Delft University of Technology
Master's Thesis in Embedded Systems

Sensing human activity with dark light

Hajo Kleingeld

Sensing human activity with dark light

Master's Thesis in Embedded Systems

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

Hajo Kleingeld
hajokleingeld@gmail.com

27th November 2017

Author
Hajo Kleingeld (hajokleingeld@gmail.com)
Title
Sensing human activity with dark light
MSc presentation
27th November 2017

Graduation Committee

Prof. dr. K.G. Langendoen (Chair)	Delft University of Technology
dr. M. Zuniga (Daily Supervisor)	Delft University of Technology
dr. C. Doerr	Delft University of Technology

Abstract

TODO ABSTRACT

Preface

TODO MOTIVATION FOR RESEARCH TOPIC

TODO ACKNOWLEDGEMENTS

Hajo Kleingeld

Delft, The Netherlands
27th November 2017

Contents

Chapter 1

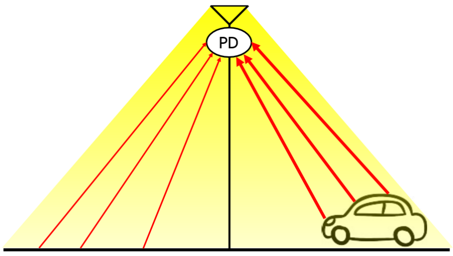
Introduction

Nowadays, 19% of the global energy consumption is used for lighting. For this reason, saving energy in lighting is vital. A simple way to save energy is to simply turn the lights off, or reduce the amount of light used when nobody is around. This thesis proposes a new method for luminaires to detect the presence of humans and objects.

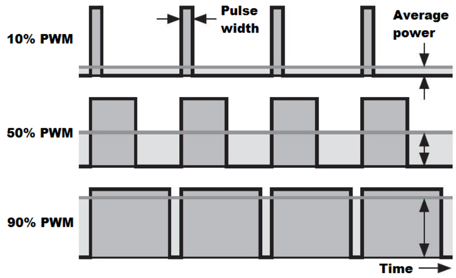
The idea of human sensing is not new. Everybody in the western world has walked into a room where the lights suddenly turned on once they entered. The most common method to create this effect is to make use of a PIR (passive-infrared) sensor. By monitoring the infra-red radiation (heat) in the area, it can detect changes in the environment and toggle the light based on these changes. This method works very well but has several drawbacks. The first is that it's unable to detect objects with the same surface temperature as the environment, like for example a car where the engine has just been turned on. Another drawback is that the PIR method has no potential for communication without the addition of extra components. The new method attempts to overcome these drawbacks by only using a photo diode and the light in the visible spectrum a luminaire normally emits.

This thesis explores the idea of detecting changes in the environment with reflections of visible light. The proposed system works in the following manner: If nothing is in the area, the light will be turned on and the luminaire will illuminate the surrounding area. Some of the light will reflect off the environment back to the light source. This light can be measured with the photo diode. The signal received is a measure of the illuminated area. If something where to change in that area, a car drives by for example, then the reflections in the environment will change and therefore the light perceived by the photo diode will change as well. These changes will then result in a detection by the system. An overview of the scenario can be seen in **??**.

This method by itself does not save energy as the light used in the system is always turned on. If we are able reduce the light output while maintaining the ability to pick up meaningful reflections, then the system would save energy. For this application we cant simply lower the light output by reducing the current flowing into the light (analog dimming). If we for example reduce the current flowing into the light by a factor 2, then the reflections will also become twice as weak, which results in a smaller signal to detect. Another method to reduce the total light output is to decrease the amount of time the current is allowed to flow into the light (digital dimming). By turning the light on and off at a rapidly we can reduce the light output. Then, if we then measure the reflections only when the light is fully turned on, we can capture a reflection without losing signal strength.



(a) raw data received by WIFI reciever-transmitter pairs.



(b) Moving variance applied to the raw data.

Figure 1.1: Analogue VS digital dimming.

1.1 Problem statement

Is it possible to create a system, that can detect the activity of humans or objects by measuring reflections of visible light while being invisible to the human eye?

- How strong is a reflection obtained from a flash in a realistic scenario and how much does this reflection vary if a human is in the area?
- What are the challenges in obtaining reflections when the light is turned on for a very short time and how can they be tackled?
- What additional signals are received by the system (beside the reflection of the flash) and what algorithm can be used to convert the received signal in a reliable logical signal: Detection or no detection?

1.2 Contributions

This thesis proposes **Dark Sensing**, a system that uses reflections of a LED controlled with a low duty cycle (1.4%), and therefore nearly invisible to the human eye, to detect changes in the surrounding area without active involvement of the environment.

- A model, estimating the change in signal (reflected light) when a object moves under, leaves or passes by the LED in different environments.
- A method to convert a captured reflection of the LED into a usable measure of the environment.
- An algorithm which analyses features of consecutive flashes which is capable of detecting objects moving under, leaving or passing through the illuminated area.
- A prototype capable of detecting 99% of all humans passing by in a realistic environment and therefore saving 98.6% of the light used in comparison to the original situation.

1.3 Organization

The thesis starts with providing background knowledge required to understand the thesis followed by an overview of the related work. It then shows that the proposed system could work with the help of a model.

Chapter 2

Background

This chapter first presents the background knowledge required to understand the thesis. The first sections describes the characteristics of lights in general. The second section explains how a model can be made which describes the reflection of light. The third section explains how a light can be dimmed, and how this affects the total amount of light outputted, the amount of light sensed by electronics and the human experience.

2.1 Characteristics of a lights

Before reading this thesis, a basic understanding of photometry is required. For this reason, this section will first introduce the the units and measures used in this document, followed by the most used method for modelling and calculating these measures.

2.1.1 Units and measures of light

Name	Symbol	unit	Description
Radiant flux	Φ_e	W	Radiant energy per unit time
Luminous flux	Φ_v	lm	Luminous energy per unit time
Luminous intensity	I_v	$lm/sr(= cd)$	Wavelength-weighted power emitted by a light sour
Illuminance	E_v	$lm/m^2(= lx)$	Amount of luminous flux impinging a surface
Luminous energy	Q_v	$lm * s$	Total amount of luminous flux outputted over time
Luminous exposure	H_v	$lx * s$	Total amount of illuminance impinging on a surfac

Table 2.1: summary of measures, units and symbols used in this thesis

2.1.2 Modelling a light

Light sources in optics are typically modelled as a point in spaces, emitting light in a Lambertian radiation pattern [x]. This pattern describes how

Name	Symbol	unit	Description
Light uniformity	U_0	-	E_{min}/E_{mean}
Reflection coefficient	ρ	-	wavelength dependent surface reflection ratio
Albedo	α	-	Impinging intensity / reflected intensity
Exit angle	ϕ	rad	Exit angle with respect to the normal of the reflective surfa
Incidence angle	θ	rad	Incidence angle with respect to the normal of the reflective

Table 2.2: summary of measures, units and symbols used for modeling light

Name	Symbol	unit	Description
Standard deviation	σ	-	Measure of variance over a set of values
Mean	μ or $\overline{X}$	-	average of a set of values
n	n	-	Number of samples in the N section of the algorithm
d	d	-	Number of samples in the D section of the algorithm
m	m	-	Number of samples in the M section of the algorithm
Threshold value	T	-	Value determining the detection threshold
Scale value	ss	-	Value changing n based on the noise in the current signal

Table 2.3: summary of measures, units and symbols used in the algorithm section

much light leaves a light-source at angle ϕ and can be calculated with:

$$I(\phi) = \Phi_{lum} \frac{m+1}{2\pi} cos^m(\phi) \tag{2.1}$$

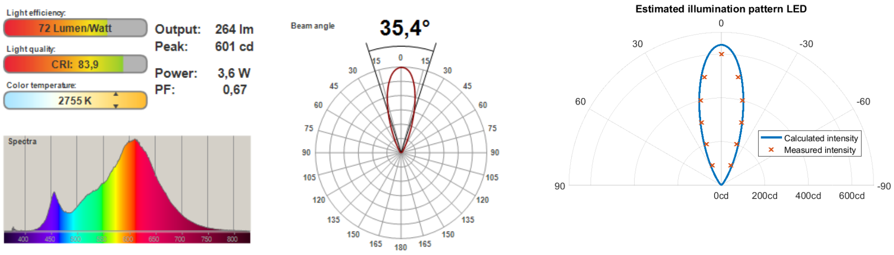
where Φ_{lum} is the luminous flux of the light and m is the order of lambertian emission calculated with $m = -1/log_2(cos\varphi_{1/2})$ where $\varphi_{1/2}$ is the angle where light is leaving the luminaire at half power.

With equation 2.1 we can now estimate the illumination pattern of any LED when the luminous flux and half power angle are known. This can be done for example with the LED in figure 2.1(a). If we choose $\Phi_{lum} = 264lm$ and $\varphi_{1/2} = 17.7°$ then we obtain the pattern shown in figure 2.1(b) which closely matches the measured irradiation pattern.

$$E_{hor} = \frac{I(\phi)\cos(\theta)}{d^2} \rightarrow E_{hor}(x,y,z) = \frac{I(\phi(x,y,z))\cos(\theta(x,y,z))}{x^2+y^2+z^2} \tag{2.2}$$

2.1.3 Modelling a reflection

Now the illumination pattern of a light bulb is known, we are able to calculate how much a light is illuminating a surface with equation 2.2. Some of the light will reflect back in the environment while the rest of the light is absorbed by the material and turned another form of energy (typically heat). The total amount of light reflecting back in the environment can be



(a) Specs of an LED

(b) Caclculated intensity pattern

Figure 2.1: Figure a shows measured specifications of an LED [11] where Figure b shows the estimated illumination pattern of the same LED.

calculated with the surface reflection coefficient $p(\lambda)$. p has a different value for each wavelength λ as not every material reflects the same colour of light. An example reflection coefficient can be seen in figure X.

$$R_{total} = E_h or * p(\lambda) \tag{2.3}$$

The next step is to determine the directions of the reflection. How much light will be reflected in what directions? There are three ways light can be distributed when reflecting of a surface: Specular, spread and diffuse. A visual representation of each of these reflection patterns is shown in Figure 2.2. Each pattern will be discussed briefly.

The **specular pattern**

The **diffuse pattern** is the

The **spread pattern** is

is, it can be approximated by setting m to infinite from eq x.

Some materials have both emit a lambertian pattern and spread pattern. these can be combined with ratio r_d (amount of diffuse reflection), as seen in equation x. This eqation can now be used to model any reflection R, as long as the

$$I = P_I \rho(\lambda) \left[r_d \frac{1}{\pi} \cos(\phi_2) + (1 - r_d) \frac{m+1}{2\pi} \cos^m(\phi_2' - \phi_2) \right] \tag{2.4}$$

$$P_{PD} = \frac{P_I \cos(\theta)}{d^2} rec\left(\frac{\theta}{FOV}\right) \tag{2.5}$$

$$E_{PD} = E_{hor} * p(\lambda) * R(r_d, \phi_2, \phi_2') * \frac{cos(\theta_2)}{d_2^2} \tag{2.6}$$

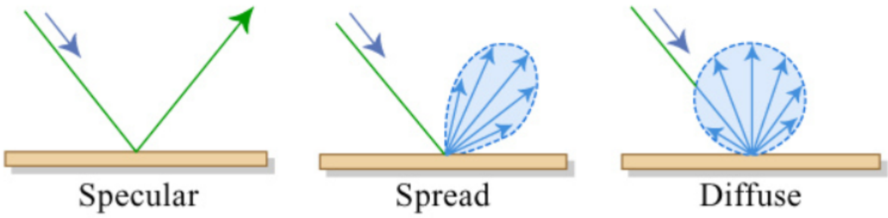


Figure 2.2: The possible ways for light to reflect when it hits a surface [1]

2.2 Dimming and its consequences

Explain what dimming of light is and means

2.2.1 Types of dimming

Explain analog dimming (intensity)

Explain digital dimming (pwm)

2.2.2 Limits of dimming

Explain the limits analog dimming:
- At some point there is not enough current to turn on the lights.
- Reduces range
Explain the limits of pwm dimming
- There is a time required for the LED to turn on
- There is a time required for the let to turn off
- Making the total on time **a bit** too short results in a huge variance in light emitted
- Making the total on time **a lot** too short results in no light

Note that this does not reduce range
LedResponse.png

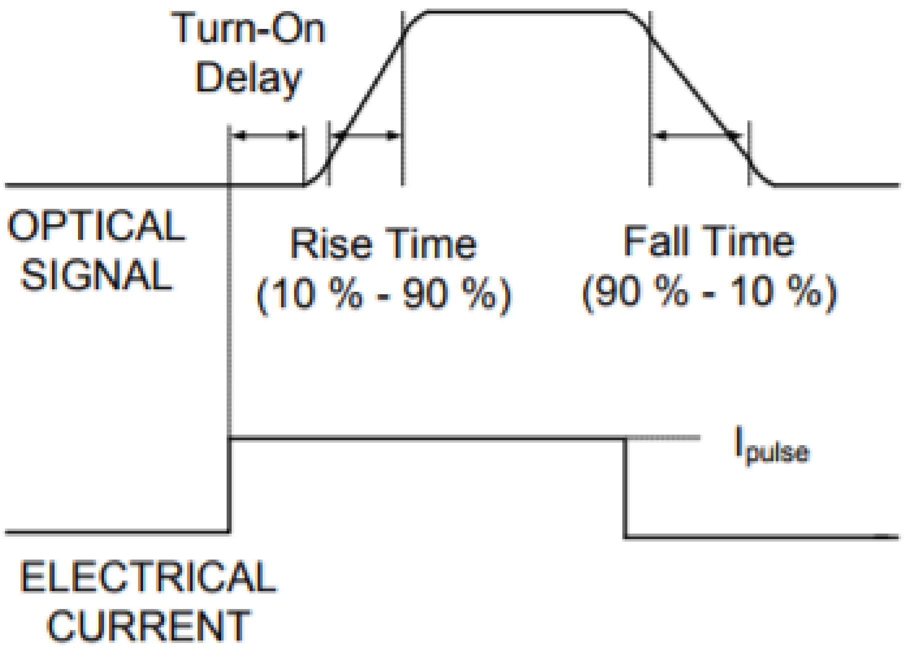


Figure 2.3: Realistic light response.

Chapter 3

Related Work

This section discusses the most important work found related to the thesis. It starts with mentioning similar techniques with the same goal: Detecting human activity. It then summarises all projects done which use light reach the same goal. This chapter finalises with used in other projects using visible light which can be applied in this thesis.

3.1 Related techniques

Passive localization, the act of sensing the presence of humans or objects which do not actively participate in the process, is a common problem and has been tackled in many different ways by companies and research groups. Several techniques used by these groups will be mentioned here while pointing out what specific ideas could be used to improve the thesis.

M. Youssef *et al.* created a detect and track application with the help of WIFI access-points (APs) and WIFI monitoring-points (MPs) and an aplication sever (AS)[17]. The MPs measure the signal strength of the APs, and transmit this data to the AS. The server runs a moving variance algorithm on all of the received signals to detect significant changes in the signal. An overview of the complete system can be seen in figure **??**.

M. Valtonen *et al* created a system which passively tracks humans with the help of capacitive street tiles[15]. Byunghun Song *et al*[18] Smart street light system [12]

3.2 Human sensing with visible light

This thesis attempts to solve the passive localization problem with only visible light a luminaire would normally emit and a single light sensor mounted on the ceiling. Several other projects have attempted similar challenges using visible light. E.D. Lascio *et al.* has created a system which uses a ceiling mounted luminaire and light sensors in the floor as seen in figure **??** [5]. A

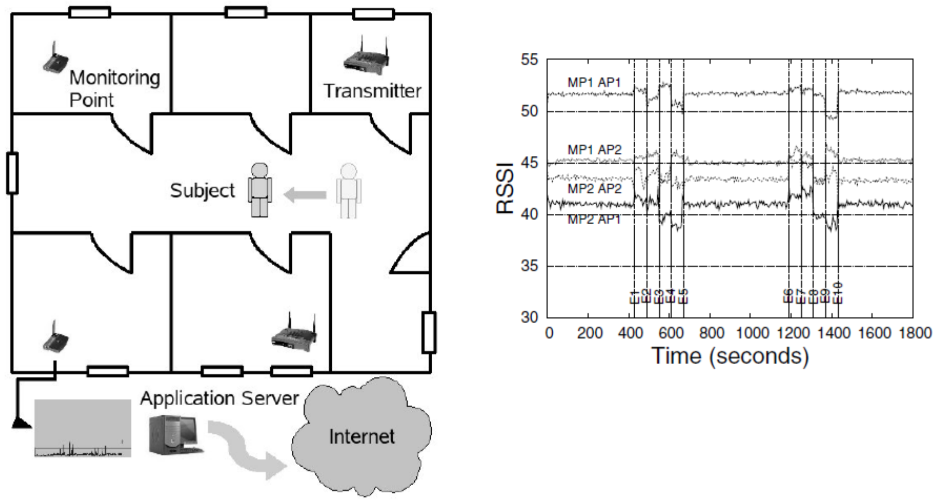


Figure 3.1: Overview of the WIFI tracking system of Moustsafa Youssef *et al.* [17]. The left figure shows an overview of the setup where the right figure shows the strength of the APs from the point of view of the MPs. E1 to E8 represent possible 'events' of bypassing persons.

Figure 3.2: Overview of the LocalLight system of E.D. Lascio *et al.* Lights on the ceiling and light sensing RFID tags on the floor.

human passing by in interrupt the light rays and cast a shadow on the photo diode, resulting in a detection.

T. Li *et al.* takes the concept of lights on the ceiling and photo diodes on the floor to the next level in [19]. By placing multiple lights on the ceiling and photo diodes on the floor, they create a pixelated image of a person standing from the point of view of each light on the ceiling (see figure **??**). These pixel-images are then used to reconstruct the original stance of the person scanned.

C. Zhang *et al* [3]

J. Zhang for example created a method for localizing and tracking objects with specular surfaces on a line[20].

Another example of indoor activity detection is the project of M. Ibrahim *et al.* They detect humans or bypassing objects with the help of multiple

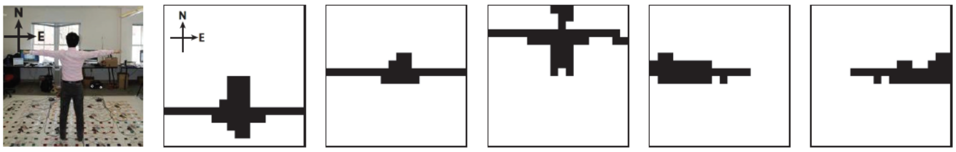


Figure 3.3: The scanned person with the resulting created pixel images.

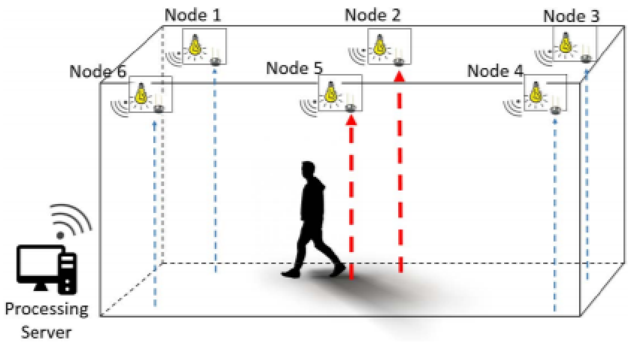


Figure 3.4: Overview of the system set-up of "Activity sensing using ceiling photosensors" project [16]. In this specific situation node 2 and 5 detect shadow caused by the person in the room, while other nodes do not.

lights and photo diodes hung at a ceiling[16]. Normally, if nobody is in the room, all lights shine on the floor, and reflect some rays of light back to each photo diode hanging on the ceiling. If an object passes by which interrupts this light ray, then a shadow is created on the floor. The photo-diode hanging at another light will notice the reduced reflection and a detection is triggered. An overview of this set-up can be seen in figure **??**.

3.3 Related visible light techniques

A lot of projects make use of visible light, but in another way

There is one other project in the VLC world that has nothing to do with human sensing. Zhao Tian *et al* explores the idea of VLC with dark light, a VLC primitive that allows light-based communication to be sustained even when LEDs emit extremely-low luminance [21] [22]. The communication works by generating high power, but short light pulses (500ns). These pulses are then used in a pulse position modulation scheme to achieve communication (1.8Kbps at 1.3m) with light while being nearly invisible to the end user.

[19]

Chapter 4

Model

A model has been made with two goal of answering two questions:

1. How strong are the reflections of flashes in a realistic environment?
2. How much will these reflections change if an object enters the illuminated area

This section describes how the model explained in section 2.1 was adjusted and implemented to answer the questions posed above

4.1 Model description

The model made is an interpretation of the phong reflection model (see section 2.1). It calculates how much of the light leaving a luminaire, bounces back via the environment to a photodiode placed next to the light source. This section will first discuss the adjustments made to the phong model, followed by an explanation of the calculation process.

4.1.1 Adjustments

The first adjustment is the removal of "time". The methods in the literature took the travelling time of light into account in order to calculate the possible inter-symbol interference. This is not required for this simulation as we are only interested in the steady state situation when the light is fully turned on and the light received by the photodiode is maximized for the current situation.

The second adjustment is the removal of "colour". The original method differentiated between different wavelengths of visible light when reflecting light of off surfaces and was therefore maintaining colour information. This is however not necessary for this model, as we do not care about the colour of the reflecting objects, but only about the total amount of energy reflected by the object. For this reason, the surface reflection coefficient ($p(\lambda)$) was replaced with the albedo of the object instead.

Albedo is a property of an object representing the percentage of energy which is reflected when sun is shining on the object. Even though albedo is based on the full spectrum of sunlight instead of only the wavelengths of visible light, it gives a reasonable approximation of the reflection coefficient in this scenario. This will be shown in section 4.2.

$$\Gamma = \int_{380nm}^{780nm} \Phi_e p(\lambda) d\lambda \rightarrow \Gamma = \Phi_{lum} \alpha \tag{4.1}$$

The final adjustment is the amount of reflections we calculate. In reality a light ray can be reflected an infinite amount of times of several different surfaces before returning back to the sensor. In the model however we only calculated one bounce (from the light to an object and back) for two reasons. The reason for this is that the first reflection provides approximately 80% of the signal where all other reflections only make up 20% of the total power[13]. This increase in accuracy

$$E_{hor} = \frac{I(\phi)\cos(\varphi)}{d^2} = \frac{I(0)cos^m(\phi)\cos(\varphi)}{d^2} \tag{4.2}$$

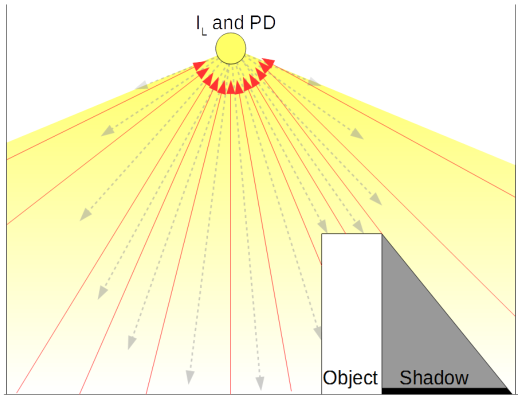
4.1.2 Calculation process

Calculating the amount of light reflecting back to the object is a three step process. The first step is to calculate the shadow casted by the object on the floor and walls. This is required as the surface where the shadow is casted can't reflect light back directly to the photo diode. It's important to note that light casting the shadow is reflected of the object instead and with that, changes the reflection pattern of the room.

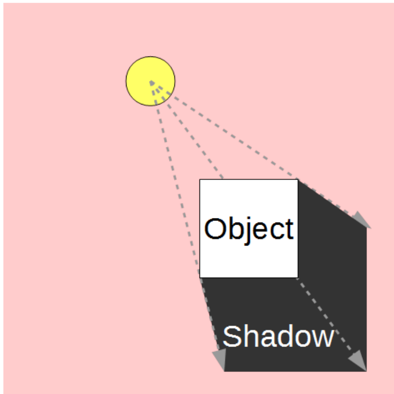
The second step is to calculate how much light reflected from all floors and walls (where no shadow is casted) to the photo diode. The finals step is to calculate how much light is reflected from each side of the object. **??** shows an overview of an environment with rays leaving the light, casting shadow and the resulting reflections.

4.2 Verification

The calculation method was verified using a scale model featuring a LED[11], a paper box and a light meter[14]. The first step of verifying the model is check if the LED is modelled properly. This was done by hanging the LED at 100cm above the floor and measuring the horizontal illuminance (E_hor) at the floor to see if the measured irradiation pattern of the LED matches the theoretical pattern produced by equation 2.1. Simulations in Appendix



(a) Sideview



(b) Topview

Figure 4.1: Overview of the calculation process. Grey lines represent light rays casted by the light. Black represents the shadow casted by the object on the floor or walls. Red lines or areas show reflections bouncing from the ground, walls or object back to the photo diode.

A that the LED in the test set-up was producing more light than in the specification.

The second step is the verification of the interpretation of the Phong model. This was done with the test set-up shown in figure 4.2. By moving a paper box across the flool in steps of 5cm and measuring the reflection at each step we obtain the red line in figure 4.3(b).

4.3 Modelling of the hallway

The hallway modelled is based on a real hallway located at the TU Delft. The hallway is 2.2m wide and 2.8m high. The floors albedo is set at 0.37, as this was calculated during the verification of the model. The albedo of the walls was set to 0.95 which represents the albedo of white plaster[7]. The reflection of these surfaces is assumed to be fully diffuse ($r_d = 1$).

Industry standards state that corridors in education buildings should be illuminated with at least $E_{mean} > 100lx$ and a light uniformity of $U_o > 0.4$[23]. These lighting requirements can be achieved using the same luminaire used during the verification process if hung in the staggered formation shown in figure 4.4(a). Calculations showing that the industry standards are met can be found in Appendix A.

The object passing by the light (representing a human) will be modelled as a cuboid 0.5m long and 0.2m wide with varying heights. Several albedos have been assigned to the cuboid to represent the different kind of clothing humans wear. The object will be moved in a straight line trough the hallway

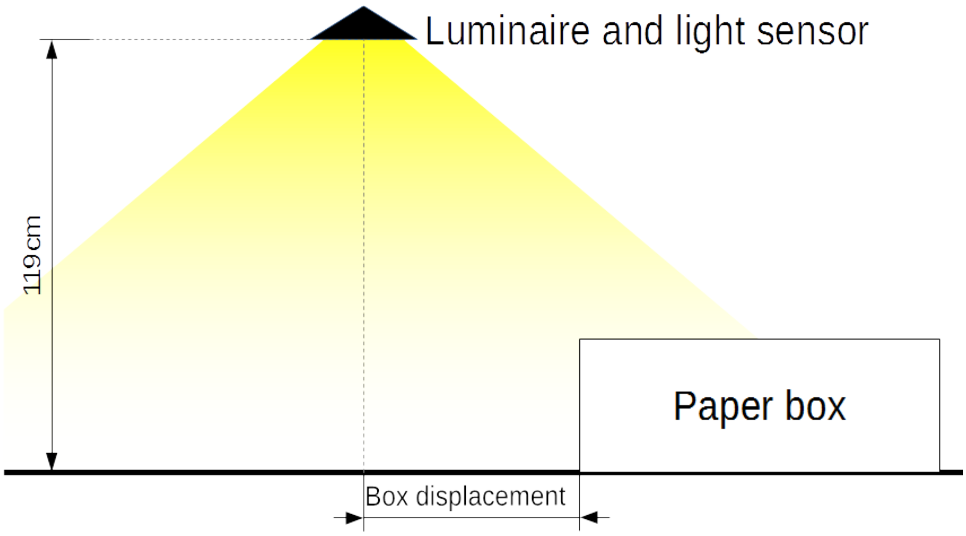
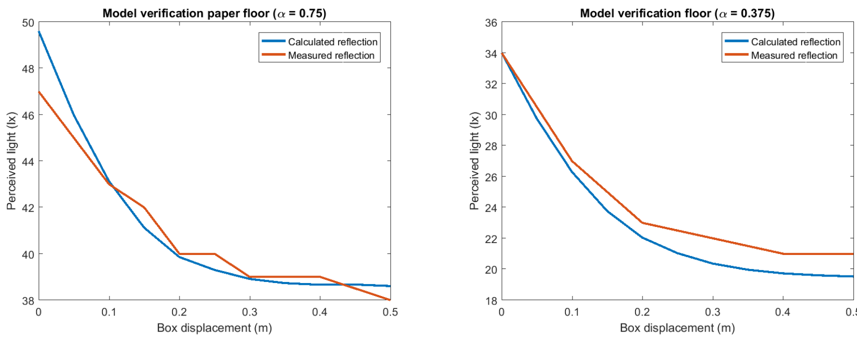
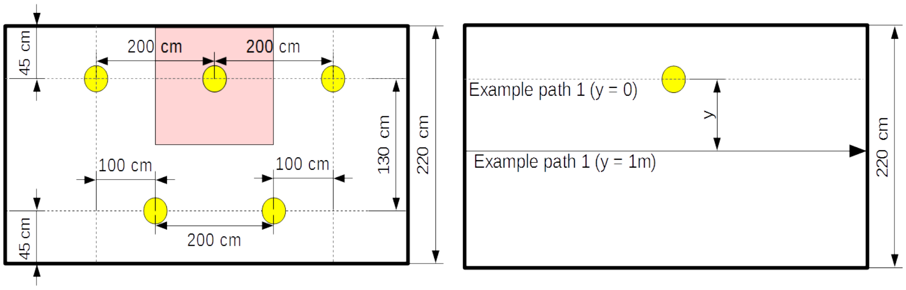


Figure 4.2: Visualisation of the model verification set-up.



(a) (b)

Figure 4.3: Both figures show that the model provides a reasonable approximation of the reality. Note that the albedo of paper was taken from [7] and the albedo of the floor was estimated with these measurements.



(a) Staggerd hallway LED pattern (b) Traveling path of the object

Figure 4.4: Figure a shows the position of the luminairs to obtain a realistic illumination pattern. Figure b shows an example travelling path of an object.

with the light at a set vertical distance y. Some example paths can be seen in figure 4.4(b).

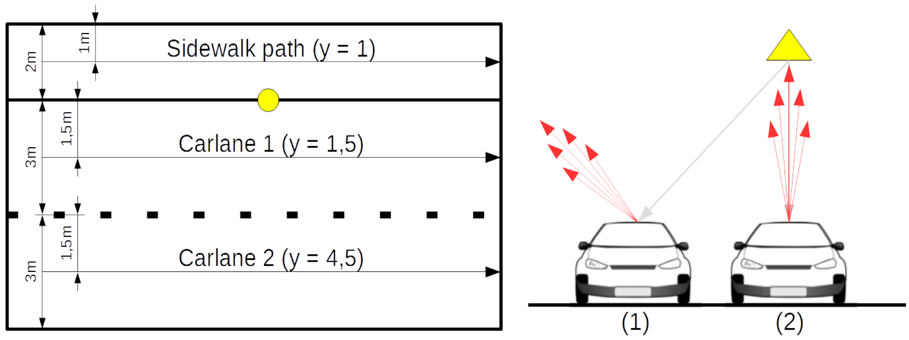
4.4 Modelling of the street

The street model is based on a real street near the TU delft. It has two lanes for a cars (each 3m wide), and sidewalk (2m wide). The albedo of the street will be modeled with different values for old ($\alpha = 0.06$) and new asphalt ($\alpha = 0.14$), as asphalt loses reflectivity if it grows older[7]. The reflections of the street are assumed to be fully diffuse ($r_d = 1$).

Industry standards state that a street with side walk should be illuminated with at least $E_{mean} > 3lx$ and a light uniformity of $U_o > 0.2$ [4].These lighting requirements can be achieved using 700lx luminaires with a halfpower angle of 60° placed every 15 meter in between the road and side walk. This set-up is visualized in figure 4.5(a). Calculations showing that the industry standards are met can be found in Appendix A.

In this model two different objects will be modelled representing humans (walking on the side walk) and cars (driving in the two driving lanes). The humans will be modelled in the same way as in section 4.3. The car will be modelled as a cuboid with the dimensions of an Opel Corsa (4m x 1,7m x 1,5m), a commonly seen small car. The object was modelled with diffuse reflection, because no reliable sources describing the specular parameters (r_d and m) of cars could be found.

Lacking the specular deflection for this specific model should not influence the results significantly. This is because no part of the car will be moved directly underneath the light and therefore no significant amount of light of the spread reflection should ever reach the light sensor. This is visualized in figure 4.5(b). This is also the reason why cars are simulated with lower albedo than humans.



(a) Topview of the street model

(b) Spread reflection on cars

Figure 4.5: A shows an overview of the model. B shows why the spread reflection component plays no part in this model for cars.

4.5 Results

Several measurements have been graphed in Figure X. Raw results can be viewed in appendix B.

$$\frac{s}{v_{object}} = T \qquad \frac{1}{f} = T \tag{4.3}$$

$$f_{WorstCase} = \frac{v}{s} = \frac{10/3.6}{m} = 0.92Hz \tag{4.4}$$

4.6 Conclusions

Influence of albedo

approximate the minimum and maximum signal frequency.

Chapter 5

Platform

A device has been made to generate, receive and analyse flashes. The complete system architecture can be found in figure 5.1. Each component and their interfaces will be discussed briefly, followed by a section showing the final build of the platform.

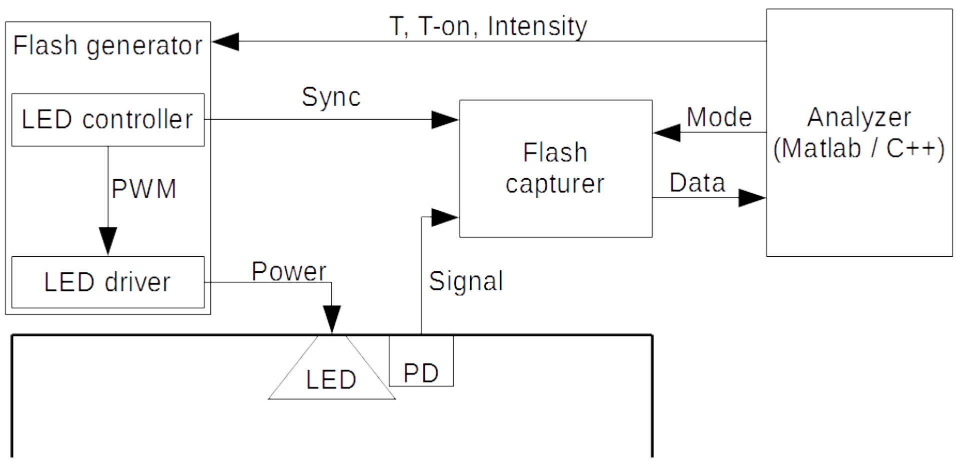


Figure 5.1: System architecture of the flash generator/analyser

5.1 system components

5.1.1 Flash generator

The flash generator is a device able to control a LED with high precision. It is able to set the period T, and the t-on time T_{on}. T Controls the frequency of the flashes and t_{on} length. Both parameters can be set with a resolution of $10\mu s$ resulting in a precisely controlled PWM signal with the help of equation 5.1. This signal is sent to a LED driver trough one of three LED drivers, which will make the actual light turn on and off at different light levels.

Besides generating the PWM signal for the light, the flash generator has another function. It sends a sync signal to the flash receiver just before generating a flash. This allows the Flash generator to be ready when the flash starts, so it does not waste time sampling if no flash is generated.

$$T = \frac{1}{f} \qquad DutyCycle = \frac{T_{on}}{T} * 100\% \tag{5.1}$$

5.1.2 Reflection receiver

The job of the receiver is to sample values while the light is being turned on and off, to then analyse the full reflected flash and extract a feature which properly represents the environment. The receiver therefore should capture flashes as precise and consistent as possible. For this reason, the receiver receives a sync signal from the flash generator, so it can start sampling at exactly the same moment every time relative to the start of the flash.

The receiver should continue sampling for a set period of time. once done, the device should do one of the following things with the received samples, depending on the mode of the analyser:

1. Send back the full flash, uncompressed, for the analysis of separate flashes.
2. Send back all compressed flashes, by extracting several features.

5.1.3 Analyser

The analyser will receive samples from the reflection receiver and is ran on a PC in the form of either a C/C++ program (real-time) or as a MATLAB script (post-time). The analyser can set the receiver to raw-. or compressed mode. If the receiver sends raw flashes to the analyser it can be used to analyse this flash. This mode is used in chapter **??** to analyse single flashes to find the ideal settings for the flash generator and reflection receiver. If the receiver sends compressed flashes, the analyser is able to analyse consecutive flashes. This mode will be used in chapter **??** to find an algorithm to determine if an object is moving in the area under the light.

The Analyser should also be able to control the flash generator if the system is running in real-time mode. It is therefore able to send a packet with T, T_on and I_{LED} to the device. This allows for real time control of the flash generator.

5.2 Implementation

The system was build by combining several of shelve parts. An overview of the actual build can be seen in figure 5.2. It shows the different components mounted on a box. This section will explain briefly how each system component is implemented and why each part was chosen.

The flash generator is implemented on an Arduino UNO[2]. This platform was chosen, as it's simple to use, does not require an operating system (OS) and has therefore no unexpected jitter. The LED used in the set-up is the same LED as modelled in chapter4[11]. The power used by the LED is regulated with a single resistor. The resistors where chosen after some experimentation with the flash generation and reception. The values and resulting power consumption of the LED can be seen in equation 5.2.

$$P_{LED} = \frac{(V_{DD} - U_{LED})^2}{R} \qquad P_{LED} = \frac{(7.2 - 3.6)^2}{[1, 3, 5]} = [4W, , 1.33W] \quad (5.2)$$

The reflection receiver is implemented on the shine platform [10]. This platform was chosen because it's a simple (no OS required) hands-on platform featuring multiple photo diodes by default. The original software of shine sampled each photo diode at 1Khz. This is way too low to see the $10\mu s$ flash resolution. For this reason the software of shine was rewritten to sample in bursts of 50 samples at 210Khz ($\pm 240\mu s$) when the sync signal is received.

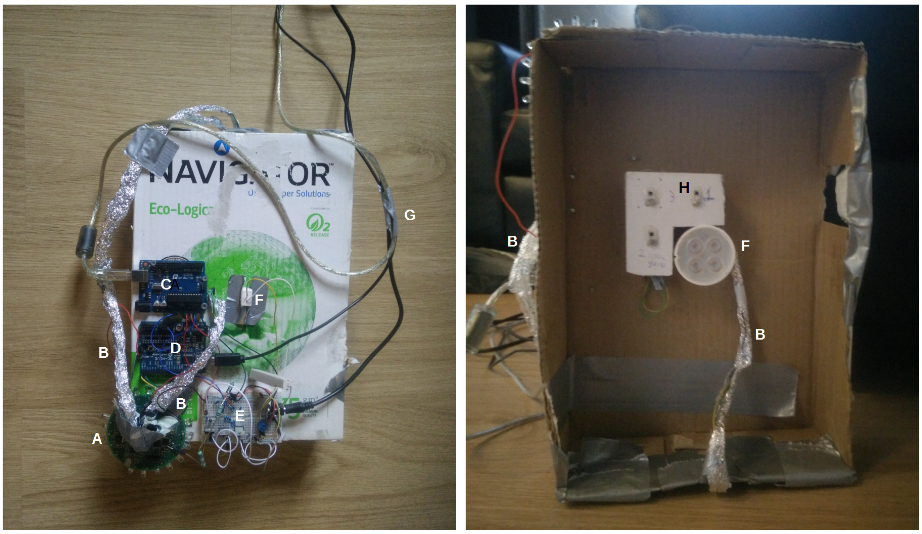
A downside of the shine platform is that it's unable to communicate directly with a computer as it does not has a FTDI interface. This problem was solved by using a processor-less Arduino UNO as bridge between the analyser and shine platform.

The receiver makes use of three photo diodes. The original sensors on shine where replaced with ones more sensitive to visible light. Each sensor is also configured in a different way. Some feature an increased amplification of the measured signal. Others have a longer wire with (intentional) bad shielding which can simulate how the system preforms in a environment with lots of electromagnetic radiation. An overview of the PD configurations can be seen in table **??**.

Another important decision concerns the amplification circuit of the photo diode. The original circuit used of shine features an analogue low-pass filter to remove ripple introduced by the amplifier (See Figure **??**). The filter has several side effects. It reduces the signal strength and increases the time the signal is invisible to the system. It was therefore chosen to remove al analogue filters from shine and deal with the ripple with the help of software if required. The ripple effect might even be useful as the it's probably dependent on the received signal and therefore a measure of light the environment.

PD#	Wire length	Gain	EMC Shield
1	Long	1000	Slecteable
2	Short	5600	Yes
3	Short	10000	Yes

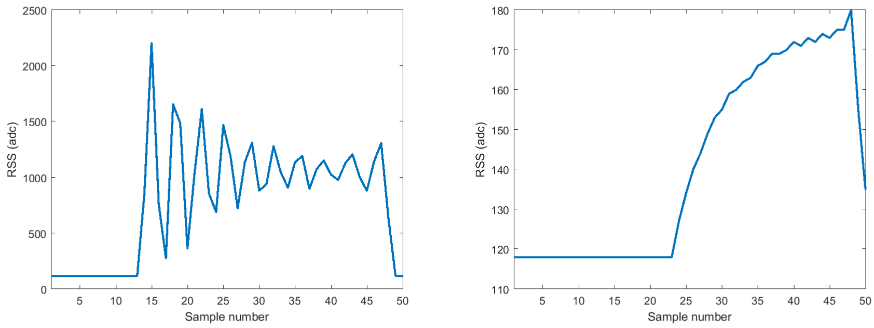
Table 5.1: Overview of photo diode configurations



(a) Top view (b) Bottom view

Figure 5.2: The platform prototype. Each letter denotes a different component:

A = Reflection receiver
B = Wires to the photo diodes
C = LED controller
D = Communication bridge between shine and the PC
E = LED driver
F = The LED
G = Wires to the analyser and power supply
H = Three photo diodes



(a) Unfiltered flash (b) Filtered flash

Figure 5.3: Two flashes captured with the platform. Left is unfiltered, right is filtered

5.3 Evaluation

The system has been build and tested. Even though the created device has a poor build quality, it has great potential for experimentation with the proposed method of activity detection. The main advantages are:

- Each building block has one clear purpose and can therefore be tackled separately from other components. It's therefore impossible that a timing error in the flash generator software affects the sampling of the receiver or vice versa.
- The build quality is poor. If the project works on this device, it will definitely work on a dedicated platform.

The next steps for the project is finding a method for extracting useful information from flashes as shown in figure 5.3(a).

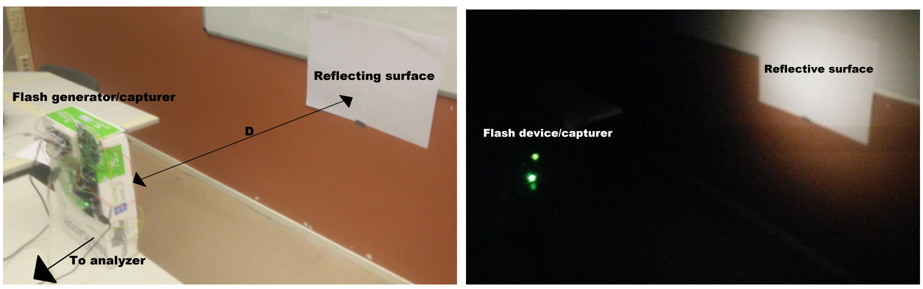
Chapter 6

Flash Analysis

The goal of this chapter is to find a method, capable of obtaining consistent measures of the environment from measured flashes. Additional goals are to achieve this with the shortest flash and with the least complex method as possible.

In this chapter, the platform is set-up as seen in figure ??. D in the figure represents the distance between the device and the reflecting surface (the floor in this case). All measurements presented in this chapter have been made in a darkroom, a room where no lights from outside can enter, so the test result won't get influenced by other illumination sources.

The set-up will first be used to get a reasonable understanding of what flashes look like and how settings of the flash generator influence the received flash. Then, several methods of obtaining these information flashes will be presented and compared. This chapter will conclude with a final settings used in the flash generator and an algorithm to obtain a consistent measure of the environment from the received flash.



(a) Test setup in illuminated environment (b) Test set-up in dark environment

Figure 6.1: Test set-up used to capture flashes in the darkroom.

6.1 Flash properties

The test set-up has several parameters which can affect the perceived flash: T_{on} (on time of the LED), I (brightness of the LED), PD (sensitivity of the photo diode) and D (distance between device and reflecting surface). This section shows how each of these parameters influences the received signal. Note that the period, T, is not present in the list as should not influence an individual flash, but only the total amount of flashes.

Figure 6.2(a) shows several responses for different T_{on}. In the figure it can be seen that all signals closely match each other, until the light is turned off. This is a useful property as this means it's possible to reduce the T_{on} with no influence on the signal, if the last part is not used.

Figure 6.2(b) shows the influence of using the different amplification circuits of the flash generator. It can be seen that the LED powered with the lower resistance (and thus a higher LED current) is perceived as brighter to the system than the lights powered with a bigger resistor. It's also observed that the LED powered with higher currents show up earlier to the system. This is because LEDs driven with higher currents turn on faster [8]. This means that if a lower LED current is used a bigger T_on is required to obtain useful information.

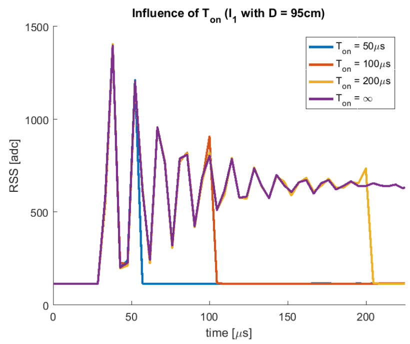
Figure 6.2(c) shows a set of measured flashes at a variance distance from the wall. It clearly shows that if the distance increases, the observed light also decreases. This is logical, as when light travels longer distances, the relative intensity of the light decreases.

Figure 6.2(c) displays what happens to the signal if it's sampled by different photo diodes (with a different amplification circuit). As would be expected, the signal with the bigger amplification perceives the signal to be several times stronger. It's however important to note that the frequency ripple on the signal is significantly lower with the strong amplifier. This makes sense as using a higher resistance in the amplification circuit (resulting in a higher gain) increases the osculation frequency of the amplifier.

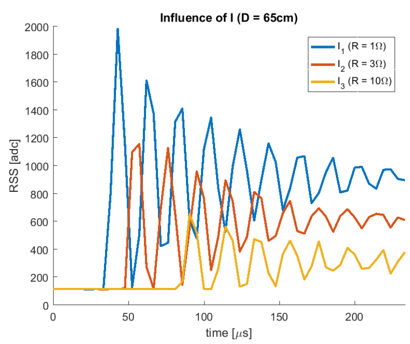
Figure 6.2(d) shows what happens when the different photo diodes are used. As expected, the RSS rises once we increase the gain on the photo diode. PD_3 almost instantly saturates as the gain is too strong when used in combination with I_1. PD_3 is therefore also displayed with I_3. Another thing which changes is the ripple frequency. This is expected, as the resistor in the feedback loop of the amplifier was changed.

6.2 Flash features

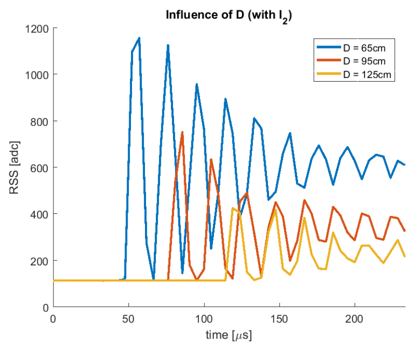
This section explores what kind of features can be extracted from a flash signal. It will then compare the methods based on required T_{on}, precision, the "Signal to Noise Ratio" (SNR) and computational complexity.



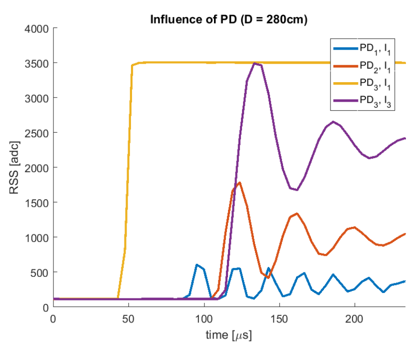
(a)



(b)



(c)



(d)

Figure 6.2: Several perceived flashes generated with different settings of T_{on}, I_{LED}, D and PD.

6.2.1 Feature considerations

The maximum of a flash response could contain useful information. Even though the light at the first maximum has not fully turned on yet, it still is some measure of the perceived light. This can be especially useful if the maximum of the flash always occurs at the exact same moment in time relative to the light turning on. If that is the case, then the maximum value of the first peak could provide us with enough information of the environment. If the maximum value of the first peak holds enough information, then a very small T_{on} can be used to obtain this value, as decreasing T_{on} does not significantly influence the height and form of the first peak.

Another possibility is the to remove the oscillation of the signal with a low pass filter and then take the maximum value of the filtered signal. This method less reliant on precise timing of the pulse. It also uses more samples of the signal and should therefore be able to obtain a value which better represents the reflections of the current environment than the maximum method. A downside to this method is that a filter designed to deal with one frequency of ripple, is not immediately suited to deal with the other possible ripple frequency.

Another method considered is to use the surface underneath the signal. This method has the advantage of being both simple and flexible. It does not matter if T_{on} is chosen big or small. It also does not care about the ripple frequency of the amplifier This method simply sums all information available to obtain a measure of the reflections.

The final possibility considered is the filtered sum method. It first uses a filter to smooth the signal to then calculated the surface underneath it. It also requires multiple filters to be designed (one for each PD amplifier). It might however give a more detailed result than the filter method, as more information is used obtaining the data point.

6.2.2 Feature comparison

A test was created to compare the effectiveness of each feature with various settings in a full scale environment ($D = 280cm$). The test was executed as follows:

1. Set the parameters for the given test (PD, I, T_on).
2. Move a highly reflective piece of cloth underneath the set-up at $185cm$ ($D = 95cm$).
3. Move the piece of cloth underneath the setup again, but no from the other direction.
4. Calculate the SNR of the received signal.

If we refere to the 'SNR' in this thesis we mean the SNR as defined in equation 6.2. This equation calculates ratio between the standard deviation

	SNR: PD_2, I_1			SNR: PD_3, I_3		
T_{on}	$150\mu s$	$200\mu s$	$250\mu s$	$150\mu s$	$20\mu s$	$25\mu s$
Maximum	35	38	40	5	5	5
Filtered maximum	39	65	66	20	27	33
Sum	45	75	95	18	20	26
Filtered sum	50	105	100	19	20	24

Table 6.1: Overview of the test results

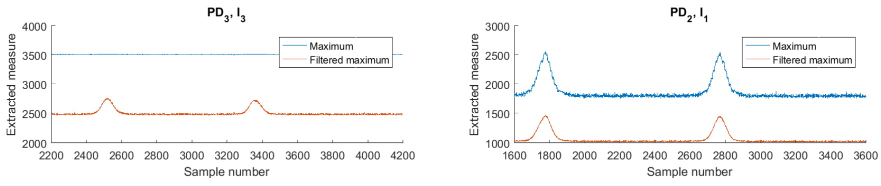


Figure 6.3: Data extracted using the maximum and filtered maximum methods with $T_{on} = 250\mu s$.

of the signal when noting is passing by and the absolute minimum and maximum of when something is. The higher the SNR, the easier it should be to distinguish between activity and no activity later on.

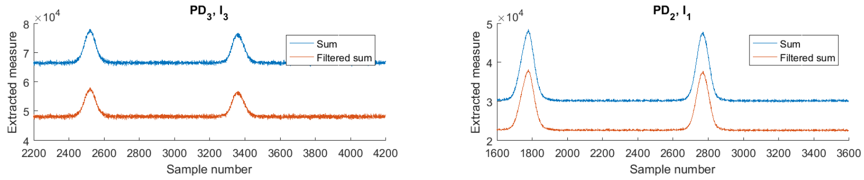
$$SNR(PD) = \left(\frac{\mu(PD_{NoEvent}) - min(PD_{event})}{\sigma(PD_{NoEvent})} + \frac{max(PD_{event}) - \mu(PD_{NoEvent})}{\sigma(PD_{NoEvent})} \right) \tag{6.1}$$

$$SNR(PD) = \frac{max(PD_{Event}) - min(PD_{Event})}{\sigma(PD_{NoEvent})} \tag{6.2}$$

The test was done with all combinations of PD and I. Only the combinations of PD_2, I_1 and $PD3, I_3$ gave potential usable results at full scale as for other combinations the flash was invisible or or too bright (saturation). Several consecutive captured features can be seen in the Figures **??** and **??**. These where then used to calculated the SNR for each scenario.

An overview of all calculated SNR values can be seen in table **??**.

The final parameter to decide is the period of the signal, T. This value has no influence on the noise measured on the signal except for the frequency. It has however a clear influence on how much light is used by the system, as decreasing T directly increases the amount of flashes. We can't choose a too low value for T as then users will observe flickering of the light. Another reason T can't be chosen too low is that certain kind of noise still needs to be filtered out of the system. It is almost guaranteed that some 50Hz component will be seen in the signal, as long as its connected to the net.



(a) (b)

Figure 6.4: Data extracted using the sum and filtered sum method with $T_{on} = 200\mu s$.

	PD_2, I_1			PD_3, I_3			# of	# of
Method	$T_{on}[\mu s]$	Noise (σ)	SNR	$T_{on}[\mu s]$	Noise (σ)	SNR	Additions	Multiplications
Maximum	15	1.9	5	20	18.8	40	N	-
Filtered maximum	25	8.4	33	25	6.5	65	N + 4N	5N
Sum	25	450	26	25	192	95	N	-
Filtered sum	25	424	23	20	142	100	N + 4N	5N

Table 6.2: Overview of the best found settings to extract each feature.

For these reasons, T was chosen to be $800\mu s$. This value results in a flash frequency of 125Hz. This value is more that the Nyquist frequency of the 50Hz. Even though literature recommends at least 200Hz to prevent the visibility of flickering, none was observed by 10 different test subjects with this setting of T. Another benefit is that the found method does not require a lot of computational power.

6.3 Conclusion

The Flash analyser will run at a frequency of 125Hz, a T_on of $200\mu s$ with maximum light intensity I_1. These settings provide a reasonable level of precision for the full scale scenario. The next step for the project is creating an algorithm for the analyser, capable of analysing a set of consecutive flashes.

Chapter 7

Analyser

The flash analyser now outputs values at 125Hz, which is a mixture of various lights and noise. The next step is to create a real-time binary classification algorithm to convert the incoming samples into a logical value: Activity detected, or no activity detected. The detection algorithm should be designed with certain goals in mind:

- **High true positive ratio** - The system does not fulfil it's purpose if it is unable to reliably detect bypassing objects.
- **Low false positive ratio** - The system is useless if it classifies everything as activity. This would result in the light being on all the time and therefore, no energy being saved.
- **Fast response time** - If the algorithm manges detect every bypassing person correctly, but it only triggers a detection when the user has already passed the light, then the system does not fulfil its purpose.
- **Low computational complexity** - If the algorithm uses too much calculations per incoming sample, the system would require a strong processor to analyse all incoming data. This makes the system expensive, if it where to eventually get implemented in the real world.

This chapter is separated in three parts. The first part shows what signals are received by the photo diode. The second part explains what methods considered to remove unwanted signals from the signal. The final part of this chapter what considerations where mode to determine threshold of the binary classifier.

7.1 Received signals

In an ideal world, the dark sensing system only perceives light it emits itself, reflected by the environment. The previous chapter already showed that this is not the case. Several other factors are influencing the measurements. Equation 7.1 has been devised and contains the most common signals the photo diode PD might receive. Each term of the equation will be discussed briefly while pointing out what this signal looks like.

$$PD = I_L\alpha + \sum_{i=1}^{n} I_{Edc_n}\beta_n + \sum_{i=1}^{n} I_{Eac_n}\gamma_n + N_{50Hz} + N(\mu, \sigma^2) \tag{7.1}$$

I_L represents the light emitted by the light. This gets multiplied with α, which represents the environment from the point of view of the system. These two terms represents the ideal response. The expected frequency of α should lie between XHz and YHz for by passing pedestrians, as shown in chapter 4. The goal of the complete algorithm is to isolate α and detect significant changes in it real-time.

The next term, $\sum_{i=1}^{n} I_{Edc_n}$, represents all constant, but slowly changing light sources in the area. An example of this is moonlight. Moonlight illuminates the surrounding area, but slowly changes over time because moon moves over time, or clouds blocking the moonlight. β_n represents the environment from the point of view of the moon.

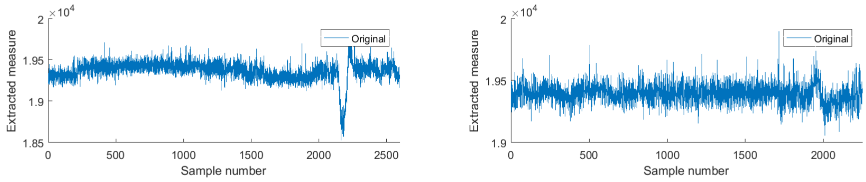
$\sum_{i=1}^{n} I_{Eac_n}$ represent all fluctuating light sources in the area. Most lights connected to the power grid fall into this category. They typically turn on and off at 100Hz in Europe. Some of the light produced by these source could reflect of off the environment γ_n and reach the system and therefore influence the received signal.

Another term in the equation is N_{50Hz}, which represents 50Hz noise from the powergrid. As long as the system is connected to the grid, some 50Hz components will be seen in the system. Especially if amplified 1000 times.

the final term, $N(\mu, \sigma^2)$, represents the noise on the measurements not created by the "predictable" sources listed above. This noise originates from the imperfections of the platform and electromagnetic noise in the environment. In the previous chapter it was shown that the noise could be approximated with a Gaussian curve and its therefore represented by its mean (μ) and variance (σ^2).

7.2 Filter methods

The goal of the filters is to get rid of unwanted signals in order to make the detection of α easier. Several digital filters types have been considered, each with different goals mind. The effectiveness (or failure) of each proposed filter will be shown, where possible, with the help of the signals shown in figure **??**. (a) shows an optimistic case with an original SNR of 18.5. (b) shows a harder case with an original SNR of 10.5. This figure will display the filter working in a much harsher condition.



(a) (b)

Figure 7.1: Two signals of a person walking underneath the set-up. Figure (a) shows an optimistic case with an SNR of 18.5 and (b) shows a harder scenario with an SNR of 10.5.

7.2.1 Low-pass filters

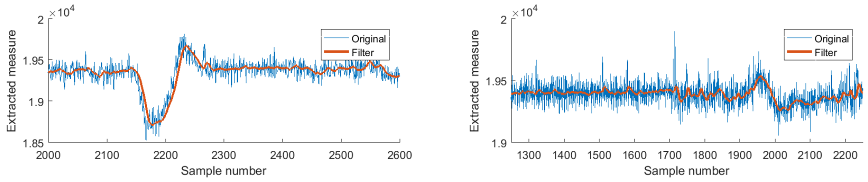
A lowpass filter can be used to remove I_{Eac_n} and N_{50Hz} from the measured signal as their frequencies are far removed from the signal we are interested in α (0.1 - 2Hz). Low-pass filters have one big downside for the system. They introduce a delay in the signal when used which is bad for the overall response time. Several filters have been tested. The final result is shown in figure **??** and is a second order IIR lowpass with its corner frequency at XHz.

With this filter, the complete N_{50Hz} component of the signal is removed and in most cases, I_{Eac_n} is removed as well. We are however unable to grantee the removal of I_{Eac_n} because of possible signal aliasing.

Signal aliasing is a phenomenon which occurs if the sample rate F_s of a system is too compared to the signal being sampled. If F_s is smaller then twice the frequency of then the sampled signal, the signal will appear as another frequency instead, an alias. This frequency is called F_{alias} and can be calculated with equation 7.2, where n is the closest integer mutiple of F_s to the signal being aliased (F_{Iac}).

$$F_alias = |F_s * n - F_{Iac}| \tag{7.2}$$

Almost all lights have a flicker frequency higher than half the sample rate of the system and will therefore alias. In Europe most lights have blink frequencies which are multiples of $50Hz$ (frequency of the power grid) and will therefore show up with an alias frequency of $25Hz$. This frequency is can still be removed with the used low-pass filter. There is however no grantee that all lights will blink at a multiple of $50Hz$. In the Americas for example the grid is powered at $60Hz$. The chanse is very high that a light there typically flickers at $120Hz$, which will alias at $5Hz$. This frequency is too low for the low pass filter to remove and will have to be death with in another way (if it occurs).



(a) (b)

Figure 7.2: A lowpass filter ($F_{cutoff} = 5Hz$) applied to the two example signals. The SNR for signal (a) increased to 76.3 from 18.4 and the SNR of signal (b) increased to 12.6 from 10.5.

7.2.2 Highpass filters

Highpass filters can be used to remove I_{Edc_n} from the signal. This is in this specific case very hard as the frequency we are interested in is very close to 0 compared to our sampling rate. It works, but it takes the filter a very long time to settle if a permanent change occurs in the environment. An example of this can be seen in figure X. In the figure a step is introduced at $t = 1000$, the signal has only settled after 11000 samples (90 seconds), which is not acceptable for the application. For this reason, the high pass filter is not part of the final algorithm.

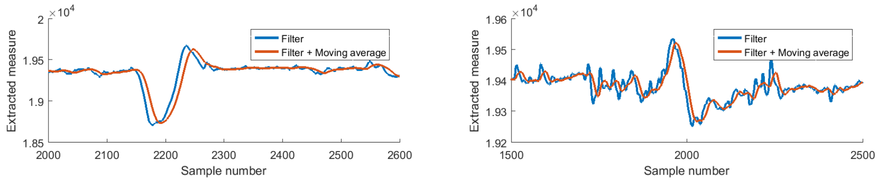
7.2.3 Moving average filters

A moving average can be used to reduce $N(\mu, \sigma^2)$ and the remaining F_{allias}. A moving average is effectively a simple low pass filter with specific frequencies being removed completely at $\frac{F_s}{n} * x$, where n is the number of tabs of the moving average and x any integer greater than 0. Therefore a make shift filter can be created instantly if F_{alias} is known, with $n = \left\lfloor \frac{F_s}{Falias} \right\rfloor$.

F_{alias} could be determined with the help of a Fourier transformation and then filtered away with a make-shift moving average. Yes, the Fourier transform would cost a lot of computation power which is against our goal of creating a computationally light algorithm, but the transform wouldn't have to be ran every sample. It is probably good enough to run it once every 10 minutes, to check if F_{alias} has been changed.

Another advantage the moving average brings is, that if the resulting noise is Gaussian, the noise gets reduced by a factor $\sqrt{(n)}$, where n is the number of tabs in the filter. It was therefore considered to scale the moving average, based on the current standard deviation of the noise with the help of equation 7.3. The presented formula calculates n, so that the

This method has two huge downsides. The first is that a moving average, capable of changing every incoming sample is computational expensive. If n changes, then the full moving average needs to be re-evaluated (n sum-



(a) (b)

Figure 7.3: A 10 tabs moving average applied to the two filtered example signals. The SNR for signal (a) increased to 80.1 from 76.3 and the SNR of signal (b) decreased slightly from 12.6 to 12.5.

mations, 1 division)) instead of using a simple update rule (1 summation, 1 division). Another downside is that if n gets too large, the response time of the system go down. For these two reasons, the scaling moving average was not implemented in the final algorithm.

$$\frac{signal}{noise} = 1 = \frac{\mu * ss}{T * \frac{\sigma}{\sqrt{n}}} \Rightarrow n = \left(\frac{T * \sigma}{\mu * ss}\right)^2 \tag{7.3}$$

7.2.4 Differential filter

The differential filter makes use of the fact that the system is not only able to sample when the light is turned on. Instead, It is possible to take samples while the light is turned off, to obtain PD_{dark}. This signal represents all the signals in the environment we are not interested in. If this signal is obtained very close to in time relative to PD ($20\mu s$), then we can assume that all fluctuating sources in both, PD and PD_{dark}, are equal. It's therefore possible to subtract the two signals, which would result in the filtered signal shown in equation 7.5.

$$PD_{dark} = \sum_{i=1}^{n} I_{Edc_n}\beta_n + \sum_{i=1}^{n} I_{Eac_n}\gamma_n + N_{50Hz} + N(\mu, \sigma^2) \tag{7.4}$$

$$PD - PD_{dark} = I_L\alpha + N(0, \sigma^2 + \sigma_{dark}^2) \tag{7.5}$$

There are several downsides to this filtering method. The first is that we are subtracting two separate measures of the same noise signal. This leads to a higher variance on the complete signal and therefore a higher noise level. Another downside of this method is that it does not work properly with the current hardware set-up because the PD_{dark}, on its own, is below the sensitivity threshold of the receiver and is therefore unmeasurable, unless there is a lot of stray light in the area.

Filter type	Goal	Notes	In final algorithm?
Low pass filter	Filter I_{Eac} and N_{50Hz}	Can't guarantee the removal of I_{Eac} due to signal aliassing	Yes
High pass filter	Filter I_{Edc}	Slow step response and therefore unusable	No
FFT based moving average	Filter F_{alias} and reduce $N(\mu, \sigma)$	Works, as long as F_{alias} is not too close to $I_L\alpha$	Yes
SNR based moving average	reduce $N(\mu, \sigma)$	Worked, but introduced huge delays for high σ and was is computational intensive	No
$PD - PD_{dark}$	Filter I_{Eac}, N_{50Hz} and I_{Edc}	Only worked in illuminated enviroments which made it unriliable to filter N_{50Hz}	No

Table 7.1: Overview of the filter methods described in this section

7.2.5 Filter overview

Several methods for removing unwanted parts of the received signal have been presented and summarised in table **??**. The final solution implements only the low pass filter and the moving average scaling based on F_{alias}. Figure X shows the remaining distribution of noise after the two implemented filters for both, the hard case and the easy case.

7.3 Detection threshold

There are several reasons why the threshold can't be static and has to self adjust overtime:

- I_dc -
-

7.3.1 Naive thresholds methods

7.3.2 Standard deviation based threshold

A threshold can be made based on the standard deviation of the signal. This method in literature is called the blabla method [**?**].

A delay can be added between the threshold and X, to increase detection ratio and detecting speed.

If the delay is too long it might increase the false positive ratio if the signal is slowly drifting upwards or downwards.

7.3.3 Variance based threshold

A threshold can be made based on the variance. By comparing

T	Chance false positive	Single occurrence @125 Hz	Double occurrence @125Hz
2	4.550026%	0.18s	0.69 days
3	0.269979%	2.96s	198.5 days
4	0.006334%	126.3s	1001 years
5	0.000057%	1178.2s	12199827 years

Table 7.2: Chance of a false positive occurring for several values of T, how often this would happen

7.3.4 Detection threshold

A naive solution to this problem would be to sample a set amount of values when there are no objects in sight. Then, take the maximum and minimum of the sampled values and if the signal ever moves out of the range of the found values, activity is detected. Even though this might work consistently in a dark room (lab environment with no lights), it fails to work in a more realistic environment. If we for example introduce a slowly rising I_{Edc} (e.g. moonlight), then the signal will eventually peak above the current maximum value and trigger a false detection.

Another way of tackling this problem would be to allow the minimum and maximum thresholds to move up and down with the mean of the signal. This results in two thresholds moving up and down together with the mean of the signal, and therefore ignores the slow changing I_{Edc}. The downside of this solution is that if the noise level ($N(\mu, \sigma^2)$) where to increases, then the signal would still cross the set threshold and trigger a false detection. The opposite is also true. If the noise level decreases, then the threshold would not scale back automatically and thus making it "deaf" to smaller changes in the signal.

This problem can be solved using the standard deviation of the signal as thresholds instead. A standard deviation scales up and down based on the deviation from the mean, meaning that if a lot of noise is present in the signal, then the detection borders would scale up and vice versa. The detection borders could be set on $\mu \pm T\sigma$, where μ is the mean, σ is the standard deviation and T is a factor determining width of the threshold

Using this method has another benefit. As the noise in our system ($N(\mu, \sigma^2)$) can be approximated with a normal curve (see section **??**), it allows us to control the amount of false positives perceived by the system by adjusting the T parameter. How T influences the chance of a false positive can be seen in table **??**.

7.3.5 Algorithm overview

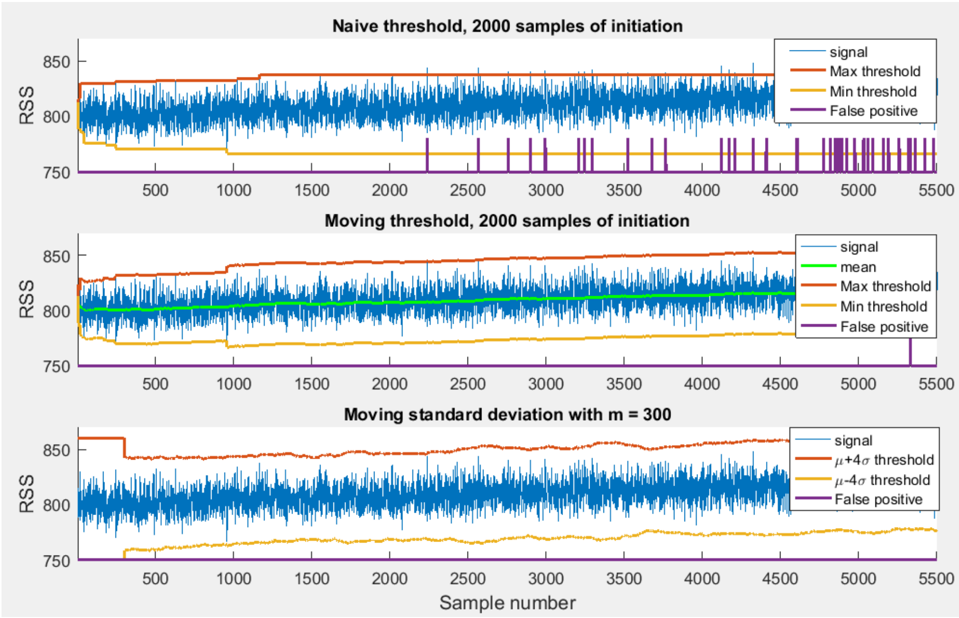


Figure 7.4: An example of how the discussed threshold algorithms respond to a slowly rising noisy signal.

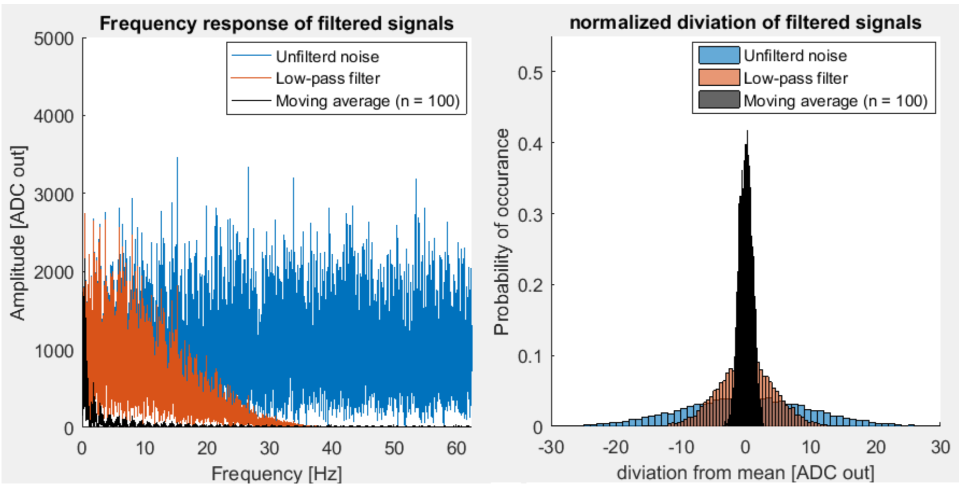
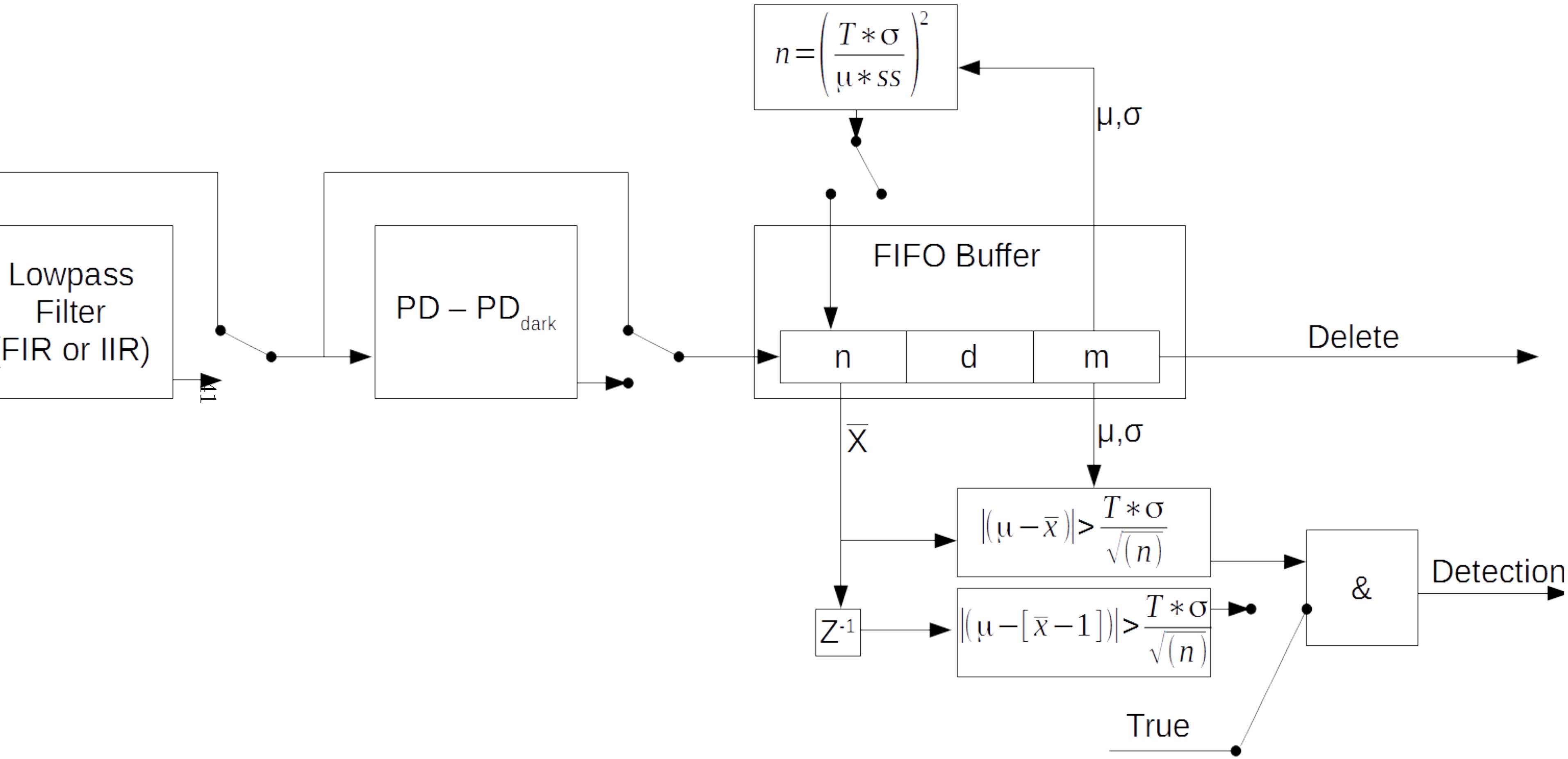


Figure 7.5: Frequency response of the noise, the noise when filtered with a low-pass filter, and then noise when filtered with a moving average.


$n = \left(\frac{T * \sigma}{\mu * ss}\right)^2$
μ,σ
Lowpass
Filter
(FIR or IIR)
PD – PD$_{dark}$
FIFO Buffer
n
d
m
Delete
$\overline{X}$
μ,σ
$|(\mu - \bar{x})| > \frac{T * \sigma}{\sqrt{(n)}}$
Z^{-1}
$|(\mu - [\bar{x} - 1])| > \frac{T * \sigma}{\sqrt{(n)}}$
&
Detection
True

Chapter 8

System evaluation

In the previous chapter an algorithm was presented, which could be capable of classifying a set of consecutive samples into two groups: Activity detected or no activity detected. The algorithm is however dependent on several values which have not been determined yet as they might, or might not be dependent on the operating environment. This chapter will therefore focus on finding the ideal values for several test environments and compare them with each other.

This chapter will first present the measures used for evaluating the system. It will follow up with describing and evaluating several pedestrian test environments. It will then do the same for a scale model simulating a street and finalizes with summarising the results.

8.1 Measures of evaluation

The system will be evaluated on three different criteria: Precision, recall and and response time. Precision is defined in equation 8.1 and gives us insight in how many situations the light would turn on unnecessarily. Recall is defined in equation 8.1 and gives us insight in how often the light fails to turns on when an object passes by.

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \tag{8.1}$$

The precise repose time is impossible to determine with the used dataset, as the starting time of the event is not defined. It is however possible that if two different settings of algorithms trigger a detection, we compare the detection times with each other.

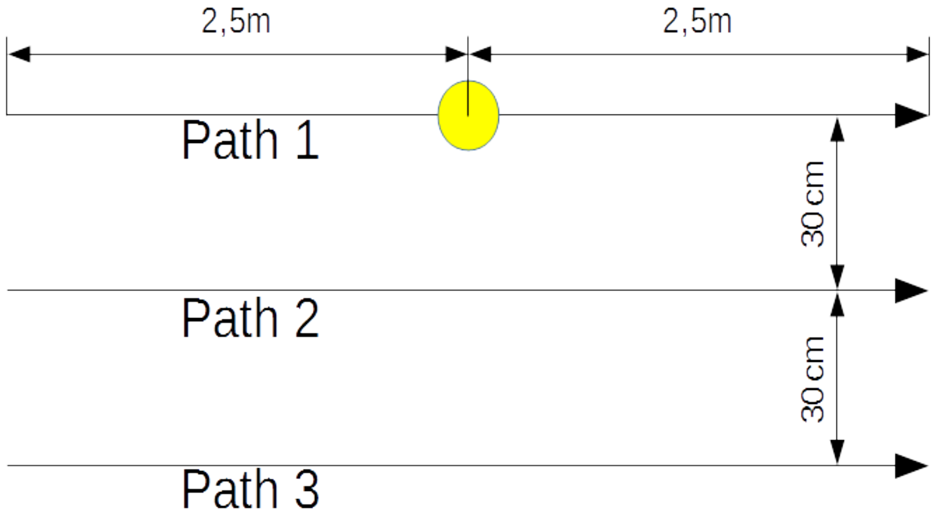


Figure 8.1: The three paths the test subjects had to walk underneath the light.

8.2 Hallway Evaluation

This section evaluates the system on detecting pedestrians walking by through a hallway. It will first explain the test set-up and procedure, followed by a section showing the ideally found algorithm settings and their detection results.

8.2.1 Test set-up

8.2.2 Results

8.3 Street

This section evaluates the system on detecting cars driving by on a road. It will first explain the test set-up and procedure, followed by a section showing the ideally found algorithm settings and their detection results.

8.3.1 Test set-up

8.3.2 Results

8.4 Conclusion

(a) Indoor (b) Outdoor

Figure 8.2: Pictures of the test set-up indoor (a) and outdoor (b). The light was suspended at 2.6m high.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

TODO CONCLUSIONS - It works with crappy hardware -

9.2 Future Work

The present work severs as a proof of concept for detecting activity in the line of sight of an LED. I personally think that the potential of this system is huge, especially if a dedicated platform is created. Below I have listed several ideas for future research, which I think have great potential, if a proper platform is created and it would be awesome if anyone would the dark sensing project to the next level.

- **Multiple units in one room** - The algorithm is currently designed for a stand-alone device. If we would hang multiple of these systems in the same room then it's likely that some of the light flashes overlap and trigger a false positives regularly. This problem could be solved by having each detector flash in another timeslot, but this requires more research.

- **Tracking** - The system is currently only detecting activity. It could also be expanded for other purposes. It might for example be possible to use multiple photo diodes, lenses or field of view blockers to track bypassing pedestrians.

- **A dark sensing network** - Multiple working units in one room is nice, but multiple units working together to track, predict and illuminate the path of a pedestrian is nicer. This could be achieved by having the devices communicate using the flashes already generated by the device (visible light communication).

Bibliography

[1] Marilyne Andersen. Types of reflection. https://www.flickr.com/photos/mitopencourseware/4815499473/.

[2] Arduino.cc. https://store.arduino.cc/arduino-uno-rev3.

[3] Jialiang Zhang Chi Zhang, Joshua Tabor and Xinyu Zhang. Extending mobile interaction through near-field visible light sensing. *MobiCom '15 Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 345–357, September 2015.

[4] Lilian Tieman Chris van der Nat, Hans Schelvis. *Bestaande bouw, Handboek Politiekeurmerk Veilig Wonen*. Centrum voor Criminaliteitspreventie en Veiligheid, december 2015.

[5] Thiemo Voigt Carlos Perez-Penichet Elena Di Lascio, Ambuj Varshney. Localight - a battery-free passive localization system using visible light. *IPSN '16 Proceedings of the 15th International Conference on Information Processing in Sensor Networks*, (60), April 2016.

[6] H.P. Lopuha L.E. Meester F.M. Dekking, C. Kraaikamp. *A modern introduction to probability and statistics*, chapter Confidence intervals for the mean, pages 341–360. Springer, 2014.

[7] H. Akbari H. Taha, D. Sailor. High-albedo materials for reducing building cooling energy use. Technical report, Lawrence Berkeley Lab., CA (United States), January 1992.

[8] H. Halbritter. High-speed switching of ir-leds (part i) - background and datasheet definition. Technical report, OSRAM, Opto Semiconductors, March 2014.

[9] National instruments. http://www.ni.com/white-paper/3000/en.

[10] L. Klaver and M.A. Zuniga. Shine: A step towards distributed multi-hop visible light communication. *12th IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS)*, pages 235 – 243, October note = 2015.

[11] Ikea 702.880.22 led1336r4. http://lamptest.ru/review/ikea-70288022-led1336r4/.

[12] TVI light. www.tvilight.com/citysense.

[13] Esmail M.A. and H.A. Photon Fathallah. Indoor visible light communication without line of sight: investigation and performance analysis. *Photonic Network Communications*, pages 159–166, October 2015.

[14] Mastech. *MS8229, Digital multimeter operation manual.*

[15] Jaakko Maentuasta Miika Valtonen and Jukka Vanhala. Tiletrack: Capacitive human tracking. *Pervasive Computing and Communications*, March 2009.

[16] Siddharth Rupavatharam Minitha Jawahar Marco Gruteser Richard Howard Mohamed Ibrahim, Viet Nguyen. Visible light based activity sensing using

ceiling photosensors. *VLCS '16 Proceedings of the 3rd Workshop on Visible Light Communication Systems*, pages 43–48, October 2016.

[17] Ashok Agrawala Moustafa Youssef, Matthew Mah. Challenges: Device-free passive localization for wireless environments. *MobiCom '07 Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 222–229, January 2017.

[18] L. Benini P. Zappi, E. Farella. Enhancing the spatial resolution of presence detection in a pir based wireless surveillance network. *Advanced Video and Signal Based Surveillance, 2007*, September 2007.

[19] Zhao Tian Andrew T. Campbell Tianxing Li, Chuankai An and Xia Zhou. Human sensing using visible light communication. *MobiCom '15 Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 331–344, September 2015.

[20] Junwei Zhang. Passive localization with visible light. MSc thesis, TU Delft, 9 2016.

[21] Kevin Wright Zhao Tian and Xia Zhou. The darklight rises: Visible light communication in the dark. *MobiCom '16 Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 2–15, October 2016.

[22] Kevin Wright Zhao Tian and Xia Zhou. Lighting up the internet of things with darkvlc. *HotMobile '16 Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*, pages 33–38, February 2016.

[23] Zumtobel. The lighting handbook. In *You concise reference book*. Zumtobel, Schweizer Strasse 30, 6851 Dornbirn, AUSTRIA, 5th edition, 2017.

Appendix A

Code repository

All code referred to in this thesis can be found at `https://github.com/hkleingeld/DarkSensing`. The folder structure shown in fig X should be self explanatory. Readme files where added to each folder to explain what each file contains and is used for.

- `root`
  - `PC`
    - `Analyzers`
    - `DarkSenseAnalyzer`
    - `DarkSenseCapture`
    - `DarkSenseRunning`
    - `Shared`
  - `Device`
    - `Superblink`
    - `Firmware`
  - `Matlab`
    - `Model`
      - `Street`
      - `Hallway`
      - `Verification`
  - `Thesis`

Appendix B

Raw Model results

This appendix contains the raw results of the model.

y	H	min $\alpha = 0.2$	max $\alpha = 0.2$	min $\alpha = 0.3$	max $\alpha = 0.3$	min $\alpha = 0.4$	max $\alpha = 0.4$	min $\alpha = 0.5$	max $\alpha = 0.5$
0m	1.4m	-0.08	0.70	-0.03	1.30	0	1.90	0	2.50
	1.6m	-0.07	1.60	-0.02	2.66	0	3.73	0	4.80
	1.8m	-0.06	3.46	-0.01	5.54	0	7.61	0	9.69
0.2m	1.4m	-0.05	0.65	-0.01	1.17	0	1.70	0	2.23
	1.6m	-0.05	1.34	0	2.22	0	3.11	0	3.99
	1.8m	-0.04	2.64	0	4.24	0	5.83	0	7.43
0.4m	1.4m	-0.07	0.28	-0.02	0.59	0	0.90	0	1.21
	1.6m	-0.07	0.56	-0.02	1.01	0	1.46	0	1.91
	1.8m	-0.07	0.91	-0.02	1.59	0	2.26	0	2.93
0.6m	1.4m	-0.08	0.04	-0.03	0.18	-0.01	0.33	0	0.47
	1.6m	-0.09	0.09	-0.04	0.27	-0.01	0.44	0	0.62
	1.8m	-0.10	0.05	-0.05	0.26	-0.01	0.47	0	0.68

Table B.1: Differences with baseline (no object) for each simulated situation

y	Colour	min $\alpha = 0.06$	max $\alpha = 0.06$	min $\alpha = 0.14$	max $\alpha = 0.14$
1.5m	Silver				
	Black				
	Red				
4.5m	Silver				
	Black				
	Red				

Table B.2: Differences with baseline (no object) for each simulated situation

Object albedo	H	min $\alpha = 0.06$	max $\alpha = 0.06$	min $\alpha = 0.14$	max $\alpha = 0.14$
0.1	1.8m				
	1.6m				
	1.4m				
0.4	1.8m				
	1.6m				
	1.4m				

Table B.3: Differences with baseline (no object) for each simulated situation

Appendix C

Flash analyser schematics

This appendix contains pictures of electronic schematic created for specifically the flash analyser. Schematics of the processor boards where not added, as only small changes small modifications (see section ??) where made to these boards. The original scematics can be found at:

- Flash analyser - `https://github.com/LennartKlaver/SDVN1`
- LED controller - `https://www.arduino.cc/en/uploads/Main/arduino-uno-schematic.pdf`

C.1 LED driver

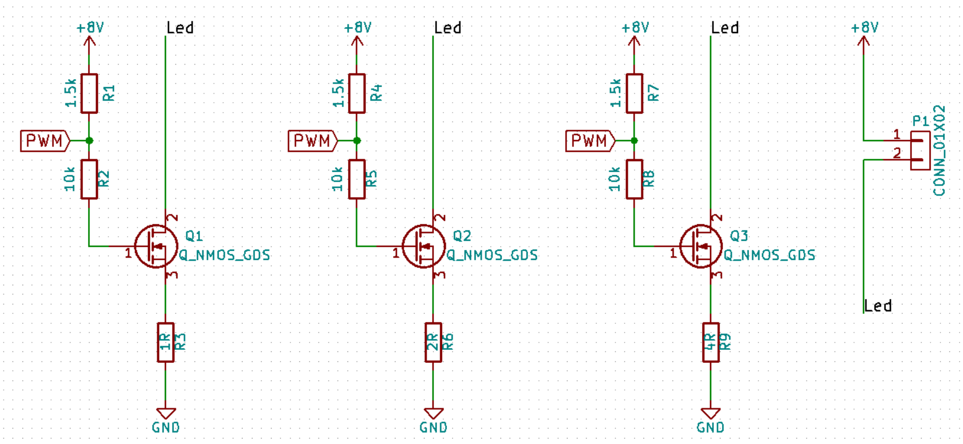


Figure C.1: Drivers used to drive the LED.

C.2 Interfaces between components

@TODO!