

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

5th December 2017

Author

Hajo Kleingeld (hajokleingeld@gmail.com)

Title

Sensing human activity with dark light

MSc presentation

5th December 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

5th December 2017

Contents

Preface	v
1 Introduction	1
1.1 Problem statement	2
1.2 Contributions	3
1.3 Organisation	3
2 Background and related work	5
2.1 Background	5
2.1.1 Dimming of an LED	5
2.1.2 The Phong model	7
2.2 Related Work	11
2.2.1 Passive localisation	11
2.2.2 Passive Visible Light Localisation	11
2.2.3 Other related projects	14
3 Model	15
3.1 Model description	15
3.1.1 Model Adjustments	15
3.1.2 Calculation process	16
3.2 Verification	16
3.3 Modelling of the hallway	19
3.4 Modelling of the street	20
3.5 Results	21
3.6 Conclusions	21
4 Platform	23
4.1 system components	23
4.1.1 Flash generator	23
4.1.2 Reflection receiver	24
4.1.3 Analyser	24
4.2 Implementation	25
4.3 Evaluation	26

5 Flash Analysis	29
5.1 Flash properties	30
5.2 Flash features	30
5.2.1 Feature considerations	30
5.2.2 Feature comparison	32
5.3 Flash period	34
5.4 Conclusion	34
6 Analyser	35
6.1 Received signals	35
6.2 Filter methods	37
6.2.1 Low-pass filters	37
6.2.2 Highpass filters	38
6.2.3 Moving average filters	39
6.2.4 Differential filter	40
6.2.5 Filter overview	40
6.3 Detection threshold	41
6.3.1 Standard deviation based threshold	41
6.3.2 Variance based threshold	43
6.4 Conclusion	44
7 System evaluation	45
7.1 Measures of evaluation	45
7.2 Hallway Evaluation	46
7.2.1 Test set-up	47
7.2.2 Results	47
7.3 Street	47
7.3.1 Test set-up	47
7.3.2 Results	48
7.4 Conclusion	48
8 Conclusions and Future Work	49
8.1 Conclusions	49
8.2 Future Work	49
A Code repository	53
B Raw Model results	55
C Flash analyser schematics	57
C.1 LED driver	57
C.2 Interfaces between components	58

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, namely *Dark Sensing*.

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, 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. Dark sensing 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 can be measured with the photo diode. The signal received is a measure of the illuminated area. If something were 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 1.1(a).

This method by itself does not save energy as the light used in the system is always turned on. If we are able to reduce the light output while maintaining the ability to pick up meaningful reflections, then the system would save

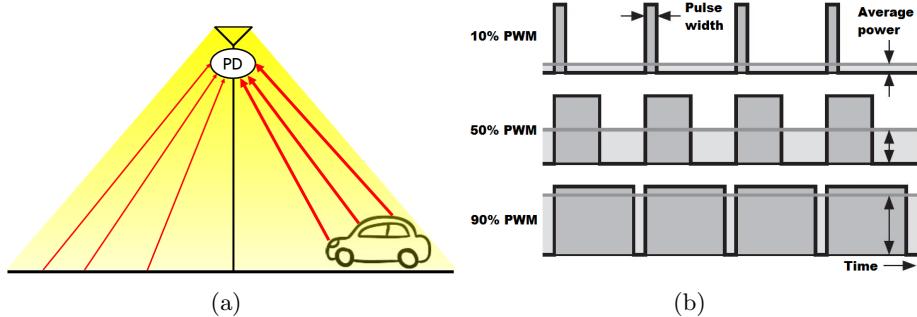


Figure 1.1: Analogue VS digital dimming.

energy. This can be achieved by turning the light on and off very quickly. Then, if the reflections are only measured when the light is fully turned on, we can capture a reflection while saving energy, because the light is only on part of the time.

The ultimate goal of the proposed system is to reduce the time the light needs to be on the nearly zero. This will lead to a light which barely consumes any energy while nobody is around but is still able to detect people, cars or other objects passing by. It might even be possible to decrease the light output to an amount which is invisible to the human eye, resulting in an unnoticeable, activity detecting, energy-saving device.

1.1 Problem statement

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

This problem can be divided into three sub-questions:

- How strong is a reflection obtained from a flash in a realistic scenario and how much does this reflection vary if an object enters 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 (besides 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

Note for proof readers: Note that the numbers here are approximations as the system has not yet been fully evaluated.

This thesis proposes **Dark Sensing**, a system that uses reflections of a LED controlled with a low duty cycle (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 95% of all humans passing by in a realistic environment and therefore saving 96% of the light used in comparison to the original situation.

1.3 Organisation

This thesis describes the development path of the new technique "Dark Sensing" from idea to a working prototype. Chapter 2 will present the required background knowledge to understand several choices made in this thesis and present the related work. In chapter 3 a model will be presented, which calculates the theoretical response of bypassing objects. Chapter 4 describes the created experimentation platform. Chapter 5 will focus on finding the ideal settings for generating an analysable flash and will explain what the best method is of extracting data from this flash. Chapter 6 explores the possibilities for analysing sets of consecutive flashes and proposes an algorithm to detect significant changes in the signal. Chapter 7 tests the prototype created and evaluates the performance of system. The thesis concludes with an evaluation of the new "Dark Sensing" technology and suggests several possible directions for future work.

Chapter 2

Background and related work

2.1 Background

This section presents the required field knowledge to understand this thesis. It therefore starts with an explanation on how a light source behaves when it turns on and off rapidly and why this is preferable over other light saving strategies. It then follows up with an explanation of how produced light travels and reflects off of surfaces.

2.1.1 Dimming of an LED

In this thesis, an LED will be used to illuminate the environment which will cause reflections in the room. It's therefore important to understand how the light responds to different methods of adjusting the light output, as this directly influences power of the reflections.

In general, there are two methods of dimming (reducing the light output of) an LED: Analogue and digital. A light which is dimmed in an analogue manner has its total light output reduced by reducing the current flowing into the LED. This leads to a light which has a constant light output directly proportional to the current flowing into the LED. If we for example want a light to produce 25% of its normal light output, we simply supply it with a quarter of the current. A graphical representation of analogue dimming is shown in Figure 2.1(a) and is marked as "average power".

Digital dimming works in another way. Instead of controlling the amount of current flowing into the LED, we control the amount of time current is allowed to flow into the LED. This can be achieved by turning the LED on and off rapidly. If we for example want to reduce the light output of an LED to 25% with the help of digital dimming, then we would turn the light on at full strength for 25% of the time, while turning it off for 75%, with the help of a Pulse Width Modulated (PWM) signal. A graphical representation of digital dimming is shown in Figure 2.1(a).

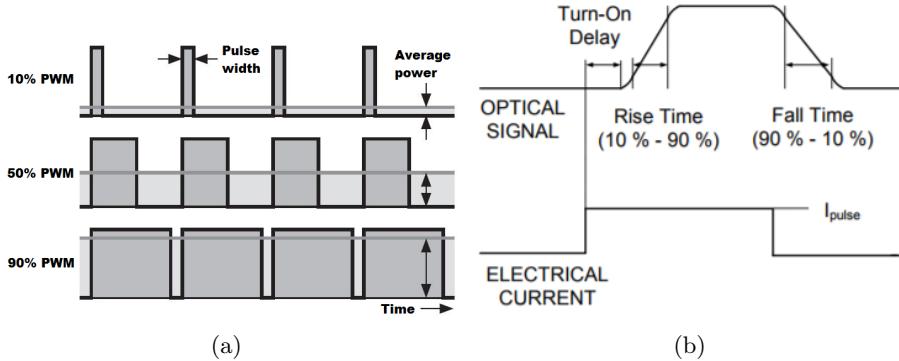


Figure 2.1: Figure (a) shows the difference between analogue and digital dimming, Figure (b) shows a realistic response of the light when a short electric pulse is applied[7].

The resulting light produced by both types of dimming are indistinguishable for humans if the switching frequency is high enough. Both methods have the same apparent brightness and use the same amount of power. For photo diodes however, there is a clear difference. The analogue signal will show up as a constant, but weak signal. The digital signal shows up as a square wave with high peaks (when the light is on) and valleys (when the light is off). This becomes especially apparent if we want the system to work at only 1% of it's original illumination level. The signal dimmed in an analogue manner will be nearly invisible as it is turned on constantly at 1% of it's original power. This in contrast to the digital signal, which only shows up for 1% of the time, but at maximum power, resulting in a shorter but much brighter peak. Because the 1% time constrain is no problem for an electronic system, it was chosen to explore Dark Sensing with this dimming method.

There is however a limit to how much the energy consumption can be reduced with digital dimming, if we want to be able to observe the signal with a photo diode. When an LED is turned on, it does not produce light at maximum intensity instantly[7]. It first has a short "turn-on delay" where the light does not output any light, followed by a "rise time" where the light "slowly" powers up until it has reached it's desired intensity level. A graphical representation of this process can be seen in figure 2.1(b). This limit on digital dimming forces a hard minimum to the amount of digital dimming the system can work with and therefore limits the amount of energy it can save.

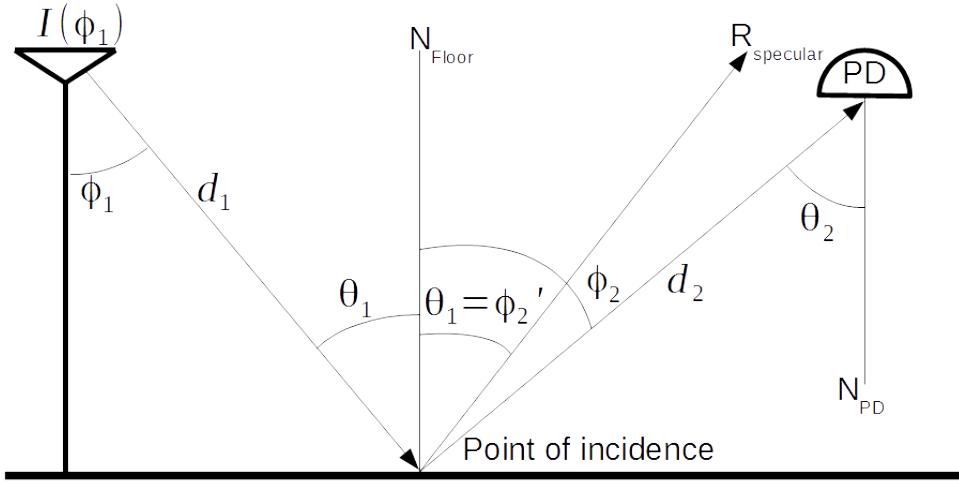


Figure 2.2: Overview of angles, vectors and distances used in the model. It represents a street light illuminating the street (I). This light is then observed by a photo diode (PD), aimed at the ground. Note that ϕ always represents an exit angle and θ always represents an incidence angle.

2.1.2 The Phong model

When a light shines on a surface, some parts of the surface appear brighter than other parts. This is caused by three major factors:

- The light used to illuminate the wall and it's position relative to the wall.
- The surface of the wall itself.
- The position of the observer relative to the wall.

If all of these factors are known, then the complete pathing of the light can be approximated with the help of the Phong model. This section presents a simplified version of the Phong model which is used in chapter 3. All used angles can be seen in Figure 2.2. The full model can be found in [17].

Modelling an LED

A light can be modelled if several parameters of the light are known, with the help of equation 2.1. This formula describes how much light is leaving the light source at angle ϕ relative to the normal of the LED.

$$I(\phi) = \Phi_{lum} \frac{\alpha + 1}{2\pi} \cos^\alpha(\phi_1) \quad (2.1)$$

The equation consists of three parts. Φ_{lum} represents the total amount of light emitted in lumen by the LED. $\cos^\alpha(\phi_1)$ represents the radiation pattern of the LED. α represents the order of Lambertian emission which describes the illumination pattern of the LED. If α is low (e.g. 1), then this equation

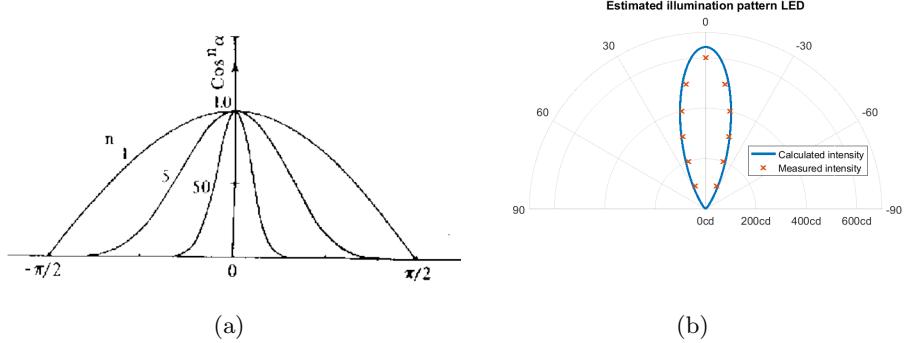


Figure 2.3: (a) shows an overview of how adjusting α changes the light cone of a simulated light source. (b) shows a modelled light cone, modelled with the shown measured light cone.

represents a luminaire with a very wide spread of light, for example a street light. If α is high (e.g. 200+), then the light source is much more focused like a laser. An overview of α values versus their angle is shown in Figure 2.3. $\frac{\alpha+1}{2\pi}$ is a normalisation factor that ensures that integrating equation 2.1 results in the total amount of light produced (Φ_{lum}), as reshaping the cone of light with α would otherwise lead to a change in produced light. An example of a modelled light with $\alpha = 14.3$ and $\Phi_{lum} = 260lm$ can be seen in Figure 2.3(b).

We can now take any light ray from the luminaire and calculate how much light hits a specific surface with the help of equation 2.2. This calculation also consists of three parts. The first part is the strength of the light ray calculated with equation 2.1. The second part, d , represents the distance the light needs to travel before it hits the surface. The final variable, θ_1 , represents the incidence angle of the light ray on the surface.

$$E_{hor} = \frac{I(\phi_1)}{d^2} \cos(\theta_1) \quad (2.2)$$

Modelling a reflection

Light impinging on a surface can reflect in three different ways: Diffuse, spread and specular. Almost all surfaces combine several of these reflection types. A visualisation of these reflections can be seen in Figure 2.4. The **specular reflection** is a so called perfect reflection. It reflects each incident ray outward, with the reflected ray having the same exit angle to the normal vector N as the incident ray. A material with this kind of property is a mirror. The **diffuse reflection** is the opposite. Instead of reflecting light in one direction, the light ray is scattered in all directions following

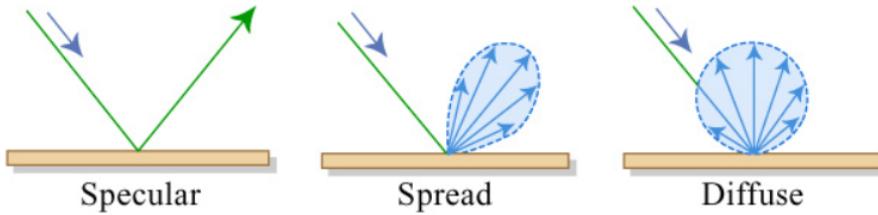


Figure 2.4: The possible ways for light to reflect when it hits a surface [1].

a Lambertian emission pattern. This leads to a point, which appears to have the same brightness, no matter the observation angle. A common material with this property is plain white paper. The final kind of reflection is the called the **spread reflection**. It's a scattered reflection, aimed in the direction of the ideal reflection. A material which has mainly this kind of reflection is matte aluminium.

All of these reflections can be modelled with the help of equation 2.3, where ϕ_2 represents the observation angle. The first part of the equation calculates how much of the impinging light is reflected off the surface and not converted into heat. This is determined by A , which represent the albedo of the observed surface.

The second part describes the actual reflection of the surface. $\frac{\alpha+1}{2\pi} \cos^\alpha(\theta'_1 - \phi_2)$ describes the spread reflection and is modelled as light source pointing in the direction of the ideal reflection θ'_1 (see equation 2.1). The higher α is chosen, the more focused the reflection. If α is chosen to be infinite, the surface is modelled as a mirror instead.

$\frac{1}{\pi} \cos(\phi_2)$ describes the diffuse part of the equation and is also modelled with 2.1 where $\alpha = 1$. This results in a diffuse reflection. The final term of the equation is r_d . This value represents the ratio between the diffuse and spread reflection.

$$R(\theta_2) = E_{hor}\rho(\lambda) \left[r_d \frac{1}{\pi} \cos(\phi_2) + (1 - r_d) \frac{\alpha + 1}{2\pi} \cos^\alpha(\theta'_1 - \phi_2) \right] \quad (2.3)$$

Modelling a photo diode

The final part missing in the model is the observer. The observer, or receiver in our case, is a photo diode which can be modelled with the help of Equation 2.4. This equation is very similar to equation 2.2, but has one major difference: The $rec(x)$ function. This function checks if the light incoming at angle θ_2 lies in the field of view of the photo diode. If it is, then $rec\left(\frac{\theta_2}{FOV}\right)$ returns 1, otherwise it's 0 and the ray of light wont be counted.

$$PD = \frac{I \cos(\theta_2)}{d^2} rec \left(\frac{\theta_2}{FOV} \right) \quad rec(x) = \begin{cases} 1, |x| \leq 1 \\ 0, |x| > 1 \end{cases} \quad (2.4)$$

Creating a 3D model

All equations shown in the previous sections can be combined into one big equation, calculating how much a point on the wall is illuminated, reflected and perceived by the observer. This equation is 2.5. It has however a lot of variables, which will make it hard to create a proper simulation.

$$PD_{tot} = I(\phi, \alpha_{light}, \Phi_{lum}) R(\theta_1, \phi_2, \lambda, r_d, \alpha_{floor}) PD(\theta_2, d_2) \quad (2.5)$$

This can be solved by making the problem concrete and simulate it in a 3D space with an *xyz* coordinate system. If we assume the floor is a plane spanning x and y (thus z = 0) and fix the positions and normals of the LED and photo diode, we can express all angles and distances as formulas of x,y and z. If we then want to calculate total amount of energy perceived by the photo diode, all we need to do is integrate over all the points of the floor (the xy plane).

$$PD_{tot} = \int_x \int_y I(x, y, z, \alpha_{light}, \Phi_{lum}) R(x, y, z, \lambda, r_d, \alpha_{floor}) PD(x, y, z) \quad (2.6)$$

This model was used to create the model used in chapter 3. That chapter will also explain what changes were made to obtain the presented model.

2.2 Related Work

This section presents the related work of this thesis. It starts with giving a short overview of several methods used for passive localisation. It shows the projects which use visible light in their localisation schemes. This section finalises by highlighting a paper which attempts to reduce the visible light used in a similar way to this thesis.

2.2.1 Passive localisation

Passive localisation is a hot topic in research and has been tackled by many different research groups in several different ways. The most common method found in literature to detect and track humans is by using Passive Infra-Red (PIR) sensors. These sensors detect the infra-red (heat) radiating from objects and draw conclusions from the observed signals. The passive infrared sensor has been around since 1982 [5] and has been used to detect humans since 1994 [10].

These days, the research in PIR sensors for detecting and tracking humans focuses in two directions. The first direction is to get more information out of PIR sensors by examining the raw data. M. Waelchli *et al.* for example created a method for estimating the location of a person within the view of the sensor [11]. The second direction is to track humans with the help of several linked sensors. An example of this, by P. Zappi *et al.* is [18]. They linked a server to multiple binary human activity sensors, in order to locate and track humans in an indoor building.

Another method for passive localisation, which popped into existence more recently, is developed by M. Youssef *et al.* [16]. They created a detect and track application with the help of WIFI access-points (APs), WIFI monitoring-points (MPs) and an application server (AS). 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 2.5.

Another approach to passive localisation is to use the tiles upon we walk as sensors. This was done by M. Valtonen *et al.* [14]. The system measures the capacity between several floor tiles and a receiving electrode. With the help of the measurements, they estimate the position of the person standing on top of the tiles.

2.2.2 Passive Visible Light Localisation

In recent years, a new method for locating and tracking humans has been explored: Passive Visible Light Localisation (PVLL). This method is focussed on using visible light and photo sensors to detect and track humans

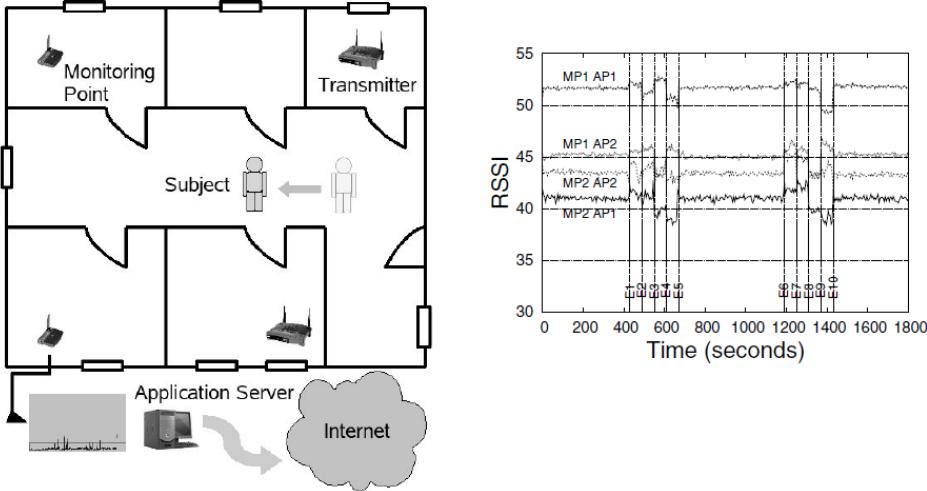


Figure 2.5: Overview of the WIFI tracking system of Moustsafa Youssef *et al.* [16]. The left figure shows an overview of the setup where the right figure shows the strength of the APs from the point of view of the MPs. E1 to E8 represent possible 'events' of bypassing persons.

and objects. Several of these project will be explained briefly, followed by a short comparison between these projects and Dark Sensing.

Local Light

Local light, developed by Lascio *et al.*[4], is a system which implements passive localization with the help of visible light. The system consists of 3 parts. A light, light sensing RFID tags and a server. The light illuminates the environment. The RFID tags are mounted in the floor, detecting the light produced by the luminaire. The tags transmit their data to a server which processes the data. An overview of the system can be seen in Figure 2.6.

The system works by detecting changes in the light intensity. If the photo diode suddenly senses a huge drop in light, because a shadow is casted on the photo diode by an object or person, the system triggers a detection. The server knows the exact location of all luminaires and photo diodes and is therefore capable to determine where the object or person is at this moment in time.

Activity sensing using ceiling photo diodes

Two different projects have been found which have developed a passive localisation scheme using several light/photo diode pairs mounted together on the ceiling. Both take a slightly different approach.



Figure 2.6: Overview of the LocalLight system of E.D. Lascio *et al.*[4]. Lights on the ceiling and light sensing RFID tags on the floor.

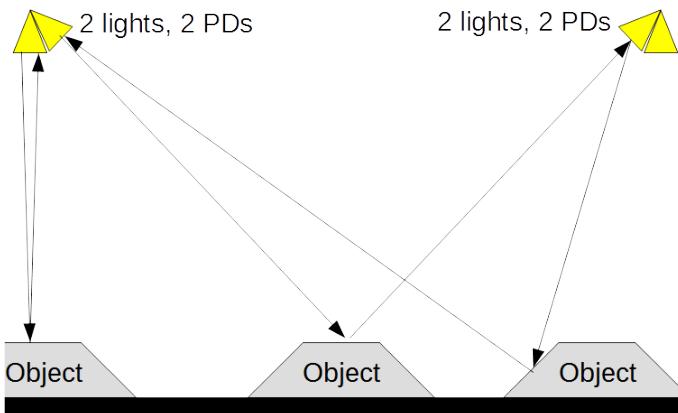


Figure 2.7: Overview of the system used by J. Zhang [19]

The first project, by J. Zhang [19], created a method capable of localising objects on a line between two light/photo diode pairs. By moving an object with three reflective surfaces underneath a light, he managed to localise them at several points on the line by using the specular component of the reflections bouncing off the object. His test set-up can be seen in figure 2.7.

The other project, by M. Ibrahim *et al.*, makes use of modulated lights. Each luminaire transmits light in a different pattern. The photo diode, which is placed next to the light, detects what patterns of light it perceives. If the photo diode does not sense one of the lights it normally does, it triggers a detection as the light was intercepted by a bypassing object. An overview of the set-up can be seen in Figure 2.8

Comparison with Dark Sensing

The Dark Sensing project differs from the existing projects in several ways. It's the only project tempting to create a sensing device, only requiring one sensor node instead of multiple and is therefore easier to install and expand. It's also the only project which attempts to detect humans with the main drive of saving energy. It's also the only project which potentially can be

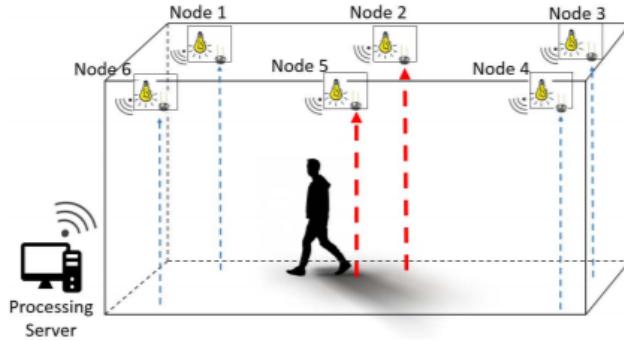


Figure 2.8: Overview of the system set-up used by M. Ibrahim[15]. In this specific situation node 2 and 5 detect no light from node 1, because a person is blocking the light.

implemented outdoor, mounted on a light post for example, as the other PVLL projects where require either a server in reach of the sensors, or other specific environmental features. Dark Sensing has the potential to be a stand alone product.

The downside of Dark Sensing is that it only focuses on detecting activity. It's therefore unable to determine where the activity is exactly. All of the other projects are way better in that specific area.

2.2.3 Other related projects

One project which has nothing to do with passive localisation, but can't miss from the related work section is "*The dark light rises*" by Z. Tian *et al.* [20] [21]. This group explores the idea of Visible Light Communication (VLC) with dark light, a VLC primitive that allows light-based communication to be sustained even when LEDs emit extremely-low luminance. 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. The goals of Dark sensing and Dark VLC are similar: Save light and therefore energy. Both projects however apply this method in other applications.

Chapter 3

Model

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

- How strong are the reflections of flashes in a realistic environment?
- How and how much, will these reflections change if an object enters the illuminated area

This section uses the model explained in section 2.1.2. It starts with what changes where made in the presented model, in comparison to the original [17] and shows that the model gives a reasonable estimation of reality. It then describes two modelled scenarios and presents the results. The chapter ends with answering the posed questions.

3.1 Model description

The model made is an interpretation of the Phong reflection model (see section 2.1.2). It calculates how much of the light leaving a luminaire, bounces back via the environment to a photo diode placed next to the light source. This section will first discuss the adjustments made to the Phong model, followed by an explanation of the simulation process.

3.1.1 Model Adjustments

The model presented in section 2.1.2 is not the complete Phong model. Several parts where simplified or removed as they should barely influence the results of the simulation.

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 photo diode is maximized for the current situation.

The second adjustment is the removal of "colour". The original method differentiated between different wavelengths of visible light, while producing,

reflecting and receiving light. It 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 (A).

$$\Gamma = \int_{380nm}^{780nm} \Phi_e p(\lambda) d\lambda \rightarrow \Gamma = \Phi_{lum} A \quad (3.1)$$

Albedo is a property of an object representing the ratio of energy which is reflected when sun is shining on it. 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 is shown in section 3.2.

The final adjustment is the amount of reflections we calculate. In reality a light ray can be reflected an infinite amount of times off 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). 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[12]. If we were to add multiple bounces, the accuracy of the model would only increase by a maximum 20%. Adding the extra bounces makes the model many more times complicated, depended on the environment we are simulating. A simple hallway model would be 8 times as complex to model and require at least 4 times as much computation power, while only providing "only" 16% more accuracy.

3.1.2 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 the light casting the shadow is reflected off 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) is received by the photo diode. The final step is to calculate how much light is reflected from each side of the object. 3.1 shows an overview of an environment with rays leaving the light, casting shadow and the resulting reflections.

3.2 Verification

The calculation method and changes in the model were verified using a scale model featuring a LED[9], a paper box and a light meter[13]. The first

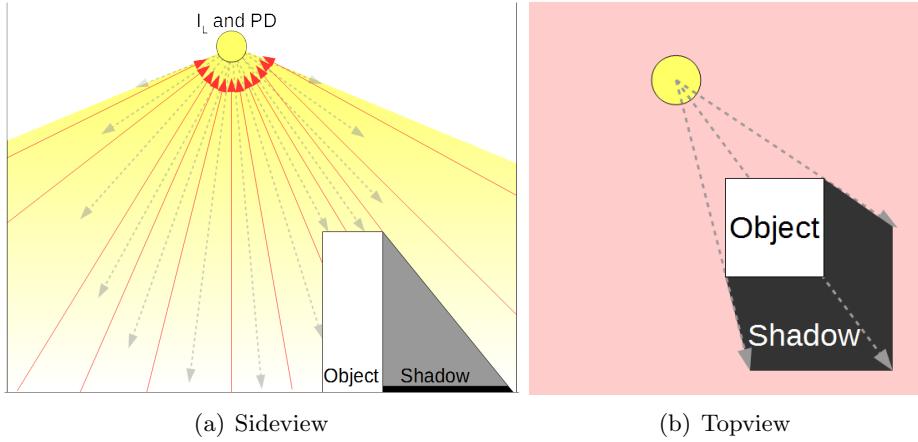


Figure 3.1: Overview of the calculation process. Grey lines represent light rays casted by the light. Black represents the shadow casted by the object on the floor or walls. Red lines or areas show reflections bouncing from the ground, walls or object back to the photo diode.

step of verifying the model is to check if the LED is modelled properly by equation 2.1. 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 2.2. Measurements and simulations in Appendix A show that the LED in the test set-up was producing more light than in the specification. These numbers were therefore adjusted for the next step of verification.

The second step is the verification of the interpretation of the Phong model. This was done with the test set-up shown in Figure 3.2. By moving a paper box across a paper covered floor in steps of 5cm and measuring the reflections in each step, we obtain the red line in figure 3.3(a). When we compare this line with the blue line generated by the model using $A = 0.75$ (albedo of paper according to [6]), then the lines closely match.

The test was repeated while using the original floor of the room. The albedo of the floor was calculated to be 0.37, based on a measurement of the floor without the paper box. The result of the second test can be seen in Figure 3.3(b). The resulting curves also seem similar. As the model has shown to reflect reality quite closely, it seems fine to assume that the model works and can be trusted some extent. It won't give exact results, but it will at least provide a proper approximation of the perceived light.

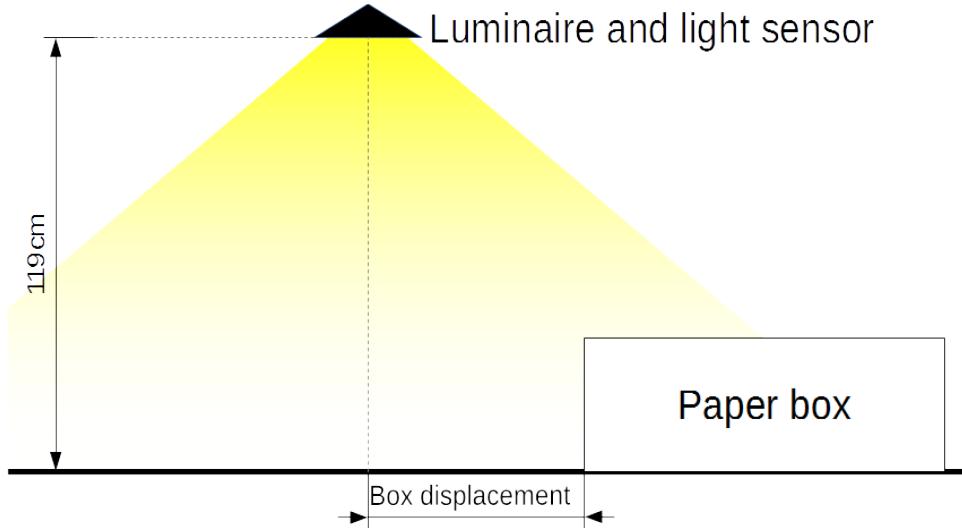


Figure 3.2: Visualisation of the model verification set-up.

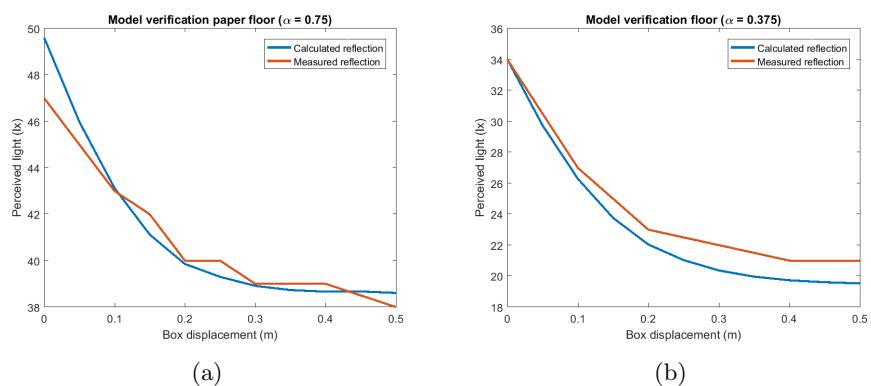


Figure 3.3: Both figures show that the model provides a reasonable approximation of the reality. Note that the albedo of paper was taken from [6] and the albedo of the floor was estimated with measurements.

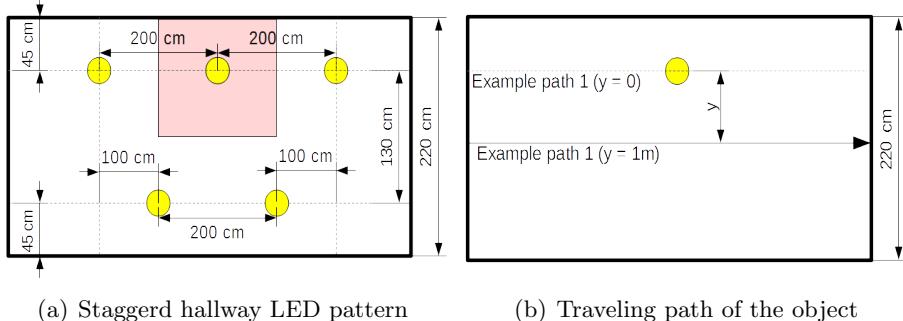


Figure 3.4: Figure (a) shows the position of the luminaires to obtain a realistic illumination pattern. The red square represents the area one light/photo diode pair should cover. Figure (b) shows an example travelling path of an object.

3.3 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[6]. 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 $U_o > 0.4$ [22]. E_{mean} represents the mean illumination level of the floor and U_o the proportional difference between E_{mean} and $E_{minimum}$. These lighting requirements can be achieved using the same luminaire used during the verification process if hung in the staggered formation shown in figure 3.4(a). Calculations showing that the industry standards are met can be found in Appendix A.

$$E_{mean} = \frac{1}{y \cdot x} \int_y \int_x E_{hor}(x, y) dy dx \quad U_o = \frac{E_{mean}}{E_{minimum}} \quad (3.2)$$

The object passing by the light (representing a human) will be modelled as a cuboid 0.2m wide and 0.5m long 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 through the hallway with the light at a set vertical distance y . Some example paths can be seen in Figure 3.4(b).

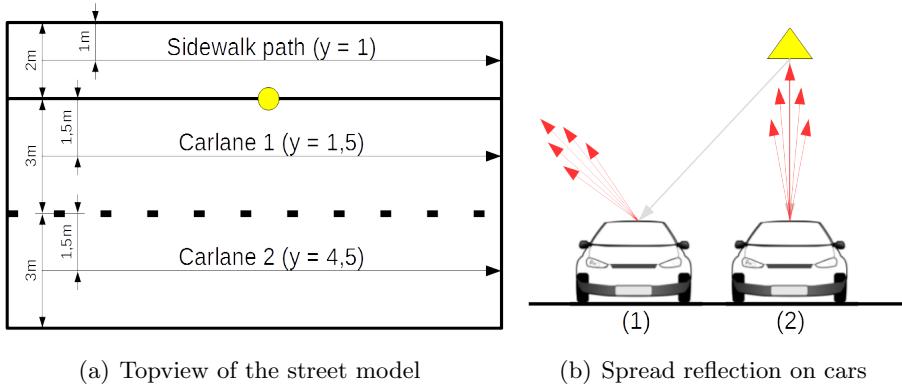


Figure 3.5: Figure (a) shows an overview of the model. Figure (b) shows that a spread (or specular) reflection will only reach the light in situation (2). This situation does not occur in the modelled scenario.

3.4 Modelling of the street

The street model is based on a real street near the TU delft. It has two lanes for cars (each 3m wide) and sidewalk (2m wide). The albedo of the street will be modeled with $A = 0.11$ which represents old asphalt[6]. 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 $U_o > 0.2$ [3]. These lighting requirements can be achieved using 700lx luminaires with a half power angle of 60° ($\alpha = 1$) placed every 15 meter in between the road and side walk. This set-up is visualized in figure 3.5(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 the hallway scenario. 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 objects were modelled with diffuse reflection, because no reliable sources describing the reflection parameters (r_d and α) of cars could be found.

Lacking the specular and spread reflections for this specific model should not influence the results significantly, as 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 3.5(b).

3.5 Results

Several simulated measurements have been graphed in Figure 3.6. All plots can be observed with the tools stored in appendix A. The plots on the left side of Figure 3.6 show the best (most deviation from steady state) and worst case (least deviation from steady state) scenarios for the simulated situations.

In general we can state that the extremer the albedo, the better the bypassing object can be observed. A high albedo leads to a huge peak in the signal. A low albedo leads to a huge drop in signal as most of the light otherwise bouncing back to the source, is now absorbed by the object itself. Another thing which can be observed is that the smaller the y distance, the more better the signal can be observed.

Figures 3.6(b), 3.6(d), 3.6(f) and 3.6(h), show the frequency spectrum of the received signals. They where obtained by calculating a Fast Fourier Transfrom (FFT) over the signal with an F_s (sample rate) by assuming a constant movement speed, $5km/h$ for humans and $30km/h$ for cars, over the observed area. These plots show that the frequencies, carrying the signals, lie between 0 and 2Hz for all simulated cases.

3.6 Conclusions

In this chapter, a version of the Phong model was implemented, verified and used to estimate the light response if a person or car would move past a light/photo diode pair. These responses gave several insights:

- The extremer the albedo, the better an event can be observed.
- The bigger the object, the better it can be observed.
- The smaller the distance to the light, the better the bypassing object can be observed.
- The expected frequency of the signal lies between 0 and 2Hz for both the street and the hallway scenario, no matter what properties the object has.
- Building one device which works in both scenarios will be hard, as the expected reflection strengths lie far apart.

The insights obtained with this model will be used in several places later on in the thesis.

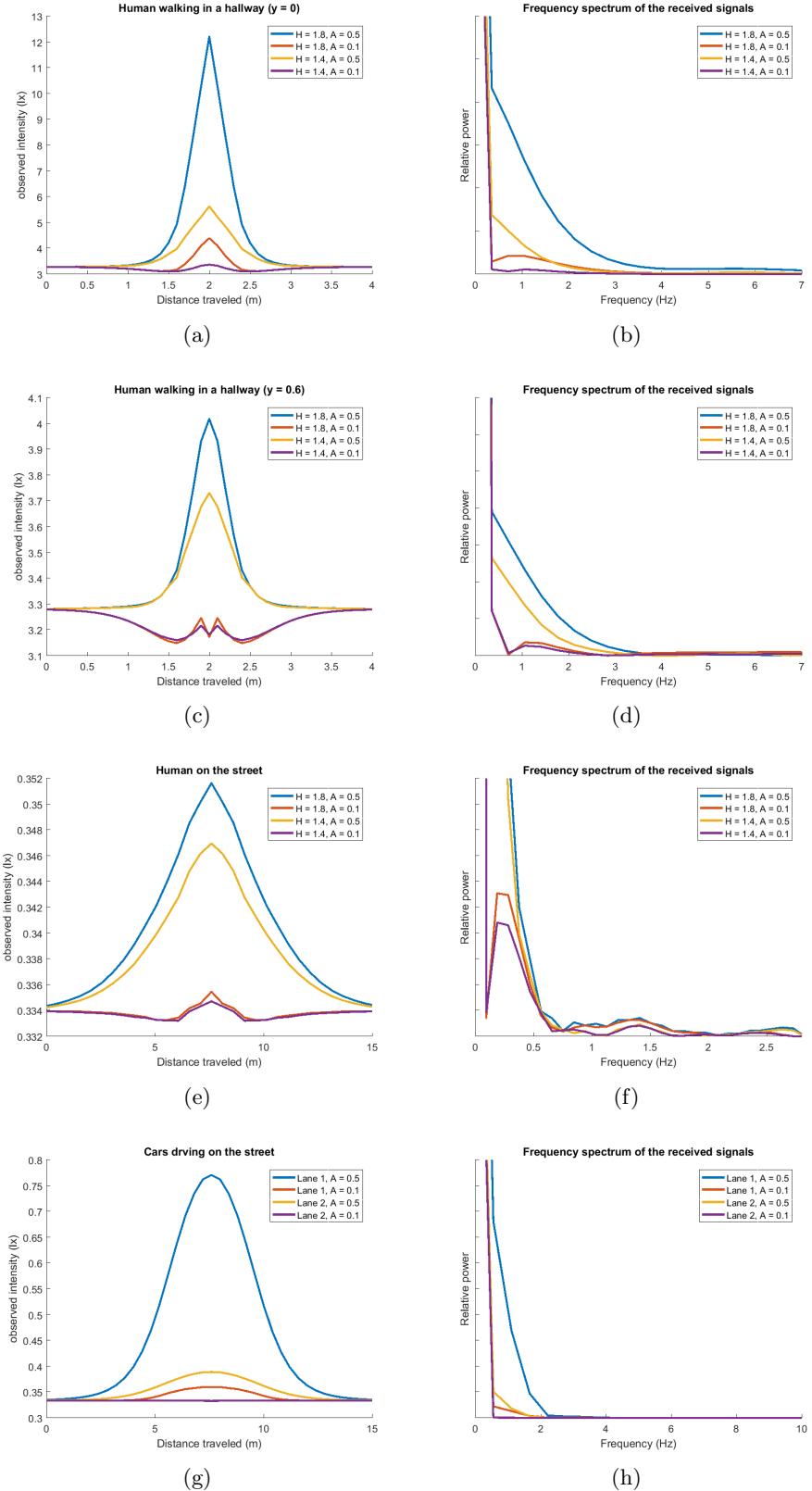


Figure 3.6: Several selected simulated responses. The left figures show the biggest and smallest responses. The right figures show the frequency spectrum of those signals.

Chapter 4

Platform

A device has been made to generate, receive and analyse flashes. The complete system architecture is shown in figure 4.1. Each component and their interfaces will be discussed briefly, followed by a section describing the final build of the platform.

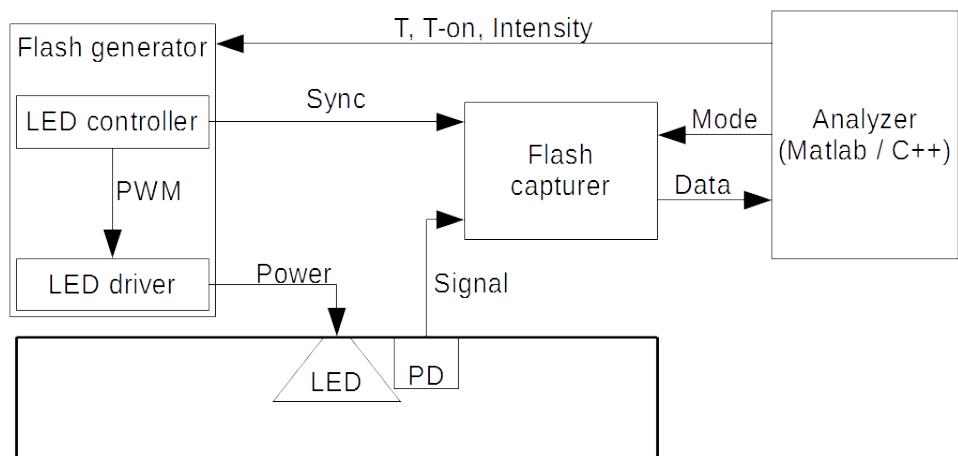


Figure 4.1: System architecture of the flash generator/analyser.

4.1 system components

4.1.1 Flash generator

The flash generator is a device able to control a LED with high precision. It's 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 4.1. This signal is sent to a LED driver through one of three LED

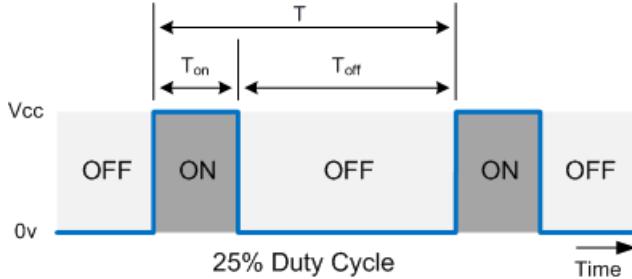


Figure 4.2: Visualization of how T and T_{on} determine the duty cycle and frequency of the flash generator.

drivers, which will make the actual light turn on and off at different light levels.

$$T = \frac{1}{f} \quad \text{DutyCycle} = \frac{T_{on}}{T} * 100\% \quad (4.1)$$

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.

4.1.2 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 should therefore capture flashes as precise and consistent as possible. For this reason, the receiver receives a sync signal from the flash generator and is therefore able to start sampling at almost 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.

4.1.3 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 these flashes. This mode is used in chapter 5 to analyse single flashes in order 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 6 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.

4.2 Implementation

The system was build by combining several of shelve parts. An overview of the actual build can be seen in figure 4.3. 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 chapter 3. The power used by the LED is regulated with a single resistor. The resistors were chosen after some experimentation with the flash generation and reception. The values and resulting LED current can be seen in equation 4.2.

$$I_{LED} = \frac{V_{DD} - U_{LED}}{R} \quad I_{LED} = \frac{7 - 3.6}{[1, 3, 5]} = [3.4A, 1.1A, 0.68A] \quad (4.2)$$

The reflection receiver is implemented on the shine platform [8]. 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 (for a total of $\pm 240\mu s$) when the sync signal is received.

A downside of the shine platform is that it's unable to communicate directly with the analyser as it does not have 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 were 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 performs in an environment with

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

Table 4.1: Overview of photo diode configurations.

lots of electromagnetic radiation. An overview of the PD configurations can be seen in table 4.1.

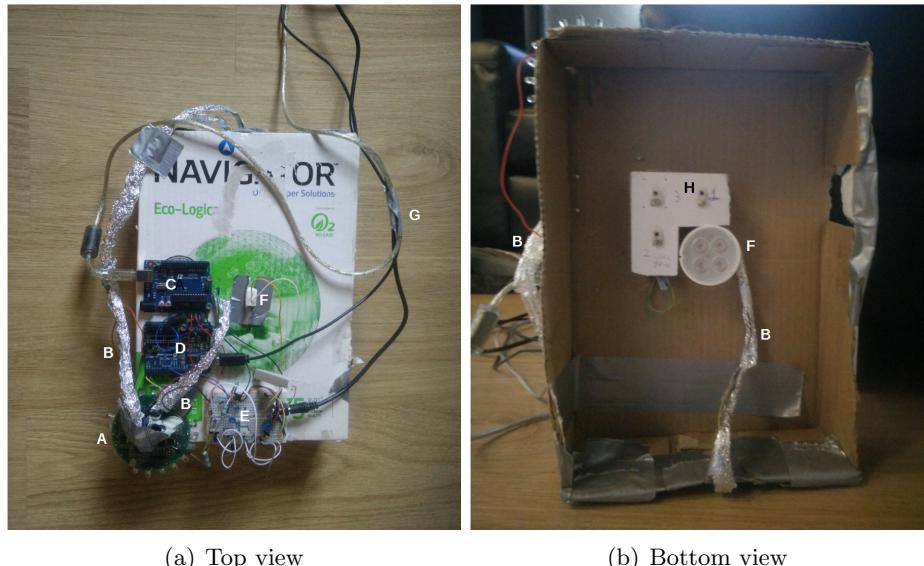
Another important decision concerns the amplification circuit of the photo diode. The original circuit used by shine features an analogue low-pass filter to remove ripple introduced by the amplifier (See Figure 4.4). The filter has several side effects. It reduces the signal strength and decreases the time the signal is visible to the system. It was therefore chosen to remove all analogue filters from shine and deal with the ripple with the help of software if required. The ripple effect might even be useful as it's probably dependent on the received signal and therefore a measure of light in the environment.

4.3 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 4.4(a).



(a) Top view

(b) Bottom view

Figure 4.3: 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

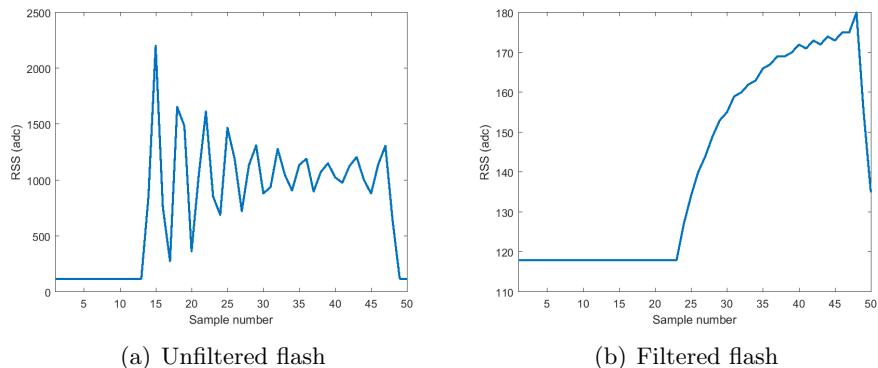


Figure 4.4: Two flashes captured with the platform. Left is unfiltered, right is filtered.

Chapter 5

Flash Analysis

The goal of this chapter is to find a method, capable of obtaining consistent measures of the environment from measured flashes. Secondary goals are to achieve this with the shortest flash and while using a small amount of computation power.

In this chapter, the platform is set-up as seen in figure 5.1. 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 for obtaining a measure of the environment from flashes will be presented and compared. This chapter will conclude with the final settings used in the flash generator and an algorithm to obtain a consistent measure of the environment from the received flash.

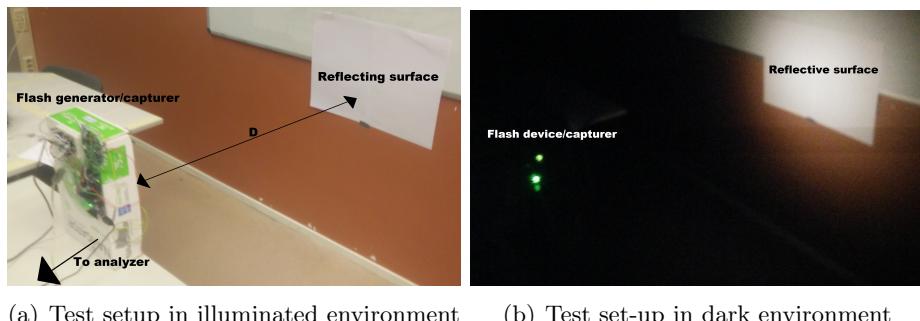


Figure 5.1: Test set-up used to capture flashes in the darkroom.

5.1 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.

Figure 5.2(a) shows several responses for different T_{on} . The figure shows 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 ending of the flash is not used.

Figure 5.2(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 [7]. This means that if a lower LED current is used a bigger T_{on} is required to obtain useful information.

Figure 5.2(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 decreases. This is logical, as when light travels longer distances, the relative intensity of the light decreases.

Figure 5.2(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 in combination with I_3 . Another noticeable change is the ripple frequency caused by the amplifier. This change is expected, as the resistor in the feedback loop of the amplifier was changed.

5.2 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.

5.2.1 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

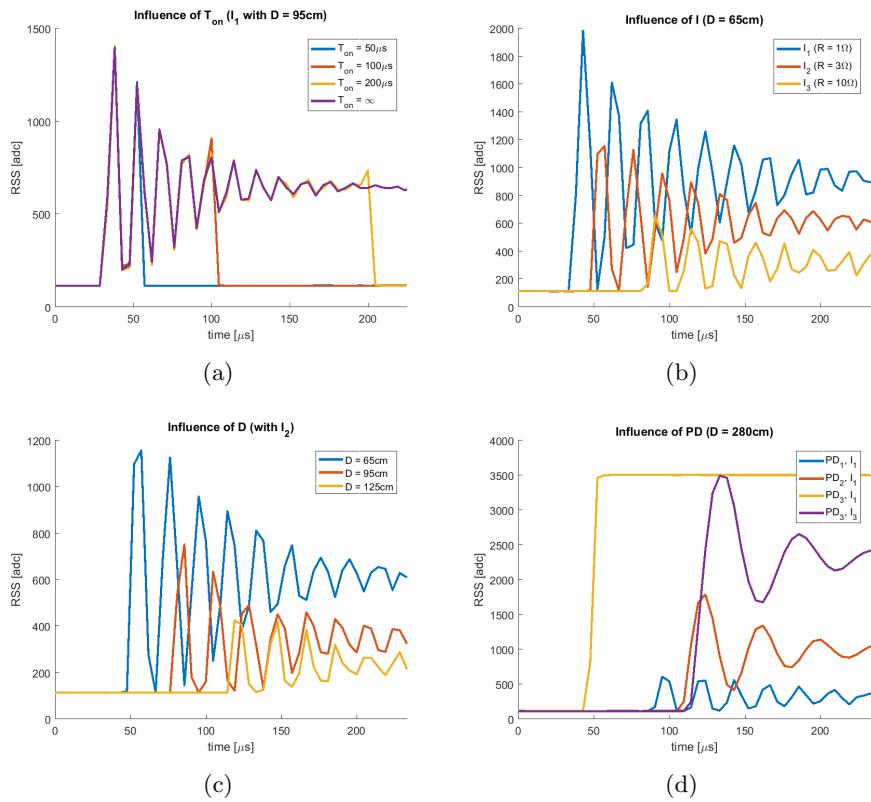


Figure 5.2: Several perceived flashes generated with different settings of T_{on} , I_{LED} , D and PD .

relative to the light turning on. If that's 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 ripple frequencies of the other amplifiers

Another method considered is to use the surface underneath the flash. This method has the advantage of being both simple and flexible. It does not matter if T_{on} is chosen big or small, it will always give a reliable result if T_{on} is not changed. 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 calculate 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.

5.2.2 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 now 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 5.2. This equation calculates ratio between the standard deviation 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.

T_{on}	SNR: PD_2, I_1			SNR: PD_3, I_3		
	150 μs	200 μs	250 μs	150 μs	20 μs	25 μs
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 5.1: Overview of the test results.

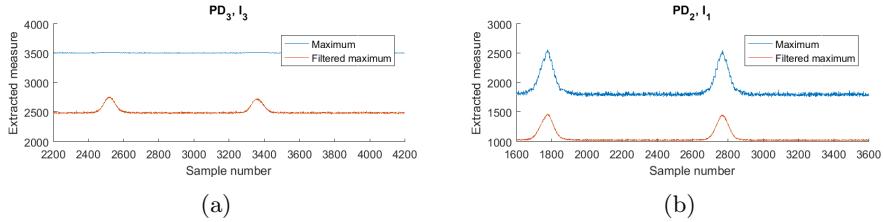


Figure 5.3: Data extracted using the maximum and filtered maximum methods with $T_{on} = 250\mu s$.

$$SNR(PD) = \left(\frac{\mu(PD_{NoEvent}) - min(PD_{event})}{\sigma(PD_{NoEvent})} + \frac{max(PD_{event}) - \mu(PD_{NoEvent})}{\sigma(PD_{NoEvent})} \right) \quad (5.1)$$

$$SNR(PD) = \frac{max(PD_{Event}) - min(PD_{Event})}{\sigma(PD_{NoEvent})} \quad (5.2)$$

The test was done with all combinations of PD and I . Only the combinations of PD_2, I_1 and PD_3, I_3 gave potential usable results at full scale as for other combinations the flash was invisible or too bright (saturation) to observe. Several consecutive captured features can be seen in the Figures 5.3 and 5.4. These were then used to calculate the SNR for each scenario. An overview of all calculated SNR values can be seen in table 5.1.

Based on the results of the SNR test it was chosen to use the Filtered sum method with $T_{on} = 200\mu s$. This method gives better results than the maximum and filtered maximum methods because more measurements are used to determine the final value, leading to lower standard error. The reason that this method works better than the sum method lies in the fact that the filtered signal is better representation for the environment that the rippled signal. This can also be seen in the difference between the maximum and filtered maximum.

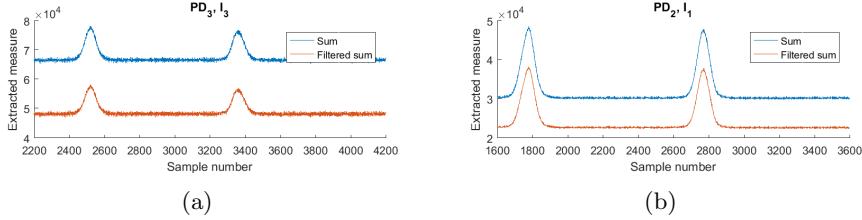


Figure 5.4: Data extracted using the sum and filtered sum method with $T_{on} = 200\mu s$.

5.3 Flash period

The final parameter to decide is the period of the signal, T . This value has no influence on the calculated SNR. It has however a clear influence on how much light is used by the system, as decreasing T directly increases the amount of flashes which occur. 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.

For these reasons, T was chosen to be $800\mu s$ resulting in a flash frequency of 125Hz. This value is more than the Nyquist frequency of the 50Hz and should therefore be filterable by the system. 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 .

5.4 Conclusion

The Flash analyser will run at a frequency of 125Hz, a T_{on} of $200\mu s$ with maximum light intensity I_1 using the filtered sum method with PD_2 to extract information from the reflections. These settings provide the best found signal to noise ratio with the given platform. The next step for the project is creating an algorithm for the analyser, capable of analysing a set of consecutive flashes.

Chapter 6

Analyser

The flash analyser now outputs values at 125Hz, which is a mixture of various light and noise sources. 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 its 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 manages detect every bypassing person correctly, but it only triggers a detection when the object 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 were 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 shows what considerations were made to determine threshold of the binary classifier.

6.1 Received signals

In an ideal world, the dark sensing system only perceives light it emits itself, reflected by the environment. Figure 6.1 shows that this is clearly not the case. Several other factors are influencing the measurements. Equation 6.1 has been devised and contains the most common signals the photo diode,

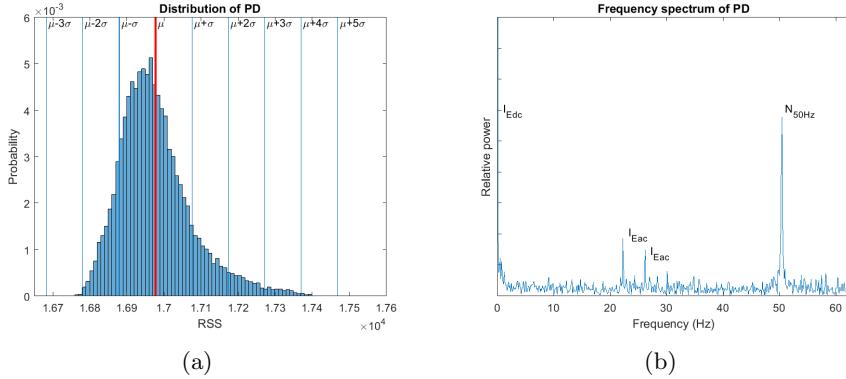


Figure 6.1: Properties of the obtained signal with the the filtered sum method.

PD , might receive. Each term of the equation will be discussed briefly while pointing out what the signal looks like.

$$PD = I_L \alpha + \sum_{i=1}^n I_{Edc_n} \beta_n + \sum_{i=1}^n I_{Eac_n} \gamma_n + N_{50Hz} + N(\mu, \sigma^2) \quad (6.1)$$

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 0.2Hz and 2Hz for by passing pedestrians, as shown in chapter 3. 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

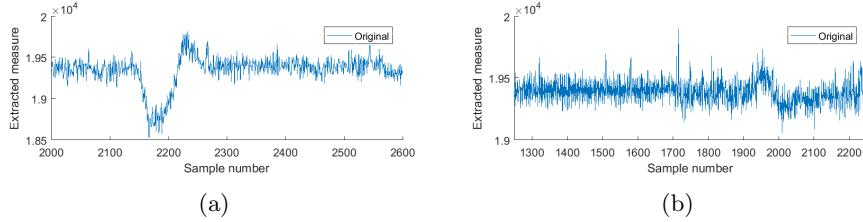


Figure 6.2: Two signals of a person walking underneath the set-up. Figure (a) shows an optimistic case with an SNR of 14.9 and (b) shows a harder scenario with an SNR of 10.5.

environment. The exact distribution of the noise is unknown, but its likely to approximate a Gaussian curve. It is therefore represented by its mean (μ) and variance (σ^2) in the equation.

6.2 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 6.2. (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.

6.2.1 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 6.3 and is a second order IIR lowpass with its corner frequency at 5Hz. Using 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 guarantee 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 than twice the frequency of 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 6.2, where n is the closest integer multiple of F_s to the signal being aliased (F_{Iac}).

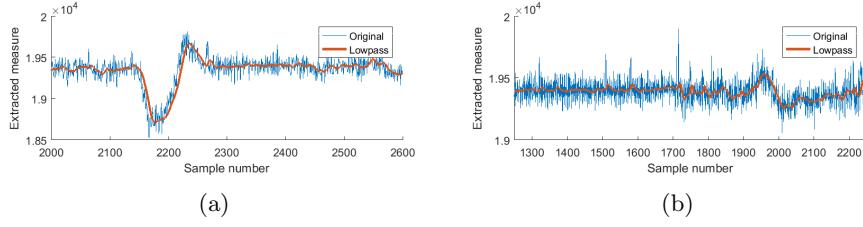


Figure 6.3: A lowpass filter ($F_{cutoff} = 5\text{Hz}$) applied to the two example signals used to remove $I_{50\text{Hz}}$ and I_{Eac} . The SNR for signal (a) increased to 27.9 from 14.9 and the SNR of signal (b) increased to 11.7 from 10.5.

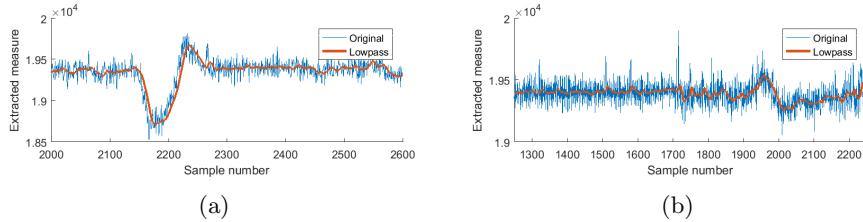


Figure 6.4: A highpass filter ($F_{cutoff} = 0.01\text{Hz}$) applied to the two example signals used to remove I_{Edc} from the signal. The SNR for signal (a) increased to 30.9 from 27.9 and the SNR of signal (b) increased to 13.3 from 11.7.

$$F_{alias} = |F_s * n - F_{Iac}| \quad (6.2)$$

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 100z or 200Hz (frequency of the power grid) and will therefore show up with alias frequencies of 25Hz and 50Hz. These frequency is can still be removed with the used low-pass filter. There is however no guarantee that all lights will blink at a multiple of 50Hz. In the Americas for example the grid is powered at 60Hz. The chance 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 dealt with in another way (if it occurs).

6.2.2 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. It works, but it takes the filter a long time to settle if the F_{Cutoff} is chosen too low. Several filters have been tested, and the final result applied to the two test signals is shown in figure 6.4.

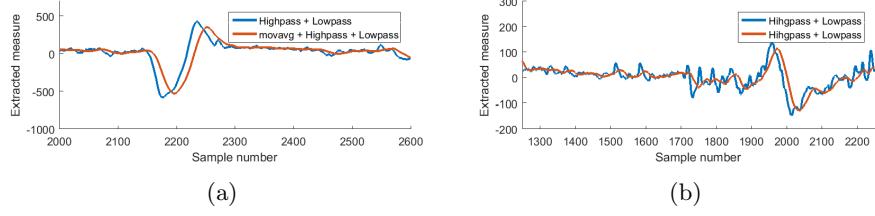


Figure 6.5: A 28 tabs moving average applied to the two filtered example signals used to remove the remaining 4.5Hz component of the signal. The SNR for signal (a) increased to 35.6 from 30.9 and the SNR of signal (b) decreased slightly from 16.6 to 13.3.

6.2.3 Moving average filters

A moving average can be used to reduce $N(\mu, \sigma^2)$ and the remaining F_{alias} . 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 $n = \left| \frac{F_s}{F_{alias}} \right|$.

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 costs 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} is detected and/or has changed.

Another advantage the moving average brings is it reduces the noise 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 6.3. The presented formula calculates n , so that the

This method has one 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 summations, 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.

$$\frac{\text{signal}}{\text{noise}} = 1 = \frac{\mu * ss}{T * \frac{\sigma}{\sqrt{n}}} \Rightarrow n = \left(\frac{T * \sigma}{\mu * ss} \right)^2 \quad (6.3)$$

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}	Takes a long time to settle if a step is received	Yes
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 computational intensive	No
$PD - PD_{dark}$	Filter I_{Eac} , N_{50Hz} and I_{Edc}	Only worked in illuminated environments which made it unreliable to filter N_{50Hz}	No

Table 6.1: Overview of the filter methods described in this section.

6.2.4 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 6.5.

$$PD_{dark} = \sum_{i=1}^n I_{Edc_n} \beta_n + \sum_{i=1}^n I_{Eac_n} \gamma_n + N_{50Hz} + N(\mu, \sigma^2) \quad (6.4)$$

$$PD - PD_{dark} = I_L\alpha + N(0, \sigma^2 + \sigma_{dark}^2) \quad (6.5)$$

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.

6.2.5 Filter overview

Several methods for removing unwanted parts of the received signal have been presented and summarised in table 6.1 and result in a better SNR. The final solution implements the lowpass, highpass and moving average with scaling based on F_{alias} . Figure 6.6 shows the remaining distribution of the noise on the signal.

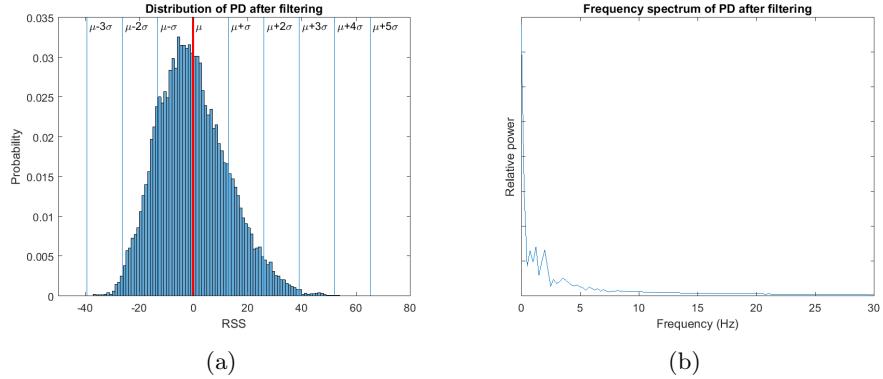


Figure 6.6: Properties of the signal after filtering. Note that (b) is zoomed in on the frequency axis compared to Figure 6.1(b)

6.3 Detection threshold

The next step in creating the binary classification algorithm is to determine classifying thresholds or rules. If an extracted value of the set of samples crosses the threshold, then the set gets classified as activity detected. A naive solution to the threshold 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 a "dirty" device in the environment turns on then this could increase the noise level in the environment. which would lead to false detections.

This failed method shows that the threshold needs to be adaptable based on the noise in the environment. Two other algorithms have been considered, which are possibly capable of adjusting their thresholds in an intelligent manner.

6.3.1 Standard deviation based threshold

The first method is based upon the standard deviation. We could set the detection threshold based on the current measure on σ . Several real-time algorithms are known to calculate rolling σ , the standard deviation over all samples which have passed by. This measure gives a good estimate of the noise in the environment until something happens. If an object passes by for example, an "extreme" change in the signal will occur and the rolling standard deviation will be deluded with non-noise samples, ruining the noise estimate.

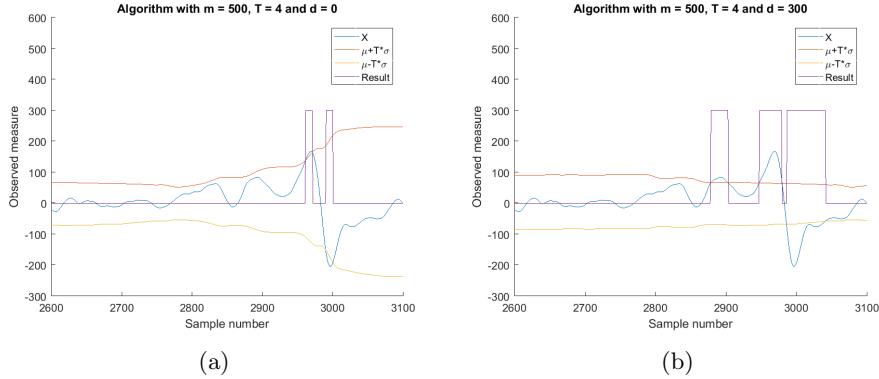


Figure 6.7: Working of the algorithm visualized with and without delay (d) between the standard deviation (σ) and the signal (X).

This issue can be solved by using a moving standard deviation. This method only uses the most recent m samples to obtain σ and bases its detection threshold on this value. This means, that if an event occurs, the algorithm only remembers it for m samples. By using a system with only a short-term memory, we can create a flexible system which automatically adjust to permanent changes in the environment.

Using the idea of moving σ , we can create a simple threshold algorithm with equation 6.6. This equation calculates the z-score for the most recent sample out of the filter section $\bar{X}[i]$ and compares it with threshold value T . Then if z is greater than T or smaller than $-T$, it triggers a detection. σ_i and μ_i in this equation represent the a moving mean and standard deviation over the most recent m samples at i .

$$z_i = \frac{\bar{X}_i - \mu_i}{\sigma_i} \quad (-T < z_i < T) \rightarrow \text{detection} \quad (6.6)$$

Figure 6.7(a) shows the described algorithm in action. It can be seen that the bypassing person is detected at sample 2962. This is a good result, but rather slow, especially since we are able to see the signal rise from sample 2875. The reason why the algorithm does not trigger a detection is because the threshold ($T\sigma$) has risen slightly due to the variance on being increased, which is caused by change in the signal itself. This problem can be solved in a very simple way. Prevent σ from being updated before an event happens. This can be achieved by adding a small delay of d samples between X and the threshold. An example of this is shown in Figure ??, where the event is detected at 2879 samples, which is 83 samples (0.66 seconds) earlier than the previous result.

Several other parameters were added to further improve the classification algorithm. The first one is to add a linear offset $k \cdot \sigma$ to the detection

threshold, because the noise distribution of the device is not balanced, meaning that there are more outliers on one side of the mean, than there are on the other side. This can be seen in Figure 6.6(a). If we were to move μ slightly more to the right with a factor 0.75σ , then the distribution would be more balanced (all samples lie between $\pm 3.5\sigma$) which results in a better overall sensitivity.

Another improvement we can make is to not look at only one observation of \bar{X} , but L consecutive ones instead. If we then only trigger a detection if l out of L samples cross the threshold then it might be possible to run the algorithm with a smaller T and therefore detect more events.

Equation 6.7 shows a mathematical representation of the threshold algorithm. A graphical overview of the complete algorithm (including the filter section) can be seen in Figure 6.8(a).

$$\sum_{i=-L}^0 \left\lfloor \frac{\bar{X}_i + k \cdot \sigma_{i-d} - \mu_{i-d}}{T \cdot \sigma_{i-d}} \right\rfloor \geq l \rightarrow \text{detection} \quad (6.7)$$

6.3.2 Variance based threshold

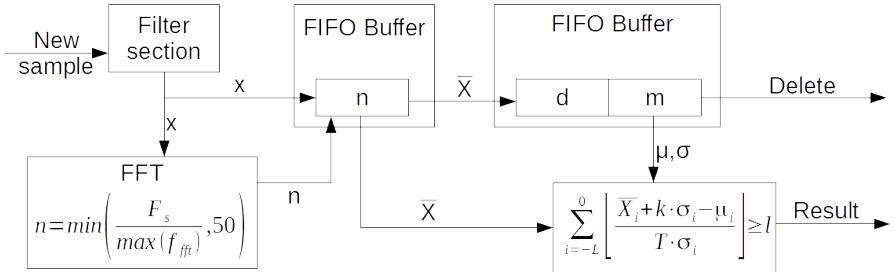
The other threshold method considered for classifying several consecutive samples is to only look at the variance of the signal. In Figure 6.7(a) it was observed that the standard deviation changes rapidly if an event occurs. If we wanted, we could simply calculate the variance (σ^2) of the signal and check if it exceeds a certain threshold. If it does, we trigger a detection. This method can be summarized with equation 6.8.

$$\sigma_i^2 > T \rightarrow \text{detection} \quad (6.8)$$

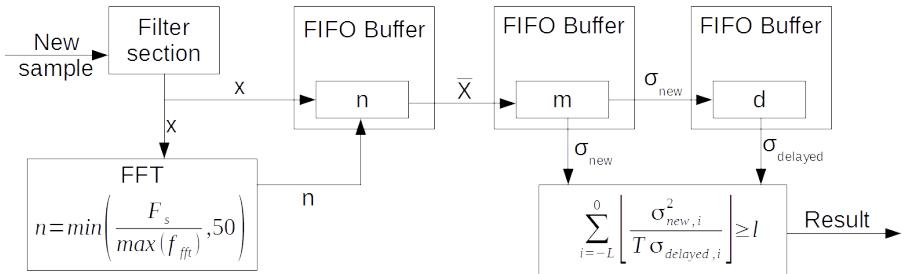
This method has similar problems as previously described methods:

- A static threshold, T , is unable to cope with changes in the environment noise.
- A rolling variance as threshold would be deluded by the events, and therefore be deceptively high.

Similar problems ask for similar solutions. It was therefore chosen to use a moving variance over m samples as measure and a delayed moving variance, delayed in time by d samples, as threshold. This results in a variable threshold with a short-term memory which is able to deal with adjustments in the environment. This method can be improved even further by adding the variables l and L as discussed in the previous subsection. By checking l out of L samples cross the threshold, we can afford to use a lower T and detect even smaller changes. A mathematical representation of this algorithm is given in equation 6.9. A graphical representation of the full algorithm



(a)



(b)

Figure 6.8: The two versions of the algorithm. (a) shows the absolute version and (b) shows the standard deviation version.

(including filters) is shown in figure 6.8(b).

$$\sum_{i=-L}^0 \left| \frac{\sigma_{new,i}^2}{T \cdot \sigma_{delayed,i}^2} \right| \geq l \rightarrow detection \quad (6.9)$$

6.4 Conclusion

Chapter 7

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.

7.1 Measures of evaluation

The system will be evaluated on three different criteria: Precision, recall and response time. Precision is defined in equation 7.1 and gives us insight in how many situations the light would turn on unnecessarily. Recall is defined in equation 7.1 and gives us insight in how often the light fails to turn on when an object passes by.

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \quad (7.1)$$

The precise response 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.

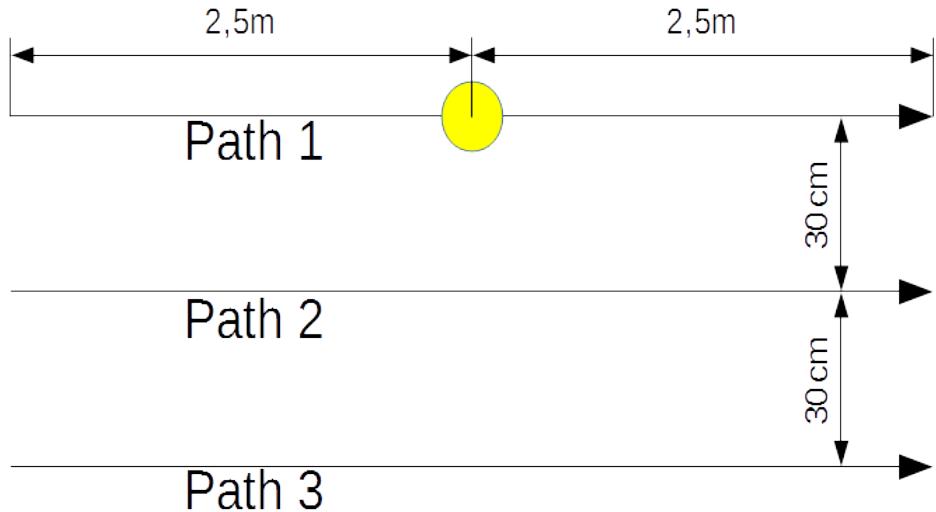


Figure 7.1: The three paths the test subjects had to walk underneath the light.

	d	m	l	L	T	k	P_{set1}	P_{set2}	$P_{set1,2}$	R_{set1}	R_{set2}	$R_{set1,2}$
Heuristic												
Evolution set 1	539	654	4	4	3.5	0.1						
Evolution set 2												
Evolution set 1, 2	475	714	4	4	4.5	0.8						

Table 7.1: Results of the hallway tests

7.2 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.

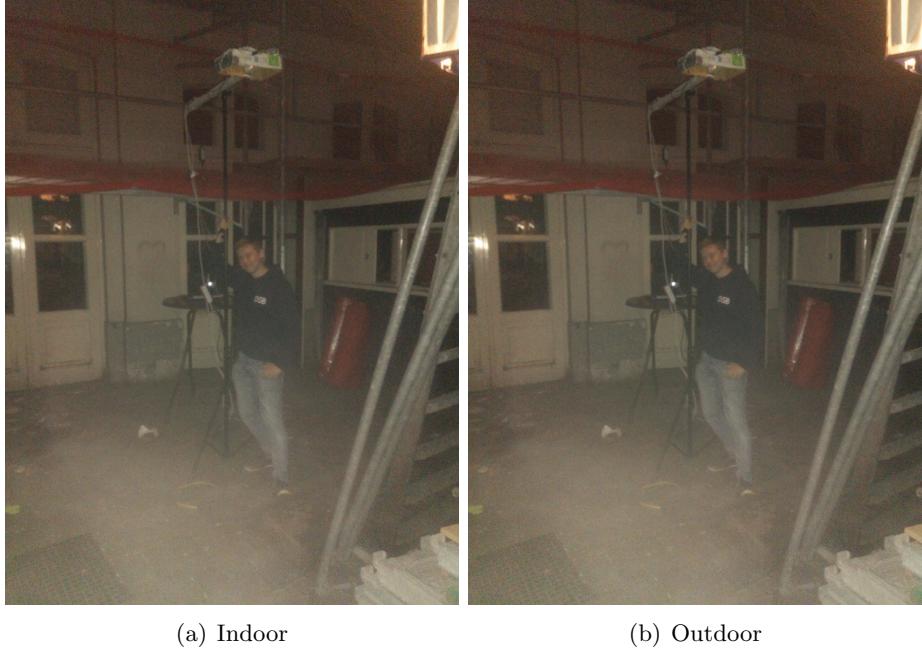


Figure 7.2: Pictures of the test set-up indoor (a) and outdoor (b). The light was suspended at 2.6m high.

7.2.1 Test set-up

7.2.2 Results

7.3 Street

This section evaluates the system on detecting cars driving by on a road with a scale model. It will first explain the test set-up and how it was scaled. It then explains the test procedure, followed by a section showing the ideally found algorithm settings and their detection results.

7.3.1 Test set-up

The street scenario is tested with a scale model. The dimensions of the scale model were determined with equations X to Y. Equation 7.2 represents an approximation of the strength of the received signal. It

$$PD_{fullscale} = \frac{E_{hor} * S_{car}}{H^2} = \frac{3 * 1}{7^2} \quad (7.2)$$

$$PD_{scale} = \frac{I_L * \frac{1}{39}}{H^2} \quad (7.3)$$

7.3.2 Results

7.4 Conclusion

Chapter 8

Conclusions and Future Work

8.1 Conclusions

TODO CONCLUSIONS - It works with crappy hardware -

8.2 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 take 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

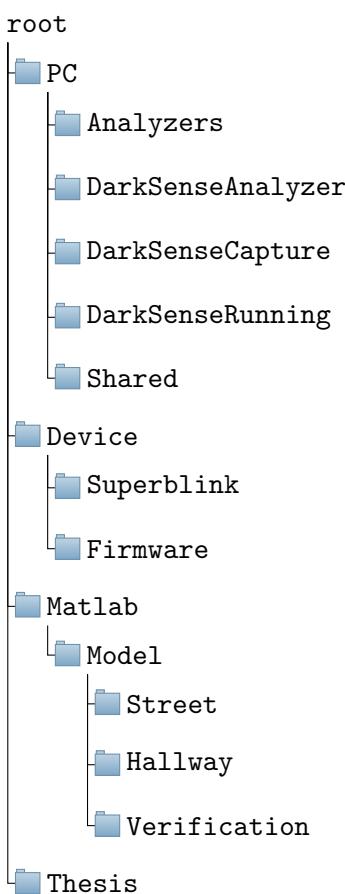
- [1] Marilyne Andersen. Types of reflection. <https://www.flickr.com/photos/mitopencourseware/4815499473/>.
- [2] Arduino.cc. <https://store.arduino.cc/arduino-uno-rev3>.
- [3] Lilian Tieman Chris van der Nat, Hans Schelvis. *Bestaande bouw, Handboek Politiekeurmerk Veilig Wonen*. Centrum voor Criminaliteitspreventie en Veiligheid, december 2015.
- [4] Thiemo Voigt Carlos Perez-Penichet Elena Di Lascio, Ambuj Varshney. Loc-alight - 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.
- [5] A. Galvin and J.K. Guscott. Passive infrared detector. Google Patents, March 1982. US Patent 4,321,594.
- [6] 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.
- [7] H. Halbritter. High-speed switching of ir-leds (part i) - background and data-sheet definition. Technical report, OSRAM, Opto Semiconductors, March 2014.
- [8] 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.
- [9] Ikea 702.880.22 led1336r4. <http://lamptest.ru/review/ikea-70288022-led1336r4/>.
- [10] K. Iio M. Fukuda, H. Numakuraand and A. Hidaka. Human body detection system. Google Patents, March 1994. US Patent 5,369,269.
- [11] S. Bissig M. Waelchli and T. Braun. Intensity-based object localization and tracking with wireless sensors. January 2006.
- [12] 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.
- [13] Mastech. *MS8229, Digital multimeter operation manual*.
- [14] Jaakko Maentuasta Miika Valtonen and Jukka Vanhala. Tiletrack: Capacitive human tracking. *Pervasive Computing and Communications*, March 2009.
- [15] 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.

- [16] 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.
- [17] Marcos F. Guerra Medina Oswaldo Gonzlez and Inocencio R. Martn. *Advances in Optical Communication*, chapter Multi-User Visible Light Communications, pages 36–62. InTech, November 2014.
- [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] Junwei Zhang. Passive localization with visible light. MSc thesis, TU Delft, 9 2016.
- [20] 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.
- [21] 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.
- [22] 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 were added to each folder to explain what each file contains and is used for.



Appendix B

Raw Model results

This appendix contains the raw results of the model.

y	H	min	max	min	max	min	max	min	max
		$\alpha = 0.2$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.5$
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	max	min	max
		$\alpha = 0.06$	$\alpha = 0.06$	$\alpha = 0.14$	$\alpha = 0.14$
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	max	min	max
		$\alpha = 0.06$	$\alpha = 0.06$	$\alpha = 0.14$	$\alpha = 0.14$
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 4) 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

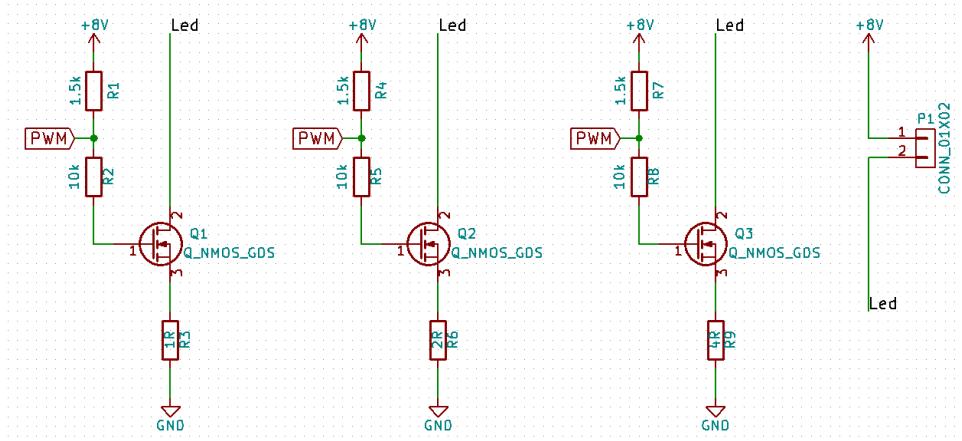


Figure C.1: Drivers used to drive the LED.

C.2 Interfaces between components

@TODO!