Delft University of Technology
Master's Thesis in Embedded Systems

Passive Localization of Robots with Ambient Light

Danielle van der Werff

Passive Localization of Robots with Ambient Light

Master's Thesis in Embedded Systems

Embedded Software Section
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Van Mourik Broekmanweg 6, 2628 XE Delft, The Netherlands

Danielle van der Werff
daniellevanderwerff@gmail.com

23rd May 2018

Author

Danielle van der Werff (daniellevanderwerff@gmail.com)

Title

Passive Localization of Robots with Ambient Light

MSc presentation

30th May 2018

Graduation Committee

prof. dr. K.G. Langendoen (chair)	Delft University of Technology
dr. Przemysław Pawełczak (supervisor)	Delft University of Technology
dr. Marco Antonio Zúñiga Zamalloa (supervisor)	Delft University of Technology
dr. ir. Guido de Croon	Delft University of Technology

Abstract

A lot of research is been being done on Visible Light Communication (VLC), which has shown to be of interest for many applications, such as localization. Since localization based on VLC requires active modulation of light sources, this limits the amount of light sources that can be used for localization. Furthermore, in some situations there might not even be a controllable light source present (for example outdoors). To extend the use of light-based localization schemes, this thesis looks into a way to achieve the same result as current VLC localization methods in a passive manner, i.e. *without* control of the light sources.

Previous work has been done on passive ambient light-based localization [38]: objects are equipped with unique barcodes, that reflect ambient light in a distinct manner. The reflected light is received by photosensors, from which their ID is obtained. However, this work has focused on identifying large-sized objects in one dimension. Using the same principle for localization of small-sized objects, and in two dimensions, are open challenges that this thesis addresses.

The work presented here forms a proof-of-concept of a passive light-based localization system for two-dimensional, real-time tracking of small-sized objects. In order to achieve this, a special enclosure has been designed, giving simple photosensors the ability to distinguish small-sized objects without compromising their FOV. With this enclosure, a single photosensor can detect barcodes down to 7 cm in size in the test set-up, while distinguishing up to three different IDs. A particle filter has been implemented to combine detections from different photosensors into a single estimate of an object's location. The localization system is designed around the robots developed by [33]. By moving these robots at a speed of 15.4 cm/s in a straight line through the test set-up, a localization error of 4.8 cm is obtained. The distance between the robots and the sensor equals 20 cm.

Preface

I have always been interested in how much can be achieved with mere light. During my bachelors in Astronomy I enjoyed learning about how much information we were able to gain from the most distant galaxies by purely observing light. After deciding to pursue a more applied path in Embedded Systems, this interest remained. Therefore, I was very interested in doing my master thesis on Visible Light Communications at the Embedded Systems group. After talking to Marco Zúñiga about possible thesis topics, a topic that struck my attention was localization based on light *without* any control of the light source. Then, the idea came for Marco to co-supervise this project with Przemysław Pawełczak, who had a MSc student (Koen Schaper) working on a small solar-powered robot. The aim was to apply the localization system I would develop to the robot Koen had been working on. I was very enthusiastic, and eager to learn more about the robot and about localization. This thesis will ultimate contribute to such robots not only using light for power, but also for localization. The broader goal for the future, to which I hope my thesis has contributed, is for the Embedded Software group to have autonomous solar-powered and solar-localized robots, working together as a swarm.

This thesis would of course not be the same without my supervisors: I would like to thank Marco and Przemysław for being amazing supervisors throughout the project and for their contagious enthusiasm, their support and their ideas. Dziękuję and gracias! I would also like to thank Koen Schaper for letting me use his robots, and Ioannis Protonotarios for making all my 3D prints. I would like to thank Ruiling Zhang, Chaitra Pai and Weizheng Wang for our discussions on our topics and their company in the lab - either in the old or new faculty building. And lastly, I am grateful for my family and friends for always supporting and encouraging me during this project, it was worth more than they know!

Danielle van der Werff
Delft, The Netherlands
14th June 2018

Contents

Chapter 1

Introduction

1.1 Motivation

GPS is currently seen as the standard localization method for road transport, aviation and smart phones. Under an open sky, a device with GPS compatibility can typically find its location within a 5 m radius [37]. When measuring the location near buildings, GPS signals can suffer from unwanted reflections before reaching the user. This especially occurs in dense urban areas, and it drastically degrades the accuracy of the location estimate. In indoor or underground environments, the signals can even be blocked entirely, rendering GPS useless. Besides this accuracy problem, it has a relatively high cost and energy consumption. Moreover, it is based on radio waves, while the radio spectrum is becoming increasingly crowded.

Alternatives are for example based on Wi-Fi, currently with a 2 m accuracy [7]. This accuracy can be brought down to submeter level if combined with other techniques, at the cost of high environment dependency and deployment costs. Other examples use Ultra Wide Band (UWB), with an accuracy of 25 cm [34]. These methods have the same drawbacks as GPS regarding cost, energy-consumption and radio-based nature. In short, new localization methods need to be found for applications requiring accurate, energy-efficient, yet cheap localization, that can operate both outdoors and indoors.

We believe that visible light offers a solution. The combination of indoor lighting infrastructure, street lanterns and the sun can often provide almost continuous illumination. Using these infrastructures for localization means a simpler and cheaper implementation process. Many studies have already been done to exploit visible light to localize objects. These methods often rely on modulation of a light source, which is commonly known as Visible Light Communication (VLC) [30], [27]. The general concept of this is shown in Figure 1.1a. An LED is toggled on and off in a specific pattern, with a frequency high enough to make the flickering invisible to the human eye.

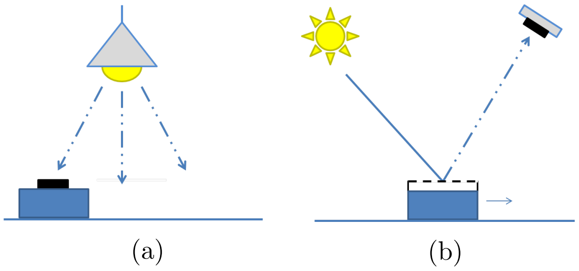


Figure 1.1: Two examples of how visible light can be used to localize objects, in an active (left) and passive (right) manner. The black and dotted boxes indicate photosensors and a reflective tag, respectively.

Objects with a photosensor and processing power can use such light sources as anchor points to compute their relative location. Since light sources are being controlled, this is called *active* visible light localization. This method is usually used for self-localization of objects; if some central entity desires to know where such an object is, an uplink communication path from the object is required.

In some situations there is no light source that can be toggled, e.g. if the sun is the only light source present. In such situations, a *passive* localization method is needed, an example of which is shown in Figure 1.1b. Ambient light shines on an object covered with a tag that reflects light in a distinct manner. When this object moves under a photosensor this unique reflection pattern is observed, which can be used as information on the identity and location of the object. Instead of a modulating light source, it is now the object that encodes data. A central server computes the location of the object, and a downlink channel would be needed to let the object know where it is if desired. The concept described here has already been studied by [38], but only for one-dimensional tracking of large-sized objects. Earlier studies on *active* visible light indoor positioning such as [41] and [14] already achieved an accuracy of less than 10 cm. An open challenge is to achieve this accuracy in a passive manner, especially in two-dimensions, for small objects.

One application that could benefit from having a passive, ambient-light based tracking mechanism, is a robot swarm: a group of relatively simple robots interacting autonomously, showing a desired collective behaviour [32]. They are currently a hot topic in research [9], [19], [17], and will become increasingly popular in our daily lives. An often crucial step for their task is localization. Since swarm robots often have limited resources, such as energy and computational power, localization methods should be straightforward and energy-efficient. The robots are usually small in size, so centimeter level accuracy is desired. At the Embedded Software group of Delft University of Technology, a small solar-powered wheeled robot has been developed by

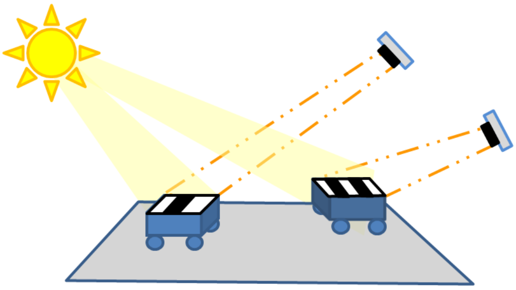


Figure 1.2: The basic concept of this thesis. Two robots and two photosensors are shown.

Koen Schaper for his MSc thesis. The focus of his work was on making the robots transiently powered, harvesting energy from light [33]. Several instances of these robots could be used to form a swarm, which due to the solar-powering, could be self-sustainable. A missing link in making these robots perform tasks in a self-sustainable manner is the localization, which has not been implemented yet. This open problem can be solved while simultaneously tackling the open challenge of passive localization: ambient light reflections can be used for energy-efficient, inexpensive localization of the small robots in a two-dimensional area. Making such a mechanism is what this thesis aims to achieve.

1.2 Research goal

Exploit ambient visible light to localize and identify small objects in a two-dimensional area, and have a server track their paths real-time, without control of the light source and with only minimal infrastructure, i.e. using simple photosensors.

Figure 1.2 shows the basic concept of this localization system. Several (small) objects move around on a surface, illuminated by some arbitrary light source. The objects considered here are the robots from [33]. The aim is to add simple sensors somewhere in the environment and supply these robots with a unique, reflective tag, and be able to track the trajectory of these objects. Ambient light is reflected from the tags, which modulate this light to encode the identification (ID) of the object. Whenever the reflected light is received by a photosensor, this detection can be combined with that from several sensors to localize the object. The sensors, the tags and the software combining the measurements form the three research parts of this thesis. The desired accuracy of this uplink localization system is in the order of a few centimetres.

1.3 Contributions

To the best of our knowledge no system has been developed before to provide two-dimensional localization based on passive exploitation of ambient light. The main contributions of this project are:

1. Design of a new mechanical enclosure to give simple photosensors the ability to detect small-sized objects while maintaining the outer bound of their field of view.
2. Analysis of different encoding schemes to facilitate the embedding of distinct reflective patterns on the external surface of objects. These unique reflective patterns provide identification information.
3. Implementation of a localization algorithm that takes as input the reflective patterns detected by photosensors and provides as output the ID and location of the robot for linear movements.

1.4 Thesis outline

The next chapter will provide background information and discuss relevant papers. Chapter 3 describes the test set-up and main challenges of this project. Chapter 4, 5 and 6 discuss each of the three main components of the research: the sensor, the tag and the algorithm, respectively. Chapter 7 presents the conclusion and topics for future work.

Chapter 2

Related work

This chapter covers required background information (Section 2.1) and discusses work that has been published in related areas: visible light localization methods (Section 2.2) and localization of robot swarms (Section 2.3).

2.1 Background

For this thesis it is important to know how light reflects from surfaces. There are several ways in which reflection can occur, and the difference between them is important for proper design of a reflective tag.

2.1.1 Light reflection

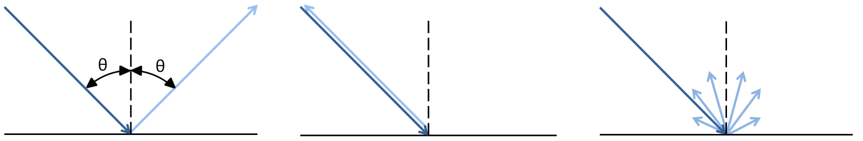
There are three ways in which light can be reflected from a surface. Each of these reflection types is discussed below, and shown in Figure 2.1.

- **Specular reflection**: a light ray hitting a surface with an angle θ compared to the surface normal, is reflected with the same angle on the opposite side of the surface normal. This reflection occurs on very smooth surfaces, such as mirrors or still water.

- **Retro-reflection**: a light ray hitting a surface is reflected back along the same direction where it came from. Retro-reflection is actually a form of specular reflection in which several specular surfaces are combined in such a manner that the net effect of the surfaces is retro-reflection. Such surfaces are found in the reflective tags of safety vests and bicycle lights for example.

- **Diffuse reflection**: a light ray hitting a surface is reflected into of range of directions, due to the unevenness of the surface. Examples of such surfaces are paper, cardboard and asphalt.

Each reflection type each has its own use in different applications. In Figure 2.1a, an observer aligned with the light blue ray will detect a high signal. Light is completely reflected to the observer, ensuring all the energy is detected. However, if this observer moves to a different location, no light will be observed. The area in which an observer can be located to detect a given signal is small. A retro-reflective surface is useful when the light source and the observer are placed at the same angle. In visible light communications this is useful if an object contains both a transmitter (LED light) and a receiver (photosensor). With a diffuse surface as in Figure 2.1c, a reflected light signal can be observed from various angles, mitigating the need to accurately align the surface and the observer. However, the received signal strength is significantly less, since the light energy is spread over various angles.



(a) Specular reflection (b) Retro-reflection (c) Diffuse reflection

Figure 2.1: Different types of reflection.

2.2 Light-based localization techniques

Various papers have already proposed light-based localization techniques, which can broadly be split into two categories. One category is *active localization*, which in this thesis is considered to be all techniques that require an active light source such as an LED or a transmitter sending a modulated light beacon. The second category is *passive localization*, which are techniques where light sources are not controlled; but the rest of the system (e.g. the object to be tracked) can have active electronic components. As stated before, this thesis aims to achieve passive localization with ambient light. The following sections give an overview of current research in light-based localization techniques, to see how the system proposed in this thesis relates to the state of the art.

2.2.1 Active localization

Visible light based localization often relies on VLC. Philips Lighting obtained an accuracy of 30 cm for indoor positioning in retail stores with VLC [23]. These stores are illuminated with LED lights sending out modulated light signals. Customers in these stores can capture the light signals with their smartphones. Using a special app on their smartphones, the received data can help customers find their way in the store or get location-based promotions and notifications. Methods such as [41] and [14] achieved a localization accuracy below 10 cm in VLC test set-ups.

An example of an active light localization method used for robots similar as the ones used in this thesis is [4]. This work presents a way of powering microbots with solar cells, and simultaneously exploit these solar cells to act as an integrated GPS. The microbots are part of a project called Alice, and are developed by [6]. The localization done by [4] has a projector as active light source, which projects a series of images onto the surface on which these microbots are situated. The pattern received at a particular location in this area is unique to this spot, so the pattern that the robots observe indicates their position and orientation within the area.

The Zooids [22] have a similar example of light-based localization for swarm robots. Zooids is an open-source and open-hardware swarm robot interface for autonomous table-top robots, that can interact with humans. The wheeled robots are 2.6 cm in diameter, and are currently localized using a large overhead projector that displays a sequence of gray-coded patterns onto the surface on which the Zooids are located. The Zooids detect these patterns using photodiodes and use these patterns to compute their location and orientation. For both [4] and [22] the locations that the robots compute are sent to a radio base station by RF signals.

The system proposed in this thesis would provide a projector-less alternative for both robot platforms, if combined with a downlink channel (a channel from the central server to the robots).

2.2.2 Passive localization

An example of passive light-based localization is LocaLight [21], which is a battery-free system for uplink indoor localization. By having light bulbs at particular locations on the ceiling, and RFID sensor tags with photodiodes on the floor, the shadow of objects is used to determine the position of the object. The authors have obtained an accuracy of 50 cm. It removes the requirement of controlling a light source, but the location of both the light sources and sensor tags must be known.

Another approach is to let users in an indoor environment measure the spectrum of the light they receive, which is then assumed to be unique for different positions throughout the indoor environment, as done by [25]. Comparing the measured color spectrum with a predetermined database indicates where the user is mostly likely situated. However, this is not robust to changes in the lighting infrastructure, and the time of day at which the positioning is performed. If something in the environment changes, a new database is required.

The paper that is the most relevant to this thesis, and where the work presented here builds upon, is [38]. It presents a system for passive com-

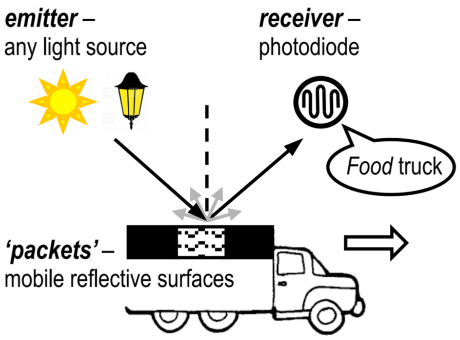

emitter – any light source
receiver – photodiode
Food truck
'packets' – mobile reflective surfaces

Figure 2.2: The concept of a passive light-based system for identifying an object, figure taken from [38].

munication in illuminated areas. The concept is shown in Figure 2.2. Light from either the sun or an existing lighting infrastructure is reflected from unique reflecting surfaces that can be placed on mobile nodes. In this work, cars were used as mobile nodes and barcodes as reflective surfaces. The reflective surfaces encode the data in a passive manner. When such an object moves, a distinct pattern of reflected light is sensed by a simple photodiode. This unique pattern is decoded into the received data. The system is used to identify objects, while they pass through the sensor's field of view (FOV). As far as the author knows, no research has been done using such a passive method for two dimensions or for smaller scale objects, where the encoding of reflective surfaces is less stable. This thesis aims to fill this gap.

2.3 Tracking robot swarms

Now that the current state of the art regarding (passive) light-based localization has been discussed, it is important to look how these localization techniques fit into robot swarm localization. There is no absolute standard when it comes to (indoor) localization of small-sized robots. The most extensively studied type of localization for robot swarms is downlink: self-localization. This can roughly be divided into two parts: sensing and computing. It gathers data from its environment or from its own behaviour (odometry), and subsequently computes its location based on whatever has been sensed and what was known beforehand (such as an initial location or a map of the environment).

2.3.1 Sensing

Sensing mechanisms used for robot swarm localization are usually based on GPS. Common alternative methods are based on vision [2], [8], [35], radio signals [12], [31] or inertial sensors (gyroscope and accelerometer data).

Other common techniques use the magnetic field, or ranging with ultrasound [5] or lasers [16]. Besides the localization methods for the Zooids and the Alice robot mentioned in the previous section, not many light-based sensing methods have been found for robot swarms as far as the author knows (not including the use of expensive and energy-consuming cameras).

2.3.2 Computing

After sensing has been done, the robot computes its location. This can either be based on a known map, or it has to determine the map from its measurements. If the map is unknown, a common algorithm to be used is Simultaneous Localization and Mapping (SLAM) [11]. By visually scanning the environment a map can be made of the surroundings of the robot, while localizing itself in it.

If a map is known, the computation of the location is often based on the robot's distance to anchor nodes: a technique known as trilateration. Another technique is fingerprinting, where a variable (such as the Received Signal Strength Indicator (RSSI) of Wi-Fi access points) has been measured beforehand at various locations in a known environment and stored in a database. When a robot performs a measurement of this variable, the given value can be compared to the values in the database, indicating what the most likely location is for the robot [26]. A different approach is dead reckoning, where a robot uses its own movement and a known initial position to compute its new position. Software techniques such as particle filters or Kalman filters can be used to enhance the location estimates described above.

The system in this thesis will estimate the location of objects on a central server. The nodes themselves are not aware of their location; the communication from this server to the nodes is left for future work.

Chapter 3

Methods

This chapter covers the test set-up (Section 3.1), the system requirements (Section 3.2) and the main challenges for each component of this thesis (Section 3.3).

3.1 Test set-up

As described, the system consists of a light source, simple photosensors, and reflective tags that will be placed on objects. The basic set-up for which the tracking system is designed is shown in Figure 3.1. The light source from the concept shown in Figure 1.2 is in this case a 60×60 cm LED light panel to have a more controlled environment. This results in a relatively uniform and constant light source, for easier testing of the system. The final system should of course also work with other, non-constant light sources. The square light panel is placed at a height of 20 cm above a table, facing down towards the table surface. A grid of photosensors is placed on the illuminating side of the LED panel, which is indicated by the grey dots in Figure 3.1.

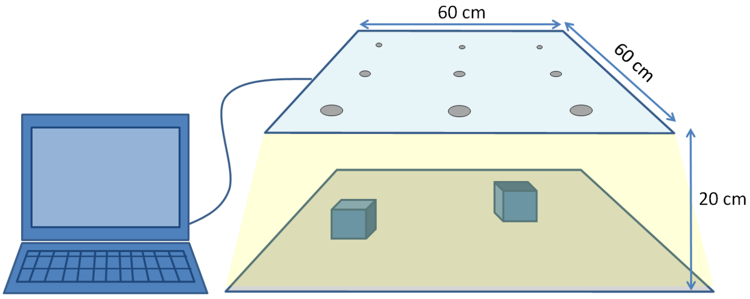
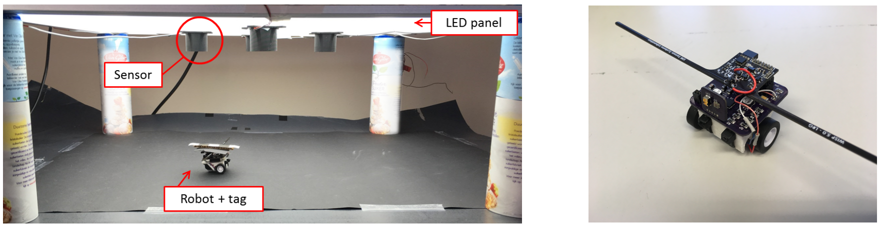


Figure 3.1: The basic set-up of this project: a light source, several objects and a grid of photosensors (depicted by the grey circles) connected to a PC.



(a) (b)

Figure 3.2: The set-up as used in the laboratory, and the robot developed by [33].

The robots are equipped with a tag, and move on the table surface. They are depicted by the blue boxes in Figure 3.1. The table surface is covered with black paper, to obtain a higher contrast with an object.

Each photosensor is connected to an Arduino Nano, with an ATmega328 microcontroller. The microcontroller is connected to a PC, and sends the voltages measured by each sensor to the PC. Using Python software on the PC, these voltages are converted to a path estimate of objects located under the LED panel. The real-life set-up is shown in Figure 3.2a.

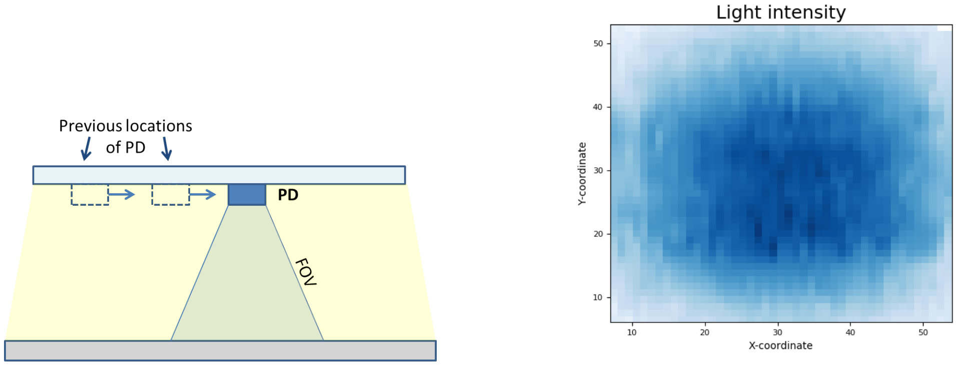
3.1.1 Robot

The localization system in this thesis is designed around the transiently-powered battery-free robot designed by [33], which is shown in Figure 3.2b. It is 3.5 by 4 cm in size, which imposes the requirement on the tracking system that it should be able to detect objects with sizes down to 3 to 4 cm. Energy is harvested from light using a solar cell, and this energy is temporarily stored on a 22 mF/4.5 V supercapacitor. Since the available light is not always sufficient, the energy from the supercapacitor is not always enough for continuous locomotion of the robot. Therefore, the robot is made to operate intermittently, having as little error in the final trajectory as possible. For this project, the robot is used with a battery, to avoid waiting time during experiments.

3.1.2 Light source

The LED panel is bought from [1] and is an SRPL 60×60 cm 40 W square LED panel with a Sharplumi LED driver, with an illuminating area of 55×55 cm. The spectrum of the emitted light has a strong, narrow peak around 450 nm and a less strong, but broader distribution around 550 nm.

Although the light intensity from the LED panel seems to be uniform throughout the square area, the intensity is expected to be the highest in the center. This should be quantified, since the light intensity can influence the tests. A scheme has been devised where an SFH203P Osram photodiode is



(a)

(b)

Figure 3.3: The left image shows how the measurements were taken: the diode moves to different locations and measures the light intensity at each point. The right image shows the location under the panel versus the amount of light measured from that location. The detected light intensity is 1.74× stronger in the center compared to the edges.

placed on the LED panel, able to shift in two directions across the panel. The diode is moved across every cm^2 under the panel, while the light intensity is measured for each location. This set-up is shown in Figure 3.3a for one direction. The field of view of the photodiode (PD) is limited by a cap, due to which light directly from the LED panel to the PD is blocked. The measured light is the reflected light from the table, without any material on it. The results are shown in Figure 3.3b. The intensity measured in the center is higher than the intensity at the edges, by a factor of 1.74. This plot explains why results differ for different locations under the panel later on (Chapter 4).

3.2 System requirements

The requirements of the localization system are as follows:

1. The system should track small size (3×3 cm) objects with centimeter level accuracy.
2. Only ambient light should be used for localization, without control of the light source(s). The user cannot control any parameter of the light, such as the intensity or the direction.
3. The system should be energy-efficient and simple, due to the limited energy and computational resources of the robot.

As future work, one could make the tag active, using for example an LC shutter. This is a glass-like material that can switch between opaque and transparent, based on a voltage that is applied to this material. The addition of such a shutter to the tag would enable an object to autonomously change its ID if desired.

3.3 Challenges

The design of each of the three main components (sensor, tag and algorithm) of the set-up has its own challenge, which is discussed in this section.

3.3.1 The sensor

As discussed in Chapter 2, the system from [38] localizes objects by placing barcode-like reflective surfaces on objects, and detecting patterns resulting from these surfaces. The two open challenges in using such a system for this project are to detect small-sized object, and to localize in two dimensions.

Challenge 1: small-sized objects

The reflective surface from [38] is a barcode with a 10 cm symbol width. Several symbols are needed to distinguish a significant amount of objects, resulting in almost meter-sized barcodes. Since the goal of this thesis is to localize small-sized objects, such barcodes would not be feasible. Decreasing the size of the barcode and keeping the rest of the system as is, makes the barcode fit on the object, but also results in inter-symbol interference (ISI).

This trade-off is shown in Figure 3.4. Photodiodes only measure one value for all light received within their FOV, so if the full barcode is in this FOV only a single value will be measured. This is shown on the left. There will be no way of distinguishing the individual bits of the barcode. In Figure 3.4 on the right, the FOV is decreased to match the size of a single barcode bit. The barcode can be detected while it crosses the FOV of the sensor, but the coverage of this sensor is severely compromised. A simple photodiode will only be able to either cover a significant area, or distinguish the individual constituents of the barcode.

As stated in Section 3.1.1, the barcode size should not be much larger than 3 cm. To distinguish a pattern within a 3 cm sized barcode, the field of view (FOV) should be less than the smallest part of the barcode that should be distinguishable. This thesis proposes a method where both properties can be achieved, while still only using a simple sensor.

Challenge 2: two-dimensional localization

The second challenge is based on the fact that the system of [38] is able to localize objects in one dimension, where the object to be localized has a fixed path and will always cross the sensor's FOV. The aim of this thesis is

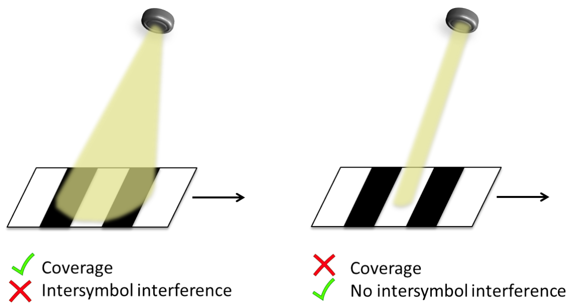
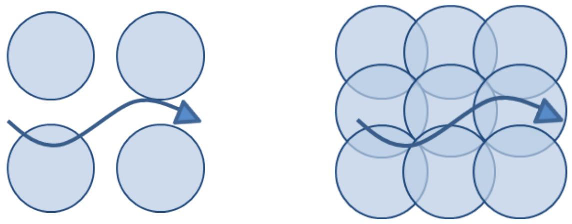


Figure 3.4: A sensor with large coverage, resulting in ISI (left) and a sensor with smaller coverage, without ISI (right).



(a)

(b)

Figure 3.5: The blue circles indicate the area in which a photosensor can detect an object (its field of view). An object moves through a path indicated by the blue line. On the left, a cost- and energy efficient system that often misses the object. On the right, an expensive system where the object is (almost) continuously tracked.

to expand this to tracking in two dimensions, where an object has a variable path and might not always be in the FOV of the sensor.

One way of doing this, is to place many sensors to increase the coverage and thereby increase the probability that the object crosses the sensor, as shown in Figure 3.5. The black dots in these images indicate the location of photosensors, and a narrow FOV is assumed for these sensors to overcome the ISI problem. The left image shows four of these photosensors. An object moving in a path as given by the red arrow will rarely cross FOV of a photodiode, and its location will therefore often be uncertain.

On the right image, there is an increased chance of an object being detected by a sensor. This increases the accuracy of the localization. However, the larger number of sensors is not energy- nor cost-efficient Moreover, in many situations it is not possible to use more than a certain number of sensors. We could use fewer sensors and compensate for the limited coverage by having software-based estimation methods of the location, such as particle filters or Kalman filters. Therefore, this results in a trade-off between the number of sensors and the computational complexity.

These two challenges form the core of the thesis, and the solutions will be discussed thoroughly in the following chapters.

3.3.2 The tag

The second part of the system is the identification part: the tag. Encoding and ID via light reflections on such a tag can be done in multiple ways. The work discussed in the previous section assumes a black and white barcode to identify an object, as was done by [38]. Such a barcode is an example of modulating the intensity of the light. Besides the intensity, other light properties can be exploited too for this purpose, such as the polarization or the color (frequency). A tag can be encoded via differently colored blocks, specific combinations of polarization sheets or different types of shapes that reflect light in different directions.

For this project, polarization, color and intensity were considered as methods of identification. The possibilities of the identification should be matched with the requirements of the system. Besides this method, it is necessary to think of an encoding scheme, how many objects can be identified, and how to convert a detected light pattern to an object ID. The chosen method here is intensity (Chapter 5).

3.3.3 Algorithm for path estimation

Section 3.3.1 discussed the trade-off between the number of sensors and the computational complexity. Since the system will not be able to have continuous coverage of the objects, the sensors only give sparse information. When two subsequent measurements are done at different sensors, a filter is needed to derive what path the object has taken in the mean time, in a real-time fashion. The trajectory of the robots can be unpredictable and is most likely non-linear. Several objects need to be tracked at once, and multiple properties of these objects need to be tracked (position, orientation and velocity). Filters such as a Kalman filter, or the variations Unscented or Extended Kalman filter, do not perform optimally for such a situation [20]. Particle filters are the most suitable for *multivariate* (i.e. tracking multiple quantities), *multimodal* (i.e. possibly tracking multiple objects) and non-linear systems, as is the case in the set-up of this project [18].

This filter is discussed more in-depth in Chapter 6. The resulting algorithm recognizes the tags from the voltages measured by the sensor, and displays the trajectory of the objects that are being tracked.

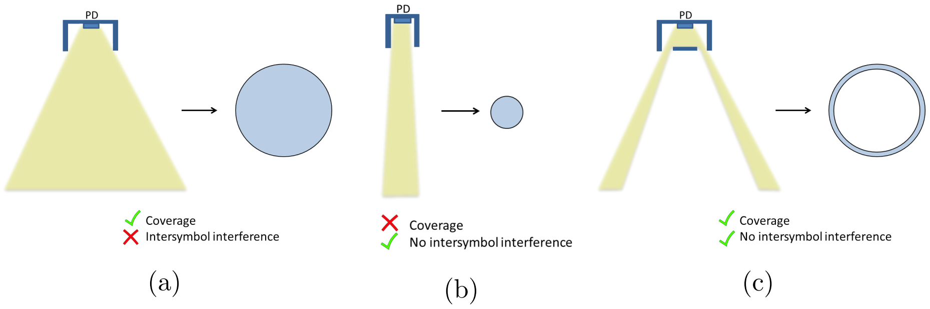
Chapter 4

The sensor

This chapter proposes a solution to overcome the FOV versus coverage trade-off (Section 4.1), for which a model has been designed (Section 4.2). The chapter concludes with a description of the final photosensor chosen for the tracking system (Section 4.4).

4.1 Field of view versus coverage

The proposed solution in this project to achieve a large coverage area for a photosensor, while having the ability to observe small-sized objects without intersymbol interference, is to not use the full FOV of a sensor as in Figure 4.1a, but to partially block out the FOV as in Figure 4.1c. Instead of a cirle, the FOV is a thin ring. If this ring is made thin enough, a single point on this ring is narrow enough to distinguish small elements of the tag. When a unique tag moves in or out of this ring, smaller objects can be observed without intersymbol interference.


PD
✓ Coverage
✗ Intersymbol interference
PD
✗ Coverage
✓ No intersymbol interference
PD
✓ Coverage
✓ No intersymbol interference

(a) (b) (c)

Figure 4.1: The figures show a cut-through depiction of delimited sensors, and their resulting FOV.

In contrast to a strongly focused FOV covering a small area, there is not only a single point, but a collection of points where this can be observed. The

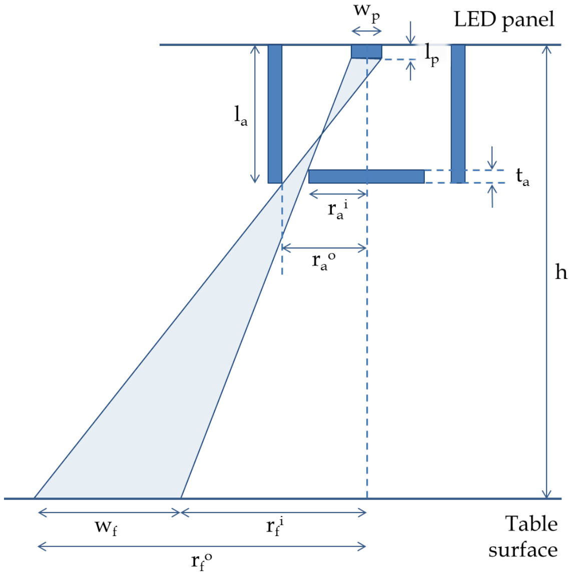


Figure 4.2: A cut-through of an aperture blocking the photosensor, and its resulting FOV. The two diagonal lines depict the fraction of light that is reflected from the table surface onto the photosensor. Only the light rays on the left side of this two-dimensional figure are shown for simplicity.

resulting FOV has the same outer boundary as in Figure 4.1a, but a much smaller tag element can be distinguished. As a result, the coverage can still be significant, while small sized tag elements can be observed throughout the covered area. Several of these rings could be combined or overlapped to track nodes within a larger area. By combining measurements from different sensors, a cheap and efficient way can be found to estimate the location of the object.

4.2 Aperture model

Both the radius and the width of the ring FOV are determined by the dimensions of the obstruction that is placed in front of the photosensor. It is useful to make a model of the effect of a particular obstruction size on the FOV. For this purpose, we use the concept of a circular *aperture* as as shown in Figure 4.2. We use a TEMD5510FX01 Vishay photodiode. The model will be used to compute optimal dimensions of an aperture to achieve the desired FOV ring.

The photosensor is placed facing downwards. A hollow cylinder determines the outer boundary of the FOV. The length and radius of this cylinder are from now on referred to as the aperture length l_a and aperture outer radius r_a^o. The obstruction is placed below the photosensor, at a distance

Value	Advantages	Disadvantages
l_a	Smaller object size	More intrusive sensor
r_a^o	Larger coverage Smaller object size	Signal loss due to angular sensitivity More intrusive sensor
w_f	Easier fabrication Higher received signal	Larger object size

Table 4.1: (Dis)advantages of increasing different parameters of the aperture model.

equal to l_a. The radius of this obstruction will be called the inner aperture radius r_a^i. Photosensors are usually less sensitive to light arriving from larger angles, meaning that this radius should not be too wide. The height of the sensor above the table/ground surface is fixed.

From the aperture parameters, both the FOV radius r_f and FOV width w_f can be computed. From Equations 4.1 and 4.2 the inner and outer FOV radii are obtained, whose difference equals w_f. The formulae are discussed in more detail in Appendix A.

$$r_f^i = \frac{r_a^i - \frac{w_p}{2}}{l_a - t_p - t_a} \cdot (h - t_p) + \frac{w_p}{2}. \tag{4.1}$$

$$r_f^o = \frac{r_a^o + \frac{w_p}{2}}{l_a - t_p} \cdot (h - t_p) - \frac{w_p}{2}. \tag{4.2}$$

With this model, one can optimize the aperture to result in a particular FOV, given a few constraints. Table 4.2 provides an overview of all parameters and constraints within the model, given the test set-up. The aperture slit, which is the difference between the outer and inner radii of the aperture, $r_a^o - r_a^i$, is denoted by w_a. Table 4.1 shows the influence of an increase in different parameters of the model, as can be derived from Equations 4.1 and 4.2. A smaller FOV width means that a smaller object can be detected by the photosensor. A larger FOV radius means a larger area is covered by the sensor.

With this model, the aperture dimensions for a desired FOV radius and width are determined. Given the size of the test-set-up, the aim for the FOV radius is 15 cm. The FOV width is aimed at below 3 cm, which means that the smallest distuinguishable element of a reflective surface is also less than 3 cm. This ensures that a reflective tag of a few centimetres is achieved. The following sections describe what aperture dimensions are needed to obtain these FOV parameters.

Constants	**Symbol**	**Selected value (mm)**	
Sensor height	h	180	The distance between the LED panel and the table surface is 20 cm, but aperture has a thickness of 2 cm.
Photodiode width	w_p	2.7	
Photodiode length	l_p	2.6	
Aperture thickness	t_a	0.7	
Variables	**Symbol**	**Constraints (mm)**	**Reason**
Aperture length	l_a	$l_p < l_a < 30$	
Aperture outer radius	r_a^o	$10 < r_a^o < 30$	A too large aperture blocks light, takes up too much space, and the photosensor is less sensitive to light from larger angles.
Aperture slit	w_a	$1 < w_a$	Limitation of the fabrication method (3D printing).
FOV inner radius	r_f^i	> 50	A barcode should not be in different segments of the FOV ring simultaneously.
FOV outer radius	r_f^o	150	
FOV width	w_f	< 30	

Table 4.2: Parameters and constraints of the aperture model.

4.2.1 FOV ring radius

Figure 4.3 shows what r_f^o is for different pairs of l_a and r_a^o, computed with Equation 4.2. All other parameters in the equation are fixed. The values in Figure 4.3 are plotted within the constraints given in Table 4.2. The lines in the figure indicate the set of (l_a, r_a^o) pairs that result in a FOV radius as indicated by the labels. Since a FOV radius of 15 cm would be desirable for the test set-up, all l_a and r_a^o combinations on or near the 15 cm line in Figure 4.3 form a valid set of aperture parameters.

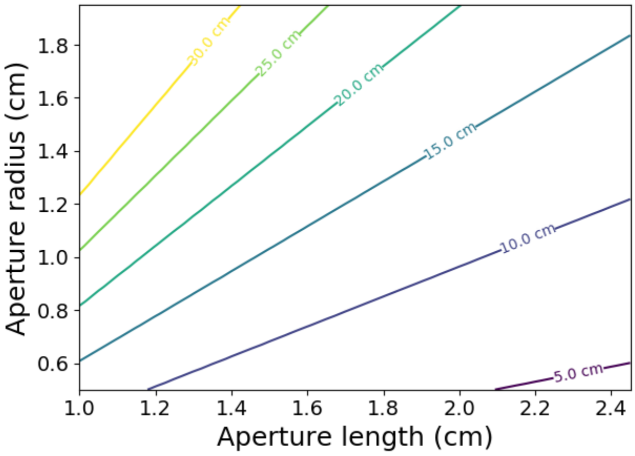


Figure 4.3: The lines in this figure show all combinations of l_a and r_a^o that result in a FOV radius as described by the label of the contour.

4.2.2 FOV ring width

From the set of (l_a,r_a^o) values that result in a FOV radius of 15 cm as shown in the previous section, we want those values that also result in the desired FOV width. By subtracting Equations 4.1 and 4.2, w_f is computed. This equation requires a value for r_a^i, which is assumed to be 1 mm, which is the smallest possible size given the fabrication method that will eventually be used for the aperture (3D printing). Figure 4.4 shows what w_f is for different pairs of l_a and r_a^o. The values of l_a and r_a^o are again bounded by the constraints given in Table 4.2.

4.2.3 Chosen aperture parameters

The (l_a,r_a^o) combination that satisfies both $r_f^o = 15$ cm and $w_f = 3$ cm is found by combining the lines in Figures 4.3 and 4.4, which is shown in Figure 4.5. The aperture parameters located on or near the red dot, result in (approximately) $r_f^o = 15$ cm and $w_f = 3$ cm. The final aperture is chosen

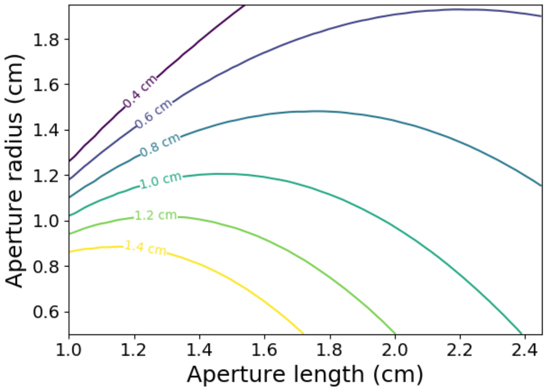


Figure 4.4: Each combination of l_a and r_a^o results in a FOV width indicated in cm by the labels of the contours.

as $l_a = 2\,\text{cm}$, $r_a^o = 1.5\,\text{cm}$ (with $w_a = 1\,\text{mm}$), which results in a FOV of $r_f^o = 16.5\,\text{cm}$ and $w_f = 3\,\text{cm}$

With these values, the angle of incidence of the light is in the range of $\theta = [35{,}37]°$. The TEMD5510FX01 photodiode for which the aperture has been optimized has a sensitivity of 90% at these angles, meaning that the signal loss due to the angle is not a concern.

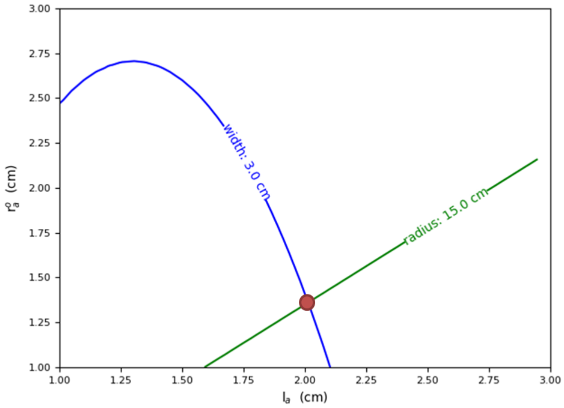


Figure 4.5: This figure shows pairs of (l_a,r_a^o) that result in $w_f = 3\,\text{cm}$ in blue, and pairs that result in $r_f^o = 15\,\text{cm}$ in green.

4.3 Performance of the designed aperture

Two shapes of apertures with parameters as computed in the previous section are compared, to find the optimal aperture. These shapes result in a circular or square FOV.

4.3.1 Circular FOV

A cylinder-shaped aperture results in a circular FOV. The image shown in Figure 4.2 holds in all directions throughout this cylinder. Of course a real aperture would not result in a hard boundary of the FOV, due to reflections within the aperture and inaccuracies in the fabrication and orientation of the aperture. The expected ring is somewhat noisy, like a hard ring convolved with Gaussian distributed noise.

The circular aperture with the obtained parameters is 3D printed in black PETG filament, resulting in the aperture in Figure 4.6. The bottom part of the design is shown on the left and has four 'arms' that are used to attach the aperture onto the ceiling. The photosensor is glued onto the center of this part. The middle image shows the cylinder part with four arms to hold the obstruction in place, which are kept as small as possible.

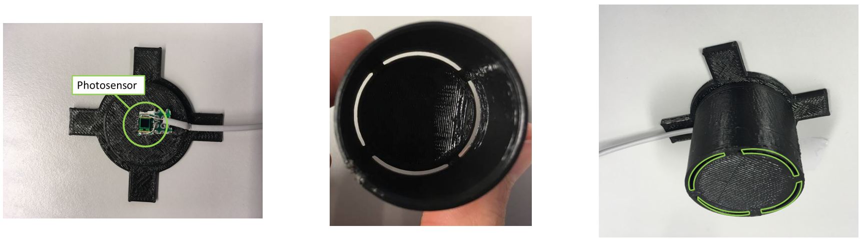
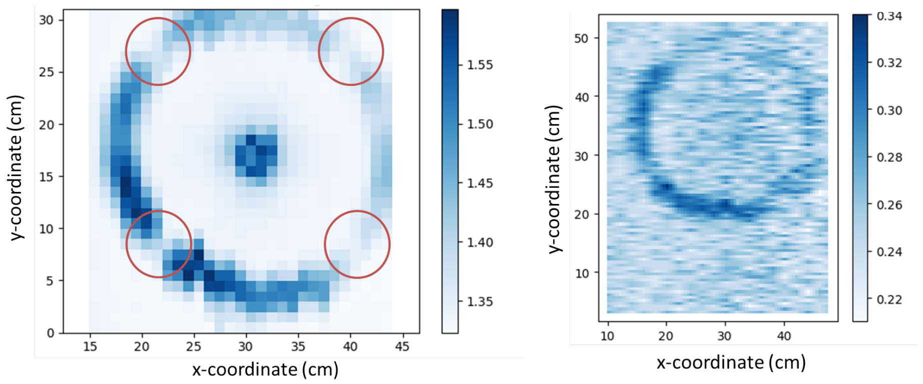


Figure 4.6: The left and middle figures show the bottom and cylinder parts of the aperture. The right most figure shows the resulting sensor, where the opening for the light rays is emphasized with the green lines.

A 3x3 cm square of wrinkled aluminium foil, which effectively functions as a diffuse reflector, is moved under the LED panel to consecutively cover every squared centimetre under the panel. For every location, the light intensity is measured by the sensor. This indicates what the FOV of the sensor is. The result of these measurements is shown in Figure 4.7a, where the ring-shaped FOV is visible. Each value is an average of 100 samples.

There are several things that can be noted from the ring-shaped FOV of Figure 4.7a. First of all, there is a strong peak in the middle. This peak is caused by inner reflections of the sensor. Light rays from directly below the sensor are reflected in such an angle within the aperture, that they reach the photosensor. This point is supported by the fact that the inner peak is not seen when a non-reflecting material is used for fabrication of the aperture. Figure 4.7b shows the resulting FOV from an aperture made out of cardboard, again measured with a 3x3 wrinkled aluminium foil. There is clearly no strong center peak in this image, while all other circumstances were kept constant. The inner peak is relatively thin, and can therefore



(a) 3D printed aperture

(b) Cardboard aperture

Figure 4.7: The left image shows the FOV measured by moving a 3x3 cm wrinkled aluminium foil under the circular aperture. The red circles show where the 'arms' of the obstruction block the incoming light. The right images shows the same measurement done with a cardboard aperture instead of a 3D printed one. The center peak is not visible here. The colorbar indicates the voltage (in Volt) that is measured by the photodiode.

function as an extra point of detection, and will therefore be used as an extra detection area to enhance localization accuracy.

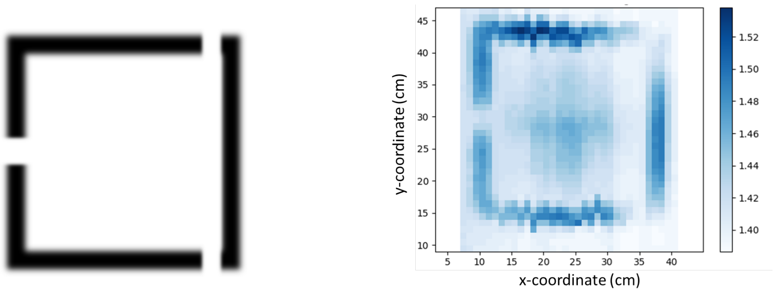
Secondly, the ring is not uniform. This is due to the non-uniformity of the LED panel (Figure 3.3b), since the sensor was not placed in the center of the panel. Lastly, the 'arms' that keep the obstruction in place can be seen in the image as well, locally diminishing the sensitivity at these locations.

The average FOV width in Figure 4.7a is 5.5 cm and the average FOV radius 13.5 cm. The FOV is computed here as all locations where the measured light intensity is five times higher than the standard deviation of the noise - meaning a 99.977% chance that the measured light intensity is actual signal and not noise. Since these measurements can of course not be performed with an infinitesimally small object, the FOV shown in the image is the sum of the object width and the actual FOV. To obtain the actual FOV width of the aperture, without the effect of the non-zero size of the aluminium foil, the foil size should be subtracted from the 5.5 cm. This makes the final FOV ring width and radius 2.5 and 13.5 cm for the photodiode. This is quite close to the modelled 3 cm and 15 cm.

4.3.2 Square FOV

A circular shape has the advantage that points throughout the ring all have approximately the same distance towards the sensor, so they all contribute the same to the total amount of light measured. Other shapes might not have this advantage, but they do have other advantages. Square shapes could for example be combined in a compact way, in which each square area of the table surface can be covered by a specific aperture.

A square aperture has been 3D printed as well, with the same w_a, l_a and r_a^o as the circular aperture. The expected FOV shape is shown in Figure 4.8a, and the measured FOV in Figure 4.8b. A FOV square has a length and width of 28 cm. The strength of center peak due to inner reflections is less high for this shape, but it is more spread out. This means that there is quite a large area under the photosensor where objects will result in a signal measured by the sensor, which due to the broad size of this center peak will suffer from intersymbol interference. Therefore, the circular aperture is chosen instead of the square.



(a)

(b)

Figure 4.8: The square aperture (left) and the resulting FOV (right), measured in the same manner as Figure 4.7a.

4.4 The final photosensor

4.4.1 Photodiode versus phototransistor

Several photosensor specifications were compared on several criteria, such as the spectral sensitivity (which should match with the spectrum of the LED panel for a high detected signal), price, angle of half sensitivity θ (which should be high enough to use the photosensor with an aperture blocking its FOV) and switching speed (which should be fast enough to detect a small robot passing by). Initially, the TEMD5510FX01 Vishay photodiode was chosen, based on these criteria. The aperture optimization described here, has been done using this diode. However, for later stages of the experiments, a more accurate detection method was required. For this purpose, a TEMT6000X01 Vishay phototransistor was chosen instead of this photodiode, due to the lower amount of noise encoutered when using a phototransistor [3].

Both sensors have a sensitivity spectrum that matches the panel light, they have large enough angles of sensitivity ($\theta_{PD} = \pm 65°$ and $\theta_{PT} = \pm 60°$)

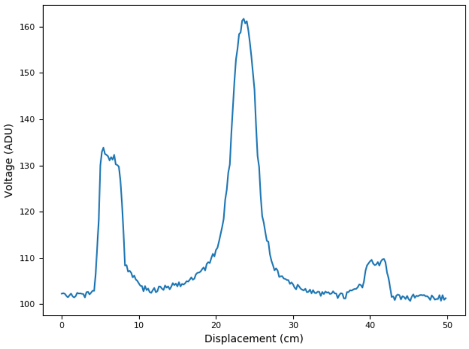


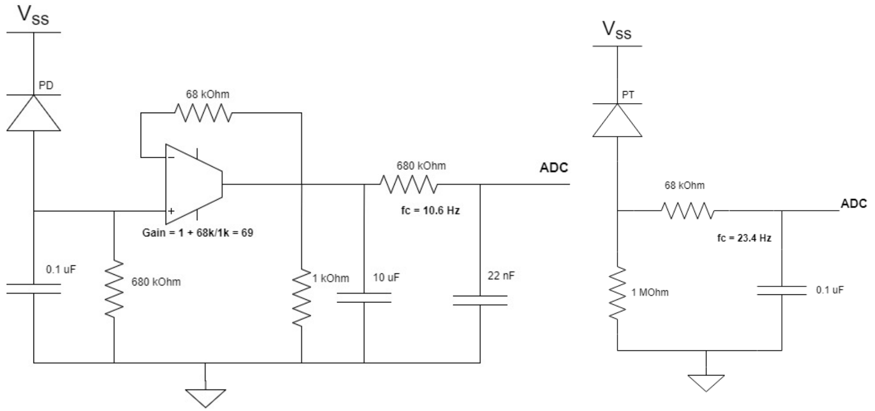
Figure 4.9: Profile of a 3x3 cm wrinkled aluminium foil crossing the full FOV of the sensor (the circular aperture in combination with the TEMT6000X01 phototransistor). The three peaks indicate the foil entering the FOV ring, crossing the center, and leaving the ring, respectively. The FOV width and radius can be derived from this profile.

and at the time of writing they were available for €1.26 and €0.86, respectively. The switching speeds are not mentioned in the data sheets, but similar diodes from the same manufacturer have switching speeds of 100 ns; it is reasonable to believe these have similar speeds. This means a switching frequency up to 10 MHz is achievable, which is much higher than the maximum sampling rate of the Arduino (10 kHz). Therefore, the switching speed does not form a bottleneck.

From this point onwards, the phototransistor is meant whenever photosensor is written. Since the detection area of the phototransistor is smaller compared to the photodiode, this results in a smaller FOV ring width when used with the same aperture. According to the model, this same aperture can be used in combination with the TEMT6000X01 phototransistor to result in a FOV width and radius of 0.78 cm and 15.5 cm, due to the smaller detecting area of the transistor. To verify this, a profile of the 3x3 cm aluminium foil is made with the phototransistor and the circular aperture. This foil is moved throughout the FOV, while the sensor measures the light intensity for each location. The result is shown in Figure 4.9. The detected ring widths are 4.0 and 3.8 cm, which without the foil width results in an actual **FOV width of 0.9 cm for the phototransistor. The FOV radius is 17.3 cm on average.** The combination of the phototransistor with this circular aperture is chosen as the final sensor.

4.4.2 Schematic of the photosensor

Figure 4.10a shows the schematic that was used for the photodiode. The cathode side of the diode is connected to an input voltage (for Arduino, this is 5V). When light reaches the diode, a current will flow towards the anode.



(a) TEMD5510FX01

(b) TEMT6000X01

Figure 4.10: The photosensor schematics for the photodiode (left) and the phototransistor (right).

This current flows through a biasing resistor of $680\,k\Omega$. The voltage over this resistor is amplified with an OpAmp by a factor of 69×Ȧ low-pass filter with a cut-off frequency of 10.6 Hz is added after the OpAmp. Finally, two decoupling capacitors are added to decrease the noise. The resulting voltage serves as an indicator of the light level, and is sent to the ADC input of the Arduino, which samples the measured voltage with a 10-bit range.

For the phototransistor, the OpAmp is not required. The schematic corresponding to this sensor is shown in Figure 4.10b. A biasing resistor with a low-pass filter suffices for the purpose of this research.

Chapter 5

The reflective tag

This chapter describes how to encode data on the reflective tag (Section 5.1), and what encoding scheme is used for this purpose (Sections 5.2).

5.1 Encoding data with light reflections

The reflective surface that is placed on objects can be passive (invariant over time) or active (having the ability to actively modulate reflected light). A passive tag should be a sequence of elements that reflect light in different ways, as for example in Figure 5.1. Only a single light value is detected at a time by the sensor, so only when an object is moving in and out of the FOV, a change in the intensity is observed. The detected pattern received by the sensor from such a tag moving through the FOV is a sequence of valleys and peaks in the measured light intensity.

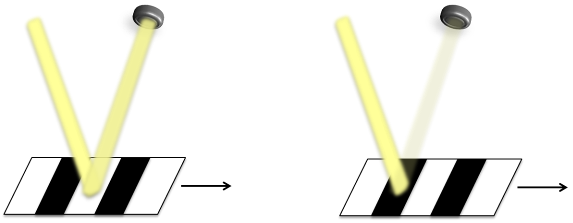


Figure 5.1: Two situations where a barcode, which passively encodes an ID, moves through the FOV on a sensor. The yellow bundles indicates the incoming and reflected light. A sensor observing this barcode will receive a different amount of light based on whatever element moves through the FOV. On the left, a high intensity is detected, while on the right almost no light is reflected to the sensor.

An active tag on the other hand, modulates itself in a certain way resulting in a varying intensity measured by the sensor. In this case, the sensor can also detect an object's ID when this object is standing still within the sensor's FOV. Since one of the main goals of this thesis is to achieve identification and localization in a manner that is as energy-efficient as possible (Section 3.2), a passive tag would be the most straightforward solution.

Since ambient light is used as a medium to transfer an object ID towards a sensor, we need to look at the different properties of light that can be exploited for this purpose. The light properties discussed here are the polarization, the color and the intensity.

5.1.1 Polarization

The polarization of ambient light reflecting on the tag can be modulated with polarization sheets. A sequence of small pieces of polarization sheets with different axes, as shown in Figure 5.2, encodes a pattern in the ambient light that is reflected from this sequence. A sensor equipped with a polarization filter will detect a high or low amount of light, depending on whether the polarization of the incoming light matches the axis of the polarization filter.

Since polarization is invisible to the human eye, this forms a non-invasive method and also adds a layer of security: humans can not see the ID of objects. The drawback is that every sheet diminishes the light intensity (with 50% if the incoming light is unpolarized) and that the detected pattern depends on the orientation of the tag compared to the sensor. There are ways to overcome the latter, as done by the PIXEL indoor localization system [40], but this significantly adds to the complexity and cost of the system. Moreover, while sunlight is unpolarized, this light can become partially polarized after being scattered in the atmosphere [13]. This might make it difficult for a polarization-based system to give the same results indoors and outdoors.

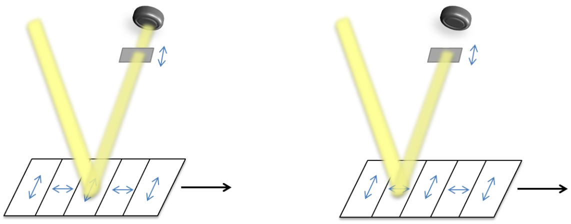


Figure 5.2: Two situations in which a sequence of reflecting polarization filters move through the FOV of a sensor. The arrows on the tags indicate the polarization axis of that element, and the yellow bundles indicate the incoming and reflected light. The sensor is equipped with a polarization filter as well, whose axis is shown with the arrow. In the situation on the left, the sensor is receiving light, while the sensor on the right does not.

5.1.2 Color

Color based encoding schemes have already been adopted, for example by KarTrak to passively encode an ID on trains [28]. The advantage of color based encoding schemes is that a single bit can have numerous values, leading to a higher data density. The disadvantage is the uncontrollability of the color of the ambient light sources. A color based encoding scheme would result in different values under different lighting conditions and no certainty can be drawn from the decoded data. Moreover, single sensors would not be sufficient to detect color, making RGB sensors necessary.

5.1.3 Intensity

A material with different reflectivities or different shades across its length can be used to encode data, as shown in Figure 5.1. As done in [38], a simple barcode with black and white stripes can be used for this purpose. Another possibility is for an object to modulate its shape in such a way that light is reflected into different directions. The data density might be less than for color-based encoding schemes, but the encoding does not depend on the orientation of the object nor the color of the ambient light.

5.1.4 Conclusion

A sequence of reflective and absorbing elements is seen as the most robust way to encode data for this thesis, due to the low dependency on the environment. The reflective elements should be diffuse reflectors, to observe these elements from a broad range of angles (as explained in Chapter 2). The reflectivity of various materials has been compared to figure out what the most suitable material is for such a tag. A sequence of black and white elements (as in [38]) is chosen as the way to encode object IDs, which in this case are made from paper.

5.2 Passive encoding schemes

The naming convention as is used in this thesis is as follows. A barcode constists of consecutive *bits* of black and white paper. Either a bit or a group of bits can be used to encode a data element. The (group of) bit(s) that encodes a data element, forms a *symbol.* A white bit is regarded as a high bit, and a black bit as a low bit.

5.2.1 Single bit

First, it is important to see what the effect is of a single bit on the light detected by a sensor. A white piece of paper is used as a single bit, being in high contrast with the black table surface.

Channel response

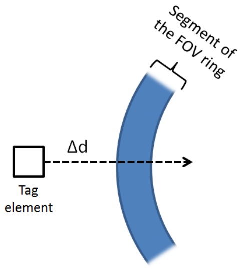
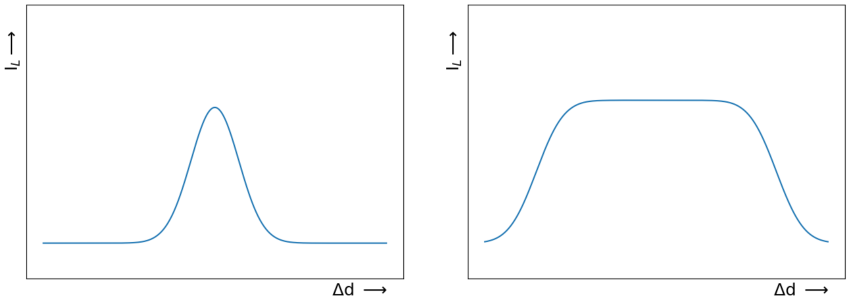


Figure 5.3: A high bit (i.e. a white tag element) moves through a FOV ring segment. The displacement Δd is a measure of location compared to the ring.

The *channel response* of a sensor describes what the effect is of a reflective material moving into the FOV of this sensor. First, only a segment of a sensor's FOV ring will be considered, shown in Figure 5.3. By moving the bit in such a manner, the effect of this element on the measured light intensity is obtained. A high bit with size s (smaller than the FOV ring width w_f) is moved across the FOV, perpendicular to the ring. The light intensity measured by the sensor, as a result from the high bit moving throughout the FOV segment, is found to be of a Gaussian shape. This is also found by [3]. A modelled response is shown in Figure 5.4a, where the detected light intensity is plotted versus the displacement of the bit compared to some reference value. This image does not include any noise, which of course would occur in a real measurement. The sum of w_f and s determines the standard deviation of the Gaussian shape that is observed.

If a bit is larger in size than the FOV ring, the detected signal will be as shown in Figure 5.4b: as soon as the object covers the entire length of the FOV segment, a maximum value is reached. Only when the bit starts leaving the FOV again, the light intensity starts to decrease. In this case, the slope of the rising and falling edges of the detected shape are determined by the FOV ring width. The width of the flat part of the signal is determined by the difference between s and w_f.

Each reflective surface that crosses the FOV, will individually result in a shape as seen in Figure 5.4. When multiple bits consecutively cross the FOV, the observed signal will be the sum of the individual Gaussians: taking into account the offset in distance between the bits. This distance between the bits, in combination with the FOV ring width, will determine what the detected signal looks like. Take for example a barcode consisting of two white bits, with a black bit in the middle. Each white bit results in a Gaussian measured by the sensor. The black bit will have no (significant) impact, since in this case the background is a similar shade of black. If the bits are enough apart, only a single bit will be in the FOV at a single time and two distinct peaks will be detected, see Figure 5.5a. When the bits are too close, inter-symbol interference will occur: both bits will be in the FOV simultaneously when the barcode crosses the FOV. The resulting observation at the sensor is then a single peak, as shown in Figure 5.5c. If

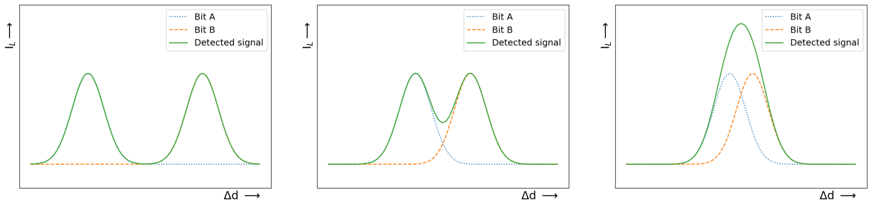


(a) $s \leq w_f$

(b) $s > w_f$

Figure 5.4: Modelled values for the light intensity measured by a sensor as a result from a single high bit moving across a point in the FOV.

the distance between the bits is somewhere in between, a peak as shown in Figure 5.5b is detected. From this image it becomes clear that two high bits *can* overlap a bit within the FOV, as long as the individual peaks are still distinguishable.



(a)

(b)

(c)

Figure 5.5: Each image shows the detected signal from a barcode with two high bits, separated by a low bit, crossing through the FOV ring, shown in green. The signals are broken down into the responses of each individual bit, which are given by the blue and orange lines. The distance between the high bits is decreased from left to right.

Objects are not only observed by the FOV rings, but also by the FOV center. When an object moves through the complete FOV, three stages of detection take place, as shown in Figure 5.6. An object enters the ring, which results in a detection. It then proceeds onto the center of the ring, and is detected here again. A detection in the center results in a higher signal, as one might recall from Figure 4.7a. The object then continues, and is observed a last time upon leaving the ring. If one makes a *profile* of a single high bit traversing the full FOV, it will look like the model in Figure 5.7a. The distances between the peaks are equal to the FOV ring radius. If, instead of a simple bit, a white-black-white barcode crosses the FOV in this

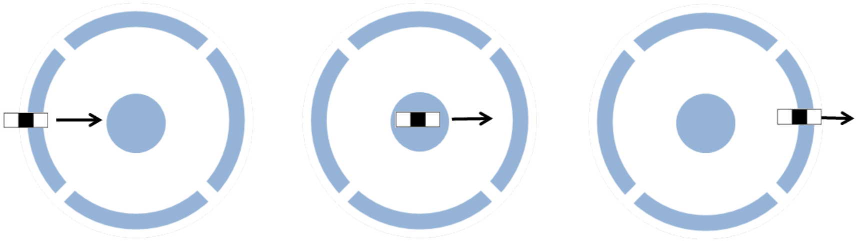
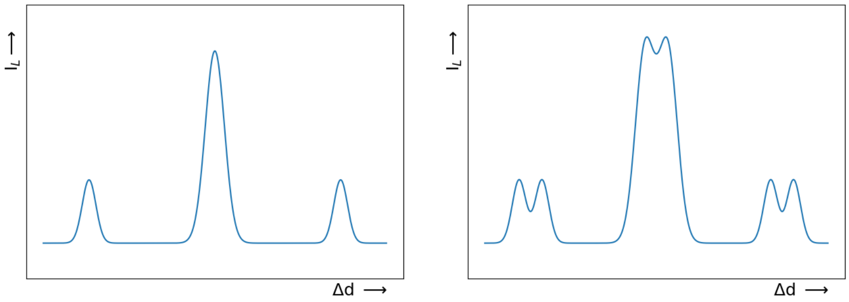


Figure 5.6: The three points at which a barcode is detected when moving in a straight line through the FOV of a sensor. The FOV of the sensor is depicted in light blue; both the ring and center points are shown.



(a) High bit

(b) Barcode

Figure 5.7: A modelled profile of an object traversing the complete FOV of a sensor. The three peaks represent the ring, center and ring detections, respectively.

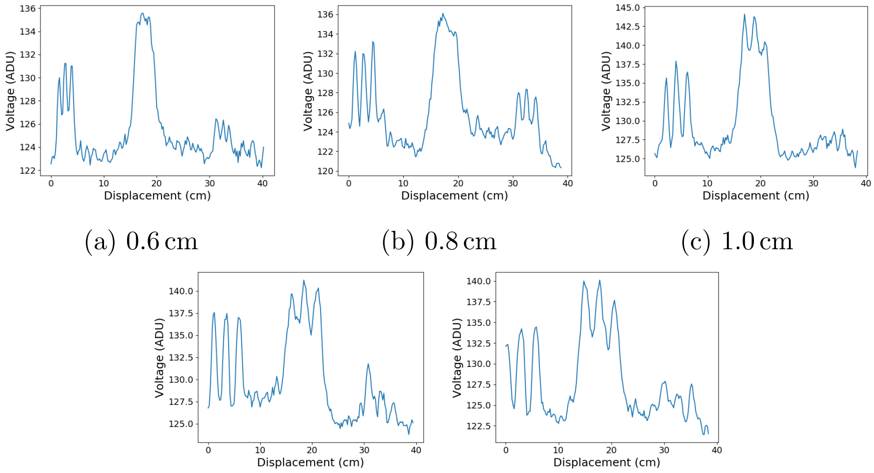
manner, a profile as shown in Figure 5.7b is observed.

The center has a wider detection area compared to the ring, as can again be recalled from Figure 4.7a. Since a center detection has a larger standard deviation, it is harder to distinguish individual peaks in this area.

Minimum bit size

The minimum width of the barcodes depends on both the minimum bit size and the minimum distances between two high bits. For simplicity, these two sizes are chosen to be equal, making the black bits have the same size as high bits. Barcodes with different bit sizes are individually moved through the FOV in the same manner as shown in Figure 5.6. The barcodes have a value of 10101, where a 1 indicates a white (high) bit and a 0 a black (low) bit. Bit sizes of 0.6 cm, 0.8 cm, 1.0 cm, 1.2 cm and 1.5 cm are considered, resulting in barcodes of 3 to 7.5 cm. Figure 5.8 shows the profiles for each size.

The experiments were performed with the circular aperture as described



(a) 0.6 cm

(b) 0.8 cm

(c) 1.0 cm

(d) 1.2 cm

(e) 1.4 cm

Figure 5.8: Profiles for a sequence of alternating high and low bits, with bit sizes as given in the captions.

in Section 4.2 in combination with the TEMT6000X01 phototransistor. The right side of the profile corresponds to a ring segment further away from the center of the LED panel than the left side. Therefore, the right side has a much lower amplitude. Secondly, the measurement results turned out to be quite dependent on the run due to noise. This can be seen in Figure 5.8c, where the right side is not detectable very well. Based on these images, a bit size of 0.8 cm is chosen as a suitable minimum. Smaller bit sizes might be detectable as well when they are moving perpendicularly across the FOV, but when more noise or a rotational offset is introduced the detected peaks might not be decodable anymore.

As described in Chapter 4, the FOV ring width determines the minimum size of a tag element to be distinguishable by the ring. However, if the center point of the FOV is used as well, a larger bit size is required. According to Figure 5.8, a barcode with bit sizes of at least 1.2 cm would be needed to detect with the center as well. In the rest of this chapter, the experiments are carried out for a 0.8 cm bit size, focussing only on ring detections.

5.2.2 Encoding schemes

Constraints

There are several constraints on encoding schemes that can be used in the given set-up. The ambient light level is assumed to be unknown, so an encoding scheme cannot be purely based on the absolute light level of a detection. Secondly, the speed of the objects is unknown and can vary per object. Therefore, there is no simple way for the barcode and the sensor to be synchronized. This means that encoding should not be based on absolute time intervals. If two high bits are located directly next to each other, the sensor will have no way of telling whether the resulting peak is a single long bit or the result of two or more short bits close to each other.

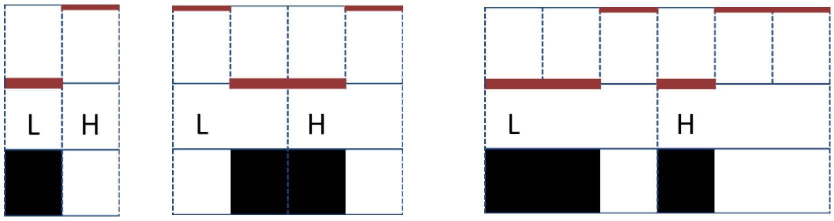
Comparison

The modalities of the barcode that are considered to encode data, are the length of the barcode elements and the number of these elements that encode a symbol. The list below gives an overview of some encoding schemes. The high and low symbol values of these schemes are shown in Figure 5.9.

- **Number of peaks:** In binary encoding schemes each bit represents a symbol, which leads to a high information density. However, since black bits at the edges of a barcode are not detected in the test set-up due to the black table background, N bits can encode only $2^N/4$ IDs. Multiple consecutive bits of the same value cannot easily be decoded since the speed is unknown, so even fewer feasible possibilities are left. The remaining options effectively encode an ID with the number of peaks. This encoding scheme scales with $O(N)$, instead of $O(2^N)$ as compared to a 'regular' binary encoding scheme. For example, with three bits, there are two distinct IDs possible. One of them has one peak (a barcode of 001, 011, 100 etc.), the other has two (101).

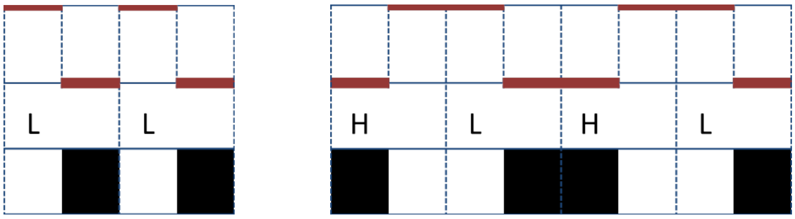
- **Manchester-like encoding:** By implementing an encoding scheme inspired on Manchester encoding, each symbol has a length of two bits. If consecutive symbols have the same value, there is still a change in the light intensity measured. However, both of the barcodes shown in Figure 5.10 would result in two observed peaks for example. Without knowing the speed of the object or the symbol length of the barcode these would not be distinguishable.

- **Relative peak height:** The absolute light levels are not reliable for decoding, but the relative light levels of peaks can be compared to one another (assuming the ambient light level will not change significantly within the time that the object crosses the FOV). With symbols as shown in Figure 5.9, a thinner and lower peak would indicate a low symbol value, and a wider and higher peak a high value. Each barcode should contain at least one of each symbol value for comparison of the peak heights. The symbol length is up to three times higher than the encoding, but is more scalable than encoding with the number of peaks.



(a) Peaks (b) Manchester (c) Relative peak height

Figure 5.9: The high and low symbols for the discussed encoding schemes. The rows indicate the light levels detected from each bit by a sensor, which symbol is encoded and the barcode layout, respectively. The Manchester standard shown here is the IEEE 802.3 standard.



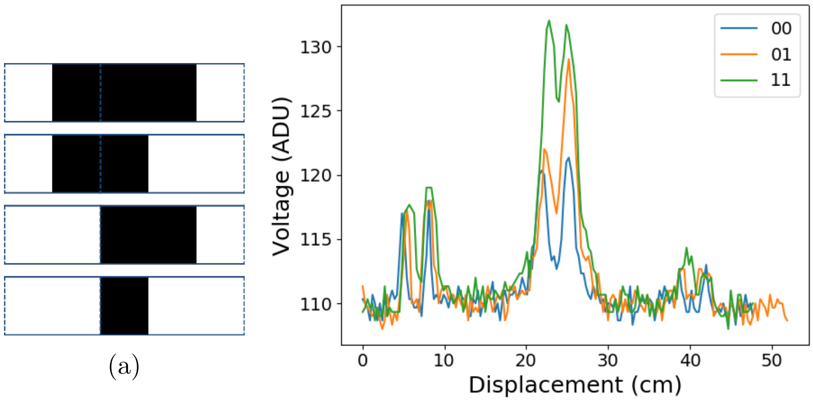
(a) (b)

Figure 5.10: Problems with the Manchester encoding scheme: both barcodes will cause two peaks in the light intensity measured by the sensor. In Figure 5.10a these are two narrower and lower peaks. In Figure 5.10b these are wider and higher peaks (there are two white bits causing the peaks here instead of one). Without any synchronisation or known barcode length there is no way of telling these two barcodes apart from the detected light patterns.

Conclusion

While encoding with the relative peak height as described above might be scalable, there are some practical problem when implementing this in the test set-up. Figure 5.11b shows profiles obtained of the barcodes in Figure 5.11a, which encode four IDs using the relative peak height. For these experiments a bit size of 0.8 cm. Clearly, the peak heights relative to each other are not as expected. The 01 and 11 barcodes become indistinguishable in the left ring part for example. This is most likely due to some non-uniformity within the FOV ring. In Figure 4.9, such a non-uniformity can be seen especially in the left-most peak of the profile: a side-lobe is visible on the peak.

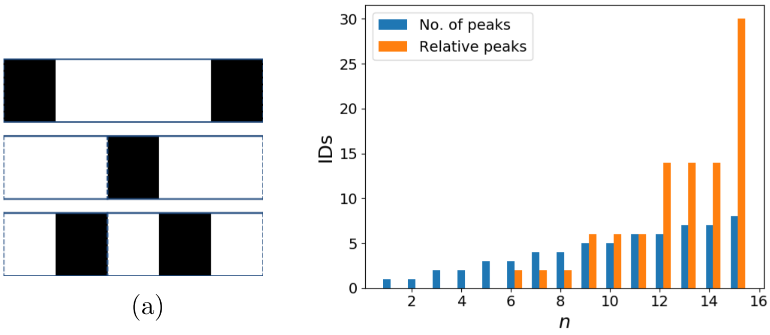
Therefore, encoding based on the number of peaks is used for now, as shown in Figure 5.12a. It is less scalable, as shown in Figure 5.12b, but in the current set-up there are not many robots used anyway and it overcomes the practical issues.



(a)

(b)

Figure 5.11: The four barcodes shown on the left encode four IDs (from top to bottom: 00, 01, 10, 11). When these barcodes move in a straight line through a sensor's FOV (as shown in Figure 5.6), the profiles are obtained as shown on the right.



(a)

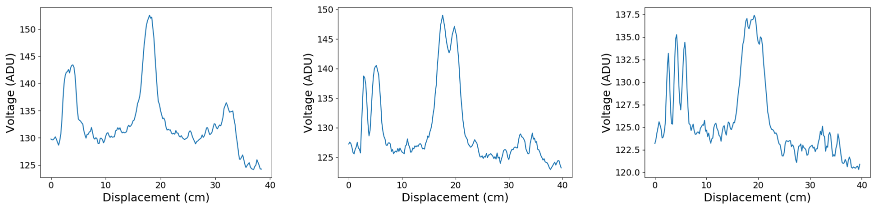
(b)

Figure 5.12: The left image shows the chosen barcodes, consisting of five bits which encode three distinct IDs. Barcodes with ID 1 and 2 are broadened to match the barcode size of ID 3. Barcodes with ID 4 or higher are made by adding a single black and white bit to the barcode with ID 3. The right image shows the number of distinct IDs that can be encoded this way with n bits, compared to encoding with relative peak heights.

5.2.3 Measurement results of the chosen encoding scheme

Detection pattern

The reflected light pattern that the sensor receives when a certain barcode moves through its FOV is measured for the final encoding scheme as shown in Figure 5.12a. Barcodes with an ID of 1, 2 and 3 are considered here, and are moved through the FOV as in Figure 5.3. A bit size of 0.8 cm is used, and a barcode width of 1 cm. The patterns that are observed when each barcode moves through the sensor's full FOV, are plotted in Figure 5.13. As can be seen in these profiles, the ring segment on the right of these images is less sensitive. This is a result of the sensor's FOV not being centered under the LED panel. Instead, this right FOV ring segment receives less light from the panel, and therefore the light that is reflected by a barcode crossing this segment, is less as well.



(a) ID 1

(b) ID 2

(c) ID 3

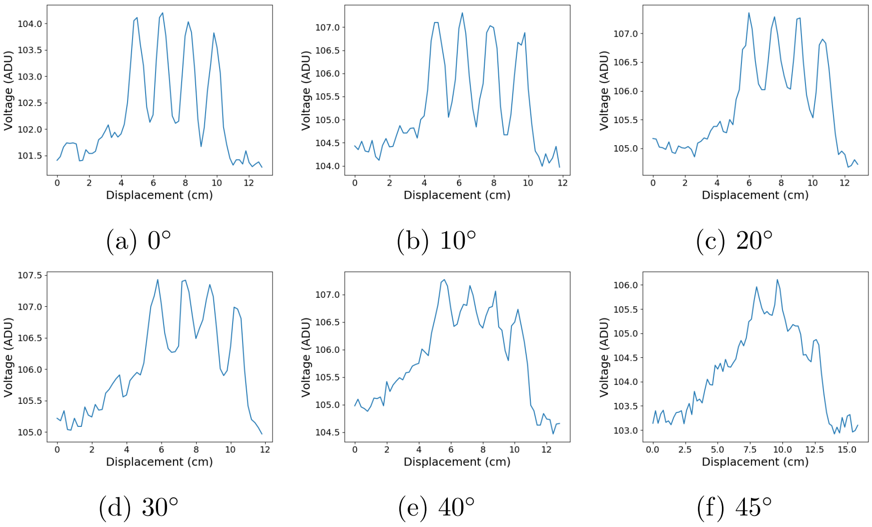
Figure 5.13: Profiles of the three unique barcodes chosen as final IDs.

Influence of rotational offset

When an object does not enter the FOV perpendicularly but under a rotational offset, the peaks with which the ID is encoded might interfere with each other. Figure 5.14 shows a barcode encoding an ID of 4, while crossing a FOV ring segment under different angles. The bit size is 0.8 cm. The measurements were taken for every 0.3 cm of displacement, where each value is an average of 100 samples. The peaks in the figures show to be readily decodable up to an offset of ±30°. For higher rotational offsets it becomes difficult to properly decode the peaks. Just as was seen for the PWM barcodes, the peak heights in Figure 5.14 are not constant. The first peak is consistently the highest, and the last consistently the lowest. Again, this seems to be an effect of non-uniformities in the FOV ring.

Influence of speed

Another aspect that needs to be taken into account is the speed of the object. To determine the impact of the speed, a barcode with ID 3 is attached to the robot of [33]. The robot is moved in a straight line through a



(a) 0°

(b) 10°

(c) 20°

(d) 30°

(e) 40°

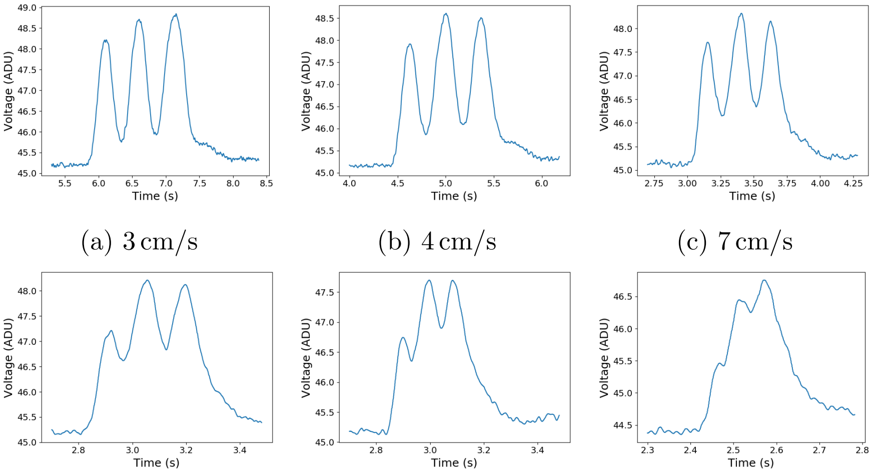
(f) 45°

Figure 5.14: A barcode encoding an ID of 4, for different rotational offsets as indicated in the captions.

sensor's full FOV, for different speeds. Figure 5.15 shows the profile that the sensor detects for each of these speeds. The sampling frequency during these experiments was $F_s = 1833$ Hz, with four sensors being sampled (only the data from one sensor was used for generating the plots). With $F_s = 1833$ Hz and the current set-up, a barcode moving at speed 17 cm/s is still properly detectable. At a speed of 32 cm/s, not enough samples can be taken to properly get the pattern.

The standard speed of the robot is around 20 cm/s. Figure 5.16a shows a profile of the robot with an ID 3 barcode attached to it with bit size 0.8 cm, moving at this speed. The presence of the robot causes some extra disturbances in the signal as can be seen, but in the ring segments the ID is still decodable. The center part of the FOV only occasionally detects the ID. With a minimal barcode size in mind, the 0.8 cm bit size barcode is feasible, and results in a barcode of 4 cm for systems to distinguish three different objects.

However, when the center peaks are desired to be detected as well, a barcode with bit size 1.4 cm is more suitable. A profile of this is shown in Figure 5.16b. For localization of the robot, which will be described in the next chapter, the center point of the FOV provides an enhancement of the accuracy, and therefore the barcode with bit size 1.4 cm is used as final barcode. Besides the extra detection point, the maximum rotational offset and speed are higher as well compared to the 0.8 cm bit size barcode.



(a) 3 cm/s

(b) 4 cm/s

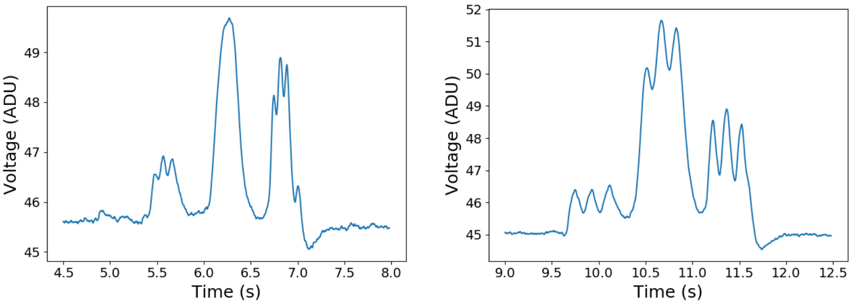
(c) 7 cm/s

(d) 11 cm/s

(e) 17 cm/s

(f) 32 cm/s

Figure 5.15: Profiles for a barcode with ID 3 attached to the robot, moving under a sensor with different speeds. The speed of the robot used in experiments is around 20 cm/s.



(a) Profile of the robot with a 0.8 cm barcode and 0.8 km/h speed.

(b) Profile of the robot with a 1.4 cm barcode and 0.8 km/h speed.

Chapter 6

The algorithm for robot path estimation

This chapter describes the software used to track the robots' paths. After a description of the purpose of the algorithm (Section 6.1), its two main building blocks are discussed: identification of barcodes (Section 6.2) and localization of these barcodes using particle filters (Section 6.3). Lastly, the results are shown (Section 6.4).

6.1 Objective of the robot path estimation

The tracking system has a trade-off between the number of sensors used and the computational complexity needed to estimate the path of objects, as mentioned in Chapter 3. The test set-up has four photosensors, whose FOVs cover the area as indicated in Figure 6.1. Robots are given a barcode as shown in Figure 5.12a, with a bit size of 1.4 cm, encoding up to three different IDs. The total barcode is 7 cm long. The robots move through the illuminated area. To estimate the location of the robots when they are not in a sensor's FOV, an algorithm is needed. The purpose of this software is to read voltages measured by the photosensors in the test set-up, and output a display showing the estimated real-time location of detected objects (i.e. robots).

The algorithm consists of two parts. The first part is detection and identification of (an object with) a barcode crossing the sensor's FOV. Each detection indicates what barcode has been detected by what sensor and when. The speed from the detected barcode can be derived as well. The second part of the software uses the combination of these IDs, timestamps, speeds and sensors to estimate the barcode location with particle filters. The following sections describe these two components in detail.

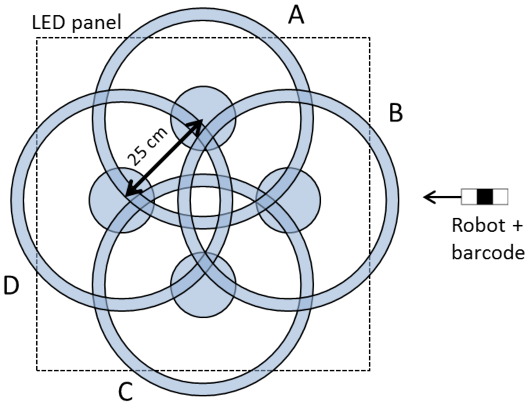


Figure 6.1: A top view of the table surface used for the test set-up. The black dotted box indicates the size of the LED panel above the table surface. The four rings and circles represent the FOV of the four photosensors pointed to the table surface. An object with a barcode moving on the surface will cross multiple FOVs.

6.2 Barcode identification

The identification algorithm converts a stream of voltages measured by a sensor to the list of detected barcodes, with corresponding timestamps and speeds.

6.2.1 Barcode detection algorithm

Filtering the measurements

The first step in the algorithm is removing noise from the signal. For this purpose, the raw data received on the PC is filtered real-time using a moving average over $k = 15$ samples. Subsequently a 30 Hz lowpass-filter of order 80 further smooths the signal. Figure 6.2 shows the raw and filtered signal of a measurement. The measurement is a robot passing through the FOV of a sensor, with the an ID 3 barcode attached to it. As stated before, this barcode has a bit size of 1.4 cm, and total length of 7 cm.

Clustering peaks

The second step is to detect and group peaks within the smoothed signal. Detection of peaks is done with the peak detection algorithm from [10]. From the resulting list of peak values, the algorithm looks for peaks that 'belong together' and are the result of a single barcode. For this purpose, the time distance between the peaks is considered. If the distance from a certain peak A to its neighbour B is significantly smaller than the distance

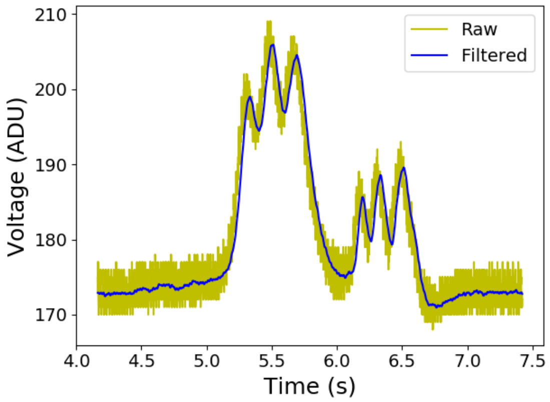


Figure 6.2: The raw and filtered signal received by the phototransistor enclosed with the aperture from Chapter 4 in the test set-up, while a robot with an ID 3 barcode crosses the FOV with a 15 cm/s speed.

from this neighbour B to its next neighbour C, peak A and B probably belong together. C is most likely the start of a new barcode. On the other hand, if the distance between A and B is much larger than the distance between B and C, it is more likely that B and C belong to a barcode that did not cause peak A. If instead similar distances are found, A, B and C are assumed to belong together. With this reasoning, peaks are clustered into groups that each represent a barcode.

Whenever the time difference between two peaks exceeds a threshold, the peaks are assumed to not belong together. This threshold is computed as 1.5 times the expected time difference between peaks for a given average speed of the object. If for example the robots move with an average speed of 15 cm/s, and the barcode bits are 1.4 cm in size, the expected time difference between peaks is $2 \times 1.4/15 = 187$ ms.

Computing the speed of the barcode

The last step is to compute the speed of a barcode from the time difference between the detected peaks. Since both the bit sequence of a particular ID and the bit size are known for the given system, not only the temporal but also the spatial distance between the peaks is known. Dividing the distance by the time difference, the speed of the object is calculated. This will only not work for ID 1, since there is only peak and no difference between peaks can be computed.

For every detected barcode, the ID, timestamp, speed and sensor number is stored. If a barcode has already been detected before, the 'history' of this barcode object is updated with the new info.

6.3 Location estimation with particle filters

By (not) detecting a barcode at a particular sensor, information is gained about whether the barcode is likely to be in or out of this sensor's FOV. To obtain an estimate of the location of such a barcode based on this information from several sensors, particle filters are used.

6.3.1 Particle filters

Particle filters, also known as Sequential Monte Carlo methods, are a set of filters that are widely used in applications requiring tracking of some nonlinear value [39], [15], [24]. The basic structure of a particle filter starts with randomly generating a large amount of virtual 'particles' and giving each of them an equal weight. These particles represent the attribute or the object that is being tracked. In case of localization, many particles are for example distributed on a known map. Each of these points has a position, velocity and orientation, referred to as their *state*. The value of these parameters are drawn from some initial distribution.

After obtaining this initial distribution, an iterative process follows. This process *predicts* for all particles a new state (position, velocity and orientation), based on the old parameters and the expected movement of the object that is being tracked. The set of equations indicating the expected movement of the object is called the *motion model*. Some noise is always added to the predicted values, in case the motion model is flawed. This prevents the particles from converging to an incorrect value. After the prediction, the process *updates* the weights of the particles, depending on how well they match the measurement.

After the update, the particles are *resampled* if needed. Resampling means that particles with a weight that is too low are discarded and new particles are placed instead. There are various ways to do resampling and to choose which particles to choose in their place.

6.3.2 Implementation

In this thesis, each object is modelled by 4000 particles that have an x and y position, an orientation θ and a speed v. Initially, the particles are given a random position across the 'map', which is the square area from Figure 6.1. Their speeds are set to an initial guess v_{avg}, which is assumed to be the average speed of the object, plus Gaussian noise. Here, the robot is set to move with a speed around 15 cm/s, so this will be taken as an initial guess. Note that this is a favorable choice, the system should be tested with outlying initial conditions as well. The orientations assigned to the particles are randomly chosen between 0 and 360°.

The motion model, which is used to predict the next state of the particles, is defined as follows:

$$v_{\text{new}} = v_{\text{current}} + \sigma_v R, \tag{6.1}$$

$$\theta_{\text{new}} = (\theta_{\text{current}} + \sigma_\theta R) \mod 360, \tag{6.2}$$

$$x_{\text{new}} = x_{\text{current}} + v\cos(\theta)\delta t, \tag{6.3}$$

$$y_{\text{new}} = y_{\text{current}} + v\sin(\theta)\delta t. \tag{6.4}$$

In Equations 6.1-6.4, R denotes a random number from the standard normal distribution, and σ_v and σ_θ the standard deviation of the speed and orientation, respectively, and δt the amount of time passed since the last predict step.

With the current set-up of four sensors, the sampling frequency is $F_s = 1884\,\text{Hz}$. The particle filter is not updated on each sample, but instead every 120 samples, so with a frequency of approximately 16 Hz. The values of the standard deviations have been estimated by measuring the standard deviation of the orientation and speed of a robot moving in a straight line. The values have been fine-tuned empirically, and optimal tracking was observed for $\sigma_v = 0.05$ and $\sigma_\theta = 20$.

After prediction the next state of the particles, their weights are updated based on the measurement. Whenever an object is detected by a certain sensor, the weights of all particles within the FOV of this sensor are multiplied by a factor of seven, which was again determined empirically. If an object is not detected by any sensor, the weight of all particles that are not in the FOV of any sensor are multiplied by a factor of 1.05. The chosen resampling method is multinomial resampling, and is taken from [20].

6.4 Results

6.4.1 Barcode identification

To quantify the accuracy of detecting a barcode, a robot is moved eight times in a straight line through a full FOV of a circular aperture with the phototransistor. The robot is equipped with a barcode encoding the value 3, and moves at different speeds (15.4 cm/s and 17.1 cm/s). Table 6.1 shows how often the barcode is detected correctly, in the center and both ring segments that are traversed by the robot. Ring segment A indicates the ring segment of the sensors that is the closest to the center of the panel. Ring segment B denotes the opposing segment, the furthest from the center of the panel.

v (cm/s)	Ring side A	center	Ring side B	v_{est} (cm/s)
15.4	87.5%	100%	37.5%	15.6
17.1	100%	87.5%	50%	17.6

Table 6.1: Detection accuracy of barcode 3 throughout five different runs, with the object moving at different speeds v. The speed estimated by the detection algorithm is indicated by v_{est}.

6.4.2 Localization estimation

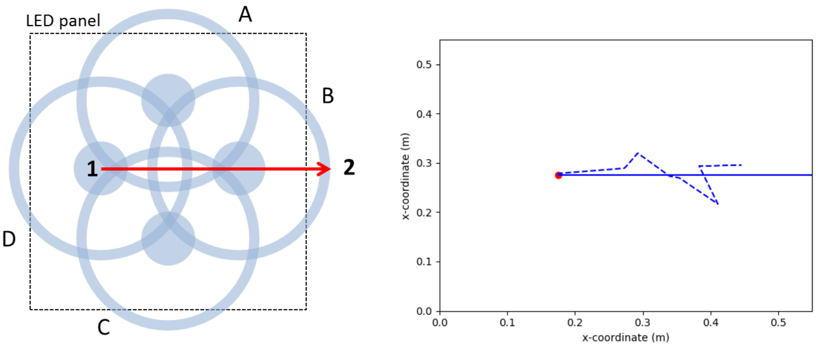
Tracking an object moving in a straight line

To show the performance of the resulting system, first only one robot with ID 3 is moved several times through the area with a speed of 15.4 cm/s in a straight line from points 1 to 2, as indicated in Figure 6.3a. An example of the estimated location is shown in Figure 6.3b. The average error in the real and estimated location for various runs of this same path, is 4.8 cm. When the experiment is repeated for a similar path, but for example between sensors A and C, instead of D and B, show similar results.

Tracking an object with non-linear trajectories

To further evaluate the performance of the tracking system, it is important to test the localization error for different trajectories, such as a circle or a random walk. However, a consistent problem that arose during these experiments, if that an object will cross the FOV too often with a rotational offset that prevents the ID to be detected. Instead, a single peak is seen, from which no direct conclusion can be drawn about the actual ID. A similar problem occurs where an object moves through the edge of the FOV center, leading to a detection of a single peak as well. Section 7.2 discusses several pointers on how this problem can be overcome in future work.

The algorithm is written in Python and can be found on [36].



(a) (b)

Figure 6.3: The left image shows the path of the ID 3 barcode through the FOVs, indicated by the red arrow. The right image shows both the real and an estimated path, shown by the continuous and dotted line, respectively. The red dots indicated the starting and ending points of the path.

Chapter 7

Conclusion and future work

7.1 Conclusion

This thesis shows a proof-of-concept of a passive light-based localization system for two-dimensional, real-time tracking of small-sized objects. Based on the concept of [38], objects are equipped with a unique reflective tag (a barcode), from which ambient light is reflected in a distinct manner. These light reflections are detected by simple photosensors, from which the object ID and an estimate of the location are obtained.

Whereas previous work on ambient passive light localization methods did not focus on small-sized objects and two-dimensional tracking, this thesis addressed two challenges to overcome these open problems. Tackling the first challenge requires detection of small-sized objects. For this purpose, a physical enclosure is designed for the photosensors that modifies the FOV in such a way that a small object can be resolved, without compromising the outer boundary of the FOV. The apertures have been designed such that the smallest distinguishable size of an object in the test set-up is 0.8 cm.

This size corresponds to the minimum bit size that a barcode crossing the FOV can have. This means that small barcodes can be used for distinguishing a modest group of objects. Several encoding schemes have been compared, after which an encoding scheme is chosen where up to three IDs can be distinguished with 4 cm barcodes, if the aim is to keep the barcode size as small as possible. If a 1.4 cm bit size is used, the barcode sizes for 3 IDs becomes 7 cm, but the full FOV of the photosensor can then be used for detecting this barcode.

The second challenge requires a software filter to combine measurements from different sensors, and make an estimate of the location of an object. Particle filters have been implemented and tested with barcodes with a bit size of 1.4 cm, and a total size of 7 cm. Four photosensors are placed in the illuminated test environment. The robots from [33] are provided with the barcodes, and move through the FOVs. With the robots moving at a speed of 15.4 cm/s in a straight line through the test set-up, a localization error of 4.8 cm is obtained if the objects cross through the centers of the FOVs. When objects only partially cross the FOV, the tracking error increases. Some limitations were encoutered due to which other trajectories are not properly localized. As a result, the system can localize objects with a one-dimensional movement, however, this movement can have various angles in a two-dimensional area.

7.2 Future work

Several topics of interest can be studied further, to improve or extend the tracking system proposed in this thesis.

7.2.1 Tracking more complex movement of objects

To track different types of movements of the robots, besides a straight line, some changes need to be made to the algorithm or the sensor. The main limitation in the current set-up that prevented trackign of non-linear movement is the detection. When an object does not enter a FOV ring perpendicularly, but instead parallel to the ring, or under a large rotational offset, the reflection pattern from the object is often detected as a single peak. This can (partially) be solved by for example a different types of apertures such as the one shown in Figure 7.1. Figure 7.1b has the advantage that if each detection point is small enough to observe only a single barcode element, there is no rotational dependency on the detected signal, at the cost of a significantly lower coverage area. Figure 7.1c would be able to detect barcodes moving in perpendicular directions, but is much less sensitive to signal at the edges of the cross compared to the center. Another idea would be to make the algorithm 'smarter', and find a way to reason whether a single detected peak is a barcode with ID 1 or an unresolved barcode with other ID (e.g. if shortly before or after an ID 3 is detected in a neighbouring sensor, while no ID 1 was observed in that area).

7.2.2 Increase distance between objects and sensor

In order to use the described set-up in a real-life environment, the distance between the sensor and the objects needs to be increased significantly. However, when the height is increased, not only a much lower light intensity is received, but a small FOV ring width becomes more challenging as well. And with this, a small barcode size cannot be maintained. With the current aperture, a thinner aperture slit is not easily manufacturable. Instead, an aperture could be made by lasercutting instead, or a lens could be added to still focus on a small FOV. In any case, the received signal should be

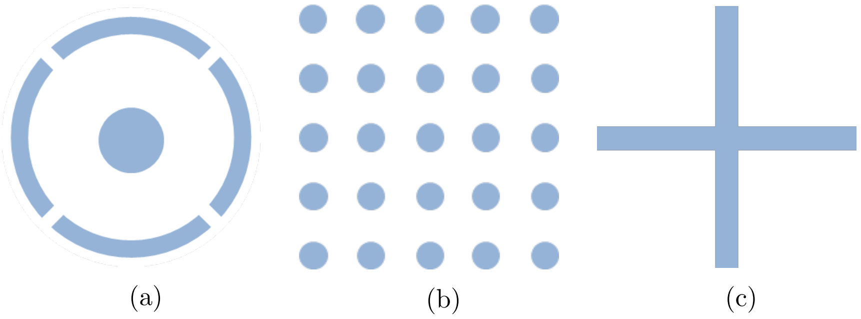


Figure 7.1: These images show the FOV of possible sensors. The left FOV is the sensors as discussed in this thesis, the other two images show FOVs that have a potential to improve the detection, but at the expense of a smaller coverage.

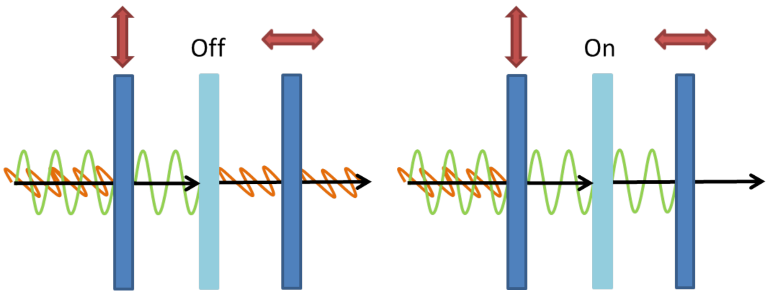
amplified and careful removal of noise sources in the environment is needed to make such a system work.

7.2.3 Active tag encoding

In contrast to a passive tag, an active tag can enable an object to change its ID when desired (e.g. when encountering something in the environment). Moreover, an object actively sending out its ID can be detected while standing still, and the small FOV width constraint can be lifted. For some applications, these points may outweigh the increased energy usage and infrastructure.

A suitable material for active encoding is a liquid crystal (LC) shutter. Essentially, such a shutter is a glass plate plate that switches between transparent and opaque states on the application of a voltage. The working of such a shutter is shown in Figure 7.2. It consists of two polarizing sheets, with a perpendicular polarization axes. They are separated by a thin layer of liquid crystal, which turns the polarization of incoming light by 90°, unless a voltage is applied. Unpolarized light entering an LC shutter is filtered by the first polarization sheet, and only light oscillating parallel to the polarization axes is transmitted. Then, depending on the voltage over the liquid crystal, the polarization of the transmitted light is rotated or not. Then these light rays continue onto the second filter, which either blocks or transmits them, depending on whether they match its polarization axis. These shutters are used in for example LCD screens, but can also have useful applications in light-based communication [40].

Currently, work is being done at the Embedded Software group of Delft University of Technology, where LC shutters are tested for the purpose of setting up a sunlight-based communication system [29]. Some initial research has been done in this thesis, based on the initial research done by



(a) (b)

Figure 7.2: An LC shutter: transparent (left) and opaque (right).

[29], regarding the feasibility of using the shutters in combination with the robot from [?]. The refresh rate was found to be high enough for different encoding schemes, and the energy used by the shutter is low enough to fit in the energy budget of the robot. The results of the energy measurements are shown in **??**, and are dependent on the operating frequency. If the toggling frequency is increased, the consumed energy will increase too. The total consumption of the robot is in the range of [29,52] mA , of which the motors use between [27,50] mA. The round shutter flickering at 60 Hz with 3.3V voltage swing would consume only 0.12% of the robot's total energy budget if placed on the robot, and switches fast enough to encode an ID in reasonable time. From this we can conclude that energy consumption will not be a bottleneck for implementation of an LC shutter as active ID encoder. Combining the shutter and the robot has not been done for this project, but based on results from [29] this can be done as future work.

Frequency (Hz)	**Current (μA)** *Round shutter*	**Current (μA)** *Rectangular shutter*
8	5.9	3.3
20	12.9	7.7
60	34.4	22.6

Table 7.1: Current drawn with 50% PWM signal at 3.3V.

7.2.4 Other directions

Other topics that form interesting related research are for example improving the scalability of the tag and collision detection (when multiple object simultaneously cross the FOV). Another topic is detection whenever an object is standing still. Currently, if the light reflected from such a object is high enough, this results in continuous peak detection and therefore incorrect decoding to barcode values.

Bibliography

[1] Led panel 60×60 cm, 40w, 4000 lumen, ordered through `www.ledlampaanbiedingen.nl`, 2017.

[2] Mohammad Aqel, Mohammad Hamiruce Marhaban, M Iqbal Saripan, and Napsiah Ismail. Review of visual odometry: types, approaches, challenges, and applications. Vol 5, Dec 2016.

[3] Rens Bloom. Channel analysis for passive communication with ambient light. Master's thesis, Delft University of Technology, the Netherlands, 2017.

[4] A. Boletis, W. Driesen, J. m. Breguet, and A. Brunete. Solar cell powering with integrated global positioning system for mm3 size robots. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5528–5533, Oct 2006.

[5] Antoni Burguera, Yolanda González, and Gabriel Oliver. Sonar sensor models and their application to mobile robot localization. *Sensors*, 9(12):10217–10243, 2009.

[6] G. Caprari and R. Siegwart. Mobile micro-robots ready to use: Alice. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005.

[7] C. Chen, Y. Chen, H. Q. Lai, Y. Han, and K. J. R. Liu. High accuracy indoor localization: A wifi-based approach. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6245–6249, March 2016.

[8] Yang Cheng, M. W. Maimone, and L. Matthies. Visual odometry on the mars exploration rovers - a tool to ensure accurate driving and science imaging. *IEEE Robotics Automation Magazine*, 13(2):54–62, June 2006.

[9] Artem Dementyev, Hsin-Liu (Cindy) Kao, Inrak Choi, Deborah Ajilo, Maggie Xu, Joseph A. Paradiso, Chris Schmandt, and Sean Follmer. Rovables: Miniature on-body robots as mobile wearables. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 111–120, New York, NY, USA, 2016. ACM.

[10] Marcos Duarte. Notes on scientific computing for biomechanics and motor control. `https://github.com/demotu/BMC/blob/master/functions/detect_peaks.py`, 2014.

[11] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine*, 13(2):99–110, June 2006.

[12] Guoyu Fu, Jin Zhang, Wenyuan Chen, Fengchao Peng, Pei Yang, and Chunlin Chen. Precise localization of mobile robots via odometry and wireless sensor network. *International Journal of Advanced Robotic Systems*, 10(4):203, 2013.

[13] Eugene Hecht. *Optics*. Addison-Wesley, Reading, Mass, 2002.

[14] H. Hosseinianfar, M. Noshad, and M. Brandt-Pearce. Positioning for visible light communication system exploiting multipath reflections. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017.

[15] Y. Huang, S. Zhang, and Y. Jing. An indoor mobile localization strategy based on particle filter in nlos environment. In *2016 Chinese Control and Decision Conference (CCDC)*, pages 6518–6522, May 2016.

[16] Minkuk Jung and Jae-Bok Song. Efficient autonomous global localization for service robots using dual laser scanners and rotational motion. *International Journal of Control, Automation and Systems*, 15(2):743–751, Apr 2017.

[17] M. Karimi, A. Ahmadi, P. Kavandi, and S. S. Ghidary. Weemik: A low-cost omnidirectional swarm platform for outreach, research and education. In *2016 4th International Conference on Robotics and Mechatronics (ICROM)*, pages 26–31, Oct 2016.

[18] N. Y. Ko and T. G. Kim. Comparison of kalman filter and particle filter used for localization of an underwater vehicle. In *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 350–352, Nov 2012.

[19] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis. Human interaction with robot swarms: A survey. *IEEE Transactions on Human-Machine Systems*, 46(1):9–26, Feb 2016.

[20] Roger Labbe. Kalman and bayesian filters in python. `https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python`, 2018.

[21] E. Di Lascio, A. Varshney, T. Voigt, and C. Perez-Penichet. Poster abstract: Localight - a battery-free passive localization system using visible light. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–2, April 2016.

[22] Mathieu Le Goc, Lawrence H. Kim, Ali Parsaei, Jean-Daniel Fekete, Pierre Dragicevic, and Sean Follmer. Zooids: Building blocks for swarm user interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 97–109, New York, NY, USA, 2016. ACM.

[23] Philips Lighting. Understanding philips indoor positioning `www.lighting.philips.com/main/inspiration/smart-retail/latest-thinking/indoor-positioning-retail`, 2017.

[24] G. Ligorio and A. M. Sabatini. A particle filter for 2d indoor localization relying on magnetic disturbances and magnetic-inertial measurement units. In *2016 IEEE Sensors Applications Symposium (SAS)*, pages 1–6, April 2016.

[25] J. Liu, Y. Chen, A. Jaakkola, T. Hakala, J. Hyyppä, L. Chen, J. Tang, R. Chen, and H. Hyyppä. The uses of ambient light for ubiquitous positioning. In *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, pages 102–108, May 2014.

[26] R. C. Luo and T. J. Hsiao. Dynamic wireless indoor localization incorporate with autonomous mobile robot based on adaptive signal model fingerprinting approach. *IEEE Transactions on Industrial Electronics*, pages 1–1, 2018.

[27] M. Mukherjee. Visible light communication-a survey of potential research challenges and advancements. In *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–8, Feb 2017.

[28] B. Nelson. *Punched Cards to Bar Codes: A 200 Year Journey.* Helmers Publishing Company, 1997.

[29] Chaitra Pai. Passive communication using static objects (in progress). Master's thesis, Delft University of Technology, the Netherlands, 2018.

[30] S. Rajagopal, R. D. Roberts, and S. K. Lim. Ieee 802.15.7 visible light communication: modulation schemes and dimming support. *IEEE Communications Magazine*, 50(3):72–82, March 2012.

[31] B. Ramirez, H. Chung, H. Derhamy, J. Eliasson, and J. C. Barca. Relative localization with computer vision and uwb range for flying robot formation control. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1–6, Nov 2016.

[32] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. In Erol Şahin and William M. Spears, editors, *Swarm Robotics*, pages 10–20, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[33] Koen Schaper. Transiently-powered battery-free robot. MSc thesis, Delft University of Technology, December 2017.

[34] H. Stoll, P. Zimmer, F. Hartmann, and E. Sax. Gps-independent localization for off-road vehicles using ultra-wideband (uwb). In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, Oct 2017.

[35] W. Su, A. Ravankar, A. A. Ravankar, Y. Kobayashi, and T. Emaru. Uav pose estimation using ir and rgb cameras. In *2017 IEEE/SICE International Symposium on System Integration (SII)*, pages 151–156, Dec 2017.

[36] Danielle van der Werff. Software for passive ambient light-based tracking. `https://github.com/dnvanderwerff/passivetracking.git`, 2018.

[37] Enge Per van Diggelen, Frank. The world's first gps mooc and worldwide laboratory using smartphones. In *Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, pages 361–369, Tampa, Florida, 2015.

[38] Qing Wang, Marco Zuniga, and Domenico Giustiniano. Passive communication with ambient light. In *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '16, pages 97–104, New York, NY, USA, 2016. ACM.

[39] Xuedong Wang, Tiancheng Li, Shudong Sun, and Juan M. Corchado. A survey of recent advances in particle filters and remaining challenges for multitarget tracking. *Sensors*, 17(12):2707, 2017.

[40] Zhice Yang, Zeyu Wang, Jiansong Zhang, Chenyu Huang, and Qian Zhang. Wearables can afford: Light-weight indoor positioning with visible light. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '15, pages 317–330, New York, NY, USA, 2015. ACM.

[41] M. Yasir, S. W. Ho, and B. N. Vellambi. Indoor position tracking using visible light. In *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, pages 1–5, Sept 2015.

Appendix A

Aperture model

This section describes how an aperture influences the FOV of a photosensor. From a given set of aperture parameters, the FOV width and radius can be computed. The aperture is shown in Figure 4.2. The FOV on the surface is assumed to be delimited by the two light rays shown. The distance between these lines and the center of the FOV ring, right under the sensor, is computed using similar triangles.

The similar triangle used to compute the inner FOV radius is depicted in green in Figure A.1. The corresponding formula is shown in Equation A.1. The subscript i is used to denote an inner radius, the subscript o is used for an outer radius. The subscripts f, a and p indicate whether a quantity belongs to the FOV, the aperture or the photosensor respectively. Furthermore, the symbols w, l and t denote the width, length and thickness. The sensor height is shown as h.

$$r_f^i = \frac{r_a^i - \frac{w_p}{2}}{l_a - l_p - t_a} \cdot (h - l_p) + \frac{w_p}{2}. \tag{A.1}$$

The outer radius of the FOV is computed analogously, using the similar triangles as depicted in blue in Figure A.2, as is described by Equation A.2.

$$r_f^o = \frac{r_a^o + \frac{w_p}{2}}{l_a - l_p} \cdot (h - l_p) - \frac{w_p}{2}. \tag{A.2}$$

The difference between the outer and inner radii of the FOV equals the FOV width.

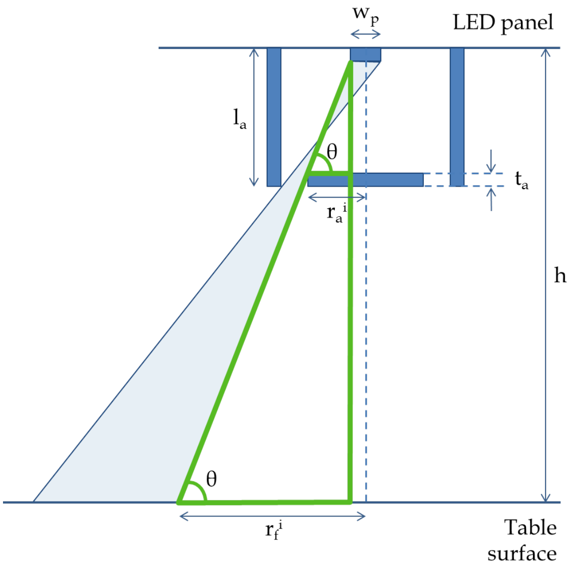


Figure A.1: The two similar triangles used to compute the inner radius of the FOV, shown in green.

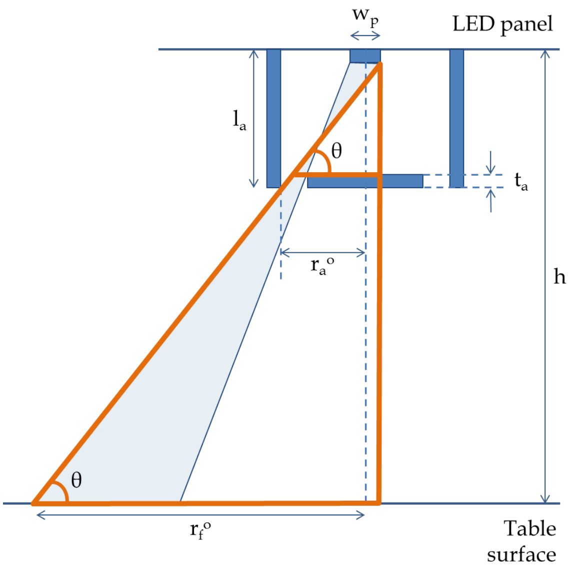


Figure A.2: The two similar triangles used to compute the outer radius of the FOV, shown in blue.