Delft University of Technology
Master's Thesis in Embedded Systems

Sensing human activity with dark light

Hajo Kleingeld

embedded
software

TUDelft Delft University of Technology

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

14th November 2017

Author
Hajo Kleingeld (hajokleingeld@gmail.com)
Title
Sensing human activity with dark light
MSc presentation
14th 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
14th November 2017

Contents

Chapter 1

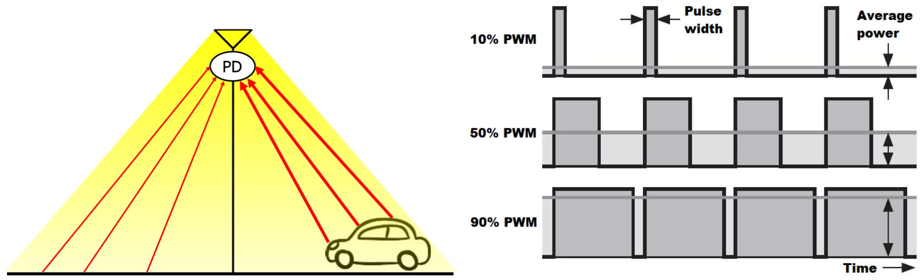
Introduction

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 1.

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 surface

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

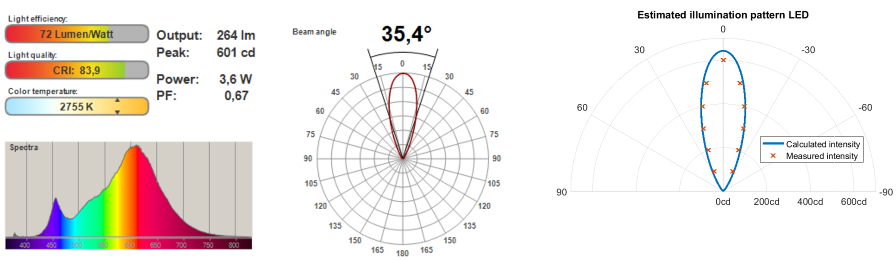
Table 2.3: summary of measures, units and symbols used in the algorithm section

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



(a) Specs of an LED

(b) Caclculated intensity pattern

Figure 2.1: Figure a shows measured specifications of an LED [7] where Figure b shows the estimated illumination pattern of the same LED.

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 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

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

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

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

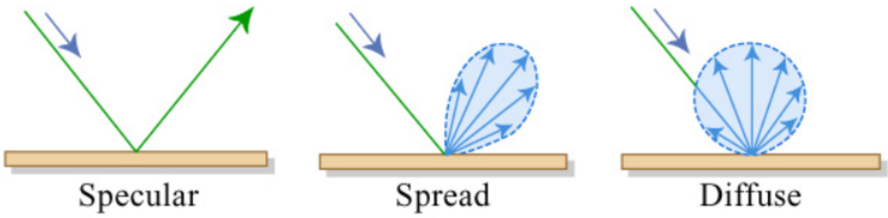


Figure 2.2: The possible ways for light to reflect when it hits a surface [1]

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

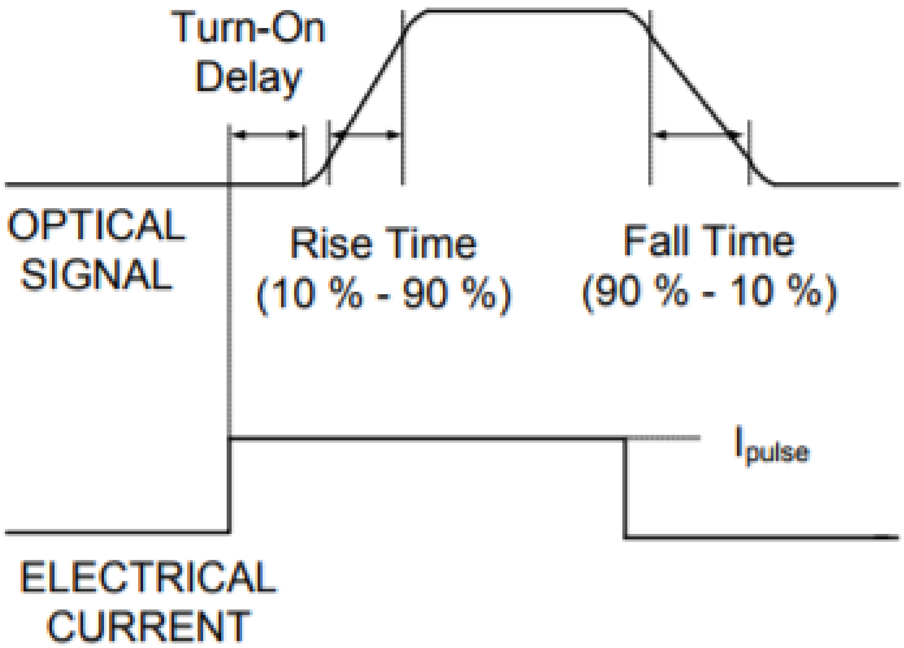


Figure 2.3: Realistic light response.

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)[13]. 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 3.1.

M. Valtonen *et al* created a system which passively tracks humans with the help of capacitive street tiles[11]. Byunghun Song *et al*[14] Smart street light system [8]

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 3.2 [4].

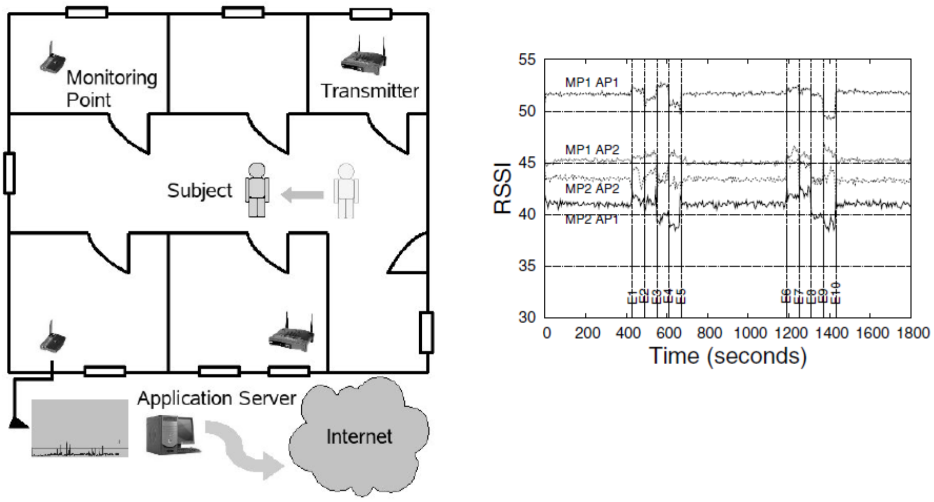


Figure 3.1: Overview of the WIFI tracking system of Moustsafa Youssef *et al.* [13]. The left figure shows an overview of the setup where the right figure shows the strength of the APs from the point of view of the MPs. E1 to E8 represent possible 'events' of bypassing persons.

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.

A 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 [16]. 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 3.2). These pixel-images are then used to reconstruct the original stance of the person scanned.

C. Zhang *et al* [2]

J. Zhang for example created a method for localizing and tracking objects with specular surfaces on a line[17].

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

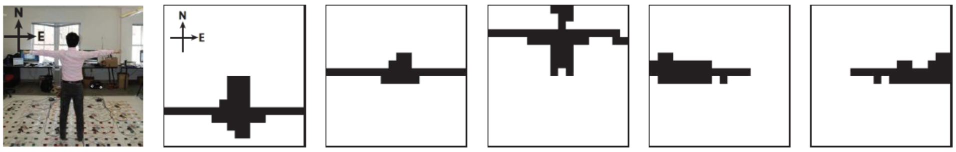


Figure 3.3: The scanned person with the resulting created pixel images.

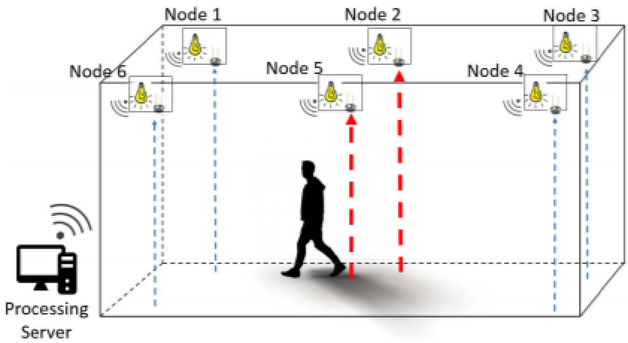


Figure 3.4: Overview of the system set-up of "Activity sensing using ceiling photosensors" project [12]. In this specific situation node 2 and 5 detect shadow caused by the person in the room, while other nodes do not.

lights and photo diodes hung at a ceiling[12]. 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 3.2.

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 [18] [19]. 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.

[16]

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[9]. This increase in accuracy

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

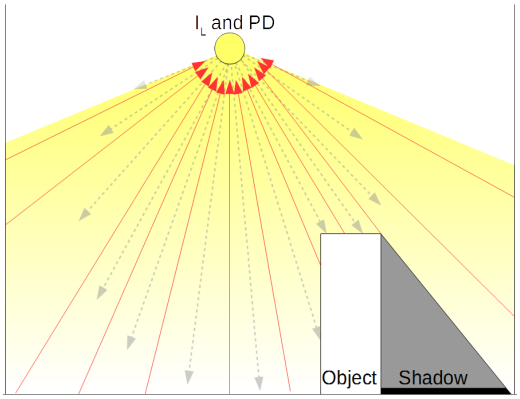
4.1.2 Calculation process

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.

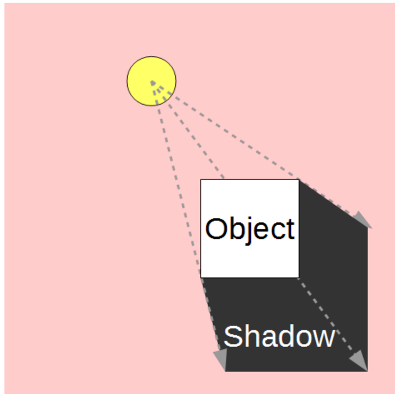
The second step is to calculate how much light reflected from all floors and walls (where no shadow is casted) to the photo diode. The finals step is to calculate how much light is reflected from each side of the object. 4.1.2 shows an overview of an environment with rays leaving the light, casting shadow and the resulting reflections.

4.2 Verification

The calculation method was verified using a scale model featuring a LED[7], a paper box and a light meter[10]. 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[15]. 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 $U_o > 0.4$[20]. 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

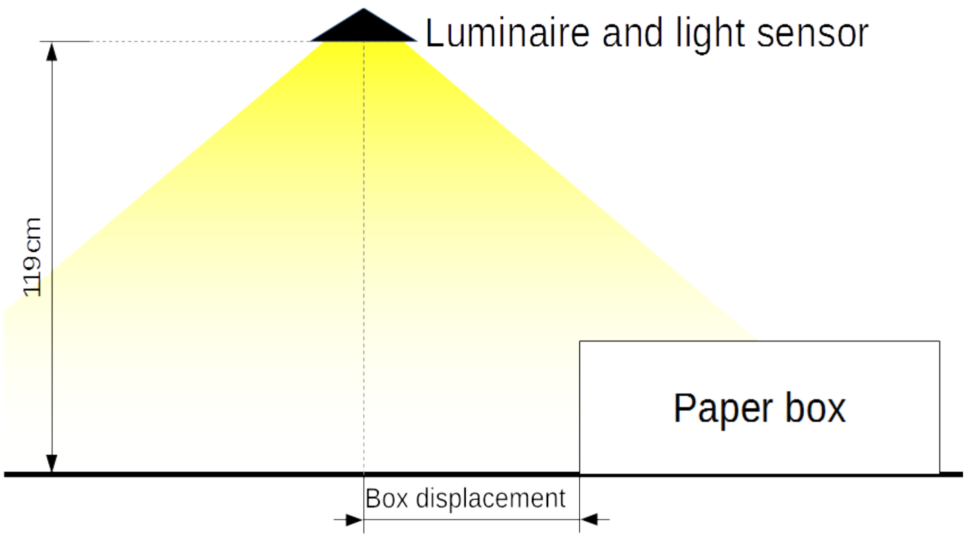
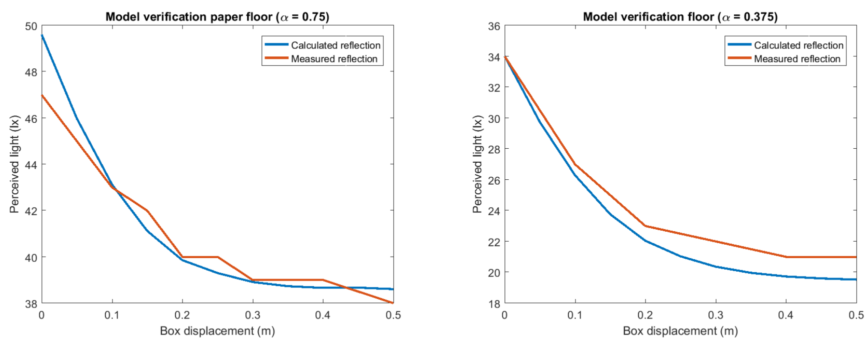
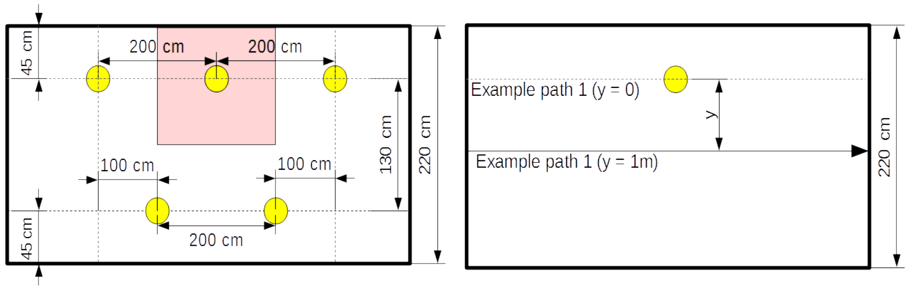


Figure 4.2: Visualisation of the model verification set-up.



(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 [15] 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).

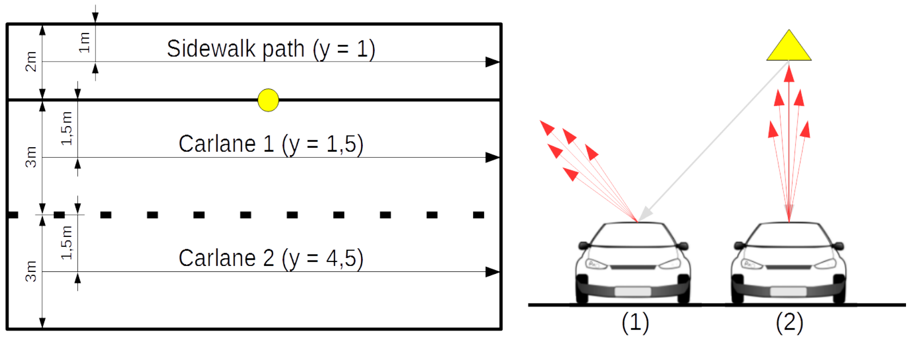
4.4 Modelling of the street

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[15]. 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 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.

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.

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} controls the amount of time the LED is turned on. Both parameters can be set with a resolution of $10\mu s$ resulting in a precisely controlled PWM signal. This signal is sent to a LED driver trough one

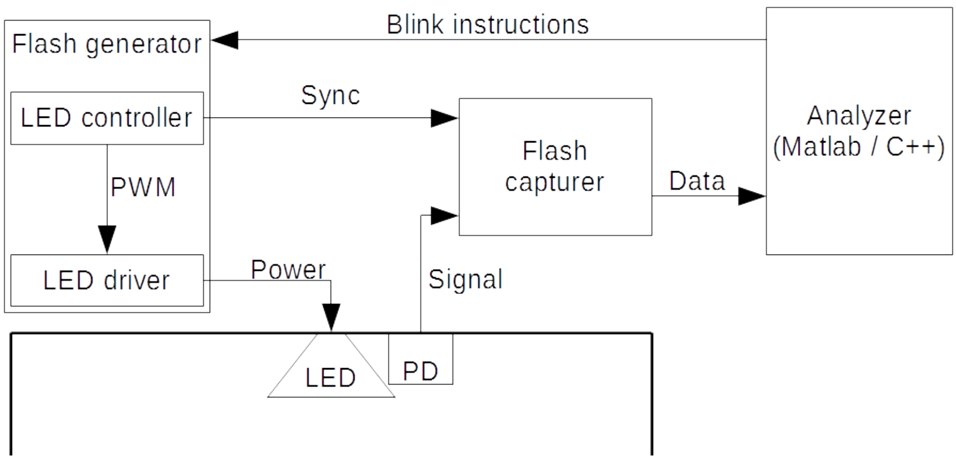


Figure 5.1: System architecture of the flash generator/analyser

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 flash and extract it's features. The receiver should start sampling when the sync signal of the flash generator arrives, until the light turns off. Then, the device should do one of the following things, depending on the system settings:

1. Send back the full flash, uncompressed, for analysis of separate flashes.

2. Send back all compressed flashes, by extracting several features.

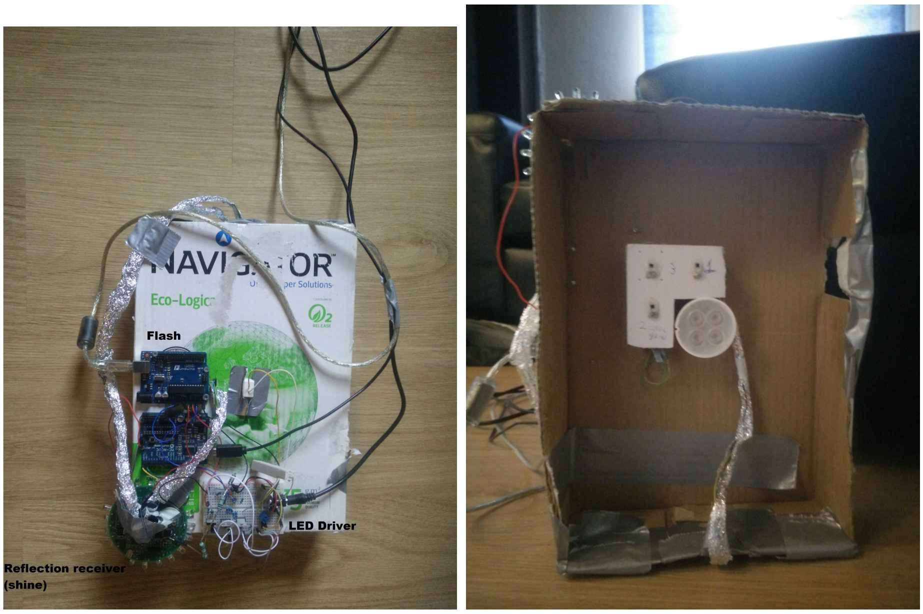
5.1.3 Analyser

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). If the receiver sends raw flashes to the anlyser it can be used to analyse this flash. This mode is used in section **??** 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 LED_{nr} to the device. This allows for real time control of the device.

5.2 Implementation

The system was build up with off the shelve parts. pictures of the acutal build can be seen in figure 5.2. This section will explain briefly how each system component is implemented.



(a) Top view

(b) Bottom view

Figure 5.2: The used platform in it's bodged glory!

Chapter 6

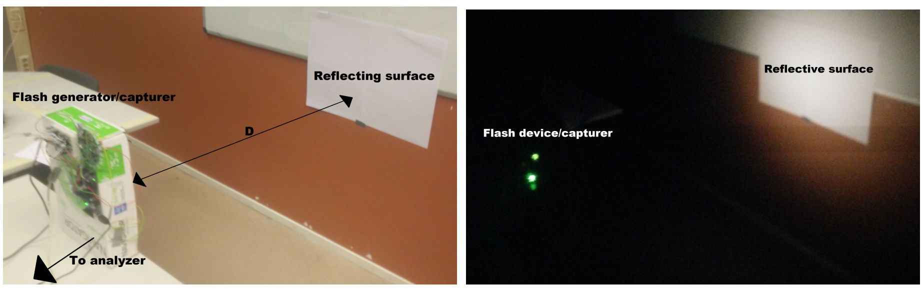
Flash Analysis

The goal of the flash analyser, as explained earlier, is to capture the flash and extract useful features. Before we are able to analyse flashes we have to find the ideal flash-generator settings. For this reason the first section of this chapter focuses on finding the ideal flash settings. The second section will focus on how and what kind of features to extract from the flash. The final section will show the variance still left in the signal.

6.1 Flash generator settings

The flash generator has several parameters which can be changed: T (period), t-on (on time of the LED) and I (brightness of the LED).

In order to find the ideal flash settings we use the test set-up shown in figure 6.1. D in the figure represents the distance between the device and the reflecting surface.



(a) Test setup in illuminated environment (b) Test setup in dark environment

Figure 6.1: Test setup used to capture flashes in the darkroom.

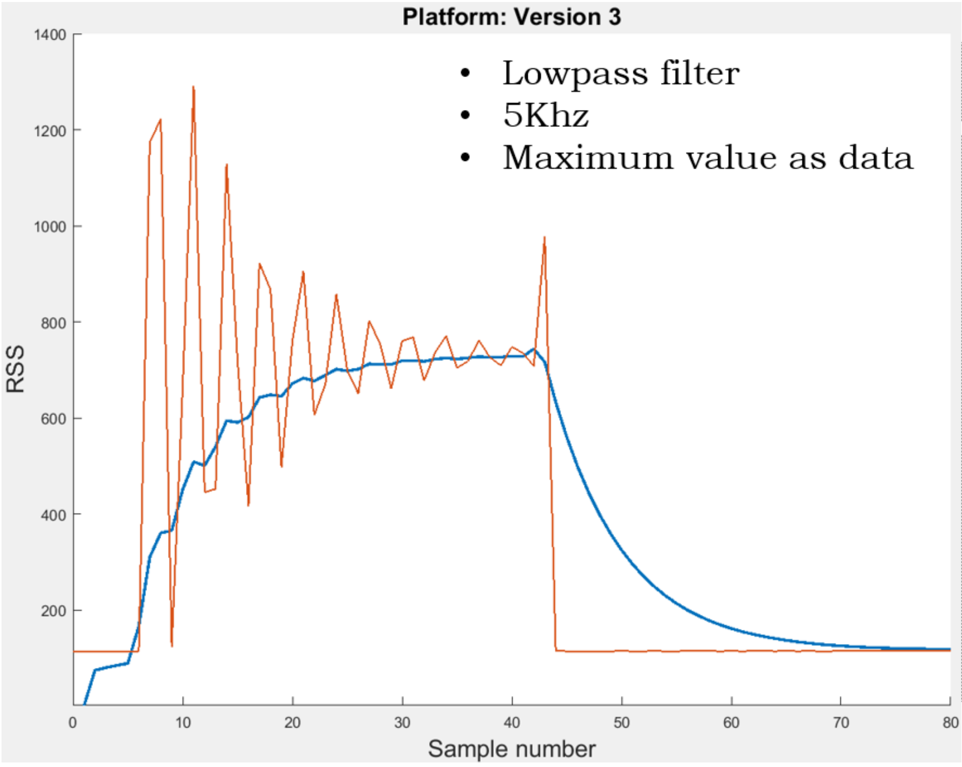


Figure 6.2: The data captured by the system before and after filtering. @=PLACEHOLDER FIGURE!@.

6.2 Feature extraction

6.3 Overview

show several captured flashes. and explain what we see. explain features extractable from the flash - maximum - minimum - surface under graph -

show real turn on delay time (or in our case, the time we are able to sense the light), rise-time, and turn-off-delay. Define turn on time.

6.3.1 Data generation

Show several filtered captured signals in one figure
point out the part where light becomes constant
pick that t-on time as minimum
Mention that a sample of darkness can be taken @ 80 samples.
That the number (80) can be reduced by picking a FIR filter instead (less ripple)

6.3.2 Results and conclusion

Show the result of several ”filtered flashes
Explain sources of the ”noise”, also show FFT and histogram of the noise
Explain that the noise could be reduced, but that this would cost a lot more processor time. (which is not recommended to run at 125Hz).
Note the noise is approximately Gaussian.

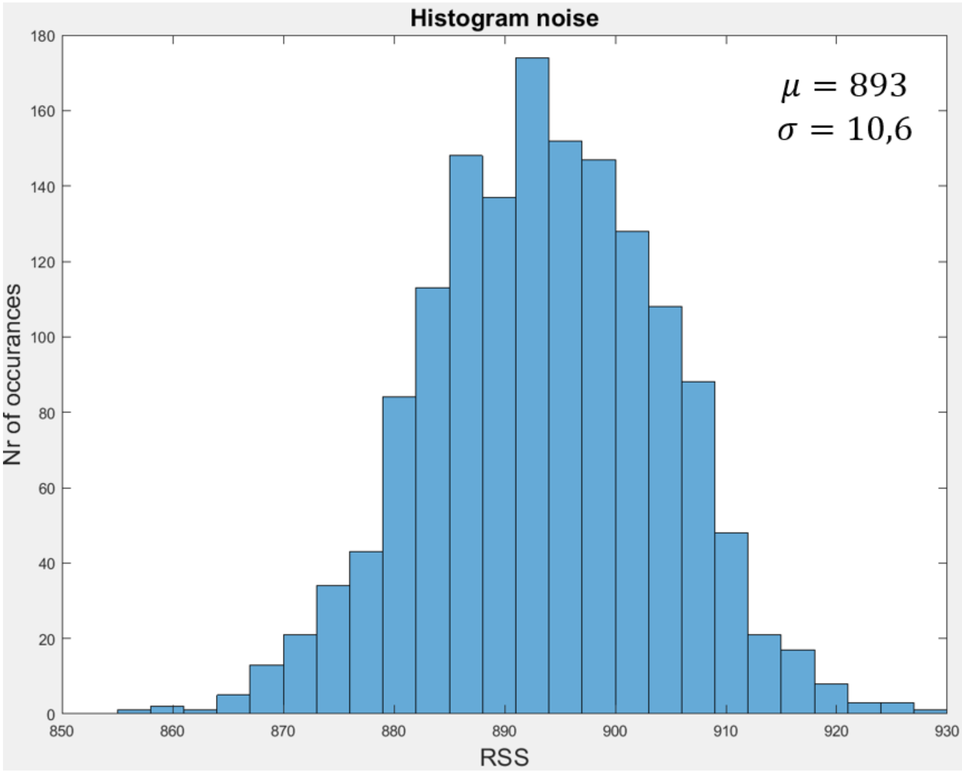


Figure 6.3: Histogram of the noise of the data sampled by the system. The distribution roughly follows a normal curve. @=PLACEHOLDER FIGURE!@.

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 an activity detection algorithm to convert the incoming samples in a logical value: Activity detected or no activity detected. This chapter is separated in three parts. The first part shows what signals are received by the photodiode. The second part explains what methods considered to remove unwanted signals from the signal. The final part of this chapter shows all considerations for the threshold algorithm.

7.1 Received signals

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

The goal of the complete algorithm is to isolate α and detect significant changes in real time. It might be possible to create a post-time algorithm with overall better detection rates, but that would be unusable for this usecase.

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.

7.2.1 Digital filters

Digital filters can be used to remove or at least reduce signals we are not interested.

highpass filters

Highpass filters can be used to remove I_{Edc_n} from the signal.

The highpass filter can ruin the the signal if chosen poorly.

Lowpass filters

Lowpass filters can be used to remove I_{Eac_n} and N_{50Hz} from the signal.

The Lowpass filter is unable to grantee the removal of I_{Eac_n} because of possible aliasing.

7.2.2 Moving average filters

A moving average can be used to reduce $N(\mu, \sigma^2)$ and the remaining F_allias.

7.2.3 Differential filter

The differential filter makes use of the signal received when the light is turned off. PD_{dark} can then be subtracted from PD remove several noise sources of the equation.

$$PD_{dark} = \sum_{i=1}^{n} I_{Edc_n}\beta_n + \sum_{i=1}^{n} I_{Eac_n}\gamma_n + N_{50Hz} + N(\mu, \sigma^2) \tag{7.2}$$

$$PD - PD_{dark} = I_L\alpha + N(0, \sigma^2 + \sigma^2_{dark}) \tag{7.3}$$

$PD - PD_{dark}$ does not work well with the current hardware set-up as the PD_{dark} is outside the range of the system.

7.3 Threshold determination

The goal of the threshold is to separate significant from insignificant changes in the signal. The challenge in creating a good threshold equation for this problem lies in the variety of signals the system has to deal with.

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.

7.3.3 Variance based threshold

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

7.4 Algorithm overview

The first step in creating a proper algorithm, is to identify what signals are affecting the measurements. For this reason, equation 7.1 was devised. It shows the composition of the received signal PD after it's down sampled to 125Hz with the method described in section 6. The equation for PD helps to understand what steps are required to create a reliable algorithm.

The first part of the equation, "$I_L\alpha$", is the amount of light generated by our luminaire (I_L), multiplied by some factor describing the environment from the point of view of the luminaire (α). This signal typically occurs in two ways. The first is an object passing by. The other way this signal can occur is if an object moves into range of the sensor, stops, and then stays still for a long time. This can be modelled as a sine wave, followed by constant, interrupting the wave. The wave is typically a low frequency signal between 0.25Hz and 2Hz (see section **??**). An example of this signal can be seen in figure **??**.

The second part of the equation, "$\boldsymbol{\sum_{i=1}^{n} I_{Edc_n}\beta_n}$", describes the impact of all constant light sources (I_{Edc_n}), multiplied by the factor describing the environment from their point of view (β). An example of a constant light source is moonlight. The final factor involving light in the equation is $\boldsymbol{\sum_{i=1}^{n} I_{Eac_n}\gamma_n}$. This describes all fluctuating light sources (I_{Eac_n}), multiplied once again by an environment describing factor from it's point of view (γ_n). This signal is typically 100Hz as most "old" lights blink or fluctuate at this frequency. Note that the sampling frequency is not twice as big as the sampled frequency, thus aliasing will occur at 25Hz ($= |F_s * 1 - F_{Analyze}| = 125 - 100 = 25Hz$)[6]. An example of these signals can be seen in figure **??**.

The last two terms in the equation have nothing to do with light, but represent noise from all other sources. $\boldsymbol{N_{50Hz}}$ represents specifically 50Hz noise. This is powerline-noise picked up by the physical wire of connecting the photo-diode to the amplifier. Normally, line-noise is barely noticeable, but as all the signals are amplified by a 1000 times, it becomes a significant disturbance. The final part of the equation, $\boldsymbol{N(\mu, \sigma^2)}$, is all the noise described in section **??**. An example of these signals can be seen in figure **??**.

The only part of equation 2.5 that holds information we can use is $I_L\alpha$, or more specifically, the changes of α. For this reason, each of the methods described in this section focus on removing the other parts of the equation, or making changes of α more detectable.

7.4.1 Removing high frequency components

Filtering higher frequencies from a signal is a common challenge in signal processing. The signals we try to isolate ($I_L\alpha$, 0.25 to 2Hz) are far removed

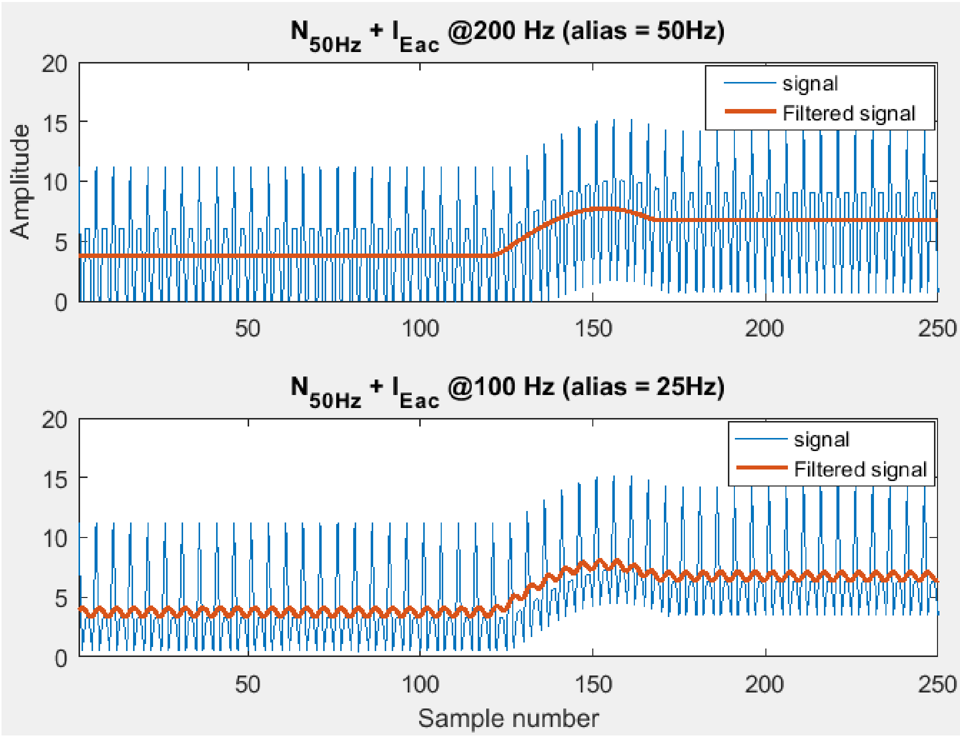


Figure 7.1: A low-pass filter, filtering the I_{Eac} and N_{50Hz} signals.

from the frequencies we are trying to suppress (I_{Eac} at 100Hz and N_{50Hz} at 50Hz). The most common method of doing this is by using digital filters. As only the lower frequencies are interesting, a low-pass filter seems to be ideal in this case. There is however one problem. $2 * I_{Eac}$ Is above our sampling frequency, and thus aliasing will occur. In this example it will appear as a 25Hz signal. This signal poses no real issue, as its still an order of magnitude away from the frequency of $I_L\alpha$, and it can still be filtered with a steeper filter. This however, is not always the case in the real world. Any light manufacturer can create lights running at any frequency, and thus there is no guarantee that there won't be a light out there in the world, that can mess the algorithm up. In all other cases this filter probably works fine. Figure 7.1 shows how a low-pass filter would remove the signals from the data.

A completely other method of removing these high frequency components is the "PD_{dark}" method. This method uses the fact that the sensor set-up can do more than just sampling when the light (I_L) is turned on. It's also possible to take samples when the light is turned off. The components in such a sample are shown in equation 7.2. If this sample is obtained a short time away from the original PD then the 50Hz and 100Hz values are roughly the same. Therefore subtracting PD_{dark} from PD gives the equa-

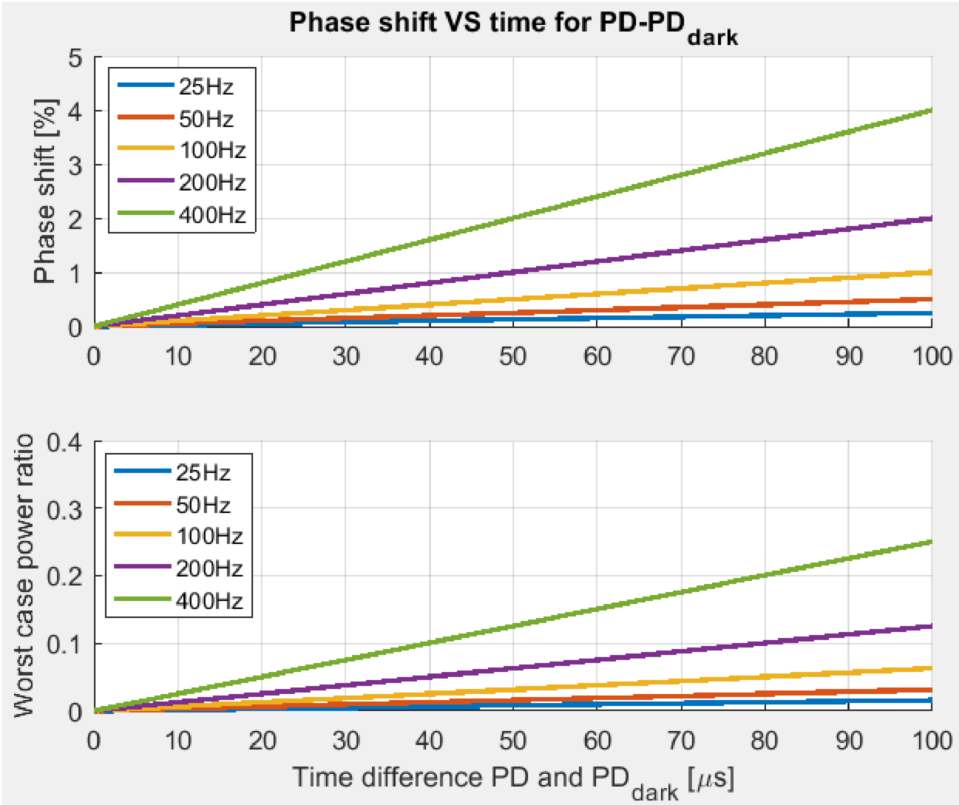


Figure 7.2: The amount of phase shift that will occur.

tion 7.3, resulting in a signal where I_{Eac}, I_{Edc} and N_{50Hz} have practically disappeared.

The signals can't be filtered completely with this method though. This is because of the turn-off time of the LED (see section 2.2), and the filters used to create a usable signal from the turning on and off of the LED (see section **??**). Both introduce a delay to the PD_{dark} sample. Figure 7.2 shows how much several signals get suppressed at various delay values.

Another downside of this method is that adding or subtracting two noisy signals, leads to signal with even more noise. In this case, an increase of approximately ($\sqrt{\sigma^2 + \sigma^2}$ =) $\sqrt{2}\sigma$[**?**], as the σ of PD and PD_{dark} are practically the same.

The huge selling point of this method in comparison to simply filtering, is that no aliasing of I_{Eac} can occur. Another positive point, is that if a LED is specifically selected for it's a short turn off time, then this filter method is able to remove almost all disturbing sinusoid signals.

7.4.2 Detection threshold

The first thing necessary to detect changes in α is a change detector, or with other words, if the signal has changed more than X, then we assume that the changes in the signal are not caused by noise, but by actual changes of α (the environment). from this point it is assumed that I_{Eac} and N_{50Hz} are no longer a significant part of the signal, as they can be practically removed by any of the methods described in section 7.4.1.

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 7.4.2.

7.4.3 Noise reduction

The final thing that needs to be done, to make the algorithm more

Averager [5]

Scaling averager [5]

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

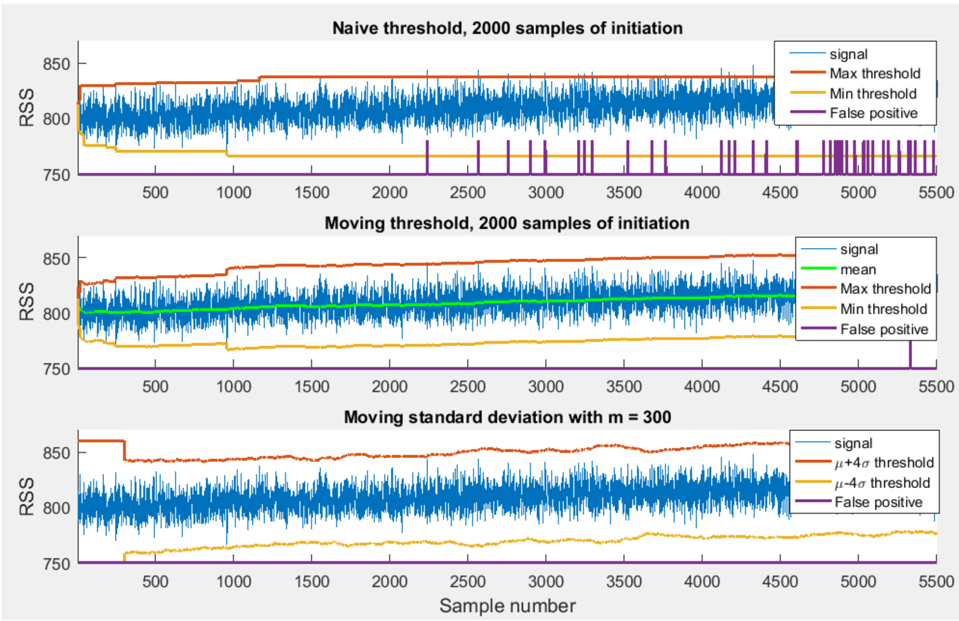


Figure 7.3: An example of how the discussed threshold algorithms respond to a slowly rising noisy signal.

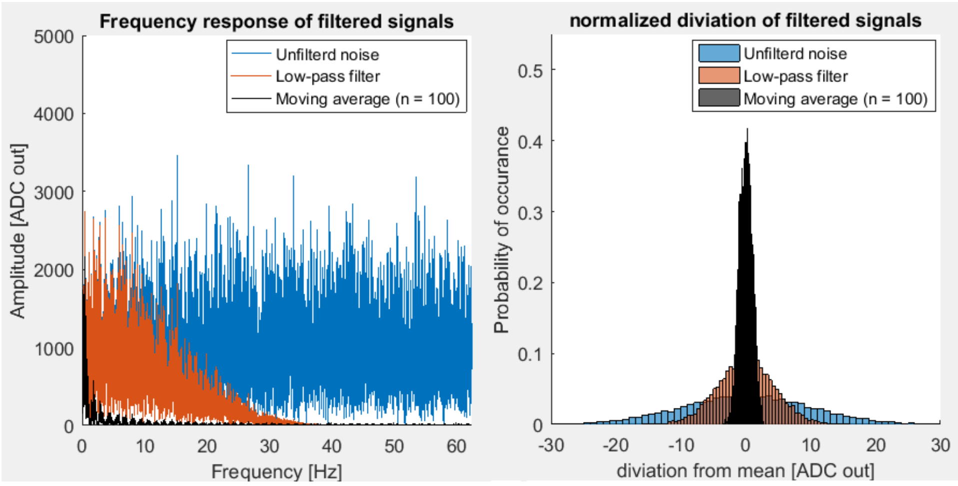


Figure 7.4: Frequency response of the noise, the noise when filtered with a low-pass filter, and then noise when filtered with a moving average.

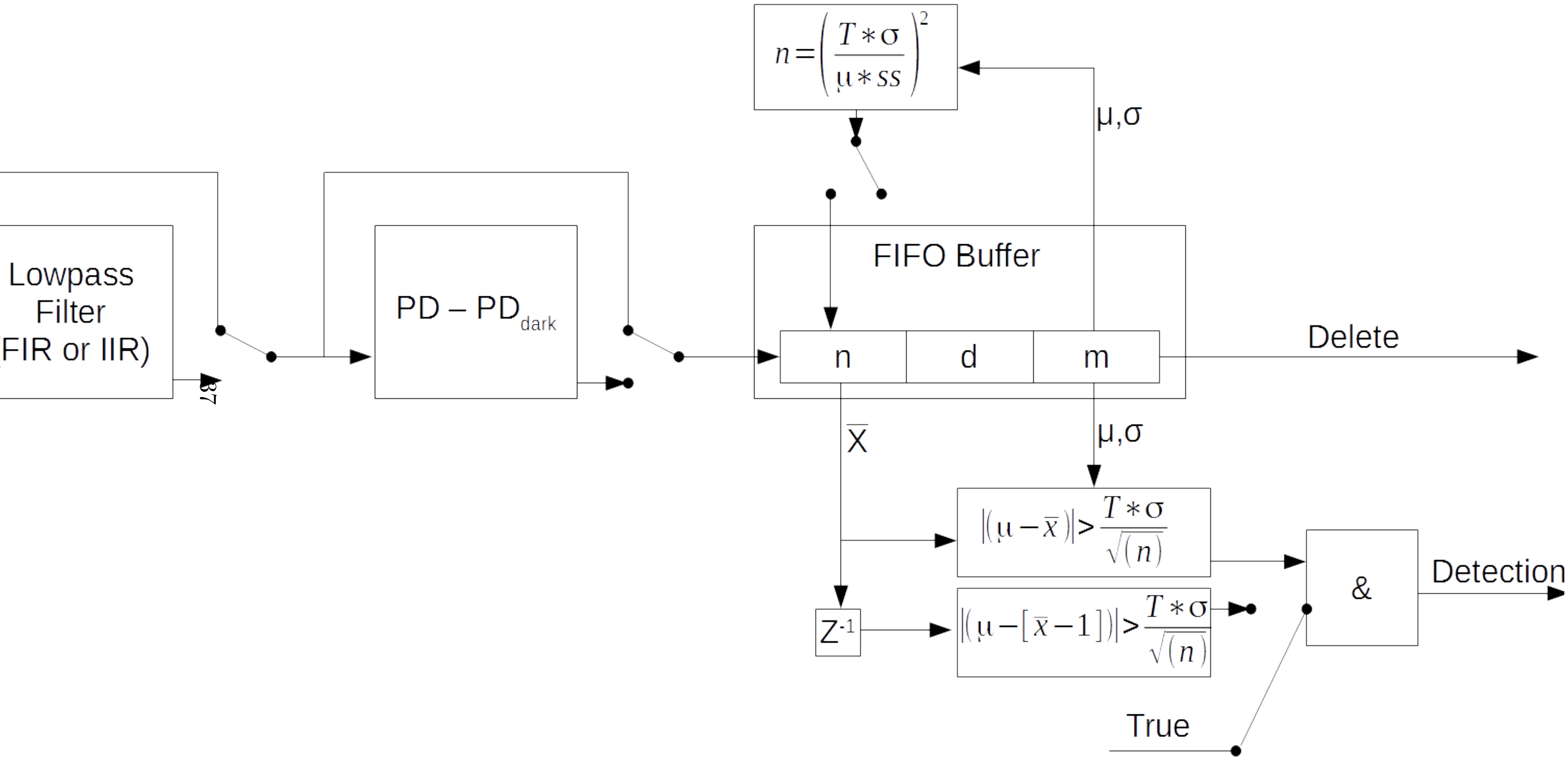
Algorithm	$I_{Edc_n}\beta_n$	$I_{Eac_n}\gamma_n$	N_{50Hz}	$N(\mu,$
Low-pass filter	None	Removed (unless unfortunate alias)	Removed	Redu
$PD - PD_{dark}$	Removed	Removed	Removed	$\sqrt{2}$ i
Moving average	None	Reduced	Reduced	Stati
Scaling moving average	None	Greatly reduced	Greatly reduced	Stati

Table 7.2: Overview of all decent filter algorithms with their effects on each signal.

$$SNR = 1 = \frac{\mu * ss}{T * \frac{\sigma}{\sqrt{n}}} \Rightarrow n = \left(\frac{T * \sigma}{\mu * ss}\right)^2 \tag{7.4}$$

$$T * \frac{\sigma}{\sqrt{n}} = \mu * ss \tag{7.5}$$

7.4.4 Algorithm overview



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

Chapter 8

System evaluation

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 exactly. It is however possible that if two different algorithms trigger a detection, we compare the detection times, and compare those.

8.1 Hallway Evaluation

8.1.1 Test set-up

The system was evaluated indoor and outdoor.

8.1.2 indoor results

8.1.3 outdoor results

8.2 Street

8.2.1 Test set-up

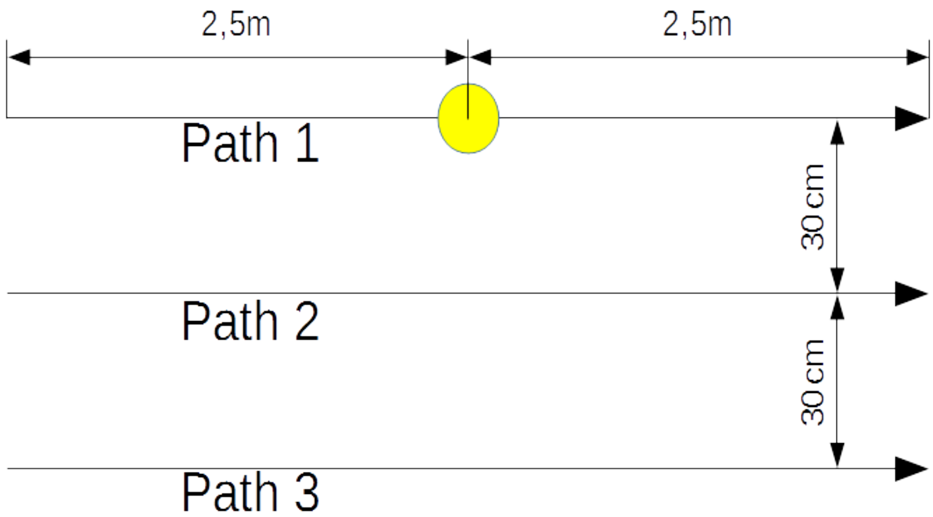


Figure 8.1: The three paths the test subjects had to walk underneath the light.

(a) Indoor

(b) Outdoor

Figure 8.2: Pictures of the test setup 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

System improvements TODO FUTURE WORK - It works with crappy hardware, try the system on a better platfrom: * shorter wires (less noise) * better LED (lower t-rise -¿ more energy saved and less visible light) * make the $PD - PD_dark$ work for possibly even better results - Test the system with a lens like in [14], to get even better tracking

Feature expansion - Implement communication algorithm using lights bouncing from the floor to the next light post, to singal it to turn on. - Figure out a method to have more of these devices working in one area (make the light flashes not interfere with each others detections)

Bibliography

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

[2] 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.

[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. 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.

[5] 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.

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

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

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

[9] 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.

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

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

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

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

[18] 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.

[19] 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.

[20] 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

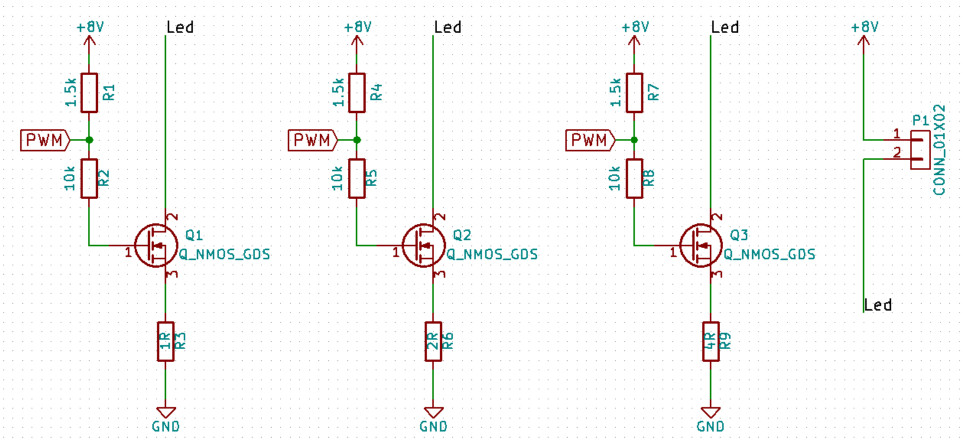


Figure C.1: Drivers used to drive the LED.

C.2 Interfaces between components

@TODO!