. Equation 2.18 shows the values suggested by Kim for the variable q.

The France Telecom model, developed by Al Naboulsi, separates the characterization of fog into two distinct situations: fog due to advection currents and fog due to terrestrial radiation.

$$\gamma_{ADV}(\lambda) = \frac{0.11478\lambda + 3.8367}{V} \tag{2.19}$$

The advection fog is formed by the collision of wet and warm air masses (Equation 2.19).

$$\gamma_{RAD}(\lambda) = \frac{0.18126\lambda^2 + 0.13709\lambda + 3.7502}{V} \tag{2.20}$$

Radiation fog is related to the ground cooling by radiation (Equation 2.20).

The Equation 2.17 was used to calculate the attenuation factor for the visible spectrum, considering a moderate fog with a visibility of 500m, it was verified that the attenuation coefficient presents a constant value along the spectrum $(300 - 700nm)$, with a value of $26dB/Km$. Considering a link range of 100m, the attenuation factor in function of distance of Figure 2.22 was obtained. It can be concluded, that even for a moderate fog condition, it imposes a severe attenuation in the transmitted signal strength.

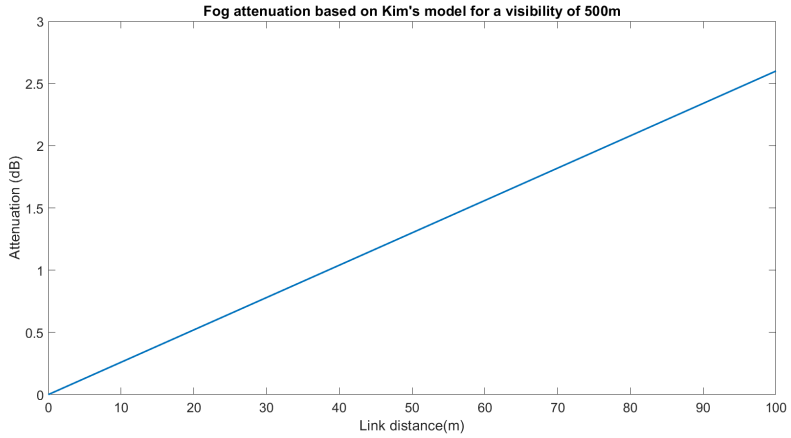


Figure 2.22: Fog attenuation for a visibility of 500m, in function of link distance.

2.3.2.5 Car's headlights emission pattern

Since the application of the setup discussed in this work is based on the signalization lighting from cars, it was considered of interest to study and simulate the emission pattern of the front car headlights. As can be verified in the appendix (section A.2), there are rules that control all the characteristics inherent to the construction, assembly and emission standard of the headlamps applied in the approved commercial cars.

Taking into account the knowledge of subsubsection 2.3.2.1, and the limits of regulation imposed by United Nations Economic Commission for Europe (UNECE) in terms of luminosity according to matrix location (section A.2), some MATLAB® simulations were elaborated.

The main concern of regulation is the safety of all road users. From drivers to pedestrians, thus limiting the optic power output from headlights by angular location prevents other users (other drivers for example) from being blinded by the brightness of the light coming from other vehicles they come across. One of the many noteworthy features is the differentiation between driving on the road through the right or left side, this creates the imposition of limiting the light emitted to the opposite lane so as not to blind the opposite drivers.

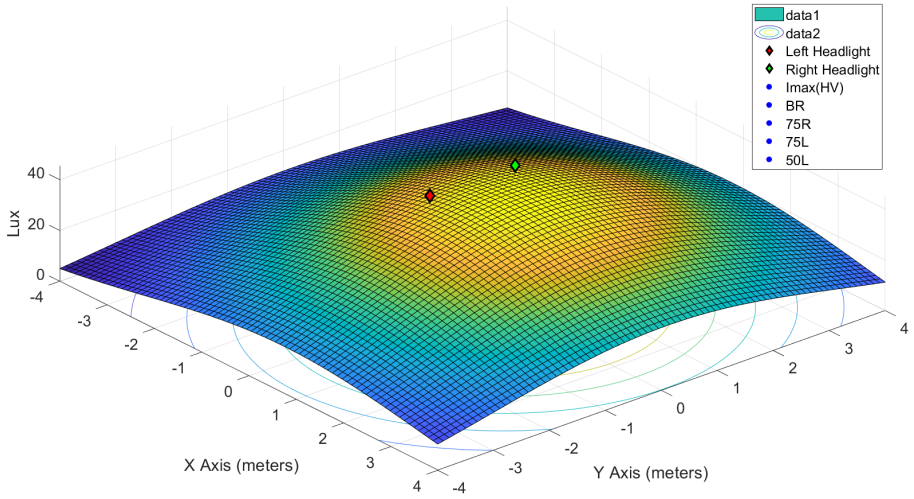
One of the most basic solution found by car manufacturers is to adjust the emission pattern of each headlamp individually. This way, is possible to avoid the blind effect in the contrary drivers by turning the headlamps light emission pattern to opposite side from where they are coming from. For example, for right-hand drivers, the emission pattern of the left headlamp is a little bit turned to the right side.

For the light sources, it was designed a simulation scheme as close as possible to the actual mounting in modern cars. The two lights were placed side by side, with a distance of 1.53m between them, the rest of the configurations, is expressed in Table 2.6.

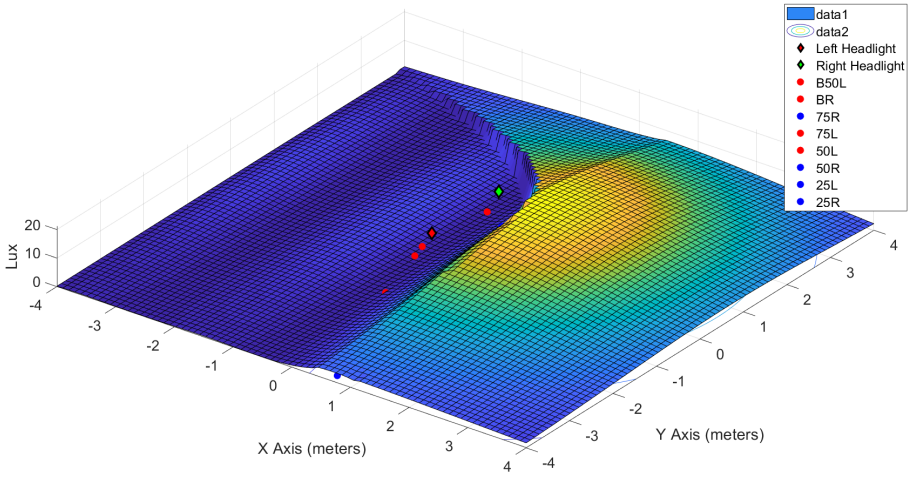
Table 2.6: Car's headlight simulation properties

Nº Sources:	Half power angle:	Luminous flux:	Test room dimensions:	Vector emission:	LED positions:
2	R: 6º L: 10º	R: 400lm L: 300lm	8x8x25m	R: [0.4, 0.2, -1] L: [0.4, 0, -1]	R: [0, 0.75, 25] L: [0, -0.75, 25]

The simulations of Figure 2.23a and Figure 2.23b for High Beam and Low Beam respectively took into account UNECE optical power limitations (consult appendix: section A.2). All calibration points are in the location specified by regulation. Red dots represent locations where power should be lower, while blue dots represent places where power should be higher than that level.



(a) High Beam



(b) Low Beam

Figure 2.23: Pattern simulation for car's headlights

For the high beam simulation, it is verified that the compliance with the stipulated rules is easily achieved. However, for the low beam simulation, due to a greater restriction, especially for the front and left flanks (these simulations were performed considering the right hand drive), shows that more complex techniques are necessary to achieve the desired results.

The results of Figure 2.23b were achieved by applying a principle close to that used in valve lamps, where the light pattern is projected through mirrors embedded in the lamp structure in order to model the light emission pattern for the required specifications. In this simulation, this was achieved by limiting the power emitted according to the angle between the light ray and the emission vector. However, there is an abrupt "filtering" between the lower and upper frontal zones, which is not close to the real situation. In reality, the cutoff region implemented by the mirror case, is more smooth and continuous.

To overcome this problem, it was studied the most current technological solutions. It was found the solution of MERCEDES® "Multibeam LED" technology[35], (also other car manufacturers are already making use of this innovation). This system makes use not of a

single LED, but instead 84 individually controllable ones (Figure 2.24). With this it is possible to build an intelligent light matrix where, for each matrix point and according to the needs of the momentary scenario, increases or decreases the brightness of each matrix individual LED.

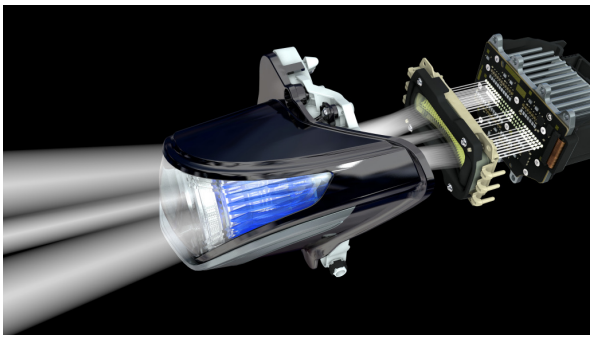


Figure 2.24: Multibeam LED headlamps on Mercedes E-Class [35]

Based on this solution, a new simulation was performed, not with 84 LEDs per headlight, but with 6, each having distinct vectors and emission power. The results achieved are much more satisfactory and close to the real scenario (Figure 2.25).

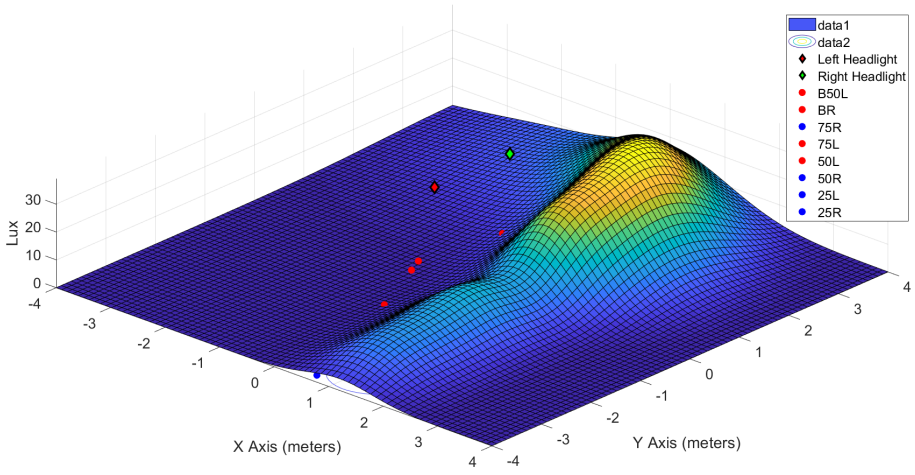


Figure 2.25: Low beam pattern simulation for 6 LEDs per headlight

2.3.2.6 Impact of street lighting

Public lighting will behave as a noise source from the point of view of the VLC outdoor receiver. In order to verify "how much" it will affect, the simulator has been modified in order to take into account its presence.

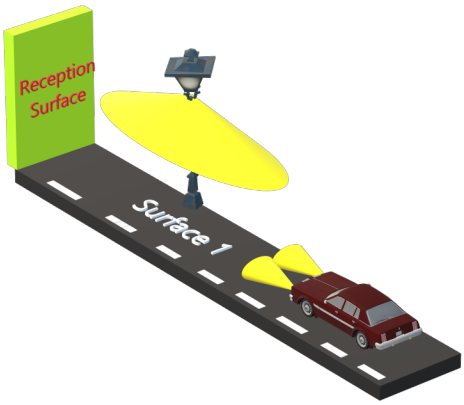


Figure 2.26: Simulation sketch for public lighting noise characterization.

Table 2.7: Public light simulation properties

Half power angle:	Luminou flux:	Vector emission:	Height:
52.5º	6600lm	[1,0,0]	8m

To evaluate the impact of the light noise emitted by the public lighting on the car2x VLC channel, an SNR analysis was performed. For this, as a signal source it was considered the car lighting (with the high beam emission pattern, previously analyzed) and as noise source the light from public lighting (with emission pattern characterized by Table 2.7). In terms of simulation dynamics, the effect of movement was considered by varying the position of the public lighting (at 0m it is placed near the reception surface, and at 25m it is exactly in the position of the car). Road reflection was also considered, having a coefficient value of 0.08 (as previously analyzed). The results of Figure 2.27, show that as expected, the worst scenario happens when the reception system is close to the public lighting source.

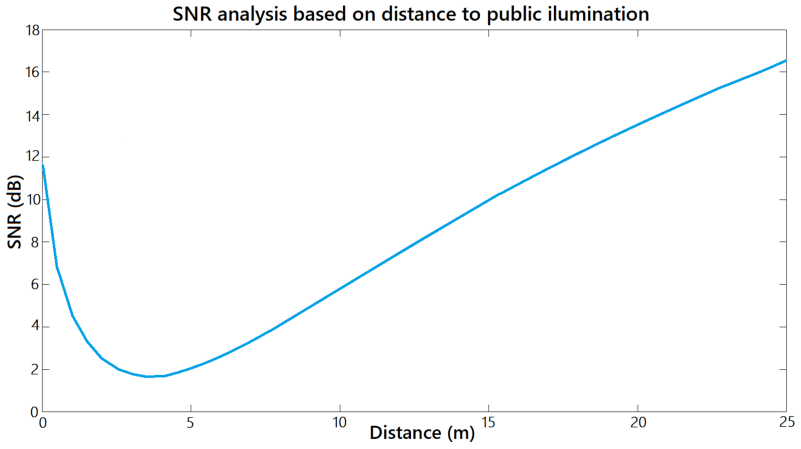
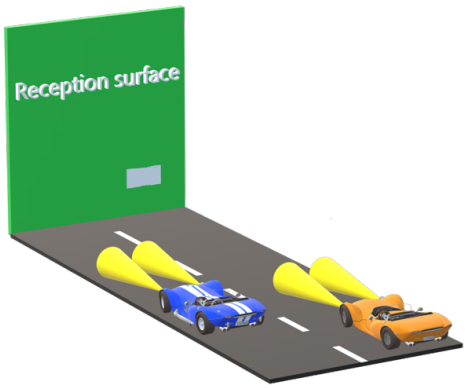


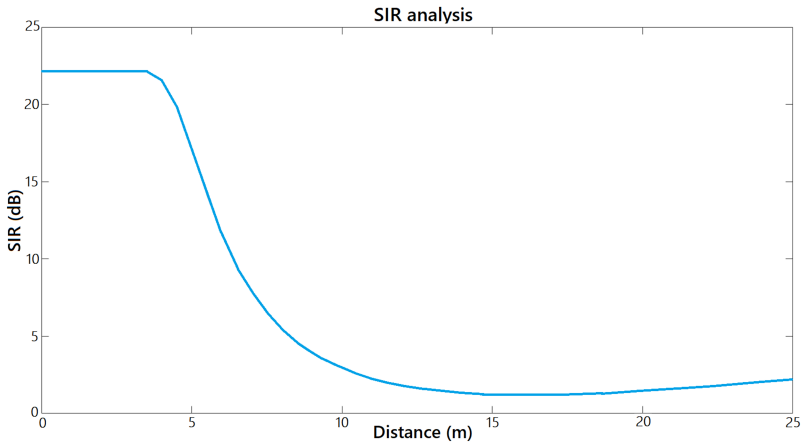
Figure 2.27: SNR analysis for public lighting impact on outdoor VLC.

2.3.2.7 Interference from other transmitters

Another situation that affects the quality of the communication channel is the interference from other users, in this case other cars that also make use of VLC technology. In other words, a RX_a makes a communication link with a determined TX_a, but simultaneous, in the same communication channel, another TX_b is also transmitting. Since the receiver (RX_a) can not distinguish both transmitters, it will receive the superposition of the two signals, where one is in fact the signal of interest, and the other is the interfering signal. The Signal to Interference Ratio (SIR) makes the relation between the intensity level of both signals.



(a) Simulation sketch



(b) SIR results depending on car positions.

Figure 2.28: SIR analysis.

For this, it was considered the situation of two vehicles running in the same direction, in parallel lanes, however only one transmits optical power considered of interest (signal), the light emitted by the other vehicle is considered as interference Figure 2.28a. Both cars are equipped with high beam headlights, and the reflection of the floor was once again taken into consideration (coefficient of 0.08). The region of interest (or reception area) has a dimension of 0.2x0.4m and is located 25m from the position of the car responsible for transmitting the signal of interest. The SIR relationship was obtained by simulating the movement of the car responsible for the emission of the interfering signal in relation to the other that emits the

signal of interest.

This results (Figure 2.28b), are interesting as they show that the highest power intensity of the interfering signal occurs when this transmitter (blue vehicle) is at an intermediate distance between the receiver position and the transmitter vehicle (orange). This is simply due to the emission pattern emitted by the car's headlights, which tend to be designed to have a large radiated surface only at a certain distance from the vehicle and not at the nearest distances.

2.3.3 The receiver: Camera reception

Cameras, like our eyes, have the task of capturing frames from the world around us, convert them into impulse signals, that are then processed. As is well known, the world around us is presented in 3 dimensions, but the image captured by the image sensor (camera) is in 2 dimensions, this will have repercussions on the geometry of the image obtained by the camera. In particular, internal and external camera parameters such as its position, lens types, image sensor size, among others, will affect the projected image.

The pinhole image is a natural optical phenomenon. The earliest historic accounts date back to 500 BC in the writings of Chinese Mozi [36]. At Virupaksha Temple in Hampi, India (700 AD) it is possible to see a real demonstration of the pinhole effect where the image of one of the temple towers is projected (inversely) on a wall of a dark room. The Arab physicist Ibn al-Haytham (956-1039)[37] was the first to make use of this phenomenon to safely project images of solar eclipses into a dark room and study them, those rooms were called as *"camera obscuras"*, *"camera"* in latin stands for *"room"* and *"obscura"* for *"dark"*. Ibn al-Haytham is considered the father of the camera, as he was the first to present a correct geometrical and quantitative analysis of the pinhole phenomenon, and was also the first to present a relationship between the focal point and the pinhole. These experiments and analyzes are considered the foundations for the invention of photography in the XIX century. The use of lenses was introduced in the XVII century and in the early XIX century was developed the first photographic camera, which was based on a dark box type system.

Essentially, the pinhole camera creates an inverted image of the outside world, the geometry is very simple as it is based on triangular geometry, however the image created is quite dark and this is due to the low energy passing through the tiny hole of the camera. The thin lens model represents in an explicit way the geometry behind this phenomenon. A pinhole camera is an idealization of the thin lens when aperture decreases to zero. The focus of the captured image depends on the distance between the ideal lens (tiny hole) and the projection plane Figure 2.29.

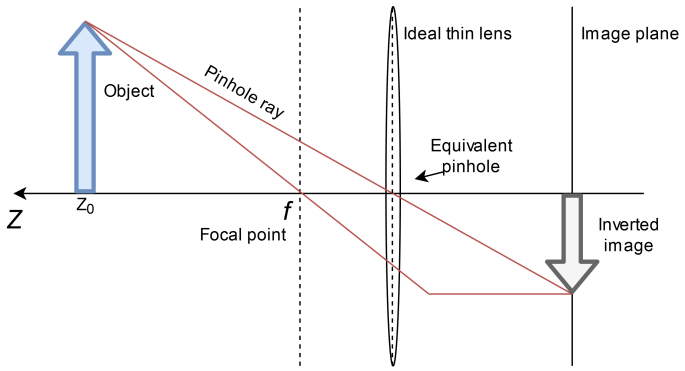


Figure 2.29: Thin lens model geometry.

The consequence of concentrating a 3D view on a photograph (i.e. on a 2D plane) is that one dimension (the depth) will be removed ($(X, Y, Z) \mapsto (x, y), \mathbb{R}^3 \mapsto \mathbb{R}^2$), and this will have consequences: parallel lines with distance will appear to be intersecting, circles will appear as ellipses, among other effects.

Remembering in planar geometry, a point can be represented by its distance to the origin through an orthogonal line to the axis, which creates a (x, y) coordinate, this is called "Cartesian Coordinates". However, that same point belonging to a 2D plane can also be represented by 3 coordinates, by projecting a third coordinate, this is called "Homogeneous Coordinates".

- Cartesian $\mapsto$ Homogeneous:

$$P = (x, y) \qquad \widetilde{P} = (x, y, 1)$$
$$P \in \mathbb{R}^2 \qquad \widetilde{P} \in \mathbb{P}^2$$

- Homogeneous $\mapsto$ Cartesian:

$$\widetilde{P} = (\widetilde{x}, \widetilde{y}, \widetilde{z}) \qquad P = (x, y)$$
$$x = \frac{\widetilde{x}}{\widetilde{z}}, y = \frac{\widetilde{y}}{\widetilde{z}}$$

It is possible to represent the pinhole model of a camera, the thin lens model, through the matrix calculations:

$$\begin{pmatrix} \widetilde{x} \\ \widetilde{y} \\ \widetilde{z} \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.21}$$

$$\widetilde{x} = fX, \widetilde{y} = fY, \widetilde{z} = Z \qquad x = \frac{\widetilde{x}}{\widetilde{z}}, y = \frac{\widetilde{y}}{\widetilde{z}} \tag{2.22}$$

This mathematical representation takes a point in the three-dimensional outer world (X, Y, Z) to which it adds the value 1 to the vector, thus having homogeneous coordinates. Multiplying

this homogeneous vector by the matrix corresponding to the pin-hole model, we obtain the coordinates of the captured image in homogeneous coordinates $(\widetilde{x}, \widetilde{y}, \widetilde{z})$. Thereafter, these image coordinates (in homogeneous form) are converted to Cartesian coordinates $(\widetilde{x}, \widetilde{y}, \widetilde{z}) \mapsto (x, y)$, thereby obtaining the 2D coordinates of the captured frame.

In the previous equation (Equation 2.21), the main matrix can be decomposed in two:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.23}$$

The matrix on the right is the scaling/zooming matrix, depends on the focal length, a larger value will zoom-in the captured image, on the other hand, if the value is small, it will capture an image with a larger field-of-view and consequently objects will appear smaller. The left matrix imposes the conversion from 3D to 2D.

Another way to represent this system is through the Central projection model. Contrary to the Thin lens model, here light does not pass through a lens, but rather is represented as a light ray, originating at the point of the outer (three-dimensional) world and heading towards an imaginary point (the origin), in the middle of the path, it "pierces" the plane of the captured image. This is the model that better represents the camera capture dynamics .

The image plane is usually very small, tends to have a square shape and is confined within the camera. In a digital camera, this image plane is composed of pixel arrays spatially distributed as a grid, when the beam of light coming from point P intersects the image sensor, it will "light up" a particular pixel. So, it is necessary to carry out a transformation of distances from meters to pixels, here it is important to pay attention to the origin of the reference that is usually the upper left point. The center of the image plane is given by (u_0, v_0) according to pixel metrics Figure 2.30.

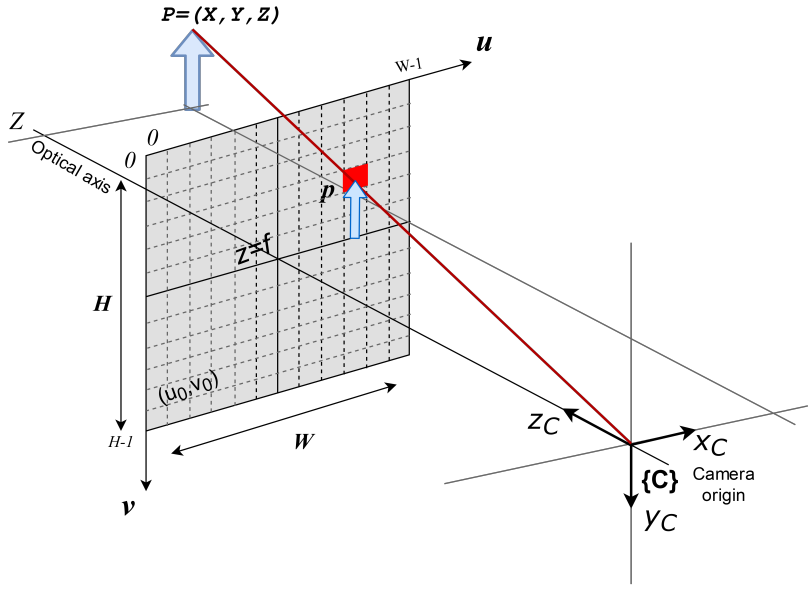


Figure 2.30: Central projection model, based on digital camera image sensor.

As such, the linear transformation present in Equation 2.24 converts the homogeneous coordinates $(\widetilde{x}, \widetilde{y}, \widetilde{z})$ to coordinates $(\widetilde{u}, \widetilde{v}, \widetilde{w})$, also homogeneous but now on a pixel scale and with the origin shifted to the top left corner.

$$\begin{pmatrix} \widetilde{u} \\ \widetilde{v} \\ \widetilde{w} \end{pmatrix} = \begin{pmatrix} \frac{1}{\rho_u} & 0 & u_0 \\ 0 & \frac{1}{\rho_v} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \widetilde{x} \\ \widetilde{y} \\ \widetilde{z} \end{pmatrix} \tag{2.24}$$

$$p = \begin{pmatrix} \widetilde{u} \\ \widetilde{v} \end{pmatrix} = \begin{pmatrix} \widetilde{u}/\widetilde{w} \\ \widetilde{v}/\widetilde{w} \end{pmatrix} \tag{2.25}$$

All transformations (zooming, metrics and reference conversion) can be presented in one equation [38]:

$$\begin{pmatrix} \widetilde{u} \\ \widetilde{v} \\ \widetilde{w} \end{pmatrix} = \begin{pmatrix} \frac{1}{\rho_u} & 0 & u_0 \\ 0 & \frac{1}{\rho_v} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R_{3\times3} & t_{1\times3} \\ 0_{1\times3} & 1 \end{pmatrix}^{-1} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

This equation, also called as Camera model, is composed by 3 matrices. The right vector $(X, Y, Z, 1)$ represents the point coordinate in the outer world in homogeneous coordinates. The first two matrices, already presented here, are known as the "intrinsic camera parameters" and characterize the camera parameters: focus, zoom, reference and geometry of the sensor image in pixels. The third matrix characterizes the camera's exterior arrangement, where R represents a 3 dimension matrix with the camera's rotation parameters and t means the camera's position in space ($[x, y, z]'$). The third matrix is known as the "extrinsic camera parameters".

The image formation can be obtained by multiplying the homogeneous coordinates of the real point by the "Camera matrix" Equation 2.26, which is a matrix that encompasses all camera parameters (both intrinsic and extrinsic). In order to obtain the values of this matrix, it is often necessary to resort to calibration functions.

$$\begin{pmatrix} \widetilde{u} \\ \widetilde{v} \\ \widetilde{w} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.26}$$

$$u = \frac{\widetilde{u}}{\widetilde{w}}, v = \frac{\widetilde{w}}{\widetilde{u}} \tag{2.27}$$

Using the Machine Vision Toolbox from MATLAB®, some simulations were performed in order to better understand the behavior of a digital camera. The intrinsic and extrinsic parameters of the camera are presented in the Table 2.8:

Table 2.8: Camera intrinsic and extrinsic parameters

Camera:	Central perspective
Focal length:	8mm
Pixel size:	(1e-5,1e-5)m
Central point:	(512,512)
Number pixels:	1024x1024
Pose:	t=(0,0,0) RPY/yxz=(0,0,0)deg
Principal point:	Center of image plane

Considering as input the parameters of Table 2.8, the Toolbox calculates the following characteristic matrices:

- Camera intrinsic matrix:

$$C_I = \begin{bmatrix} 800 & 0 & 512 \\ 0 & 800 & 512 \\ 0 & 0 & 1 \end{bmatrix}$$

- Camera matrix:

$$C = \begin{bmatrix} 800 & 0 & 512 & 0 \\ 0 & 800 & 512 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

A circumference was the object considered to be "photographed". It has a radius of $5m$, the center placed at point $(0, 0)$, however, as the tests progressed this position was changed. It belongs to a plane parallel to the $XY plane$ and its height has also been changed according to the needs of the simulation. The camera was at point $(0, 0, 0)$ and has never been changed, however its capture angles have been changed as shown in Figure 2.31.

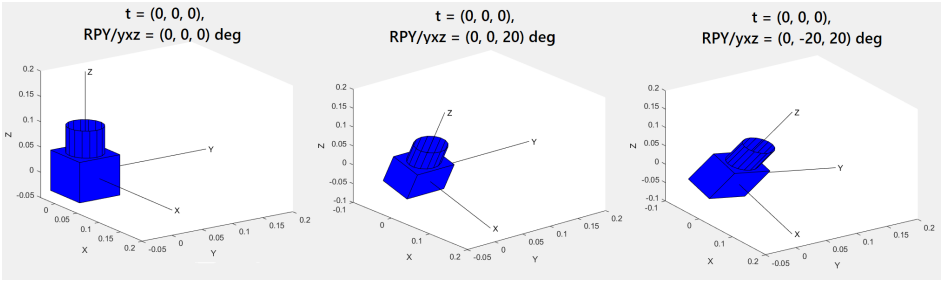


Figure 2.31: Different camera poses during the simulation.

Some simulations were then performed, varying both the physical characteristics of the system (distance from object to camera) and the spatial position of the camera, as well as intrinsic camera parameters (focal length for example). The results are shown in Figure 2.32.

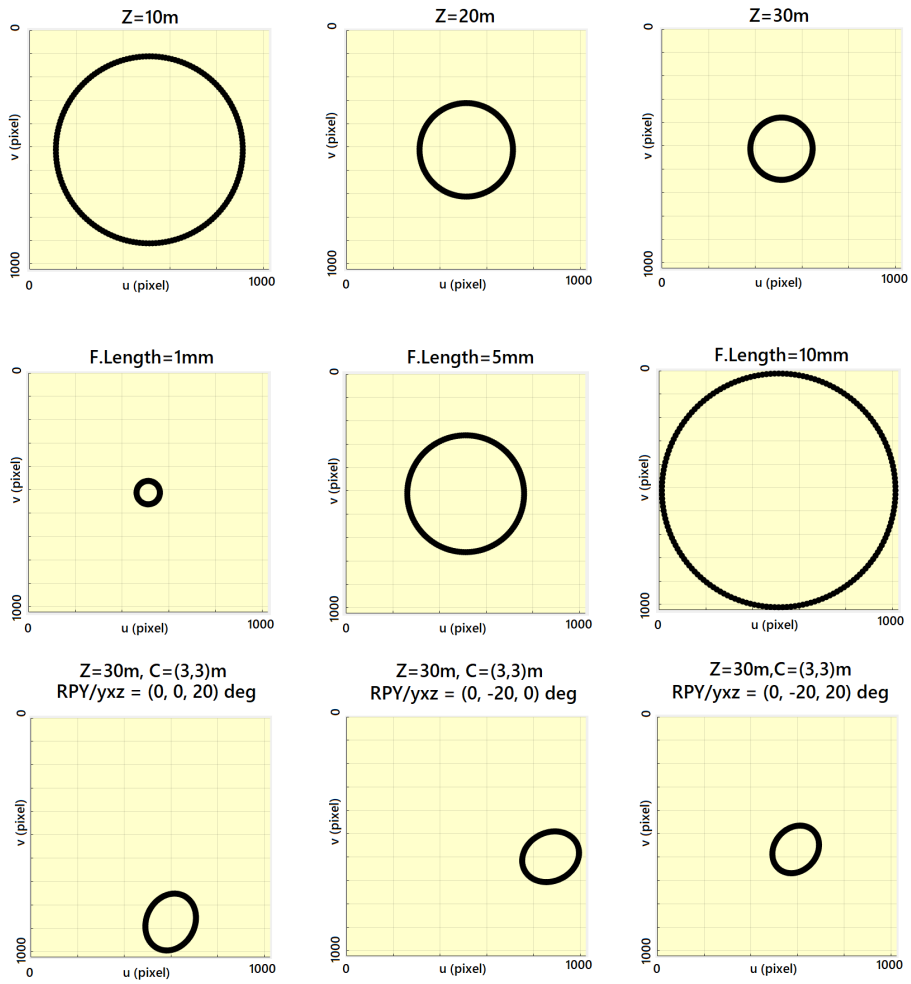


Figure 2.32: Simulation results for different intrinsic and extrinsic camera parameters.

The various simulation results of Figure 2.32, demonstrate that as expected, if the capture object moves away from the camera, the corresponding captured 2D representation also gets smaller. Changing the camera position in relation to the object, creates an interesting result. As already analyzed, the conversion from 3D to 2D dimensions makes that objects with a determined shape, when converted to 2D and depending of the camera position, appear with different one. This is visible in the simulations here presented, a circular shaped object, when captured by the camera in different poses, results in captured figures with dissimilar shapes (elliptical). Also, by modifying the intrinsic camera parameters (in this case the focal length), makes the object appear with different sizes. By increasing the focal length, its imposed a zooming effect, as the results of Figure 2.32 show.

2.4 OCC modulation based on Rolling Shutter effect

Rolling shutter is commonly used in CMOS cameras. This frame-capture method is characterized by not exposing the entire sensor (photosensitive cell matrix) simultaneously to light, but by exposing smaller parts (usually arrays or sensor rows) over time. Another capture

method is the Global Shutter, which exposes the entire sensor, i.e. the entire matrix, at the same instant. One way to understand the rolling shutter method is to imagine a slit sliding along the top-down sensor where only the area enclosed by that slit is exposed to light (from a engineering point of view, the camera driver selects which array of cells collects light in that instant of time), this means that the upper part of the sensor does not capture light at the same time as the lower part of the sensor, and this is the main characteristic of this method.

Global Shutter cameras generally use Charge-Coupled Devices (CCD) instead of Rolling Shutter cameras that use CMOS sensors. There are several reasons for choosing one or another type of sensor. There is a trade off between several reasons, from process speed, power consumption, system complexity, application type and cost. Global Shutter sensors tend to have a slower processing speed when compared to Rolling Shutter sensors.

2.4.1 OOK modulation

One of the simplest and most commonly used modulation in OCC technology is to flash the transmitter LED, i.e. by polarizing it directly and reversely, for short periods of time. This type of modulation also used in other systems, is known as On-Off Keying (OOK). The light intensity radiated by the LED is modulated by the current flowing through it. The OOK modulation makes use of this property, and considers only two possible states: state *OFF* or "*0*", with a defined current level, and state *ON* or *1*, with a current level higher than the previous one. Normally, the state *OFF* is applied with LED effectively off (i.e.,the current flowing through it close to 0A), and the state *ON* is applied by forcing a current thought the LED high enough to produce the necessary emitted radiation.

However this technique faces some important problems:

- **Luminous intensity radiated by the transmitter (LED)**: When transmitting consecutive ones and zeros, OOK will suffer from unbalanced light intensity provided by the LED. Considering a time period of 10 symbols, if 8 consecutive ON symbols and 2 last OFF symbols are transmitted, the average brightness will be higher than the inverse case of 8 OFF and 2 ON symbols, which means that in these two time periods the emitted brightness by the LED would be different. Any type of lighting device, must radiate a constant luminous intensity, at least during the minimum time period perceptible by the human eye. Otherwise, it would be visible to the naked eye, the light source oscillating its luminous intensity.
- **Clock signal**: Another problem derived from the previous point is that when transmitting more than one symbol of the same type (ON or OFF) the clock signal inherent to the transmission is lost. Some adaptations to the modulation are possible, it is the case of OOK-Manchester, where there is always a intensity level transition for each symbol, and consequently the receiver can always remove the clock signal. In addition, the signal now has a constant average intensity level over time, which also corrects the phenomenon of brightness oscillation presented in the previous point.
- **Oscillation frequency limits**: Since one of the purposes of VLC communications is maintain the symmetry between transmission of data and lighting, the comfort of

the light produced by the transmitting devices is a major concern when designing the system. The parameter that regulates the minimum detectable limit of oscillation of a light source by human vision is called Critical Flicker-Fusion Frequency (CFF). Its value depends on different factors [39], from ambient light level, exposure time, if the user uses one or both eyes, among others, the minimum value is close to 36Hz, however some people report feeling uncomfortable at values below 80Hz.

2.4.1.1 Undersampling-based modulation

As already mentioned, the OCC system design requires consideration of human perception of light source flicker. The commercial camera's normally perform a low frame rate (usually 30 FPS), this makes impossible the implementation of a transmission frequency equal to the reception frame rate, as the human user would perceive the flicker effect. With the purpose of create a flicker-free communication, undersampling-based modulation methods were proposed to ensure no perceptual intensity fluctuation. The camera's frame rate is lower, but the image sensor has a cutoff frequency higher ($\approx 1000Hz$) than the human flickering perception ($\approx 100Hz$). This means that a signal based on undersampling modulation can be captured by a camera with the appropriate exposure setting, and not perceived by the human eye.

Undersampling Frequency Shift On-Off Keying (UFSOOK)is a form of direct current differential coding [29], similar to Frequency Shift Keying (FSK), different ON-OFF keying frequencies indicate the transmitted bits. In UFSOOK the two possible bits (logic 1 and 0) are modulated into two different frequencies (both higher than the human flickering threshold). The sampling rate is defined by the camera frame-rate, and the sampling period is defined by the camera's exposure speed. The idea is to sample the signal waveform in a precise instant such that during the transmission of the "logic 1 frequency" the captured frame will see the LED ON, and during the "logic 0 frequency" it will see the LED OFF. To guarantee this effect, the two frequencies are projected such that the "logic 1 freq." is $f_{camera} \times (n \pm \frac{1}{2})$ and the "logic 0 freq." is $f_{camera} \times n$, where f_{camera} is the camera frame-rate, and n is an integer value. The Figure 2.33 represents a pattern example for this type of modulation. In this technique, two transmitted frames samples represent one bit

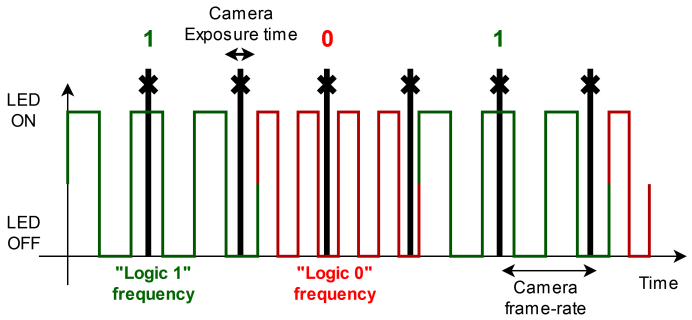


Figure 2.33: UFSOOK pattern example.

Undersampling Phase Shift On-Off Keying (UPSOOK) modulation is similar to the well known Phase Shift Keying (PSK)[29], where the logic bits are modulated not throught different frequencies, but instead by changing the phase of the corresponding carrier signal.

2.4.2 Rolling shutter effect-based modulation

Commercial cameras usually make use of CMOS image sensors with rolling shutter. In this technology an array of pixels is used to capture the incident light in a progressive manner by exposing each row of pixels to the light rays. Rolling shutter image sensors, when illuminated with high frequencies and with relatively low exposure time tend to form different illuminated bands indicating the "ON" and "OFF" status of the incident signal **??**. The voltage at each individual pixel is given by:

$$v_P(t) = \frac{A}{C_{PD}} \int_{t-T_{exp}}^{t} \Re \cdot x(t) d(t) \tag{2.28}$$

Where A is the gain, C_{PD} the equivalent capacitance and $\Re$ the sensor responsivity. The received optical signal for this individual pixel at instant t is represented by $x(t)$. Important note that the exposure time must be higher or at least equal to the symbol period ($T_{exp} \geq T_{sym}$). The number of symbols captured depends on the resolution, exposure time, area of the ROI, among other parameters. The system response is given by:

$$h(t) = \frac{A}{C_{PD}}(u(t) - u(t - T_{exp})) \tag{2.29}$$

where $u(t)$ is the unit step function. Converting to the frequency domain, results in a low-pass-filter effect :

$$H(f) = \mathcal{F}\{h(t)\} = \frac{AT_{exp}}{C_{PD}} \frac{sin(\pi f T_{exp})}{\pi f T_{exp}} e^{-j\pi f T_{exp}} \tag{2.30}$$

As can be verified, the DC gain is proportional to T_{exp}, increasing the T_{exp} will reduce the cut-off frequency, this means that a trade off between the gain and the required bandwidth must be accomplished.

It is now important to highlight two characteristics of video/photographic cameras. The first is the Camera Sampling Rate, this property controls the camera's Frames per Second (FPS), or in other words, the number of frames the camera is capable of capture in a 1 second. High-frame-rate cameras contribute to high-data-rate communications. Although, high-frame-rate cameras are usually more expensive when compared to standard cameras with frame rates of 30FPS's normally. Another important feature is the Shutter Speed, this property indicates the total amount of time that the sensor has been exposed to light. The shutter speed should be more than twice the frame rate for secure communications. Considering a camera frame rate of 30 FPS's, this means that the time reserved per frame is $33ms$, this means that the shutter speed should be at most (or not exceed) $16.5ms$, to ensure that the requested frame rate is respected and that enough time is allowed for the camera Graphic Processing Unit (GPU) to process the data burst generated by each capture.

Since, most of the currently available commercial cameras have low frame rates, researchers have been forced to study some undersampling techniques in order to communicate using low-frame-rate cameras. Undersampling, as already explained, means that receptor samples the transmitted signal at a lower frequency than the transmitted. In this technique, making use of the rolling shutter phenomenon, becomes an advantage.

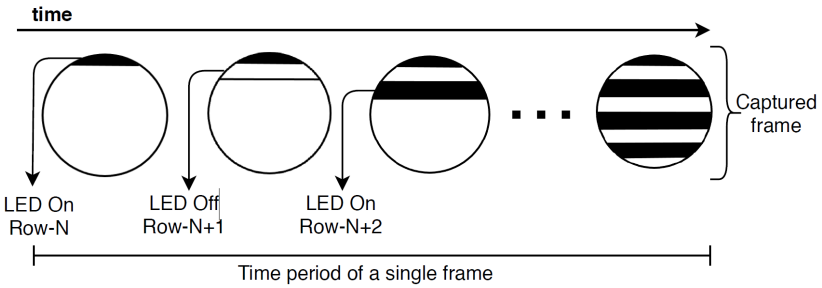


Figure 2.34: Rolling Shutter capture method.

When the OOK modulation technique is applied to support the transmission, it has two possible intensity levels: ON and OFF. This consequently results in dark and bright stripes [18] in the LED image captured by a Rolling Shutter sensor Figure 2.34. The number of stripes and their thickness are used to carry the transmission data [40]. The number of stripes varies proportionally with the size of the light source. However, the thickness of the stripes depends strictly on the ON and OFF frequencies of the LED. Different frequencies result in different thickness stripes. The width of the bright and dark stripes can be expressed by Equation 2.31:

$$S = \frac{1}{f \cdot t_r} \tag{2.31}$$

In this equation, t_r represents the time needed to read-out a single row of photosensitive cells in the image sensor, and f means the LED flickering frequency. Assuming that each sensor cell row is exposed to light for the same amount of time and that transmission follows a uniform and constant pattern (in other words, it presents a square wave transmission of "1010101"), the total number of detectable stripes during an exposure is given by:

$$n = 2 \cdot f \cdot t_e, \ [t_s \geq 2f_r] \tag{2.32}$$

Here, t_e denotes the time of single exposure for a particular frame and t_s indicates the shutter speed, that should be at least two times the transmission frequency to respect the Nyquist theorem.

The number of stripes depends on several factors as already noted, however other extrinsic factors such as LED size, communication distance and area of each photosensitive sensor cell, will affect the number of stripes captured in the image [40]. Equation 2.33 presents the total number of stripes for a circular-shaped LED:

$$n_s = \frac{A_i f_o^2 (f_{on} + f_{off}) t_r}{\rho^2 d^2} \tag{2.33}$$

As can be verified, the value depends of the effective area of the light source A_i (i.e. the projected area on the image sensor), the focal length f_o, the pixel edge length ρ and the Euclidean distance between transmitter and camera position d.

It can be concluded that the width of bright or dark bands is proportional to the symbol rate of the transmitter and the rate of the image sensor to capture preview images. Using

image processing techniques, these bands can be converted into binary arrays, and extract the corresponding data. Although, other modulation techniques can be employed, by using ON-OFF stripes with different frequencies corresponding to different modulated symbols. This way, its applied a principle similar to the Frequency-shift keying (FSK) modulation, which digital information is transmitted through discrete frequency changes of a carrier signal. Figure 2.35 shows these two different modulations techniques using rollings shutter effect.

Figure 2.35: OOK and FSK modulation using the rolling shutter effect.

2.4.3 MIMO functionality

Typically, there are two optical MIMO approaches for VLC, namely, non-imaging and imaging MIMO. In non-imaging MIMO systems, each receiver collects light from all sources simultaneous and after that, by signal processing, distinguishes the different transmitters. Main example are the photocell receivers and multi color modulation. On the other side, imaging MIMO systems employ an imaging diversity receiver structure to distinguish the light from different transmitters. Since the purpose of this work is to apply image sensors receivers, it makes sense to analyse the imaging MIMO systems.

As already explained, the image sensor is composed by a matrix of independent photosensitive cells. In partnership with the lens, optical signals from different locations within the camera's Field of View (FOV) will be projected at different coordinates of the image sensor and will therefore be processed independently and uncorrelated. This allows the implementation of a imaging MIMO system.

Figure 2.36: An example of MIMO application in car traffic.

Figure 2.36 presents an example of VLC imaging MIMO application. The photograph can be considered as an instantaneous frame captured by a camera. In this "frame" it is visible

three different cars, that represent three different communication links with the camera. Here is the first advantage of this technique, which is the possibility to simultaneous receive and differentiate data from different transmitters, by employing AI algorithms that allow the recognition and detection of cars into image frames. The second advantage is the possibility to differentiate Region of Interest for the same transmitter, in Figure 2.36, for CAR 2 it is possible to recognize two ROI (the two taillights). Making use of this feature, the data rate can be improved by applying modulation techniques that make use of this spatial characteristic and so, modulate the symbols independently for different ROI. With this, the available coding symbols increase exponentially for each added taillight.

2.5 Concluding remarks

In conclusion, the VLC-OCC system has several advantages, from being a type of communication not yet regulated (in terms of restrictions of the electromagnetic spectrum), combining illumination with the transmission of information and the fact that most of the infrastructures necessary for the diffusion of this technology already exist. However, it also faces several obstacles, the main one being the communication channel behavior. Application as a solution for communication between vehicles is possible, and here, reception by image sensors is fundamental in order to obtain satisfactory results both in terms of distances and resilience to phenomena that impair the system's SNR.

CHAPTER 3

Implemented system

The system implemented, as already presented, is based on two distinct complementary technologies: VLC and OCC. Since the purpose of this work is to find a solution for a communication link between vehicles, the use of LED headlights or taillights, which equips any modern vehicle, is the most obvious choice. From the reception point of view, a digital camera is the key element, however, there are some restrictions that are important to take into account, such as: hardware specifications, firmware support and freedom to configure properties of the camera.

3.1 V2V application (Intelligent Transport System)

The purpose of ITS is to create an intelligent network that optimizes traffic efficiency as well as the safety of all users. Also comfort is one of the purposes of this concept, delivering useful and real-time information to the user, makes it possible to make decisions that prioritize their safety and comfort. The ITS system, although closely associated with driving aids, extends to many more areas, from traffic control in Traffic Management Centers (TMCs), emergency care, assistance and protection of high-risk users (pedestrians, childrens, cyclists, among others). ITS is not only restricted to land transports, it also applies to air, rail and sea transports, but will not be subject of study in this work.

Studies show that on average 40% of the population spends daily 1 hour on the road [41], this shows that transportation systems are a fundamental part of every citizen's life. Due to this brutal influx, the traffic of vehicles is becoming increasingly congested. According to 2010 numbers, the United States had on average 797 vehicles registered per 1000 people, Portugal, which ranks 26th position in the table, has 548 vehicles per 1000 people. It is expected that in the short term, these numbers will continue to grow. Unfortunately, and as expected, the risk of accidents also rises. In 2017 [42], in Portugal, there were 130,000 road accidents, which caused 509 deaths and more than 44,000 injuries. According to experts, the responsibility for such numbers is distributed from poor road condition, mechanical problems of vehicles, lack of supervision, and other reasons. Nevertheless, all are unanimous that much of the

responsibility for such numbers is due to the human factor. Several strategies have been developed and applied to solve this problem. An example of this is the implementation of new transport policies that restrict vehicle access to certain urban areas, the investment in new infrastructure that through the construction of new roads or reconstruction of existing ones, and the third, is the optimization of the existing infrastructures.

The purpose of ITS is the analysis (based on computational algorithms) of the data collected by the various auxiliary systems (cameras, GPS, sensors, Radars, etc), act on the transport system in order to prevent the occurrence of accidents or other types of congestion.

The ITS system is divided into 6 distinct application areas [41]:

1. Advanced Traffic Management System
2. Advanced Traveler Information System
3. Advanced Vehicle Control system
4. Advanced Public Transportation System
5. Advanced Rural Transportation Systems
6. Advanced Commercial Vehicles Operations system

Whether the functions of these components can fully be realized depends on how data is collected and processed into useful information.

The work process/application of the ITS system methods is based on [43]:

- **Data collection**: This is the fundamental pillar of the whole system. Data is collected via different types of hardware devices, from sensors embedded in vehicles and infrastructures, GPS, cameras, surveillance systems, among other forms. This information is critical for proper and accurate action.
- **Data Transmission**: For efficient action, responsiveness is one of the key points. Different types of transmissions are present in various phases and areas of the system. First of all is the communication between the data collection systems and the processing servers, the next is the communication between the same servers and the actuators, in addition there is still communication between the various individual users. Different techniques are already used (internet, LTE, Bluetooth, Dedicated Short Range Communication (DSRC), among other types of telecommunications), but many others may be applied in the future (such as VLC-OCC).
- **Data Analysis**: All the collected and received data must be sampled, error rectified, synthesized and applied to adaptive logical analysis systems. Post-statistic analysis or AI are used to predict scenarios and act timely. Much of this process is made in large cloud based servers. Small microcontrollers scattered across the ITS system can do much of the processing or ease server load through edge processing techniques.
- **Traveler/User Information**: One of the main methods of action is to inform, advise or even directly impose guidelines for the user to make the best decisions for himself and all users of the system. One such example is Travel Advisory Systems (TAS), whose purpose is to inform traveling users of updates. This system delivers real-time information such as travel time, travel speed, delays and accidents on roads. Information can be delivered through various forms, as mentioned above.

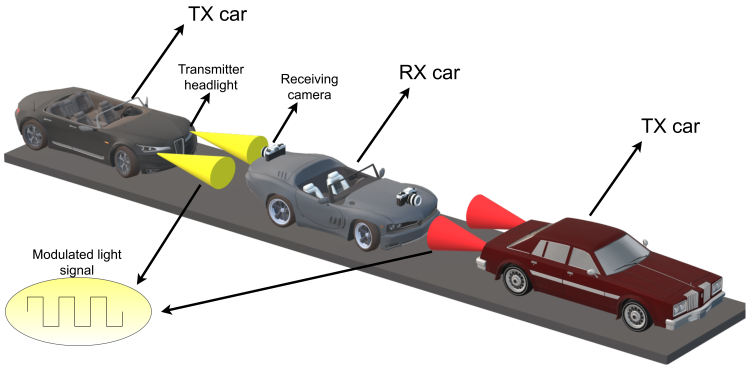
In conclusion, the ITS system is very comprehensive and dispersed, covering everything from traffic management to commercial vehicles operation control. The purpose of this work is to focus on a narrower area that can be included in both the Advanced Traveler Information as in Advanced Vehicle Control System areas and is considered a Data Transmission process, which is the communication between vehicles (V2V) or between vehicles and infrastructures (V2I).

3.1.1 Optical V2V and V2I overview

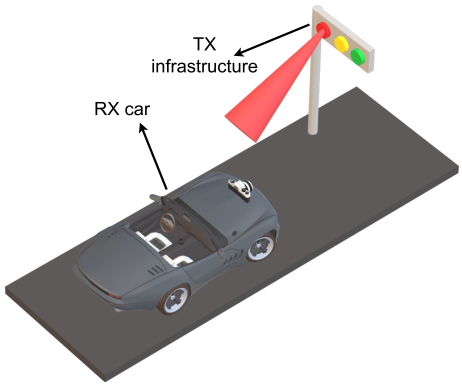
V2V communication is one of the key sectors for transmitting information between road users and thus strengthening data exchange within the ITS system [19][20]. It is considered one of the most unanimous solutions to increase road safety and traffic efficiency. Some solutions are already in place is the 5.9 GHz DSRC[44] with a reserved bandwidth of 75 MHz. This reserved and regulated band is based on IEEE P1609.x/D5.8 communication protocols and is presented as being a solution with high data rates and low latency. However, when applied to a scenario with high number of users, this communication technology tends to present low packet reception rates and long delays. Besides that, the identification and location of the transmitter by the receiver is difficult in populated environments.

In Figure 3.1a is presented a V2V system, where two possible implementations are used (with the rear or front lights), the information to be sent by the TX vehicle is carried on the emitted light using modulation techniques, and later received by the camera, which is properly positioned in the RX vehicle. This type of transmission is characterized by offering a one-way communication link (TX$\rightarrow$ RX), so if the receiving vehicle needs to answer to the transmitter, the only solution (using the same type of VLC-OCC technology) is implementing the reverse system. In addition, and as it is known, the OCC is a LOS communication, which implies that for a successful transmission, the car headlight must be in the receiver camera's FOV. V2I is very similar to the V2V system[45][4], only differing in one point, communication takes place between the vehicle and the infrastructure present on the roads (traffic lights or other sources). While in the V2V scenario the transmission of data is restricted between vehicles, in this case the vehicle communicates with an infrastructure that is connected to the core of the ITS network. High-priority alert communications from top network authoring layers (for example a TMC) must at some point of the communication root pass by this type of transmission.

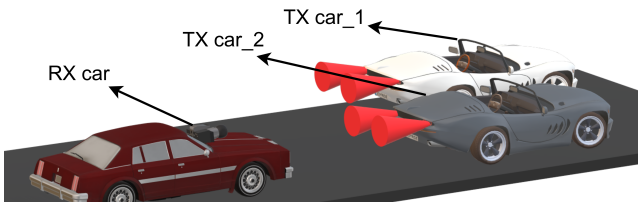
Another interesting possible transmission, already discussed in subsection 2.4.3, is the MIMO solution. Figure 3.1c exposes an example of this situation, where two cars transmit simultaneously information through their taillights to the car that follows them. The only imposition is the fact that the transmitters must be into the receiver camera field of view.



(a) Vehicle to vehicle transmission.



(b) Vehicle to infrastructure transmission.



(c) Vehicles to vehicle transmission (MIMO implementation).

Figure 3.1: V2V different implementations.

3.2 System conceptual architecture

The practical system implemented is based on the guidelines studied in section 2.3. It is divided into two distinct sectors, these are the transmitter and the receiver. Figure 2.6 from section 2.3 must be remembered, there is presented these two sectors (taillight module as transmitter and camera system as receiver). In the process of choosing and adjusting both blocks, factors such as complexity, resilience to external disturbances, freedom of configuration/adjustment, size occupied by the setup, power consumption, cost and proximity to the real application scenario were always taken into account.

3.2.1 Transmitter architecture

The purpose of the transmitter system is to convert the input data into optical symbols by a modulation technique. This way, is possible to transmit through the OWC the information from the transmitter side to the receiver. Figure 3.2 presents the logic behind this conversion that will be analyzed in detail over the next sections. The input digital buffer is truncated into parts of three bits. Each of this package of three bits corresponds to one and only symbol (from a set of eight possible choices). The symbol, that is explained in detail in the further subsubsection 3.2.1.1, is composed by two independent sectors (upper and bottom). By imposing a square wave signal with an uniform frequency through each of this sectors is possible (by the Rolling Shutter Effect) to visualize in the capture frame a symbol composed by stripes in the upper and bottom regions.

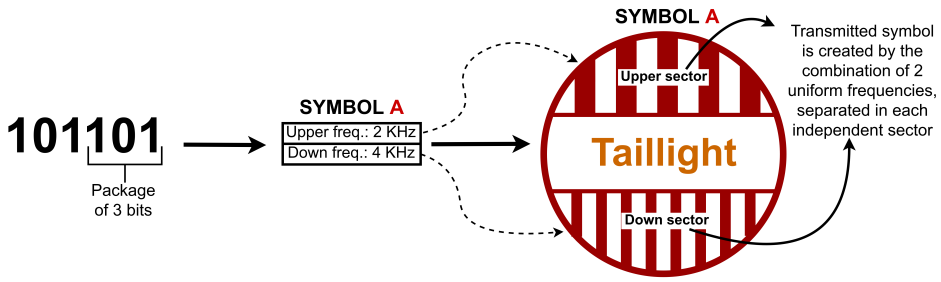


Figure 3.2: Conversion from digital bits to optical symbols logic.

The transmitter consists of three main blocks (Figure 3.3):

- **Microcontroller**: This component acts as the "brain" of the transmitter, its purpose is to convert high level input signals (bits) into control signals that properly activate the driver controller. In addition, it imposes the clock signal for the entire transmitter system, as well as data processing before transmission.
- **Driver**: This electronic circuit has the objective of regulate and control the light emitted by the LED matrix by modulating the current delivered to it. It acts according to control pulses sent by the microcontroller. The driver circuit consists of integrated circuits (Op-Amps, oscillators, multiplexers, etc), transistors, capacitors, among others.
- **Car rear light (LED matrix):** This component is of high importance, since it is the one that effectively emits light and information into the communication channel. It was chosen a properly UNECE homologated automotive LED tail light. Thus the actual outdoor practical implementation condition is always safeguarded.

These three parts, together, constitute a system that converts input data (bits) into optical signals to be carried by the free-space optical channel.

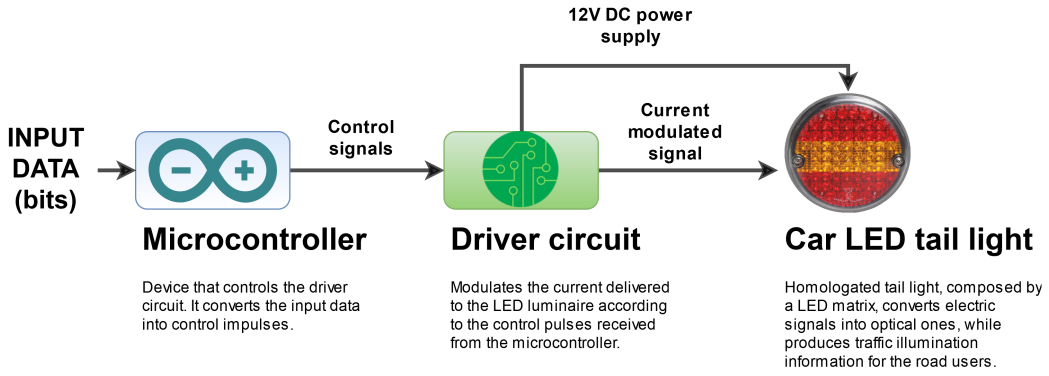


Figure 3.3: Transmitter scheme.

3.2.1.1 Transmitted symbols

The choice of shape, appearance and technique used for the creation of symbols to be transmitted by the taillight was a challenge. Several factors come into the "equation", its complexity affects both the transmitter side (from the point of view of information modulation to apply to the signal), as well as the difficulties imposed on the receiver for its correct decoding. Since one of the objectives of this work has always been to show the system's feasibility, applying a too complex information modulation technique could improve the transmission data rate, but would certainly restrict the conditions of application of the system. The main idea was always to show its application in a real outdoor world, and as such, the chosen way was to make make the solution as simple as possible.

The shape and arrangement of the LEDs in the taillight (Figure 3.11) has an interesting geometry, when the stop or presence signal is active, the red light is emitted simultaneously, but in two physically separated zones. From this, came the idea of modulating each zone independently, and thus creating a symbol that was the spatial conjunction of the two distinct areas.

Three modulation frequencies were then defined: 2KHz, 4KHz and 6KHz. The distance band between the different frequencies was defined as 2KHz so that the differentiation between them was as easy as possible to verify. Also the band chosen between 2KHz and 6KHz was determined, since below 1KHz the stripes would become too thick and it would be possible that in a given capture instant the stripes would "fall" outside the ROI bounded by the taillight area, as well as above 8KHz the stripes become too thin and difficult to differentiate. At this point, there are 3 different arguments (frequencies) for two distinct zones, where spatial arrangement (or order) matters for symbol differentiation. This means that we have $3^2 = 9$ possible symbols. It should be noted that this number can be maximized in two possible ways, either by increasing the range of frequencies that can be used, or by increasing the different and independent available taillight zones. Figure 3.4 shows the 9 different symbols used for modulating the information to be transmitted. The frequencies corresponding to the different symbols and their arrangement according to the two zones are shown in Table 3.1.

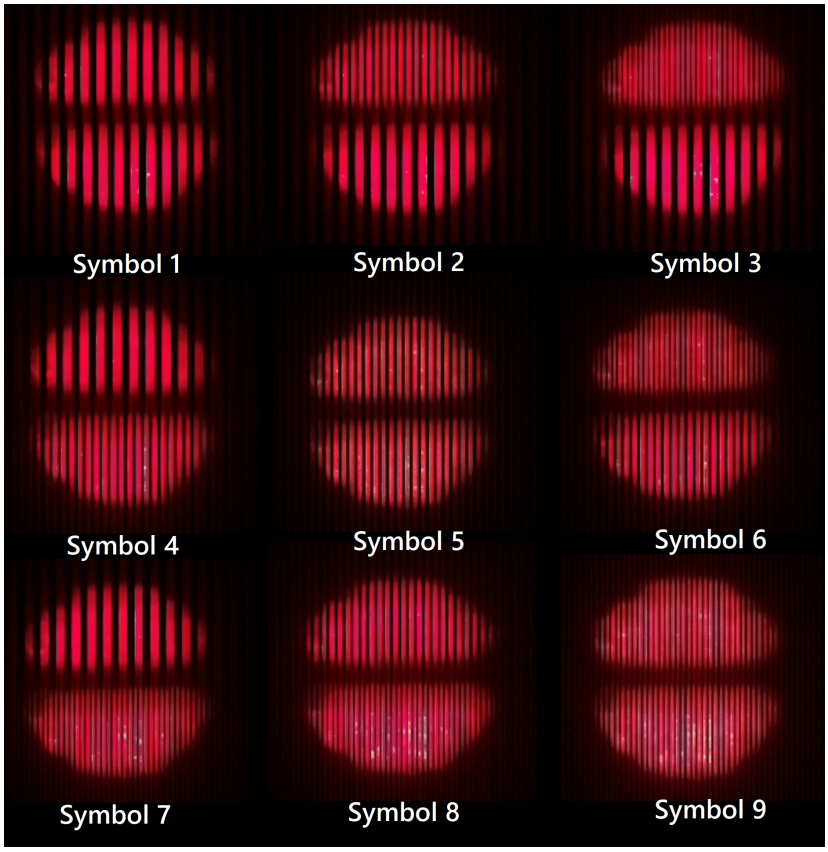


Figure 3.4: Different symbols used for data transmission.

Table 3.1: Symbols and corresponding frequencies.

Symbol:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Upper freq. (kHz):	2	4	6	2	4	6	2	4	6
Lower freq. (kHz):	2	2	2	4	4	4	6	6	6

3.2.1.2 Microcontroller

The microcontroller used was a Arduino Nano[46]. A small, complete, and breadboard-friendly based on the ATmega328 (Arduino Nano 3.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable. Table 3.2 presentes some main specifications of this board.

Table 3.2: Arduino Nano ATmega328 specs

Operating Voltage:	5 V
Flash Memory:	32 KB
SRAM:	2 KB
Clock Speed:	16 MHz
DC Current per I/O Pin:	40 mA
Input Voltage:	7-12 V
Power Consumption:	19 mA

The firmware was divided into two independent parts. In the main function, which corresponds to a infinite while loop, the data is received via UART (or serial port) service, the information received (bits array) is stored into a buffer, so that, it can be accessed while other one is processed. Information is transmitted to the free-optical space via modulated symbols (subsubsection 3.2.1.1). The system was designed to transmit one of nine (possible) symbols per frame, one of the symbols is reserved for synchronization procedure between transmitter and receiver, thus leaving 8 possible symbols to carry information. With 8 symbols, it is possible to encapsulate 3 bits. The "$Ready\_flag()$" control signal is used by the comparator unit (which converts the packet of bits into symbols) to request the buffer for the next 3 bits packet.

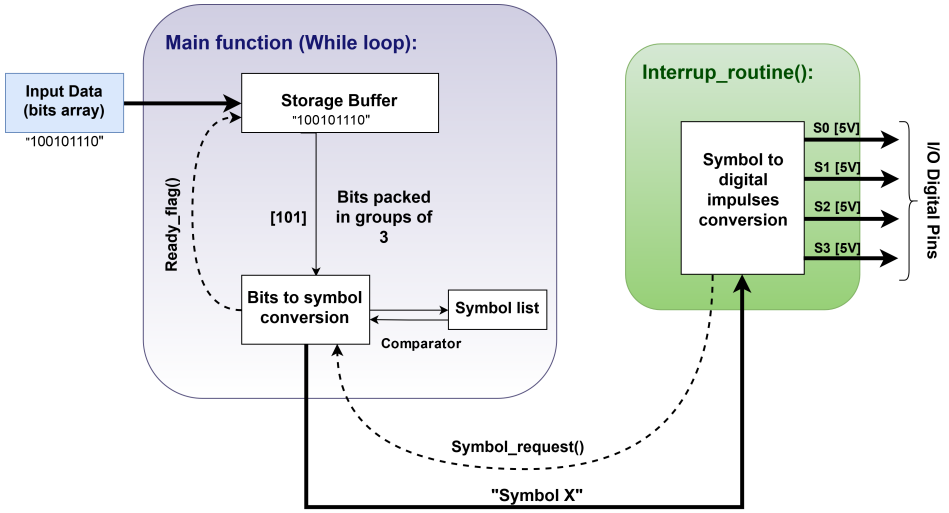


Figure 3.5: Microcontroller work-flow.

The second part of the software is an interrupt handler that updates the output of the I/O pins according to the symbol to be transmitted. The advantage of inserting this part of the code into an interrupt routine, is that the attendance is as accurate as possible, since the microcontroller processor is forced to attend this handler in the exact moment that it occurs. This ensures (or as close as possible) the clock accuracy of the transmitted signals, which is calibrated by the frequency set by the interrupt handler. An interrupt flag is generated through an increment process that compares the value of a user-configured register with an internal register that increments based on Arduino clock (16 MHz). Thus, when the interrupt flag "awakes", the microcontroller processor pauses the attendance of any other instruction and answers the code present in the interrupt handler. This code consists in requesting the main function for the symbol to be sent in the present frame, and then configuring the Arduino's digital ports accordingly.

In section A.9 its presented the Arduino code used for random symbol generation and corresponding driver control.

3.2.1.3 Driver circuit

The driver's electronic circuit is responsible for controlling the car LED tail light according to control signals from the microcontroller. It acts over the LED tail light, changing the lighting patterns to emit, by modulating the current flowing throw it. In addition, it is also responsible for conditioning the signal from the microcontroller and converting it properly to act on the LED's tail light matrix input.

Prior to the driver development, some considerations were taken:

- Qualitative and quantitative evaluation of the input control signals versus the output signals to be generated.
- Available components.
- Power evaluation (consumption and power to be delivered to the LED matrix).
- Portability, size and other logistical considerations.
- System specifications (bandwidth, jitter, noise imposed on signals throughout system propagation, among others).

The driver must act according to the distinct and independent zones of the LED matrix. The LED taillight was designed by the manufacturer to support voltages of the order of 12V and powers up to 4.8W when in the ON state (subsubsection 3.2.1.4). Therefore, the power to be delivered to the LEDs would necessarily be an imposition to take into account. In addition, the entire circuit should be projected to be powered with a DC 12V voltage, as this is the voltage provided by a car battery.

The analog logic of the circuit can be verified in the Figure 3.6.

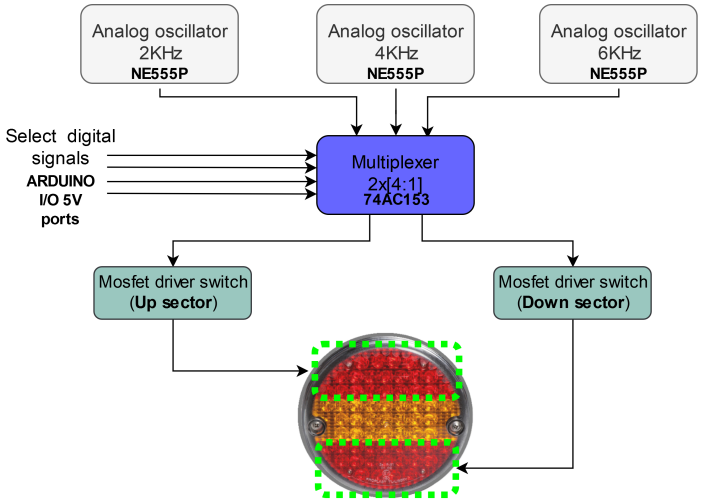


Figure 3.6: Electronic driver logic.

Looking at the driver by parts, the first part of interest are the oscillators. The circuit consists of 3 independent analog oscillators. Each with a different operating frequency: (2, 4 and 6KHz). Each of the analog oscillators makes use of an NE555P integrated circuit. The choice of this IC was due to the fact that it was an available component in the laboratory storage. The circuit was designed as specified in the manufacturer's datasheet to generate an a-stable mode of operation (Figure 3.7).

Analyzing the datasheet, two functions stand out:

$$Frequency \approx \frac{1.44}{(R_A + 2R_B)C} \tag{3.1}$$

$$Low\ to\ high\ ratio = \frac{t_L}{t_H} = \frac{R_B}{R_A + R_B} \tag{3.2}$$

The purpose of these oscillators is to generate a uniform square wave, with a duty cycle close to 50%. However, according to the previous functions, it is found that this is impossible (it would require R_A to be 0Ω and thus, short-circuit Vcc to the Discharge Port of the integrated circuit). This is because the triggering capacitor (C) charges through $R_A + R_B$, but only discharges through R_B. For a duty cycle of 50% the charging and discharging currents should be the same, which is impossible, since in this circuit the duty cycle will always be greater than 50%. There are some adaptations that through sourcing/sinking directly from the output port can achieve a 50% duty cycle, however, the output signals from the oscillators are needed as inputs for others driver ICs, which require an input current to properly operate, so, lowering this current is not advisable. Thus, knowing that the signals to be generated do not necessarily demand a duty cycle of 50%, but a signal with a constant period, getting a duty cycle close to 50% instead of exactly 50% is acceptable.

Therefore, and following the configuration of Figure 3.7, R_A and R_B were applied by a $100K\Omega$ potentiometer, and C was regulated by the addition of parallel capacitors until the required frequency was obtained.

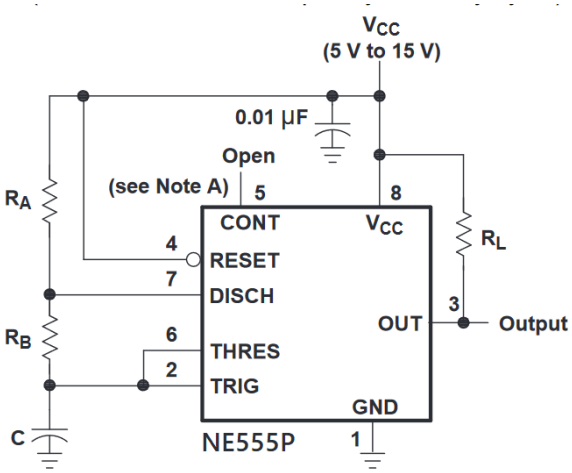


Figure 3.7: Oscillator scheme.

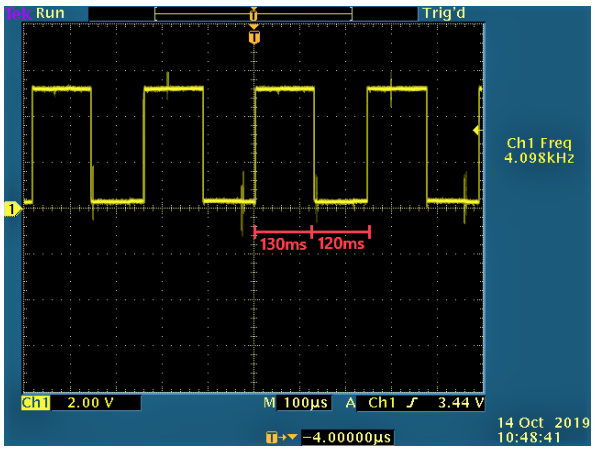


Figure 3.8: Output of 4KHz oscillator.

Figure 3.8 shows a print screen of the output signal of one of the three oscillators, in this case, from the 4KHz one. It is noticeable that the duty-cycle is not 50%, but instead 52%, as explained above. The operating frequency is not exactly 4KHz, this is due to adjacent disturbance sources, largely due to the fact that the circuit was mounted into a breadboard, and inherent to this type of mounting are several sources of capacitance and parasitic inductance, that affect the signal precision. The oscillator is supplied with a DC voltage of 5V, hence the peak value found in the figure is very close to this value.

The second part of the driver logic are the multiplexers. These components choose the signal from a set of inputs (according with the select control signal), and pass it to the output port. The objective of this implementation, is, through external control pulses generated by the microntroller, select which signal from the oscillators (2, 4 or 6KHz) should pass to each of the sectors (upper or lower), according to the symbol to be transmitted by the LED taillight.

For this purpose, two separate 4:1 multiplexers were used, one for each sector. Select ports have been connected to Arduino digital I/O ports: D3, D4 control the select ports S0 and S1 respectively from multiplexer 1, whose output controls the lower sector. Arduino's ports: D5 and D6 control the select S0 and S1 of multiplexer 2 (upper sector) respectively. Table 3.3 shows the truth table. The state *"1"* represents the high state 5V, and the *"0"* the low state 0V.

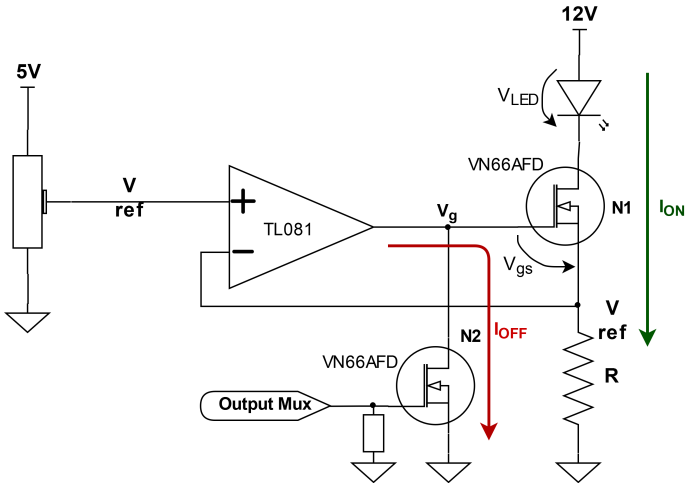
Table 3.3: Multiplexer's truth table

Select mux 1		**Selec mux 2**		**Out mux 1**	**Out mux 2**
D3 (S0)	D4 (S1)	D5 (S0)	D6 (S1)	Lower sector sig.	Upper sector sig
0	0	0	0	2KHz	2KHz
0	0	0	1	2KHz	4KHz
0	0	1	0	2KHz	6KHz
0	1	0	0	4KHz	2KHz
0	1	0	1	4KHz	4KHz
0	1	1	0	4KHz	6KHz
1	0	0	0	6KHz	2kHz
1	0	0	1	6KHz	4KHz
1	0	1	0	6KHz	6KHz
1	1	1	1	0Hz (DC)	0Hz (DC)

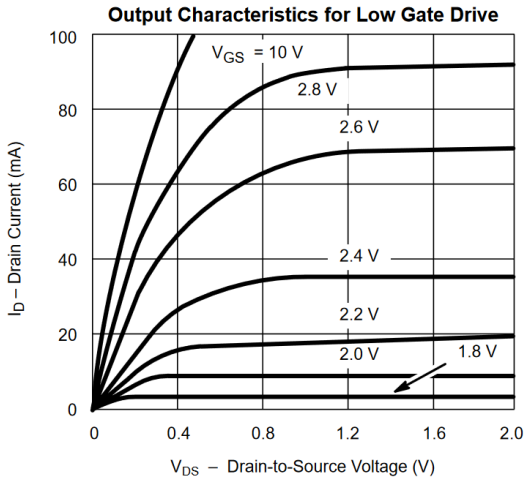
Since the frequencies imposed on each sector are related to the different symbols to be transmitted, it was added an "Idle" state, which imposes a DC level on the output. This state was created as a solution in case of need to emit a traffic signal light (stop or presence) and simultaneously no data to be transmitted in, i.e., the modulation is disabled. In this case, the lighting effect of the LED taillight will always be guaranteed.

The last logical part of the driver is the current regulating MOSFET switches. The purpose of this part of the circuit is to "flash" the LED matrix according to the control signal from the multiplexers, and at the same time ensure a constant current level that will not damage the structure of the LEDs. The circuit presented in Figure 3.9a was used.

The circuit has two possible operating states: ON or OFF. Looking at the ON state first, it is observed that the operational amplifier (TL081) operates in closed loop (*Output Mux* is in Low state, so *N2* is "cut-off" and therefore in open circuit). For this reason, it will impose on R a electrical potential difference equal to the voltage set on the Op-amp's positive differential gate. To achieve this, the Op-amp will set an output voltage level in such a way that it imposes a current (I_{on}) sufficient to guarantee a V_{ref} voltage on R. In this schematic, all the individual taillight sector (composed by an amount of LEDs) is represented by a single LED.



(a) Mosfet switch.



(b) Mosfet VN66AFD I_{DS}-V_{DS} graph from datasheet.

Figure 3.9: LEDs driver circuit analysis.

However, it is necessary to take into account several important factors:

- **Max forward current through LEDs**: The tailight luminaire is made up of **SMD 3528** LEDs with red light emission. According to the datasheet, these have a typical $V_{forward}$ between 2.2V and 2.6V, and a maximum current of 25mA. Each lighting sector consists of 21 elements (19 LEDs plus 2 zener diodes) arranged in 3-element parallels (section A.4). This means that the maximum allowable current for each sector is 0.175mA and the $V_{forward}$ of the all matrix is around 7V.
- **MOSFET operation voltages**: From the previous point, it is possible to understand the maximum value of current I_{on}. In order to ensure that the value never exceeds this limit, the following precautions have been taken:
  1. Ensure that the MOSFET current I_D does not exceed the maximum value of I_{ON} by checking the graph of the I_{DS}-V_{DS} ratio presented in the VN66AFD mosfet datasheet. It was found that for a V_{GS} less than 10V and V_{DS} less than 0.5V, the current I_{DS} will always be less than 100mA.

2. The system is supplied with a 12V DC voltage, subtracting the potential drop imposed by the LED array as 7V, thus leaving 5V for the mosfet V_{DS} plus the potential difference in resistance R. The choice of the reference voltage value V_{ref} and resistance R, must therefore take into account the previous point.
3. The TL081 operational amplifier is powered with 12V DC voltage. For this assembly, it will then have a maximum V_{out} close to 11V, due to the fact that it is not a Rail-to-Rail Op-amp. This value has impact in the characterization of the V_{GS} of mosfet N1.

Considering the points presented above, the circuit was dimensioned with the values presented in Table 3.4. Figure 3.10 shows the voltage on mosfet N1 source port (which also means the potential difference imposed on R), its possible to verify that the maximum value is as expected close to 3.3V.

Table 3.4: Mosfet switch regulator value

V_{ref}**:**	**R:**	I_{on}**:**	V_{DS} **N1:**	V_{GS} **N1 max:**
3.3V	33Ω	100mA	1.8V	7.7V

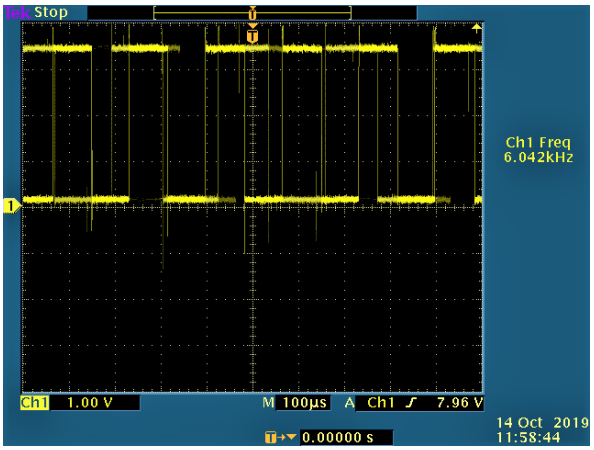


Figure 3.10: Electric potential difference of R during a transmission of symbols.

Now looking to the OFF state of the system. This is done when *Output Mux* is "1" (5V), consequently N2 is biased and enters into conduction regime. This "short circuits" the output of the operational amplifier to a value close to 0V. The **TL081CP** is equipped with *output short circuit protection* and unlimited output short circuit life, this is a necessary feature in this application, as it safes from "burning" when dealing with the OFF state of the system.

Note finally, that the circuit behaves like an inverter, because when *Output Mux* is in high state, N2 will be polarized and N1 depolarized, and consequently the LED will be off.

The general schematic of the driver can be found in section A.3.

3.2.1.4 Car LED tailight

The last part of the transmitter is the car LED taillight. This piece despite its simplicity is a fundamental element of the entire system. The choice fell on this type of car light due to

several reasons:

1. Simplicity of the radiation diagram when compared to the headlight.
2. Simple construction and mounting.
3. Portability
4. Low power consumption.
5. Available part, by the enterprise partner of this work: *Exatronic*.

This taillight, in its initial state, consisted of a set of inputs: (presence light, stop light, turn signal light and ground). The user controlled its operation by imposing a voltage (12V or 24V) on the terminal of the function that wanted to activate. In terms of electronic circuit, the taillight consisted of two circuits for regulating current through switched sources (one for the red LED sector and another one for the orange LED sector).

The first change to be made to the taillight circuit was to "pull out" the current regulating circuit. This is because, this type of circuit does not allow relatively high frequency changes to its power supply, thus making it impossible to modulate light. In this way, direct access to the array of LED's became possible. The second and last amendment was to make the two sectors, responsible for emitting red light, independents of each other. This was possible by cutting the pcb lanes that interconnected the two zones. The middle sector (responsible for emitting light for the turn signal) was left untouched throughout all this work.

The taillight setup consists of 4 different parts, as shown in Figure 3.11.

Figure 3.11: Taillight components (from left to the right): PCB with recessed LEDs, plastic mirror, plastic color lens and plastic encapsulation.

Attached to the external plastic cup of the taillight, it is possible to observe the inscription on Figure 3.12.

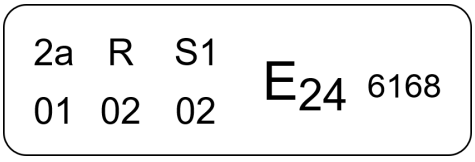


Figure 3.12: UNECE homologation inscription on taillight.

These inscriptions correspond to UNECE homologation and are interpreted as follows:

- **2a 01**: A rear direction indicator lamp (category 2a) approved in accordance with the 01 series of amendments to Regulation No.6.

- **R 02**: A red rear position lamp (R) approved in accordance with the 02 series of amendments to Regulation No.7.
- **S1 02**: A stop-lamp with one level of illumination (S1) approved in accordance with the 02 series of amendments to Regulation No.7.
- **E24**: Device approved in Ireland (E24).
- **6168**: Approval number 6168.

The UNECE regulation that rules this type of automotive component is *Regulation No.7*. One of the concerns of this work, has always been to try to ensure maximum compliance with existing regulations by the competent authorities, in order to ensure the application of the system in a real environment. One of them was the luminous intensity emitted by the taillight. For this, *Regulation No.7* has been consulted, it can be seen that for the stop light function of category S1 (the category of this lamp), it must present a luminous intensity between 60cd to 260cd. As mentioned earlier, this taillight was previously subject to approval before being applied to this work. The results shared by *Exatronic* for the Stop Light function are shown in section A.5, and a part from this report is shown in Figure 3.13, in particular the radiation diagram and measured luminous intensity values.

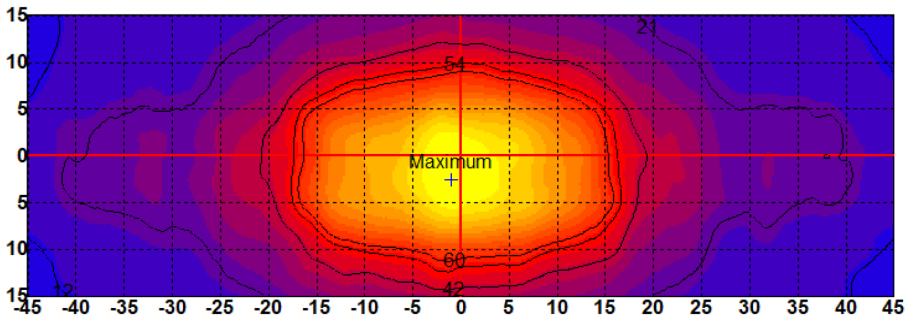


Figure 3.13: Taillight Stop light 100% LED's, homologation results (candelas).

According to the same report, the results in Figure 3.13 have the main characteristics shown in Table 3.5.

Table 3.5: Taillight Stop light report, main characteristics.

Active LEDs:	All (both sectores)
Maximum measured luminous intensity:	125.26cd
Maximum position:	H=-1.00º, V=-2.50º
Current:	0.097A

In order to verify the compliance of this regulation, with the circuit diagram of the PCB changed as presented above and making use of the driver developed for this work, the luminous intensity emitted by the light in a dark camera was measured. For that, it was used the lux meter model HD450 from EXTECH. It is also important to note that the taillight was active in symbol emission mode.

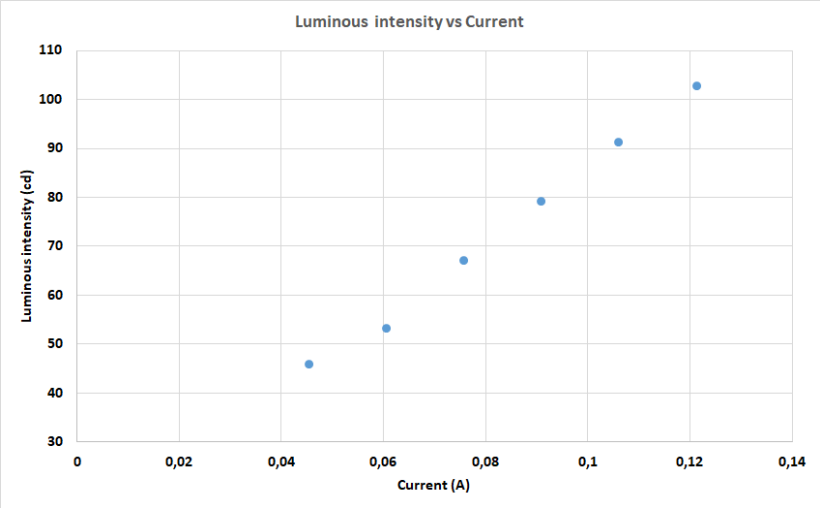


Figure 3.14: Transmitter luminous intensity measure.

The graph in Figure 3.14 shows the results obtained. In order to better understand the variation of luminous intensity as a function of the current flowing through the LEDs, the current has been changed by varying the *Reference Voltage* of the driver circuit (Figure 3.9a), keeping the same value of the resistor **R** (33Ω). The luminous intensity shown in the graph corresponds to the total light emitted simultaneously by both sectors (upper and lower), however the current presented in the same graph corresponds to only one of the sectors, the total current consumed is therefore twice the value.

As noted in subsubsection 3.2.1.3, the driver has been designed to impose a current limit of $100mA$ in each of the sectors. Figure 3.14 shows that for this current value, the light has a luminous intensity close to **90 candelas**, and so, it is in compliance with UNECE requirements. It should also be noted that comparing the total current consumed for this configuration ($\approx 200mA$) with the current consumed by the taillight without being subjected to any modulation of its light ($97mA$), it is found that for similar values of luminous intensity twice the maximum current is required. This is due to the type of modulation used, as explained above, the light emitted by the taillight is modulated in current through square waves with duty-cycles close to 50%. This requires that for a similar luminous intensity level it has to be twice the maximum current intensity, as the LED will only be active half the time period when compared to a DC emission (and assuming that the LED has a linear light emission in function of the current that runs through it).

3.2.2 Reception based on Image Sensor

The receiving system can be reduced to the camera and the frame processing block (in this case the computer). The camera chosen for the purpose was the Raspberry Pi Camera Module v2, an official product from the Raspberry Pi Foundation. This version of the camera has the ability to capture up to 8 megapixels and with sensitivity to visible and infrared light. This device is only connectable to small single-board computers made by Raspberry Pi. The main reason for choosing this camera was the freedom of possible configurations offered to the user,

this property is fundamental to the requirements of this work. In addition, all camera module configuration is based on C or Python programming language, enabling integration with other programs or codes. Among many features presented by the manufacturer, Table 3.6 presents the most relevant.

After some tests, it was verified that at a distances larger than 2m, the symbol captured became too small. The decoding algorithm makes use of a neural network, which needs an input object with a minimum of "features quality" for properly classification, in this case: width, size and location of the stripes. Considering this, capture frames with ROI too small would become a decoding problem. To avoid this constrain, it was attached to the front of the camera a zooming lens with a focal length of 8mm.

Table 3.6: Raspberry Pi Camera module v2 characteristics

Still resolution:	8 Megapixeis	**Camera focal length:**	3.04 mm
Sensor:	Sony IMX219	**Horizontal field of view:**	62.2 degrees
Sensor resolution:	3280x2464 pixels	**Vertical field of view:**	48.8 degrees
Sensor area:	3.68x2.76mm	**Global or rolling shutter:**	Rolling shutter
Pixel size:	1.12umx1.12um	**Fixed focus:**	1 m to infinity
Optical size:	1/4"	**Sensitivity:**	680 mV/lux-sec
Lens focal length:	8mm	**LED flash:**	no

The only possible way to interact with the camera module is through Raspberry Pi, however, despite the numerous capabilities of these small computers, the fact that they are equipped with processing units (GPU and CPU) with low computing power, does not allow the frame reception and decoding algorithm to run on them. The only possible solution is to use a machine with greater processing power for this phase of the process. As such, it was necessary to design a system that interacted with the camera, acquired the frames and then sent them to the computer, where the heavier processing algorithms would be running.

The solution found makes use of a server-client interaction. The server, on the most powerful machine, i.e. the computer, which listens for a connection from the Raspberry Pi, and a client which runs on the Raspberry Pi and sends a continuous stream of images to the server. The protocol used is quite simple, first the length of the image will be sent, then this will be followed by the bytes of image data. If the length is 0, this indicates that the connection should be closed as no more images will be transmitted. Figure 3.15 shows the logic diagram of the receiving system, as well as the pseudo-code of the client and server scripts.

The server script creates a client (Raspberry Pi) interaction socket. Upon receipt the client feedback, opens a connection and makes a file-like object that serves as frame reception buffer. The connection once created is kept in an infinite loop as long as incoming frame packets have length larger than 0. When the last frame arrives, the server will also receive information that the next data packet has size 0, meaning that there is no next frame and that may close the connection.

The client script after creating the socket and the server connection channel, initializes

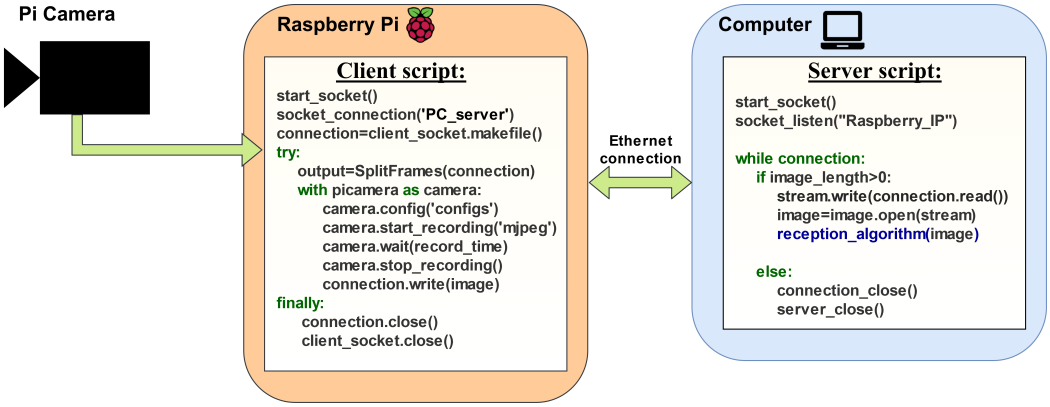


Figure 3.15: Receiver server-client protocol.

the camera according to the settings passed by the user (frame-rate, shutter-speed, brightness, among others). From here comes the capture method, there are several methods, some more focused on the transmission speed, others more focused on the transmitted image quality. In reality and as expected, there is a compromise between transmission speed and image quality. The camera is capable of capturing a sequence of images extremely fast by utilizing its video capture capabilities with a JPEG encoder (via the *use_video_port* parameter). However, there are a few things to be careful about when using this technique:

- When using the video port, only the video recording area is captured, depending on the camera's sensor mode, this area may be smaller than normal image capture area.
- Captures typically appear "grainier" with this technique. Captures from the still port use a slower and more intensive denoising algorithm.

It is in the interest of this work to perform processing on the frames after capture, so it is appropriate to capture the video and decode the frames after, rather than dealing with individual JPEG captures. By creating a custom output (custom output is an object that admits a set of inputs and parameters and allows the application of methods), it separates the frames out of an MJPEG video recording and writes them individually in the connection buffer. This custom output was named as *SplitFrames*. Along with the frame, the custom output is also composed of a method that indicates the size of the frame to send, this value is passed to the server. After everything is set up, the client script will drop frames during the time period indicated by the user.

In section A.6 and section A.8, are presented the configuration codes for the client (Raspberry Pi) and server (computer) respectively. For the interaction with the camera it was used the *Picamera* library. This is a package that provides a pure Python interface to the Raspberry Pi camera module for Python 2 or Python 3. This library allows the configuration of all camera parameters, from the capture process to the method and type of data packet transmission to the Raspberry GPU. From many possible, here are some configuration parameters of interest for this work [47]:

- **.iso**: Retrieves or sets the apparent ISO setting of the camera. The ISO value affects the sensitivity of the image sensor, more sensitive values will make the photo more bright,

but also increases the grain of the photo. This is a principle that it is also applied in film photography.

- **.awb_mode**: Retrieves or sets the auto-white-balance mode of the camera. Since the color temperature varies depending of the characteristics of the light source, adjusting a photo's white balance helps remove the color casts that different color temperatures produce. The possible modes are: 'off', 'auto','sunlight','cloudy', among others.
- **.brightness**: Brightness refers to the amount of light received through the lens of the camera.
- **.contrast** : Contrast is the difference in brightness between light and dark areas in a scene.
- **.shutter_speed**: Shutter speed is the length of time camera shutter is open, exposing light onto the camera sensor. In practice, the Raspberry Camera does not have a mechanical shutter, which opens and closes and therefore exposes the sensor to the light, in reality, the camera electronic driver controls which line is exposed to light and for how long.
- **.zoom**: Retrieves or sets the zoom applied to the camera's input.
- **.resolution**: Retrieves or sets the resolution at which image captures, video recordings, and previews will be captured. It depends of the *sensor_mode* in use.
- **.framerate**: Retrieves or sets the frame rate at which video-port based image captures, video recordings, and previews will run. Frame rate is the amount of individual video frames that camera captures, per second.

Table 3.7: Camera configuration parameters.

Shutter speed	**ISO**	**AWB mode**	**Brightness**	**Contrast**	**Resolution**	**Frame rate**
$9\mu s$	0	auto	50	0	640x480pixels	15fps

All of these parameters have a huge impact on the final image delivered by the camera. Table 3.7 presents the camera parameters configured.

From all, there are two parameters with huge relevance to the communication system. They are Frame rate and Shutter speed, and are both related. The camera's frame rate, as shown above, defines the number of frames captured and processed by the GPU in a unit of time, in this case 1 second. From a system perspective, it is critical that the camera frame rate is synchronized with the symbol rate transmitted by the taillight. Since there is no connection between the receiver and the transmitter to send reception feedback, the only way to ensure maximum synchronization is to rely on the requested frame rate from the camera. Thus, the section that will therefore impose the frame rate of the system is the receiving unit, specifically the processing block that will decode the frames, in this case, the computer. For this purpose some tests were performed, where the time needed to run an interaction was measured (in other words, process and decode a frame and find the corresponding symbol sent). The graph Figure 3.16 shows the results. This test was performed using the Python function *time.clock()* for time measurement. It should be noted, however, that the processing algorithm

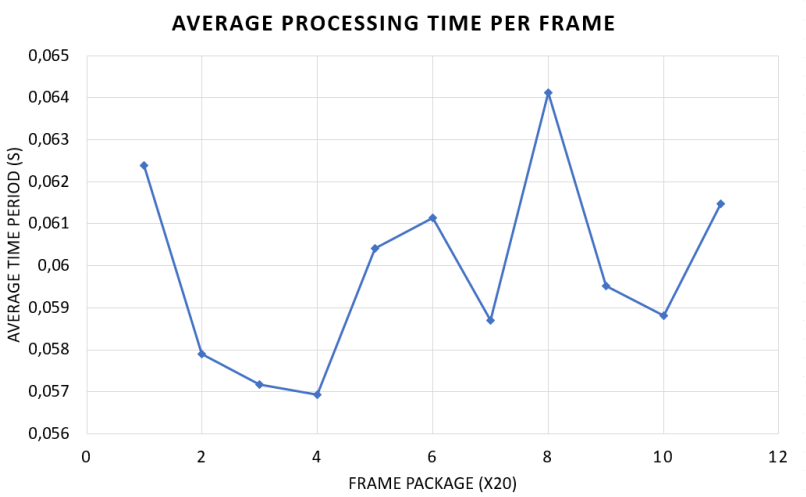


Figure 3.16: Average processing time per frame.

runs on a Non-Real-Time Operating System (RTOS), which means that the measured time is not accurate due to the fact that the CPU is not only dedicated to processing the program in question, but to answer other instructions at the same time. However, for the test in question, all programs that were not necessary for the receiving system were closed, making the measurement of the elapsed time as close to the real as possible. As expected, the average processing time for each frame of a packet of 20, does not have a constant value. The results obtained have an average value close to $0.0599s$, with a maximum of $0.06412s$, which represents an average frame rate of $16.7fps$ and a minimum of $15.6s$. In order to maintain a safety margin, the value of $15fps$ was chosen as the system frame rate.

The second important parameter is shutter speed. This parameter, also known as exposure time, represents the time period at which the sensor inside the camera is exposed to light. Recalling the effect of the capture method based on the Rolling Shutter phenomenon (subsection 2.4.2), the value of this parameter will have a huge impact on how the transmitted symbols will be captured by the camera. Knowing that the technique used to create the different symbols to be sent is based on modulating the current that flows through the different sectors of the taillight with a square wave and a duty-cycle close to 50%, thus making the LEDs flash at a determined frequency. However, for the camera and according to the Rolling Shutter phenomenon, the Shutter Speed parameter will affect the shape of the stripes in the captured image. Figure 3.17 shows the differences of the captured image to 4 different values of the shutter speed. From Equation 2.31 and Equation 2.33, it was expected that by increasing the exposure time, the stripes would become thinner and the total number increase. However, the practical results show a different behavior, increasing the exposure time doesn't make the stripes thinner, in the opposite, makes them "look thicker". This is because, in fact, increasing the exposure time, means that each row will be more time exposure to the light and consequently capture more photons. The result is brighter stripes that look thicker. For shutter speeds larger than $200\mu s$, the stripes become undetectable.

For the case with a shutter speed of $9\mu s$, the Equation 2.33 was verified. Considering the

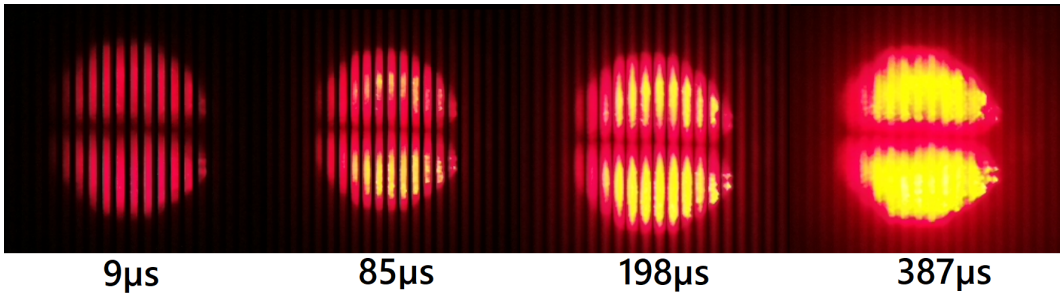


Figure 3.17: Impact of different shutter speeds on captured frames.

camera parameters of Table 3.6, the following variables were obtained: It should be noted

Table 3.8: Variables calculated for Equation 2.33 verification.

Ai:	**f0:**	**fon=foff:**	**tr:**	**rho :**	**d:**
1.13cm2	11.3mm	2KHz	18.8ns	1.12um	3m

that the taillight has a radius of $6cm$, focal length is equal to the sum of camera focal length plus lens, the stripes frequency is $2KHz$ and unique, and the exposure time per row is equal to the total exposure time (shutter speed) divided by the number of rows (480, because resolution used is 640x480). With this, the result obtained was 9.18 stripes, which is close the number observed in Figure 3.17 that is 9 (considering just the "clear" stripes). This shows the accordance of the practical result with the theoretical equation.

The shutter speed value is limited by a maximum and a minimum, the maximum is determined by the minimum exposure time required by the camera, and depends on several factors (sensor_mode, sensor gain, among others). It is therefore not possible to determine an exact value. By setting the value of "*shutter_speed*" to 1, the camera will apply the smallest possible value according to the other settings. After some tests, the minimum value never was smaller than $9\mu sec$. The maximum value is determined by the camera's minimum frame rate, high frame rates means that more frames are needed for one period of time, meaning that the exposure time per frame will consequently decrease. For a 15fps, the maximum exposure time is $66.7ms$. In order to obtain the maximum impact of the Rolling Shutter phenomenon, the shutter speed has been set to the minimum possible.

The last parameter to take into account when setting up the camera was the resolution. As mentioned before, the communication between the camera and the computer was based on a client-server system, where the transmission frame rate obeys a trade deal between frame quality and number of frames delivered to the server. In order to maximize the number of frames transmitted, as well as to make the decoding of frames as light as possible for the reception algorithm, the chosen resolution was the minimum possible, with a value of $640 \times 480 pixels$.

3.3 Reception based on Convolutional Neural Network

The approach chosen for processing the received frames containing the inherent transmitted symbols was based on the application of a Convolutional Neural Network. This choice took into account several factors, which can be summed up as:

- It is a reception process that is not very applied on communication via optical cameras.
- Since the application environment of this system (outdoor) has conditions (luminosity, reflections, other sources of interference) quite irregular, this justifies the application of a more robust reception system to various ambient situations compared to a purely deterministic system such as image processing.

Of course, this type of solution also has several drawbacks. Given its complexity and the inherent computational effort needed, its implementation is not easy and requires attention to several details.

Firstly it will be explained the logical concept behind Convolutional Neural Network (CNN) and then the architecture of the applied model.

3.3.1 Convolutional Neural Network for image classification

Convolutional Neural Networks[48], is a specific type of neural networks used for grid-like data processing. Its application ranges from sound processing (1D grid) to image processing (2D). In either case, order and spatial location are of enormous importance to their value and the CNN makes use of this characteristic to optimize its algorithms. The "convolutional network" denomination comes from the fact that it makes use of the convolution mathematical operation. This is one of the main differences from other neural networks that make use of matrix calculus to propagate the signals along the neural network. Convolution is an operation of two independent functions among an axis. It is defined as the integral of the product of one function times the reversed and shifted version of the other. Equation 3.3 shows the base formula:

$$(f * g)(t) = \int_{-\infty}^{+\infty} f(\tau)g(t-\tau)d\tau \tag{3.3}$$

One particular property is the commutativity of this operation:

$$(f * g)(t) = (g * f)(t) = \int_{-\infty}^{+\infty} f(t-\tau)g(\tau)d\tau \tag{3.4}$$

One possible application of this type of operation is the weighted averages calculation. In CNN logic, the Equation 3.3 takes the principal role of all process. This equation is the connection link between consecutive layers, and can be described as:

- The function $f(\tau)$ is considered as input argument, in other words, is the "output image" that comes from the previous layer.
- The second function $g(\tau)$ is named as *kernel* or *filter*, and contains the "neural values" of the present layer that will transform the input argument.
- The result of this operation (the output of the layer) is named as *features* or *features maps*, and evidences the feature or features that the present layer wants to highlight (could be a shape, color, pattern, etc).

Of the various applications with neural networks, usually the input is a multi-dimensional array, in the specific case of images it is a 3D array (2 space dimensions plus 1 color dimension). The kernel, made up of parameters that are adapted by learning algorithms, can also have multiple dimensions. This means that the integration domain is not infinite, but finite according to the number of array elements. Looking at an application whose input is images (I), they have a 2D dimension (considering only one color component), meaning that it will be of interest to use a two-dimensional kernel (K). Thus the convolution operation will necessarily deal with two domains:

$$S(i,j) = (I * K)(i,j) = \sum_{w} \sum_{h} I(w,h)K(i-w, j-h) \tag{3.5}$$

When applied to two dimensional arrays, the convolution operation can be viewed as a doubly block circulant matrix, where the kernel matrix circulates along the matrix I and for each position it is calculated the dot product (Figure 3.18).

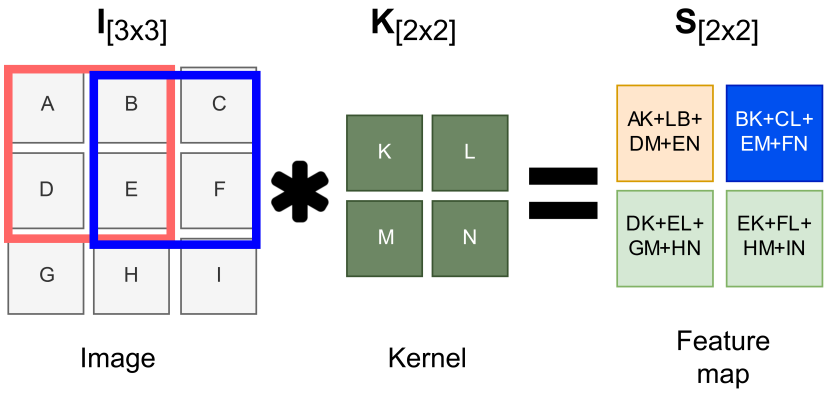


Figure 3.18: 2D convolution process.

In a practical way, the mathematical process explicit on Figure 3.18, can be verified in the example of Figure 3.19. Here, an input image with 10x10 pixels is delivered as input to the CNN layer. Its applied three possible kernel layers, each one highlights a different feature (horizontal or vertical edges). The output of this layer (an 8x8 image) evidences the presence of this features in the input image.

The Convolutional Neural Network is a complex connection of individual layers (like the one presented above), where the output of one individual layer will be the input of the next one. This means, that the features extracted from the last layers are very complex, at least for the human perception.

From this convolutional calculation comes up some questionable details. Looking for the example of Figure 3.19, the output image presents a size smaller then the input one. This is due to the fact that is impossible to perform the convolution operation around the border of the input image. To overcome this, exits some techniques, named as *Padding*, where is attached pixels (zeros usually) all around the input image border. Otherwise, the image result of the convolution operation is always smaller than the input [49]. Other important detail is the step size of the convolution operator, in Figure 3.18 this value was 1, which means

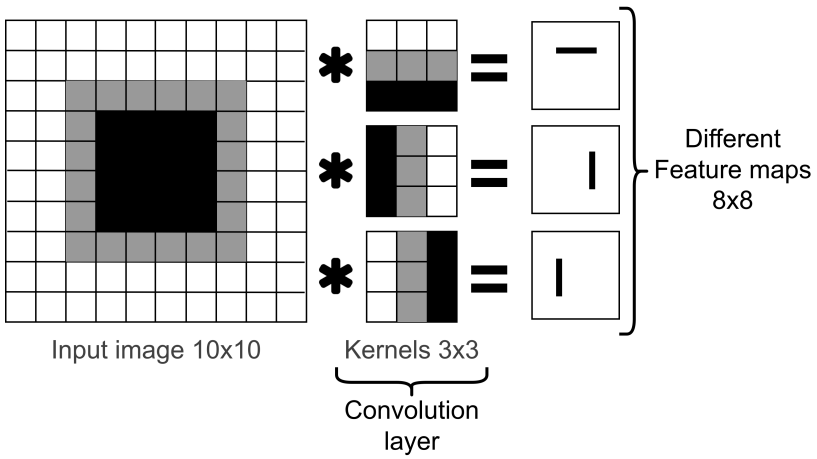


Figure 3.19: Practical example of the convolution process of an input image, and features highlighted.

that the *Kernel* operator is sliding over all the possible matrix positions of the input image. However this value could be larger than 1, which means that the *Kernel* would "jump" input image pixels, and consequently the output result would present a smaller size. This property is used in the *"pooling_layers"* (another type of layers that make part of the CNN), with the purpose of decrease the amount of data that flows through the CNN layers [49].

As shown in Figure 3.18, the feature map is obtained by the convolution operation between the input image with a *kernel*. At this point should be evident that the *kernel* matrix holds the adjustable parameters that characterize the network perception. This parameters are usually adjustable and optimized by training processes. From this property comes two important notes:

- The convolution operation performed inside each CNN layer makes use of the same *kernel*, this means that in total, the network stores fewer parameters, which both reduces the memory requirements and improves its statistical efficiency.
- Computing the output requires fewer operations, which benefits the processing elapsed time.

The main purpose of the Convolutional Neural Network for image processing is the classification of objects, in other words, look to an object and find its class (a car, a bicycle, a plain, etc). For that, makes use of convolutional operations with different *kernels*, with the objective of evidence different features that compose this object. After that, it reunites all this different features and makes a decision of which class belongs the object.

The Figure 3.20 presents the full architecture of a CNN. From the input image to the output classification. Until here, it was used the "layer" nomenclature to define the convolutional operation of an input image with a *kernel*, however, a real CNN "layer" is more complex than that.

A typical layer consists of three stages (Figure 3.20). In the first stage, the layer performs a convolution with a *kernel* to produce a set of linear activations (basically features values). In the second stage, each linear activation is run through a nonlinear activation function (it

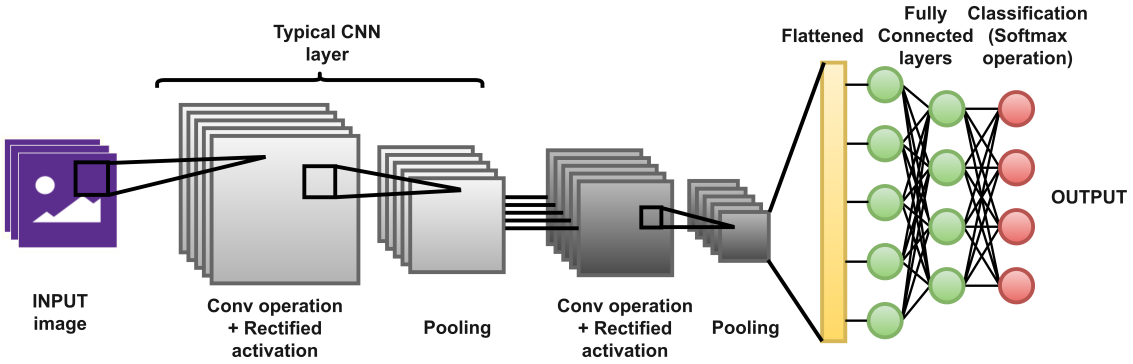


Figure 3.20: Typical CNN architecture for image processing.

applies a threshold to the feature values to annul the negative ones). In the last phase is used a pooling function to adjust the output of the layer further. A pooling function has the purpose of decrease the output feature map size, by applying different techniques (changing a set os values by its average, for example).

Figure 3.20 shows a simple example with just two layers, however, in reality a CNN makes use of a larger number. Each layer aims to highlight certain features that become increasingly complex as the layers deep in. In the last layer a flattening transformation is applied to transform the values into a 1 dimension array. The values of this array are then delivered to a fully connected network, where the input activation values are classified. Finally the fully connected network outputs, pass to a softmax function, which normalizes those values into a probability distribution and outputs the image class and a corresponding probability value.

3.3.2 Single Shot multibox Detector model for image detection

A classifier network takes an image as input and produces a single output: the probability distribution over the classes. But this only gives a summary of what is in the image as a whole, it does not work so well when the image has multiple objects of interest. Thus a problem arises: in addition to classifying an object it is also necessary to obtain its location in the space limited by the 2D image. If there are other objects present in the same image, it is also necessary to obtain its independent classification and location.

The problem with a model that analyzes an image for objects is that even if the network knows that there are more than 1 object present in the image, it has no way of distinguishing them and thus will try to identify them simultaneously, resulting in wrong classifications.

The solution found was to divide the image according to grids (Figure 3.21). Those grids (also called as *"box predictors"*) have different sizes in order to be able to draw bounding boxes with different sizes. In this way it is possible to apply the classification model to each of these independent cells without interference from objects present in other positions of these grids [50]. In total, the number of predictions will be equal to the number of cells per feature map, times the number of different feature maps times the number of detectors. This value is always the same for any image, so it is necessary to apply a *non-maximum suppression algorithm*. This algorithm is similar to the *softmax* function applied to CNN classification networks, so that, according to the confidence value, the majority number of predictions can

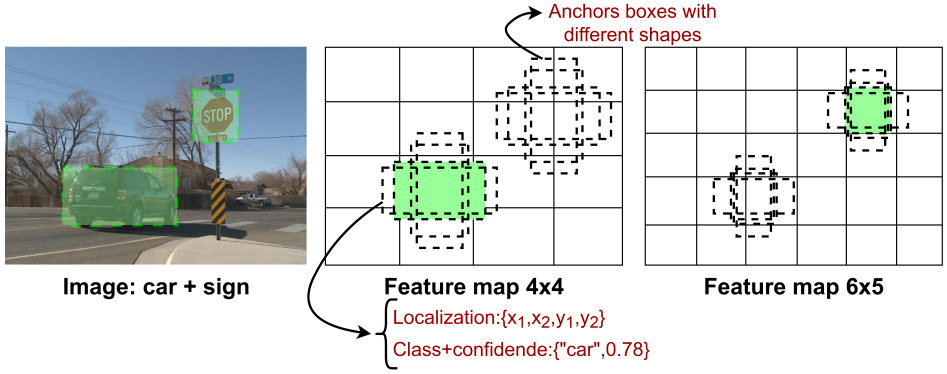


Figure 3.21: Box predictors for independent object detection.

be considered as false and ignored. There are several models for object detection based on this logic. High-end models like Faster R-CNN[51] first generate so-called region proposals (areas of the image that potentially contain an object) and then make a separate prediction for each of these regions. It brings great results, but is a slow solution since it requires running the detection and classification portion multiple times. That is why one-stage detectors (like Single Shot multibox Detector (SSD) or You Only Look Once (YOLO)) just require a single pass through the network and predicted all the bounding boxes and respective classification. That is the best choice for real-time solutions [52][53].

Applying a grid to the image makes possible to locate objects in different positions. However, another constrain arises: more than one object may appear per cell. For this, is used in each cell, detectors with different shapes, called "*anchors*", each of these bonding boxes specializes in a certain object shape (Figure 3.21). It is important to understand that these anchors are chosen beforehand. They are constants and as they will not change during training.

3.3.2.1 The model used: ssd_mobilenet_v2_coco

The model used is based on a SSD structure[50] and makes use of a base network (responsible for classifying objects) *MobileNet_v2*. The dataset used to train the weights of the base network is Common Objects in Context (COCO). The implementation of the model is based on the Tensorflow API that is an end-to-end open source platform for machine learning development. It runs over Python programming language.

Next, the model and its properties will be presented:

- **Single Shot Detector**: This type of method for detecting objects in an image has already been explained above. Being an architecture that prioritizes processing speed, it is a good choice given the need to apply the system in a real time solution. It makes use of 6 different grid shapes to detect objects of different sizes (Figure 3.22). This SSD version has 6 grids with sizes 19x19, 10x10, 5x5, 3x3, 2x2, and 1x1. The SSD grids range from very fine to very coarse [52], in order to get more accurate predictions over a wider range of object scales. The fine 19×19 grid, is responsible for the smallest objects, the exact opposite from the 1x1 grid. An interesting feature of this architecture

is that not only does it take the output of the last base network layer but also the outputs of several previous layers (Figure 3.22). Since the objective of the base network (*MobileNet_v2*) is to convert pixels from the input image into features that describe the contents of the image, different layers will evidence accordingly different output features. The convolution layer inside each *box_preditor*, depending on the shape that looks for, will need different sets of features, that's why, each *box_preditor* is connected to different depths of base network layers [53].

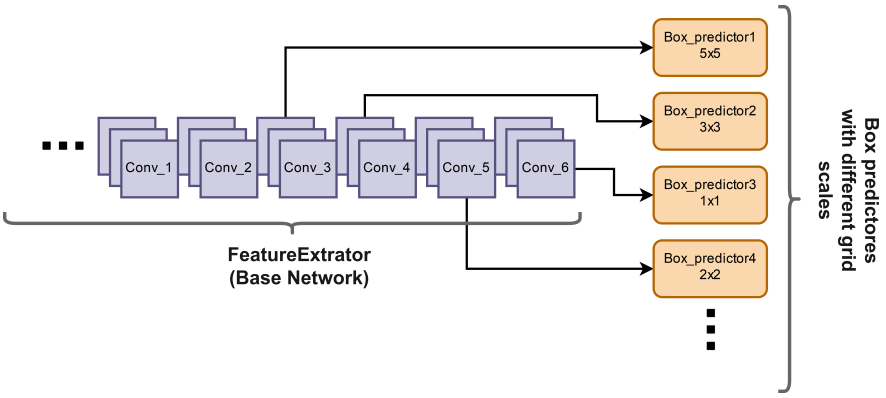


Figure 3.22: SSD architecture.

- **MobileNet_v2**: This neural network used as a feature extractor (or base network) was designed by Google and is intended to run on mobile devices[54]. It therefore has two key characteristics: lower computation needs, and processing speed. The big idea behind MobileNet_v2 is that convolutional layers, which are essential to computer vision tasks but are quite expensive to compute, can be replaced by so-called depthwise separable convolutions. Depthwise convolutions are different from common convolutions in the fact that it divides the process into two phases [55][56]. Unlike a regular convolution it does not combine the input channels but it performs convolution on each channel separately. For an image with 3 channels, the depthwise convolution creates an output image that also has 3 channels. The depthwise convolution is followed by a pointwise convolution. The purpose of this pointwise convolution is to combine the output channels of the depthwise convolution to create new features. When compared, regular convolutions and depthwise convolutions have the same inputs and outputs, however, due to this separation process, depthwise is much faster. MobileNet_v2 also makes use of depthwise separable convolutions, but instead, it has the architecture presented on Figure 3.23. The first layer, "*expansion_convolution*", a 1x1 convolution has the purpose of expanding the number of channels in the data before it goes into the depthwise convolution. The second layer, already presented, is responsible for performing all the feature extractions expected for that *conv_layer*. The last layer, "*projection_convolution*", makes the opposite of the *expansion_layer*, it projects data with a high number of dimensions (channels) into a tensor with a much lower number of dimensions. This kind of *conv_layer* is also called as bottleneck layer because it reduces the amount of data that flows through the network. So the input and the output of the block are low-dimensional tensors, while

the filtering step that happens inside block is done on a high-dimensional tensor. The second new property from MobileNet_v2 is the residual connection where the input tensor is added to the output of the *conv_layer*, this exists to help with the flow of gradients through the network.

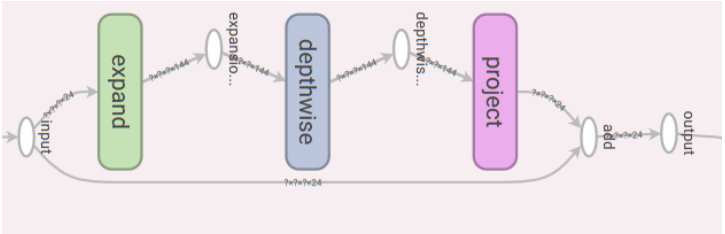


Figure 3.23: Conv_layer extracted from TensorBoard graph visualizer.

- **COCO dataset**: COCO is a large-scale object detection, segmentation, and captioning dataset. In this case it was used to train the weights of the MobileNet_v2 network, capable of classifying up to 80 different classes of objects (people, cars, bicycles, etc). This way, the base network was previously trained to collect as many features as needed to correctly classify various types of objects.

All this tools together construct the structure used to decode each of the captured frames and obtain two important informations: which symbol is sent and its spatial location (very important in a MIMO implementation).

Resuming, the capture frame is delivered as input to this structure. The SSD architecture slices the input image by using different *box_predictors* with different sizes. In each of this small boxes is applied the classification base network (*MobileNet_v2*), that will try to classify the symbol there present. In the end, is expected to obtain the classification of the symbol present in the capture frame, as well as its position inside the 2D frame.

With this, is expected to reinforce the system with sufficient robustness to cope with phenomena that degrade the quality of the received signal (sunlight with different angles of incidence, reflections, interference from other light sources, among others).

In section A.7 is presented one of the various code files for CNN configuration.

3.4 Model training for symbol classification and detection

Having already explained the CNN model used, it is necessary to train it, so that it can recognize the different symbols correctly. This phase of the process is of great importance, the proper choice of the dataset has huge repercussions on the results obtained. The model used, as mentioned in the previous section, comes previously trained with the COCO dataset. In other words, the base network also referred to as feature extractor, has the parameters of the kernels belonging to the different layers already properly trained to collect as much as possible features from the input image. For that, Tensorflow was used to apply a transfer learning training methodology. With transfer learning, instead of starting the learning process from scratch, it starts from patterns that have been learned when solving a different problem.

This way it leverage previous learnings and avoid starting from scratch (which would take a large amount of time):

- Use pre-trained weights, remove last layers to compute representations of images.
- Train a classification model from these features on a new classification task.
- The network is used as a generic feature extractor.
- Better than handcrafted feature extraction on natural images.

With this, the core of the network (MobileNet_v2) keeps its parameters intact, adjusting only the parameters of the sector responsible for the classification: the last layers on the network and the convolutions within the *box_predictors*.

Over the course of this work, more than one dataset was created, which was used to train the network and subsequently test it. One of the problems immediately evident is its size. According to the literature, a minimally adequate dataset should have over 10000 images, properly divided between the training branch and test. Something impossible to accomplish, considering the deadlines and objectives to be met. As such, a decision was made to take 200 photographs for each of the symbols, out of these 200 images, 150 would be devoted to testing and 50 to training. Thus totaling 1800 photographs taken for each dataset.

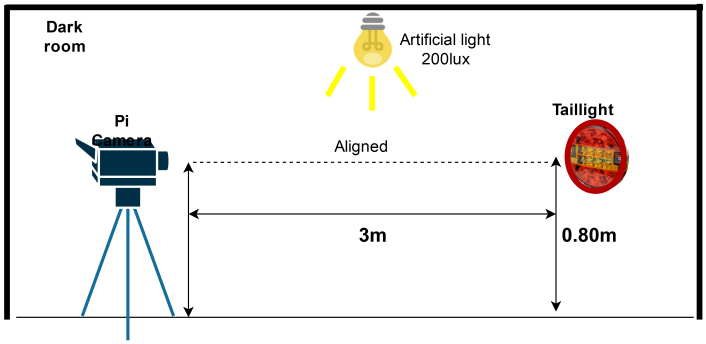


Figure 3.24: Conditions for obtaining the dataset.

The conditions for capturing the images that make part of the dataset are shown in Figure 3.24. The distance of $3m$ between the camera and the taillight is fixed, the height of $0.80m$ from the ground is the same for both devices. The alignment between them was always guaranteed at an early stage. The fact that the setup was mounted inside a dark room prevents any outside light source from interfering. However, light inside the room was kept ON, with a brightness close to $200lux$, for two reasons:

- Minimum brightness required for proper setup handling.
- Make the system resilient to interference from other light sources. The taillight is equipped with interior mirrors, which in the presence of other light sources would reflect and change the lighting pattern captured by the camera. However, for a true robustness, the dataset should embrace different luminous levels, since in the outside world, the luminous intensity varies along the day.

The camera has been configured according to Table 3.7, the same settings as it was during reception of signals. This is because, it is a good practice to train the network with images similar (geometrically and aesthetically) to the ones that it will have to decode during a

transmission. After capturing the photographs for each symbol, it is necessary to label them, this process consists of drawing a bounding box around the object to be detected as well as attaching a label to it with the corresponding class of the symbol. This bounding box with the label of the corresponding class is named as *ground_truth* in AI terminology (Figure 3.25).

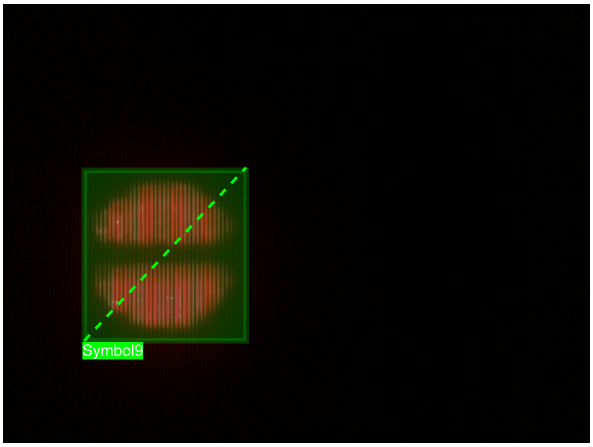


Figure 3.25: Training image after label process.

This process has to be done by hand and requires caution, especially when drawing the bounding boxes, since during the training the network will consider this inputs to be 100% correct when training *box_predictors*.

During training, the loss function and back propagation are applied end-to-end, as well as the hard negative mining and data augmentation strategies. Hard negative consists in during the matching process, choose the negative examples with highest confidence loss, ignoring the other ones, this leads to faster optimization. Data augmentation is used to make a more robust model, it changes the aspect of the input images, and with that, the dataset is expanded. The model makes predictions of different detections and classifications, the dataset contains the ground-truth boxes that say which objects are actually present in the training image, and so to train this kind of model it is required a loss function that compares the predicted boxes to the ground-truths. The images are separated by training and test sets. During training the network goes through the training images and adapts all the parameters that are necessary to classify the object. In the testing phase, the network uses the training images, and without any access to the *ground_truth* label, does its prediction. This prediction is then compared to the *ground_truth* of the image, and using a loss function, a conclusion is made about the proximity of this result to the reality. The overall objective loss function is a weighted sum of the localization loss and the confidence loss:

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{x,l,g}) \tag{3.6}$$

The x, c, l, g stands for: predicted class, true class, predicted box and ground truth box, respectively. In the equation, N represents the number of matched default boxes, L_{conf} simply compares the predicted class with the *ground_truth* class, and the $L_{x,l,g}$ compares the predicted box with the *ground_truth* using the Jaccard Index, also known as, Intersection

over Union. The variable α permits to adjust the weight of each of this equations for the total value of the loss function. This whole process of learning, testing, analyzing results and self-adjusting the weights of the various kernels is called back propagation.

This entire process is applied by the Tensorflow library, which trains the network and returns the loss function value for each training iteration. Training should be performed until a loss function value of less than 1 is presented, however, for the problem in question it was impossible to reach this value (even with interactions close to 30000), being at best close to 1.5. One possible reason for this, is the dataset size used. With a bigger and robustness dataset, the network will be able to optimize its parameters even better, and consequently perform better predictions.

3.5 System performance evaluation

After having the transmitter and receiver setup properly designed and ready, it was time to proceed to the testing phase. At this stage, the objective was firstly to verify the existence of a communication link between TX and RX. As might be expected, the system did not deliver satisfactory results at first. Due essentially to the conjuncture of the symbols used, with the characteristics of the camera. The captured ROIs were too small relative to the overall frame size. This made it impossible for the network to have enough data to perform the correct decoding. Then came the idea of adding a zoom lens (already described in subsection 3.2.2). The dataset was again captured and processed, the network trained, and is according with this configuration that all the results presented in this work are based on.

3.5.1 Confidence test at fixed distance

Considering the training scheme of Figure 3.24, the first test consisted in a continuous transmission for each symbol, the receiving camera and the taillight transmitter were placed exactly in the same position as in the training stage.

A continuous transmission was then performed for each symbol for 2 minutes, which at 15FPS frame-rate corresponds to 1800 transmitted symbols. Each symbol is received and processed in real time, and the network output corresponding to the most confident symbol is recorded in a .txt document for further analysis in MATLAB®. The neural network, for each frame, recognizes innumerable objects (in this case symbols), however most with very low values of confidence. An appropriate decision threshold must then be applied, this is also one of the objectives of this measurement.

Only the object with the highest confidence value was considered as register output because only one transmitter (taillight) was used, as the neural network organizes the output tensor in order of the confidence parameter, it makes sense to only consider the highest one.

For 8 of the 9 symbols (2-9), it can be seen from the Figure 3.26 that the average confidence value (which is given between 0 and 1) is very close to the value 1, in other words, the neural network has excellent precision on the symbol it detects. However, for symbol 1, which is composed of the upper and lower frequency of 2kHz, has a very low confidence value (around 0.05). When this "problem" was detected, the neural network was again configured and

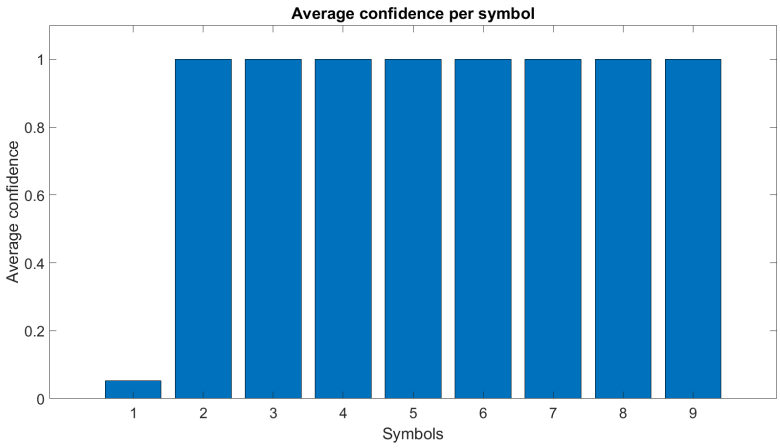


Figure 3.26: Average confidence for each transmitted symbol for a 3m straight distance.

trained, however the result was the same. A possible cause for this phenomenon could be the difficulty of the network to remove sufficient features of the stripes generated by the rolling shutter effect for the 2kHz frequency, however this possibility has been set aside, as other symbols that also make use of this frequency are detected without any difficulty. It was then assumed that the problem is possibly due to an error in the training configuration of the neural network (labeling, number of classes to consider or Idx of each class). However, given the time constraints and satisfactory results for the other 8 symbols, it was decided to proceed with the work sequence.

Another important aspect to consider is the decision threshold value. Given the results of the Figure 3.26, it was then considered a threshold of 0.5, i.e. above this value it is considered a valid detection, below it is considered a non-detection.

3.5.2 Synchronization and mixed detection phenomenon

After verifying the recognition by the receiving system of the transmitted symbols, it was time to start the Symbol Error Ratio (SER) tests according to several conditions (alignments, luminosity, obstruction of the symbols, etc.). For this, the Arduino has been programmed to generate random symbols and transmit them by the taillight, in parallel in a text file with the name "TX", each symbol sent is recorded. In the reception sub-system, the frame is received and subsequently decoded to a symbol, its value is also recorded in a text file called "RX".

The test is allowed to run for a long time (around 3 hours), which, under the usual test conditions, generates up to 162000 symbols. At the end of transmission, TX and RX files that store all sent and received symbols correspondingly, are compared against using a MATLAB®script and the SER value is obtained (Figure 3.27).

It was then verified a peculiar phenomenon. For a transmission with an exact period of time, the number of received symbols was always superior (in some units) to the number of sent symbols. This had huge repercussions in the SER measurement, since the symbols Tx-Rx were no longer aligned in time, making the correct measurement of the ratio impossible.

After some tests, it was found that in average, in every 16000 transmitted symbols was

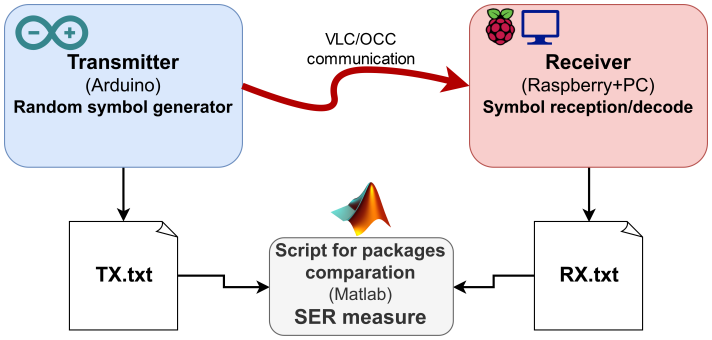


Figure 3.27: SER measuring procedure.

always received one more, i.e. 16001 symbols. However, the position of this "intruder" is not exact within the data package, so it is very difficult to detect.

For this, it was then projected a communication protocol in order to be able to identify the exact location of these incorrect bit(s) and consequently its removal. Considering then a probability of occurrence of this phenomenon of $p_e = 1/16000$, the solution found was the creation of fixed size data packages (Figure 3.28).

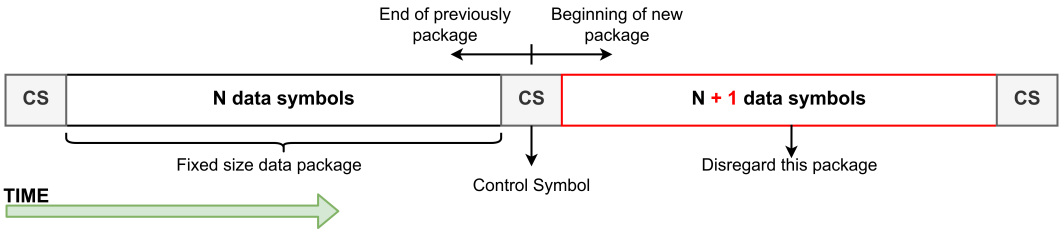


Figure 3.28: Data symbols encapsulation.

This makes it possible to identify the affected packages. If the received data packet has a size different than specified ($Nsymbols$), it is considered wrong and discarded. Recalling that this type of communication occurs only in one direction ($Tx \mapsto Rx$) it is impossible for the receiver to ask the transmitter to repeat the transmission of the damaged data packet. It is known in the telecommunications literature that there are data encodings that allow the exact identification of the wrong bit (in this case symbol). However, this type of solution was not studied or applied in this work.

There is a variable that needs to be defined, it is the data size of the frame (N). This choice was not random, and two factors were taken into account: encapsulation efficiency and data transmission efficiency. Encapsulation efficiency is defined as the ratio between the size of each packet reserved for carrying data symbols to the size reserved for control symbols. As shown in Figure 3.28, each data packet uses only one control symbol, which serves to symbolize the end of that frame and the beginning of the next one. Thus efficiency is given by:

$$\eta_D = \frac{N}{N+1} \tag{3.7}$$

Data transmission efficiency is the ratio of data lost upon detection of a corrupted packet to the total amount of data sent. In this case, the probability of occurrence of the phenomenon has impact on the relationship:

$$\eta_e = \frac{\frac{1}{p_e} - N}{N} = 1 - N \cdot p_e \tag{3.8}$$

For the variable p_e it was considered the value of the probability of occurrence of the phenomenon presented above ($p_e = 1/16000$). The value of N must correspond to the best compromise between the two relations (Figure 3.29).

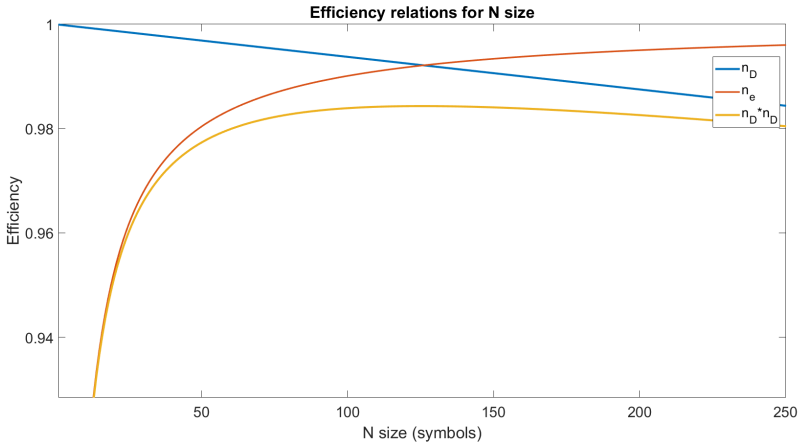


Figure 3.29: Efficiency relations for the package fixed data size.

Analyzing the graph in Figure 3.29, it can be observed a maximum efficiency for the relationship between η_D and η_e for an N of 125 symbols. This was the value then chosen as the fixed frame size. The symbol used as "CS" was ”2”, and so, this symbol is reserved for packet control and no more available as data.

By applying this type of protocol to the transmission test, and using a MATLAB®script, it is possible to process all received packets and identify those damaged by this problem (those with data length different from 125). In order to better understand this phenomenon and the apparent reason for its existence, a 3 hour continuous transmission test was performed, corresponding to approximately 162000 symbols transmitted at 15FPS. It should also be noted that an error in detecting the CS (classifying it as another symbol, or identifying any other symbol as a "2") causes at least 2 packets to be considered invalid. In order to avoid this, the test was performed under the closest possible conditions to those imposed during the capture of the training dataset (distances and brightness of the room), thus maximizing the recognition capacity of the symbols by the neural network.

The graph from Figure 3.30 is organized according to the index of the packets, in other words, the first packet corresponds to index 1, the second 2, and so on over time. The number of wrong symbols is considered as the total data packet length (125). Since it is not possible to recover data coherently from a damaged packet, so, at least one sent data packet has been lost when this phenomenon occurs. This results in the fact that the total wrong symbols received due to this problem is multiple of N.

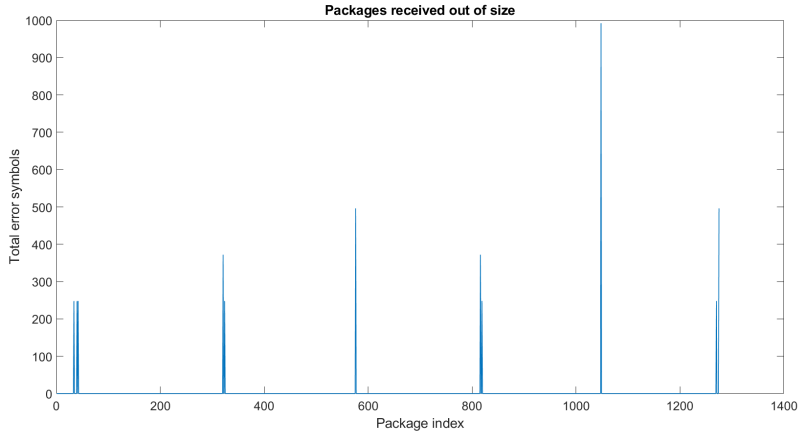


Figure 3.30: Received packets out of size during a continuous transmission.

In this test, 163501 symbols were transmitted and 163507 received, which corresponds to a difference of 6 more symbols received than sent. Analyzing the graph in Figure 3.30, there are exactly 6 peaks of errors, and these occur with relative periodicity (between 27500 and 30000 symbols). In addition to each peak, an average of 500 symbols (4 data packets) are affected. This brings two important facts:

- The phenomenon does not have the same frequency as previously scaled ($\approx 1/16000$) for the same test configurations, i.e., it can be assumed that is affected by extrinsic parameters.
- Contrary to what was initially considered, not only 1 data packet is affected, but instead 4 at least. It is concluded that the phenomenon does not occur in an exact moment, but over a period of time.

3.5.2.1 Tx-Rx asynchronous hazard

After analysis and discussion, it was concluded that the problem is due to non-synchronization between the transmitter and the receiver. Each system has its own clock: the transmitter is given by the frequency of interruption, and the receiver by the frame rate of the camera. Since there is no communication link between Rx→Tx, it is difficult to implement a two-clock reconfiguration/synchronization dynamic system. Assuming (in an ideal case) that the interruption frequency of the Arduino is exactly 15Hz and that consequently the symbol transmitted by the taillight is updated to this exact frequency, the same cannot be guaranteed on reception. In the documentation provided by the camera manufacturers, no frame-rate inaccuracy problem is identified. However, after some research on raspberry pi developers/users web forums [57][58], it was found that more users identified the same issue. The fact is that it is not possible to ask the camera to impose an exact capture frame rate. It always tends to capture at a slightly faster speed. Several theories are pointed to this characteristic, since the capture system (camera module + raspberry) consists of two distinct clocks: raspberry GPU clock and the module's internal crystal oscillator. So we have two clocks for the same system, so there is no "universal time signal". Another possible reason is due to the method used for

frame rate measurement by the capture tool. When the GPU asks for a new frame capture, it only begins to count the frame time-period from the moment the first line of pixels from the camera module arrives. From this instant until the end of the capture, the time-period will depend solely on the camera module performance. Which is known to be affected by ambient brightness, sensor mode, sensor gain, among other parameters.

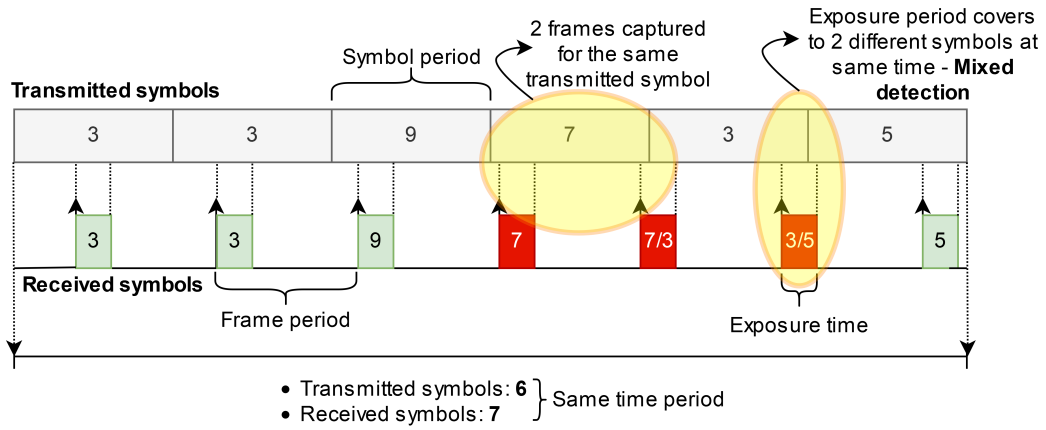


Figure 3.31: Transmitter- Receiver asynchronous hazard.

The Figure 3.31 gives a succinct illustration of the phenomenon in question. On one side there is the transmitter that emits the symbols with a certain time period. On the other, the receiver (camera), which captures the frames at a certain frame rate, however this frame rate is not exactly the same as the transmitter's "symbol rate", but rather a $FPS + \Delta$. In the case of Figure 3.30, and assuming that the transmission symbol rate is exactly 15Hz, the delta value can be calculated:

$$FPS + \Delta = \frac{163507 \cdot 15}{163501} \approx 15.00055FPS \Leftrightarrow \Delta \approx 0.0037\% \tag{3.9}$$

In this case, its a small value, but in a continuous transmission it has impact. Considering again the Figure 3.31 if the capture frame rate is slightly faster than the transmission frequency, it means that at a given instant, during the transmission of the same symbol, the camera will capture 2 frames, hence the existence of more symbols received than sent.

The second important point which justifies the fact that in Figure 3.30 each peak of affected packets is greater than 1. This phenomenon does not occur instantaneously, but over time, with a "sliding" effect. As each frame capture depends on an exposure time during which the sensor is exposed to light, if during that period the symbol transmitted by the taillight changes, therefore the captured frame will have not 1 but a mixture of 2 symbols. This can be seen in the captured frame of Figure 3.32.

When a frame like the one in Figure 3.32 is delivered as input to the neural network, it cannot decode correctly, since more than 1 symbol is present for the same ROI. And that is the reason for the existence of more than 1 packaged affected.

This type of problem is quite common in video processing, situations where capturing both a sound signal and an image signal for example, since they are two distinct systems, one can

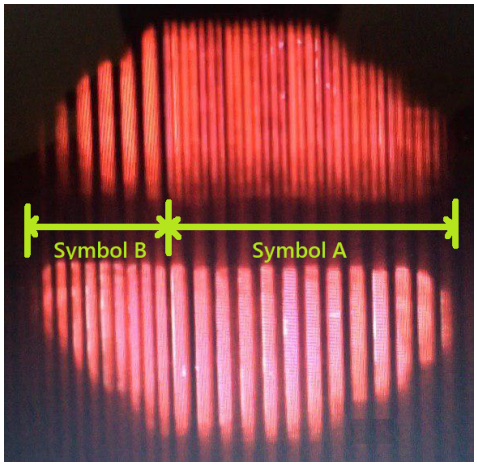


Figure 3.32: Captured frame affected by mixed detection.

be accelerated in relation to the other. To correct this, one or more video frames are usually "ignored" to ensure synchronization between the video and sound signal. In the problem here evidenced, the same solution cannot be applied, since all received symbols represent valid information. Although an attempt was made to solve the issue by imposing controlled delays among other techniques. However, no satisfactory results were obtained.

As approached above, in a continuous and long transmission, this hazard would affect the correct measurement of the SER by imposing an error offset that would not correspond to reality. To avoid this, a MATLAB®script has been developed that identifies the damaged received packets and then removes them from both the sent and received data archives, before calculating the SER. Part of this script is presented in section A.10. This strategy was applied in all the tests below, where a SER measure was needed.

3.6 Concluding remarks

This chapter described the entire system developed to support the communication between the two terminals. The entire system, consisting of hardware, firmware and software, has been designed to meet the required operational requirements (UNECE regulations, eye safety, among others). In addition, several barriers were found and properly documented. The performance results show the system's viability, thus reacting positively to the challenge imposed and anticipating a possible practical implementation.

CHAPTER 4

Practical results

After concluded all the performance tests, with the results already presented in the sections of the previous chapter, it was time to carry out several practical tests in various conditions. Rather than assessing the performance of the system according to the distance, other challenges were imposed, like alignment or symbol obstruction. Also the MIMO capability was verified and discussed. In the end, an exhaustive transmission test in the real outdoor environment was done to verify the resilience of the system to noise light sources (sun, streetlight, reflections, among others).

4.1 Misalignment Rx-Tx

In a practical and real implementation, vehicles are not always fully aligned. One of the barriers to VLC-OCC communications is the necessity for line of sight between the transmitter and receiver. However, even if the light source is in the camera's FOV, it may yet, not be able to capture the modulated information through light in the way needed for correct demodulation. So, the radiation diagram emitted by the light is of huge importance. Automotive rear and headlight are equipped with mirrors (Figure 3.11), which have the purpose of modulating the light emitted according to a radiation pattern.

The purpose of this test is to verify the misalignment limits by measuring the SER while varying the angle of alignment between the taillight and the camera.

As a training scheme (Figure 4.1), the taillight was placed in a static position while the camera, always maintaining a distance of 3m to the taillight, was rotated from 18° to 0° (in relation to the normal of the plane containing the taillight). The test was done in just one way ($18° \mapsto 0°$), and not into full angle range ($18° \mapsto -18°$), since the radiation pattern emitted by the taillight is symmetric to the normal angle. As measuring step, 3° was used, resulting in 7 distinct measuring points.

For each measured point, a 3 hour transmission was considered, resulting in approximately 160000 symbols, imposing a 10^{-5} order limit for SER. The MATLAB®scripts designed to

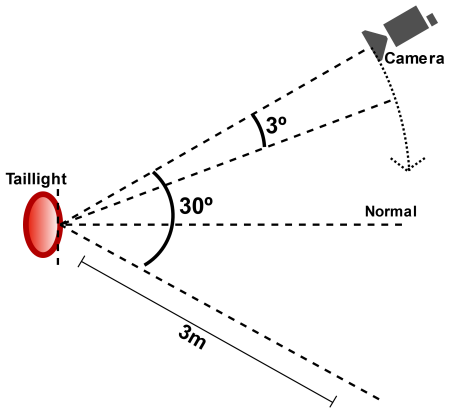


Figure 4.1: Test scheme for misalignment limits measuring.

remove the impact of asynchronous hazard were applied to the results, in order to obtain more precise SER results.

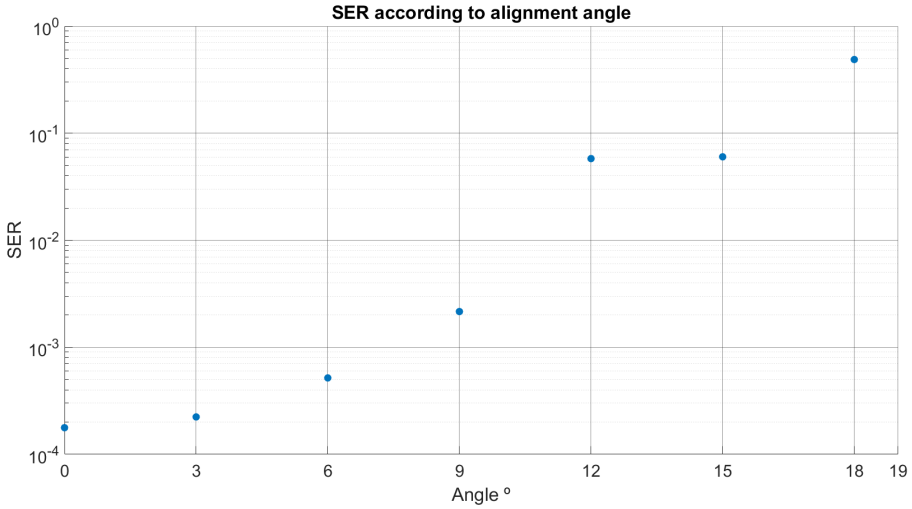


Figure 4.2: SER results for alignment test.

The results of the Figure 4.2 shows a sharp drop in SER results for angles larger than 9°. Comparing with the taillight radiation diagram, available at section A.5, it can be visualized that along the horizontal axis, exits a considerable decay of radiated power near the ±12° mark. This variation, in the order of 60/50 cd to 20/12 cd, agrees well with the results of the Figure 4.2. This shows the impact that the mirrors inside the taillight have on the delimitation of the radiation diagram emitted by it and that consequently affects the information modulated through light (VLC).

4.2 Region of Interest obstruction

When driving, its impossible to maintain an exact central position of the vehicle between the road lines delimitations. This means that the ROI (taillight), will appear into different

frame positions. But, a worst situation can happen, which is the case where the ROI is hidden behind an obstacle, and due to that, makes impossible the correct capture of the full symbol.

Obviously, this brings repercussions to the decoding process, since the CNN makes use of features that extracts from the input frame, obstruct the full symbol figure means that less features can be extracted. This results in a decrement of the CNN classification capability, and consequently in symbols misclassified. In a worst case, the reception algorithm will not detect the ROI at all.

With the purpose of chasing the geometrical condition limit that this setup is able to support and its inherent resilience capacity, a test setup was idealized. It consists in purposely obstruct the ROI with an opaque object (something similar to a journal or a magazine). Keeping the alignment between the taillight and the camera, and with a link distance of 3m, it was demanded to the taillight to transmit a continuous burst of random symbols. While this, the front of the taillight was gradually obstructed with the opaque object for each measured point (Figure 4.3).

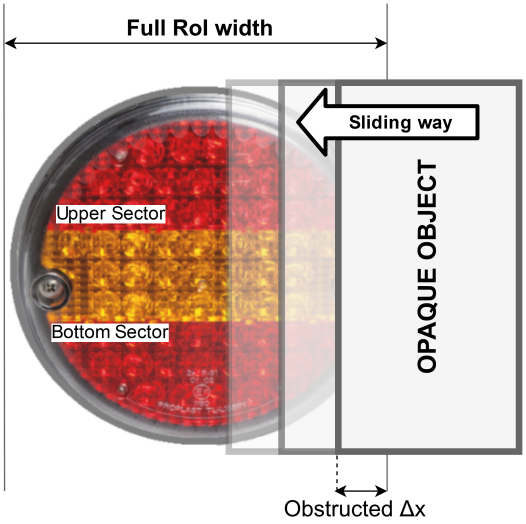


Figure 4.3: Region of Interest obstruction test scheme.

It should be noted, the way that the taillight is hidden takes into account the characteristics of the symbols (subsubsection 3.2.1.1), which is composed by two indented sectors (upper and bottom). Obstructing a full sector (by placing the opaque object in a vertical way) means that it would be lost an essential part of the symbol. To avoid this, the opaque object was sliding in a horizontal way, maintaining always a part of both sectors visible to the camera.

The taillight has an approximate width (diameter) of 12cm, considering this, it was defined an obstruction pace of 0.75cm. Considering the start position as 0cm (no obstruction), and with a continuous random symbol transmission of around 25000 symbols, a SER measure was calculated for each point. Once again, the same SER measuring technique applied in the previous tests was used.

The Figure 4.4, shows the graphical results. From this SER values it can be concluded, that the system even for an obstruction close to half symbol (4.5-5.25cm), presents a satisfactory

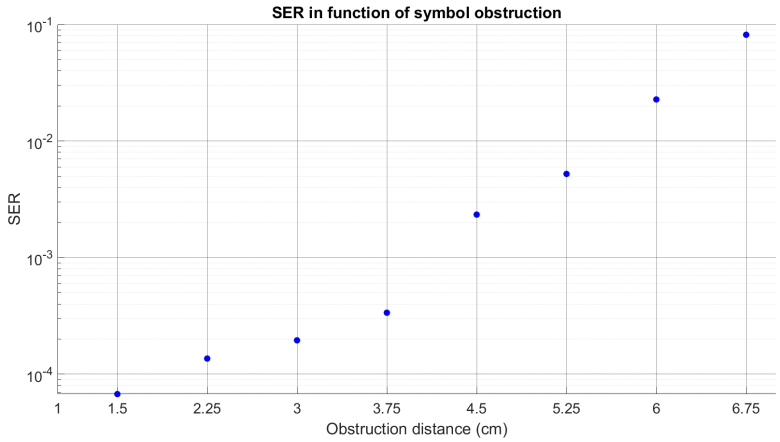


Figure 4.4: SER results for the symbol obstruction test.

transmission quality. This results were not expected, since the training dataset does not includes training data for this kind of situation. All the training photographs included the full ROI. Once again, the resilience to unusual conditions is evident.

4.3 MIMO ability

One of the features of this system is the ability to communicate between several transmitters and several receivers simultaneously, in other words, a MIMO system. In the system presented in this paper, this perspective can be seen as the communication between several light sources (rear or front lights) and one or more cameras. This has several advantages, as it allows to increase the data modulation capacity and consequently increasing the transmission rate. In addition, the use of the neural network as a tool for object detection and classification also enables to spatial locate the symbols within the frame boundaries. This property may allow, in a more advanced state of the work, to differentiate the light sources and assign them to different entities (automobiles) and thus encapsulate the received information.

Unfortunately, due to limitations of available material (all results were developed with access only to one taillight and one camera), it was impossible to mount a real MIMO system. To work around this problem, and to be able to at least verify the network's ability to identify and locate different symbols in the same frame, an alternative technique was applied. The technique consisted of capturing several frames with the raspberry camera, this one was aligned with the taillight at 3.5m away. After capturing the frames, and making use of tools for image manipulation (Photoshop), the ROI of each frame, was cut and then pasted in pairs in the same frame. The ratio size of each ROI was kept the same in all symbols. After this, each of these "new frames" was delivered as input to the network. As output was asked the identification of each symbol, the confidence of each of these identifications and the corresponding box prediction location.

A set of 16 different frames was created. All of them presented a correct classification of the two symbols present per frame. The location, as shown in Figure 4.5, is not 100% accurate, this is mainly due to the fact that the dataset used for network training is not large

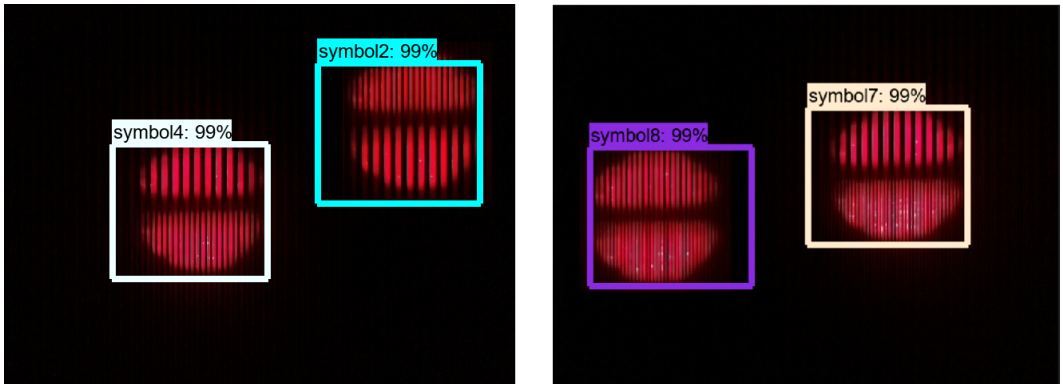


Figure 4.5: Two different "drawn frames" for MIMO capability testing. In each frame is exhibited the classification and location of each symbol.

enough, ROI labeling was not exact and the frame have a low resolution (only 640x480pixels). Either way the results show the system's MIMO capability.

4.4 Indoor and outdoor SER analysis in function of distance

The test consists of evaluating the transmission error limits in two different environments: indoor and outdoor. Another indicator with relevance in terms of system capacity is the range of distances acceptable to a properly communication. For this, maintaining the geometrical alignment between the transmitter (taillight) and the receiver (camera), the distance between both was been changed within a range $[0.5, 5.5]m$. The tests were first made inside a dark room and later outside in broad daylight.

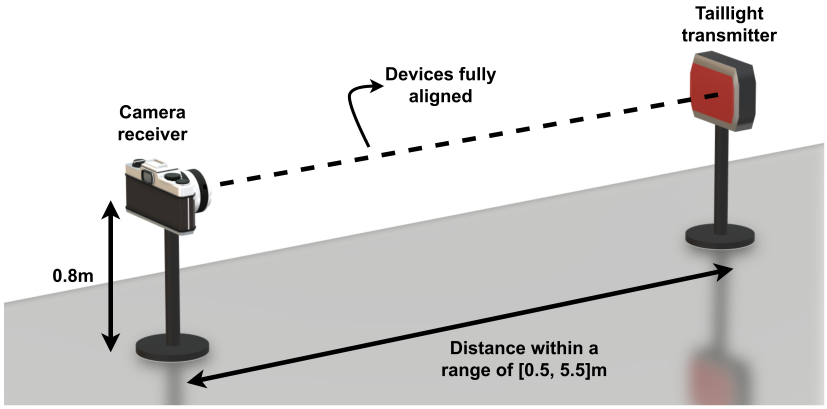
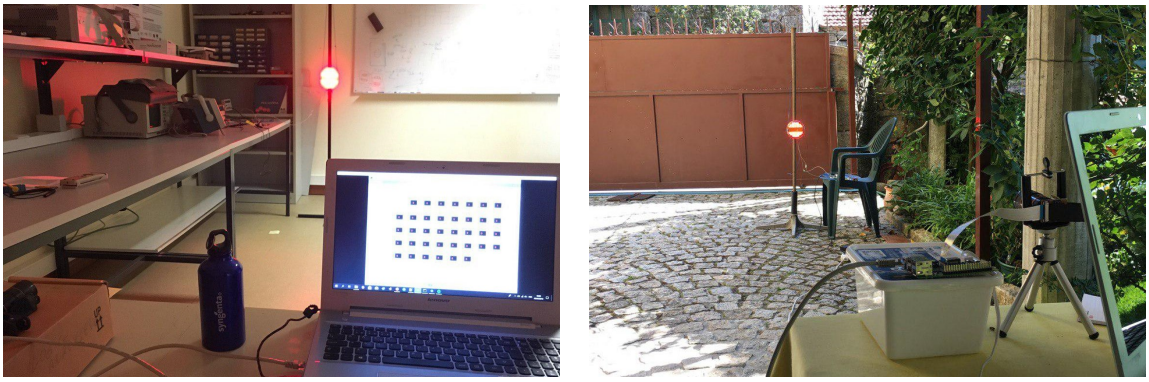


Figure 4.6: SER in function of distance, test sketch

With a range of distances between 0.5 and 5.5m, the measured pace was 0.5m, resulting in 11 distinct points. As measurement limit for the SER, it was defined the 10^{-6} order, meaning that the continuous transmission should contain at least 1 million symbols. Using a symbol rate of 15 symbols per second (15 FPS), this corresponds in time to approximately 19 hours of transmission for each measured point.



(a) Indoor test conditions.

(b) Outdoor test conditions.

Figure 4.7: Photographs from both test environments.

The test when performed indoor, suffers the impact of the artificial illumination within the room. It has a low brightness value (300lux), and can be considered constant over time. However, when measurements are made in an outdoor environment, the system is subjected to uncontrollable disturbances that vary its impact over time, as the primary source is the Sun. As its position varies along the day, and considering the camera and taillight positions as static, the angle of incidence on both devices will vary, which means that their impact is not constant over time.

The asynchronous phenomenon already discussed in subsubsection 3.5.2.1, in a continuous and long transmission like the one presented here, would affect the correct measurement of the SER, to avoid that, once again, the MATLAB®script developed to remove this hazard was applied.

The transmitted symbols were randomly generated, the symbol 2 is reserved for delimiting the data packet size (as explained before).

The results obtained can be visualized in the Figure 4.8.

As expected, the results for very close ($< 1m$) and very far ($> 4.5m$) distances between the camera and the taillight show a SER with a order of 10^{-1}. This is due to the fact that when camera is close to the taillight, the symbol (ROI) will occupy the entire area of the frame. For far distances, the ROI of the symbol is very small, which in a situation where the symbol in question makes use of the frequency of 2 kHz (which generates as is known thicker stripes), may lead to the ROI not containing enough stripes needed for a correct CNN decoding. Between these two values, it appears that the results follow a very similar pattern for both test situations. However and as expected, in the indoor situation, the results are better. Note also that the best results for both conditions are for distances around [3, 4]m, exactly the distance at which all images of the symbols that were used to constitute the training dataset were captured.

For the outdoor test, it was not indicated any exact value for the luminous intensity radiated by the Sun. This is due to the fact that the test duration is too long, resulting in a transmission environment that faces different stages of the day (morning, afternoon and part of the night). This means, that the luminous intensity received cannot be represented as an exact

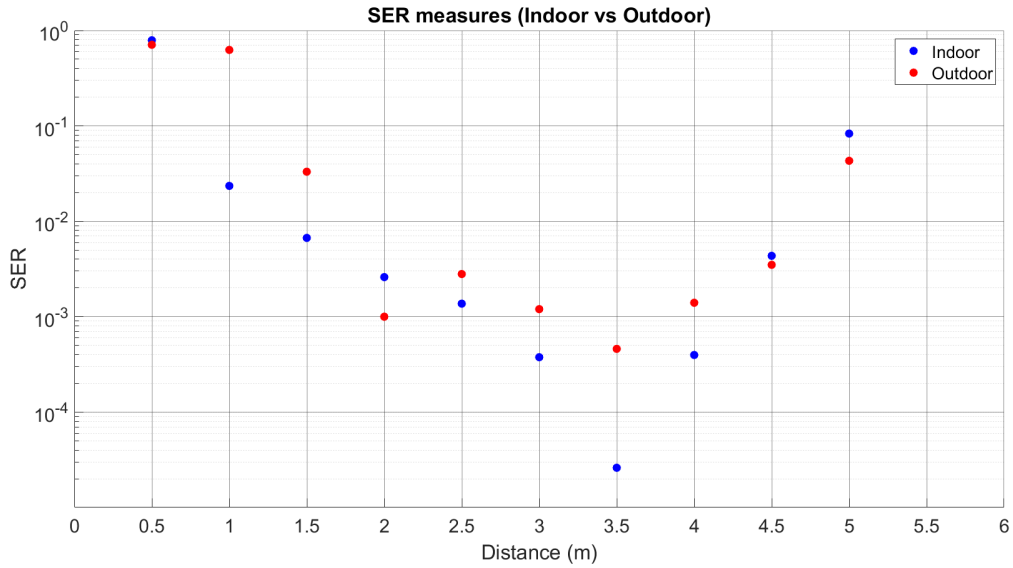


Figure 4.8: SER results in function of distance for both (indoor and outdoor) conditions.

value, but instead by a maximum and minimum limit. During the day, the maximum value measured was around 110000lux, at midday with clear blue sky, the minimum value (during day time) was 40lux at sunset. This tests were performed during the Summer (July/August), when the effects of solar light are stronger. However, should be noted, that in terms of impact in the communication transmission quality, is not during the period of maximum luminous intensity (midday) that its effect is more destructive. This instead, happens during the sunrise (early morning) and sunset (afternoon ending), which is exactly when the Sun position is more oblique to the earths surface. During this periods, sun rays are almost aligned with the individual system components (camera or taillight), resulting in a glare effect, and consequently increase of the noise value.

4.5 Concluding remarks

The results show the system's resilience in various conditions. Even for situations that were not considered as a requirement in the initial system design, the results show its capability to overcome this constrains. For the outdoor test, at distances different from the one for which the network was previously trained, the quality of the transmission declines. This shows that the correct choice of training dataset has a huge impact on CNN performance. However, the ability of the developed system as a viable communication solution in a real outdoor environment is evident, considering all the impositions (link distance, luminous intensity radiated, outdoor resilience, among others factors).

CHAPTER 5

Conclusion

5.1 Conclusions

The main objective of this dissertation was the study of VLC/OCC as communication technology applied to ITS, and the elaboration of a setup for real outdoor application. As such and in order to obtain the necessary bases for the correct development of the system, a state of the art research work was first carried out. During this phase, it was focused on the study of the fundamental units that make part of the whole system (LEDs and camera), as well as their dynamic characteristics. In addition, simulations were also performed in order to better understand the behavior of the communication channel. The conclusions and results of this phase are presented in chapter two.

After the research and study phase concluded, the communication setup was developed. The transmitter was the first subsystem to be designed and assembled, then the receiver, composed by the camera and the processing unit (in this case a computer). Some barriers were found, which were properly analyzed and documented. All this considerations are exposed in chapter three, as well as all process documentation for the design of the entire system.

After the whole system conceived, several tests were performed to verify the performance and limits of the communication link (alignment, obstruction of symbols, among others). In order to verify if the developed setup offered the necessary resilience to the disturbances imposed by the real outdoor environment, exhaustive SER transmission tests were performed. Chapter four it is composed of these results and subsequent analysis.

In conclusion and doing an overall looking for all the work done, it can be seen that this technology, characterized by the modulation of visible light as communication medium, has an enormous potential and much to explore further. Its application is not reserved for ITS, but for all other environments where both lighting and data transmission are two fundamental and necessary concepts. However, and looking at the purpose of this work, several obstacles were faced, from choosing a modulation technique appropriate to the imposed needs (UNECE regulation, physical limitations of the used taillight and eye safety), as well as problems in the synchronization of the transmitter with receptor. Moreover, the low bitrate of the system

projected in this work can be considered in contradiction to the actual standards. However, it should be noted that the prototype developed is intended to be applied as a redundant communication solution, where high transmission rates are not an imperative necessity while the ability to behave as an alternative and viable solution is. And for this, the results obtained in this dissertation are favorable: SER ratios of the order of 10^{-4} for outdoor environments with strong disturbance of sunlight and at link distances close to 4-5 meters.

Also, the fact that all the implemented system was made with off-the-shelf components (raspberry, camera, taillight driver and portable computer), shows the inherent hardware limitations that all the setup faced, but most importantly, the ability to overcome this constrains and present satisfactory results. This also means, that investing a little bit more (better cameras and processing unit) could consequently increase the system capabilities (symbol-rate, link distances, etc). Another factor to highlight is the usage of components that are already applied into the traffic system: LED taillight (that was untouched in terms of mechanic assembly to fulfill the traffic regulations), and the digital cameras, that are presented in front of all modern cars. This means, that this type of communication is ready to be applied in the real world, without the need of remarkable modifications in the actual interested clients (cars and traffic lights).

Concluding, all the results and analysis presented in this work undeniably show the suitability of this system as a communication link between V2V or V2I, imposing itself as a solution to the problem of RF spectrum overcrowding.

5.2 Future work

This dissertation presents a proof of concept and the results show its viability. However, there are several details that if refined can greatly improve the capabilities of the work already done. In the receiving block, the optimization of the processing and decoding algorithm, by creating from "zero" a CNN properly designed for this problem (recognition of the transmitted symbols) as well as increase the training dataset (with samples at different distances, alignments and brightness levels). This solution may improve both the data transmission rate and the allowable communication range. In addition, implementing another type of solution for frame capture, other than Raspberry Pi (for example an USB WebCam), in order to circumvent the frame-rate acceleration phenomenon and consequent system non-synchronization. Finally, augment the number of available symbols (either by increasing the frequencies or increasing the different independent areas of the head/taillight). This solution, if implemented can improve the data transmission rate.

APPENDIX A

Annex

A.1 Light measurement: Radiometry and Photometry

The physical properties of electromagnetic radiation throughout the spectrum are characterized by radiometric units [21]. Radiometric units can characterize light in terms of physical quantities: photon number and energy, optical power, etc. However, when considering the perception of the human eye, radiometric units become irrelevant. To characterize the light and color perception detected by the human eye another type of units is required. These units are the photometric units [59]. Photometry is a sub field of Radiometry, measures are obtained from the corresponding radiometric measures through a weighted average. The weighting function can consider the human visual perception in one of the three defined conditions:

- **Photopic vision** - relates to human vision at high ambient light levels (e.g. during daylight), it applies to luminance levels $> 3cd/m^2$.
- **Scotopic vision** - relates to human vision at low ambient light levels (e.g. at night). The color senstivity decreases in this regime. It applies to luminance levels $< 0.003cd/m_2$.
- **Mesopic vision** - relates to to light levels between the photopic and scotopic vision regime ($0.003cd/m_2 < mesopicluminance < 3cd/m_2$)

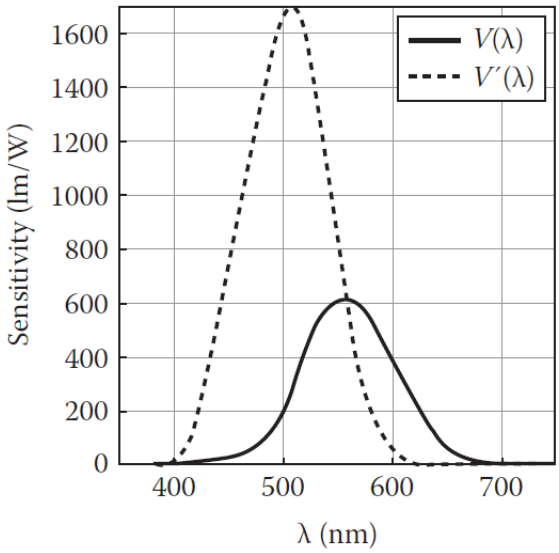


Figure A.1: Photopic and Scotopic human eye sensitivity [21]

.

Figure A.1 shows both curves for Photopic and Scotopic regimes. It can be seen the differences: the Scotopic curve ($V(\lambda)$) has a higher wavelength peak, however it has a lower sensitivity compared to the Photopic curve ($V'(\lambda)$). The Mesopic regime, which relates the intermediate of the two situations, is expressed by Equation A.1 as a linear combination between the other two regimes:

$$V_m(\lambda) = (1 - x) \cdot V'(\lambda) + x \cdot V(\lambda) \tag{A.1}$$

Mesopic curve is used for light simulations because it is in a middle position. The value of x is a constant ranging from 0 to 1, depending on photopic luminance.

- ***Radiant and Luminous Flux:***

  Radiant flux (Φ_e) or radiant power, is a radiometric unit. Is the power (sweep of the entire spectrum) that reaches a certain surface, it's expressed in watts:

$$\Phi_e(W) = \frac{dQ_e}{dt} \tag{A.2}$$

  In Equation A.2, Q_e means the total power emitted by a radiant source per unit time (t).

  Luminous flux (Φ_v) is a photometric unit. Characterizes the quantity of energy emitted by a light source per second in all directions. The unit of luminous flux is Lumen (lm):

$$\Phi_v(lm) = K \cdot \int_{380}^{780} P_e(\lambda) \cdot V(\lambda) \cdot d\lambda \tag{A.3}$$

  In Equation A.3, K represents a constant (683 lm/W for photopic), $P_e(\lambda)$ means the Power Spectrum Distribution of the light source and $V(\lambda)$ comes from Figure A.1 which

represents the sensitivity function. The integral appears because only the energy values for the visible light spectrum (380 to 780 nm) are of interest.

- ***Radiant and Luminous intensity:***

  Radiant intensity (I_e) is a Radiometric unit, which represents the emitted, reflected, or received radiant flux per unit solid angle (steradian), its unit is watt per steradian (W/sr):

$$I_e = \frac{d\Phi_e}{d\Omega} \tag{A.4}$$

  In Equation A.4 Φ_e represents the emitted power per unit solid angle (Ω).

  Luminous intensity (I_v) measures the emitted light into a given direction, in other words, is the luminous flux that is radiated by the light source in a given direction within the unit of the spacial angle. If a light point source emits (Φ_v) lumens into a small spatial angle $d\Omega$, the luminous intensity is then $I_v = d\Phi_v/d\Omega$. The S.I unit is lm/sr or candela.

- ***Irradiance and Illuminance:***

  In radiometry, irradiance (E_e)is the radiant flux received by a surface per unit area. The S.I. unit of irradiance is the watt per square meter (W/m^2):

$$E_e(W/m^2) = \frac{d\Phi_e}{dA} \tag{A.5}$$

  Illuminance (E_v) or illumination, determines the amount of light that covers a surface. If Φ_v is the luminous flux and A is the area of the given surface, then the illuminance (E_v) is determined by $E_v = d\Phi_v/dA$. The S.I. unit is the lumen per square meter or lux.

- ***Radiance and luminance:***

  Radiance (L_e) is a radiometric quantity, is usually used for extended light sources (sources that cannot be described as point source). Defined as the radiant flux (Φ_e) emitted from as area (A) per unit of solid angle (Ω). Its S.I. unit is the watt per steradian per square meter:

$$L_e(W/m^2) = \frac{d^2\Phi_e}{dA \cdot d\Omega} \tag{A.6}$$

  Luminance (L_v) is the luminous intensity emiited by the surface area of $1m^2$ of the light source. Mathematically, it's $L_v = dI_v/dA$, where I_v is the luminous intensity and A is the area of the surface area perpendicular to the source in a given direction. The S.I. unit is the candela per square meter (cd/m^2).

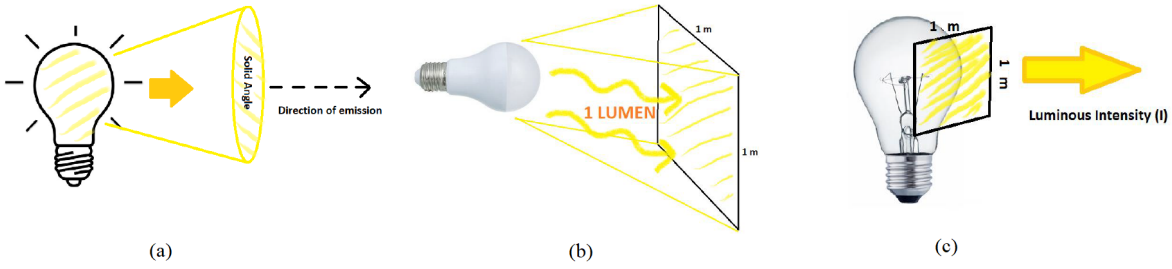


Figure A.2: Photometric units: (a) Luminous intensity (b) Illuminance (c) Luminance

A.2 WP.29 regulations

The World Forum for Harmonization of Vehicles Regulations is a working party (WP.29) of the Sustainable Transport Division of the United Nations Economic Commission for Europe. The purpose of this delegation is to respect, legislate and apply the terms signed in the 1958, 1997 and 1998 agreements for the control of the technical characteristics of construction and circulation of wheeled vehicles circulating within the countries covered by the European Union Agreement. Many non-European countries (Egypt, Japan, Australia...) are now contracting parties to the 1958 Agreement, the regulations are officially entitled "UN Regulations". As of 2015, there are 135 UN Regulations appended to the 1958 Agreement, most regulations cover a single vehicle component or technology.

The regulation number 112 is the one responsible for Headlamps Asymmetric lightning control. Described as:" Uniform provisions concerning the approval of motor vehicle headlamps emitting an asymmetrical passing beam or a driving beam or both and equipped with filament lamps and/or light-emitting diode (LED) modules".

This document regulates all the details inherent to the vehicle lighting setup, from its components (lenses, coatings ...), lighting, color, as well as the tests for the necessary approval.

Some (of the many) points required by this legislation are:

1. Headlights equipped with LED modules must comply the requirements from annex 10 from regulation n° 112.
2. The combination of LED with other light sources is not allowed.
3. LED modules that are not homologated as replaceable must in no condition be changed.
4. Light intensity produced by the headlight must be measured at 25m from the source using a photoelectric cell with a side length of 65mm. LED modules should be measured in voltage conditions of 6 3 V, 13 2 V or 28 V.
5. The luminous flux produced by the main LED beam modules shall be measured as described in paragraph 5 of Annex 10 and shall have a value of 1000 lumens or more.
6. In addition, to LED modules, headlamps shall be checked by using a standard colorless filament lamp designed for a nominal voltage of 12V.

Measurements are made on a test bench, where the luminaire is properly aligned with the center of the receiving plane. Measurement points in the receiving plane are placed in spherical coordinates relative to the headlamp.

In terms of emission patterns and brightness limits, the regulations are divided into two branches: right-hand drive or left-hand drive. During this application, the headlamps must meet different test values. Within Europe, when driving a vehicle with right-traffic headlamps in a left-traffic country or vice versa for a limited time (as for example on vacation or in transit), it is a legal requirement to adjust the headlamps temporarily so that their wrong-side beam distribution does not dazzle oncoming drivers. This may be achieved by methods including adhering opaque decals or prismatic lenses to a designated part of the lens.

The luminous intensity distribution of the principal passing beam headlamp (low beam) shall incorporate a “cut-off” (see Figure A.3), which enables the headlamp to be adjusted correctly for the photometric measurements and for the aiming on the vehicle. The "cut-off"

shall provide (for right hand traffic beams): (i) a straight "horizontal part" towards the left; (ii) a raised "elbow-shoulder" part towards the right.

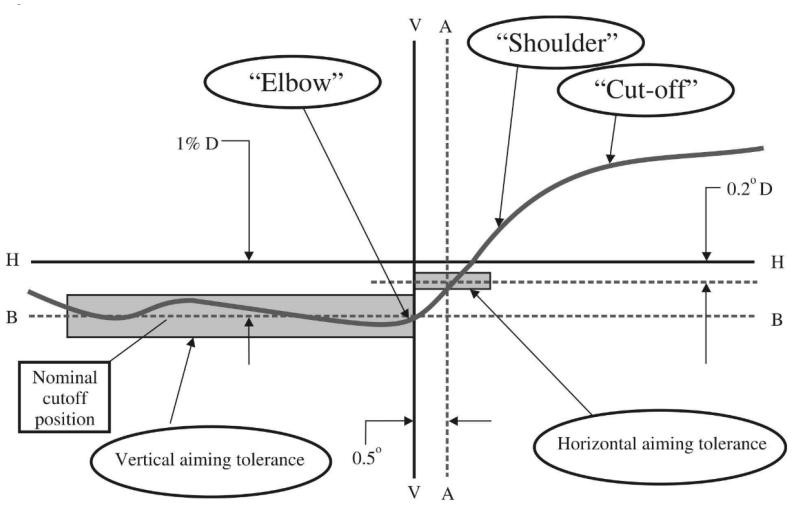


Figure A.3: "Cut-off" frontal view for a right hand passing beam (low beam).

The illumination levels produced on the screen by the passing beam (for right hand drive) shall meet the requirements of the Figure A.4 and Table A.1:

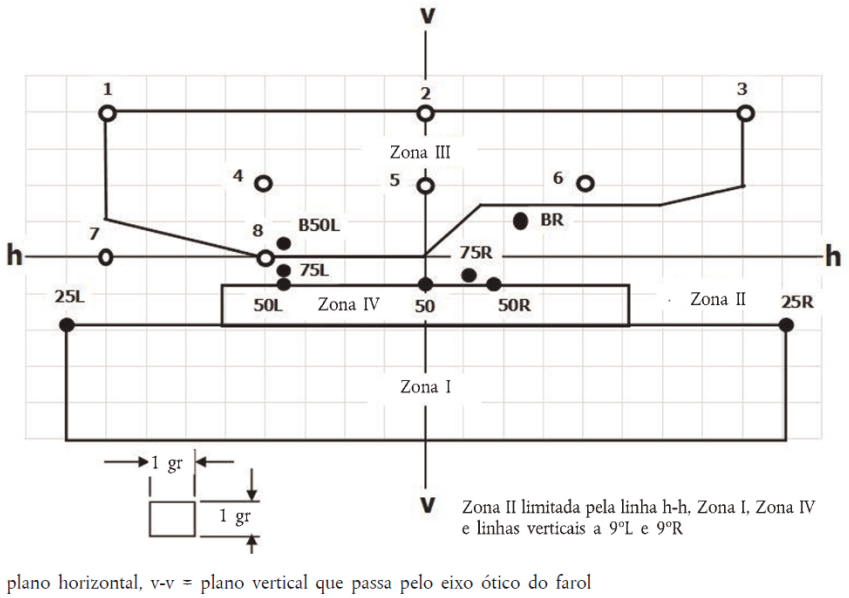


Figure A.4: Geometrical distribution of the measurement points for a right-hand low beam drive headlight.

Table A.1: Brightness limits for each control point (U-Up, D-Down, R-Right, L-Left).

Point:	Coordinates (m)	Max (Lux):	Min (Lux):
B50L	0.25U, 1.50L	0.56	
BR	0.44U, 1.09R	2.8	
75R	0.25D, 0.50R		8.16
75L	0.248D, 0.75L	19.96	
50L	0.37D, 0.75L	21.12	
50R	0.37D, 0.75R		8.16
25L	0.75D, 3.9596L		2
25R	0.75D, 3.95R		2
zone IV			2.72
zoneI		28.16	

The illumination levels produced on the screen by the high beam (for right hand drive) shall meet the requirements of the Figure A.5 and Table A.2:

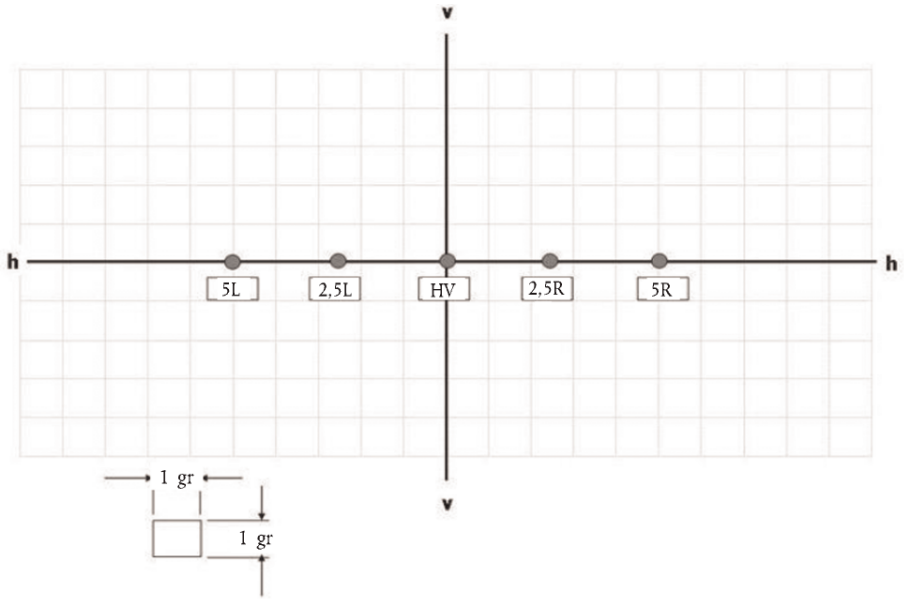


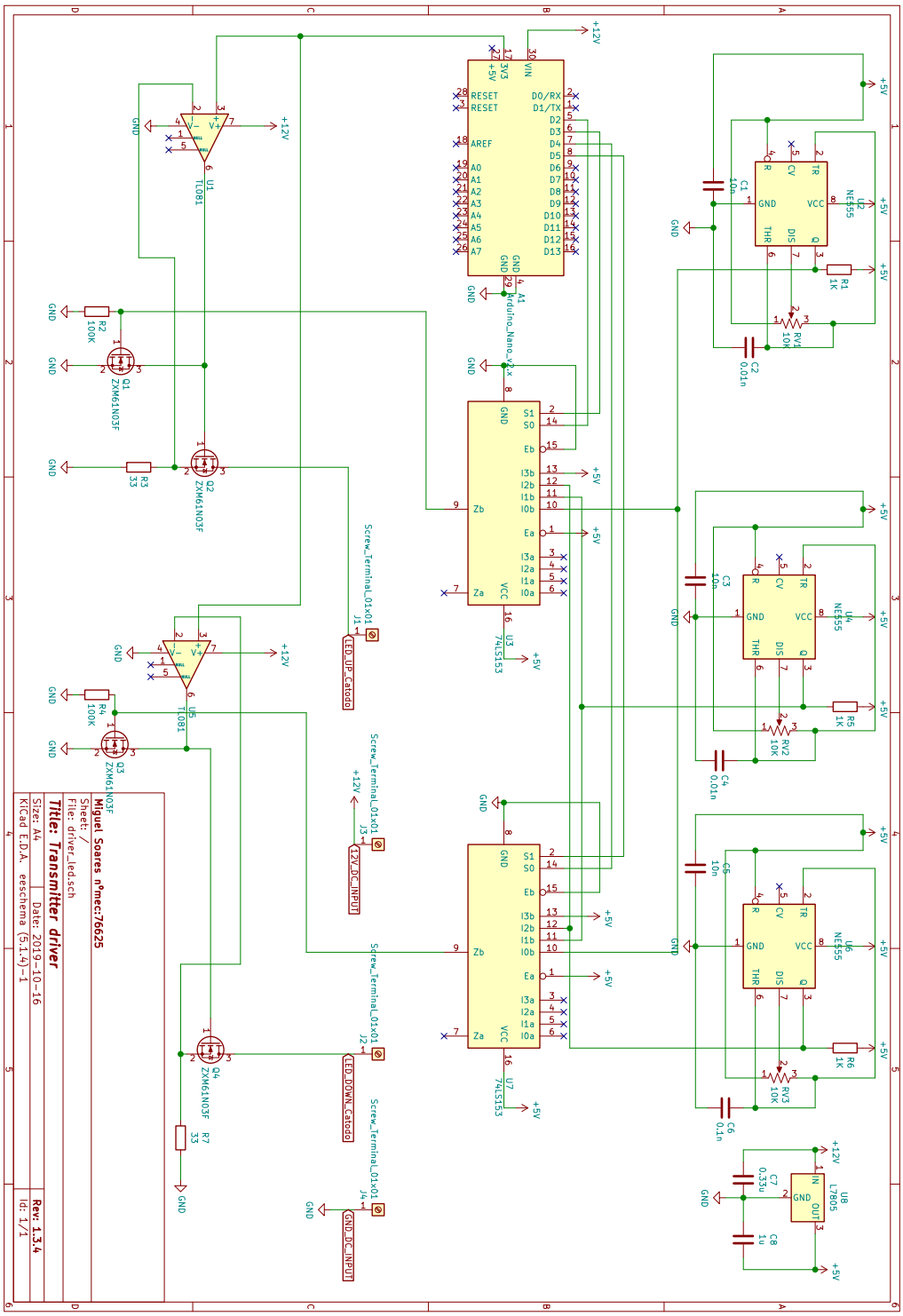
Figure A.5: Geometrical distribution of the measurement points for a right-hand drive high beam headlight.

Table A.2: Brightness limits for each control point (U-Up, D-Down, R-Right, L-Left).

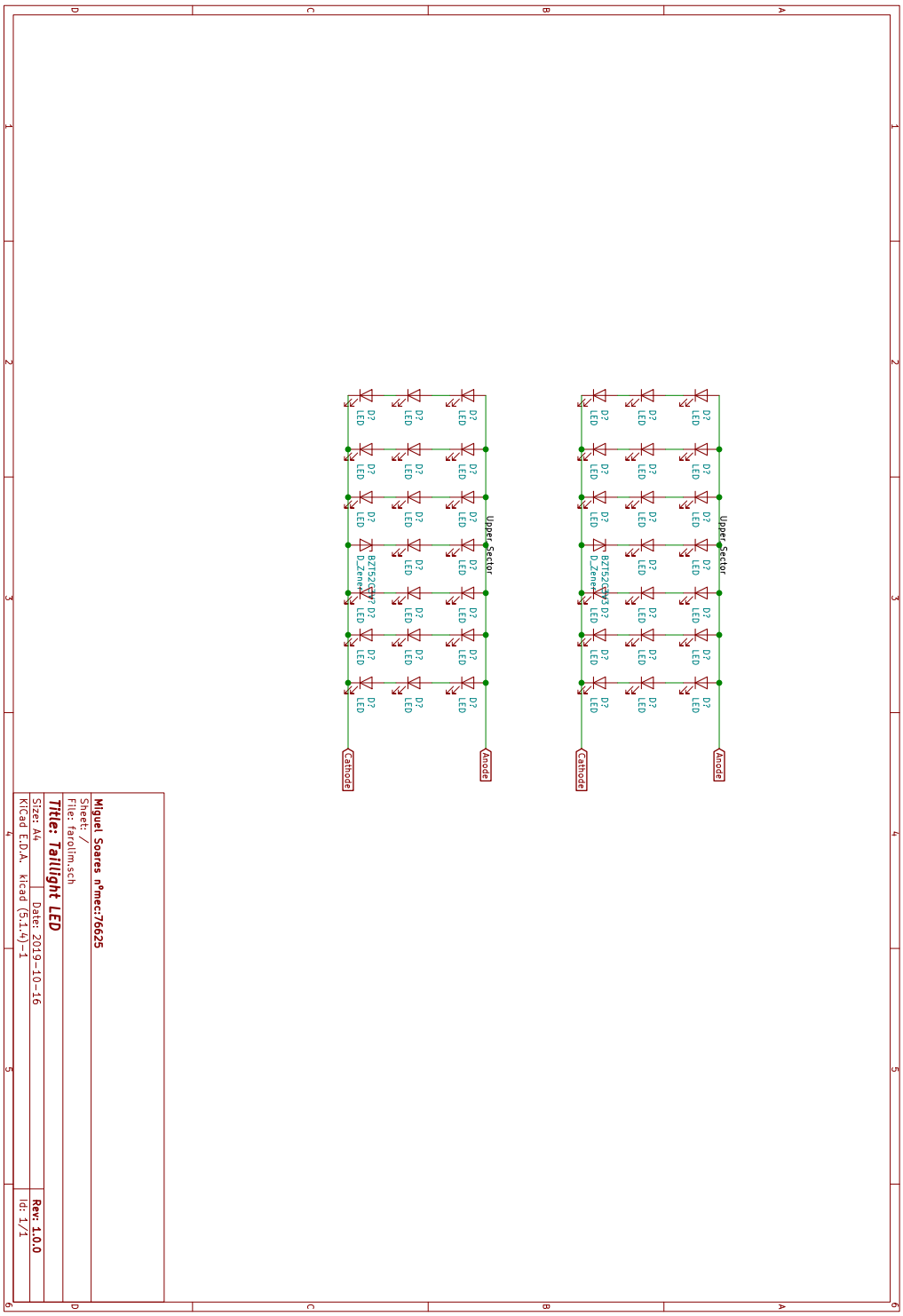
Point:	Coordinates (m)	Min (Lux):
Imax(HV)		43.2
BR	0U, 2.1872L	5.44
75R	0U, 1.0915L	21.6
75L	0U, 1.0915R	21.6
50L	0U, 2.1872R	5.44

For further details, it is advisable to consult the full regulation: **https://www.unece.org/fileadmin/DAM/trans/main/wp29/wp29regs/R112rev2_e.pdf**.

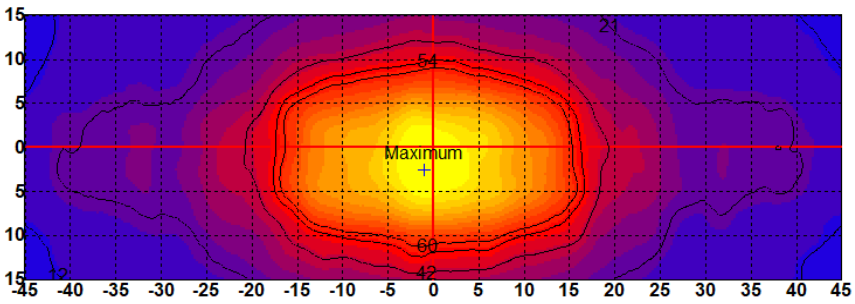
A.3 Driver circuit schematic



A.4 Taillight circuit schematic



A.5 Homologation report - Stop light luminous intensity



Abs.: 12 21 42 54 60 cd

Aspöck	Program: ECE R7 Stop Lamp 100% LED's		Maximum: 125.26
	Name: 18/4052_Hamburguer_S1	Lamp type:	H: -1.00°
	Number: 18/4052	Lamp no.:	V: -2.50°
	Report: 180726_Hamburguer_S1_T05	Lamp flux: 0.00 lm	
	Test No.: 05	Voltage: 13.492 V	Operator: Joana S
Isoline diagram	Comment: sample 1.	Current: 0.097 A	Date: 26/07/2

Mod. F. 414-00/10

A.6 Raspberry Pi client code

```
1
2  import io
3  import socket
4  import struct
5  import time
6  import picamera
7  from fractions import Fraction
8  #import errno
9  #from socket import error as socket.error
10 finish=0
11 start=0
12 class SplitFrames(object):
13     def __init__(self, connection):
14         self.connection = connection
15         self.stream = io.BytesIO()
16         self.count = 0
17
18     def write(self, buf):
19         if buf.startswith(b'\xff\xd8'):
20             #Start of new frame; send the old one's length
21             #then the data
22             size = self.stream.tell()
23             if size > 0:
24                 self.connection.write(struct.pack('<L', size))
25                 self.connection.flush()
26                 self.stream.seek(0)
27                 self.connection.write(self.stream.read(size))
28                 self.count += 1
29                 self.stream.seek(0)
30             self.stream.write(buf)
31
32 client_socket = socket.socket()
33 client_socket.connect(('192.168.137.1', 8000))
34 connection = client_socket.makefile('wb')
35 try:
36     output = SplitFrames(connection)
37     with picamera.PiCamera(resolution = (640,480),framerate = 15) as camera:
38         camera.exposure_mode = 'auto' #off
39         camera.awb_mode = 'off'
40         #camera.iso=1000#300
41         #camera.brightness=40
42         #camera.clock_mode = 'raw'
43         camera.shutter_speed = 1
44         camera.rotation = 90
45         #camera.sensor_mode = 1
46         #camera.framerate_delta = 1/256
47         #camera.zoom = (0.3,0.3,0.7,0.7)
48         time.sleep(3)
```

```
49          start = time.time()
50          #camera.sensor_mode = 7
51          camera.start_recording(output, format='mjpeg')
52          camera.wait_recording(10)
53          camera.stop_recording()
54          finish = time.time()
55          #Write the terminating 0-length to the connection to let the
56          # server know we're done
57          connection.write(struct.pack('<L',0))
58          #print(finish-start)
59
60  except socket.error:#finally:
61      connection.close()
62      client_socket.close()
63      print('Sent %d images' % (output.count))
64      print(finish-start)
65  finally:
66      print('Sent %d images' % (output.count))
67      print(finish-start)
```

Listing A.1: Python code for Raspberry Pi client configuration

A.7 Convolutional Neural Network configuration code.

```
model {
  ssd {
    num_classes: 9
    image_resizer {
      fixed_shape_resizer {
        height: 300
        width: 300
      }
    }
    feature_extractor {
      type: "ssd_mobilenet_v2"
      depth_multiplier: 1.0
      min_depth: 16
      conv_hyperparams {
        regularizer {
          l2_regularizer {
            weight: 3.9999998989515007e-05
          }
        }
        initializer {
          truncated_normal_initializer {
            mean: 0.0
            stddev: 0.029999999329447746
          }
        }
        activation: RELU_6
        batch_norm {
          decay: 0.9997000098228455
          center: true
          scale: true
          epsilon: 0.0010000000474974513
          train: true
        }
      }
      use_depthwise: true
    }
    box_coder {
      faster_rcnn_box_coder {
        y_scale: 10.0
        x_scale: 10.0
        height_scale: 5.0
        width_scale: 5.0
      }
    }
    matcher {
      argmax_matcher {
        matched_threshold: 0.5
        unmatched_threshold: 0.5
```

```
      ignore_thresholds: false
      negatives_lower_than_unmatched: true
      force_match_for_each_row: true
    }
  }
  similarity_calculator {
    iou_similarity {
    }
  }
  box_predictor {
    convolutional_box_predictor {
      conv_hyperparams {
        regularizer {
          l2_regularizer {
            weight: 3.9999998989515007e-05
          }
        }
        initializer {
          truncated_normal_initializer {
            mean: 0.0
            stddev: 0.029999999329447746
          }
        }
        activation: RELU_6
        batch_norm {
          decay: 0.9997000098228455
          center: true
          scale: true
          epsilon: 0.0010000000474974513
          train: true
        }
      }
      min_depth: 0
      max_depth: 0
      num_layers_before_predictor: 0
      use_dropout: false
      dropout_keep_probability: 0.800000011920929
      kernel_size: 3
      box_code_size: 4
      apply_sigmoid_to_scores: false
      use_depthwise: true
    }
  }
  anchor_generator {
    ssd_anchor_generator {
      num_layers: 6
      min_scale: 0.20000000298023224
      max_scale: 0.949999988079071
      aspect_ratios: 1.0
      aspect_ratios: 2.0
```

```
        aspect_ratios: 0.5
        aspect_ratios: 3.0
        aspect_ratios: 0.33329999446868896
      }
    }
    post_processing {
      batch_non_max_suppression {
        score_threshold: 9.99999993922529e-09
        iou_threshold: 0.6000000238418579
        max_detections_per_class: 100
        max_total_detections: 100
      }
      score_converter: SIGMOID
    }
    normalize_loss_by_num_matches: true
    loss {
      localization_loss {
        weighted_smooth_l1 {
        }
      }
      classification_loss {
        weighted_sigmoid {
        }
      }
      hard_example_miner {
        num_hard_examples: 3000
        iou_threshold: 0.9900000095367432
        loss_type: CLASSIFICATION
        max_negatives_per_positive: 3
        min_negatives_per_image: 3
      }
      classification_weight: 1.0
      localization_weight: 1.0
    }
  }
}
train_config {
  batch_size: 15
  data_augmentation_options {
    random_horizontal_flip {
    }
  }
  data_augmentation_options {
    ssd_random_crop {
    }
  }
  optimizer {
    rms_prop_optimizer {
      learning_rate {
        exponential_decay_learning_rate {
```

```
          initial_learning_rate: 0.004000000189989805
          decay_steps: 800720
          decay_factor: 0.949999988079071
        }
      }
      momentum_optimizer_value: 0.8999999761581421
      decay: 0.8999999761581421
      epsilon: 1.0
    }
  }
  fine_tune_checkpoint: "ssdlite_mobilenet_v2_coco_2018_05_09/model.ckpt"
  num_steps: 200000
  fine_tune_checkpoint_type: "detection"
}
train_input_reader {
  label_map_path: "data/object-detection.pbtxt"
  tf_record_input_reader {
    input_path: "data/train.record"
  }
}
eval_config {
  num_examples: 8000
  max_evals: 10
  use_moving_averages: false
}
eval_input_reader {
  label_map_path: "data/object-detection.pbtxt"
  shuffle: false
  num_readers: 1
  tf_record_input_reader {
    input_path: "data/test.record"
  }
}
```

Listing A.2: Python code for CNN configuration.

A.8 Computer server code, with symbol decoding and storage.

```
1
2  import numpy as np
3  import os
4  import six.moves.urllib as urllib
5  import sys
6  import tarfile
7  import tensorflow as tf
8  import zipfile
9  import time
10 import io
11 import socket
12 import struct
13
14
15 from collections import defaultdict
16 from io import StringIO
17 from matplotlib import pyplot as plt
18 from PIL import Image
19
20 import cv2
21
22 from utils import label_map_util
23
24 from utils import visualization_utils as vis_util
25
26 # Start a socket listening for connections on 0.0.0.0:8000 (0.0.0.0 means
27 # all interfaces)
28 server_socket = socket.socket()
29 server_socket.bind(('0.0.0.0', 8000))
30 server_socket.listen(0)
31 #io.DEFAULT_BUFFER_SIZE = 1200000000
32 # Accept a single connection and make a file-like object out of it
33 connection = server_socket.accept()[0].makefile('rb')#'rb'
34
35 # What model to download.
36 MODEL_NAME = 'simbol_zoom'
37
38 # Path to frozen detection graph. This is the actual model that is used for
      the object detection.
39 PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'
40
41 # List of the strings that is used to add correct label for each box.
42 PATH_TO_LABELS = os.path.join('data', 'object-detection.pbtxt')
43
44 NUM_CLASSES = 9
45
46 # ## Download Model
47
```

```
detection_graph = tf.Graph()
with detection_graph.as_default():
  od_graph_def = tf.GraphDef()
  with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
    serialized_graph = fid.read()
    od_graph_def.ParseFromString(serialized_graph)
    tf.import_graph_def(od_graph_def, name='')

# ## Loading label map
# Label maps map indices to category names, so that when our convolution network predicts `5`,
#we know that this corresponds to `symbol5`.  Here we use internal utility functions, but anything that
#returns a dictionary mapping integers to appropriate string labels would be fine

# In[7]:

label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map, max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)

# In[8]:

def load_image_into_numpy_array(image):
  (im_width, im_height) = image.size
  return np.array(image.getdata()).reshape(
      (im_height, im_width, 3)).astype(np.uint8)

try:
  #cnt = 0
  buf = []

  file = open('rx.txt','w')#'wt'
  with detection_graph.as_default():
    with tf.Session(graph=detection_graph) as sess:
      while True:
        image_len = struct.unpack('<L', connection.read(struct.calcsize('<L'))[0]

        image_stream = io.BytesIO()
        image_stream.write(connection.read(image_len))

        image_stream.seek(0)
        image = Image.open(image_stream)
        #ret, image_np = cap.read()
        # Expand dimensions since the model expects images to have shape: [1, None, None, 3]
        image_np_expanded = np.expand_dims(image, axis=0)
```

```
92          image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')
93
94          scores = detection_graph.get_tensor_by_name('detection_scores:0')
95          classes = detection_graph.get_tensor_by_name('detection_classes:0')
96          # Actual detection.
97          (scores, classes) = sess.run([scores, classes],feed_dict={
    image_tensor: image_np_expanded})
98
99          if scores[0][0] > 0.5:
100           ##print(str(classes[0][0].astype(int)))
101           file.write(classes[0][0].astype(str)[0]) # Buscar primeiro elemento
102           #file.write(str(1))
103           #cnt=cnt+1
104         else:
105           file.write(str(0))
106           ##print(0)
107         #for i in range(5):
108         # if scores[0][i] > 0.015:
109         #   print(classes[0][i])
110         #print(cnt)
111         #cnt = cnt + 1
112 except KeyboardInterrupt:
113   file.close()
114   connection.close()
115   server_socket.close()
116   print(cnt)
```

Listing A.3: Python code for computer server decoding.

A.9 Microcontroller - random symbol generation code

```
volatile int S0I = HIGH;
volatile int S1I = HIGH;
volatile int S0S = HIGH;
volatile int S1S = HIGH;
long randNumber;
void setup()
{
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  Serial.begin(9600);

  randomSeed(analogRead(0));

// TIMER 1 for interrupt frequency 15.00000002400096 Hz:
cli(); // stop interrupts
TCCR1A = 0; // set entire TCCR1A register to 0
TCCR1B = 0; // same for TCCR1B
TCNT1  = 0; // initialize counter value to 0
// set compare match register for 15.00060002400096 Hz increments
OCR1A = 16665; // = 16000000 / (64 * 15.00060002400096) - 1 (must be <65536)
// turn on CTC mode
TCCR1B |= (1 << WGM12);
// Set CS12, CS11 and CS10 bits for 64 prescaler
TCCR1B |= (0 << CS12) | (1 << CS11) | (1 << CS10);
// enable timer compare interrupt
TIMSK1 |= (1 << OCIE1A);
sei(); // allow interrupts

}

ISR(TIMER1_COMPA_vect){

  randNumber = random(2,10);

  switch (1) {
  case 1:
    S0I = LOW;
    S1I = LOW;
    S0S = LOW;
    S1S = LOW;
    break;
  case 2:
    S0I = HIGH;
    S1I = LOW;
    S0S = LOW;
    S1S = LOW;
```

```
    break;
  case 3:
    S0I = LOW;
    S1I = HIGH;
    S0S = LOW;
    S1S = LOW;
    break;
  case 4:
    S0I = LOW;
    S1I = LOW;
    S0S = HIGH;
    S1S = LOW;
    break;
  case 5:
    S0I = HIGH;
    S1I = LOW;
    S0S = HIGH;
    S1S = LOW;
    break;
  case 6:
    S0I = LOW;
    S1I = HIGH;
    S0S = HIGH;
    S1S = LOW;
    break;
  case 7:
    S0I = LOW;
    S1I = LOW;
    S0S = LOW;
    S1S = HIGH;
    break;
  case 8:
    S0I = HIGH;
    S1I = LOW;
    S0S = LOW;
    S1S = HIGH;
    break;
  case 9:
    S0I = LOW;
    S1I = HIGH;
    S0S = LOW;
    S1S = HIGH;
    break;
}

  digitalWrite(3,S0I);
  digitalWrite(4,S1I);
  digitalWrite(5,S0S);
  digitalWrite(6,S1S);
  Serial.print(randNumber);
```

```
99 }
100
101 // Below mentioned code runs for ever(infinite loop)
102 void loop() {
103 }
```

Listing A.4: C++ code for Arduino microcontroller.

A.10 MATLAB - package asynchronous hazard detection script.

```
1
2  % IMPORTAR OS DOIS FICHEIROS A COMPARAR PRIMEIRO!!!
3  clc
4  clear
5
6  %tx = "21111123333324444425555526666627777728888829999992000002";
7  %rx = "21111123333322442445555526666627777728888829999992000002";
8
9  tx=split(tx,"")';
10 rx=split(rx,"")';
11
12 N = 124;%124
13 i_rx=[];
14 i_tx=[];
15 rx(1)=[];
16 tx(1)=[];
17 rx(length(rx))=[];
18 tx(length(tx))=[];
19
20 SS='2';
21 k=1;
22 m=1;
23
24 for i=1:length(tx)
25     if tx(i)==SS
26         i_tx(k)=i;
27         k=k+1;
28     end
29 end
30
31 for i=1:length(rx)
32     if rx(i)==SS
33         i_rx(m)=i;
34         m=m+1;
35     end
36 end
37
38 %%
39 dist=0;
40 buf_rx=[];
41 p=1;
42 i=1;
43 s=1;
44 e=1;
45
46 packet_error=0;
47 detection_error=0;
48 while i<=(length(i_rx)-1)
```

```
    p=0;
    ready=0;
    dist=i_rx(i+1)-i_rx(i);
    if dist==(N+1)
        buf_rx=rx((i_rx(i)+1):(i_rx(i+1)-1));
        jump=1;
        ready=1;
    else
        while dist~=(N+1)
            p=p+1;
            dist=(i_rx(i+p+1) - i_rx(i+p));
        end

        jump=floor((i_rx(i+p)-i_rx(i))/(N+1));
        i=i+p-1;
        packet_error=packet_error+jump*N;
        array_packer(i)=jump*N;
        %Array with the index of the erroneous received data packages

    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if ready==1
        buf_tx=tx((i_tx(s)+1):(i_tx(s+1)-1));
        detection_error=detection_error+numel(find(buf_rx~=buf_tx));
        %array_tx(e)=i_tx(s);
        %array_rx(e)=i_rx(i);
        array_detection(e)=numel(find(buf_rx~=buf_tx));
        e=e+1;
    end

    s=s+jump;
    i=i+1;
end
total=detection_error+packet_error;
```

Listing A.5: Matlab script for erroneous data size packages detection.

References

[1] S. Arnon, *Visible light communication*. Cambridge University Press, 2015.

[2] R. Dominguez, E. Onieva, J. Alonso, J. Villagra, and C. Gonzalez, "Lidar based perception solution for autonomous vehicles", in *2011 11th International Conference on Intelligent Systems Design and Applications*, IEEE, 2011, pp. 790–795.

[3] N. Kumar, N. Lourenco, M. Spiez, and R. L. Aguiar, "Visible light communication systems conception and vidas", *IETE Technical Review*, vol. 25, no. 6, pp. 359–367, 2008.

[4] E. Eso, Z. Ghassemlooy, S. Zvanovec, A. Gholami, A. Burton, N. B. Hassan, and O. I. Younus, "Experimental demonstration of vehicle to road side infrastructure visible light communications", in *2019 2nd West Asian Colloquium on Optical Wireless Communications (WACOWC)*, IEEE, 2019, pp. 85–89.

[5] G. Vidhya Krishnan, N. Ramalingam, T. Durka v, M. a, M. v, and S. Shanmuga priya, "Vehicle communication system using li-fi technology", *International Journal Of Engineering And Computer Science*, Mar. 2017. DOI: 10.18535/ijecs/v6i3.47.

[6] A. R. Ndjiongue, H. C. Ferreira, and T. M. Ngatched, "Visible light communications (vlc) technology", *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 1–15, 1999.

[7] J. Vučić, C. Kottke, S. Nerreter, K. Habel, A. Büttner, K.-D. Langer, and J. W. Walewski, "230 mbit/s via a wireless visible-light link based on ook modulation of phosphorescent white leds", in *2010 Conference on Optical Fiber Communication (OFC/NFOEC), collocated National Fiber Optic Engineers Conference*, IEEE, 2010, pp. 1–3.

[8] N. Fujimoto and H. Mochizuki, "477 mbit/s visible light transmission based on ook-nrz modulation using a single commercially available visible led and a practical led driver with a pre-emphasis circuit", in *National Fiber Optic Engineers Conference*, Optical Society of America, 2013, JTh2A–73.

[9] A. G. Bell, "Art. xxxiv.–on the production and reproduction of sound by light", *American Journal of Science (1880-1910)*, vol. 20, no. 118, p. 305, 1880.

[10] ——, *Diagram of the Photophone. The image is taken from Alexander Graham Bell's 1880 paper.* 1880. [Online]. Available: http://histv2.free.fr/bell/bell5.htm.

[11] N. Zheludev, "The life and times of the led—a 100-year history", *Nature photonics*, vol. 1, no. 4, p. 189, 2007.

[12] T. S. Perry, "Red hot [light emitting diodes]", *IEEE Spectrum*, vol. 40, no. 6, pp. 26–29, 2003.

[13] T. Komine and M. Nakagawa, "Integrated system of white led visible-light communication and power-line communication", *IEEE Transactions on Consumer Electronics*, vol. 49, no. 1, pp. 71–79, 2003.

[14] J. Mundy and S. Kavanagh, *What is li-fi?*, https://5g.co.uk/guides/what-is-li-fi/, Last accessed on 2019-7-14, 2017.

[15] N. Iizuka, *Optical communication repeater*, Japan patent: JP,2000-056545, Sep. 2001.

[16] S. Haruyama, "Advances in visible light communication technologies", in *European Conference and Exhibition on Optical Communication*, Optical Society of America, 2012, We–3.

[17] X. Wu, P. Liu, and S. Liu, "New structure of using image sensor communication in smart house with smart grid", in *The First International Conference on Future Generation Communication Technologies*, IEEE, 2012, pp. 32–35.

[18] N. Iizuka, "Image sensor communication—current status and future perspectives", *IEICE Transactions on Communications*, vol. 100, no. 6, pp. 911–916, 2017.

[19] I. Takai, T. Harada, M. Andoh, K. Yasutomi, K. Kagawa, and S. Kawahito, "Optical vehicle-to-vehicle communication system using led transmitter and camera receiver", *IEEE photonics journal*, vol. 6, no. 5, pp. 1–14, 2014.

[20] T. Yamazato, I. Takai, H. Okada, T. Fujii, T. Yendo, S. Arai, M. Andoh, T. Harada, K. Yasutomi, K. Kagawa, *et al.*, "Image-sensor-based visible light communication for automotive applications", *IEEE Communications Magazine*, vol. 52, no. 7, pp. 88–97, 2014.

[21] Z. Ghassemlooy, L. N. Alves, S. Zvanovec, and M.-A. Khalighi, *Visible light communications: theory and applications*. CRC press, 2017.

[22] H. Willebrand and B. S. Ghuman, *Free space optics: enabling optical connectivity in today's networks*. SAMS publishing, 2002.

[23] A. R. Ndjiongue and H. C. Ferreira, "An overview of outdoor visible light communications", *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 7, e3448, 2018.

[24] Z. Ghassemlooy, W. Popoola, and S. Rajbhandari, *Optical wireless communications: system and channel modelling with Matlab®*. CRC press, 2019.

[25] I. Moreno and C.-C. Sun, "Modeling the radiation pattern of leds", *Optics express*, vol. 16, no. 3, pp. 1808–1819, 2008.

[26] B. Turan and S. Ucar, "Vehicular visible light communications", *Visible Light Communications*, p. 133, 2017.

[27] H. Wang, J. Zhong, D. Feng, J. Meng, and N. Xie, "Nanoparticles-modified polymer-based solar-reflective coating as a cooling overlay for asphalt pavement", *International Journal of Smart and Nano Materials*, vol. 4, no. 2, pp. 102–111, 2013.

[28] I. Ashdown, "Photometry and photosynthesis: From photometry to ppfd (revised)", *Wordpress Blog*, Nov. 2015.

[29] Z. Wang, Q. Wang, W. Huang, and Z. Xu, *Visible light communications: Modulation and signal processing*. John Wiley & Sons, 2017.

[30] M. S. Islim, S. Videv, M. Safari, E. Xie, J. J. McKendry, E. Gu, M. D. Dawson, and H. Haas, "The impact of solar irradiance on visible light communications", *Journal of Lightwave Technology*, vol. 36, no. 12, pp. 2376–2386, 2018.

[31] T. Bangert, *Spectrum of sunlight*, [Online; accessed June 21, 2019], 2015. [Online]. Available: `http://www.eecs.qmul.ac.uk/~tb300/pub/Images/Spectrum_of_Sunlight_en.pdf`.

[32] C. Erlick, J. E. Frederick, V. Saxena, and B. N. Wenny, "Atmospheric transmission in the ultraviolet and visible: Aerosols in cloudy atmospheres", *Journal of Geophysical Research: Atmospheres*, vol. 103, no. D24, pp. 31 541–31 555, 1998.

[33] F. Nadeem, E. Leitgeb, M. S. Awan, and G. Kandus, "Optical wavelengths comparison for different weather conditions", in *2009 International Workshop on Satellite and Space Communications*, IEEE, 2009, pp. 279–283.

[34] M. Gebhart, E. Leitgeb, M. Al Naboulsi, H. Sizun, and F. de Fornel, "Measurements of light attenuation at different wavelengths in dense fog conditions for fso applications", 2004.

[35] *Multibeam led headlamps in the new e-class.* [Online; accessed September 17, 2019], Mar. 2016. [Online]. Available: `https://www.mercedes-benz.com/en/innovation/multibeam-led-headlamps-in-the-new-e-class-video/`.

[36] A. P. in History of alt. proc., *Pinhole history*, [Online; accessed September 26, 2019], Apr. 2017. [Online]. Available: http://www.alternativephotography.com/pinhole-history/.

[37] M. Bellis, *A comprehensive look at the history of photography*, [Online; accessed September 26, 2019], May 2019. [Online]. Available: https://www.thoughtco.com/history-of-photography-and-the-camera-1992331.

[38] M. S. Ifthekhar, M. A. Hossain, C. H. Hong, and Y. M. Jang, "Radiometric and geometric camera model for optical camera communications", in *2015 Seventh International Conference on Ubiquitous and Future Networks*, IEEE, 2015, pp. 53–57.

[39] A. Eisen-Enosh, N. Farah, Z. Burgansky-Eliash, U. Polat, and Y. Mandel, "Evaluation of critical flicker-fusion frequency measurement methods for the investigation of visual temporal resolution", *Scientific reports*, vol. 7, no. 1, p. 15 621, 2017.

[40] M. Hasan, M. Z. Chowdhury, M. Shahjalal, V. T. Nguyen, Y. M. Jang, *et al.*, "Performance analysis and improvement of optical camera communication", *Applied Sciences*, vol. 8, no. 12, p. 2527, 2018.

[41] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey", *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[42] A. P. Celso Paiva Sol, *Maior sinistralidade rodoviária. a culpa é só dos condutores?*, [Online; accessed August 5, 2019], 2018. [Online]. Available: https://rr.sapo.pt/2018/03/13/pais/maior-sinistralidade-rodoviaria-a-culpa-e-so-dos-condutores/noticia/108010/.

[43] M. Choudhary, M. ChoudharyCorrespondent, and M. Choudhary, *What is intelligent transport system and how it works*, Jan. 2019. [Online]. Available: https://www.geospatialworld.net/blogs/what-is-intelligent-transport-system-and-how-it-works/.

[44] W.-H. Shen and H.-M. Tsai, "Testing vehicle-to-vehicle visible light communications in real-world driving scenarios", in *2017 IEEE Vehicular Networking Conference (VNC)*, IEEE, 2017, pp. 187–194.

[45] A. E. Marcu, R. A. Dobre, and M. Vlădescu, "Key aspects of infrastructure-to-vehicle signaling using visible light communications", in *International Conference on Future Access Enablers of Ubiquitous and Intelligent Infrastructures*, Springer, 2017, pp. 212–217.

[46] [Online; accessed August 10, 2019]. [Online]. Available: https://store.arduino.cc/arduino-nano.

[47] *Picamera*, [Online; accessed October 17, 2019]. [Online]. Available: https://picamera.readthedocs.io/en/release-1.13/index.html.

[48] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[49] C. Ollion and O. Grisel. (). Convolutional neural networks, Heuritech and Inria, [Online]. Available: https://m2dsupsdlclass.github.io/lectures-labs/slides/04_conv_nets/index.html#1 (visited on 10/24/2019).

[50] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector", in *European conference on computer vision*, Springer, 2016, pp. 21–37.

[51] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks", in *Advances in neural information processing systems*, 2015, pp. 91–99.

[52] M. Hollemans, [Online; accessed October 26, 2019]. [Online]. Available: https://machinethink.net/blog/object-detection/.

[53] H. Gao, *Understand single shot multibox detector (ssd) and implement it in pytorch*, [Online; accessed October 20, 2019], Jul. 2018. [Online]. Available: https://medium.com/@smallfishbigsea/understand-ssd-and-implement-your-own-caa3232cd6ad.

[54] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[55] M. Hollemans, [Online; accessed October 22, 2019]. [Online]. Available: https://machinethink.net/blog/mobilenet-v2/.

[56] ——, [Online; accessed October 23, 2019]. [Online]. Available: https://machinethink.net/blog/googles-mobile-net-architecture-on-iphone/.

[57] [Online; accessed July 20, 2019]. [Online]. Available: https://www.raspberrypi.org/forums/viewtopic.php?t=58033.

[58] [Online; accessed July 20, 2019]. [Online]. Available: https://www.raspberrypi.org/forums/viewtopic.php?p=689525#p689525.

[59] E. F. Schubert, J. Cho, and J. K. Kim, "Light-emitting diodes", *Kirk-Othmer Encyclopedia of Chemical Technology*, pp. 1–20, 2000.