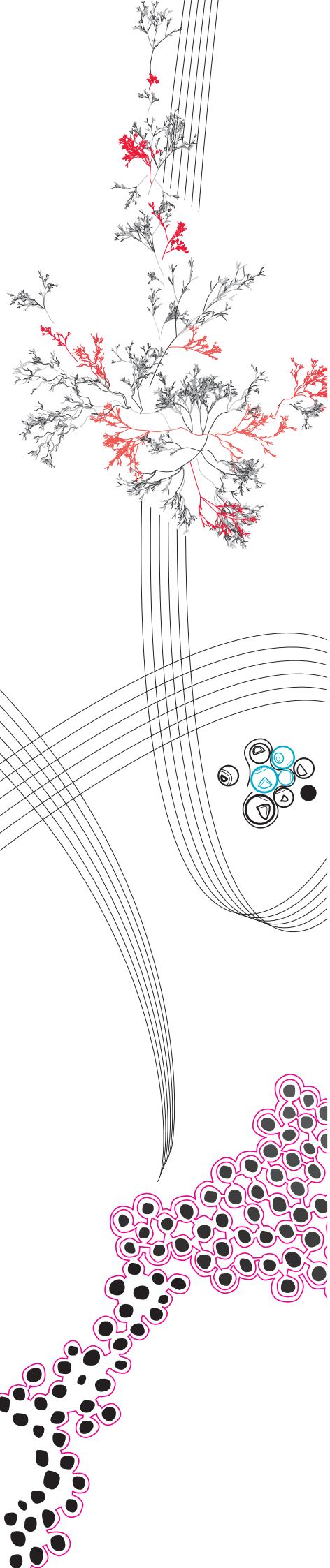


MASTER THESIS

PRACTICAL CYBER-ATTACKS ON AUTONOMOUS VEHICLES

Bas G.B. Stottelaar

Faculty of Electrical Engineering, Mathematics and Computer Science
Services, Cybersecurity and Security Research Group

Committee:
Prof. Dr. Frank Kargl
Prof. Dr. Ir. Raymond Veldhuis
Dr. Jonathan Petit
Dipl.-Inf. Michael Feiri

May 4, 2015
Version: 2123ddc

UNIVERSITY OF TWENTE.

ABSTRACT

This thesis explores the field of Autonomous Vehicle (AV) sensor technologies and potential cyber-attacks on sensors. The research on AVs is increasing tremendously, as the first vehicles are due to hit the road by 2020. Unfortunately, the literature on cyber-attacks on AVs is limited and theoretical. The first part of this work addresses the available sensor technologies, including limitations, attacks and countermeasures. Examples of sensor technologies include Laser Image Detection and Ranging (Lidar), Tire-pressure Monitoring System (TPMS) and Global Navigation Satellite System (GNSS). In the second part of this thesis, practical attacks on the hardware layer of Lidar and camera sensors will be demonstrated on actual hardware (MobilEye C2-270 Advanced Driver Assistance System (ADAS) and ibeo LUX 3 Lidar system). Camera-related attacks include blinding and auto controls confusion attacks. The Lidar attacks include jamming, relaying and spoofing attacks. The attacks are evaluated according to an external attacker model with limited money and knowledge. The experiments are proof-of-concept, and are conducted in a lab environment. It was found that the MobilEye C2-270 is sensitive to low-cost near-infrared light sources, but these light sources cannot blind it. However, a low-budget low-power visible lasers can. The Lidar was susceptible to jamming, relay and spoofing attacks using low-cost hardware. Counterfeit signals can also influence the tracking software. Three examples of the impact of the attacks on the application level have also been shown, including an attack on sensor fusion. The last section of this work discusses several countermeasures that can mitigate or limit the demonstrated attacks.

ACKNOWLEDGEMENTS

Without the enthusiasm of my supervisors I would never haven chosen this topic. Their way of thinking helped me a lot, and got me through the easy and hard times. When I first contacted Jonathan and Michael to talk about this thesis topic, I was told that it would be hard and that the outcome would be unknown. Nevertheless, it would be a very practical topic and the result would be eye-opening. Thanks to you guys, I got the possibility to challenge myself and show what I could do. I hope that I have properly wrote down all the practical things I did.

Furthermore, many thanks to my girlfriend Mirjam, family and friends for supporting and providing feedback. Even though my due date changed many, many, many times, you still kept believing in what I did, even in hard times. This will mark the end of my study career at the University of Twente. Special thanks to Dirk, Kevin, Lambert, Marijn and Ties for helping me to improve the text. I have concluded that I am better at writing program code than text.

Also, many thanks to Geert Jan Laanstra. He provided me a place to work for more than one year, and his knowledge was valuable while reverse engineering the Lidar, doing the measurements and debating camera countermeasures. It is also thanks to him that I can still use my eyes. Playing with lasers can be very dangerous!

Finally, I would like to thank V-Tron B.V. in Deventer and Ibeo Automotive Systems GmbH in Hamburg for providing the MobilEye C2-270 and ibeo LUX 3 to experiment with. Without their generosity, this work would not have been possible. I had fun playing with the devices and find attack possibilities while simultaneously reverse engineering the hardware in terms of operation.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 PROBLEM STATEMENT

When the first 'World Wide Web' server was put online in 1991 by Tim Berners-Lee, he would certainly not have expected that Cybercrime would be such an issue as it is today. The same was probably true, when the first Autonomous Vehicle (AV) was invented back in the eighties, even before the internet was invented. Initial research projects such as Stanford's autonomous line following robot named 'Cart' (1970), can be considered preliminary work of current automated vehicles. It was not up to 1986 before the first car, named 'VaMoRs', was driving autonomously on an actual street, achieving speeds up to 96 km/h. This project was led by the German pioneer in driverless cars Ernst Dickmanns [27].

DARPA Grand Challenge

Since the year 2000, more research has been carried out in the field of AVs, with notably results such as Google's Driverless Car (2010), VisLab's BRAiVe (2012) and the Mercedes' S-class (2014). Before these cars existed, challenges such as the Defense Advanced Research Projects Agency (DARPA) Grand Challenge (2005), DARPA Urban Challenge (2007) and the Grand Cooperative Driving Challenge (2011) had to gradually raise the bar.

There are many advantages of having self-driving vehicles, and the appear on the commercial market by 2020 [102, 35]. Disney's cartoon 'Magical Highway' (1958) has already visualized how the future will look like. Comfort is an obvious advantage, but in the current society, the practical advantages of a AVs become clearer every day. Due to an increase of congestion on the road (especially in The Netherlands), the productivity decreases and money is wasted on fuel and time. Cooperative AVs enhance traffic flow. With regard to road safety, smart vehicles are likely to decrease the number of injuries and fatalities. A computer can be tremendously faster in many tasks than humans will ever be.

Current research such as [16, 46, 4, 22, 72] focuses on the autonomous technologies. Even if these autonomous technologies consider malicious input, they lack on security and cyber-attacks as depicted in Figure 1[1]. From a security-by-design perspective this is wrong, because a decision made by an AV is as good as the sensors can perceive. A faulty observation can lead to dangerous situations.

1 It could be argued that tamper resistance is covered by 'correctness'. Nevertheless, the author believes this is not the case.

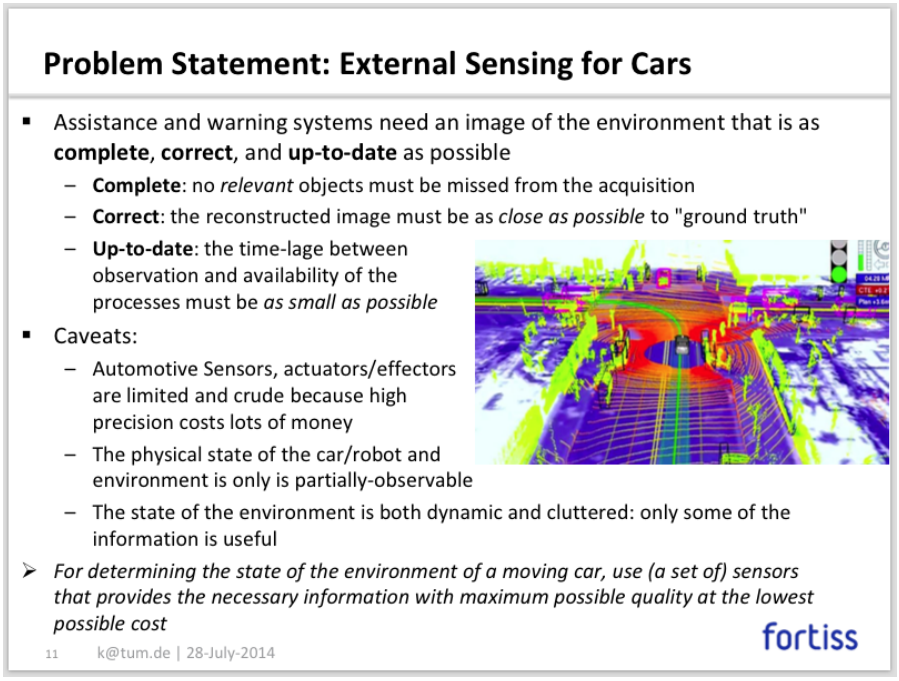


Fig. 1: In the problem statement on external sensing in this presentation from [72], tampering is not listed as a problem source.

Initial thoughts on cyber-attacks on autonomous cars were raised by a hacker with the name 'Zoz', during DEF CON 21 in 2013 [23]. The work of Petit [112] can be considered the first to elaborate on potential cyber-attacks on AVs in literature. In particular, these attacks have in common that they can be mounted externally (thus no physical access to the car), on existing sensors such as (stereo) camera vision, Global Navigation Satellite System (GNSS), Laser Image Detection and Ranging (Lidar) and Radio Detection and Ranging (Radar). However, both [23] and [112] are theoretical and have not conducted experiments on existing hardware. There is a need for practical research regarding this topic, as attacks on sensors can eventually cost lives.

1.2 RESEARCH QUESTIONS

Based on the problem statement, this study will address the following three research questions. The overall objective of this work is to find out if sensors can be influenced remotely, in such a way that the sensor either breaks or reports invalid information with the intention to crash or stop a vehicle. A survey on the sensors that are used in AVs will indicate which sensors are of interest to this work.

1.2.1 What types of attack can be mounted?

The types of attack that can be mounted is part of survey on autonomous vehicle technologies in Chapter 3.1. This chapter will point out which sensors are of interest to attack.

1.2.2 How likely are the attacks to happen and what are their consequences?

A decision made by an AV is as good as the sensors can perceive. A faulty observation can lead to dangerous situations that can eventually cost lives. Therefore, the consequences of the attacks depend on the application. For instance, if the lane-keeping application is attacked, it will have less consequences than when the Collision Avoidance System (CAS) is attacked. The latter is directly involved with preventing a crash when it happens.

1.2.3 What is the amount of effort that has to be put into the attacks, in terms of time and money?

For the attacks that are mountable, it is interesting to know if they are sophisticated or not. If they are, the attacker may require a lot of time and money to mount them.

1.3 CONTRIBUTIONS

Current literature on cyber-attacks is rather theoretical, such as [112] and [84]. Other works such as [24] and [103] limit their works to in-vehicle systems and communication busses. This thesis will contribute the following to literature.

AWARENESS OF THE ISSUE After an extensive literature study, the conclusion is that there are many applications available that add autonomy to an AV. Most of the applications use a camera system, such as lane-keeping and traffic sign recognition. Other applications include Lidar for range-finding and CAS. In most of the literature, malicious input and threat models are not considered. This work raises the issue, in particular for sensors that are commonly used in a AV at the time of writing.

DEMONSTRATION OF ATTACKS Several experiments that are concerned as proof-of-concept attacks on Lidar and camera hardware, without any prior knowledge of the systems. In addition, the influence of the attacks on the application-level is demonstrated.

THREAT MODEL An attacker model with attack scenarios that are likely to happen. This threat model is based on an attacker with limited money and limited time. It is debated that the attacks do not require expensive hardware.

1.4 ORGANIZATION

The structure of the rest of this thesis is as follows. In Chapter 2, definitions and backgrounds of AVs are established, together with a relevant attacker model and likely attack scenarios. Chapter 3 introduces sensors that are common for autonomous vehicles, including potential attacks. The experiments are conducted in 4. The sensors that are of interest will be discussed in here, including the experiments and results.

To conclude the thesis, Chapter 5 discusses limits of this work and possible countermeasures to overcome the attacks on the sensors. Finally, Chapter 6 will end this work with a conclusion and a proposal for future work.

2 DEFINITIONS AND ATTACKER MODEL

The complexity of vehicles is increasing rapidly. Not only from a technological point of view, but also from a societal point of view. In general, newer vehicles are equipped with more sensors and newer technologies [42] than their predecessors. Examples of these new technologies include Collision Avoidance System (CAS), lane keeping and parking assist. These technologies help to make vehicles safer, but also help the driver by offloading tasks. Depending on the tasks that can be offloaded to the vehicle, it can be called an Autonomous Vehicle (AV). Section 2.1 will explore the degrees of automation.

A definition of cyber-attacks will be given in Section 2.2, including a comparison with traditional cyber-attacks. An attacker model will then follow in Section 2.3, with a brief introduction of three frameworks for security modeling. An attacker model defines the capabilities of what an adversary can do and what it can not do. This is needed to reason properly about security requirements.

At the end of this chapter, in Section 2.4, the attacker model is extended with attack types and scenarios. This will be relevant for the rest of this work.

2.1 DEGREES OF AUTOMATION

What can be considered an AV, depends on the technologies (and limitations) that can offload a driver in controlling a vehicle. There are three major frameworks for classifying the autonomy of vehicles. These frameworks establish a global definition of what can be considered a AV and what can not, for instance for policy makers. The first is [15] by the German Bundesanstalt für Straßenwesen (BASt), second is [97] by the American National Highway Traffic Safety Administration (NHTSA) and last is [121]. All three frameworks are ordered, and rank autonomy of vehicles from no-automation (no tasks offloaded from a driver) to what can be considered a self-driving car (all tasks offloaded from a driver).

Five degrees of automation

In this work, the [97] classification is followed. The levels of automation are presented below.

LEVEL 0 – NO-AUTOMATION The actions performed by the car are the result of human actions, without any automation involved. This does not imply that the car does not have any electronics on board (e.g. Drive-by-wire or CAN bus).

LEVEL 1 – FUNCTION-SPECIFIC AUTOMATION This type of automation characterizes itself by the 'shared authority'. The driver enables one system, and shares control over the vehicle, but it continues monitoring the vehicle and the environment. It could be called 'hands-off, eyes-on' driving. In case of troubles, the driver can overrule the application immediately. Applications include Adaptive Cruise Control (ACC) and lane-keeping. A car can have multiple function-specific features, but in this case, the features work independently of each other.

LEVEL 2 – COMBINED FUNCTION AUTOMATION Same as above, but when one or more systems are combined as one specific application. The driver shares more authority with the individual systems. Compared to the function-specific automation, this allows the driver to be physically disengaged from the vehicle, by not touching the steering wheels or the pedals. However, the driver still can, and is expected to in case of danger, overrule the controls.

LEVEL 3 – LIMITED AUTOMATION Multiple systems and applications take over full control of the vehicle (including safety-critical functions), and the driver is expected to take over control when the automated systems are incapable of control, or limited by geographical boundaries. Current state of the art cars, such as the Google Driverless Car, are examples of this category.

LEVEL 4 – FULL AUTOMATION The vehicle is expected to have full control over all functions. It is not expected to have a driver available at all times during the trip. As of writing, no cars of this category are available, mostly due to legal reasons. This includes vehicles without a 'steering wheel'.

In [112], another distinction is made between 'autonomous automation' and 'cooperative automation'. While this work primarily discusses technologies classified as the first category, the definitions of both are presented below for completeness.

AUTONOMOUS AUTOMATION In this type of automation, information about the environment is fully gathered from on-board sensors, without any active communication between other vehicles or infrastructure.

COOPERATIVE AUTOMATION Vehicles communicate with each other and share information about the environment. Communication is not limited between cars (Vehicle-to-Vehicle (V2V)), nor between cars and infrastructure (Vehicle-to-Infrastructure (V2I)).

Throughout this work, the term AV will correspond to a 'Limited Automation' or 'Full Automation' vehicle. These two levels are (the future, and are) the most interesting ones when sensors can be remotely triggered to fail.

2.2 CYBER-ATTACKS

2.2.1 Definition

This thesis addresses cyber-attacks on AVs. Up to this point, no definition of 'cyber-attack' was presented. Multiple definitions of 'cyber-attack' exist in literature. These definitions typical address software and computer networks. For instance, [79] defines a cyber-attack as "deliberate actions to alter, disrupt, deceive, degrade, or destroy computer systems or networks or the information and/or programs resident in or transiting these systems or networks." Another definition by [57] defines a cyber-attack as "any action taken to undermine the functions of a computer network for a political or national security purpose." Although the definition states that a cyber-attack has a political or national security purpose, their interpretation does state that 'any action' can include "hacking, bombing, cutting, infecting, and so

forth", as long as the objectives attack the function of a computer network. These definitions do not fit the particular goal of this work very well: using a laser pointer to blind a camera sensor would be more closely related to vandalism than to a cyber-attack.

Safety requirements

It can be discussed that even the laser pointer attack is a cyber-attack when it is used to influence the decision making software. As an example: in [58], a case study identified several safety requirements for a prototype AV. The authors defined that "in cases where the GPS signal is lost or jammed, the vehicle is able to continue to plan its path by taking measurements from IMU in conjunction with other on-board sensors (such as Lidar)." This means, that if an attacker can block or jam the Global Positioning System (GPS) signal, it can also control the AV by attacking on-board sensors such as a Laser Image Detection and Ranging (Lidar). An attack can be one that causes the sensor to operate outside operating characteristics, thus violating safety requirements.

One way to extend the definition of a cyber-attack to cover the attacks in this work, is by including 'safety' in the definition. This is a reasonable modification, considering the attacker model. An attacker inevitably attacks the safety controls of an AV with the intention to influence the decision making software. This makes safety at least as important as security. For this work, the definition of attack from [128] is modified to include safety: "An assault on system security *or safety* that derives from an intelligent threat, i.e., an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services *or safety controls* and violate the security *or safety* policy of a system."

2.2.2 Types of attack

With a definition of cyber-attacks established, the following types of cyber-attack have been identified in the context of AVs. This listing is based on the ones presented in [128] and [117], and show how typical types of attack fit in the context of AVs.

DENIAL-OF-SERVICE ATTACKS In a denial-of-service attack, an attacker tries to prevent the delivery of a service to legitimate users. In practice, a certain service is flooded with many requests from fake users, in such a way that legitimate users cannot be served in an orderly fashion. This is not the only way to mount a denial-of-service attack. Other ways include crashing or compromising a service so it will be disabled. An example that is analogous to AVs would be that a pedestrian detection system would fail to track a pedestrian because an attack put too many mannequins aside of the road, in such a way it overloads the tracking algorithm.

REPLAY ATTACKS A replay attack is an attack in which a message is recorded and played back on another moment. If the message is badly protected (e.g. no timestamps, nonces or session tokens), it could result in the same action triggered twice, even when encrypted. Analogous to AVs, an example would be one where the Lidar signal is recorded and played back on a later moment to inject false objects, even if the underlying format of the signal is unknown to the attacker. To some extent, this attack is similar to a replay attack, where the message is not stored but transmitted directly.

INJECTION ATTACKS With injection attacks, an attacker potentially knows the format of a message. It then injects a message to trigger a certain response. An example for AVs would be that a traffic recognition system based on shapes would be triggered because an attacker put fake traffic sign shapes on the side of the road.

MODIFICATION ATTACKS A modification attack captures a message from a sender, alters it, and sends it to the original receiver. The same as with a replay attack or injection attack, the attacker does not need to understand the message format. Although the nature of such an attack may imply such an attack happens real-time, it may even happen at a later moment (e.g. message is stored). For an AV, an analog would be a situation where a traffic sign is wrongly identified because it is (slightly) modified.

2.3 ATTACKER MODEL

According to [124], "security is about Trade-offs, not Absolutes". A company can invest in the security of a product, either in software or hardware. However, if the risk for an attack is low and the cost is high, one may decide to not invest in countermeasures. Therefore, there is need for a framework to decide on the security requirements of a system. There are several frameworks to do proper security modeling of a system. Representable frameworks include attack trees [127], Failure Mode and Effect Analysis (FMEA) [125] and Common Vulnerability Scoring System (CVSS) [101].

Attack trees are a top-down approach for security modeling, introduced in the '90s. It is a tree graph with an ultimate goal as root node. An example of an attack tree is presented in Figure 2. To achieve this ultimate goal, a path of several subgoals, represented by child nodes, should be achieved. By default, the nodes in a tree are disjunct, but some nodes can be conjunct. This is useful if several subgoals should be fulfilled before the parent subgoal is completed. Nodes can also be augmented with variables such as cost and feasibility. According to [89], a major advantage of an attack tree is the decomposition of goals, so it is easy to see which countermeasure will have the biggest effect of advantage.

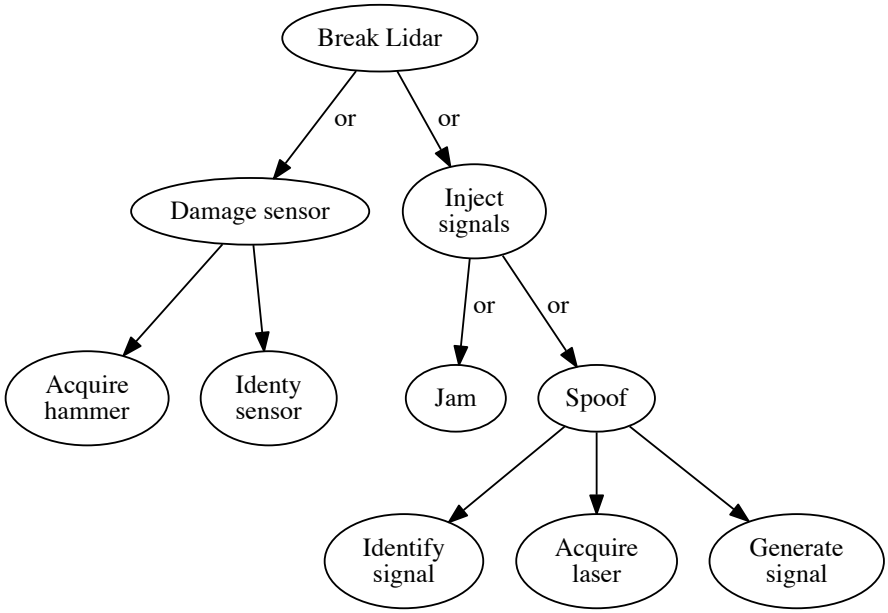


Fig. 2: Example of an attack tree. The ultimate goal is the root node. The child nodes have to be completed first.

FMEA is a much older framework, and dates back to the '50s. It was used by the United States Department of Defense to improve the reliability of military equipment. It focuses more on failure modes of actual hardware. As [125] shows, it can also be used for security requirements modeling. One of the biggest advantages of FMEA, is its age. It is well adopted in the field of engineering, for instance to guarantee product safety. As [132] mentions, it is important to conduct an FMEA carefully, so it can be used, for example, in court.

The last framework is CVSS. It focuses on three areas of interest ('Base', 'Temporal' and 'Environmental') to calculate a vulnerability score in the range 0.0 (minor) - 10.0 (critical)[1]. CVSS is used for security modeling in vehicles. For instance, in [103] the authors have focused on the topic of the increasing number of software components in a car, including connectivity with other cars, smartphones and more. They used CVSS to analyze the risk involved, and came up with a rough damage figure that clearly calls for action. Since CVSS is focused on software vulnerabilities, it is not a good candidate for security modeling of AVs, in which attacks are not limited to software only.

Attack trees, FMEA for security modeling and CVSS have in common that they involve actors that want to misbehave (anti-goals), according to [32]. This is the opposite of risk analysis, where failure of a product is also an important cause. It is therefore necessary to have a persona of these actors, an attacker model. With an attacker model, one can reason if a problem is critical or not. The attacker model should be realistic [106]. If it is modeled too powerful, it is most likely that all security requirements are impossible to fulfill. If it is modeled with not enough capabilities, it will be unrealistic.

A first category of attacker models are the formal methods. These models assume certain formal properties, that can be checked with model checker tools or proven with mathematics. An example of such an attacker model is the Dolev-Yao threat model [28]. This model is used to prove the security of cryptographic protocols. In this model, the attacker can replay, intercept

1 To give an idea of a major bug, the 2014 discovered 'Heartbleed' vulnerability was classified as major [146]. Although it was big news, it only received a vulnerability score of 5.0 due to low exploitability.

and inject messages, using the cryptographic methods exposed by the protocol. The other category of attacker models are the more practical models. They do not have formal properties and cannot be proven, but resemble a persona (frequently used in Human-machine Interaction (HMI)) with specific attacker capabilities and properties. In [112], the following properties are presented, which are adapted for this work:

INTERNAL VERSUS EXTERNAL The internal attacker has physical access to the vehicle. For example, it has direct access to the internal Controller Area Network (CAN) bus. The external attacker does not have access to the car, so only remote attacks can be mounted from a distance.

MALICIOUS VERSUS RATIONAL A malicious attacker seeks no personal benefits from the attacks, and aims to harm the vehicle and/or drivers. The rational attacker seeks personal profit, and hence, is more predictable in terms of attack means and attack target.

ACTIVE VERSUS PASSIVE The passive attacker can listen to communications only. An active attacker can do the same, but it can also inject and spoof false signals or block signals.

LOCAL VERSUS EXTENDED A local attacker has limited locations to mount an attack. An extended can mount (or extend) an attack over multiple locations. For example, a local attack would be blinding the camera at one spot, but spoofing the navigation system of a moving car for a longer distance requires the attacker to follow the car.

INTENTIONAL VERSUS UNINTENTIONAL An intentional attacker mounts an attacks on purpose, while the unintentional attacker generates signals that have unintended side-effects. The unintentional attacker may not even know he is attacking.

Additionally, the following three general properties are added to the attacker model:

AMOUNT OF TIME An attacker has either limited or unlimited time. With limited time, it is assumed one or multiple steps in an attack are time bounded, or lose interest after a certain amount of time time (e.g. brute forcing keys or product evolution).

DETECTABLE VERSUS UNDETECTABLE A detectable attack(er) leaves clear traces, such as damage due to installation. An undetectable attack is hard to detect. Any goals reached are hard to trace back to an attack, and may look like it was caused by other things.

AMOUNT OF MONEY The amount of money an attacker is willing to spend, or can spend, to reach it is target goal.

Attacker model

The attacker model will be used to evaluate the attacks in Section 6.2. The best description for an attacker that fits the purpose of this work, is a *limited time* and *limited money* attacker with the *intention* of *actively* disrupting components *undetectably* and *externally*. The attacker is not limited to any regulations that may apply, such as a transmitting license. This follows the classification suggested by [112]. Time is chosen to be limited, because it is assumed that new technologies will follow fast, thus old technologies will eventually be superseded by better ones. Furthermore, it is assumed that most of the time will be invested in preparing the attack. As an indication

for the amount of time, this thesis will 'prepare' several attack on existing sensors in a time span of six months, without any prior knowledge. The reason for choosing limited money, is because the goal of this work (and further research) is to show that with inexpensive hardware (from sources like eBay), attacks on existing sensors can be mounted. With unlimited money, there are easier means of disabling hardware, or even destroy it. For instance, take an Electromagnetic Pulse (EMP) cannon and integrate it in the road. Each car that moves over will be disabled. Not to mention, as pointed out in Section 3.1.4, hardware can get less expensive over time.

2.4 ATTACK SCENARIOS

There are many scenarios of how an attacker can mount an attack, depending on the sensors used. For this work, the following three scenarios have been designed to discuss the likelihood of an attack.. Although more scenarios are possible, the scenarios below have in common that attacks can be mounted while the target car is driving at high speed, as opposed to low-speed activities such as parking. It is the goal of the attacker to either cause as much damage as possible, such as crashing a car, or to force a car into minimal risk condition, i.e. stopping it safely. If it can be put in minimal risk condition, this implies that an AV can detect faulty sensors or tampering.

FRONT/REAR/SIDE ATTACK In a front/rear/side attack, the attacker mounts the required hardware to mount an attack in another car. Depending on the hardware, this can be installed without anyone else noticing. The car is then used to drive in front of (or behind of, or next to) the target car. When positioned, the attack is executed once or multiple times. The advantage of this attack scenario is that it allows an attacker to keep the same distance to the target AV for a longer period.

ROADSIDE ATTACK A roadside attack is mounted stationary. In this scenario, the attacker can mount the required hardware in objects on the side of the road, such as the guard rail. The attack is not limited to one installation point, but can be spread over multiple installation points, potentially connected to each other (e.g. for replay attacks).

SCENERY ATTACK In a scenery attack, the scene is changed by the attacker in such a way the target AV is unable to perceive the original scene, or perceives too much. For instance, extra traffic signs are placed, or existing ones are modified to present the wrong information.

EVIL MAID/EVIL MECHANIC ATTACK Several attack surfaces evaluated in [24] and [65] include full physical access to the vehicle. In [111], the term 'Evil Mechanic' was introduced as an extension of the 'Evil Maid' by [120]. Such an attacker has short-term physical access to the car, e.g. when it is parked or left for maintenance. A similar scenario is applicable for this work: if the sensor can be influenced remotely from the roadside, it can also be mounted on the vehicle. For instance, an attacker can mount mount a jamming device on a (carrier) vehicle that jams other cars without noticing.

All of the attack scenarios involve general-purpose locations. The attacker does not need special access to a certain area, or similar.

High-speed/Low-speed

A distinction is made between low-speed and high-speed situations. A low-speed situation is considered to be less than 50 km/h or 13.8 m/s and takes into account incoming traffic, pedestrians and more (e.g. city traffic). A high-speed situation is one on the highway with a speed of approximately 130 km/h or 36.1 m/s. It does not account for incoming traffic and pedestrians. The reason for this distinction is that an attack in one situation does not have to be effective in the other. For instance, in a city a vehicle has to take care of not driving into pedestrians, whereas on a highway the vehicle should make sure it does not crash.

In situations where an immediate action is required, it is assumed that a AV needs (far) less time to decide on that action than the response than a human. On average, the response time of a human is one second. In addition, it is assumed that a AV does not have a better braking system compared to a traditional car. An AV cannot take more time to analyze a dangerous situation before it responds to it, because AVs tend to be superior in making a decision compared to humans. This is a trade-off in terms of safety and robustness. Taking more time adds more distance to the braking distance, but will produce less false positives. Taking less time shortens the braking distance, at the cost of more false positives. In case of high-speed scenarios, it is presumed that an AV will brake as soon as it decides it has to: every tenth of a second adds approximately 3.33 meters to the total braking distance.

3 AUTONOMOUS VEHICLE SENSORS

3.1 SENSOR TECHNOLOGIES

A typical modern car is equipped with many sensors. In [42], fourteen types of sensors are listed, which can be applied to ten different application fields. Most of the sensors are only accessible to the internals of the vehicle. These applications make sure the vehicle keeps running. Only a few of the application types are involved with perception of the world. Perception is the process of converting the physical environment into digital signals for further processing, such as measuring forces or measuring distance. Table 1 lists the fourteen sensor types.

Sensor perception

Tab. 1: Classification of vehicle sensors, according to [42].

Sensor Type	Technologies	Applications
Rotational Motion	Hall Effect, Magnetoresistor, Wiegand Effect	Engine Diagnostics
Pressure	Piezoresistive, Capacitive	Vehicle, Engine Diagnostics
Angular and Linear Position	Potentiometer, Hall Effect, Camera, Magnetostrictive Pulse Transit Time	Transmission, Breaking, Steering
Temperature	Sillicon, Thermistor, Resistive Temperature Detector	Safety, Comport and Convenience
Mass Air Flow		Engine Control
Gas Exhaust		Engine Diagnostics
Engine Knock		Engine Control
Linear Acceleration	Piezoresistive, Capacitive, Resonant-beam, GPS	Navigation, Security
Angular Rate		Navigation
Solar, Twilight and Glare		Comfort and Convenience
Moisture/Rain		Comfort and Convenience
Fuel/Fluid Level		Breaking

Tab. 1: Classification of vehicle sensors, according to [42] (continued).

Sensor Type	Technologies	Applications
Near-distance Obstacle Detection	Ultrasound, Micro-wave Radar, RF capacitance	Safety, Comfort and Convenience
Far-distance Obstacle Detection	Millimeter-wave Radar, Lidar, Thermal Imaging, Camera	Safety

Most of the sensors are connected to an internal communication network, such as the Controller Area Network (CAN) bus [39] or the Drive-by-wire bus [47, 43]. This makes these types of sensors interesting attack targets. Despite that, such attack vectors[1] generally require physical access to the car, which is out of the scope for the attacker model introduces in Section 2.3. The work of [24] discusses external attacks, but their work is limited to gaining entrance via exploitable input and output channels, such as Bluetooth, keyless entry systems and wireless maintenance ports.

This chapter will introduce the most important sensors used in a typical Autonomous Vehicle (AV) that is described in research and in popular publications. Their limits and attack vectors will also be discussed.

To prevent any confusion about the terms 'sensors' and 'application', in the rest of this work, when referring to sensors, sensors that perceive the environment are meant. Applications refer to the practical uses of the sensors. For example, Laser Image Detection and Ranging (Lidar) is a sensor, while collision avoidance is an application.

3.1.1 Lidar

Lidar is a type of range-finding sensor. Briefly, it works by emitting a light pulse and measure the time it takes to reflect off a distant surface, called a ping. The time is a measure for the distance. Most speed measurement devices, such as the ones used by the police, are based on this principle.

For completeness: Radio Detection and Ranging (Radar) and Sound Navigation and Ranging (Sonar) are two other but similar methods of range-finding. Radar uses microwave radio pulses while Sonar uses (ultra) sound for pulses. The advantages of Lidar over Radar include the higher spatial resolution (10 cm versus 1 meter according to [87]), making it possible to have better resolution images when used for scanning. Pedestrians can be separated from cars at this resolution. Sonar is not a feasible technique. For sound waves in air, the speed of sound is approximately 880,000 times slower than the speed of light (at room temperature and atmospheric pressure). However, the energy of Radar waves and Lidar pulses are quickly absorbed by the water molecules, making them unusable for underwater operations.

Spatial resolution

In The Netherlands, Lidar has the advantage of not requiring a transmitting license for longer distances, as opposed to Radar. Short-distance Radar for collision detection (up to approximately 40 meters) is permitted in vehicles without a license [1].

1 An attack vector is a point to attack, for instance the CAN bus protocol.

Measuring distance

To measure the distance, Equation 1 is used. c is the speed of light in a vacuum ($\approx 3 \cdot 10^8 m/s$), n is the refraction index of the transferring medium and t the time of flight. Without factor half, the output is the total distance travelled back and forth.

$$d = \frac{1}{2} \cdot \frac{c \cdot t}{n} \tag{1}$$

When Lidar is mounted on a rotateable head, it can be used to generate a two-dimensional or three-dimensional image of the world, by quickly rotating the head. Figure 3 shows how this works. The resolution depends on the number of steps per revolution. A typical system can do over 20,000 individual measurements per second [87].

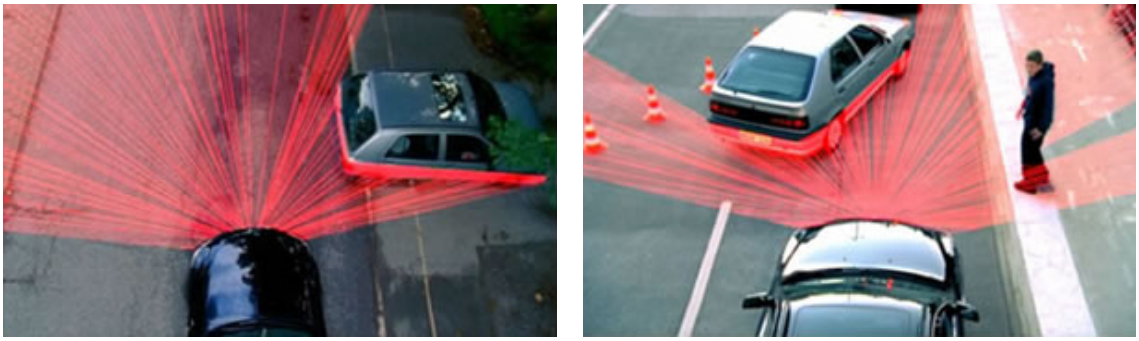


Fig. 3: How Lidar perceives the world. Any object in line of sight will reflect back to the Lidar. Note that, in practice, Lidar uses invisible light.

There are two ways of obtaining the speed of a remote object. The first one uses range differentiation, where two distance measurements within a known interval reveal the speed. The other approach uses the Doppler effect. By using the Doppler effect, the shift of frequency due to movement between sender and receiver, the speed of a remote object can be measured with the Equation 2. T1 and T2 are the period of the reflected light. c is the speed of light and n is the refraction index of the transferring medium.

$$v = (\frac{T_1}{T_2} - 1) \cdot \frac{c}{n} \tag{2}$$

Applications

As mentioned above, Lidar is used for different applications. The most common ones are Adaptive Cruise Control (ACC), Collision Avoidance System (CAS) and object recognition (in general).

ACCs systems are used in cars for many years. A typical ACC controls the gas throttle, to slow down if a vehicle in front comes closer, or speed up (to a desired speed) when there is room. The driver can still override the acceleration if they like. According to [155], 90% of the traffic accidents are the result of human error. ACC systems can help reduce this number, by enlighten long and repetitive driving tasks. In the work of [151], no significant difference was found when comparing Lidar and Radar based ACC systems.

The technology behind CAS is almost identical to the technology behind ACC. The major difference is that a collision can occur at any time, and the vehicle has limited braking power. If it is assumed that the reaction speed of a human can be ignored (which *could* be true for a AV), a typical vehicle driving 130 km/h would at least need 70 meters to stop. Therefore,

a short-range radar system would be insufficient, because it perceives too late. Volvo is an example of a vehicle manufacturer that have implemented CAS [55]. It uses Lidar to track objects, fused together with camera imaging to identify objects. In case of an approaching collision, it will automatically hit the brakes.

With regard to current research, object recognition is another application of interest. When a Lidar sensor is mounted on a rotatable mirror, it can be used to provide vision in two or three dimensional view (see Figure 4). In most cases, shorter range is preferred, but with higher angular resolution[2]. For example, the commercially available Ibeo Lux HD [10] has an angular resolution of 0.125 °. The device can classify cars and pedestrians. One way to classify certain objects, is by using a depth map. For instance, a pedestrian will appear as a small object on the depth map, while a car will appear as a much bigger object. Combined with speed information and tracking algorithms (such as a Particle Filter (PF), discussed in Section 3.2.2), objects can be classified and tracked. Other object recognition applications include terrain classification [75] and lane detection [135, 50].

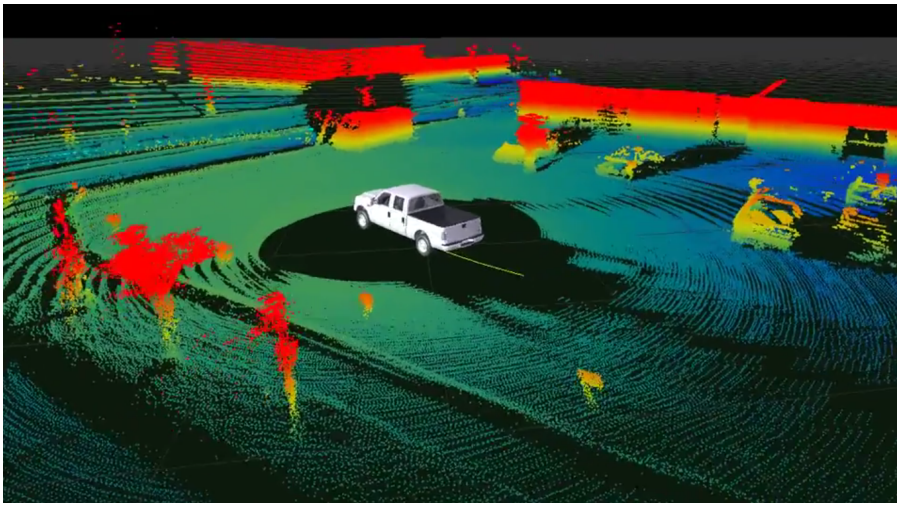


Fig. 4: Three-dimensional view of a 360 ° Lidar. The color represents the height. Image taken from [107].

Attacks

Unfortunately, there is no literature that describes an attack on Lidar directly. Since Lidar is the preferred technique in speed measurement devices, jammers are widely available on the (black) market. However, a Lidar can only see things that are reflected by the signal. If the signal does not return (due to absorption, transparent objects or range limits), it will assume there is 'nothing'. For a 360 ° view, most of the world will be classified as 'nothing'.

Reflective objects can confuse a laser beam. Objects that are far away could be brought nearby, which is major problem for CASs. Also, some objects on the road are reflective by design. Lane markings reflect some of the signal, so it will be visible in the perceived image.

Wavelength of laser

Lidar uses light of a specific wavelength, and different wavelengths yield different results. In [122], different wavelengths were chosen and examined, regarding reflective properties on car parts and absorption. Their work discussed that the atmospheric absorption is the primary factor for limiting the allowable wavelengths for Lidar applications. Most of the light is attenuated by water molecules in the air, depending on the wavelength. Lasers typically

2 Smallest angle between two objects at the same range that allows an observer to still distinguish them.

use a wavelength of 800 nm - 2800 nm (near-infrared band). But lasers with a wavelength in the range of 700 - 1400 nm are not eye-safe[3]. This limits the maximum transmission power. On the other hand, it was concluded that lasers with a wavelength of 8100 nm (mid-infrared band) work fine, and even allowed ten times more power than the 1560 nm wavelength laser. Unfortunately, the costs and size of the optics and lasers currently outweigh the performance. From [6], it is known that, for geographical mapping from planes, lasers of 1064 nm are used (near-infrared). In cases where water surfaces are mapped, lasers of 532 nm (green) are used, to minimize absorption by the water.

Absorption of light due to rain or snow can reduce the remission rate drastically. For example, the ibeo LUX 3 has a range up to 200 meters, but when only 10% of the light reflects due to non-optimal weather conditions such as rain or snow, its range drops to 50 meters [10]. Non-optimal weather conditions are currently a major limitation for the Google Driverless Car [48].

Signal noise

In an interview with an expert from DARE!![4], it was told that new technologies in cars cause problems with existing road infrastructure systems. For instance, CAS and ACC applications using Lidar or Radar are causing interference with older infrastructure systems. These systems were never built to work with so many 'noisy' signals on the same frequencies. This will become a bigger problem when every car will, eventually, be equipped with Lidar. At DARE!!, they develop speed gun detectors and jammers, based on Lidar. For this to work, their systems need to know which type of speed gun is sending the signal, so they can send a pulse back *before* the next one will arrive. Effectively, this means the speed gun will read a slower speed. It is worth mentioning that 'just jamming' will work too, but the speed gun will read that it was jammed (In The Netherlands, that is forbidden).

3.1.2 GNSS

This section will describe the currently available Global Navigation Satellite System (GNSS). The main task of GNSSs is to provide localization and time synchronization services.

Systems

There are multiple GNSS systems available. The most famous one is Global Positioning System (GPS) (originally and officially called Navigation Satellite Time And Ranging (NAVSTAR)), developed in 1973. Initially, GPS was only available for the United States Department of Defense. Since 1983 it has been accessible for civilian use, but it took until 1994 before the system was ready actually ready for civilian use. The GPS network consists of 24 satellites, of which three are backup [74]. There are five operating frequencies, of which two are relevant, the L1 and L2 code. All satellites operate on the same frequencies, and use Code Division Multiple Access (CDMA) to simultaneously access the bandwidth. The used codes are called the Pseudorandom Noise (PRN) codes of 1023 bits, which uniquely identify the broadcasting satellite. The GPS data is transmitted via the Coarse/Acquisition (C/A) code. This is the unencrypted navigation data. The encrypted (military) signal is called the Precision-code, also broadcasted by every satellite.

Pseudorandom Noise codes

3 Near-infrared light does not trigger the blink reflex of an eye. That is why even low-power near-infrared lasers are dangerous.

4 DARE!! is a company specialized in Electromagnetic compatibility (EMC) compliance testing. See `http://www.dare.nl` for more information.

It has it is own PRN codes, but it is in the order of 10^{12} bits long. When locked onto the signal, the receiver will receive the Y code, which is the encrypted signal with a unspecified W code. Only authorized users can decipher this. In later GPS satellites, extra features are added. This included the use of the L2 signal, a pilot signal for easier lock-on and forward error correction.

Global Navigation Satellite System (GLONASS) it the the Russian alternative. Its development began in 1976, and had full coverage in 1995. But due economic crisis during in the '90s, the system's coverage degenerated, and it was not until 2011 full coverage was reached again [98]. There are 28 satellites in orbit [40], of which 24 are required for full constellation.

The European project Galileo and the Chinese project BeiDou are still under active development. Galileo will be the first civil GNSS system, as opposed what GPS and GLONASS are. It is a project executed by the European Space Agency (ESA), and permission was granted in 2003. Currently, there are 4 satellites launched and operational, of the total 30 by 2019. Three satellites are used as a backup system. China is working on their own GNSS that is called BeiDou, also known as COMPASS. The idea was conceived in the 1980s. By 2012, regional coverage was completed. Global coverage is expected to be in service by 2020. In total, 35 satellites will be launched.

There are several methods of augmenting GNSS data, to get a better estimate of the location. Three of these methods are Satellite-based Augmentation Systems (SBASs), Assisted-GPS and Differential-GPS. SBASs is the first method. Such systems are commonly used in airplanes, for critical phases such as the landing phase. They consist of a few satellites and many ground stations. A SBAS only covers a certain GNSS for a specific area. The standardized systems are:

- North America — Wide Area Augmentation System (WAAS) to complement GPS.
- Europe — European Geostationary Navigation Overlay Service (EGNOS) to complement GPS, GLONASS and Galileo.
- Russia — Wide-area System of Differential Corrections and Monitoring (SDCM) to complement GLONASS.
- Japan — Multi-functional Satellite Augmentation System (MSAS) to complement GPS.

Assisted-GPS is widely deployed on mobile phones. When a receiver is searching for satellites, an almanac is consulted. The almanac, downloadable from the internet, tells the receiver which satellites are likely to be visible with respect to time and geographical area (e.g. per cell tower). With this information it takes less time to scan the ether for available satellites, so a position can be obtained faster. Differential-GPS works different, and requires two receivers. One receiver is fixed at a known position. The other is the actual receiver. It is assumed that the GPS signal that hits both receivers is attenuated the same way, resulting in position errors. Because the reference receiver knows its exact position, it can work the triangulation equations backwards, therefore calculating the error. This error is then transmitted to the actual receiver, which in turn, can correct the error.

Accuracy

For every GNSS, the accuracy is greatly dependent on and influenced by external factors [54, 76], presented below. These factors are not only applicable to GNSS applications, but every other wireless transmission application.

PROPAGATION ERRORS AND SPACE WEATHER The satellites orbit around earth at a height of approximate 20.000 km. At this height, signals can be affected in many ways. When signals hit the earth, they have to pass through the ionosphere (upper part of atmosphere). This layer is always hit by sunlight, and therefore ionized. These ionized particles tend to slow down radio signals coming through. This slowing down causes the satellite to look further away for the receiver. After the ionosphere, there is the troposphere. Here, the reflective index changes, which has a small impact on the signals.

According to [76], 'space weather', greatly influenced by the sun, affects signals too. Almost every day, the sun emits solar flares into space. High-intense ones happen a few times per year (X-class solar flares). During a flare, radio waves, X-rays and gamma-rays are swept into space. These rays have little to no effect on the earth itself, but it induces extra current in the satellites and ionizes particles in the atmosphere. The extra induced current can damage satellites, while the ionized particles can attenuate signals.

Multi-path Effects

MULTI-PATH EFFECTS GNSS requires exact timing in the order of nanoseconds to determine position. If satellite signal reaches earth, it can reflect on buildings and other objects, causing an increase in travel time. This influences the measurements. For stationary measurements, this looks like if the measurement jumps between multiple points. Using good quality antennas can reduce multi-path effects. Alternatively, avoid using satellites that have low elevation.

SATELLITE POSITION GEOMETRY With triangulation, a better fix is yielded in case two satellites have a greater angle between them. It is called *Dilution of Precision* when this is not the case. Figure 5 illustrates this.

RECEIVER CLOCK ERRORS Again, a measurement is time-dependent. Clocks that are off by a few parts can affect a measurement, since it might advertise a satellite to be closer or farther away.

SATELLITE ORBIT ERRORS Even though a satellite 'floats' 20.000 km above the earth, it is a real challenge to keep it up there. Wrong heights affects the time of flight of the signal.

VISIBLE SATELLITES At least three satellites are required to yield a latitude and longitude. A fourth one adds altitude. Having more visible satellites allows the receiver to select the best visible ones, or to combine measurements.

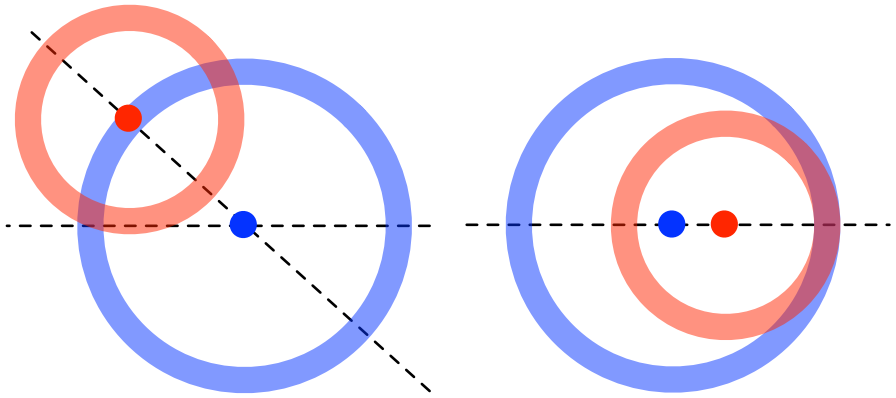


Fig. 5: Dilution of precision explained. The dots are satellites, the area is the estimated position. Image simplified from [152].

(a) Great angle and a small area of overlap: more accurate.

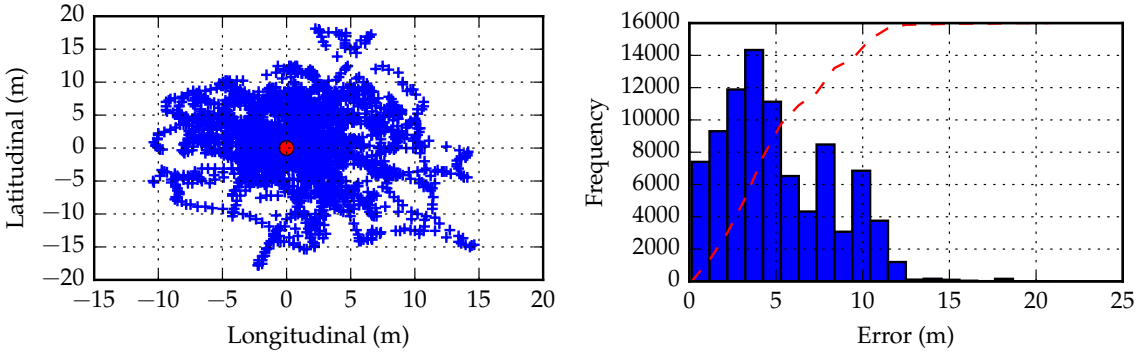
(b) Small angle and a big area of overlap: less accurate.

Selective Availability

The performance of GPS in terms of accuracy and signal acquiring increased during its development. Both the military and civilian GPS signal have the same accuracy, but the military signal has additional capabilities that allow for ionospheric correction. This reduces radio degradation caused by the atmosphere of the Earth [145]. But before May 2000, GPS satellites had 'Selective Availability' turned on. With this technique, the U.S. Department of Defense intentionally decreased the accuracy. Without this technique, the worst case accuracy is 7.8 meters at 95% confidence level [143, 144]. With this technique enabled, it is accuracy will be 100 meters. It is believed that the next generation satellites (GPS-III) will not be equipped with Selective Availability anymore [144]. GLONASS satellites, which have already been launched, do not have Selective Availability on board [41].

A simple experiment was conducted to measure the accuracy of GPS. Using a Navilock NL-402U GPS receiver, over 88,000 recordings were collected in a period of 24 hours, at a frequency of 4 Hz. The sensor was positioned stationary, indoors on the second floor and directly in the front of a window with clear line of sight to the sky. The weather was partially cloudy, during the day, without rain. The results are plotted in Figure 6. Conversion from longitude and latitude degrees to a distance relative to the center point (determined with Google Earth), is calculated via the Haversine function [37].

Fig. 6: x-y plot and histogram of 88,828 GPS position sampled over a period of 24 hours, while stationary.



(a) x-y plot of the latitude and longitude position recordings.

(b) Histogram of GPS errors. 95% of the recordings have 11 m or less error

GLONASS has similar accuracy compared to GPS, but since GLONASS orbits at a lower height, it has improved accuracy at higher latitude (towards

north and south pole), according to [139]. Unfortunately, there is no data available on BeiDou and Galileo.

To improve reliability, it is a good option to combine results of two or multiple sources. This improves accuracy, availability, but also integrity. Newer GNSS receivers are designed to work with multiple systems at the same time. In [99], it is shown that a combination of GPS and GLONASS have more accuracy over GLONASS-only. The results, measured at a few different Russian stations, are presented in Table 2.

Tab. 2: A comparison of the combined accuracy of GLONASS only and GPS. The stations are located in Russia. Table modified from [99]. Lower is better.

Station	Error of navigation (p=0.95)					
	Latitude (m)		Longitude (m)		Altitude (m)	
	Single	*Combi*	*Single*	*Combi*	*Single*	*Combi*
Bellinsgauzen	4.80	2.69	5.23	2.29	11.44	6.26
Gelendzhik	5.60	2.83	6.28	2.60	14.08	6.86
Irkutsk	6.35	3.08	6.39	2.86	10.52	5.98
Kamchatka	5.73	3.03	5.25	2.40	12.72	6.07

Navigation

For navigation applications such as turn-by-turn navigation, the accuracy of GPS is sufficient. By fusing position data with acceleration data from an Inertial Measurement Unit (IMU), the accuracy is within reasonable margins for navigation. Unfortunately, for AVs the accuracy is not high enough. Besides position information, a vehicle needs to know where it drives on the road, so called lane-level navigation (sub-meter accuracy).

Lane-level navigation

According to [17], one way of achieving lane-level accuracy, is by using Enhanced Maps (Emaps). An Emap is a standard map, augmented with more information, such as road characteristics, traffic signs, lane definitions, road markings, speed limits, curves and more [137]. The Google Driverless Cars fuses Lidar and camera vision with Emaps for road scenery understanding.

According to [29], Emaps and regular maps can be classified as one of the following three classes. They represent (but are not limited to) the amount of detail that is represented in each map.

MACRO-SCALE Most regular maps are considered to be on macro level. At this level, the roadway network consists of links (roads) and nodes (e.g. intersections), mostly represented as series of polylines (including shapes). Optionally, attributes can be associated with links and nodes, such as road type, speed limit and the number of lanes. A typical navigation system will try to find the shortest path between point A and B. The order of magnitude for navigation accuracy is about 10 meters. Note that due to this error, the accuracy of nodes, links and shape do not represent the ground truth [136].

MESO-SCALE At meso scale, the vehicle operation is considered to be on link-level. More features can be associated, such as multiple lanes (in contrast to only the number of lanes), on/off ramps, etc. Navigation at this level takes the lanes into account, so the order of magnitude for the navigation accuracy will be around 3 meters.

MICRO-SCALE Typically, this scale is used for specific tasks and does not take navigation into account. It is not limited to GNSS applications, but every system that can build up the environment (such as vision-based systems). Examples include lane keeping, traffic sign recognition and more. Sub-meter navigation accuracy is possible, with the right sensor systems.

The research on Emaps is sparse. While the work of [29] is dated, the reasons are still valid: the accuracy of GPS is somewhere between meso and macro, and since macro scale navigation has sufficient features, no effort is put into meso or micro navigation for commercial purposes (yet). However, as [136] mentions: this will slow down applications such as lane-level navigation.

Detailed maps can reduce the position error. By knowing where road segments are, a GNSS position reading can be corrected. In [14], a system is proposed where an Emap enhanced the GPS position, with support of vision. Figure 7 gives an overview of the algorithm.

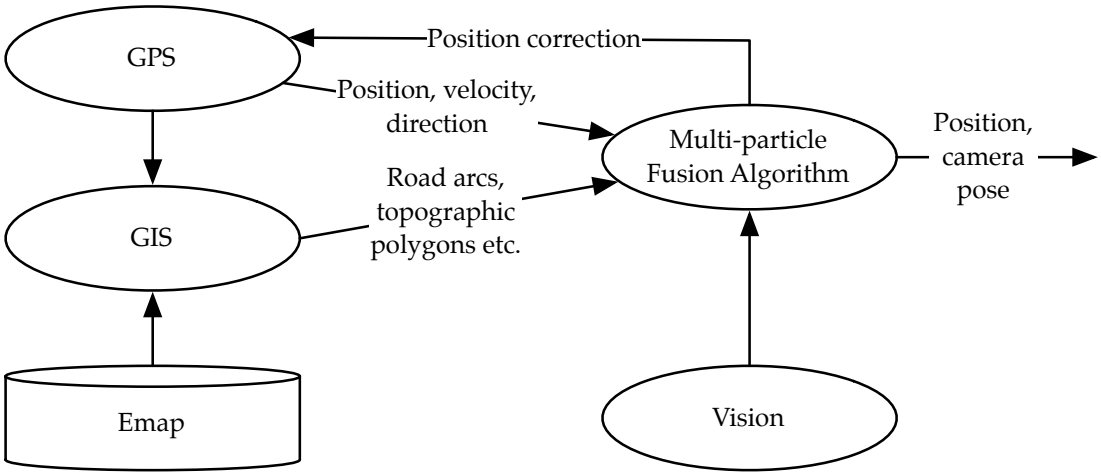


Fig. 7: Sensor fusion with Emap. Image adapted from [14].

For a given position, the Geographical Information Systems (GISs) is queried for the most probable road segments. The result is a set of connected road arcs, which model the road segments stored in the Emap. Arcs represent continuous lines, since this is what connects road segments. An algorithm attempts to find the longest biarc to fit all the road segment data points within a predefined error tolerance bound. The set of road arcs are then used to initialize the multiple particle filter, which is used to track the real road segments via the camera. The result of the tracking algorithm is then used as a feedback for the GPS measurements.

The researches tested the system in the United Kingdom, and found that the GPS error could be reduced to one meter, as long as the road is stored in the Emap. The tracking and overlaying system works well for flat environment (Figure 8), thus the presence of vertical curvature-forming road bumps and slopes increases the error. Furthermore, roundabouts and road junctions break the system, making it, at the time of writing, unusable for AVs.

Fig. 8: Visual overlay of algorithm result over camera images. Images taken from [14]

As with regular maps, Emaps should be up-to-date. On the meso-scale level, more information is available, thus providing more information that *could* change over time. The road map and its attributes change frequently, and not all changes are the responsibility of one party. Therefore, these changes should be incorporated in a map very quickly.

There are two important questions that arise from the problems. First, if Emaps are used for navigation purposes, what should an AV do when it encounters a (new) situation where Emaps lack information[5], or provides wrong information. Solutions could include downloading latest change sets on-the-fly, or Vehicle-to-X (V2x) enabled infrastructures which provides alternatives. Second, when the AV uses an Emap to validate it is micro scale observations, downloading changes on the fly may not be sufficient. What if the map suggest to take a certain off ramp, while the environment does not find it. Should the car take the ramp? Or if the map tells the AV that a speed limit applies, while traffic sign recognition says otherwise?

The work of [136] proposes a system to monitor the integrity of lane-level positioning by using Emaps. As opposed to the set of arcs used by [14], the road segments are modeled by clothoids. A clothoid has a generic shape, and an algorithm finds the best parameters to model a road segment. The algorithm outputs two parameters that indicate how much the current position can be trusted, based on a particle filtering system, combining GNSS readings, odometer information and IMU values. The authors acknowledge that, to achieve full integrity in navigation, efficient means for removing GNSS outliers and mitigating multi-path effects are highly recommended.

Attacks

Besides the accuracy problems mentioned in the previous sections, there are a few attacks possible on one or more GNSSs. Typically, an attack can *jam*

5 The Dutch Ministry of Transport introduced 14 new traffic signs in September 2014 [118], that are applicable as of January 2015. This would require all Emaps to be updated in a time span of four months.

the signal, or it can *spoof* the signal. In literature, most attacks address GPS, since it is the oldest system, with the biggest impact.

Transmission power of GPS

A GPS signal is transmitted with approximately 27 watt (comparable to a light bulb) of power. When the signal hits the earth, it signal strength will be roughly $10^{-16}W$ or $-160\text{dB}W$ [26]. This is approximately six times weaker than the background noise on earth, hence, PRN codes are used. The receiver correlates the received signal with the PRN code, and it will correlate very well if it matches. This process can be called a 'mathematical signal amplifier', and that is a reason why satellites do not require high-power transmitters [138]. If an attacker is in the possession of a GPS spoofer or jammer, only a few milliwatts is enough to override the satellite signal at reasonable distances. There have been incidents with devices that unintentionally jammed the GPS signal, due to the lack of electronic shielding, so called Continuous Wave Inference (CWI) (see Figure 9). Since the use of GPS' is not only limited to navigation, but for time synchronization too, the loss of service due to a (unintentional) jammer could have drastically impact (for instance, [141] mentions synchronization of the power grid or high-frequency trading).

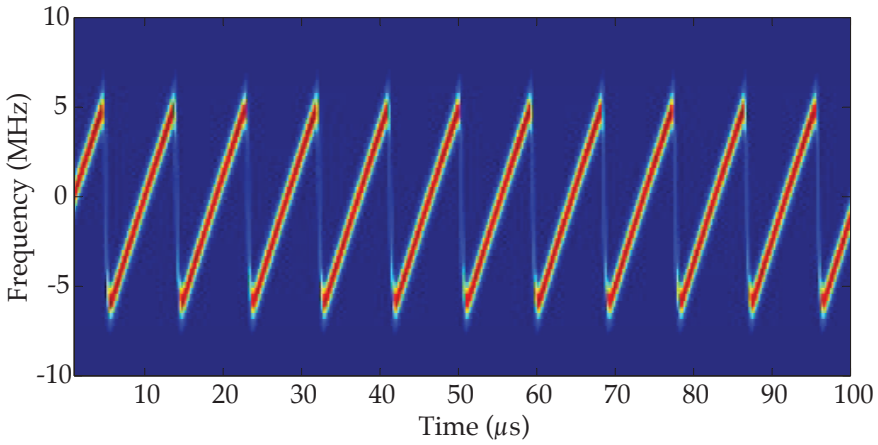


Fig. 9: CWI inference from a car GPS jammer. The color indicate the transmit power. Graph modified from [20].

Because the GPS signal is so weak[6], a GPS receiver 'scans' the frequency band for potential satellites. It does this by using the PRN codes, to acquire the C/A signal. Jamming the signal is relatively easy, if one emits counterfeit signals on the right frequency. The receiver cannot lock on the right signal. Even locked receivers can be forced to lock on the counterfeit signal, usually by gradually increasing the transmitting power, so the receiver is 'lifted' from the locked signal [114]. Spoofing involves more work. It requires more sophisticated hardware, because it has to trick the receiver to follow the spoofed signal and provide the fake navigation data (remember that triangulation with a GNSS requires at least three satellites, so an attack has to spoof at least three satellite signals). Given the data from a spoofed and real GPS receiver, there is no way to differentiate between the real and spoofed signals.

Even the encrypted military signal can be attacked using jamming or spoofing. If an attacker spoofs or emit counterfeit signals on the military signal, a receiver may switch back to the unencrypted signal. Then, if the attacker also controls this signal, it can still attack an encrypted receiver.

Another attack described in [26] uses high-power signals on the right frequency that causes the Automatic Gain Correction (AGC) circuit in the

6 The same is true for other GNSS systems

receivers to tune down the sensitivity. In this case, the weaker and *real* GNSS signal will be too weak to pick up. This attack works well, because the AGC circuit is one of the first parts to receive the signal, even before it is processed.

One event showed that attacks mentioned above can happen. In 2013, a radio navigation research team from The University of Texas managed to spoof the GPS signal of a 80 million dollar yacht in the Mediterranean Sea [142]. The yacht's navigation relied on the civilian GPS signal, and the attackers slowly overpowered the signal with a counterfeit signal. This way, they managed to alter the direction of the yacht.

It is also possible to attack the software side of a GNSS receiver. Although this will be a black-box attack, in [100], a few attacks have been tested, such as the the 'middle-of-the-earth attack', 'week number attack' and 'date-desynchronization attack'. For all of these attacks, a special transceiver is used, who can capture, modify and retransmit the signal in real-time. For the first attack, the 'semi-major axis[7]' information of the satellite is set to zero. In this case, the altitude calculations failed in some receivers, potentially due to division by zero errors. Because most GPS receivers cache this information, it will continue to crash, until the cache is cleared. The second and third attack are quite similar, and involve triggering overflows due to bogus data. Most receivers accepted invalid week numbers, which would be a problem for time-critical applications. But receivers that store the time in a Unix-timestamp, a 32-bit integer (which will roll over in 2038), crashed when the timestamp was altered to exceed the maximum value.

Not only hard and software are susceptible to attacks. In [130], a social type of attack is demonstrated, where researchers have shown that it is very easy to fake traffic situations in Google Waze. Their attack exploited common traffic patterns such as traffic jams, and emulated the behavior on (virtual) devices by influencing the GNSS data.

Countermeasures

Any countermeasure presented below have the goal to mitigate or detect the attacks presented in the previous section. They do not fix the common accuracy problems mentioned in Section 3.1.2.

In [150], a few relatively easy countermeasures are presented to detect and overcome jamming and spoofing. The most easy one is keeping track of the absolute signal strength, a so-called receiver autonomous integrity monitoring system. Since GPS is very weak, it is relatively easy to transmit a counterfeit signal that is a couple of magnitudes stronger. Additionally, if the receiver measures the relative signal strength, any deviations could be an indication of an adversary. It is worth noting that new technologies allows an attacker to copy the signal (strength) in more detail. By using these new technologies, integrity monitoring will fail to detect tampering.

Another option is to monitor the satellite identification codes. Spoofed data may use invalid codes, or codes that are not plausible. Software can identify spoofed data that use impossible codes. Also, the timing information from real satellites is regular and predictable. Any deviation from this timing can indicate tampering. Finally, a sanity check is suggested. By using a IMU, the received data can be 'verified': if the receiver did not move according to the acceleration measured by the IMU, something is likely to be wrong.

7 Consider this the height of the satellite, or the 'larger radius' of an ellipse.

Another approach is to verify a signal with another signal. A first approach is demonstrated by [96], where an untrusted signal is correlated with a known and secure signal. It needs two receivers, and the difference between the signal phase[8] is used to detect an adversary. The idea is that the phase difference between two antennas can be calculated if the positions are known, and that a real GNSS signal comes from the sky. So if antenna A measures a certain phase of the C/A signal, it is possible to calculate what antenna B should register. If this is not within a certain margin, an adversary could be spoofing one of the antennas. A similar method can be found in[114]. It introduces an alternative, by cross-correlating the military signal of the defended receiver to the military signal of a secured receiver on the same frequency. If the correlation is large enough, by an appropriate statistical measure, then the null-hypothesis of no spoofing is accepted. Otherwise, a spoofing alert is issued for the signal. The algorithm can detect an attack in approximately 1.2 seconds. Both methods require two separate antennas with a secure link between them. The secured antenna needs to be thoroughly secured to guarantee authenticity, because it requires little power to jam or spoof GPS over a large distance. This makes it less practical for mobile applications such as AV.

Cross-correlation

In [20], a method of removing the inference signal is presented, called 'Adaptive Notch Filtering'. This method, demonstrated in Figure 10, is based on the cancellation principle. It has reduced computational requirements and for its good performance in the presence of CWI and due to the fact that commercial jammers produce a swept CWI. A notch filter, in contrast to a band-pass filter, rejects a certain part of an input signal. In the work, this method is extended by dynamically choosing what parts of the signal should be muted or not, by looking at the characteristics of the jammed signal.

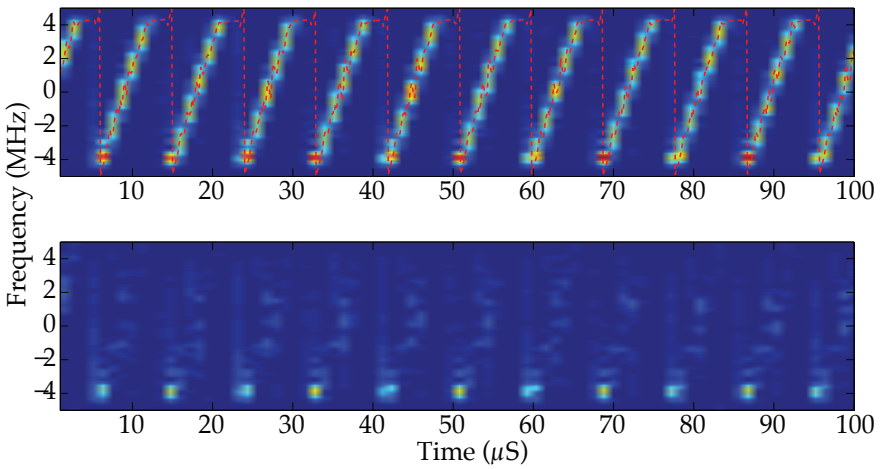


Fig. 10: Adaptive Notch Filtering applied to CWI. This is the same image as in 9, but then with the algorithm applied. Graph modified from [20].

Navigation Message Authentication

The last countermeasure uses encryption to achieve Navigation Message Authentication (NMA). With NMA, the authenticity of GNSS messages can be verified, therefore knowing if the data is from a real satellite, or if it spoofed. It can also be used, with some extensions to the current protocol, to protect against replay attacks or security code estimation attacks, according to [63]. Unfortunately, the current GPS protocol does not have support for NMA. Via extensible civil navigation messages, it is possible to add NMA. These messages are transmitted each six seconds, and can carry approxi-

8 The phase denotes the part of the sinusoidal wave that passed.

mately 238 bits of payload. Because of this limited size, it is not possible to pick any scheme for signing messages. In addition, verifying a signature should be computational inexpensive, to support small and cheap receivers. In [70], a hybrid ECDSA-TESLA scheme is proposed to periodically broadcast authenticated messages, which the receiver can use to authenticate the satellite. Elliptic Curve Digital Signature Algorithm (ECDSA) ensures message integrity. Timed Efficient Stream Loss-Tolerant Authentication (TESLA) is used to improve the efficiency, because it is loss-tolerant. A new ECDSA signature is sent periodically. In-between, it is updated with TESLA. This lowers the overall computational cost, as a receiver has to verify less ECDSA signatures. The time is used to protect against replay attacks. Other schemes exists, but the proposed one is efficient in terms of payload size, link requirements and provided security.

Galileo will have NMA as a built-in feature, according to [108]. Although the new GPS-III will have an 'enhanced security architecture', there is no confirmation of what this will actually include [83].

3.1.3 Camera

A camera is an optical device that can perceive the world as a digital video signal. It is frequently found in AVs for many applications. For example, it is used to detect traffic signs or to understand road scenery. An AV uses this to decide what to do next, and to understand what is not possible.

First, the two types of sensors are introduced. This is interesting, because it shows the hardware limits that should be considered. Then the applications will be presented, concluding with some attacks.

CCD and CMOS

There are two types of image sensor, the Complementary Metal Oxide Semiconductor (CMOS) and the Charge-Coupled Device (CCD) sensor. Strictly speaking a CMOS or CCD is an electrical component, and the actual sensor consists of a grid of them (called pixels). The number of components (thus pixels) refer to the megapixel count. Both sensor accumulate an electrical charge proportional to the amount of light received (photon-to-electron conversion), independent of color. The charge (a voltage) is then converted to a digital signal by an analog-to-digital converter. The voltage limit and the number of steps to represent a voltage as a digital value are a measure for the maximal brightness. If too many photons are converted to electrons, the voltage limit may be reached. This will then result in a white pixel.

To perceive individual colors, a pixel can consist of three individual subpixels with color filters mounted on top. Alternatively, one can have three times the sensor hardware, use a prism to separate colors and combine the three separate images into one via software. The amount of converted photons is expressed as the Quantum Efficiency (QE) ratio:

$$\mathrm{QE} = \frac{\mathrm{electrons/sec}}{\mathrm{photons/sec}} \tag{3}$$

There are a few differences between CCD and CMOS, according to [80]. The most important differences are listed below.

NOISE A CCD tends to pick up less noise, because the sensor size can be smaller and uses less components and circuitry.

QUALITY The CCD sensor works better in low-light conditions, because CCD is more sensitive to light.

SHUTTER CCD uses a global shutter, so it takes pictures at once. CMOS uses a rolling shutter, one line at a time. A global shutter can cause lag in moving pictures, but a rolling shutter causes deformed images such as curved straight lines. This is a concern in military target acquisition applications. Figure 11 shows why.

POWER CONSUMPTION Power consumption is important for mobile devices. Typically, a CMOS sensor uses less energy, because the circuitry to convert the analog signal directly into digital signals, at the cost of sensor size.

Fig. 11: The effect of rolling shutter. The blades are deformed because each line of pixels is samples at a later time frame. Image taken from [153].

Image quality

For AV applications such as object detection and tracking, cameras provide moving images. The image quality is affected by several factors and can cause that objects will get unnoticed, or increase processing time, according to [77].

In many cases, the camera can only be of limited size. For example, VisLab's BRAiVE AV has ten cameras installed [22, 52], including ones in the side mirrors. Better optical systems require more space, but can provide sharper pictures, allow for worse light conditions and reduce glare. To show the problem, a full-frame sensor of a professional Digital Single-lens Reflex (DSLR) camera is typically 36x24 mm, the iPhone 5s camera sensor sensor is 4.54x3.42 mm. This is 55 times smaller, but this does not reflect in the number of megapixels (16+ versus 8), meaning the physical CMOS or CCD sensor is smaller and receives less light in an iPhone 5s.

According to [72], there are also other problems with camera-based solutions. Cameras need lenses and lenses can distort the image (e.g. fisheye view or barrel distortion). This requires software correction of the images, before they can be properly used. In setups that rely on multiple cameras to provide vision, cameras tend to be calibrated regularly, to minimize the distortion between cameras.

Multi-band Images

In [67], multi-band images were used to further improve images quality by capturing far-infrared images (700 nm - 1200 nm) together with normal images (400 nm - 700 nm). The advantages of this, is that temperature is included, because of the infrared light. In most cases, this allows for better distinction of scene objects, which can be seen in Figure 12. This technique

can help to perceive objects better when light is limited, such as during the night. Nevertheless, for instance in [50] it was shown that Lidar outperformed camera vision significantly, for a certain lane detection algorithm under bright and dark environments.

(a) Visible spectrum **(b)** Infrared spectrum

Fig. 12: Normal capture compared to far-infrared capture. Some details are very well visible in the far-infrared image, such as the traffic sign, poles and the roof. Image taken from [67].

Even if the resulting image is of acceptable quality, there can be other problems that can affect algorithms performance. In the next section, a few applications will be discussed, where the performance greatly depends on the image quality, such as lighting conditions and shadows.

Applications

There are many implementations that use the camera as their primary source for a certain application (either detection, tracking or classification). The list below lists a few of the application encountered in the literature. A few applications are a conjunction of multiple sensor technologies, but mainly focus on the camera.

- Lane detection [149, 25, 14]
- Horizon/vanishing point detection [73]
- Object detection and tracking (vehicles, pedestrians) [52, 38, 69, 82]
- Traffic sign recognition [13, 68, 93, 81]
- Headlight detection [158, 33]
- Terrain classification [116, 5, 67, 134]

Because the tremendous number of applications, this section does not go into detail of each application[9]. Instead, a few common techniques are discussed. Most applications share a common task: extract interesting regions from an image (segmentation), extract features from these regions and classify them with common classifiers such as AdaBoost classifiers [156] or Support Vector Machines [105]. Segmentation is the most interesting step regarding the topic of this work. This determines what will be considered in the next steps. To find these interesting regions, several approaches are possible.

The first one is color channel thresholding, frequently used for finding traffic signs, lanes and headlights. By looking at different color channels (e.g. red), interesting segments can be detected. According to [93], it is better to use the hue, saturation and intensity for splitting colors, since it better

9 It is also not the intention of this work to go into application details.

models the human eye. Figure 13 shows an example of this process. Because it may not be certain that the color is actual red, improvements to this process include creating a gradient of colors before searching, so it is known which colors are likely to be expected. This can account for worse light conditions. In addition, looking for shapes or symmetry can also reveal interesting area's, with edge detection algorithms where a change in color may denote an edge [67, 93].

Fig. 13: The result of thresholding an image, looking for red hues. Image taken from [93]

In [68], a rather different approach is considered. They introduce a heat map of places where traffic signs are expected. This idea is intuitive, because traffic signs have a great chance to appear on the right or in the overhead. This limits the chance of a traffic sign appearing in other regions of the camera's view.

A Motion-based approach is discussed in [38]. This approach defines the differences between two successive images to detect regions that have changed. These regions can then be classified as an object, for instance as a car or pedestrian. This method is relatively simple, but is very sensitive to background changes. Research is devoted to the dynamic modeling of the scene background, to overcome this problem.

Haar-like features are another way of finding regions of interest. This method is frequently used for (but not limited to) face detection algorithms. As with other object detection algorithms, each pixel could be of interest, so it requires many cycles to brute force search an image. Haar-like features solve this problem by presenting 'filter' with increasing complexity. If a pixel is not of interest, no further processing will be applied onto a pixel.

A feature describes a change in contrast of a group of pixels, as opposed to the pixel intensity [154]. Some examples of Haar-like features are given in Figure 14. An object-to-detect consists of a several of these of features, also called a cascade Haar-like feature. For instance, a rectangle requires several edge features. For each image, an integral image is calculated. An integral image is an array of the sum of the pixels' intensity values that are located directly above the pixel at a certain location. With this representation a cascade of features can be compared to the integral image, to see if the cascade yield any matches. The details are outside of the scope of this work, but the idea is to place Haar-like features on the grayscale image and see where it 'fits'. A feature is flexible, so it is easy to scale them and detect objects of different sizes. If all features match, then the object is detected.

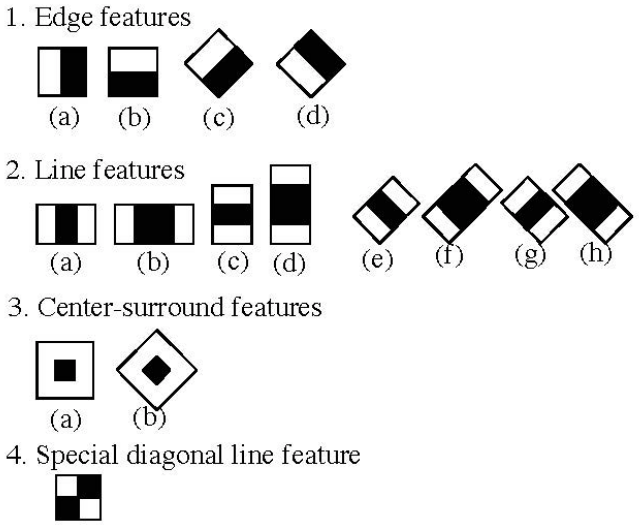


Fig. 14: Common Haar-like features. A feature represents a change in contrast. Image taken from [154].

Stereoscopic Images

An interesting topic is the use of two cameras to provide three-dimensional vision. With such a view, it is possible to see depth in images. The technique requires two cameras on the same height. The two images will be combined, whereas object closer to the camera will have a smaller shift between both images. From this information, a depth map can be created. When calibrated correctly, this map can be used for several other algorithms, such as range detection. Because of the depth map, it is also possible to distinguish foreground objects from the background. Figure 15 presents a simple setup of stereoscopic imaging.

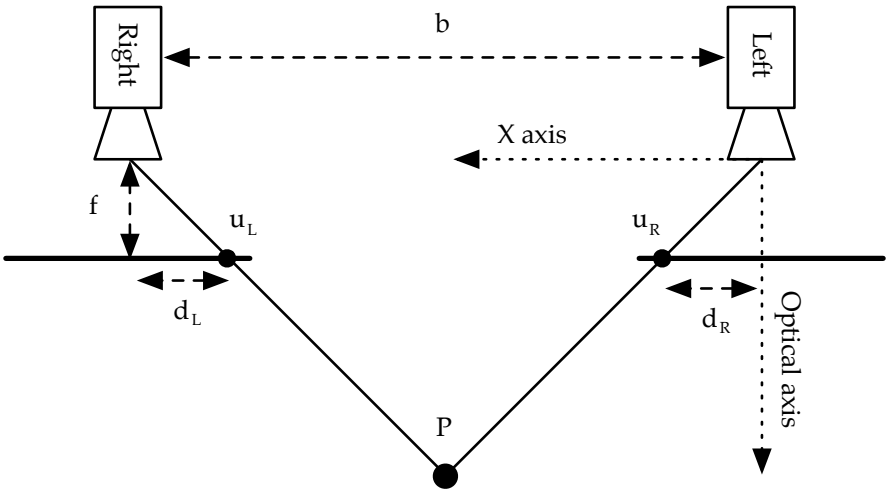


Fig. 15: A simplified setup of stereoscopic camera vision system with two cameras. Point P is a point in the world, projected by u_L and u_R on the images.

In the figure f is the focal length. This can be calculated by placing an object of known width in front of the camera. Then, the ratio between the number of pixels of the width of the object, and the real width, is the focal length. b is the distance between the cameras. d_L and d_R is the displacement of the point u_L and u_R, which represent the same point P. To calculate the depth of point P, one calculates the following [69]:

$$Z = \frac{b \cdot f}{|d_l - d_r|} \tag{4}$$

In 2014, VisLab s.r.l released a product called '3DV', which uses two cameras that can output a three-dimensional map of the world, and detect ob-

jects. However, the system still has issues with adverse weather conditions, especially snow.

Attacks

The first attack will physically attack the sensor. A CCD or CMOS sensor can be (partially) destroyed with a laser[10]. Even a low-power laser can burn the sensor at an instance, with irreversible damage. Low-power laser of less than 5 mW are easy to find on the internet, or they can be harvested from CD/DVD writers [91]. In a white paper published by [91], the following experimental conclusions were drawn:

- A Class II laser of 1 mW was never able to permanently disable a CCTV camera when at least 3 meters away from the lens.
- A green Class IIIa laser of 5 mW was able to disable a CCD-based camera at 15 meters which resulted in a permanent white screen.
- A red Class IIIa laser of 5 mW was not able to disable a CCD-based camera (see above) at 100 meters.
- Lasers in the Class IIIb power range (both red and green) are able to destroy the CCD or CMOS sensors, resulting in white images.
- Outdoor applications would be realistically more vulnerable than indoors due to the simple nature of the risks associated with attempting to disrupt/damage a camera system.

A military weapon called a 'Dazzler' is intended to (temporary) blind camera vision (or human vision). This grade of hardware can operate up to 1000 m [8], and is not much bigger than a automatic rifle. Figure 16 shows an example of such a weapon and how the beam looks like.

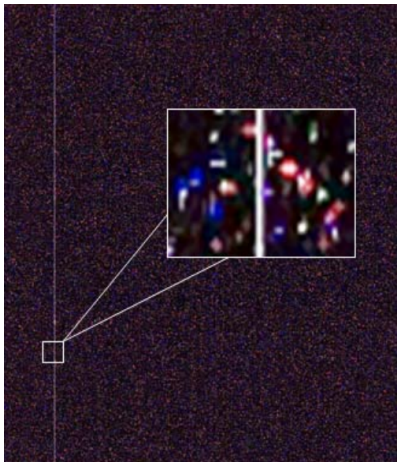
(a) Example of a Dazzler, the 'PHaSR'. **(b)** Beam produced by a Dazzler.

Fig. 16: An example of a non-lethal Dazzler weapon and the a beam. Images taken from [2] and [133].

According to [18], when a powerful laser hits the image sensor, it can reduce the quality of the silicon. This reduces the quality of the pixel's charge transfer property. When it reduces the quality, the pixel gets stuck, and may be always on or off. If it gets destroyed, the signal cannot travel via neighbor pixel anymore, since pixels are typically multiplexed (row/column wise). In this case, it will destroy a complete row and column. Figure 17b shows this as a vertical line. This attack does not require the camera to be powered on. As long as the image sensor is exposed, this attack works.

10 A burn happens almost instantaneous. See http://vimeo.com/13450755 or http://vimeo.com/56074271 for example footage of how DSLRs are damaged by laser shows.

(a) The white spots are the results of laser burns.



(b) Magnification of a black image, with dead pixel row and stuck pixels.

Fig. 17: Laser damaged CMOS sensor. Images taken from [18].

Another attack can be mounted on the auto exposure. While this attack is not described in literature, cameras tend to normalize the lighting conditions via an iterative process [147, 36, 71]. When light (e.g. sunlight) is exposed on the image sensor, it will tune down it is sensitivity and exposure to improve the image quality. Sometimes, this gives undesired effects, in cases in which the auto exposures tunes down due to headlights at night. This could hide information in the background, such as traffic signs, road edges or pedestrians. The Google Driverless Car is susceptible to this problem [48]. Potentially, this can work with infrared light too. Infrared light is invisible to the human eye, but most cameras are (highly) sensitive to it. Even though cameras have infrared filters installed, they still pass a bit of infrared light.

Considering the different applications presented, there are applications that can be spoofed. For instance, traffic sign recognition can be spoofed by placing traffic signs on places they should not be. This can also happen unintentionally, e.g. in bill boards. It is also possible to 'hide' them, by surrounding traffic signs with other shapes or colors, to confuse shape or color detection algorithms. Furthermore, lane detection could be confused by painting additional lines on the road, or by using different colors. This is already the case at road construction sites. Lastly, some applications have limited capabilities. Object or pedestrian tracking is usually limited because of computational power or resolution. It would be very easy to cause a denial of service by presenting many objects that should be tracked.

In [56] and [30], the authors have experimented with fooling Haar-like features for facial recognition. In the first, a rather artistic approach is taken, where the authors came up with several hair coupes to prevent a face recognition system based on Haar-like features to prevent working. The second takes a similar approach, with modified glasses that emit infrared light on places where the Haar-like features would normally match.

3.1.4 TPMS

A Tire-pressure Monitoring System (TPMS) is a small device equipped on the valve of each tire, regularly providing information about the tire to the car's Electronic Control Unit (ECU). This information at least includes tire

pressure, but can also includes temperature and acceleration. Since 2008, new cars in the United States are required to be equipped with a TPMS system, as mandated by the TREAD Act in 2007 [131]. A similar law exists in Europe, that was adopted in 2012 [34].

Currently, there is no evidence of AVs incorporating tire pressure sensor data in decisions making algorithms. This separates a TPMS from the other sensors. but if future AVs should operate on their own, it is expected that spoofing or jamming a TPMS can immobilize a car [24]. An AV would not be smart to ignore a flat tire. Further more, the privacy issues are of a concern too.

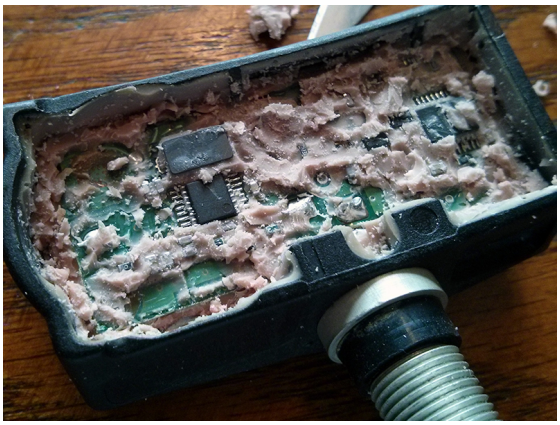
Hardware

A TPMS sensor is a sealed battery-powered device, that fits on the air nozzle of each tire. Figure 18 shows an old one, collected from a garage. Typically, the battery should last five to seven years [88]. There is no standardized protocol, hence every car manufacturer has its own proprietary hardware design and communication protocol. Most sensors work on the general purpose 315 MHz, 433 MHz or 866 MHz frequency modulated using either Frequency Shift Keying (FSK) or Amplitude Shift Keying (ASK). In addition, some sensors listen on a secondary frequency in the range of 125 KHz, that allows them to be woken up from sleep mode, using a special device. This is useful during installation and configuration, by an authorized dealer. The sensor is required to broadcast its status each 60 to 90 seconds.

Usually, the range is limited and just enough for the car to pickup the signal. But the researchers from [119] found out that a range of 40 meters is possible with low-cost hardware.

Fig. 18: Front and back of a TPMS sensor of the brand Renault. The size is approximately 6x3cm. The cylinder on the bottom side is installed on top of the tire valve.

(a) The unique identifier is shown above the CE sign. The operating frequency is shown on the right.



(b) Inside the sensor, after removing the glue. The coin cell battery sits below the printed circuit board in the left.

Security and privacy

While the TREAD Act only mandated the use of TPMS systems, no requirements were specified regarding the security and privacy of these systems. Researchers have argued about the safety and privacy concerns.

In [119], it was shown how easy it is to track and monitor cars equipped with TPMSs. The authors have reversed engineered the protocol of two brands of cars. Both cars used TPMS sensors that operated on the 315 MHz or 433 MHz frequency. With the help of Software Defined Radio (SDR), they

Software Defined Radio

were capable to receive the messages broadcasted by the sensors. Eventually they deciphered the messages by looking at which parts of the message were constant over time, and found out that the unique device identifiers did not change over time for those two sensors. Three conclusions can be drawn from their work. First, the immutable unique identifier of 32 bits, bound to the sensor, does not change over time. This poses a privacy issue, because cars can be tracked from a large distance, without the driver knowing it. In [119], it was calculated that it takes at least 110 SDRs to capture one broadcasted message per 60 seconds, assuming a car drives at a speed of 60 km/h. More sensitive antennas could drop the number of required devices significantly. The second issue is the protocol they observed. Broadcasted messages are not encrypted nor authenticated, so it was possible to replay and spoof (impossible and invalid) messages, fooling the car the tires were flat while they were not. At some point, the authors even managed to damaged the ECU of one of the two cars, that had to be replaced eventually. Third and last, they pointed out that battery exhaustion attacks can be conducted, by triggering the activation signal repeatedly. This even works when the car is not driving.

For a long time, the 'conclusive answer' on why security and privacy was not considered, was that hardware required to track was too expensive, and an attack was thus not feasible. But recently, SDRs have become very cheap, and can be bought for five 5 dollars on eBay. Furthermore, in [119], a professional SDR was used that costs at least 1,000 dollars per radio. With the cheaper alternative, the costs of the attack drops from 110,000 dollars to only 550 dollars[11]. In 2012, Jared Boone presented a talk on a conference [19], in which he showed an open source toolkit to extract unique identifiers from broadcasted messages, received with cheap SDRs. His toolkit supports FSK or ASK modulation, and includes several statistical analysis to inspect the fields of a message (e.g. Cyclic Redundancy Check (CRC) calculation).

Countermeasures

Some countermeasures can be implemented to improve and secure TPMSs, according to [119] and [19].

The first suggestion is to improve the ECU software. Clearly, the ECU trusted readings that were spoofed (and even physically impossible). Even plausibility checks would improve the design. Second improvement is the data packet format. At least all fields that uniquely track a car should be encrypted. The key should only be shared between the sensor and the ECU, during installation time. In the two sensors [119] observed, no sequence numbers were integrated. This made replay attacks possible. Last improvement should prevent the abuse of the trigger signal to exhaust batteries. A simple protocol could be implemented, where the ECU and the sensor share a common counter. The ECU generates an one-time hash, and authenticates itself with this hash to the sensor. This way, the sensor is not required to power up it is transmitter, in case an attacker replays messages.

One physical issue that remains, is the limited form factor. Its size constrains the energy capacity. This limits the use of encryption, since that would wear out the battery at a higher pace [60]. A solution over time could include more energy efficient Microcontroller Units (MCUs) and higher capacity batteries.

11 110 devices times $5 low-cost SDRs versus 110 devices times $1000 professional SDRs.

3.2 SENSOR FUSION

In the considered literature of the previous chapter that address applications, sensor fusion algorithms are commonly used to combine and correct data from multiple sensor in such a way that data quality can be improved to yield better (verified) results. The two popular choices in works are Kalman Filter (KF) and Particle Filter (PF).

Both data fusion algorithms will provide an answer to the question 'How to get accurate data from inaccurate data?' As an intuitive example, consider a car that accelerates forward, which one tries to localize. The acceleration and position is measured, with error. The position error is bigger. The basic idea behind sensor fusion is that if the acceleration sensor measures acceleration in a certain direction, the position should be on the same line, and not in another direction. If so, attach less weight to the position. After all, both sensors have a certain error. In essence, this is how GNSS data is combined with an IMU for inertial navigation, or with an odometric sensor for dead reckoning.

Dead Reckoning

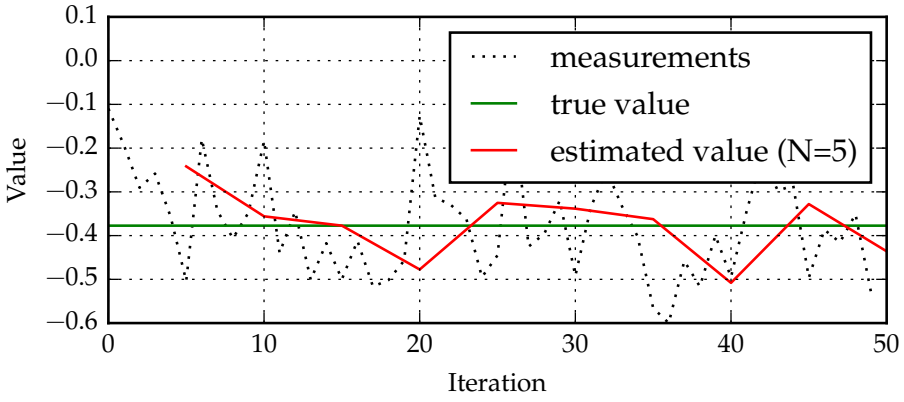
The KF is presented first, then the PF. After the introduction and the mathematics, a brief introduction to attacks and countermeasures is presented.

3.2.1 Kalman Filter

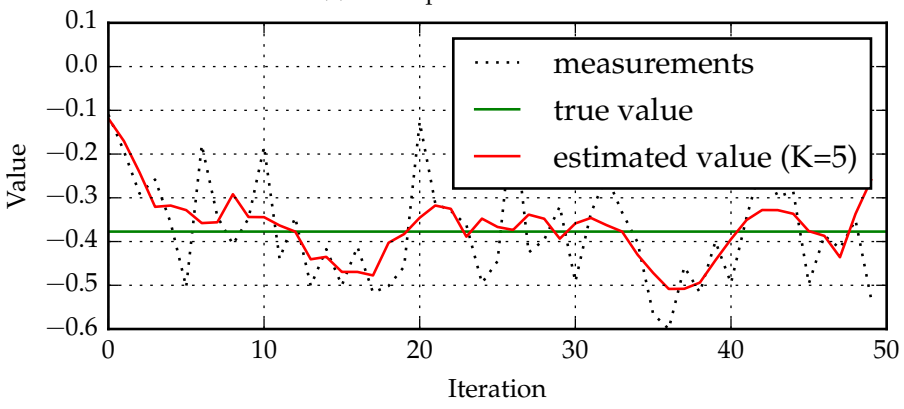
Rudolf Emil Kálmán published the initial variant for the KF in the 1960s [66, 51]. Shortly after the publication, it has proven its use as part of the Luminary 99 Lunary Module Guidance Computer during NASA's Apollo 11 Space Program [51]. It is not only used in the field of robotics, but also in fields such as meteorology [45] or stock exchange [94].

The KF is an optimal recursive discrete processing algorithm. Optimal, since it only depends on the criteria chosen to evaluate the performance, and recursive, since it does not require the previous data to be stored or reprocessed. It is also uni-modal, at any point in time, there is one estimation available. This includes the advantage that it does not imposes huge memory space requirements, making it an efficient algorithm, even for devices with memory constraints, e.g. MCUs. Discrete refers to the fact that it can only predict for full iterations, not between iterations.

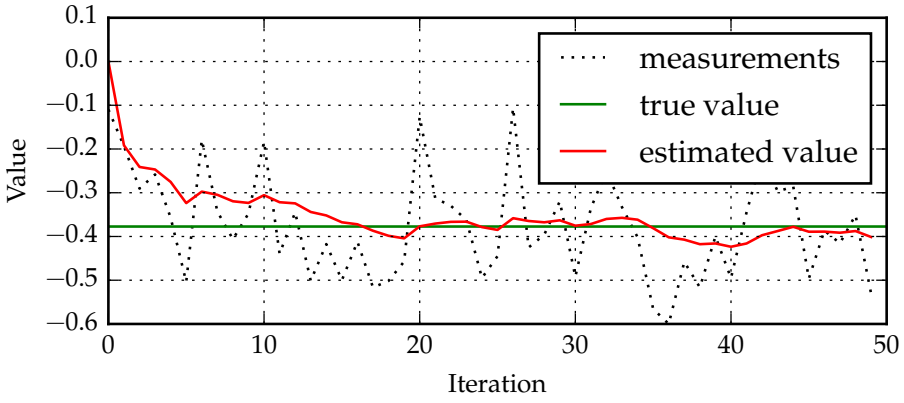
There are several other techniques for filtering (see Figure 19). One could consider taking ten samples, and calculate the average sample. This will reduce outliers, or noise, but the problem is, that it takes valuable samples away. In the case of GPS, with an interval of 10 Hz, the sensor output rate is reduced to 1 Hz. For many automotive applications, this is not desirable. Estimating the error on-line is not possible, because the actual error is unknown during movement.



(a) N-sample mean filter.



(b) Rolling K-window filter.



(c) Kalman filter.

Fig. 19: Several examples of data filters applied to the same data set of a random simulation. In each filter, the true value is not involved in any of the calculations, and is only plotted to give an idea of the error. Note the missing data with N-sample mean filtering, since it takes N samples to determine the first point, and so on.

Generally speaking, the standard or discrete KF has four steps, spread over two phases: a correction phase and a prediction phase. Figure 20 shows these. The steps form a cycle, which is repeated indefinitely. In this appendix, the variable i refers to the iteration cycle. Thus, $i+1$ indicates the next cycle, and so on.

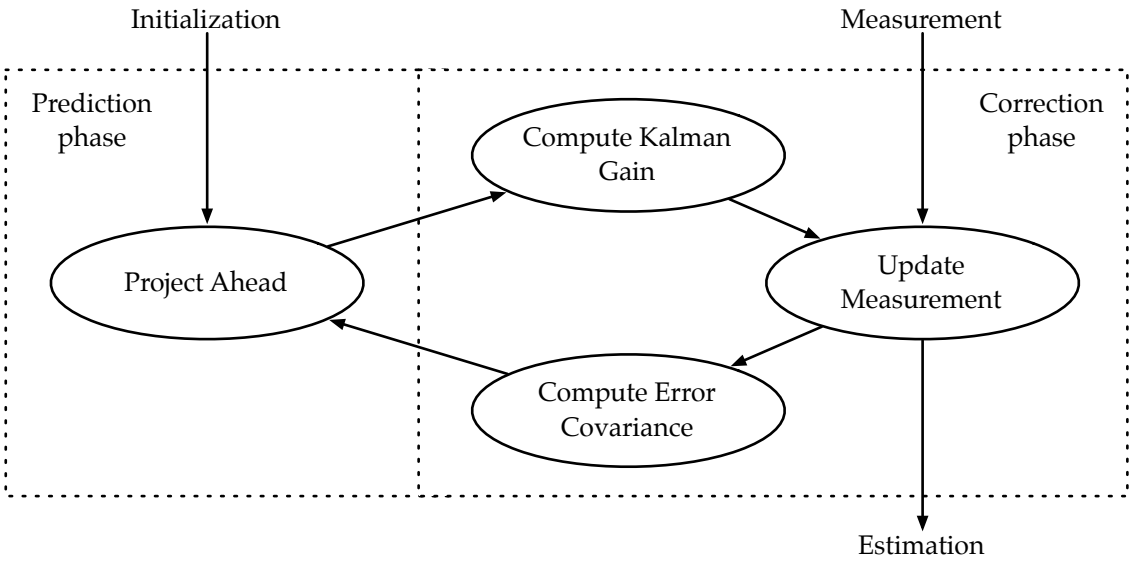


Fig. 20: The four steps of a KF, spread over two phases.

Steps of Kalman Filter

Single-variable system

To understand the KF better, the equations below are applicable for a single-variable KF. These equations do not include matrix operations. In the next section, for a multi-variable filter, the equations below will be related to their matrix equivalents.

Equation 5 and 6 are the in the 'project ahead' step. In this step, the new values of prediction of the estimate x and covariance P are determined. For this explanation, they are assumed to be equal to the previous estimate and covariance, plus the 'process covariance' Q. The estimate refers to the value that is being estimated by the KF (x_i in this case), while the prediction refers to the variable that is being predicted during the process ($\hat{x}$ and $\hat{P}$), indicated with a circumflex symbol ($\hat{()}$).

$$\hat{x} = x_{i-1} \tag{5}$$

$$\hat{P} = P_{i-1} + Q \tag{6}$$

In the 'Kalman gain' step, the gain is computed. The gain can be interpreted as measure of how much the predicted value will be corrected. R denotes the *measurement covariance*.

$$K = \frac{\hat{P}}{\hat{P} + R} \tag{7}$$

The *update measurement* step involves the sensor measurement z_i. Here the Kalman gain is applied to the error between the estimation and the real-world value, to correct the predicted value $\hat{x}$. The output x_i is the estimate, and is an output of the system.

$$x_i = \hat{x} + K * (z_i - \hat{x}) \tag{8}$$

Finally, in the last step, the *error covariance* is updated for the next cycle.

$$P_i = (1 - K) * \hat{P} \tag{9}$$

Multi-variable system

A multi-variable system is more common to use, but are more complex. Typically, GPS with IMU dead reckoning uses a 6-variable KF, to estimate the X and Y coordinates from position, acceleration and velocity data. The equations from the previous section are rewritten, to provide a general framework for a KF. Where appropriate, equations are split to be more clear.

Equation 10 and 11 are a extension of Equation 5 and 6 in the 'project ahead' step. In the single-variable system, x represents a single value, while in this system, it is a vector. Furthermore, $\mathbf{A}$ is the state transition matrix, and describes the transformation to apply to x_{i-1} and P_{i-1}. The single-variable system did not represent $\mathbf{A}$, but if it would, it would be equal to the identity, since the prediction of the new estimate is directly related to the old estimate. $\mathbf{B}u$ is the force added to the prediction. A practical overview of what these variables contain, will follow in the example.

$$\hat{x} = \mathbf{A}x_{i-1} + \mathbf{B}u_i \tag{10}$$

$$\hat{\mathbf{P}} = \mathbf{A}\mathbf{P}_{i-1}\mathbf{A}^T + \mathbf{Q} \tag{11}$$

For the Kalman gain, Equation 7 is translated to Equation 12. The matrix $\mathbf{H}$ is the observation matrix. This matrix basically tells which measurements should be considered in the current cycle. Note that is not possible to divide a matrix by another, but it is possible to multiply with the inverse.

$$\begin{aligned} \mathbf{S} &= \mathbf{H}\hat{\mathbf{P}}\mathbf{H}^T + \mathbf{R} \\ \mathbf{K} &= \hat{\mathbf{P}}\mathbf{H}^T\mathbf{S}^{-1} \end{aligned} \tag{12}$$

The real-world measurement is involved in the update measurement step, by equation 13. Note that the new estimate always lies between the real-world measurement and the previous estimate.

$$\begin{aligned} \tilde{y} &= z_n - \mathbf{H}\hat{x} \\ x_i &= \hat{x} + \mathbf{K}\tilde{y} \end{aligned} \tag{13}$$

And again, the *error covariance* is updated. $\mathbf{I}$ denotes the identity matrix.

$$\mathbf{P}_i = (\mathbf{I} - \mathbf{K}\mathbf{H})\hat{\mathbf{P}} \tag{14}$$

Practical concerns

The KF assumes one can model the noise of various sensors and systems into a covariance matrix. The better the noise is modeled, the better the estimation will be. However, since the filter has to converge to a 'good' estimation first, one typically has to discard the first few samples [53]. Figure 19 illustrates this. Converging could lead to practical implications. For example, take a GPS receiver at 10 Hz, filtered via a KF. Assuming that the first 100 samples should be discarded, an additional time of 10 extra seconds should be considered, before the system is ready.

3.2.2 Particle Filter

The initial version of the PF algorithm was presented in the '50s, with several improvements over the years. The first publication regarding the PF as it is known today, is from Gordon et al. [49] in 1993. Not only is fusing sensor

data a typical application of a PF, tracking object (in visual applications) is a popular use too [110, 129].

Multi-modal

A PF is a multi-modal filter. This means it can have multiple beliefs or 'guesses' for an estimated value. It is a recursive algorithm, but compared to a KF, it is less advanced to implement. This comes at a cost: the algorithms performance scales linear with the number of particles, and it is very likely to require a powerful computing platform.

The typical flow of a PF is depicted in Figure 21. The output is not an exact estimate, but a set of weighted particles. A common practice is to take the weighted point average [140], resulting in a single point that represents the cluster. However, it would be safer to say that this single point is the current best belief of the estimate.

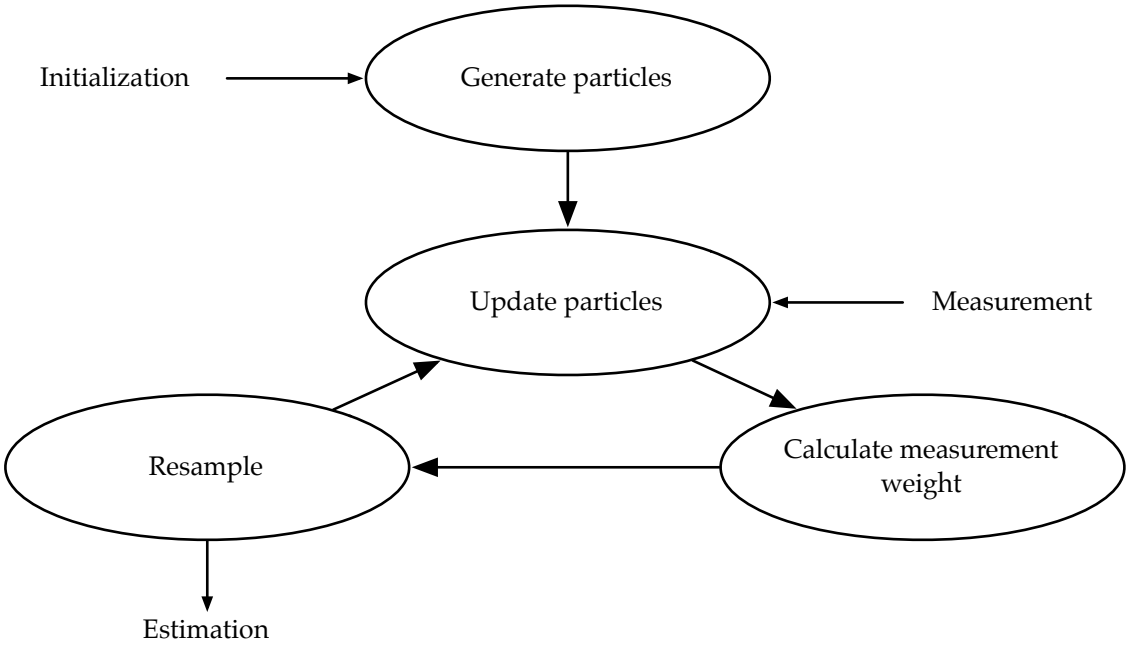


Fig. 21: A typical PF flow chart. Note that the estimation result is a set of particles.

Bayesian Filtering

This section is based on the work of [129]. To be consistent with the previous section, the i is used as iteration/time symbol, as opposed to k.

Bayesian Inference

A PF estimates the posterior density of the state-space by using the 'Recursive Bayesian Estimation' equations (also referred to as Bayesian Inference). It uses a sampling method rather than any action methods (as with KF) to avoid any issues with (non)linearity of data. The posterior probability of estimate x (the hypothesis) is calculated given the observation z (the knowledge). Equation 15 summarizes this, and figure 22 shows how the posterior is affected by the prior and the likelihood.

$$\text{Posterior} = \frac{\text{Prior} \cdot \text{Likelihood}}{\text{Evidence}} \tag{15}$$

More formally, where x is the state and z is the measurement:

$$P(x|z) = \frac{P(x)P(z|x)}{P(z)} \tag{16}$$

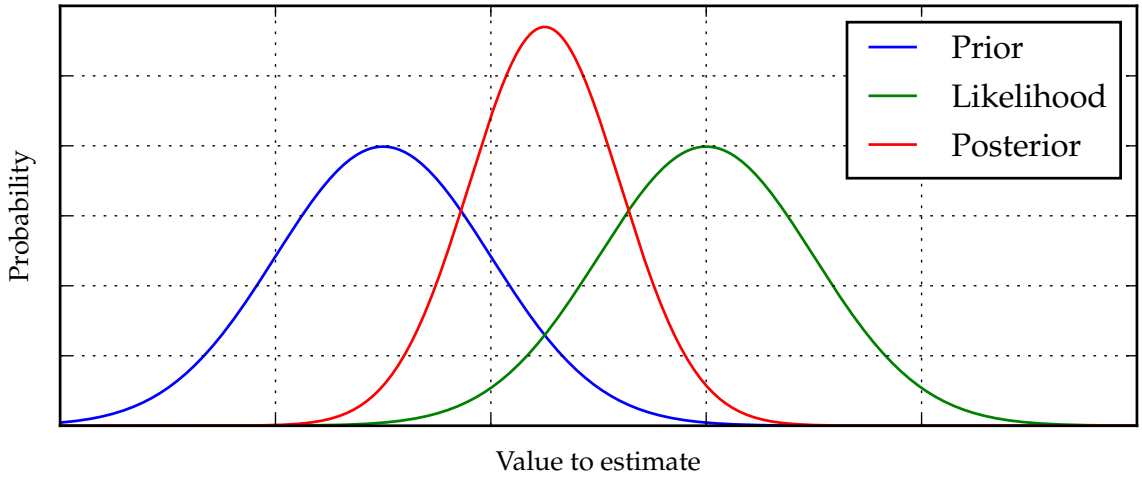


Fig. 22: How the posterior is affected by the prior and the likelihood. If the likelihood mean would increase (or the prior mean decrease), the posterior would follow the likelihood more. Displayed curves are normal distributions and are examples.

The PF is based on a Hidden Markov Model (HMM), the future depends on the past via the present. Basically, the past is completely ignored (or hidden) and represented by the present. Of course, this assumes there is some initial probability known to the system. As with the KF, the state of the system on time i in a PF is most likely a matrix $\mathbf{X}_i$.

$$P(\mathbf{X}_i|\mathbf{X}_0,...,\mathbf{X}_{i-1}) = P(\mathbf{X}_i|\mathbf{X}_{i-1}) \tag{17}$$

Furthermore, another part is the conditionally independence assumption of observations, given the state. In statistics, when two variables are independent, if and only if $P(A \cap B) = P(A) \cdot P(B)$.

$$P(z_k, z_i, ..., z_j|\mathbf{X}_k) = P(z_k|\mathbf{X}_k)P(z_i, ..., z_j|\mathbf{X}_k) \tag{18}$$

What makes it recursive, is that for each correction iteration, the posterior probability can be used as the the prior probability for the next iteration. Intuitively, it is possible to get a better estimate (using new data), based on previous estimations. As presented in figure 21, there are two major steps. Equation 19 is the prediction step and 20 is the correction step. In the equations below, z_i is a measurement vector at time i, thus $\mathbf{Z}_{i:j} = [z_i, ..., z_j]$.

$$\begin{aligned} P(\mathbf{X}_i|\mathbf{Z}_{1:i-1}) &= \int P(\mathbf{X}_i, \mathbf{X}_{i-1}|\mathbf{Z}_{1:i-1})d\mathbf{X}_{i-1} \\ &= \int P(\mathbf{X}_i|\mathbf{X}_{i-1}, \mathbf{Z}_{1:i-1})P(\mathbf{X}_{i-1}|\mathbf{Z}_{1:i-1})d\mathbf{X}_{i-1} \\ &= \int P(\mathbf{X}_i|\mathbf{X}_{i-1})P(\mathbf{X}_{i-1}|\mathbf{Z}_{1:i-1})d\mathbf{X}_{i-1} \end{aligned} \tag{19}$$

$$\begin{aligned} P(\mathbf{X}_i|\mathbf{Z}_{1:i}) &= \frac{P(\mathbf{X}_i)P(\mathbf{Z}_{1:i}|\mathbf{X}_i)}{P(\mathbf{Z}_{1:i})} \\ &= \frac{P(\mathbf{X}_i)P(z_i, \mathbf{Z}_{1:i}|\mathbf{X}_i)}{P(z_i, \mathbf{Z}_{1:i-1})} && \text{(Split matrix)} \\ &= \frac{P(\mathbf{X}_i)P(z_i|\mathbf{X}_i, \mathbf{Z}_{1:i-1})P(\mathbf{Z}_{1:i-1}|X_i)}{P(z_i|\mathbf{Z}_{1:i-1})P(\mathbf{Z}_{1_i-1})} && \text{(Bayes Rules)} \\ &= \frac{P(\mathbf{X}_i)P(\mathbf{Z}_i|\mathbf{Z}_{1:i-1}, \mathbf{X}_i)P(\mathbf{X}_i|\mathbf{Z}_{1:i-1})P(\mathbf{Z}_{1:i-1})}{P(\mathbf{X}_i)P(z_i|\mathbf{Z}_{1:i-1})P(\mathbf{Z}_{1_i-1})} && \text{(Eq. 17 + 18)} \\ &= \frac{P(z_i|\mathbf{X}_i)P(\mathbf{X}_i|\mathbf{Z}_{1:i-1})}{P(z_i|\mathbf{Z}_{1:i-1})} \end{aligned} \tag{20}$$

Particle Density Function

In systems where (a part of) the underlying model is nonlinear or non-Gaussian (e.g. nonlinear processes or rule-based processes), Equation 19, which computes the Particle Density Function (PDF), cannot be solved due to the integrals. When the state space is discrete, or when discretizing the continuous variables, solving the integrals is possible. If not, it should be approximated, and this is where the particles come in.

Particles

Since most PDFs cannot be calculated, they can be represented by a set of weighted particles. The particles, represented as a vector $\mathbf{x}_i^k$, meaning the kth particle (or sample) at time i (not to be confused with the state matrix $\mathbf{X}_i$). Each particle has an associated weight w_i^k, and all weights should be normalized to one (e.g. $\sum_{k=1}^{N} w_i^k = 1$). The PDFs can be approximated as follows:

$$P(\mathbf{x}_i|\mathbf{Z}_{1:i}) \approx \sum_{k=1}^{N} w_i^k \delta(x_{0:i} - x_{0:i}^k) \tag{21}$$

Since it is not possible to pick particles from $P(\cdot)$, the weights come from another importance density function $Q(\cdot)$, from which it is possible to sample directly. $Q(\cdot)$ can be chosen freely, as long as $Q(x) = 0$ implies $P(x) = 0$. For example, one could sample from a Gaussian or linear distribution.

$$w_i^k \propto \frac{P(x_{0:i}^k|Z_{1:i})}{Q(x_{0:i}^k|Z_{1:i})} \tag{22}$$

Resampling Methods

The core part of the PF is the (re)sampling function. There are many different resampling methods [44, 61], including the ones below.

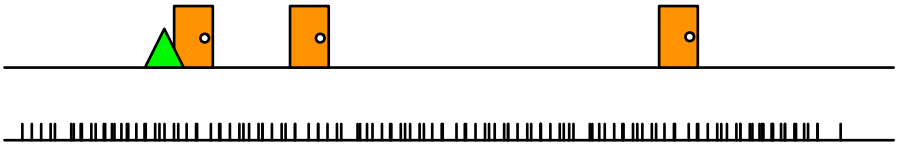
- Sequential Importance Sampling (SIS)
- Sequential Importance Resampling (SIR)
- Multinomial Resampling
- Stratified Resampling
- Systematic Resampling
- Residual Resampling
- Wheel Resampling

SIR is essentially the same as SIS, but includes resampling of the particles. Resampling in general solves the degeneracy problem, where all-but-one of the particles' weight is zero. In other words, most particles cover unlikely states. With resampling, this problem can be solved, because a new set of particles (with replacement) is created for each iteration, in essence respawning particles.

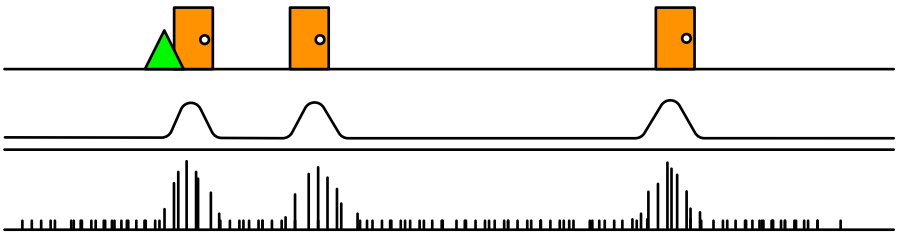
The are several resampling methods to choose from in the literature, but this is out of the scope in this work. The steps of the 'Resampling Wheel' method are presented below. This method is non-deterministic [140]. A circle is divided in N slices, where each slice is as wide as its weight. The perimeter is the sum of all weights. The algorithm starts at a random selected edge, and traverses the perimeter a random selected weight. It then adds the particle (with replacement) where it stopped traversing to the new set particles.

1. Calculate the maximum weight of the set particles w_i.
2. Pick a random particle index m
3. Initialize a new list empty list.
4. For each $n = 0..N$:
   a) Pick $\beta = 2 \cdot \text{random}() \cdot w_{i,max}$
   b) As long as $\beta > w_i^n$, calculate $\beta = \beta - w_i^n$, and $m = m + 1 \pmod N$
   c) Add particle with index m to the new list of particles.
5. Post-process the new set of particles, e.g. calculate a weighted average position.

To give an idea how the complete PF works, Figure 23 represents the classical robot localization problem in an one-dimensional world, with multinominal resampling. The robot can sense doors, but does not know it is location yet. The robot is presented in green, and has sensors to determine the location (with some error) of the orange doors. The black bars are particles at a position with some weight. The likelihood $P(z|x)$ is represented by the continuous curves, and the particles weight w_i^k are the black lines, and change in proportion to the likelihood.



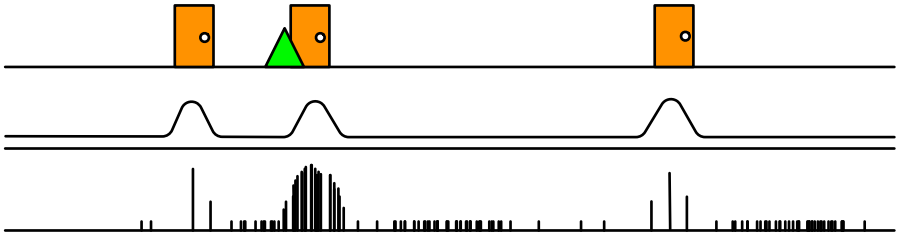
(a) Particles (black bars) are spread uniformly over the state space. It represents the PDF.



(b) Robot observes two doors. The PF is multi-modal, so there could be multiple observations and beliefs.

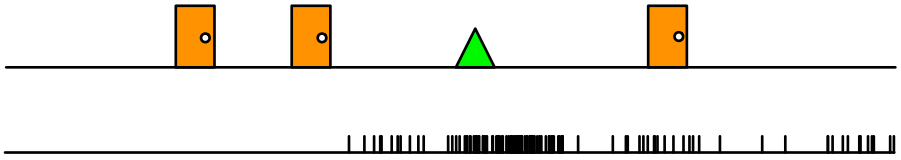
(c) After resampling, less-likely particles *are likely* to make room for very-likely particles. Weights are redistributed. Note the three dense clusters of particles.

(d) Robot moves to the right, and so do all the particles. The three clusters move along. The movement may introduce more noise.



(e) Robot makes another observation, and the particle weight is changed again.

(f) Particles are resampled. The dense cluster of particles show with high belief where the robot is located.



(g) Robot moves again. Since there are no observations, the particles will account for noise added by each movement. The cluster will spread until another observation is made.

Fig. 23: PF applied to a classical example of localizing a robot. The figure is based on the one in [44], with intermediate steps c and d added for clarity.

Practical concerns

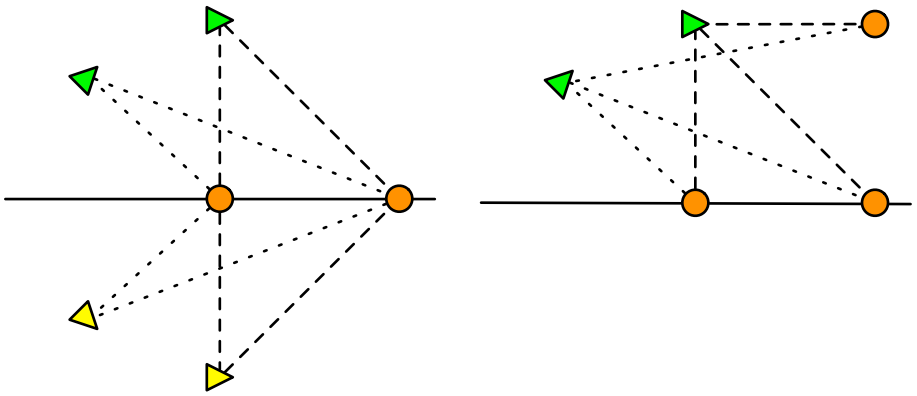
The number of particles, represented by N, determines the quality of the PDF. Having many particles will result in a better resemblance of the PDF for the given state space, at the cost of extra computations. If $N \to \infty$, the PF will converge to the true PDF. In practice, one will just try different values for N, until it yields acceptable results. It is very unlikely to see PFs with less than 50 particles.

Markov Property

According to [31], a PF assumes that the Markov Property holds. This is the stateless property of a stochastic process. The belief in the next posterior is affected *only* by the likelihood and the current prior. A random walk with the restriction of visiting the same route twice is one that violates the assumption. Another example is given by [126]. In this example, an urn with two red balls and one green ball is given. Each day, a ball is drawn, without putting it back. If one knows a red ball is drawn today, the chance a red ball will be drawn tomorrow is 50 percent, because the only options left are

$P(r, r, g)$ and $P(g, r, r)$. However, if one knows yesterday's and today's balls are red, it is 100 percent certain that tomorrow's ball will be green. This is where the Markov Property contradicts, since the probability distribution of tomorrow is not only affected by today's outcome, but also by yesterday's outcome.

Depending on the implementation of the PF, ambiguity can be a problem, as discussed above. Figure 24 demonstrates the issue. A measurement vector can be ambiguous if its reading can be projected on multiple (or infinite) locations in the defined world, as is the case in the left figure. In the right figure, with three beacons, a triangle can be constructed that fixes the coordinates of the object. It depends on the situation how many beacons are a minimum. Clearly, in the cannonball case study (see Appendix A.2), three beacons that form a triangle is a minimum, but this assumes the cannonball *can* measure the distance to the three beacons at all times. This assumption may not hold in practice, e.g. radio-transmitting beacons that are not in range.



(a) Two beacons **(b)** Three beacons

Fig. 24: With two beacons (orange) on the same plane, a PF measurements can be ambiguous (yellow). It is not until a third beacon is added that makes the measurements unambiguous.

3.2.3 Attacks

Although sensor fusion algorithms are meant to improve the reliability of sensor data, it does not protect against attacks, drift or faulty sensors[12]. From an attacker point of view, its goal should be to maximize the amount of uncertainty in the sensor fusion algorithm. For this to be effective, an understanding of the algorithms is required.

In [157], the authors describe three attack strategies for the KFs and its different variants. It assumes that an attacker has some means of injecting fake observations into the sensor.

MAXIMUM MAGNITUDE-BASED ATTACK The attacker tends to achieve the maximum deviation of original observation z, without passing a threshold for anomaly detection.

WAVE-BASED ATTACK When observing z_i, the reverse value of z_{i-1}, is continuously injected.

POSITIVE OR NEGATIVE DEVIATION ATTACK The attacker tends to achieve the maximum (or minimum) deviation of original observation z, along with the direction of increase (or decrease). It is comparable to drift.

12 Drift and faulty sensors as result of technical failure stay out of the scope for this work.

The above attack strategies require the attacker to know about the sensors it is attacking. For instance, it needs to know which thresholds apply without triggering anomaly detection.

Drifting

According to [23], another method of defeating sensor fusion, is by making sure sensor updates do not occur. For instance, an GPS sensor fused with IMU will eventually drift off, when the GPS signal is jammed.

3.2.4 Countermeasures

There are two types of countermeasures: correcting countermeasures and detecting countermeasures. The first type detect issues, and corrects it. It can tolerate up to a certain amount of noise. The second type does not have the ability to correct issues, but can detect issues. It is worth noting that countermeasures are domain specific, thus a detailed explanation the methods is out of the scope of this work.

According to [64], a first line of defense is a plausibility check. In their work, based on 'Marzullo Sensor Fusion'. For all sensors, confidence intervals are constructed from sensor specifications as provided by the manufacturer. Therefore, it is assumed that the majority of the sensor observations stay within their confidence interval. A disadvantage of this method is that it only works for sensors who perceive the same physical variable, not where multiple sensors complement each other.

[157] presents two countermeasures to the KF-based attacks. The first method is an enhanced unscented KF technique. The idea is that the Kalman Gain K (see Section 3.2.1) decreases when the deviation between the estimate x and the observation z increases. Due to this, a new observation will have a smaller effect on the next estimation. The second method is based on statistical Cumulative Sum (CUSUM) change detection, where an on-line non-parametric algorithm basically keeps summing the observations, and if it exceeds a certain threshold, it gives a warning. It assumes that 'average' value is more or less constant. A similar statistical anomaly detection method is introduced by [86], where an innovation variance testing detector (χ^2-based) is combined with an Euclidean distance detector.

4 ATTACKING AUTONOMOUS VEHICLE SENSORS

The previous chapter provided an evaluation of the sensors used in a typical Autonomous Vehicle (AV) [16, 46, 4, 22, 72].

The decision to pick Laser Image Detection and Ranging (Lidar) and camera sensors is threefold. First, the literature makes extensive use of the camera as a source of information, while at the same time Lidar is upcoming. Second, the use of sensors such as Radio Detection and Ranging (Radar), Tire-pressure Monitoring System (TPMS) and even Global Navigation Satellite System (GNSS) require a license for operation, because they emit radio waves or can be attacked by radio waves. Third, camera and Lidar can be used in a lab environment for controlled experiments, without being integrated in an actual vehicle.

This work uses a top-down approach. All of the experiments that will be conducted, are the result of the evaluation of the attack vectors on the sensors studied in Chapter 3. Multiple experiments are conducted to test a wider range of attacks, with the intention to provide a proof-of-concept. While this list of experiments is not exhaustive, the experiments that were conducted for this work, are selected on the criteria below:

- The experiment can be conducted in a lab environment.
- A clear output can be registered, based on an input action.
- Required hardware can be acquired with ease.

It is important to note that the devices-under-test are considered black boxes, of which the hardware layer is attacked. Even though the technical specifications and datasheets are available, the exact internal workings are not documented. No internal signals will be used and no detailed information on the hardware is assumed to be known. With respect to the attacker model, this is a valid assumption. Because of the limited money and limited time, the attacker cannot reverse engineer all systems in the world, and can only apply generally used techniques. The attacker is aware of what the hardware is supposed to do, but is not aware of how it works internally. By trying enough inputs and analyzing the outputs, it is possible to infer the internal workings [109].

In Section 4.1, the methodology and results on camera systems will be presented. Section 4.2 will do the same for Lidar. The results will also be presented in this chapter. However, the analysis and implications will be discussed in Chapter 5. In this chapter, the following experiments will be conducted:

- Testing the MobilEye C2-270 for light sensitivity (Section 4.1.2)
- Blinding the MobilEye C2-270 (Section 4.1.3)
- Confusing the MobilEye C2-270 auto controls (Section 4.1.4)
- Understanding the ibeo LUX 3 (Section 4.2.2)

- Jamming the ibeo LUX 3 (Section 4.2.3)
- Relay attack on the ibeo LUX 3 (Section 4.2.4)
- Spoofing the ibeo LUX 3 (Section 4.2.5)

A lot of hardware was used during the experiments. To make this chapter more readable, detailed descriptions including part numbers, technical specification and figures are available in Appendix D. In addition, Appendix B presents a spectrometry of all the light sources used in this chapter.

4.1 CAMERA

This section describes all the experiments conducted on the camera system. The camera system used is the MobilEye C2-270, and is borrowed from V-Tron B.V. in Deventer for testing. It is an Advanced Driver Assistance System (ADAS) that can the assist the driver in four tasks:

- Headway monitoring and warnings. Sounds the alarm in case a collision is approaching.
- Pedestrian collision warning. Sound the alarm when a pedestrian is approaching.
- Lane departure warning. Notifies driver of lane changes without signaling.
- Intelligent headlight control. Automatically dim the headlights in the dark, when oncoming traffic is detected.

This system is based on a camera only, which is installed on the windshield, under the rear view mirror (see Figure 25). It is noteworthy that this system is not sold specifically for full vehicle automation, but for function-specific vehicle automation [92][1].

Fig. 25: MobilEye C2-270 installed (on the windshield) in the autonomous car of the Dutch Automated Vehicle Initiative (DAVI). Photo courtesy of DAVI.

1 See Section 2.1 for a definition of the different degrees of automation.

4.1.1 Calibrating the hardware

The MobilEye C2-270 is a camera system that needs to be calibrated, in order to make accurate estimations of distances and that it knows where the car signals originate from. The camera would normally be installed on the windshield (facing forwards) under the rear-view mirror, but for the experiments conducted, it was not necessary to have it installed in a actual vehicle, nor having accurate distance measurements[2].

During the calibration process, several parameters can be configured. The following parameters are important for the experiments. In practice, the setup parameters would be a combination of a car and truck.

HEIGHT OF THE CAMERA The system should know at what height the camera is mounted. For this setup, a height of of 1.30 meter was specified.

DISTANCE TO FRONT BUMPER The distance between the camera and the front bumper, covering the hood. Since it is impractical in a lab scenario to 'simulate' a hood, this setting was set to minimal value of 0.1 meter.

HEIGHT OF THE HOOD Given a camera image, the system needs to know which part of the image to ignore because of the hood. Because there is no hood involved, this setting was set to zero (full image).

WIDTH OF THE CAR For lane keeping assistance, it is required to know the width of the car. For this setup, a width of 2 meters was specified.

SENSITIVITY SETTINGS The sensitivity of the alerts can be configured, e.g. at what distance the alarm should be triggered. For this experiment, the default values were used, as defined in the alerts configuration manual.

In order to finish one of the last steps of the calibration process, the external car signals had to be configured. These signals provide speed information, turn signals state and break state. For example, the MobilEye C2-270 will warn the driver when it detects a lane change without the turn signals turned on. Furthermore, some features are only available above (or below) a certain speed. At first, the Controller Area Network (CAN) bus was attempted because it used only a three wires. It turned out to be harder to use because protocol messages had to be reversed engineered from an actual car. Luckily, the MobilEye C2-270 has support for legacy signals. All signals are represented by a high-low signal, except the speed signal which is represented by a tachometer pulse signal.

Since the device is used in a lab session, the signals have been imitated by a simulator. This self-designed simulator consists of a hardware piece and a software piece. The software directly controls the inputs (CAN bus only) and outputs (CAN bus and legacy) of the simulator. For example, a complete video can be augmented with signal information (speed changes, turn signals), so an experiment can be repeated multiple times in a lab environment under the same conditions. The figures below show both components of the simulator.

As far as the documentation describes, there is no direct output of the camera image. However, the software does include a tool to retrieve the live camera image via the EyeCan box, at a lower resolution than the maximal

2 For this research, it is sufficient to blind or hide objects.

resolution. Unfortunately, this tool, shown in Figure 26, does not yield more information about objects it tracks or more.

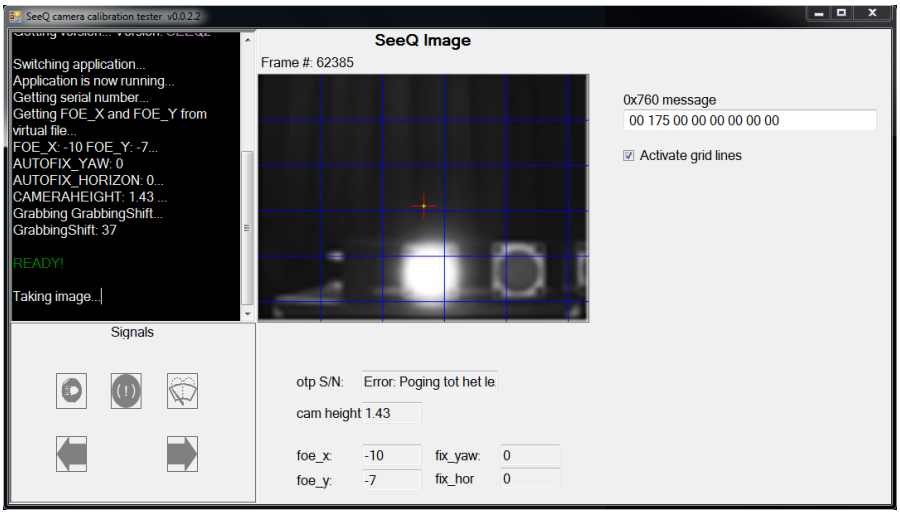


Fig. 26: MobilEye C2-270 SeeQ Camera Calibration Tester. All images from the MobilEye C2-270 camera in this report are extracted from screenshots such as the one below.

4.1.2 Testing sensitivity

The goal of this experiment, is to find out which light source the MobilEye C2-270 is most sensitive to.

Spectral sensitivity

Light consists of photons, and photons are of a certain wavelength. The human eye will respond to light between approximately 390 nm and 780 nm. Cameras have a different sensitivity range, depending on the sensor technique and lens filtering system. Figure 27 shows how sensitive Complementary Metal Oxide Semiconductor (CMOS) and Charge-Coupled Device (CCD) sensors are compared to the human eye. Furthermore, an eye (or camera) is not equally sensitive to each wavelength[3]. Camera sensors are much more sensitive to near-infrared light (approximately 800 nm to 2800 nm), if not filtered by a filter inside the lens. This is useful, because any near-infrared light available, will help to see better in the dark.

3 According to [148], the human eye is most sensitive to 430 nm (blue), 540 nm (green) and 575 nm (red). Green is the most sensitive color to the eye.

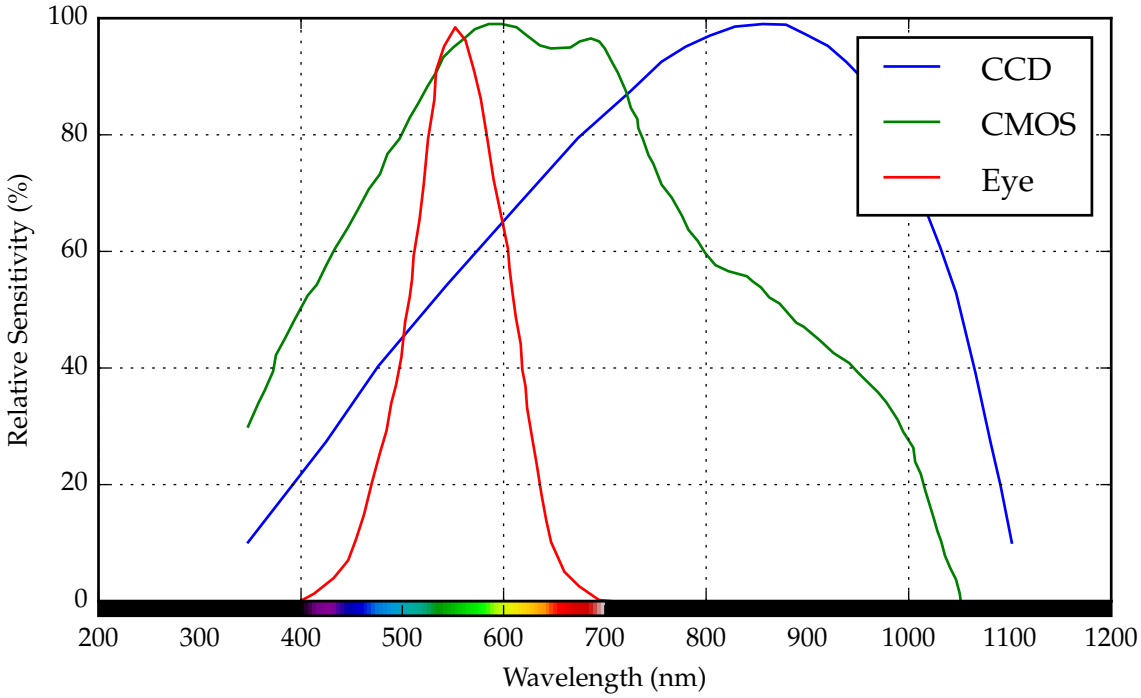


Fig. 27: Spectral sensitivity of the human eye, a CMOS sensor and a CCD sensor. The values are typical and relative. Cameras are much more sensitive to near-infrared light. Graph data taken from [113].

In this experiment, several light sources will be emitting light directly into the MobilEye C2-270 camera, at a fixed distance. The image sensor inside is a CMOS sensor. At the beginning, the off-state image will be recorded, then the on-state image. In the on-state, the light source will emit at the maximum rated power, according to the datasheet for that light source. The response of the camera will be recorded, to see which light source has most influence on the image. Because it is unknown if this camera has any filtering inside the lens system, a modified webcam without any filtering will be used as a reference.

To analyze the results, a tonal distribution for each image is created, as suggested by [21]. A tonal distribution is a histogram that shows the distribution of the color values. For this work, the grey levels are used, ranging from 0 to 255. The value lowest is black, the highest value is white. Furthermore, [21] also lists several methods of comparing images and histograms. For this work, the correlation value (Equation 23) is chosen to calculate similarity between the tonal distributions. The correlation value will only tell how similar one tonal distribution is to another tonal distribution. Therefore, it is only relevant in comparing images that have been captured in the same setup under the same circumstances. A low correlation value indicates that that two tonal distributions are less similar, therefore the two images are less similar. In the presented setup, this can only be the effect of a light source influencing the image sensor.

Tonal distribution

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H_1})(H_2(I) - \bar{H_2})}{\sqrt{\sum_I (H_1(I) - \bar{H_1})^2 \sum_I (H_2(I) - \bar{H_2})^2}} \tag{23}$$

As it turns out, there are numerous LEDs that emit visible light on the market, but the number of near-infrared emitting LEDs is small. It was hard to find the same LEDs that have an identical viewing angle, maximal current and intensity, but with another wavelength. In total five near-infrared light sources were tested for their influence on the MobilEye C2-270 camera.

1. Osram SFH4550 IR 850 nm LED

2. Osram SFH4258 IR 860 nm LED
3. Ledsee IR 875 nm LED
4. Honeywell SEP8705-3 880 nm LED
5. Ledsee IR 940 nm LED

All of these light sources have a relatively small viewing angle[4] (between 3 - 15 °). Refer to Appendix B for a spectrometry experiment, in which the different light sources have been tested for overlapping wavelengths. In addition to the selected LEDs above, the following light sources have also been selected. Two of them are lasers and one is a matrix of 5x5 940 nm LEDs (same as used above). The matrix has individual LED mounts that can be tilted horizontal and vertical. This makes it possible to combine multiple LEDs to focus the light beam on one smaller spot. This matrix was self-designed and printed with a 3D printer.

1. Ledsee 650 nm diode point laser
2. Osram SPL-PL90 905 nm diode laser
3. IR 940 nm 5x5 LED matrix

Inverse-square Law

Light is scalable: more light means more photons and more photons will induce more electrons in the image sensors (Quantum Efficiency (QE)). To increase the amount of light, more LEDs should be added. The amount of LEDs to add can be approximated with the 'Inverse-square Law', shown in Equation 24. I is the intensity and d is the distance.

$$I \propto \frac{1}{d^2} \tag{24}$$

This law assumes that the light source is point-light source (e.g. a light bulb). The opposite of a point-source is a laser, since a non-divergent laser will have the same intensity beam independent of the distance. The details are out of scope for this work, but if it is assumed that an LED falls in between. The inverse-square law can still be used to calculate an upper limit. In practice, less LEDs will be needed, depending on the viewing angle. To have the same intensity at twice the distance (denoted by d), four times more light is required, as shown in Figure 28.

4 The viewing angle is defined as the angle where half of the brightness from the center is measured.

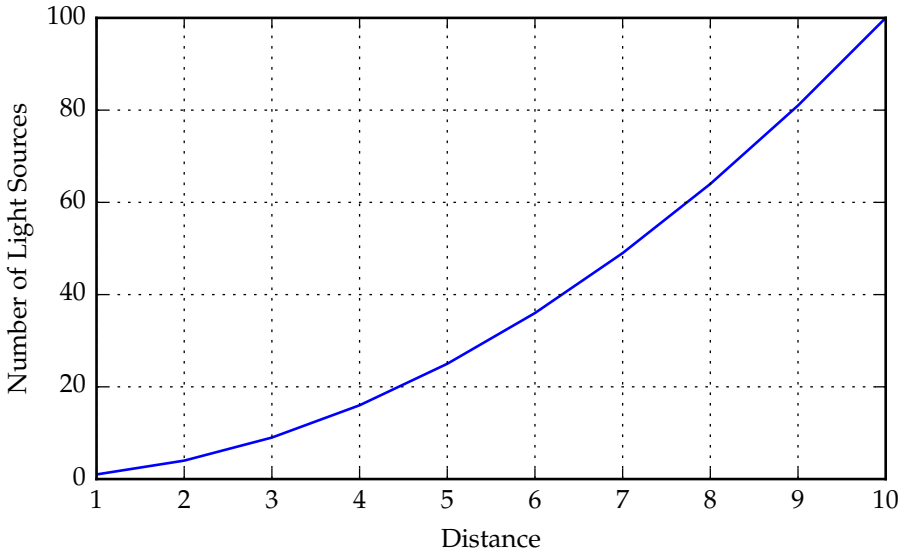


Fig. 28: Inverse-square law of light sources. As the distance increases, the number of light sources required to have the same power grows exponentially.

Based on Figure 28, an upper limit for the costs can be estimated, if the experiment would be repeated at larger distances. The prices are presented in Table 3[5]. Do note that these costs do not involve cheaper alternatives or similar products. For comparison, the costs of the light sources in Section 4.1.3 are also included.

Tab. 3: Costs of the light sources (in dollars) by extending the 50 cm experiments. Prices converted from Euros to Dollars, including VAT. Lower costs is better.

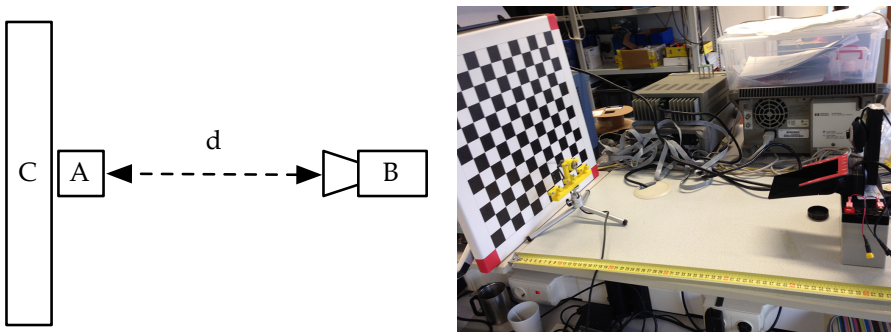
Light source	Unit Price ($)	Units	Price ($)		
			0.50 m	1.0 m	5 m
850 nm LED	0.50	4	2.00	8.00	200.00
860 nm LED	1.10	3	3.30	13.20	330.00
875 nm LED	0.22	4	0.88	3.52	88.00
880 nm LED	0.92	4	3.68	14.72	368.00
940 nm LED	0.06	4	0.24	0.96	24.00
650 nm laser	4.27	1	4.27	4.27	4.27
905 nm laser	43.25	1	43.25	43.25	43.25
940 nm 5x5 LED matrix	0.06	25	1.50	6.00	150.00
365 nm spot	26.31	1	26.31	105.24	2631.00
White spot	58.53	1	58.53	234.12	5853.00
850 nm spot	8.61	1	8.61	34.44	861.00

For this experiment, the setup as shown in Figure 29 was designed. A distance of 50 cm was chosen as a trade-off between the number of LEDs and noticeable influence on the image. The environmental light intensity in the lab was measured with the Tenma 72-6693 light intensity meter at approximately 800 lx[6]. A checkerboard pattern was chosen as a background.

5 Prices are as of writing. Where applicable, volume discount is applied. It does not include the extra costs such as shipping, related electronics etc. The exact part numbers can be found in Appendix D. Sources considered: eBay, AliExpress, Farnell, RS-Components and Mouser.

6 On a cloudy day, the outside light intensity is approximately 1000 - 2000 lx.

A checkerboard pattern produces an average tonal distribution and the MobilEye C2-270 uses that to control the auto controls, according to the datasheet [7].



(a) Schematic setup.

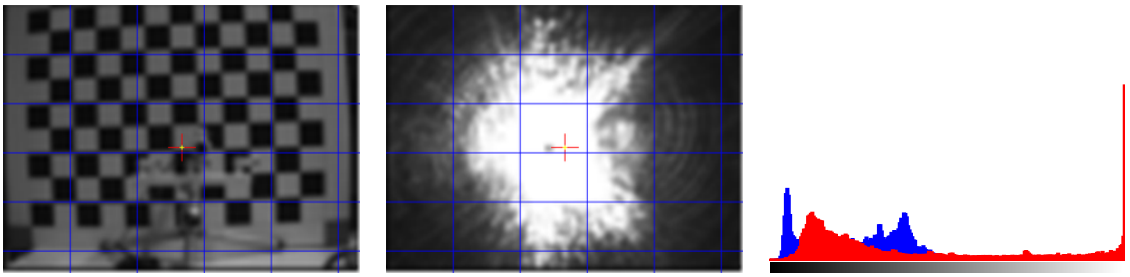
(b) Actual setup.

Fig. 29: Setup of light sensitivity setup. A is the light source, B is the MobilEye C2-270 and C is a checkerboard patterned background.

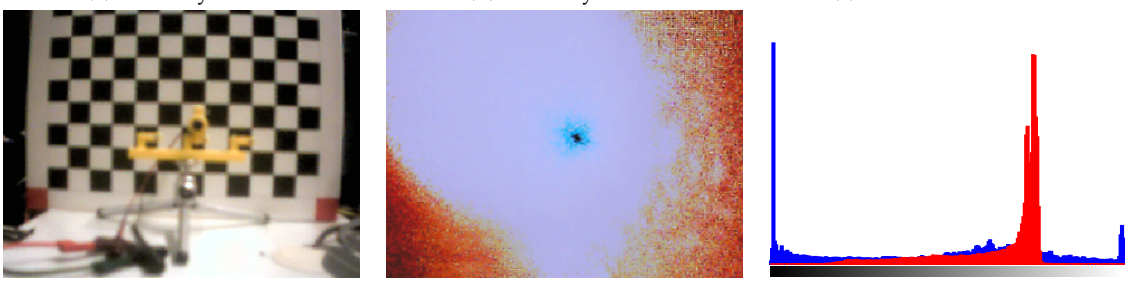
All of the resulting output images can be found in Appendix C.1, due to the high number of images. The three most-interesting images are shown below.

Every group consists of six images of a certain light source. Images (a) and (b) were captured by the MobilEye C2-270, images (d) and (e) were captured by the reference camera. Both images were taken on the same moment, after both cameras were adapted to the new light conditions. Two tonal distributions were added in images (c) and (f). Blue corresponds to the off-state, red to the on-state. The tonal distributions present the distribution of the number of pixels per grayscale value, with a total of 256 bins. All images are 320 x 240 pixels, and all tonal distributions have the same domain and range.

Figure 30, 31 and 32 are three notably results. Figure 30 shows a clear shift in tonal distribution from the visible light 650 nm laser, as denoted by the red peak. The complete background is invisible. In Figure 31 and 32, the near-infrared 850 nm and 860 nm light sources have little effect on the MobilEye C2-270, but have influence on the reference camera. Nonetheless, the MobilEye C2-270 is sensitive to near-infrared light. This conforms to the CMOS spectrum of Figure 27.

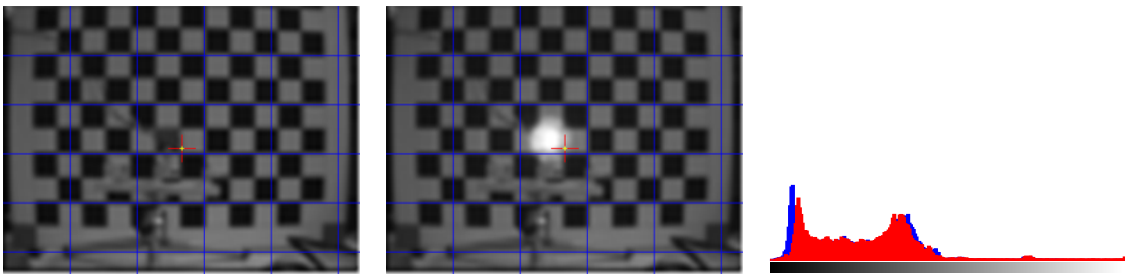


(a) MobilEye - Off **(b)** MobilEye - On **(c)** Tonal distribution.



(d) Webcam - Off **(e)** Webcam - On **(f)** Tonal distribution.

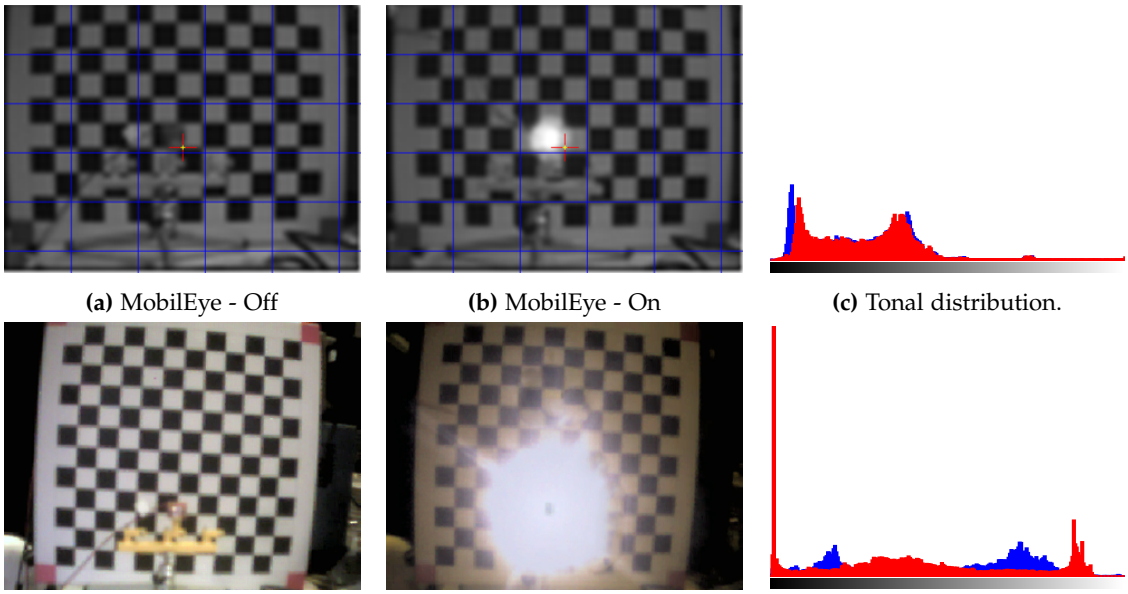
Fig. 30: 650 nm laser @ 50 cm. Blue is off-state, red is on-state.



(a) MobilEye - Off **(b)** MobilEye - On **(c)** Tonal distribution.

(d) Webcam - Off **(e)** Webcam - On **(f)** Tonal distribution.

Fig. 31: 850 nm LED @ 50 cm. Blue is off-state, red is on-state.



(a) MobilEye - Off **(b)** MobilEye - On **(c)** Tonal distribution.

(d) Webcam - Off **(e)** Webcam - On **(f)** Tonal distribution.

Fig. 32: 860 nm LED @ 50 cm. Blue is off-state, red is on-state.

The results are summarized in the Table 4 below. It presents the correlation between the off-state (blue) and on-state (red) tonal distributions per light source. A high value indicates more similarity between two images, whereas a low value indicates less similarity. Therefore, lower values are more interesting.

Tab. 4: Correlation of tonal distributions between off-on observations. Lower correlation values indicate less similarity between observations. Lower is better.

Light source	Visible Light	Distance	MobilEye C2-270	Webcam
850 nm LED	no	50 cm	**0.783**	0.898
860 nm LED	no	50 cm	**0.777**	0.855
875 nm LED	no	50 cm	0.979	0.857
880 nm LED	no	50 cm	0.994	0.824
940 nm LED	no	50 cm	0.995	0.847
650 nm laser	yes	50 cm	**0.152**	0.038
905 nm laser	no	50 cm	0.995	0.847
940 nm 5x5 LED matrix	no	50 cm	0.941	0.753

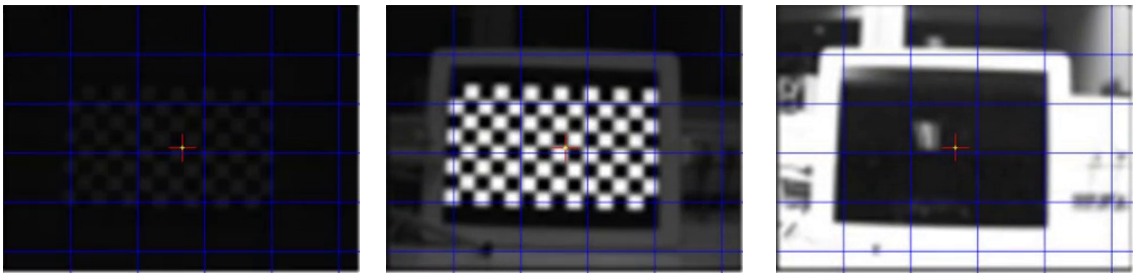
The best results have been achieved using a 650 nm laser (0.152). According to Table 3, this is the most cost-effective approach. The second-best option is the 850 nm near-infrared LED. The costs are a higher than the 940 nm 5x5 matrix, but the effect is more noticeable (0.941 versus 0.783). Although the laser is the most effective, it is also the most noticeable, especially in the dark. The near-infrared light sources are easier to aim, since they have a wider viewing angle.

4.1.3 Blinding the camera

From the previous section, it is clear what type of light source the MobilEye C2-270 is most sensitive to. In this experiment, similar to [30], the goal is to blind the camera image fully or partially, by emitting light into the camera. This type of attack does what the military weapon (the 'Dazzler') is intended to do, as mentioned in Section 3.1.3.

Blinding occurs when the camera is not able to tune the auto exposure or gain down anymore. In this case, the light cannot be dimmed, which results in an overexposed image. A common technique is to maintain an average luminosity based on a histogram, according to [36]. Figure 33 shows an example of three images from the MobilEye C2-270 with different lighting conditions. In the first subfigure, a high-intensity light source has just been turned off. In the second subfigure, it stabilized to the new light source (iPad screen) and in the last subfigure, the screen is turned off. In both the first and second image, there is no contrast with the background.

Fig. 33: The effects of auto controls. Three images from the MobilEye C2-270 after different lighting conditions. It takes at least two seconds to fully adapt to new conditions.



(a) After light source turned off. **(b)** After adapting to iPad image. **(c)** After iPad image turned off.

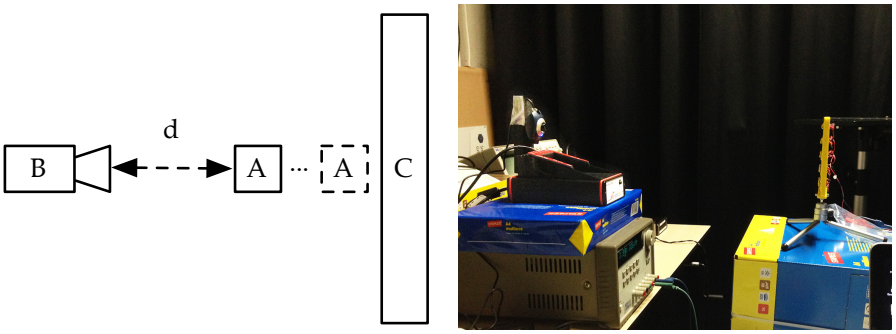
Three variables have been identified that have influence on the output of this experiment. The first variable is the environmental light. If the camera is positioned in a bright environment, the auto controls are adapted for that particular environment. In bright environments more light is required to raise above the environmental light. The second variable is the light source, and the last variable is the distance between the light source and the camera.

The results from the near-infrared sensitivity test show that 650 nm laser is the most effective. However, from the list of near-infrared light sources, the 850 nm LED is the most effective, with 940 nm LED matrix as runner-up. The set of light sources used in the previous section have a small viewing angle, limited to approximately 5 - 15 °. Although this is much wider as a laser beam, it is still hard to emit light into a camera throughout a roadside or scenery attack.

To scale up this experiment, the following light sources have been tested to blind a camera. The LED spots have more power a viewing angle of 40 degree due to the enclosure. The ultraviolet spot is added to this list, to test situations in which direct sun light is involved. The white spot is added to the test set because it covers a broader spectrum of colors (daylight) in the visible spectrum. It produces three peaks in the spectrum with wavelengths of 450 nm, 550 nm and 610 nm.

1. IR 940 nm 5x5 LED matrix
2. IR 850 nm LED spot
3. UV 365 nm LED spot
4. White LED spot

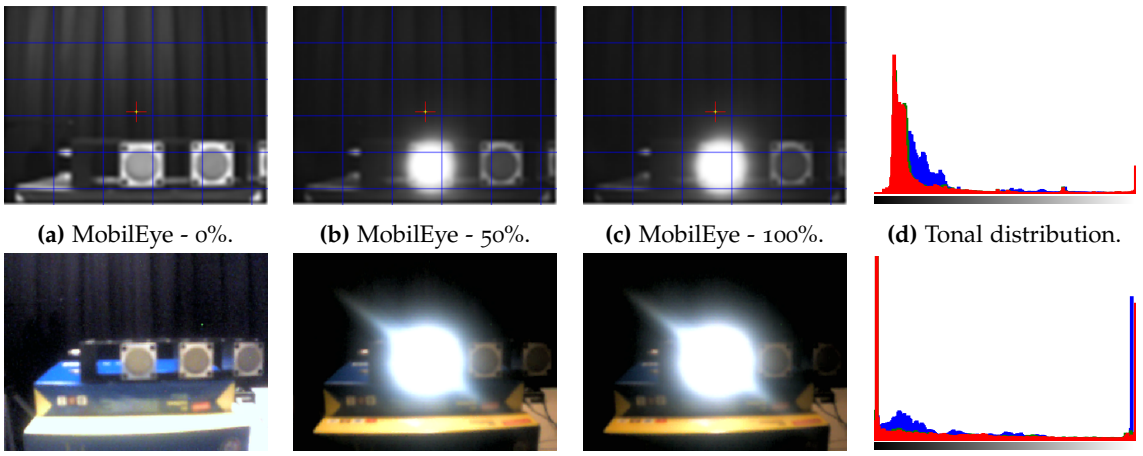
In Figure 34, the setup is presented. It looks similar to Figure 29, but in this case, four distances (50 cm, 100 cm, 150 cm and 200cm) have been tested, at several output powers (0 percent, 50 percent and 100 percent of nominal current rating[7]) in a dark (0 lx) and light (250 lx) environment. The background is a black curtain. This will make sure the MobilEye C2-270 will be most sensitive to the light source. In practice, this will be similar to a night time scenario. Similar to the previous experiment, the webcam is as a reference camera, because as shown in the previous experiment, this camera does not block near-infrared light.



(a) Schematic setup. **(b)** Actual setup.

Fig. 34: Setup of the blinding experiment. The light source A is positioned at several distances in front of camera B. C is a black background.

Three images from the result set are shown below to show the output. The tonal distributions show how many pixels have a certain tonal value. In Figure 35, it can be noted that the light source affects the amount of black tones in the image. The details in the background much are harder to see. The same effect can be seen in the webcam images of Figure 36, but in these images, the MobilEye C2-270 is almost not affected by the light source. Figure 37 shows that, in dark, the effect of the light source is almost not noticeable.

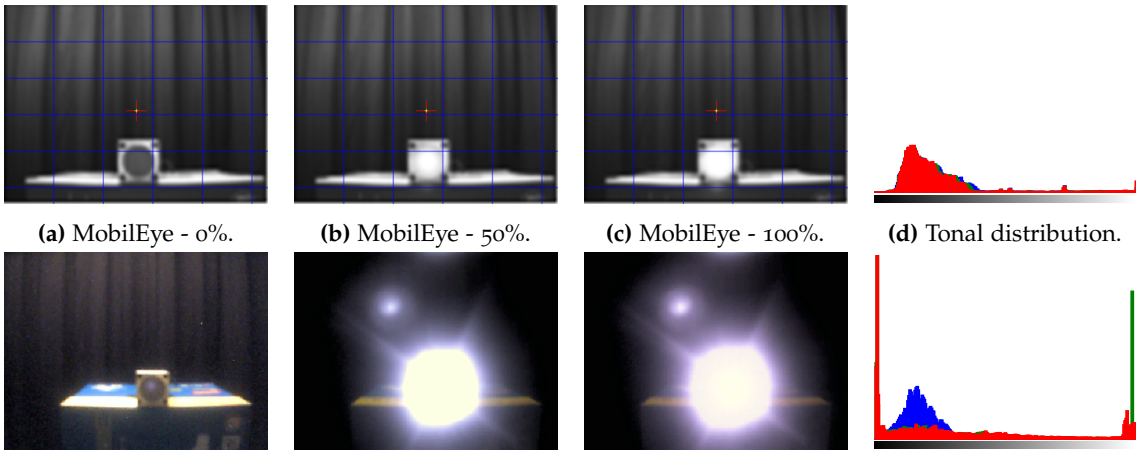


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 35: White Spot in light @ 50 cm. Blue is 0%, green is 50%, red is 100%.

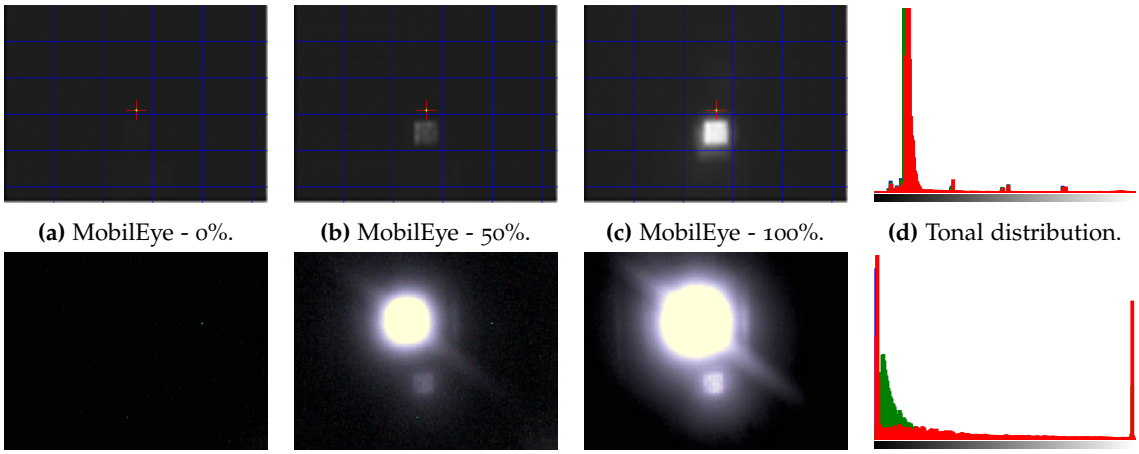
7 The light intensity of a LED is linear with the current through it.



(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 36: 850 nm Spot in light @ 50 cm. Blue is 0%, green is 50%, red is 100%.



(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 37: 940 nm 5x5 LED Matrix in dark @ 200 cm. Blue is 0%, green is 50%, red is 100%.

Analogous to the results of the previous experiments, the results are presented and interpreted the same way. The raw images can be found in Appendix C.2. The correlation values are presented in Table 5 below. The correlation values between the 0% - 50% and 0% - 100% image are presented, to see if the amount of power influences outcome. A high value indicates more similarity between two images, whereas a low value indicates less similarity. Therefore, lower values are more interesting.

Tab. 5: Correlation between 0% - 50% and 0% - 100% power observations. Lower correlation values indicate less similarity between observations. Lower is better.

				0% and 50% power		0% and 100% power	
Light source	**Visible Light**	**Setting**	**Distance**	**MobilEye C2-270**	**Webcam**	**MobilEye C2-270**	**Webcam**
365 nm LED spot	yes	dark	50 cm	0.437	0.601	**0.084**	0.208
365 nm LED spot	yes	dark	100 cm	0.860	0.955	0.524	0.816
365 nm LED spot	yes	dark	150 cm	0.993	0.987	0.858	0.946
365 nm LED spot	yes	dark	200 cm	0.691	0.992	0.758	0.995
365 nm LED spot	yes	light	50 cm	0.992	0.215	0.985	0.096
365 nm LED spot	yes	light	100 cm	0.999	0.693	0.998	0.616
365 nm LED spot	yes	light	150 cm	0.999	0.991	0.998	0.848
365 nm LED spot	yes	light	200 cm	0.998	0.994	0.996	0.986
White LED spot	yes	dark	50 cm	**0.098**	0.607	0.109	0.599
White LED spot	yes	dark	100 cm	**0.120**	0.301	0.118	0.822
White LED spot	yes	dark	150 cm	0.280	0.438	0.230	0.288
White LED spot	yes	dark	200 cm	0.748	0.803	0.323	0.522
White LED spot	yes	light	50 cm	0.492	0.119	0.400	0.113
White LED spot	yes	light	100 cm	0.901	0.251	0.777	0.165
White LED spot	yes	light	150 cm	0.946	0.520	0.941	0.357
White LED spot	yes	light	200 cm	0.924	0.677	0.927	0.513
850 nm LED spot	no	dark	50 cm	0.173	0.802	0.165	0.838

Tab. 5: Correlation between 0% - 50% and 0% - 100% power observations. Lower correlation values indicate less similarity between observations. Lower is better (continued).

				0% and 50% power		0% and 100% power	
Light source	**Visible Light**	**Setting**	**Distance**	**MobilEye C2-270**	**Webcam**	**MobilEye C2-270**	**Webcam**
850 nm LED spot	no	dark	100 cm	0.716	0.138	0.779	0.114
850 nm LED spot	no	dark	150 cm	0.966	0.589	0.796	0.848
850 nm LED spot	no	dark	200 cm	0.971	-0.044	0.911	0.819
850 nm LED spot	no	light	50 cm	0.989	0.040	0.977	0.037
850 nm LED spot	no	light	100 cm	0.996	0.071	0.997	0.039
850 nm LED spot	no	light	150 cm	0.997	0.437	0.996	0.135
850 nm LED spot	no	light	200 cm	0.996	0.565	0.997	0.352
940 nm 5x5 LED matrix	no	dark	50 cm	**0.161**	0.764	0.613	0.592
940 nm 5x5 LED matrix	no	dark	100 cm	0.727	-0.006	0.096	0.915
940 nm 5x5 LED matrix	no	dark	150 cm	0.970	0.039	**0.086**	0.927
940 nm 5x5 LED matrix	no	dark	200 cm	0.994	0.153	**0.069**	0.959
940 nm 5x5 LED matrix	no	light	50 cm	0.985	0.401	0.832	0.255
940 nm 5x5 LED matrix	no	light	100 cm	0.998	0.125	0.951	0.127
940 nm 5x5 LED matrix	no	light	150 cm	0.994	0.459	0.969	0.108
940 nm 5x5 LED matrix	no	light	200 cm	0.999	0.730	0.986	0.179

The results in the MobilEye C2-270 columns show that the correlation value increases as the distance increased. The environmental light has influence on the results, as the correlation values in light conditions are all in the range of 0.95 - 1.0, except for the white LED spot. At 100 percent power in dark, the 365 nm LED spot has the least influence, followed by the IR 850 nm 5x5 LED Matrix and White LED spot. The IR 940 nm 5x5 LED Matrix has the most influence, but this may be caused by the fact that these LEDs have a smaller viewing angle, and bundle their power. At 100 percent in light conditions, the White LED spot is the winner.

Two of the MobilEye C2-270 deviate. At 200 cm in dark, the 365 nm LED spot, a correlation of 0.691 in the 0% - 50% correlation, it performs better than 0.993 at 150 cm. The other value is the 940 nm 5x5 LED Matrix at 50 cm in dark. The 0% - 100% correlation value is 0.613, while at a greater distance (thus less light), the correlation value is 0.096. There is no explanation for the first outlier. The images and histogram do not show any anomalies. One explanation for the second outlier is that at a distance of 50 cm, the auto controls have not been fully adjusted. The individual LEDs are clearly visible. At 100 cm, the light is more concentrated, from the camera point of view. The webcam results are quite in line in the visible spectrum, but there are some outliers. It is believed that these outliers have also been caused by sensor faults. For instance, the tonal distribution of 850 nm spot in dark @ 50 cm shows that at 50%, the image has more white tones compared to the 100% image. For some reason, emitting more light into the image sensor caused the webcam to observe more grey values.

In dark situations, the 940 nm 5x5 LED matrix is the most effective. As Table 3 show, the 940 nm LEDs are by far the cheapest of all the light sources used in this experiment. Although this experiment did not succeed to fully blind the image using near-infrared light sources, these light sources can be used to blind objects. For instance, by mounting several LEDs on a vehicle that should normally be recognized, the MobilEye C2-270 cannot recognize them anymore. This fits the scenery modification scenario. In general, blinding a camera will work best from a front/rear/side attack, since the light sources should be positioned carefully to emit the most light into the image sensor.

4.1.4 Confusing the auto controls

According to the datasheet [7] of the MobilEye C2-270 camera sensor, it is equipped with auto exposure control and auto gain control. It is undocumented if both auto controls are enabled, but for optimal image quality in darker environments, it is presumed to be. Auto exposure control will determine the shutter speed for each frame, while auto gain control can amplify the electron charges from the image sensors after exposing it to light[8]. According to the datasheet, both controls measure the current scene luminosity and desired output luminosity by accumulating a histogram of pixel values. This value is then used to calculate the desired exposure and gain value. Both controls need some time before they are stable, because it is an iterative control process. For the images in Figure 33, it took up to two seconds. On the other hand, having a too fast of a loop control makes the image very unstable in terms of brightness.

Unfortunately, this attack could not be attempted on the MobilEye C2-270, because there was no way of accessing the full video stream at a decent

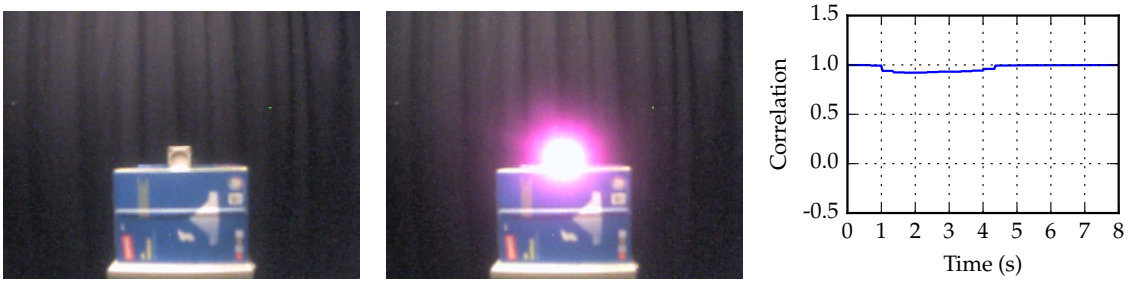
8 Refer to Section 3.1.3 for more information.

frame rate. The only video stream that was available, had a rate of less than approximately five frames per second. This is too low to see what happens in intermediate frames. It is believed that this attack applies to the MobilEye C2-270, as some influence was observed in a similar setting while capturing Figure 33.

To show the potential of this attack, the webcam was used instead, with the only limitation that it does not have an auto exposure method. This webcam can output at 30 frames per second at 320 x 240 pixels, but it turned out that frames were dropped[9] while the camera was auto controlling the gain. Fortunately, this demonstrates the potential of such an attack: it temporary 'blinds' the output. For this experiment, the setup was the same as the previous setup, explained in Figure 34. Compared to the previous experiment, this experiment does not focus on 'hiding' an object from the camera by blinding it, but on influencing the auto controls in the period before the image recovers and stabilizes. The longer it takes to stabilize to the new environmental conditions, the longer the car is vulnerable to objects it cannot detect. This attack distinguishes itself from situations like driving out of a tunnel, because in that case, the camera can more gradually adapts to the new conditions.

All of the light sources of the previous experiment were re-used, with the addition of the 650 nm laser (only at 50 cm due to safety regulations). For each take, a video was recorded. Each take starts in a 'starting condition'. Then, the light source is turned on to full power in one shot, and the video is stopped when the camera has adapted to the new light source, for as far as possible. It is assumed, that when the light source is turned on, the camera needs some time to adjust to the new lighting conditions, and recover the image. In this period, the camera is vulnerable because it cannot perceive any relevant information. To analyze each video, a tonal distribution is created for each frame, after which each consecutive tonal distribution is correlated with the first one (the 'starting condition'). The time between the first drop in correlation, and the first rise (if applicable) is measured, and denotes the vulnerable period.

Figure 38, 39 and 40 are three examples that are interesting and useful to explain the different results. Figure 38 is an example where the light does not have any influence on the image. Therefore, there is no clear vulnerable period to measure. Figure 39 has two levels[10]. It is not until approximately four seconds before the camera starts to adjust, but it does not recover. Figure 40 shows two clear levels, after which it starts recovering the image.

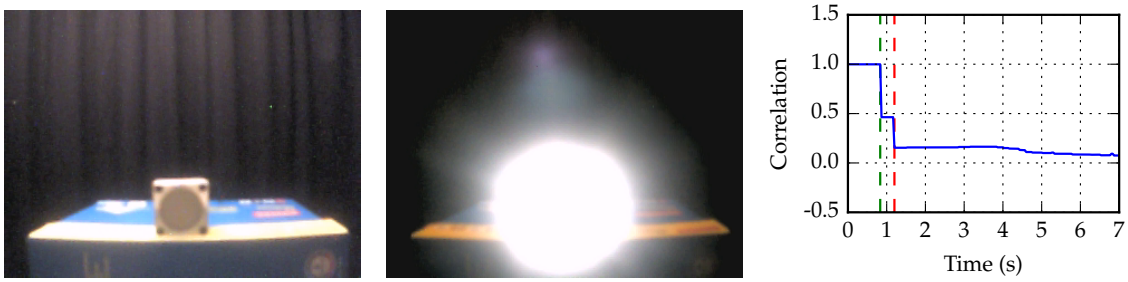


(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 38: 365 nm spot in light @ 100 cm. Green line is time of start, red line is time of stop.

9 This means that the camera was unable to send new images.

10 The frame rate drops, possibly because the sensor cannot adjust the auto controls and keep up 30 FPS. Therefore, the same image is outputted for several frames.



(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 39: White spot in light @ 50 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 40: 940 nm 5x5 LED matrix in dark @ 100 cm. Green line is time of start, red line is time of stop.

The results are presented in Table 6. The raw images can be found in Appendix C.3. The of the vulnerable period is measured in seconds. The correlation score is the lowest score calculated over all frames. A high value indicates more similarity between two images, whereas a low value indicates less similarity. Therefore, lower values are more interesting, and indicate how much the image was blinded.

Tab. 6: Blinding times (in seconds) and the lowest correlation values. Longer times indicate longer blindness, lower correlation values indicate less similarity. Higher times is better, lower correlation values is better.

Light source	Visible Light	Setting	Distance	Blinding Time (s)	Minimal Correlation
365 nm LED spot	yes	dark	50 cm	0.67	0.201
365 nm LED spot	yes	dark	100 cm	0.63	0.706
365 nm LED spot	yes	dark	150 cm	—	0.969
365 nm LED spot	yes	dark	200 cm	—	0.981
365 nm LED spot	yes	light	50 cm	0.97	0.504
365 nm LED spot	yes	light	100 cm	—	0.921
365 nm LED spot	yes	light	150 cm	—	0.945
365 nm LED spot	yes	light	200 cm	—	0.939
White LED spot	yes	dark	50 cm	1.67	0.116
White LED spot	yes	dark	100 cm	1.33	0.409
White LED spot	yes	dark	150 cm	0.43	0.470
White LED spot	yes	dark	200 cm	0.77	0.551
White LED spot	yes	light	50 cm	0.37	0.076
White LED spot	yes	light	100 cm	0.40	0.079
White LED spot	yes	light	150 cm	0.73	0.367
White LED spot	yes	light	200 cm	0.37	0.474
650 nm laser	yes	dark	50 cm	∞	-0.100
650 nm laser	yes	light	100 cm	∞	-0.011

Tab. 6: Blinding times (in seconds) and the lowest correlation values. Longer times indicate longer blindness, lower correlation values indicate less similarity. Higher times is better, lower correlation values is better (continued).

Light source	Visible Light	Setting	Distance	Blinding Time (s)	Minimal Correlation
850 nm LED spot	no	dark	50 cm	4.67	-0.017
850 nm LED spot	no	dark	100 cm	2.97	-0.001
850 nm LED spot	no	dark	150 cm	—	-0.035
850 nm LED spot	no	dark	200 cm	4.30	-0.064
850 nm LED spot	no	light	50 cm	5.50	-0.033
850 nm LED spot	no	light	100 cm	1.67	-0.021
850 nm LED spot	no	light	150 cm	2.67	0.0267
850 nm LED spot	no	light	200 cm	5.00	0.1229
940 nm 5x5 LED matrix	no	dark	50 cm	5.30	-0.012
940 nm 5x5 LED matrix	no	dark	100 cm	5.47	-0.014
940 nm 5x5 LED matrix	no	dark	150 cm	1.67	-0.017
940 nm 5x5 LED matrix	no	dark	200 cm	4.67	-0.017
940 nm 5x5 LED matrix	no	light	50 cm	**6.00**	-0.016
940 nm 5x5 LED matrix	no	light	100 cm	3.17	-0.041
940 nm 5x5 LED matrix	no	light	150 cm	4.33	-0.022
940 nm 5x5 LED matrix	no	light	200 cm	1.33	-0.027

In line with the results of Section 4.1.3, the environmental light has influence on the results. The 365 nm LED spot has the least influence, as it cannot blind the camera nor let it drop frames. In this experiment, the 650 nm laser was also tested. The results show that the camera does not recover from the intense beam, therefore the blinding time is infinity.

The 650 nm laser has the most influence, but is much harder to aim a beam at a moving image sensor. The beam is only approximately 1.5 mm wide. Furthermore, it is a visible light source. The 940 nm 5x5 LED matrix has the most influence, for the same reason as the previous section. The light can be much better aimed at the camera than the laser beam can.

4.2 LIDAR

This section is dedicated to the experiments with the ibeo LUX 3 Lidar. The required hardware is borrowed from Ibeo Automotive Systems GmbH in Hamburg. It is a four-layer laser-based ranging system, mounted on a rotating head to provide view up to 110 °. The maximal range is up to 200 meters, depending on the weather conditions. The four layers refer to the number of scanning rays. Each layer is slightly tilted with respect to the road, so the Lidar can operate on uneven roads (e.g. ones with bumps, hills, etc.). Even though it is a multi-layer Lidar, it cannot provide a three-dimensional view, but only four layers of two-dimensional planes.

The ibeo LUX 3 contains an embedded object tracking system that can track the objects listed below [11].

- Car
- Truck
- Bike
- Pedestrian
- Unknown small
- Unknown big
- Not classified

The maximum number of objects that can be tracked is 65. It uses a Kalman Filter (KF) for tracking objects. Each object, when detected, will be augmented with an object identification number for tracking purposes and lifetime information.

4.2.1 Interfacing the hardware

The ibeo LUX 3 consists of two devices. The first device is the Lidar itself. The other device is the connection box, that functions as a gateway between the Lidar and computer. This device would normally be mounted in (or on) the bumper of a vehicle, but in the lab it was mounted on a drivable table. Figure 41 shows this setup.

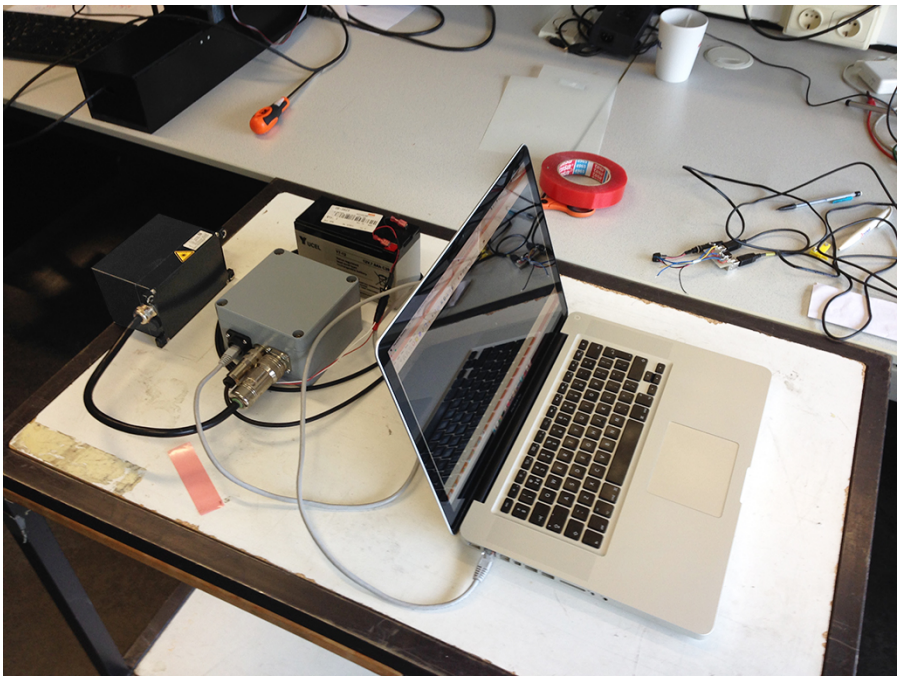


Fig. 41: Typical test setup of the ibeo LUX 3. The Lidar is the box on the table on the left. It is connected via the grey control box to the notebook. The setup is battery powered.

To interface the Lidar with the computer, the accompanying software called 'Ibeo Laser Viewer Premium' was used. It can be connected via Ethernet and provides a real-time representation of what the Lidar registers. Each measurement is drawn on circular grid as a point with a color. The software can also record the data for later use. Figure 42 shows a screenshot of the main window of the software.

The device parameters can also be changed from within the software. The default settings were used, which include a constant angular resolution. For the display settings, only valid pulses are shown. A valid point is one which passes the pre-processing stage that normally filters out dirt, ground and clutter.

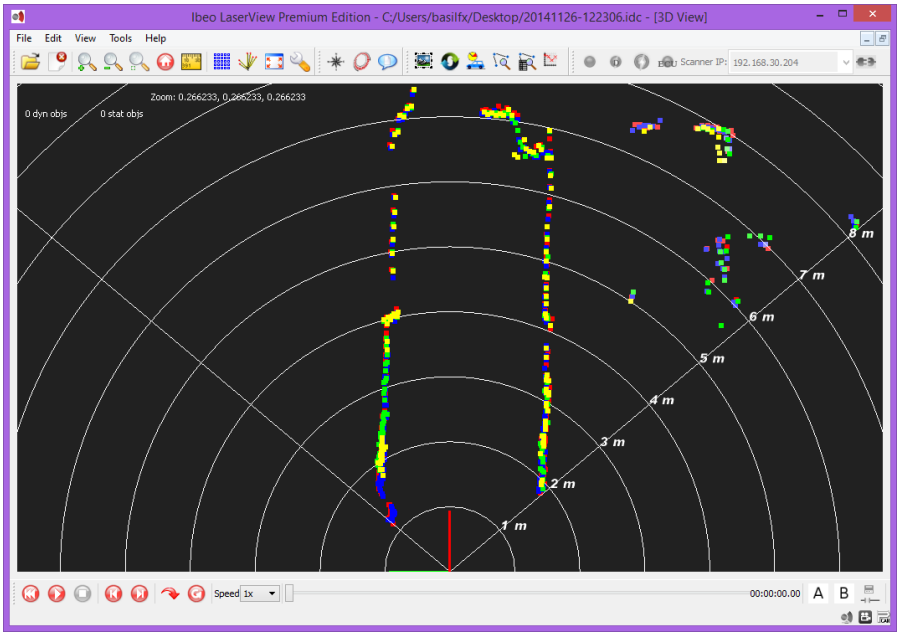


Fig. 42: Screenshot of the Ibeo Laser View Premium software. The dots represent the laser reflections. In this case a scan of a hallway was made.

Each of the four layers is represented by a color (red, green, blue and yellow). Furthermore, it supports up to three echoes. For example: a laser beam hits a window. Part of the light is reflected and triggers a measurement. This is the first echo. Most of the light will travel through the window, and reflect off a rain drop. This triggers a second echo. The last bit of light will be reflected by the actual object, triggering the third echo. The layer to color mapping is shown in Table 7.

Tab. 7: Layer to color mapping, as used by the ILV Premium software.

Layer	Color	Echo 1	Echo 2	Echo 3
4 (Highest)	Yellow			
3	Green			
2	Blue			
1 (Lowest)	Red			

4.2.2 Understanding the Lidar

A brief description of how Lidar works, has been presented in Section 3.1.1. In short, Lidar emits a near-infrared light pulse and measures the time of flight to the calculate distance to an object. This experiment should reverse engineer how the ibeo LUX 3 Lidar system works. In particular, the next four questions:

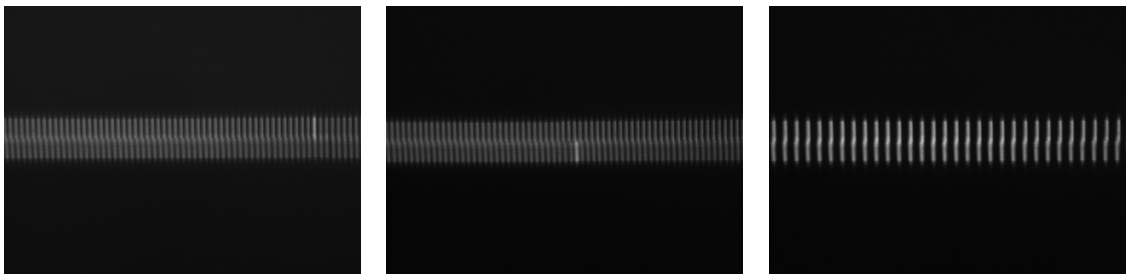
- Can the ibeo LUX 3 laser beams be visualized?
- How accurate is the ibeo LUX 3?
- How does it respond to surfaces like glass and a mirror?
- Does the ibeo LUX 3 implement any countermeasures against jamming/spoofing?

The last question addresses the possibility that some sort of countermeasure is implemented against jamming or spoofing attacks. As discussed in 3.1.1, it assumed that the Lidar rotates and emits a pulse each scan step. It is unclear if this pulse pattern is constant, can be identified and if it carries information.

The first sub-experiment conducted, answered the first question if Lidar could be visualized. From the datasheet [10], it was known that the ibeo LUX 3 uses eye-safe 905 nm infrared light for their light pulses. As it turns out, this wavelength is a de facto standard, as many other Lidar-based products use the same wavelength [78, 85], including police speed measurement devices [95]. The spectrometry experiment in Appendix B has verified that the emitted light by the Lidar is of the specified wavelength.

From the experiments in Section 4.1.2, it is known that the emitted wavelength can be captured with camera that is sensitive to infrared light. So, in a setup similar to the light sensitivity experiments (see Figure 29), the Lidar

output was visualized for different scan frequencies, using a Basler acA2040-25gmNIR high-speed near-infrared camera. The results are presented in Figure 43.



(a) 12.5 Hz **(b)** 25 Hz **(c)** 50 Hz

Fig. 43: Lidar pattern visualized at a scan frequency of 12.5 Hz, 25 Hz and 50 Hz. All images are of the same size and same distance.

The figure shows what the Lidar pattern looks like. In Figure 43c, the distance between two lines increased. According to the operating manual [12], the distance increases as the angular resolution drops at a higher scan frequency.

Question number two can be answered from the above figure as well. According to the manual, the angular resolution of Lidar depends on the scan speed. The minimal constant angular resolution is 0.25 ° at 12.5 Hz or 25 Hz, and 0.5 ° at 50 Hz. The problem is that the Lidar pulse pattern is not continuous. Instead, it probes at specific positions. This is demonstrated in Figure 44, where each line is one laser beam. If object A and B are of the same size, B will not be noticed because no beam hits it. In practice, each laser beam diverges a bit: a beam is narrow in the beginning and wide at the end.

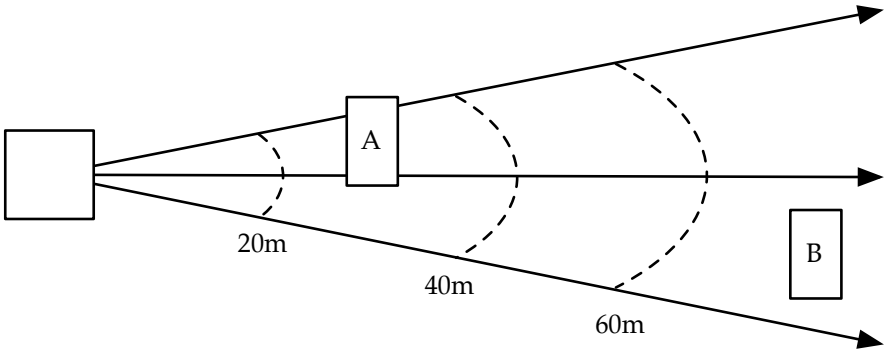


Fig. 44: Resolution of Lidar. Object A and B are of same physical size. In practice, the beams would not go through A, but they are shown for demonstration purpose.

To verify the angular resolution, the number of lines in Figure 43c are counted. In total, 42 lines were projected on a surface that is 59.4 cm wide. This image is taken at a distance of 50.0 cm of the Lidar. According to the manual, the angular resolution is measured between the lines, thus the width of the gap. The width of one line, including the gap is 1.41 cm. The width of the gap is susceptible to interpretation, since the line fades out. By changing the brightness to lowest and maximal setting (in Adobe Photoshop), it was found that the gap is between 0.73 cm and 0.96 cm wide. Figure 45 represents this.

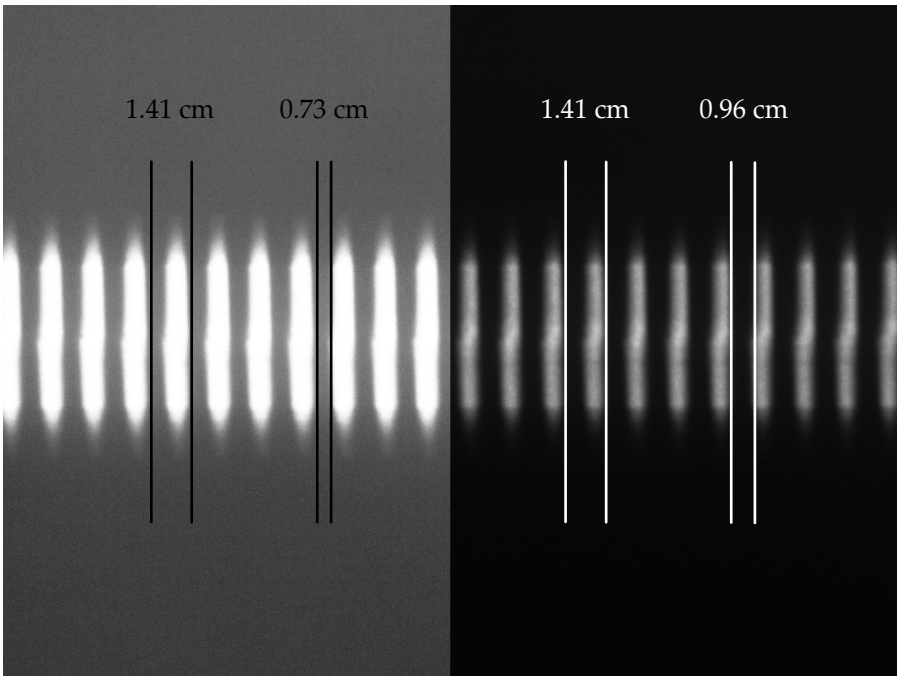


Fig. 45: Measuring the angular resolution at a distance of 50 cm. This image is a magnification of Figure 43c. Half of the image has maximal brightness, the other half has minimal brightness.

Note how the lines are not perfectly straight in Figure 45. The bottom half corresponds to the two lower layers and the top half to the two top layers. With this information, the angular resolution can be calculated via Equation 25.

$$\begin{aligned} \alpha &= \tan^{-1}\left(\frac{\text{Opposite}}{\text{Adjacent}}\right) \\ \alpha_{worst} &= \tan^{-1}\left(\frac{0.96\ \text{cm}}{50.0\ \text{cm}}\right) = 1.10\ ^\circ \\ \alpha_{best} &= \tan^{-1}\left(\frac{0.73\ \text{cm}}{50.0\ \text{cm}}\right) = 0.84\ ^\circ \end{aligned} \tag{25}$$

In the best case, the angular resolution is almost 1.7x larger than the specified 0.5 °. To achieve this angular resolution, the width of the gap should be 0.43 cm[11].

It is now possible to calculate how big an object should be at a specific distance, to not get noticed with certainty[12]. Equation 26 shows this. In other words, from a Lidar point of view, the object should be within the gap. For instance, at 20 meter the object must be smaller than 0.29 meter. This number exceeds the calculated 0.088 meter in the operating manual (same distance). At a distance of 100 meter, the width of the gap is 1.47 meter.

$$\text{Opposite} = \tan(\alpha_{best}) \cdot \text{Adjacent} = \tan(0.84) \cdot 20\ \text{m} = 0.29\ \text{m} \tag{26}$$

Lidar pulses are highly absorbed by rain or snow. If only 10 percent of the light reflects off a surface, the range drops to only 50 meter[13], according to the specifications. Other surfaces can also attenuate the amount of light that reflects, according to [122]. The ibeo LUX 3 can detect up to three echoes. An echo occurs when the original pulse is received more than once.

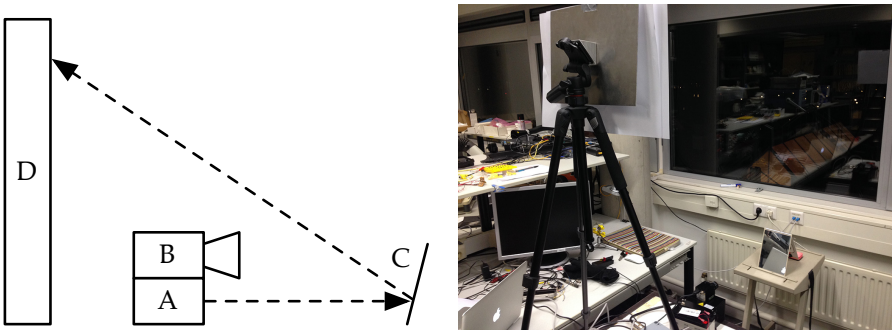
11 Page 3-13 of the operating manual shows two different angular resolutions, where 0.125 ° is the minimal resolution for 12.5 Hz. Therefore, these calculations assume 0.5 ° is the minimal resolution for 50 Hz, as mentioned on page 3-9.

12 Or the opposite, to get noted. This will be explained in the spoofing attack.

13 In a high-speed scenario, a Lidar provides only a bit more than 'one second of vision'.

In the ILV software, it is shown by a less bright color. It is interesting to see what happens if a light pulse reflects off a mirror or a glass, and if these observations are still processed in the tracking algorithm.

The second sub-experiment is represented by Figure 46. As can be seen, the object-to-detect is positioned behind the Lidar itself.



(a) Schematic setup.

(b) Actual setup.

Fig. 46: Setup of the Lidar mirror experiment. The ibeo LUX 3 is represented by A, B is an IR-sensitive camera, C is a mirror and D is a flat surface.

In Figure 47, the image captured by the near-infrared camera is shown. The Lidar representation is shown in Figure 47. The Lidar observes a surface in the front, while it is actually positioned in the back. The reflection shows up in the representation at approximately twice the distance: the distance from the Lidar to the mirror plus the distance from the mirror to the surface. The flat surface is detected as the second echo. The embedded tracking software does not detect the mirror.

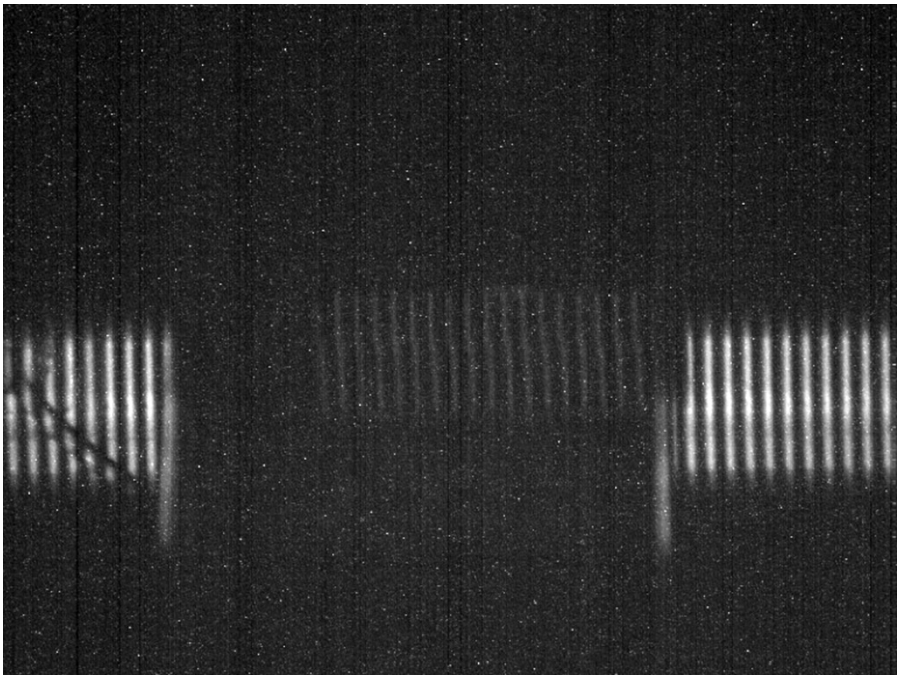


Fig. 47: Result of Lidar mirror experiment as captured by the near-infrared camera. The mirror is positioned in the center. The lines on the flat surface are visible in the mirror, but with less intensity.

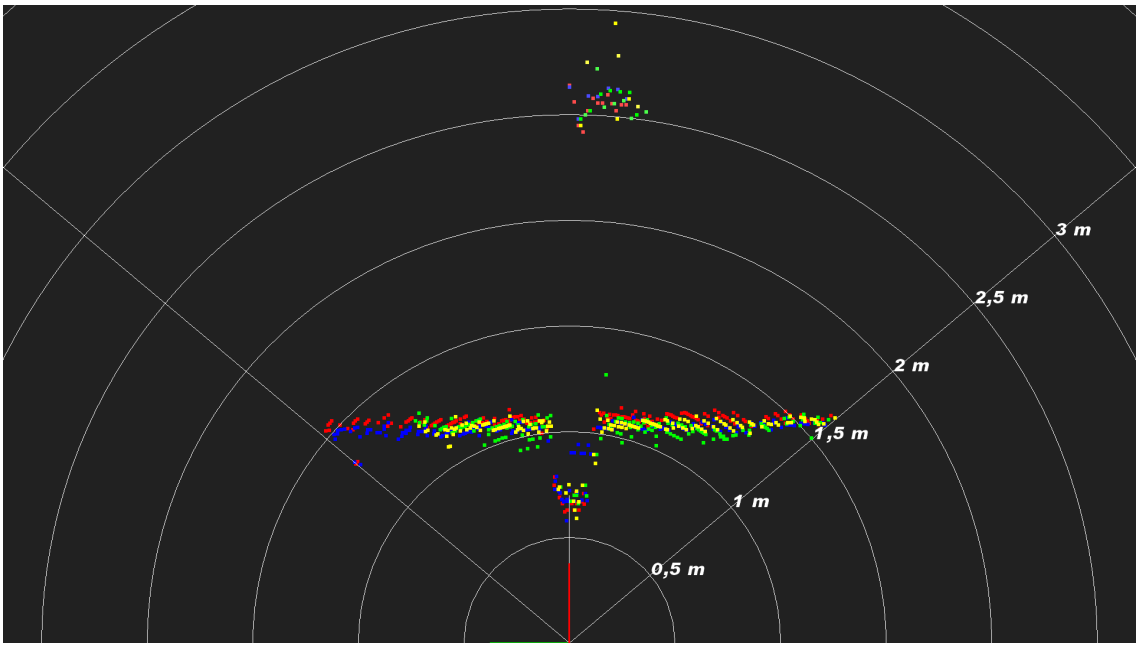
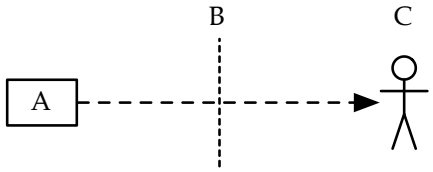
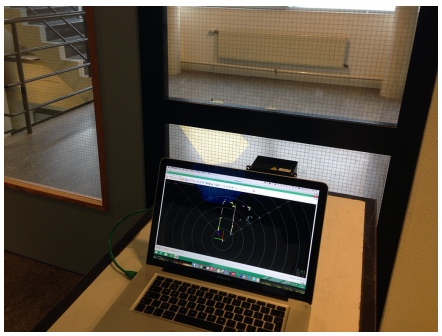


Fig. 48: Result of Lidar mirror experiment. The mirror is detected in the second echo at approximately 3 meters.

In the third sub-experiment, the Lidar is positioned directly in front of a glass window. The goal is to see how the Lidar reacts to this situation, specifically if the tracking algorithm ignores echoes or not. This is important, as an echo could also be the result of reflection off a rain drop. As the Lidar records, a pedestrian walks on the other side of the glass. A setup is depicted in Figure 49. The result is observed.



(a) Schematic setup.



(b) Actual setup.

Fig. 49: Setup of Lidar glass experiment. The ibeo LUX 3 is represented by A, the glass window by B and the pedestrian by C.

The Lidar representation is shown in Figure 50. In this figure, the pedestrians are detected and marked by 'Ped': once when walking up and once when walking down. The glass window is not shown in the representation because the distance to the glass falls within the minimal operating range of the Lidar (0.30 m). However, the reflections are of the second echo. This result shows that, if used in practice, it will also detect object of the second echo, through glass other transparent objects such as rain.

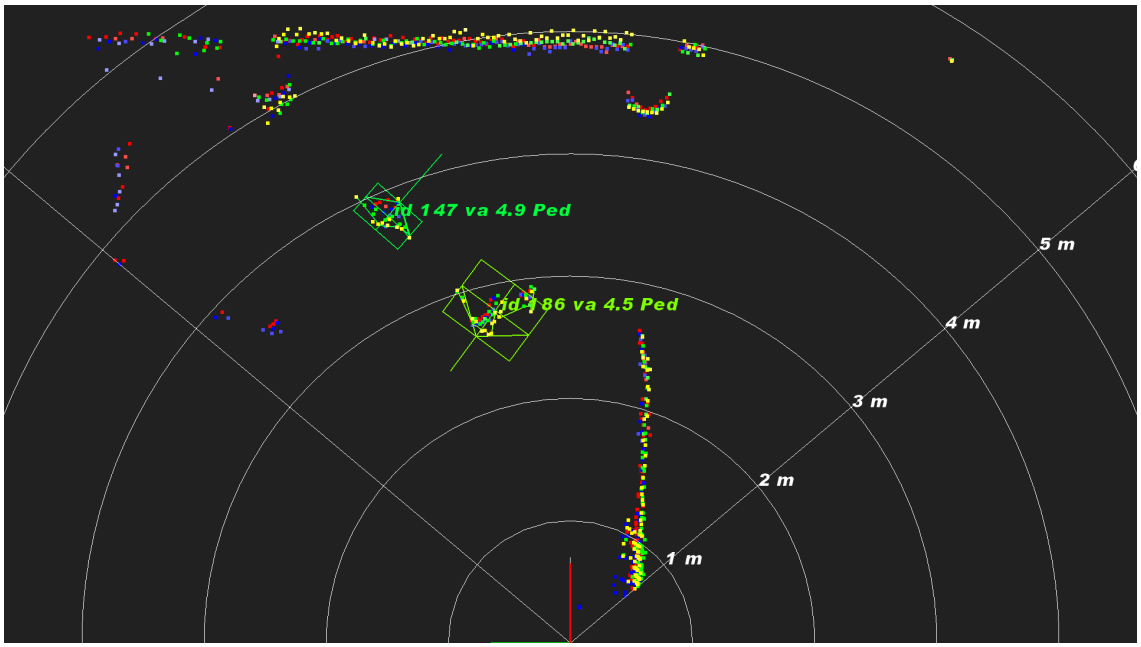


Fig. 50: Result of the Lidar glass experiment. The pedestrians are detected through the glass and marked as 'Ped'.

The sub-experiments have also led to an answer on the last question, if the ibeo LUX 3 implements any mechanism to prevent or detect jamming or spoofing attacks. From the way the pulse signals are visualized with the high-speed camera, it does not indicate any protection mechanism is incorporated. The high-speed videos recorded for this experiment show a constant on-off pattern on the exact same position and with the same intensity, for each scan. There does not seem to be any channel to carry information:

- Varying the intensity is not possible, since that would influence the range.
- Changing the width between pulses would create horizontal gaps
- Making the pulse lines longer would create vertical gaps.

This section has shown how to visualize the Lidar signals. It was calculated if the resolution is as specified and tested how different materials (glass and mirror) influence the representation. The next section will use these results to jam and spoof signals.

4.2.3 Jamming the signal

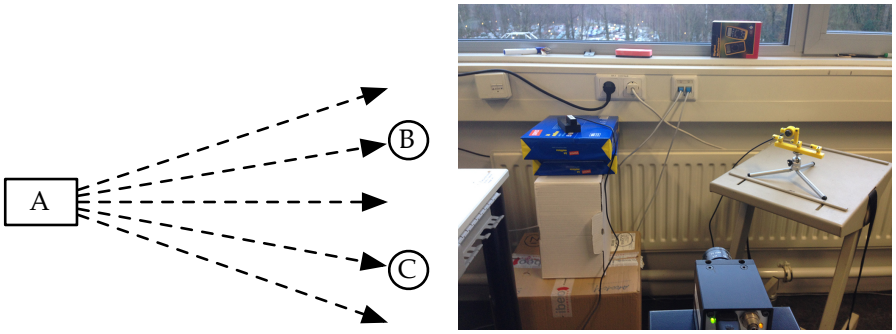
In this experiment, the goal is to jam the signal. For this experiment to succeed, it is necessary to emit similar light pulses of the same wavelength and timing as the ibeo LUX 3 Lidar does.

Transceivers

The previous experiment has shown that the ibeo LUX 3 uses light with a wavelength of 905 nm. This means that if a photodetector is used that is sensitive to this wavelength, it is possible to receive the signal. An anonymous company sponsored three transceivers. The transceivers contain a voltage amplification circuit to boost the photodetector output. The emitting laser is an Osram SPL-PL90, which costs 43.25 dollar (see Table 3). According to the datasheet [104], applications for this laser diode include range finding, with a range up to 100 meter. The receiving photodetector is an Osram SFH-213, which costs 0.65 dollar. The spectrometry experiment in Appendix B

verified that the laser emits light of the right wavelength. The output of the transceiver is a voltage signal that corresponds to the intensity of the pulse. An oscilloscope can be directly attached to visualize the signal. To emit a pulse, a high voltage should be connected to the input wire of the transceiver.

To test if the Lidar pulses could be visualized by the oscilloscope, the setup in Figure 51 was used. The distance between the Lidar and the wall is approximately 1 meter. The oscilloscope is not required to mount the jamming attack. It is only used to identify the signal characteristics[14].



(a) Schematic setup.

(b) Actual setup.

Fig. 51: Setup of Lidar patterns visualization. The Lidar is represented by A, the two transceiver devices by B an C.

In figure 52, the oscilloscope output is visualized. The pink channel corresponds to transceiver B. At 5 ms/div[15], the pink signal repeats each 20 ms, which is equal to 50 Hz. This matches the configured scan frequency. As explained in Section 4.2.2, the width of one pulse is small. At a distance of 1 meter, multiple pulses (for different scan steps) hit the sensors. Therefore, the width of the pulse in 52 does not represent one pulse, but includes multiple other pulses. If this attack would have been conducted on a large distance, this effect is less noticeable. For instance, at 20 m and 50 Hz, the gap between each pulse was calculated to be 0.29 m wide.

14 An attacker could also have found the characteristics online. It is a matter of time before a database exists with all the relevant specification.

15 Milliseconds per division. This corresponds to the width of one cell in the grid, measured in time.

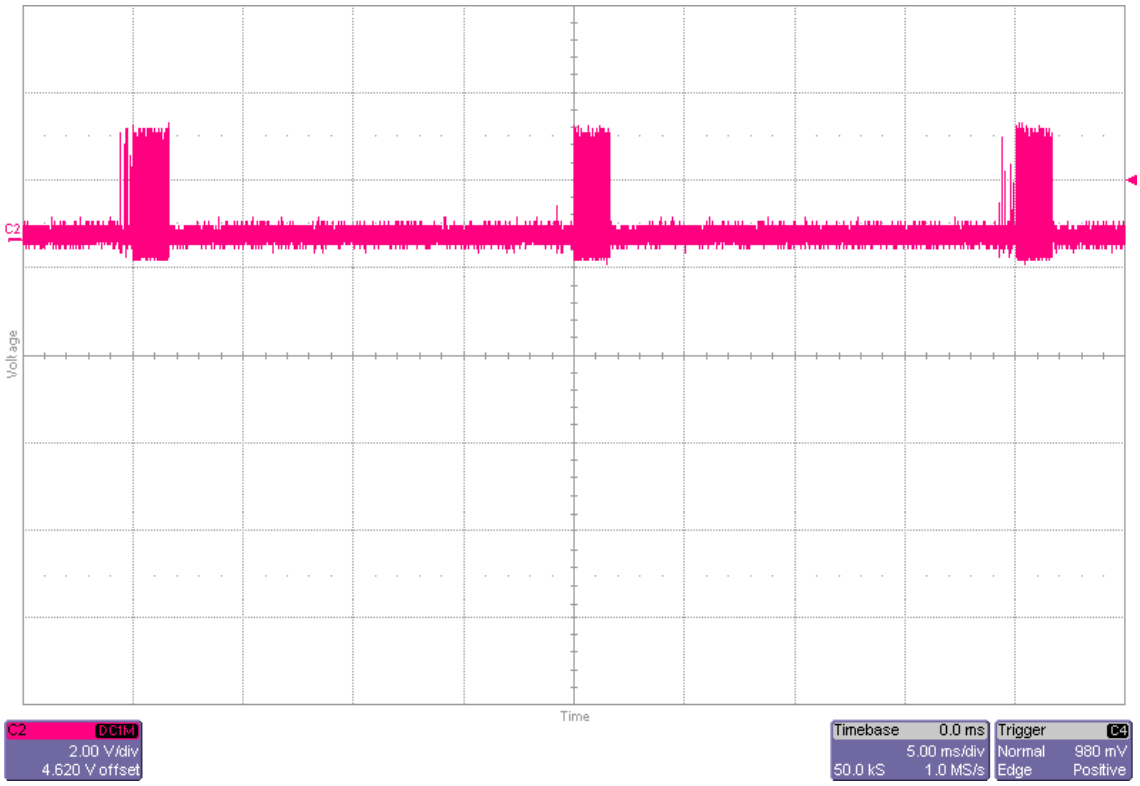


Fig. 52: Visualization of three Lidar pulses. The width of the pulse plus the gap equals the scan frequency.

After connecting transceiver C to the oscilloscope, the rotation of the Lidar could be visualized in Figure 53. The burst on the yellow channel happens approximately 5.5 ms later (note that the scale has changed to 1 ms/div). The transceiver is still sensitive to scattered light, as a small amount of light is still detected on the pink channel after the burst finished. This can be related to reflections in the lab.

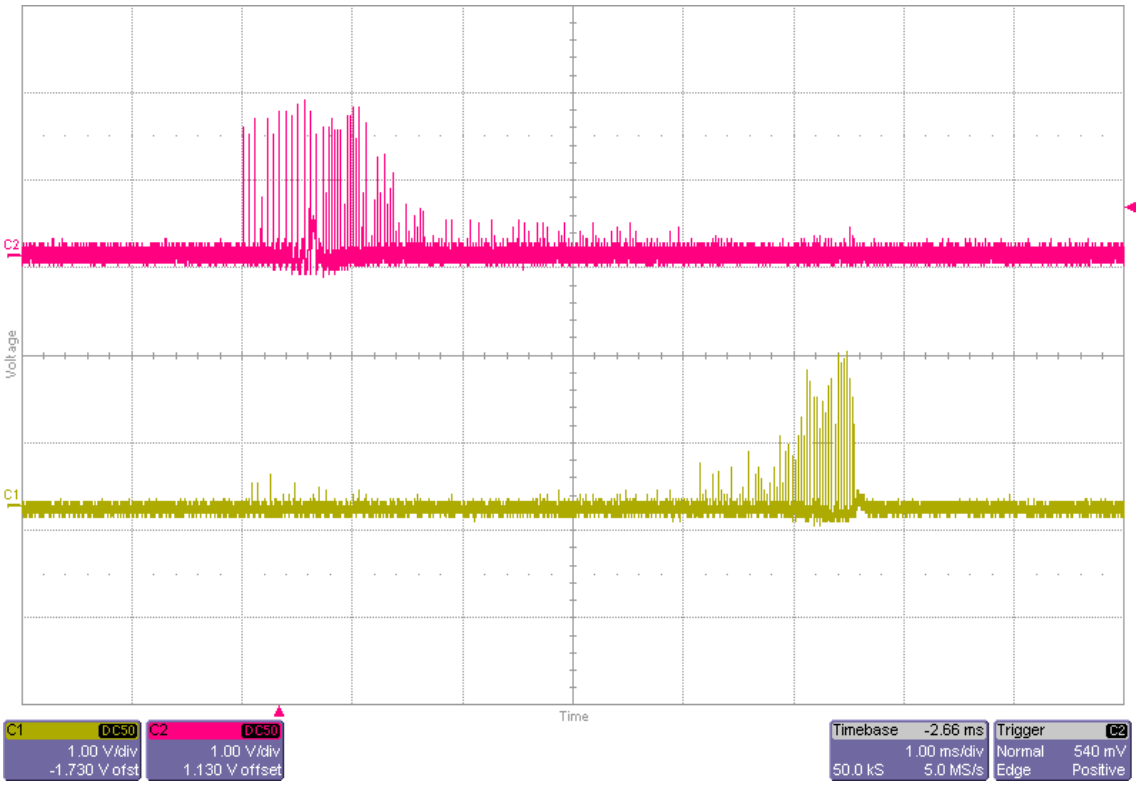
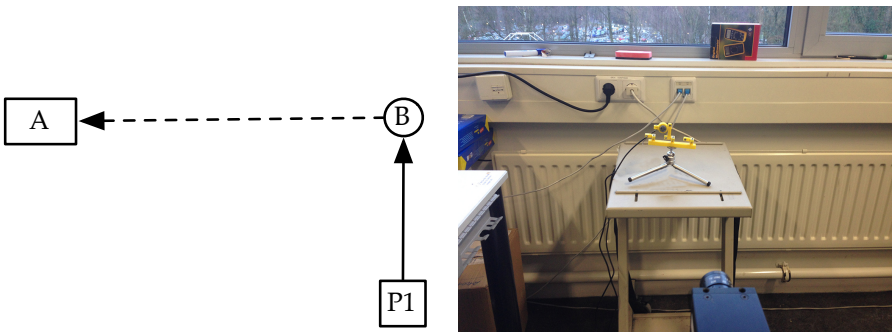


Fig. 53: Visualization of one Lidar pulses measured by two distant transceivers. The gap is proportional to the distance between both transceivers.

With the information of 52 and 53, it is possible to start jamming by transmitting a similar signal. The signal should have the same characteristics as the original signal. This can be achieved by generating a pulsed signal of the right frequency and pulse width. One of the transceiver's input is connected to the output of the Philips PM 5715 pulse generator. This is a device that can generate a square-wave pulse signal with variable width, period and frequency. The setup is presented in 54. There is no feedback from the Lidar to the pulse generator, thus the pulse generator keeps generating a counterfeit signal indefinitely.



(a) Schematic setup.

(b) Actual setup.

Fig. 54: Setup of a Lidar jamming attack. The ibeo LUX 3 is represented by A, the transceiver by B and the pulse generator by P1.

In Figure 55 the oscilloscope output is shown. It is zoomed into one of the repeating bursts. The yellow channel is connected to the transceiver's output, and represents the original signal. The blue channel is connected to the output of the pulse generator, and represents the signal that is jammed back. A simplified representation of the signals is shown in Figure 56. In the figure, the frequency of the counterfeit signal is slightly too high. Therefore, the signal is not in sync with the original signal.

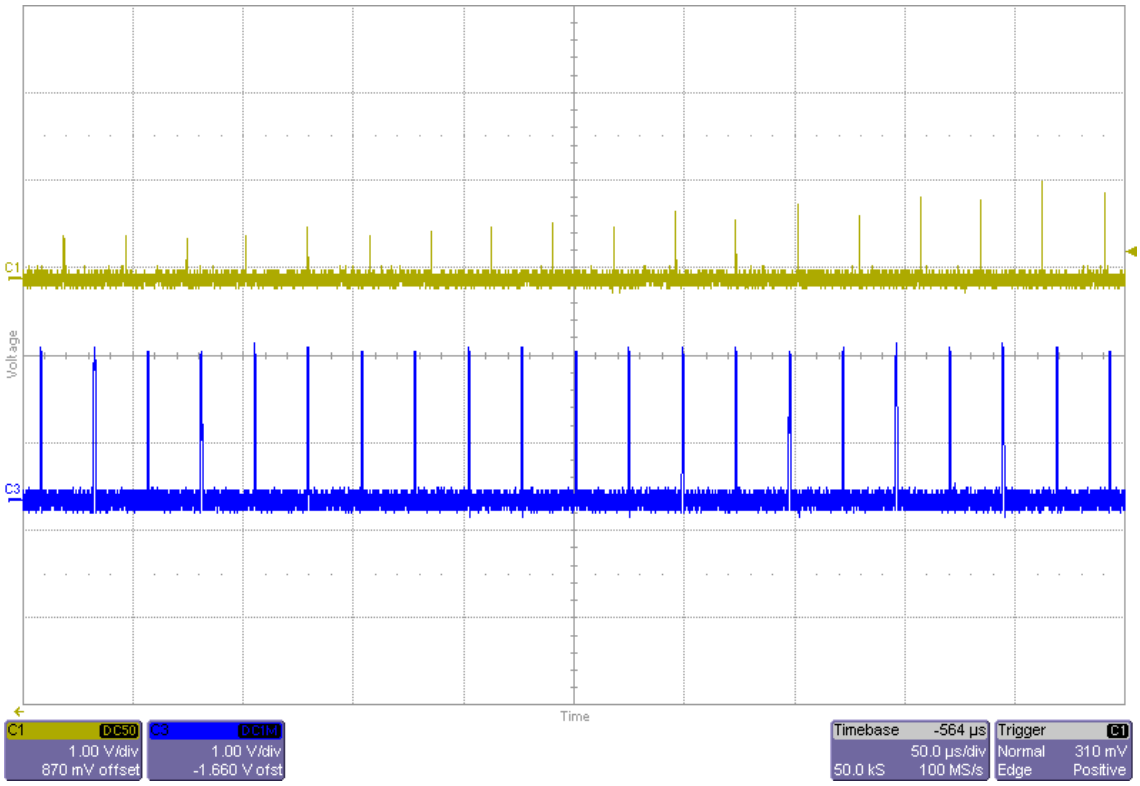


Fig. 55: Lidar jamming signal visualized on the oscilloscope. The yellow channel is the original signal, the blue channel is the generated counterfeit signal.

The frequency of the original Lidar pulse signal on the yellow channel is measured at approximately 30 kHz. The pulse width is measured at approximately 100 ns. The distance between each pulse is the same over time. It is worth mentioning that the transceivers are fast enough to receive and transmit on these speeds. The smallest pulse width that can be generated by the transceivers is 1 ns.

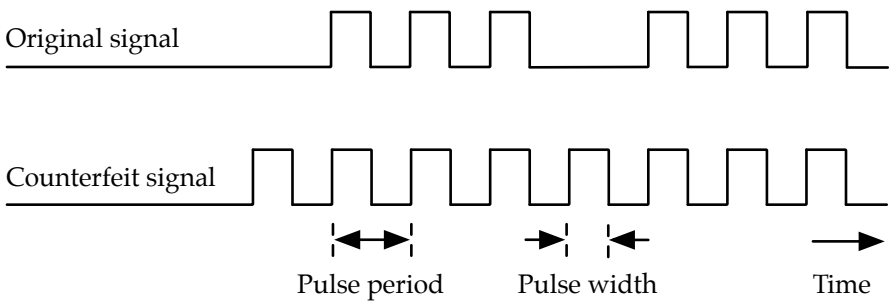


Fig. 56: Simplified representation of the signals shown in Figure 61, and how they relate over time. The pulse width and pulse period are the two variables that can be controlled.

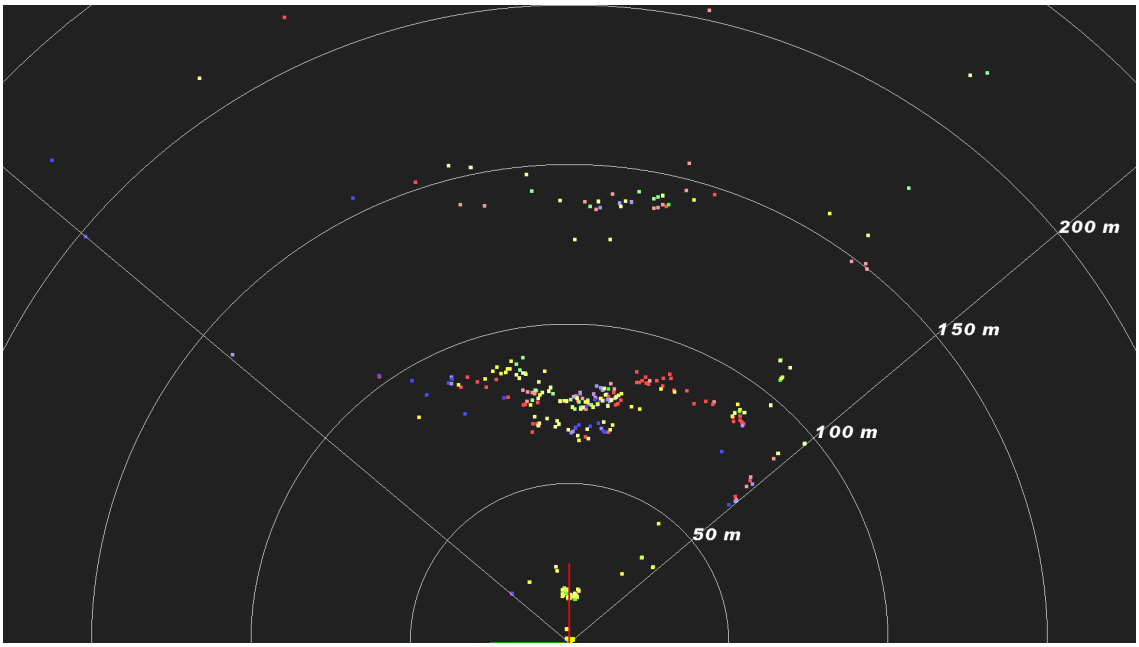


Fig. 57: Result of Lidar jamming attack. All points are fake, except for the ones less than two meters. A few points appear outside the operating range of 200 meter.

The ibeo LUX 3 does receive and process the counterfeit signal, as shown in Figure 57. Therefore, this experiment succeeded. The effect is only visible as the second echo, but as shown in Section 4.2.2, the ibeo LUX 3 will also track objects of the second echo. During the experiments it was noted that the transceiver has a lot of influence, regardless of the position. According to the datasheet of the laser of the transceiver, the beam produced has an viewing angle of approximately 9 °, which is a lot bigger than the angular resolution of the Lidar.

The pulse generator is not required for a jamming attack. At a pulse width of 100 ns, it can be easily replaces by a low-cost Microcontroller Unit (MCU) with a minimal speed of 10 MHz. All of the hardware that is required can be fitted in a small battery-powered device, making the attack compact and less detectable. The most expensive part is the laser diode, which costs 43.25 dollar. This attack can be mounted in a front/rear/side attack or a roadside attack. A battery-powered device can also be mounted on a vehicle by an evil mechanic.

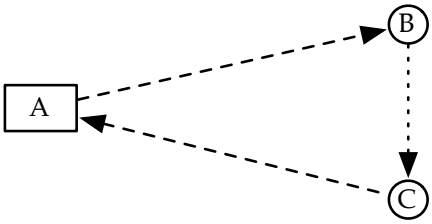
The next section will attempt to relay a signal from one position to another position.

4.2.4 Relaying the signal

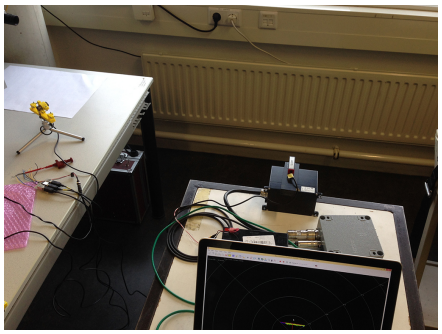
In the previous section, the Lidar was only jammed by generating a counterfeit but similar signal into the field-of-view. This experiment will shown that it is possible to relay the original signal from another position. There are many possibilities with this attack. One example would include to show it is possible to receive a signal from one side, and emit it from the other side. Another example would include the possibility to relay a signal from a Lidar system to completely different Lidar system.

To relay the signals, the two transceivers are used again. The signal received by the first transceiver will be directly fed into to the second transceiver by connecting the output to the input wire, as shown in Figure 58. In the setup, both transceivers are positioned one meter away from each other, but

they do not have to be at the same physical position for a relay attack. Figure 59 shows the representation.



(a) Schematic setup.



(b) Actual setup (rear attack).

Fig. 58: Setup of a Lidar relay attack. The ibeo LUX 3 is represented by A, the two transceivers by B and C. The dashed lines are Lidar signals, the dotted line is a communication channel between B and C.

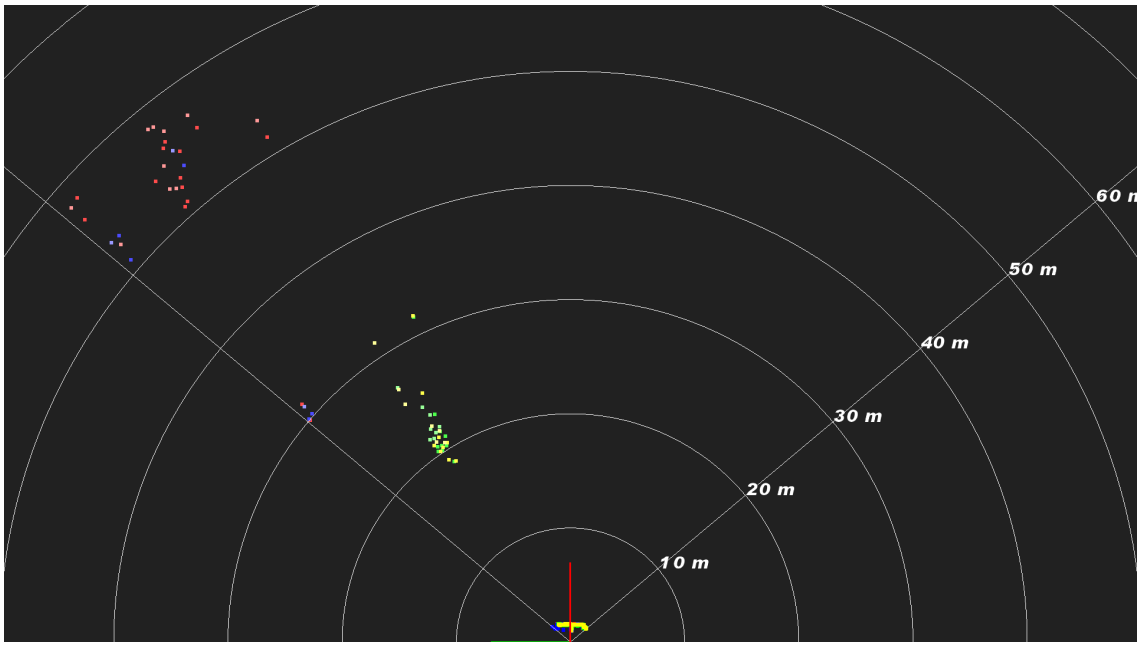


Fig. 59: Result of Lidar relay attack. Light pulses are received from the left, and relayed from the right. The Lidar does not detect this.

The setup in Figure 58 also performs well if the transceivers are positioned behind the ibeo LUX 3. Since the Lidar signals reflect in the distance, some of the reflected light that travels back will also travel past the ibeo LUX 3. If a transceiver receives it over there, the same signals can be retransmitted from another location. Therefore, a direct line of sight is not required to perform a relay attack with these transceivers.

A relay attack is most likely to happen in a roadside attack. An attacker may receive Lidar signals from vehicle at one location and relay them to another vehicle from a completely different location. The attacker only needs two transceivers. No other hardware is required, as the output of one transceiver is connected to the input of the other.

4.2.5 Spoofing the signal

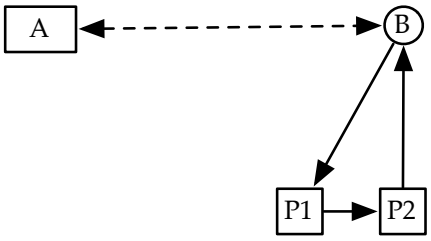
In this section, the jamming attack is extended. So far, the only signals injected are generated pulses that try to resemble the original signal. This experiment will use the original signal as a trigger point to to actively spoof

the ibeo LUX 3, with the intention to re(p)lay objects and control their position.

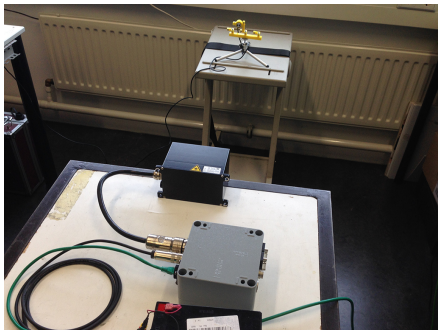
Light travels with a speed of approximately $3 \cdot 10^5$ km/s, or 1 meter every 3.33 ns. With a maximum range of 200 meters for the ibeo LUX 3, the signal travels this distance back and forth in approximately $1.33 \cdot 10^{-6}$ s, or 1.33 µs. This means that the Lidar should listen for at least 1.33 µs for incoming reflections. To successfully inject signals into the Lidar, the counterfeit signal should arrive within this window. The earlier the Lidar receives the signal, the closer it will be to the Lidar. Therefore, if the attacker delays the original signal before it relays it, it can control the position of the objects. Do note that if, for instance, the attacker is at 200 meters, the attack window is smaller since the first 200 meters have already been travelled by the pulse.

A counterfeit signal is generated via external control logic, consisting of two pulse generators. The output of the transceiver is connected to the trigger input of the HP 8011A pulse generator. This pulse generator has the option to delay a signal. As soon as this pulse generator is triggered, it will delay the output. The output of this pulse generator is connected to the input of the second pulse generator a Philips PM 5715. A fixed number of square-wave pulses can be generated as soon as it is triggered. The output of this pulse generator is then connected back to the input of the transceiver. to the trigger input of the first generator. Figure 60 summarizes this setup.

The delay, number of pulses, number of copies, pulse width and pulse period are the variables that can be controlled. As soon as one pulse triggers the control logic, a similar signal is generated of a fixed number of pulses. By tuning the pulse width and pulse period of a signal using an oscilloscope, the counterfeit signal can resemble the original one.



(a) Schematic setup.



(b) Actual setup.

Fig. 60: Setup of a Lidar relay attack. The ibeo LUX 3 is represented by A, the transceiver by B and the control logic by P1 and P2 (not shown in actual setup). P1 is triggered by the Lidar signal, which controls P2.

In Figure 61, a capture of the oscilloscope is shown. The pink channel represents the actual Lidar signal, the green line is the trigger line and the blue line is the generated signal. As soon as the transceiver receives the Lidar signal, the delay is added by the first pulse generator, After the delay, the green channel becomes high. In this period, the second pulse generator will generate a fixed number of pulses, similar to the ones discussed in the jamming attack. This is shown on the blue channel. This process repeats as long as the green channel is triggered, which controls the number of copies.

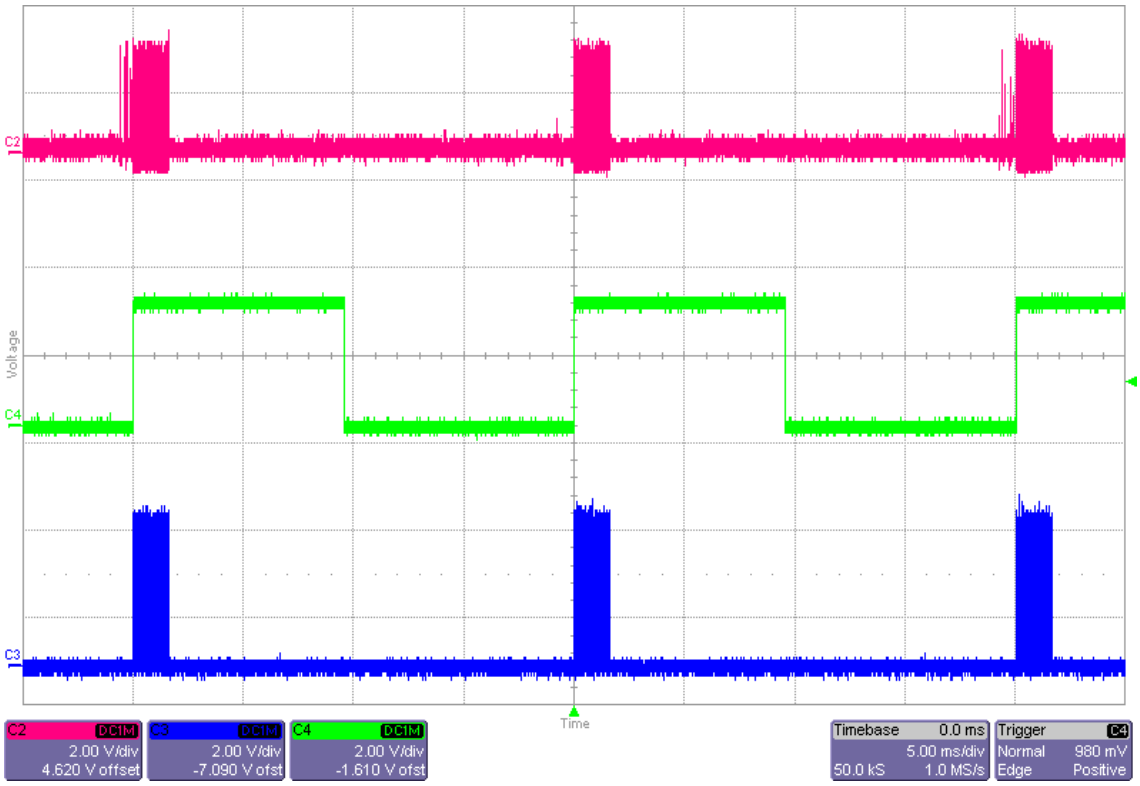


Fig. 61: Result of the Lidar injection attack. Pink channel is the actual Lidar signal, green is the trigger signal and blue is the spoofed signal.

The introduced delay is not visible in 61, as the timebase is orders of magnitudes bigger than the delay itself (nanoseconds versus milliseconds). A schematic overview of one burst is presented in Figure 62. It shows how the trigger delay and the number of copies affect the counterfeit signal.

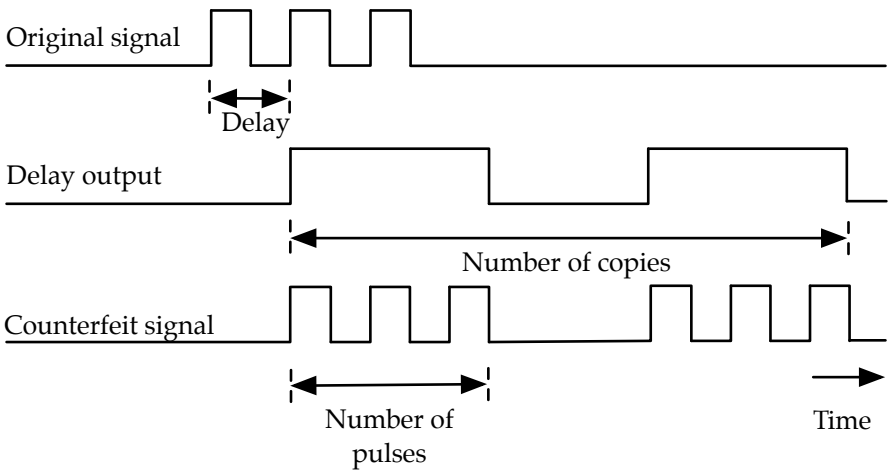


Fig. 62: Simplified representation of the signals shown in Figure 61, and how they relate over time. The delay, number of copies and number of pulses are the variables that can be controlled.

On the Lidar representation, this looks as Figure 63. The points shown are of the second echo. They resemble a copy of the wall shown at approximately 1 meter. By tuning the delay variable, it is possible to make the wall appear closer or further away, until the signal falls outside of the attack window.

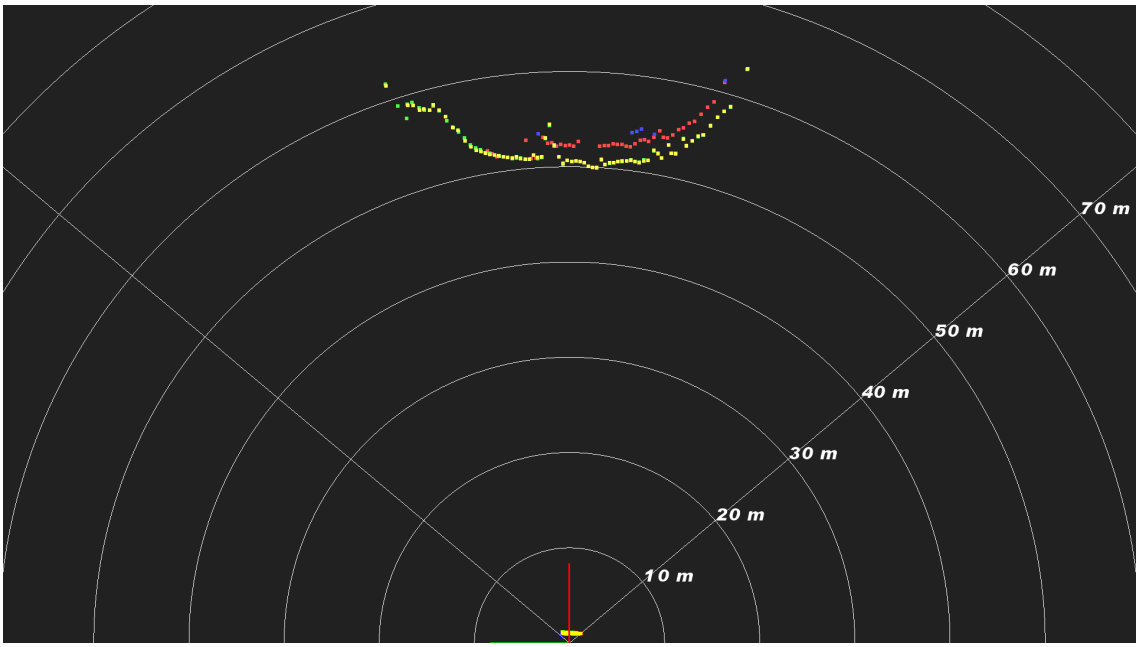


Fig. 63: Result of the Lidar spoofing attack. The injected points resemble a copy of the actual wall at 1 meter.

The first pulse generator can be configured to output multiple pulses when it is triggered. Therefore, it is possible to inject multiple counterfeit pulses in a sequence. By increasing the frequency of the signal on the green channel in Figure 61, multiple pulses on the blue channel can be generated. Figure 64 shows the result, where multiple copies of the wall are shown at regular spaced intervals. The first copy of the wall is of the second echo, the others are a mix of the second and third echo, until it fades out.

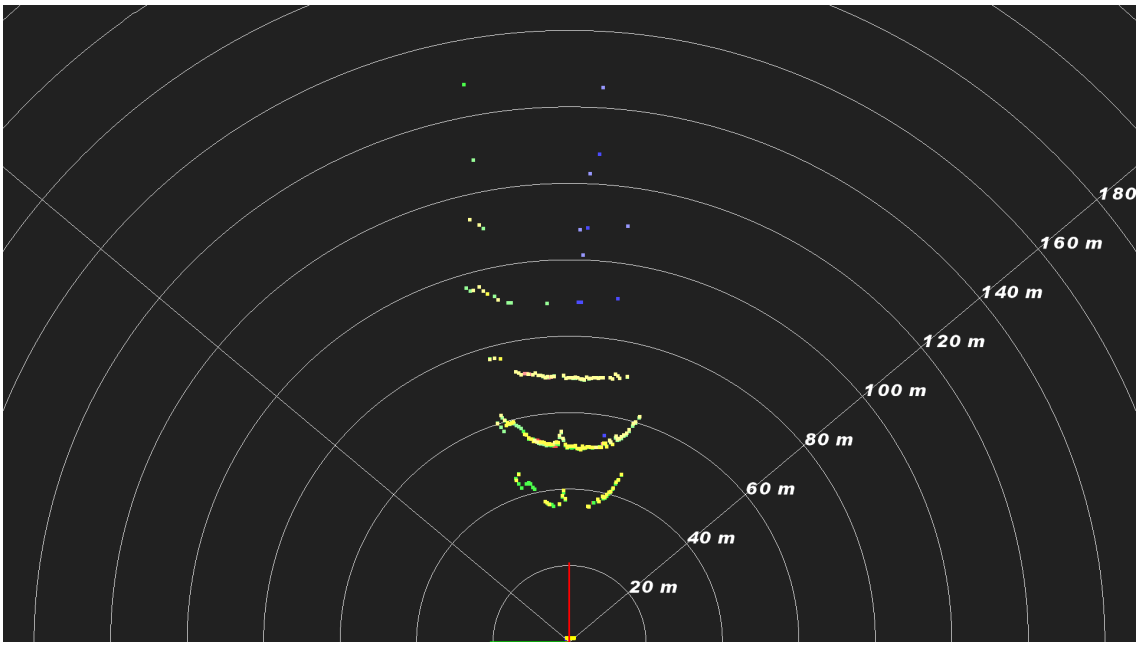
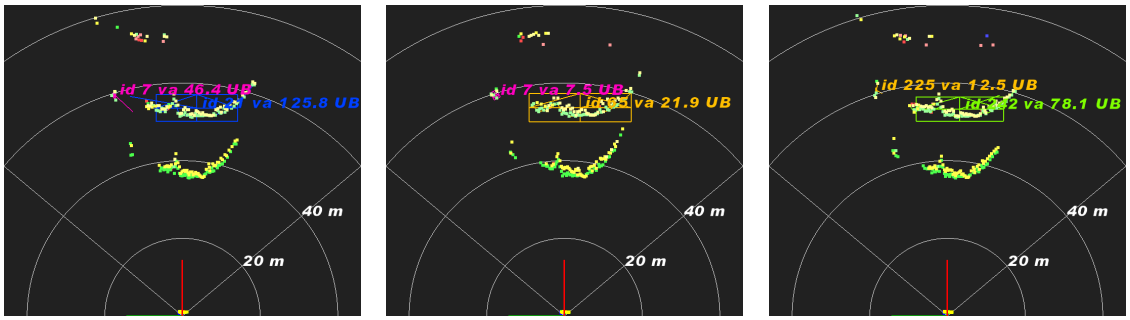


Fig. 64: Result of the Lidar spoofing attack. The points shown are multiple copies of the wall at 1 meter.

Although it is shown that a wall can be positioned

In the beginning of this section it was mentioned that the ibeo LUX 3 associates an object number with detected objects. During the experimentation, it was noted that the ibeo LUX 3 classifies the walls as 'Unknown big' (and sometimes even as a 'Car'). It was not able to keep track of the objects, as

the identification number changed rapidly. This is demonstrated by Figure 65. This figure uses the same data set as Figure 64), but this time with tracking turned on, as denoted by the rectangles around it. In less than 0.46 s, the second wall is identified as three new objects. This indicates that the ibeo LUX 3 classifies the same spoofed object as a new object, therefore it is unable to follow the movement over time.



(a) t = 10.37 s, ID = 21 **(b)** t = 10.59 s, ID = 65 **(c)** t = 10.83 s, ID = 242

Fig. 65: Tracking the second wall over time. A new color represents a new object. For clearance, the tracking boxes for the other objects are not shown.

Both attacks in visualized in Figure 63 and 64 do not happen close to zero meter, but far beyond. In none of the experiments, it was possible to generate points close to the Lidar, e.g. within 5 meter. For instance, in 64 it was possible to inject a reflection of the wall at approximately 40 meter. In low-speed situations, this is not a problem, as an AV (and even a human) has enough time to decide on an action. For high-speed situations this is major problem, as it takes approximately one seconds to travel 40 meter, leaving almost no time to brake or maneuver.

Figure 66 relates timing to the success of the jamming and spoofing attacks. In most of the attacks demonstrated, the counterfeit pulse is received by the Lidar after the first echo is received (the original pulse). This makes a point appear further away, as the Lidar thinks it travelled a longer distance. It can also happen that the counterfeit pulse is received in the gap, after the 1.33 µs attack window. In this case, it will not be noticed. According to Figure 52, this gap can be as long as 20 ms, which is several magnitudes larger than the attack window.

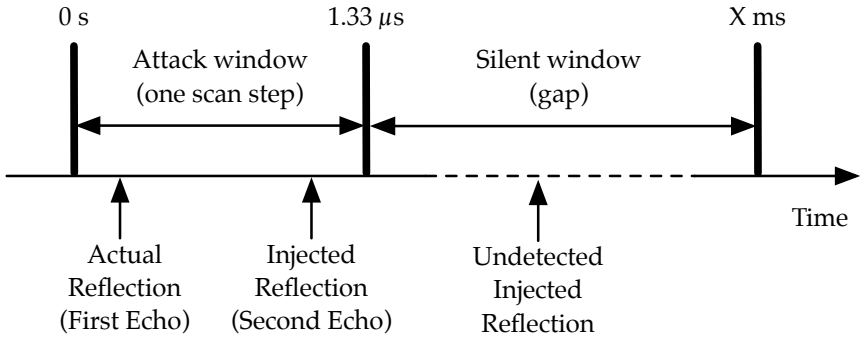


Fig. 66: Lidar attack window. The arrows indicate what would happen if the attacker's pulse hits the Lidar at that time.

Hardware and Cable delay

There are two important causes that are are responsible for this issue. The first cause is related to the pulse generators used. These pulse generators are analog devices and can only be controlled by rotating knobs. It is very hard to tune a knob to the exact right setting. Another issue is caused by the speed of the circuitry. When the pulse generator is triggered by an input signal, the hardware inside adds a bit of delay before an output signal is generated. A digital pulse generator that has a keypad as input can solve

the issue partially, but using dedicated hardware with low switching timing overhead can reduce the delays even more. This could be realized with an Field-Programmable Gate Array (FPGA) or a Digital Signal Processor (DSP). The second cause is the length of the cables. In the setups presented, the average cable length was five meter. According to [62], the speed of an electric signal is approximately two-thirds the speeds of light, which means that every meter adds approximately 6.66 ns of delay. By reducing the cable length, this issue can be lessened.

The relay attack in Section 4.2.4 directly connected the output of one transceiver to the input of the other. No additional delay was introduced. The closest result achieved was 20 meter, while the original object was located at 1 meter. Therefore, the counterfeit signal arrived at least 64 ns after the original reflection[16]. The average cable length of five meter accounts for 33.3 ns. The rest is introduced by the transceiver circuits.

Dedicated hardware or shorter cables will not an attacker to inject a pulse signal before the original signal is received. The moment the hardware decides to emit a counterfeit pulse is triggered when the original pulse hits the transceiver. If all delays are omitted, it would be possible to send a pulse back that arrives at the same time. However, it is possible to synchronize to the pulse timing and calculate the next attack window. This means that when pulse N is detected, the counterfeit pulse will be fired so it arrives in the window of $N + 1$. The delay parameter in this section tried to achieve the same effect. A general purpose MCU of less than 20 dollar is sufficient for synchronizing on nanosecond scale.

An attacker can mount this attack in a front/rear/side attack or a roadside attack. It requires at least one transceiver. Hardware is required to synchronize to the Lidar signal of the target car and to emit a counterfeit signal that can be controlled. This attack is more sophisticated because small variations in timing or delay can have a big impact on the location of the spoofed objects. Nonetheless, the hardware that is required can be low-cost. Because it is assumed that other Lidar systems will use the same wavelength and technologies (see Section 4.2.2), this attack may directly work on other systems.

4.3 CONCLUSIONS

This section will conclude the key results of this chapter.

Two attacks have been performed on the camera. The first was a blinding attack (Section 4.1.3). From the results, it can be concluded that the the White LED spot has the most influence in bright environments. The 940 nm 5x5 LED matrix works best and is much cheaper to scale, according to 3. The auto controls confusion experiment (Section 4.1.4) could not be demonstrated on the MobilEye C2-270, but the effects have been demonstrated on the reference camera. The 650 nm laser ensured that the image did not recover, so that is the most effective option at the cost of detectability. The 940 nm 5x5 LED matrix is the runner up.

Attacks on the MobilEye C2-270 can be mounted from different attack scenarios. The best option would be from a front/rear/side attack, as this gives the attacker the most time to direct a light source into a camera. While the MobilEye C2-270 is not sensitive enough to fully blind the camera using near-infrared light, it can still detect it, and allows an attacker to either

16 Refer to Section4.2.5 for more information on these calculations.

blind objects that should be detected, or inject new objects (e.g. matrix traffic signs). Confusing the auto controls is limited to front/rear/side attack, because it assumes that the attacker continuously switches the light on and off. This is not possible if the attacker is not dynamic.

Three attacks have been demonstrated on the ibeo LUX 3 Lidar system. The first was a jamming attack (Section 4.2.3) using a continuously generated pulse signal that resembles the original signal. The second attack showed that the Lidar is sensitive to relay attacks (Section 4.2.4), even from behind. The last attack was the spoofing attack (Section 4.2.5), where an original observation was spoofed, including multiple copies.

The jamming and relay attacks are less sophisticated than the spoofing attack. The jamming and spoofing attack need at least one transceiver (approximately 44 dollar), while the relay attack requires two. The range for emitting a counterfeit signal is approximately 100 meter, although the gap between the scan steps is approximately 1.47 meter at this distance. Therefore, multiple photodetectors will be required. Section 4.2.2 showed this. All of the attacks can be mounted from the roadside. Spoofing and jamming can also be mounted from front/rear/side attack. Because the hardware for a jamming attack can be very compact, it can be battery powered and even installed by an evil mechanic on another vehicle. If an attacker designs one attack and shares the blueprints (e.g. via internet), the only thing another attacker has to do is acquire the hardware connect the wires.

5 DISCUSSION

5.1 IMPACT ON APPLICATION LEVEL

The attacks that have been demonstrated in the previous chapter are directed at the hardware layer. This section discuss the impact of the attacks on the application layer briefly. The application level involves processing the sensor data, for instance cleaning the data, classifying the data and fusing the data with other sensors. The processing steps for the MobilEye C2-270 and the ibeo LUX 3 are not documented. However, it is interesting to see how the application level processes malicious input and if it can detect malicious input. The impact on camera systems, Laser Image Detection and Ranging (Lidar) systems and sensor fusion will be discussed.

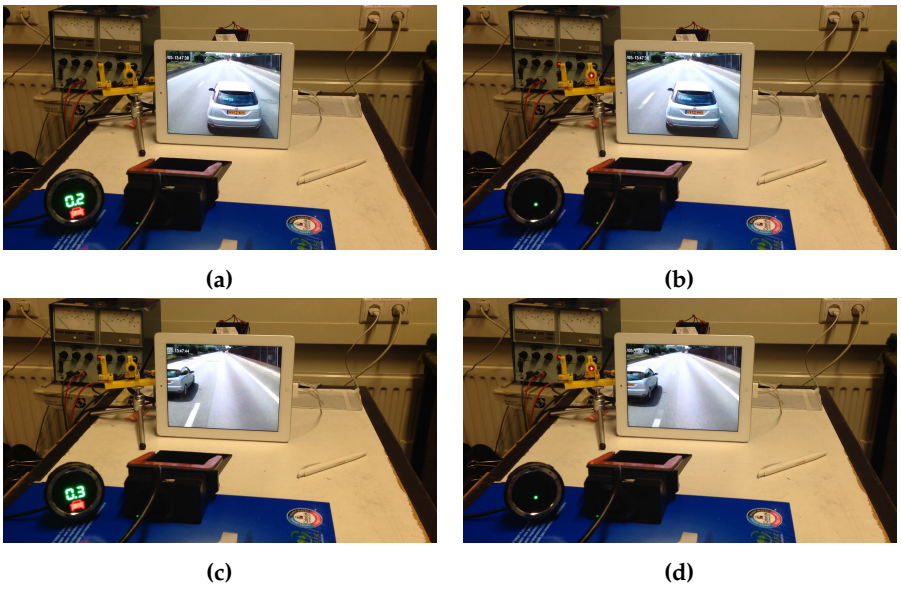
The videos that have been recorded for this section, including others from the experiments in Chapter 4, can be found at `http://goo.gl/SVkvor`. The videos are annotated.

5.1.1 Camera

In the first experiment, an iPad is put in front of the MobilEye C2-270 camera. The objective is to show that the Collision Avoidance System (CAS) fails to work when the camera is blinded. The MobilEye C2-270 is connected to a simulator, that is setup to report a speed of 130 km/h[1]. A video is played with footage from a dashboard camera, as if the MobilEye C2-270 was installed in an actual car. First, it is shown in 67a-c what would happen without any tampering, then a laser pointer is turned on, with the intention to blind the camera in Figure 68b-d. Instead, the display is blank when it is tampered with. This attack is repeated in Figure 68, to show that this attack is reproducible for different videos.

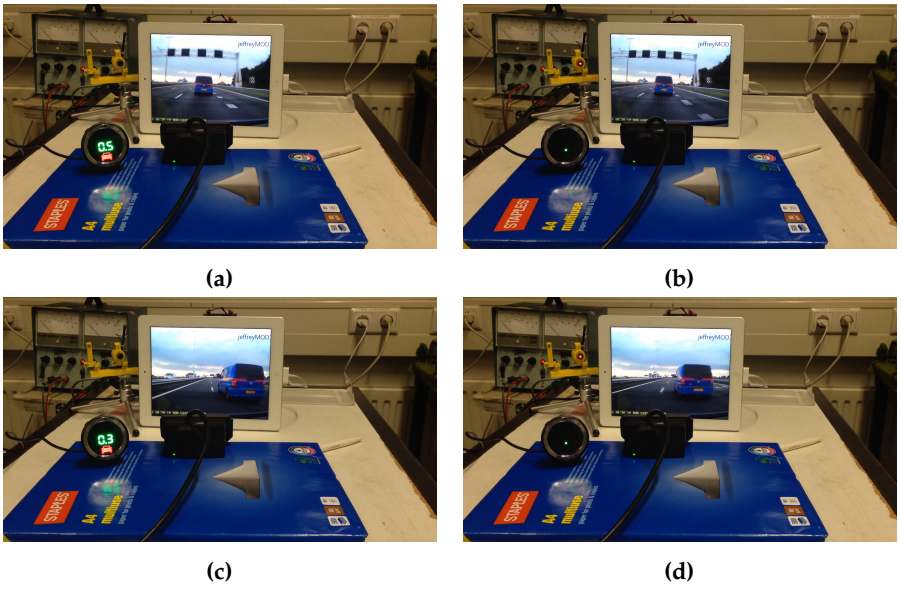
In all figures, the display is positioned in the lower-left corner. When it detects an approach collision, it will notify the by user sounding an alarm and showing the time-to-impact (in seconds). Although the 650 nm Laser pointer[2] is mounted close to the camera to optimize success level, it does show the potential of this attack. Even for short periods of blinding, the camera can be blinded and have troubles detecting the approaching vehicle. For both attempts, the MobilEye C2-270 does not detect the malicious input nor signal the driver.

1 At least 40 km/h is needed to trigger collision warnings.
2 The same laser pointer from earlier experiments.



(a) (b) (c) (d)

Fig. 67: MobilEye live blinding experiment. Four frames of the same sample, in which the laser is turned on and off.



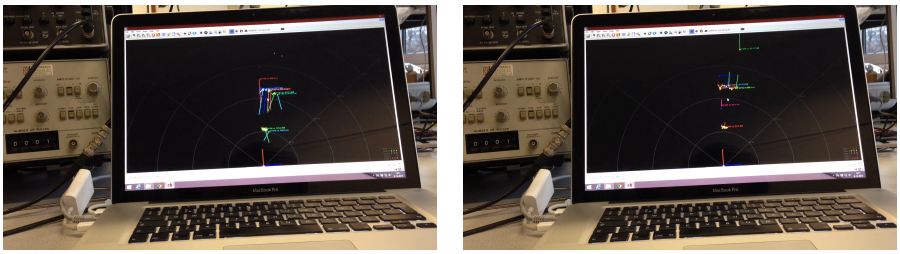
(a) (b) (c) (d)

Fig. 68: Second MobilEye live blinding experiment. Four frames of the same sample, in which the laser is turned on and off.

5.1.2 Lidar

The ibeo LUX 3 contains an embedded tracking system. The tracking system can group points and classify them as a car, truck, bike, pedestrian or unidentified object. It can also track objects over time. This allows the ibeo LUX 3 to determine an object's direction. It was demonstrated in Section 4.2 that the Lidar is sensitive to counterfeit pulses generated by an adversary. Furthermore, it was possible to control the distance where objects appeared, by varying the delay of the trigger signal for the second pulse generator.

The frames below are almost sequential, and show how the position of the objects reverses immediately. While this is no hard job for the tracking software, it can potentially confuse the decision system of an Autonomous Vehicle (AV).



(a) **(b)**

Fig. 69: ibeo LUX 3 live experiment. The objects change direction (the lines), as instructed by the attacker.

Furthermore, there is a limit on the number of objects that can be tracked by the ibeo LUX 3. By introducing noise or more sophisticated objects, a denial-of-service attack can be mounted on the Lidar, by introducing a large number of noise or spoofed objects. Real objects that should have been detected can than be missed. Extra tracking will also cost more cycles.

5.1.3 Sensor fusion

Sensor fusion was introduced back in Section 3.2.2, and introduced the Kalman Filter (KF) and Particle Filter (PF). Both algorithms take sensor data as measurement input from one or more sensors. The output of a sensor fusion algorithm is a new state, e.g. where the algorithm thinks an object is. In literature, the PF is generally used for tracking objects, whereas the KF is frequently used for fusing Global Positioning System (GPS) with Inertial Measurement Unit (IMU) sensor data.

Figure 70, 71 and 72 below are based on the cannonball tracking case study in Appendix A.2. In this case study, where the cannonball can measure the distance to three beacons, the trajectory of the cannonball is tracked with a PF. Do note that the figures presented in this section are a base on simulations only, and are not based on the internals of the ibeo LUX 3[3].

In Figure 70, the beacons (colored dots) change position clockwise every iteration step. Therefore, the cannonball will observe more measurement noise. This attack is similar to the 'Wave-based attack' mentioned in Section 3.2.3, where the measurement is reversed continuously.

3 The ibeo LUX 3 does not use a PF but a KF, according to [11].

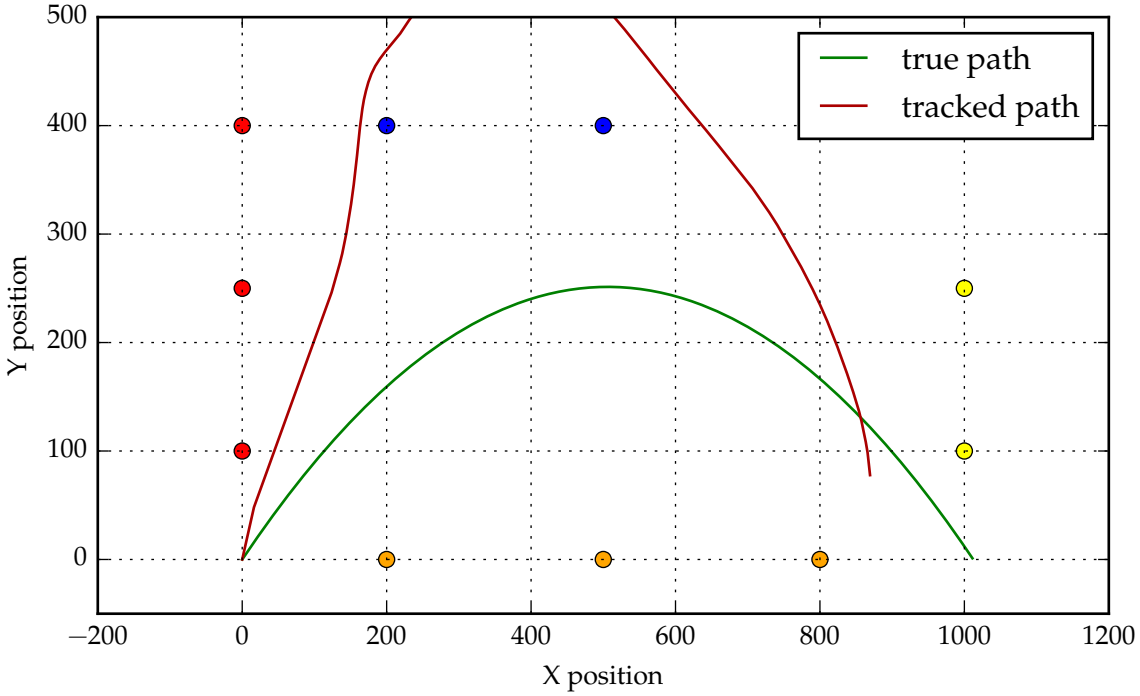
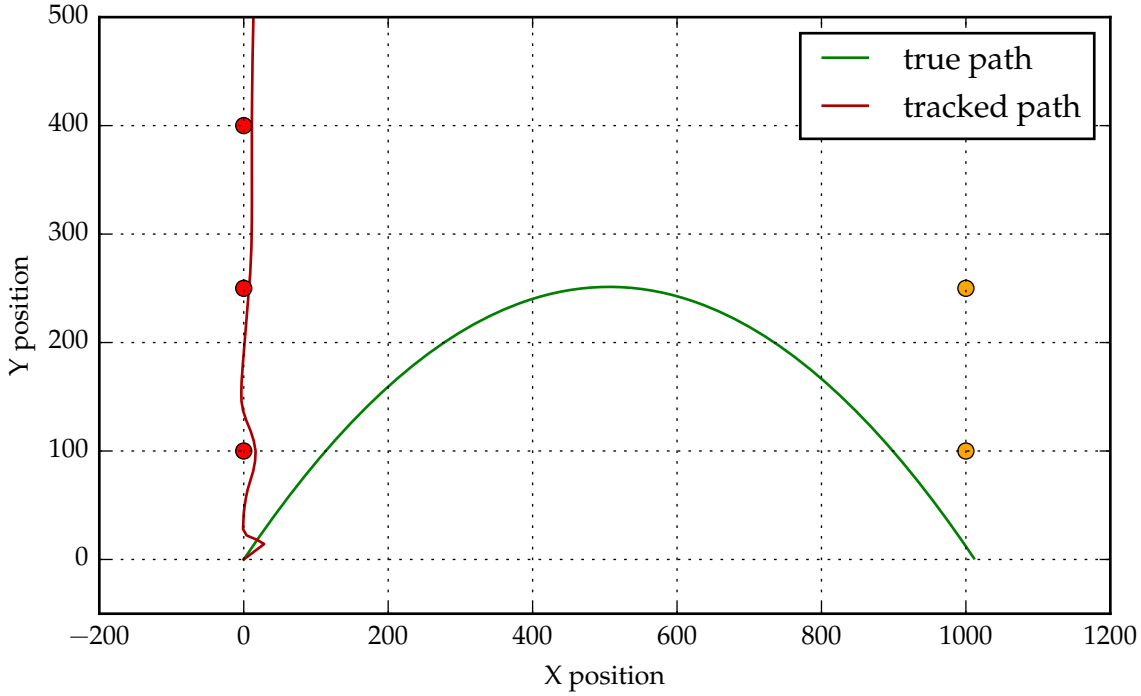


Fig. 70: Spoofing the PF with alternating the beacons. The colored dots represent the beacons, which change clockwise each iteration step.

In Figure 71, three beacons move from the right side (orange) to the left side (red). Initially, the cannonball moves up. From a cannonball point of view, the beacons are moving towards the cannonball. Therefore it assumes that it is moving in the right direction (upwards).

This situation resembles the Lidar spoofing setup from Section 4.2.5. In there, it was explained how an attacker could change the position of counterfeit wall by varying the delay. If the cannonball is replaced with a AV and the three beacons with a wall, the attack explained what happens if the attacker makes the AV believe that the wall is approaching.



In Figure 72, the beacons have been relocated three times, in a horizontal way. The first time is denoted by the orange beacon (the other two are out of range of the plot), the second time by the red color and the third time by the blue color. The shape does follow the true path, but due to ambiguity (see Section 3.2.2), it first moves to the left and then recovers.

Fig. 71: Spoofing the PF with moving beacons. The colored dots represent the beacons, which gradually moved from left to right.

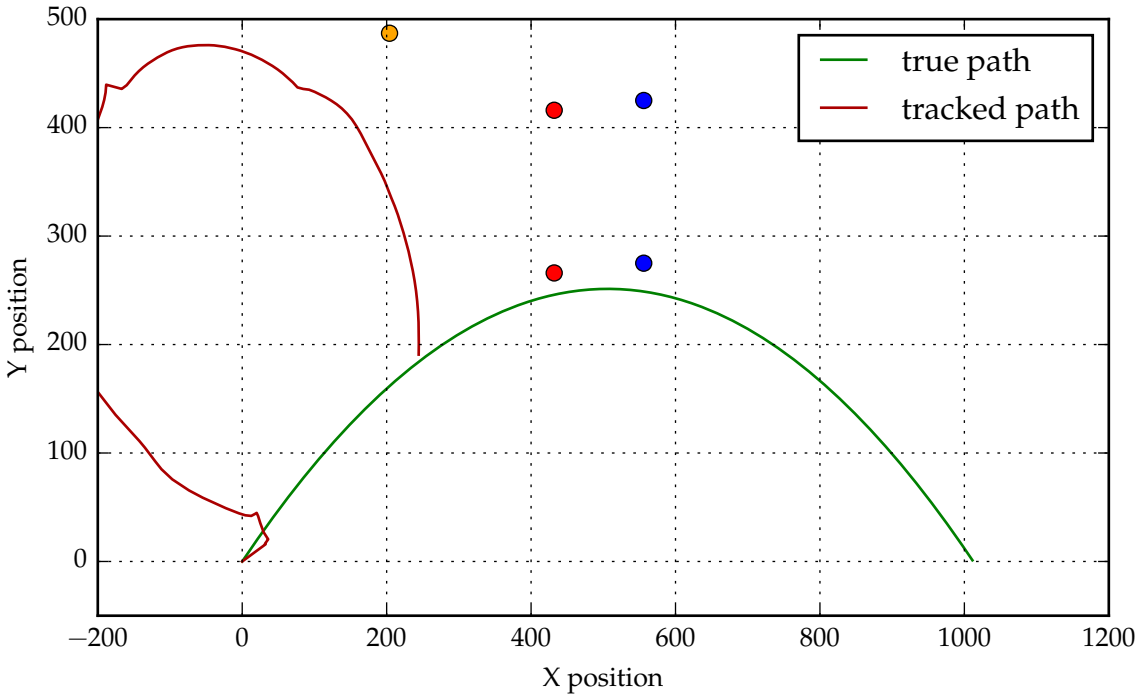


Fig. 72: Spoofing the PF with random beacons. The colored dots represent the beacons, which have been randomly positioned in a vertical line.

5.2 COUNTERMEASURES

Some countermeasures can be applied to mitigate or reduce the attacks as demonstrated. It is worth noting that these countermeasures have been established from knowledge gained during the experiments and not tested in during the experiments. The main focus is tampering detection (or to limit tampering), not fault detection, e.g. if the sensor still performs as specified.

Do note that no internal knowledge of the system is known. It may therefore be that some countermeasures have already been applied. In that case, the countermeasure should be considered a design recommendation for other systems.

5.2.1 Camera

Up to a certain limit, it is possible to protect cameras from being tampered with. There is a trade-off between protecting the camera from tampering, sensitivity, image quality, camera size and price. Most of the countermeasures require the camera to be modified. This not only raises the costs, but will also cost more space. For instance, in [52], cameras have been integrated in the side mirrors, a place where space is limited.

Redundancy

By using multiple cameras it is harder for an attacker to blind all of the cameras at the same time. The experiments have shown that using a 5 mW laser is the most effective way to temporary blind a camera. Unfortunately, due to the small beam width this attack was only limited to a singe image sensor at a time. At a distance of 50 cm, the width of a focused beam was measured at approximately 1.5 mm. The size of the MobilEye C2-270 lens was measured at approximately 5 mm.

By introducing multiple cameras that perceive the same image (or at least overlap), the attacker has to put more effort into the attack to blind both cameras at the same time. This does require more space to fit the cameras and a pair of cameras need to be carefully calibrated so the overlapping image is not misaligned. Software should blend the separate images together. As long as the cameras have a static position with respect to each other, the parameters for blending the images together have to be setup only once. Other challenges of this countermeasure include synchronized capturing and maintaining the same exposure [9]. Introducing extra cameras will not create new attack vectors, as it only extends vision.

Introducing extra cameras may not protect from military grade weapons such as a 'Dazzler' (see Section 3.1.3). According to [8], the width of these laser beams can be configured up to 12 cm, at the expense of output power on the same area and range. This makes it a lot easier for an adversary to aim at a camera sensor, even if the camera sensors would be a lot bigger (e.g. more area to blind). If multiple cameras are used to complement each other, then it is also possible that the 'Dazzler' will hit several camera sensors at the same time (see Figure 16).

Optics and materials

Integrating a removable near-infrared-cut filter, a technique that is available to security cameras, can filter near-infrared light on request. The filter can be applied by switching an electromagnet. During day time, the filter is applied to yield a better image. During night time the filter is removed to make use of infrared light for night vision. When the filter is applied, it will also block infrared light sources, hence this countermeasure is only effective against during day time.

To improve this countermeasure, the filter could also be applied when the camera decides it is needed, for instance when it is jammed (see next countermeasure), or when the auto controls cannot be optimized for the bright lighting conditions anymore. In this case, it is assumed that jamming the sensor is already in progress. This may introduce a new attack vector, as an attacker may repeatedly attack the auto controls(as demonstrated in Section 4.1.4) to let the camera apply the filter or remove it. Depending on the quality of an near-infrared-cut filter, the camera may be damaged.

Another option is to use photochromic lenses. These types of lenses can change color to filter out specific types of light. An example includes glasses with darkening lenses in sunlight. The type of lenses (or coating on the lenses) will determine the type of light it will filter. According to [159], vanadium-doped zinc telluride (ZnTe:V) is a material that can filter light with a wavelength of 630 nm - 1300 nm. High-intensity beams will make the material more opaque, therefore filter more. The advantage of these type of materials is that they do not affect the image in low-light conditions.

Spectral analysis

This countermeasures will only help to detect a jamming attack and is similar to the spectrometry experiment in Appendix B.

A spectrometer such as the AvaSpec-2048 USB that is used in this work, uses a dispersive element such as a prism. A prism will decompose an incoming light beam into several beams per color, because the refraction index is wavelength dependent. By positioning an image sensor such as a Charge-Coupled Device (CCD) or Complementary Metal Oxide Semicon-

ductor (CMOS) to receive this beam, the wavelengths and intensity can be measured.

The light sources used in this work have a characteristic wavelength. For instance the red laser has a steep peak at 650 nm. Since environmental light has influence on the amount of light needed to blind a camera, it is assumed that it needs at least the same intensity to be visible to the camera. Therefore, if the light spectrum is observed over time, it can be possible to detect whether an attacker is pointing a light source into a camera.

To perceive a camera image and do a spectral analysis at the same time, the incoming light should be split with a beam splitter. A beam splitter is an optical device that splits an incoming beam. One part will pass-through, while the other part will reflect off. Both beams will have a lower intensity, which may be undesired for the camera system.

A schematic overview of the proposed setup is presented in Figure 73.

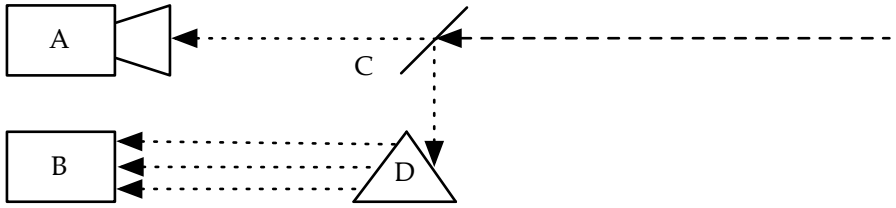
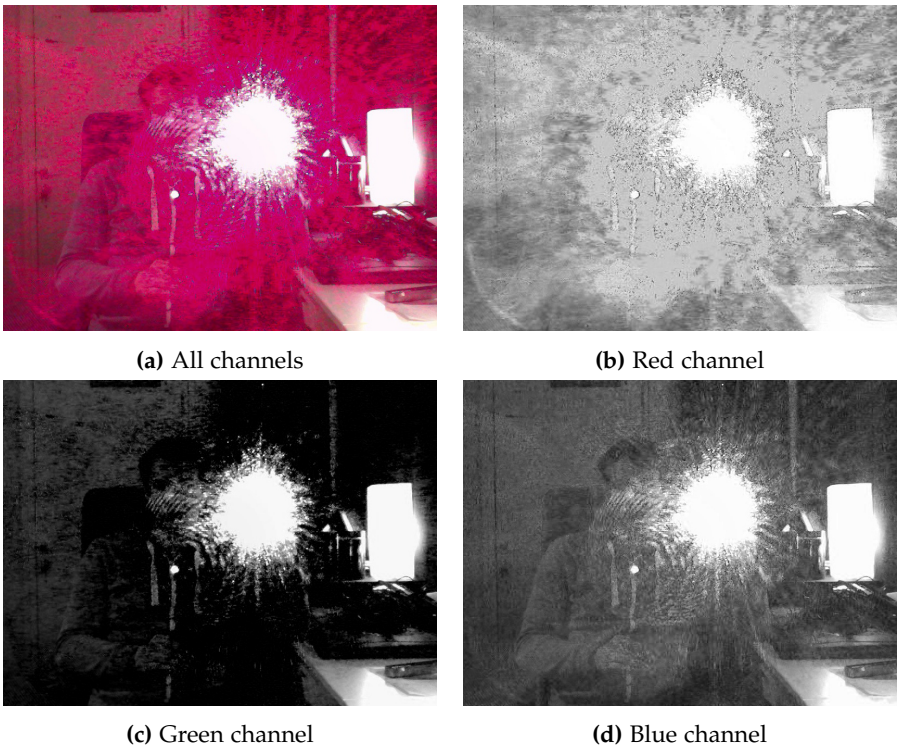


Fig. 73: Schematic setup of a spectral analysis setup. The camera is represented by A, the spectrometer by B, the beam splitter by C and the prism by D. The arrows indicate light beams.

Image channel separation

This approach is similar to some camera applications to find objects, as discussed in Section 3.1.3. The MobilEye C2-270 is a Red/Clear camera [7]. The resulting image is a two-channel image only. The red channel is most sensitive to near-infrared light, so if that channel is jammed, it can use the other channel to filter the near-infrared light. The same approach works by extending it to a multi-band channel camera, such as explained in Section 3.1.3. This countermeasure can help to restore some detail of the image, but will not recover the full image. However, this countermeasure can be implemented in software efficiently, thus requires no extra space in a camera.



(a) All channels **(b)** Red channel

(c) Green channel **(d)** Blue channel

Fig. 74: Illustration of image channel separation. The same image has been decomposed into several channels with software. Although the laser overexposed the image, the blue channel image still shows detail.

An attacker may decide to use several light sources to rapidly change color. This renders the above countermeasure less effective, since all channels may be overexposed. Military grade examples of light sources that can change rapidly are wavelength-agile laser [123].

5.2.2 Lidar

For the Lidar, the following countermeasures can prevent some of the attacks from happening, or help to detect them. An expert in the field of signal jamming[4] was considered to validate the countermeasures.

The countermeasures below can be implemented in software[5]. A modification of the sensor hardware is not necessary to implement these countermeasures, but the firmware can be changed to implement some of the countermeasures proposed (at the expense of range or accuracy. However, no information can be provided to indicate if the countermeasures are already implemented in the object tracking software of ibeo LUX 3, as the sensor was only tested on raw data level.

Redundancy

The experiments have shown that it is possible to jam and spoof on the ibeo LUX 3, that uses 905 nm wavelengths. According to [87] and [122], it is possible to use different types of wavelengths for Lidar vision[6]. Although some wavelengths have drawbacks in terms of range, combining multiple wavelength Lidar makes it harder for the attacker to both signals at the same time. According to [122], the costs for the required hardware will exceed the budget for the attacker model considered for this work.

4 In particular, jamming speed measurement devices.
5 With the exception of redundancy, as it requires a completely different design and featureset.
6 Do note that the wavelengths should not overlap. For instance, using a 850 nm Lidar will still influence the 905 nm Lidar.

Another way of adding redundancy would be to use Vehicle-to-Vehicle (V2V) communication[7]. If an attacker mounts a front/side/rear or roadside attack, the chances are it will only affect a single vehicle. If other AVs share their measurements, the attacked AV could compare the measurements with what other vehicles observe. It is assumed that a comparison with other vehicles requires the vehicle to detect tampering, ask other vehicles to share their data, validate the data (e.g. an attacker may intentionally share incorrect data), compare it and decide on an action. This process may cost too much time and utilize an expensive link between vehicles.

Random probing

There are two ways of achieving this. As shown in Section 4.2.2, the pattern is repeated at a fixed interval. This interval depends on the scanning speed, and thus the rotation of the mirror inside the ibeo LUX 3. Furthermore, the attacker needs to synchronize on this interval, so it knows exactly when to fire a pulse back. By varying this period non-predictably, it will be harder for the attacker to synchronize on the original signal. This countermeasure can be hard to implement and can be problematic for rotating Lidars. They mostly require a constant rotation speed and need to know exactly at which angle they fired a pulse. Varying the rotation speed can degrade the lifetime of the device, as it can introduce damage to the system. However, slight alternations may be sufficient, without affecting the lifetime too much.

Another option that takes less effort to implement, is to (non-)predictably[8] skip certain pulses. This can be realized in the software that is controlling the laser emission. When a pulse is skipped, it introduces an effect that is similar to varying the scan speed. If the Lidar skips a pulse, it can still listen for incoming pulses. If it notices a response, this may indicate an attacker is tampering. It depends on the application whether this is acceptable or not. However, at a scan frequency of 50 Hz, missing a few pulses will probably not have much effect on the resolution, especially close range.

Both of these countermeasures can be implemented in software[9] and are effective to roadside attacks. A roadside attacker now has to synchronize on the pulse train and synchronize on a (non-)deterministic code. If both sequences are long, it can take several seconds, at which the target may already passed the roadside attack. Any 'misfire' from an attacker lets the Lidar know if an attacker is tampering with the signal.

Probe multiple times

This countermeasure is only effective against random jamming. If an attacker is not in sync with the pulse signal generated by the Lidar, counterfeit pulses will appear at random intervals in the attack window, as demonstrated in Section 4.2.3. For instance, if the Lidar measures three times at a the same position and it measures three different distances (e.g. 40 m, 10 m, 150 m), this measurement is likely to be invalid.

Probing multiple times does introduce three new problems. The first one is that it decreases the scan frequency. Probing four times will effectively convert a 50 Hz device Lidar into a 12.5 Hz device. Second, the measurements should be corrected, to compensate for any movement of the vehicle in between the measurements. This should not be a major limitation, since

7 This countermeasure will also work for camera-based system.
8 Embedded hardware may be limited to generate true random sequences.
9 Assuming the software can control the exact speed of the rotating mirror.

most modern vehicles advertise the speed of the vehicle via the Controller Area Network (CAN) bus. Lastly, the software should decide whether a measurement should be marked as invalid or not. Removing any outliers up to a certain limit will have a small impact on resolution, but at a close range, this may not be a problem. Another option can be to average the measurements using a rolling average or KF. This has been explained in Section 3.2.1.

This countermeasure can be implemented in software.

Shorten the pulse period

In Section 4.2.5 it was calculated that the ping period is approximately 1.33 µs, the time it takes for the light pulse to travel forth and back. This gives the attacker an attack window of less than this period.

By lowering the ping period, the attacker has a smaller time frame to mount an attack. Lowering this period will also lower the maximum range. By halving the period to 0.66 µs, the range of the ibeo LUX 3 will decrease to 100 meter. If this is an acceptable range, depends on the application. Depending on how the ping period is determined, this countermeasure may require a change in hardware.

The effectiveness of this countermeasure depends on the type of attack. For instance, in a front/rear/side attack, this countermeasure is typical less effective, as the attacker is allowed to constantly move around, e.g. have a jammer installed in the bumper and drive in front of the target. For a roadside attack, this has more effect, as the maximal range decreases. It is still effective, but only at shorter distances.

Increase tracking limits

A denial-of-service attack can be mounted on the Lidar by jamming or spoofing a large number of objects. The limit of objects the ibeo LUX 3 can track is 65. This number is a rather small number, especially if the Lidar would operate in three-dimensional world. Adding support for more objects that can be tracked will increase the computational overhead. However, as hardware is getting faster and cheaper, it is only a matter of time.

5.3 LIMITATIONS

There are limitations to every experiment, including the experiments in this thesis. Some limitations are a direct result of the requirements that were defined back in Chapter 4.

5.3.1 One sensor does not make it an AV

In all of the proof-of-concept experiments conducted in Chapter 4, all sensors were tested separately. However, one sensor does not make a vehicle an AV. In the literature, a typical AV [16, 46, 4, 22] uses a combination of sensors, most notably a combination of Lidar, Global Navigation Satellite System (GNSS) and camera.

It is worth noting that the attacks on the ibeo LUX 3 and MobilEye C2-270 were not detected by the systems. There was no signal to indicate if the device noticed malicious input. This may be completely different for other systems and implementations available on the market, but it is believed that

fully automated vehicles that currently exist will also fail to detect malicious input. GNSS is an example of a sensors that considers loss of signal (e.g. due to jamming), and will then use IMU as 'backup'. A similar alternative for camera or Lidar was not found in the literature.

5.3.2 Proof-of-concepts only

All of experiments are a proof-of-concept only, so they are maximized to achieve success. When the desired goal is reached, it is considered a positive result.

The testing conditions are another limitation. Since all proof-of-concepts experiments have been conducted in the lab, the results will only be valid for these conditions. Conditions that affect performance include the amount of environmental light (no sunlight conditions were tested), weather conditions (rain, snow, fog) and of course range (limited due to lab size). Nonetheless, all of the experiments conducted are particular plausible scenarios, and their outcomes show why that particular experiment should be further explored.

In practice, most experiments will be a lot harder to conduct. For instance, it will be a lot harder to emit a laser beam into a camera if the target is moving. Also, windshields may limit the effects of a light source, e.g. it may diverge the beam. Computer vision and robotics may automate this process and can do this more accurate. However, this will exceed the cost the attacker model and may require more time than is defined.

Regarding the Lidar experiments, no range tests have been performed. The laser part of the transceiver has a range up to 100 meter with a viewing angle of 9 °. Therefore, it will be easy for an attacker to emit a laser beam at a large distance on a moving vehicle, which is the only part required for a jamming attack. But for relay and in particular spoofing attacks, receiving the original signal is more important. As larger distances between the attacker and its target, the gap between the Lidar pulses will also increase, as proven by Section 4.2.2. At a distance of 100 meter, the gap between two sequential Lidar pulses is approximately 1.47 meter. Since a photodetector has an aperture of only 5 mm, it is very likely pulses will not be detected.

5.3.3 Experiment limitations

The attacks presented in this work address both camera and Lidar systems and are not limited to a single attack. From a black-box testing point-of-view, this is a reasonable choice, since it is not clear what the effects will be of an attack on a device. However, this limits the comprehensiveness of the attacks.

In the camera experiments, different light sources have been used. These light were the ones that were available on the market, and could be bought for reasonable prices. As noted before, all light sources are all a tiny bit different, in terms of viewing angle, output power, and emitted wavelength. Therefore, it was not possible to conduct a survey, based on equal parameters. Choices of the light sources for later experiments were solely based on interpretation of the preceding results.

The results of the Lidar related experiments are sufficient, but they could have been improved if more decent hardware was used to do pulse synchronization and pulse generation. The pulse generators used during the setups were the only two available that provided the required functionality of de-

laying a signal and generating a fixed number of pulses. Configuration of the parameters happens via turnable knobs, which is hard to tune precisely.

Lastly, most of the experiments combine several fields of studies. Most of the attacks involve knowledge of electrical engineering (Lidar) or physics (camera). The required knowledge to do the experiments was gained via experts. Therefore, designing and testing attacks on such systems requires multi-disciplinary knowledge.

6 CONCLUSIONS AND FUTURE WORK

6.1 SUMMARY

Chapter 1 introduces the context of this work. The problem statement introduces the lack of research on malicious input on sensors. A decision by a Autonomous Vehicle (AV) is as good as a sensor can perceive. Any form of tampering that influences a sensor perception can therefore be catastrophically and cost lives.

In Chapter 2, the background on AVs was established, addressing topics such as the levels of automation and attacker models. This work considered an attacker model that has limited money and limited time, that intentionally wants to attack an AV without having access to the car's internals. Possible attack scenarios have been established, including potential attack scenarios such as a 'Front/rear/side attacks', 'Roadside attacks' and 'Scenery attacks'.

Commonly used automotive vehicle sensors and applications have been introduced in Chapter 3. A typical modern car has fourteen or more different sensors integrated, and this number will increase when AVs hit the market. There have been several publications on AVs that are currently available for research [16, 46, 4, 22, 72]. Most of these vehicles use Laser Image Detection and Ranging (Lidar) and camera sensors. Lidar is a laser-based ranging system and camera vision plays an important role in many applications.

Potential attack scenarios and proof-of-concept attacks and have been discussed in Chapter 4, focusing on Lidar and camera systems. The MobilEye C2-270 Advanced Driver Assistance System (ADAS) was tested for light sensitivity, auto exposure attacks and blinding attacks. It was found that it was sensitive to near-infrared light, but not as sensitive as compared to a reference camera without any near-infrared filtering. The MobilEye C2-270 was sensitive to a 650 nm laser, but this is visible light source. The blinding attacks and auto exposure attacks using near-infrared light did work on the MobilEye C2-270, but it was not possible to fully blind the camera, which was possible with the reference camera. The correlation value was calculated between the tonal distributions of subsequent images. This metric was used to compare the performance of the light sources in the different setups, with the addition of measuring the blinding time in the auto exposure attacks experiment. The ibeo LUX 3 system was used for the attacks on a Lidar system. The experiments have shown that it is susceptible to reflective materials, jamming with a laser, relay attacks and spoofing attacks. All of the attacks have had influence on the perception, as signals could be injected. In general, none of the systems include tampering detection, as far as the experiments have concluded. To receive the signal and emit a counterfeit one, transceivers were used that were sensitive to light with a wavelength of 905 nm, and emit light with the same wavelength.

The attacks have been demonstrated on the hardware layer. In Chapter 5, the implications of the attacks have been demonstrated on the application layer, including a demonstration of what happens when an attack is di-

rected a sensor fusion algorithm such as a Particle Filter (PF). Furthermore, countermeasures and limitations of this work have also been discussed. A particular countermeasure for camera-based systems is to add redundancy, for the Lidar introducing unpredictability to the pulses will improve. The major limitation to this work is that it was conducted small-scale in a lab environment.

6.2 RESEARCH QUESTIONS

The research questions were introduced in Section 1.2. In this section, the questions will be answered based on the attacker model and attack types of Chapter 2, the survey on autonomous vehicle sensor in Chapter 3 and the experiments conducted in Chapter 4.

The actual experiments have only been conducted on a Lidar and camera system. Therefore, the answers will address these sensors only.

6.2.1 What types of attack can be mounted?

This work has demonstrated two attacks on the MobilEye C2-270 camera system and three attacks on the ibeo LUX 3 Lidar system. It is discussed that the list of attacks that have been demonstrated is not exhaustive.

The first attack on the MobilEye C2-270 was a blinding attack. In a blinding attack, an attacker wants to prevent the camera from observing the environment, either fully or partially. Although it can use visible light, using near-infrared light makes is less detectable for humans. The second attack influences the auto controls of a camera system. In order to optimize the light conditions, image sensors can automatically tune the exposure and gain. By influencing the controls, the camera adapts to a new situation that may not be optimal for the current environment. Most auto controls have an iterative tuning process, that takes time (can be in the order of seconds). Even if this attack is mounted at regular intervals, the camera may not perceive the scene for a longer period.

Three attacks have been demonstrated on the ibeo LUX 3 Lidar system. The first was a jamming attack, in which a similar but counterfeit signal was emitted in the direction of the ibeo LUX 3. This introduced a lot of noise. The second attack showed that the Lidar is sensitive to relay attacks. Its own signal could be received and emitted from another location in the direction of the ibeo LUX 3, believing that it was emitted from the original position. This attack was even possible from behind. Therefore, no line of sight is required. The last attack was the spoofing attack, where an original signal was spoofed. By controlling two parameters, the position of the counterfeit object and the number of copies could be controlled.

6.2.2 How likely are the attacks to happen and what are their consequences?

The attacks that have been demonstrated could be conducted with hardware that can be bought without any restrictions. This as opposed to attacking sensors such as Global Navigation Satellite System (GNSS) or Radio Detection and Ranging (Radar) that require a license for transmitting. That said, the attacker model considered for this work in Section 2.3 is not limited to any regulations that may apply, as it intentionally wants to either stop a vehicle or crash it.

To establish the likelihood of the attacks that can be mounted on Lidar and camera, four types of scenarios where attacks are plausible have been defined in Section 2.4.

- Front/rear/side attack
- Roadside attack
- Scenery attack
- Evil maid/Evil mechanic attack

The first three attack scenarios have in common that they are remote and do not require physical access to the car. The evil maid/evil mechanic requires short-term access to the outside of the car, so it can mount the hardware required for the attack onto a car. None of the attack scenarios require special locations. In Section 2.4, a distinction was made between low-speed and high-speed scenarios. For instance, a low-speed scenario happens in a city center, that involves more interaction with the environment (pedestrians, cyclists, vehicles). This as opposed to a high-speed scenarios, for instance on the highway. From an attacker point of view, in both scenarios a vehicle can cause damage to itself and/or the direct environment, potentially causing injuries or fatalities.

The best option to mount attack on the MobilEye C2-270 would be from a front/rear/side attack, as this gives the attacker the most time to direct a light source into a camera. Confusing the auto controls is limited to front/rear/side attack, because the attacker needs to emit a beam of light into a image sensors. This is much harder if the attacker is not dynamic. If the attacker can influence the camera, applications such as Collision Avoidance System (CAS) have been shown to fail in detecting approaching vehicles. While the MobilEye C2-270 is not sensitive enough to fully blind the camera using near-infrared light, it is still sensitive to it. This allows an attacker to either blind objects that should be detected, or spoof objects that are invisible to the human eye (e.g. matrix traffic signs).

All of the attacks mounted on the ibeo LUX 3 can be mounted from the roadside. Spoofing and jamming attacks can also be mounted from the front/rear/side. Because the hardware for a jamming attack can be very compact, it can even be installed by a evil mechanic. It has been demonstrated that the Lidar will detect, classify and track objects that have been spoofed. For CAS, the AV could be tricked into hitting the brakes by introducing object spontaneously. Lastly, the tracking software has a limit on the number of objects it can track. It is debated that spoofing or jamming attacks could mount a denial-of-service attack by inserting a large number of objects. This way, objects that really should be tracked could be missed.

6.2.3 What is the amount of effort that has to be put into the attacks, in terms of time and money?

The amount of time required can be split into attack preparation and mounting the attack. Attack preparation depends on knowledge of the attacker. This work has shown that it is possible to attack camera and Lidar systems without prior knowledge of the systems, in a timespan of six month. This includes reverse engineering the operation of the hardware. The time required to mount an attack is negligible. With the right preparation, the attacker can mount an attack from the mentioned scenarios that do not require special access. This fits the definition of an attacker with limited time.

The attacker model also considered limited money. The attacks on the MobilEye C2-270 have shown to work best with a 650 nm laser. Unfortunately, this is a visible light source and this will be quickly detected by a human. The second best option is the 940 nm near-infrared LEDs. While the 850 nm have been proven to be more effective on the image sensor, the costs outweighs the effect by a factor eight. Combining multiple LEDs to form a single beam has proven to be effective.

It is shown that the jamming and relay attacks are less sophisticated than spoofing attacks. Jamming and relay attacks do no require synchronization with the original signal. Therefore they are easier to mount. The most-expensive part of the transceivers that have been used to jam and spoof the signal, is an Osram SPL-PL90 laser diode. This laser diode costs 43.25 dollar. The photodetector is an Osram SFH-213 and costs 0.65 dollar. Although the experiments have made use of pulse generators and an oscilloscope, these devices are not required to mount an attack. The pulse generators can be replaced with dedicated hardware such as a Field-Programmable Gate Array (FPGA), Digital Signal Processor (DSP) or Microcontroller Unit (MCU). This can make the attack battery powered and less detectable. This hardware can be bought for less than 20 dollar. The range of the laser diodes is large enough, therefore jamming attacks are the easiest to mount. If an attacker designs one attack and shares it (e.g. via internet), the only thing another attacker has to do is buy the hardware connect the wires. Spoofing attacks will require the attacker to catch the signal, which may be problematic at a distance of 100 meter. This is caused by the gap of 1.47 meter between scan steps[1].

The overall objective of this work was to find out if sensors can be influenced remotely, in such a way that the sensor either breaks or reports invalid information, with the intention to crash or stop a vehicle. Based on the achieved results, the conclusion is that Lidar and camera sensors can be influenced in such a way that they report invalid information. If the vehicle will crash is currently more likely than that it will stop. The latter assumes that a sensors can detect malicious input, but both sensors did not sound an alarm when they received malicious input.

6.3 FUTURE WORK

There are many things future experiments can improve upon this work. With this work, the problem of the lack of sensor security is raised. Attacks have been demonstrated using representative hardware, but it would be highly recommended to study the effect of these attacks on different hardware. In particular, similar hardware is interesting with respect to black-box testing. It can help expose knowledge on countermeasures that could not be

1 Refer to Section 4.2.2 for the calculations

detected during this work. For instance, if a camera system with the same image sensors as the MobilEye C2-270 shows it is sensitive to near-infrared light, it is likely that the MobilEye C2-270 includes any light filtering in the lens system.

6.3.1 Camera

Three additional camera-related attacks have not been studied in this work. The first is using high-power lasers to damage image sensors and study the feasibility and effects. It is interesting to know if an attacker can damage an image sensor similar to a MobilEye C2-270 from a large distance. This experiment was not possible due to lab safety regulations.

The second attack addresses hidden traffic sign, such as matrix boards based on near-infrared lights. A human (driver) will not see this, but as the experiments in Section 4.1 have shown, cameras will see this. It is believed that applications involving color thresholding or Haar-like classifiers will detect these fake traffic signs, as explained in Section 3.1.3. The MobilEye C2-270 does not include traffic sign recognition. Therefore, this attack could not be tested. The last attack would 'transform' the image, such as rotating it with mirrors [59] or stretching it with Fresnel lenses. Some camera applications of Section 3.1.3 assume that objects of interest are located in a specific region of the camera. By rotating or stretching, these assumptions will not hold and the applications will fail to detect objects. This attack fits best in the definition of an evil mechanic.

6.3.2 Lidar

There are several parts of the experiments that can be improved. In Section 4.2.5 hardware and cable delays were discussed. To lessen the effect, dedicated designs consisting of hardware such as a FPGA, a DSP or a MCU can add to the results of this thesis and make it possible to spoof and jam on shorter ranges.

In the lab experiments, only one Lidar system was used. In practice, multiple Lidar systems will be used at the same time at the same location. This will make synchronization harder if the transceiver pickup multiple signals. A attacker has to distinguish between the origin of a signal. Future work should investigate this issue.

The ibeo LUX 3 is a multi-layer Lidar that is not able to produce a three-dimensional view. Although it is likely the same techniques apply to make a three-dimensional image, there are no results on this.

The experiments have shown that it is possible to jam or spoof signals and have shown what it requires, but have not managed to inject an object from scratch. The only objects spoofed were copies of the 'original' observation. Future work could be conducted on spoofing objects that, for instance, can be loaded from a file. The spoofing attacks have indicated that it is possible to influence a single scan point of the Lidar. Dedicated hardware will be required that can control the laser at a decent pace. Because of the wide viewing angle of the laser diode in the transceiver, it is believed that a single laser is sufficient.

6.3.3 Application level

In Section 5.1, three examples of the impact on application level have been presented. These examples have not been thoroughly analyzed, because the details on the internals of the MobilEye C2-270 and ibeo LUX 3 are unknown. Furthermore, sensor fusion is a topic on its own.

Future work could look at anomaly detection using sensor fusion. For instance, the Kalman Filter (KF) utilizes a mathematical model to validate if a state change is plausible, e.g. how a Global Positioning System (GPS) receiver can use Inertial Measurement Unit (IMU) data to conclude it has not moved. Another interesting direction to consider is machine learning algorithms, especially if machine learning algorithms can identify malicious input, or if they can be influenced too.

6.3.4 Countermeasures

All of the countermeasures presented in Section 5.2 are also considered future work. The experiments were only limited to performing the attacks, therefore no time was invested in testing the countermeasures. The following countermeasures have been proposed:

- Increase redundancy by adding cameras to overlap fully or partially.
- Limit the effects of high-intensity light sources on image sensors via certain optics and materials.
- Detect jamming attacks on cameras via spectral analysis.
- Use multiple lasers with non-overlapping wavelengths to add redundancy to the Lidar.
- Split image into separate channels to detect single-wavelength attacks.
- Introduce random probing to detect jamming attacks on Lidar.
- Probe multiple times with a Lidar to raise the confidence in a measurement.
- Shorten the pulse period by limiting the maximum range of a Lidar.
- Increase the tracking limits of the Lidar.

A SENSOR FUSION: A CASE STUDY

The Kalman Filter (KF) and Particle Filter (PF) have been discussed in Section 3.2. To provide better understanding of the math that is involved, a case study is presented in this appendix. In this case study, based on the introduction of [90], both filters are applied to a cannon ball in flight. The KF will 'smooth' its trajectory, while the PF is used to track it is trajectory.

To simulate a cannon ball in flight, a simplified kinematic model is used. With this discrete model, presented in equation 27, the position x and y of the cannon ball can be determined for every time step t. The constant $g = 9.81 m/s^2$ represents the gravitational force on earth.

$$\begin{aligned} x(t) &= x_0 + V_{0x}t \\ V_x(t) &= V_{0x} \\ y(t) &= y_0 + V_{0y}t - \frac{1}{2}gt^2 \\ V_y(t) &= V_{0y} - gt \end{aligned} \tag{27}$$

Note that this is a very simple model, and does not involve any other forces that influence the cannon ball. If the model would involve the other forces, it will be more accurate. But event a very simple model give pretty good estimates, and therefore correct the measurement errors very well.

A.1 KALMAN FILTER

In this simulation, it is assumed that the cannon ball will have a position sensor that can register is X and Y position at reasonable pace. The model presented above is used, with the addition of noise to the X and Y position of the ball. This results in differences in the actual position and the registered position. The KF should reduce these difference as much as possible. In figure 75, the model is plotted, together with the output of the KF.

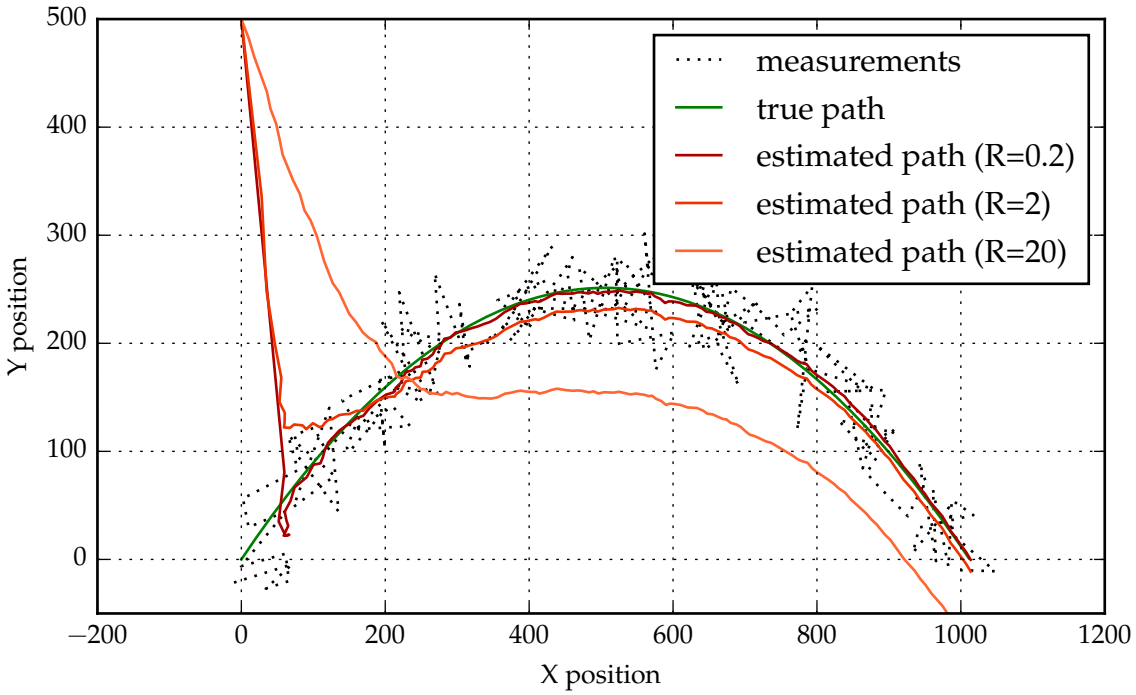


Fig. 75: The KF applied to a simulation of a cannonball in flight, with different values for the measurement error covariance matrix **R**.

The continuous kinetic model can be converted into the following discrete matrix notation, which almost looks similar to equation 6. The time step is indicated by Δt, and is different from the iteration count, but they are related (e.g. each iteration is exactly one second).

$$\begin{bmatrix} x_i \\ V_{xi} \\ y_i \\ V_{yi} \end{bmatrix} = \begin{bmatrix} x_{i-1} + V_{xi-1}\Delta t \\ V_{xi-1} \\ y_{i-1} + V_{yi-1}\Delta t \\ V_{yi-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{2}g\Delta t^2 \\ -g\Delta t \end{bmatrix} \tag{28}$$

Fully converted, it yields equation 13. The vector u_i remains constant in this model, but could hold different values for each iteration. From the equation, it can be shown how x_i and y_i are based on V_{xi} and V_{yi}.

$$\hat{x} = \mathbf{A}x_{i-1} + \mathbf{B}u_i$$

$$\begin{bmatrix} x_i \\ V_{xi} \\ y_i \\ V_{yi} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} x_{i-1} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{2}g\Delta t^2 \\ -g\Delta t \end{bmatrix} \tag{29}$$

With **A**, **B** and u_i specified, the other three parameters can be defined. The observation matrix **H** converts the measurements to the predicted state, and can be used for 'preprocessing' the results. Since the measurement values map directly to the state, no conversion is required. This map can also be used to 'disable a sensor temporarily', i.e. when a sensor reading is not available. Matrix **Q**, will be the process error covariance. This could be the error covariance due to the model not being completely accurate. Since the equations are directly taken from the kinetic model, the matrix is completely zero. Lastly, the measurement error covariance matrix **R** defines the measurement error. For this case study, it is chosen arbitrary, but in practice, it would be provided by the manufacturer of the sensor. As an alternative,

a method such as Autocovariance Least-Squares (ALS) can be used to estimate the covariance matrix [115, 3]. To give an example of the influence of the covariance matrix $\mathbf{R}$, see figure 75. Higher values allow more 'flexibility', but increase the error.

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{30}$$

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{31}$$

$$\mathbf{R} = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix} \tag{32}$$

Before the first iteration, initial values should be provided. Matrix $\mathbf{P}_0$ is the initial guess for the covariance of the state. It is outside the scope of this work on how to determine the right values for $\mathbf{P}$. The state vector x_0 contains the initial values. Note that, $y_0 = 500$, which is far from an accurate estimation of the Y position. This is done on purpose, to show how quickly the algorithm converges.

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{33}$$

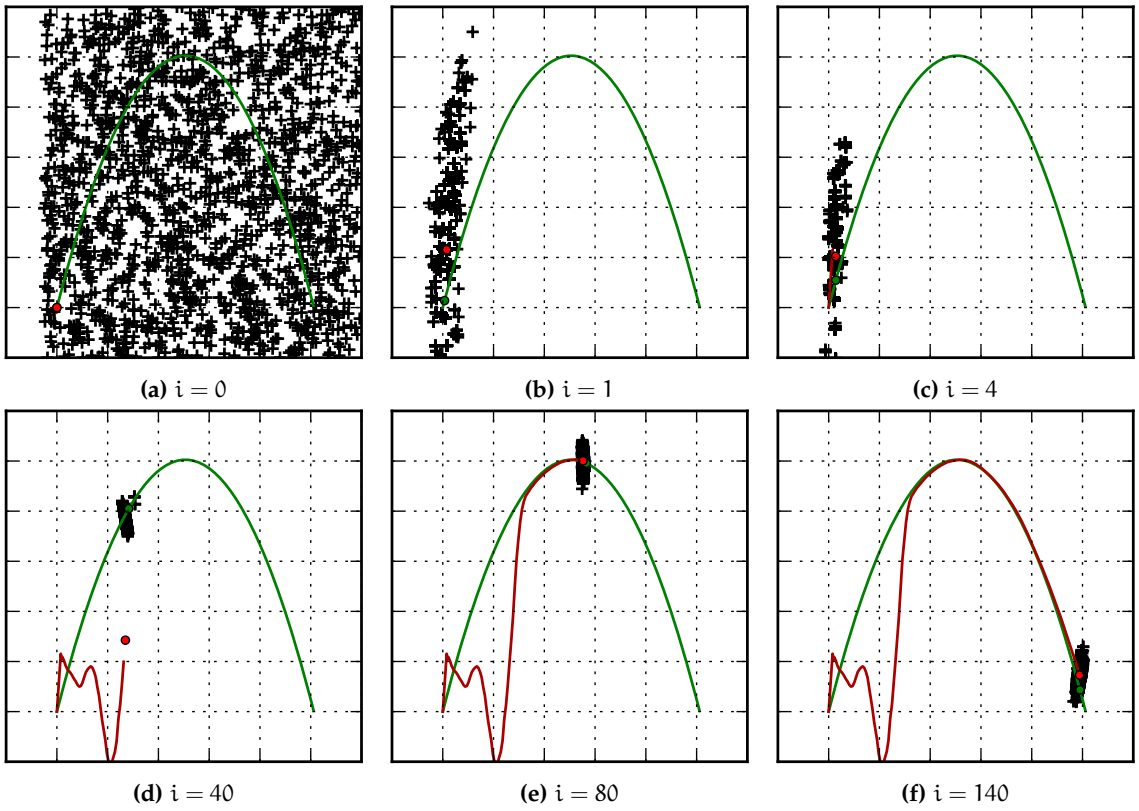
$$\mathbf{x}_0 = \begin{bmatrix} 0 \\ 100 * \cos(\pi/4) \\ 500 \\ 100 * \sin(\pi/4) \end{bmatrix} \tag{34}$$

With the definitions presented above, the KF is complete, and can be iterated. Figure 75 shows the output of the filter, applied on the noisy measurements.

A.2 PARTICLE FILTER

In this part of the case study, the PF is used to track the cannonball's position. The cannonball can sense it position to three 'beacons' in a two-dimensional world: one at the start, one at the end, and one at the peak. The exact positions do not really matter (refer to Section 3.2.2). A measurement vector does not only contain the distance to the three beacons, it will also include a direction. While the direction of the cannonball and particles are not drawn in the figures, they play an important role because the cannonball and particles move relative to the previous movement. Therefore, if an particle points 180 degrees the other way, it will move away from the cannonball.

Figure 76 shows six graphs of several iterations. At $i = 0$, 1000 particles are spread around in the predefined world, at random positions. These particles cluster very quick, as shown for $i = 1, 4$. As mentioned above, there are only three ground beacons. This is not an ideal situation, because the absolute distance to the three points, as observed by the cannonball, can be ambiguous since the measurement can be projected onto the other side ('in the ground'). Section 3.2.2 illustrates this. While not drawn for $i = 40$, there exists another cluster that is mirrored and therefore, the red dot (mean position) is not on the modeled trajectory. It is not until $i = 80, 140$ when the other cluster dies out, and the mean position follows the trajectory. Important to notice: due to the randomness of the particles and the resampling, multiple runs of this simulation could lead to different results.



(a) $i = 0$ **(b)** $i = 1$ **(c)** $i = 4$

(d) $i = 40$ **(e)** $i = 80$ **(f)** $i = 140$

Fig. 76: Six iterations of the PF applied to track the cannonball in flight. The red dot is the weighted mean position, the green dot is the actual position of the cannonball.

The final tracked path is shown in Figure 77. It is a little less smooth compared to the KF in Figure 75. This can be partially explained by the fact that the mean position of all the particles is considered to be the estimated position of the cannonball.

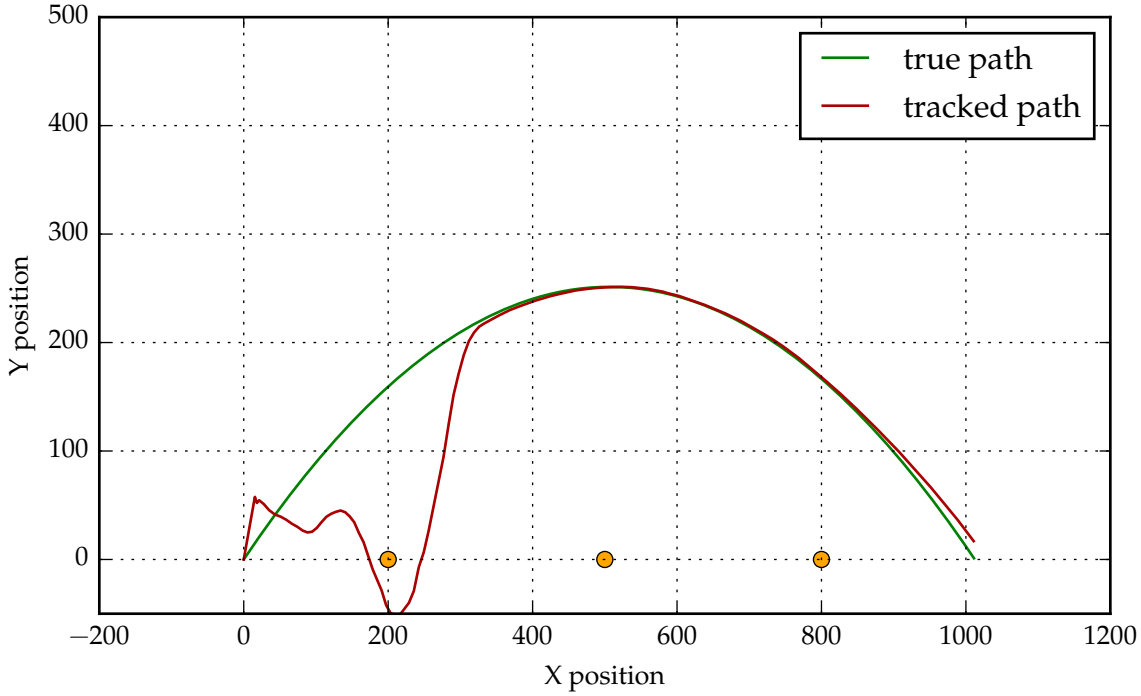


Fig. 77: The PF applied to track a cannonball in flight. The orange dots are the beacons on the ground.

B SPECTROMETRY

This appendix describes the spectrometry experiment. In this experiment, the light sources used throughout this work have been tested for their wavelengths emitted. For the camera related experiments in Section 4.1, this experiment helped to see if light sources overlapped, as a single LED does not emit light of exactly one wavelength. Regarding the Laser Image Detection and Ranging (Lidar) experiments in Section 4.2, it was needed to see if the ibeo LUX 3 emitted on the same wavelength as the transceivers.

In addition, the AvaSpec-2048 USB spectrometer was used to generate a spectrometry plot of the light sources. The same setup as in Section 4.1.2 (in particular Figure 29) was used, but the camera was replaced with a spectrometer[1]. In this plot, the Y-axis is measured in counts. This is the return value of the analog-digital convert of the Charge-Coupled Device (CCD) chip inside the spectrometer.

For all measurements, an integration time[2] of 18 ms, 28 averages and 1 smoothing were chosen. These values do not really matter for a relative measurement, as long as the same values are chosen for each measurement. Furthermore, the same dark measurement was loaded before each measurement, for calibration purposes.

1 Except for the 880 nm LED and the 905 nm laser. They were moved closer to yield an acceptable amount of light

2 Comparable to shutter speed: higher values will accumulate more light

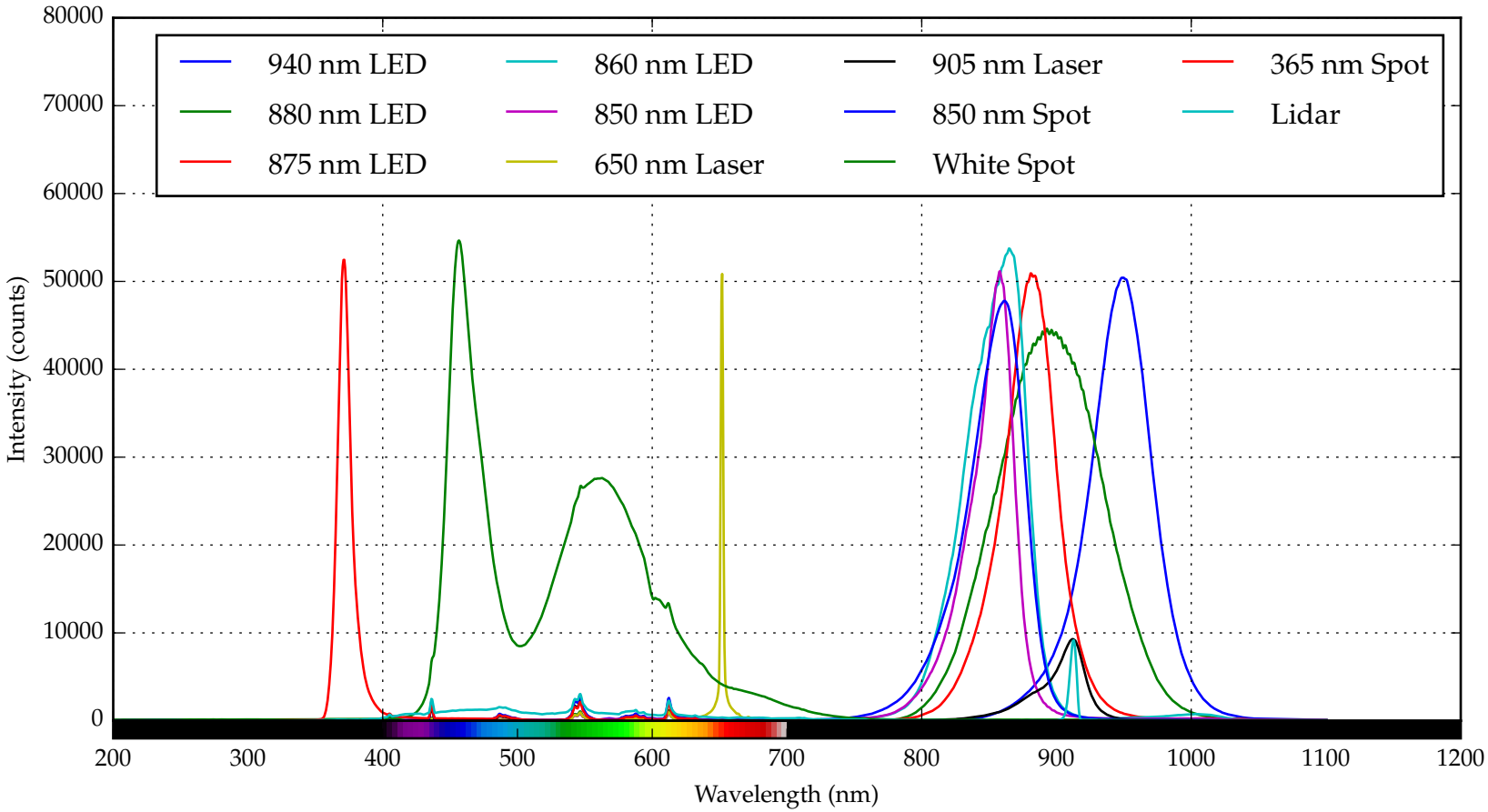


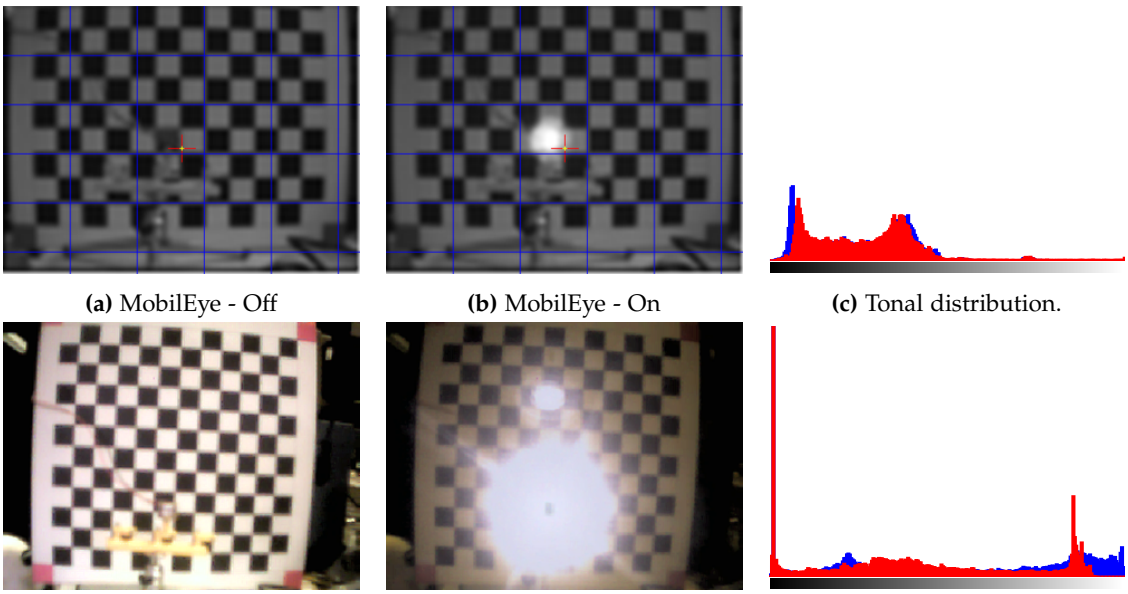
Fig. 78: Spectrometry of the light sources mentioned above. Note that the 5x5 matrix is not included, since it uses the same LEDs as the normal 940 nm LEDs.

The 905 nm laser yields a small number of counts, even though it is the most powerful laser. This comes from the fact that it is a switched laser, one that does not emit continuously. The 650 nm laser is continuously, and yields a small spike on the graph. The small peaks at 450 nm, 550 nm and 625 nm are the reflections of the fluorescent light in the lab. They can be ignored.

C RESULTS OF CAMERA EXPERIMENTS

C.1 TESTING SENSITIVITY

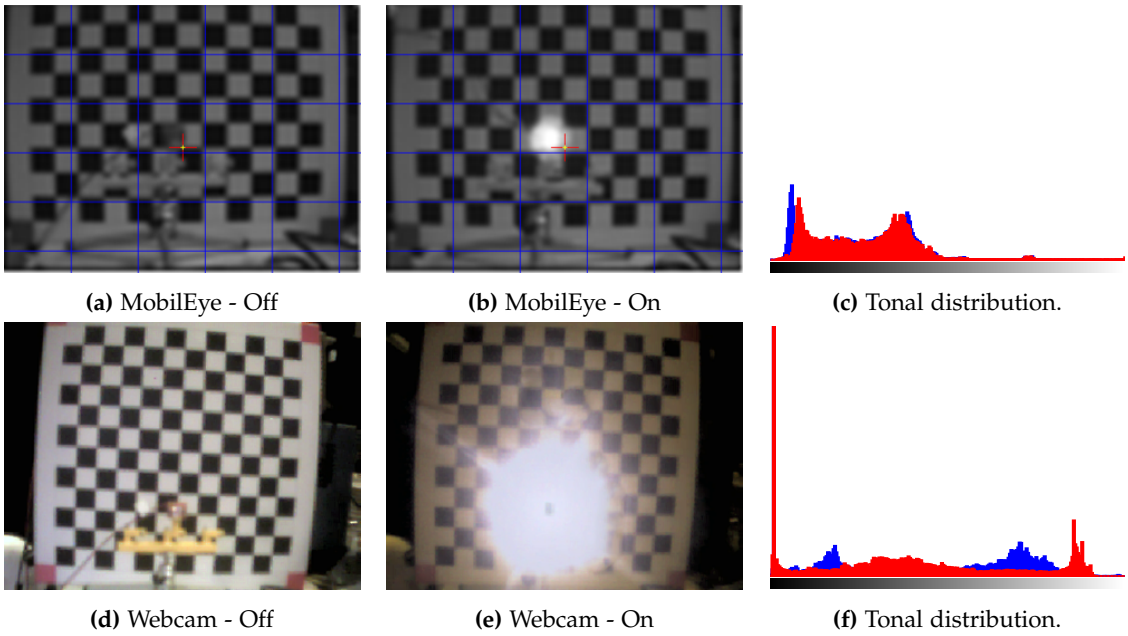
Refer to Section 4.1.2 for an explanation and interpretation of the results. In short, each group of six images consists of two images for the MobilEye C2-270, two images for the reference webcam and two tonal distributions (blue is off image, red is on image).



(a) MobilEye - Off **(b)** MobilEye - On **(c)** Tonal distribution.

(d) Webcam - Off **(e)** Webcam - On **(f)** Tonal distribution.

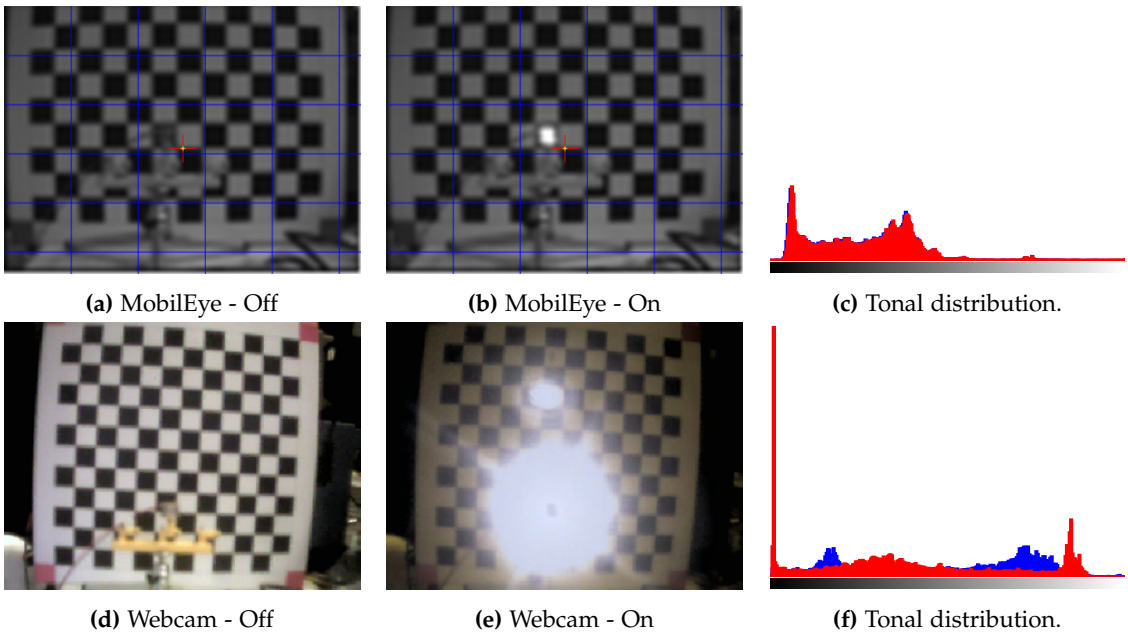
Fig. 79: 850 nm LED @ 50 cm. Blue is off-state, red is on-state.



(a) MobilEye - Off **(b)** MobilEye - On **(c)** Tonal distribution.

(d) Webcam - Off **(e)** Webcam - On **(f)** Tonal distribution.

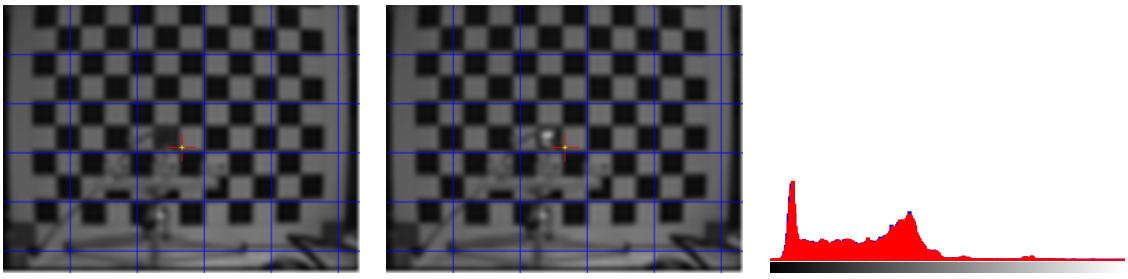
Fig. 80: 860 nm LED @ 50 cm. Blue is off-state, red is on-state.



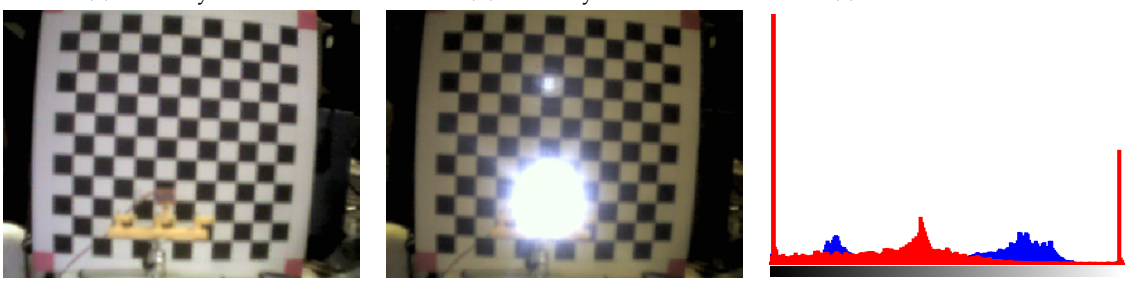
(a) MobilEye - Off **(b)** MobilEye - On **(c)** Tonal distribution.

(d) Webcam - Off **(e)** Webcam - On **(f)** Tonal distribution.

Fig. 81: 875 nm LED @ 50 cm. Blue is off-state, red is on-state.

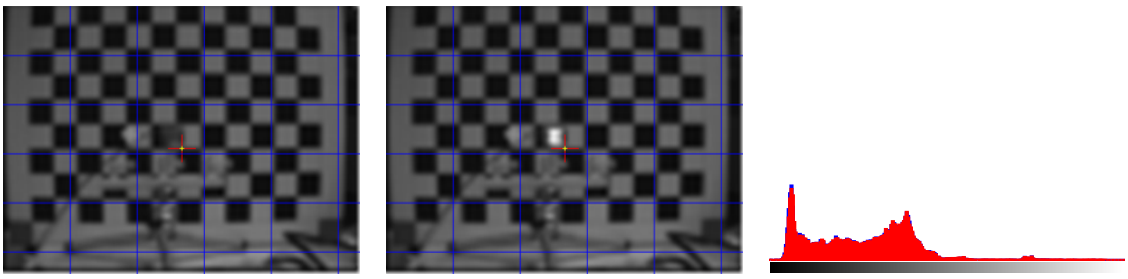


(a) MobilEye - Off **(b)** MobilEye - On **(c)** Tonal distribution.

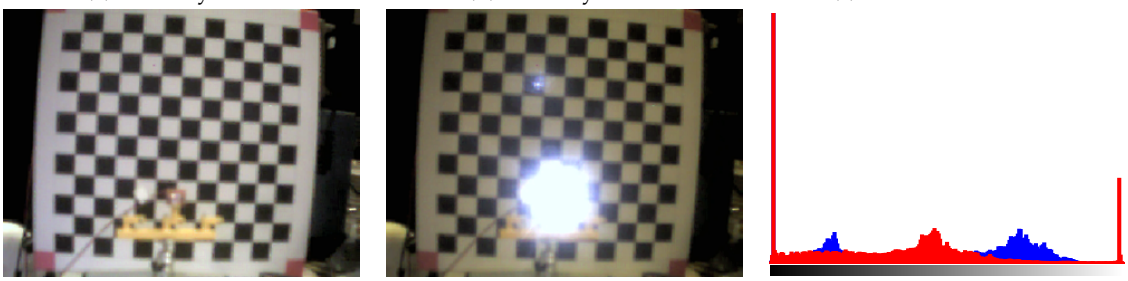


(d) Webcam - Off **(e)** Webcam - On **(f)** Tonal distribution.

Fig. 82: 880 nm LED @ 50 cm. Blue is off-state, red is on-state.

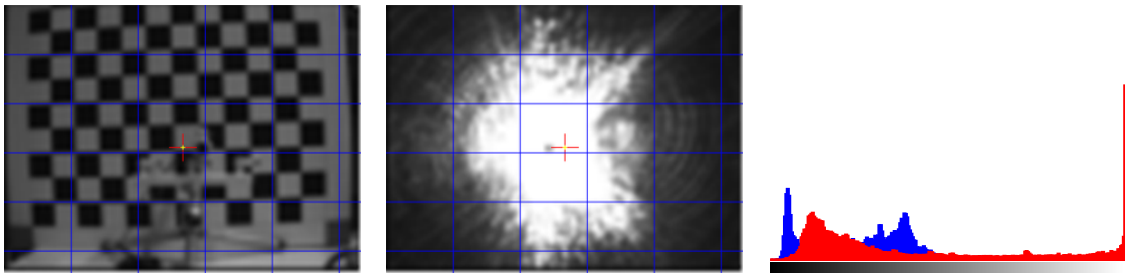


(a) MobilEye - Off **(b)** MobilEye - On **(c)** Tonal distribution.

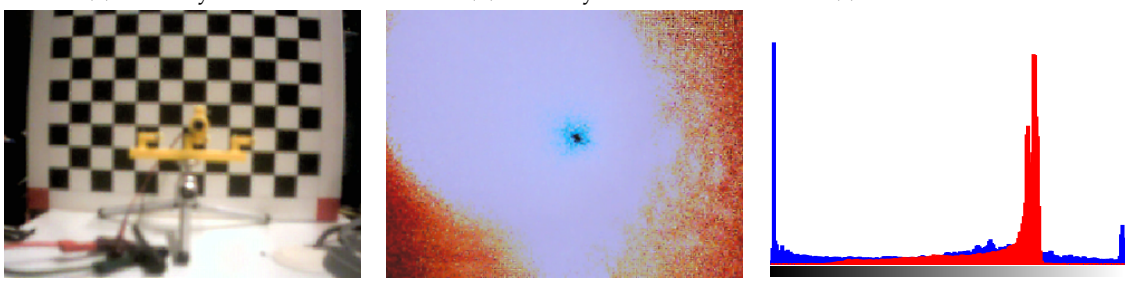


(d) Webcam - Off **(e)** Webcam - On **(f)** Tonal distribution.

Fig. 83: 940 nm LED @ 50 cm. Blue is off-state, red is on-state.

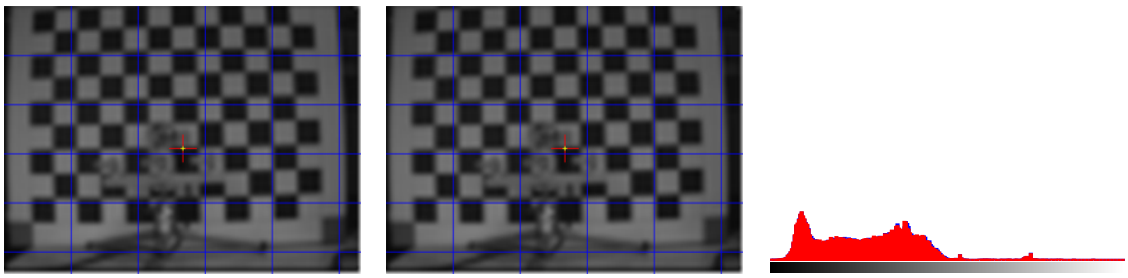


(a) MobilEye - Off **(b)** MobilEye - On **(c)** Tonal distribution.

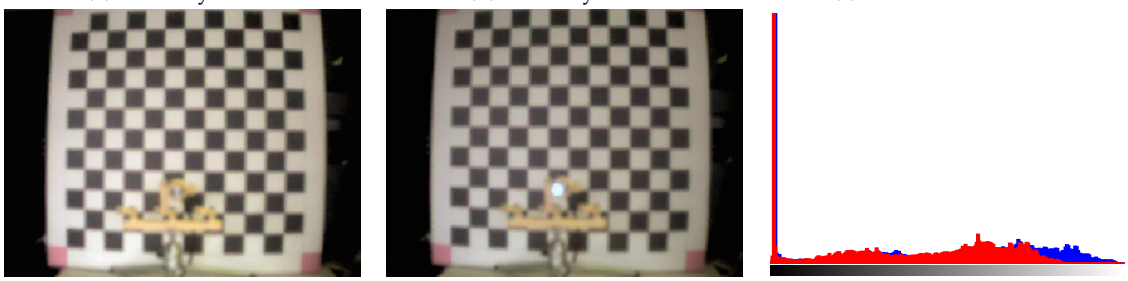


(d) Webcam - Off **(e)** Webcam - On **(f)** Tonal distribution.

Fig. 84: 650 nm laser @ 50 cm. Blue is off-state, red is on-state.

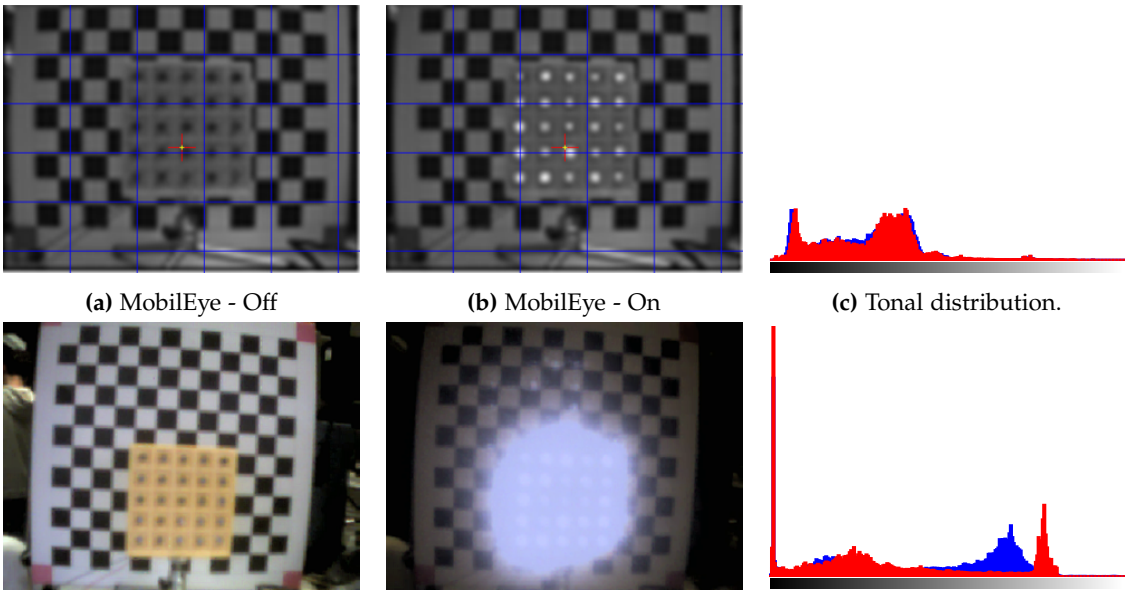


(a) MobilEye - Off **(b)** MobilEye - On **(c)** Tonal distribution.



(d) Webcam - Off **(e)** Webcam - On **(f)** Tonal distribution.

Fig. 85: 905 nm laser @ 50 cm. Blue is off-state, red is on-state.



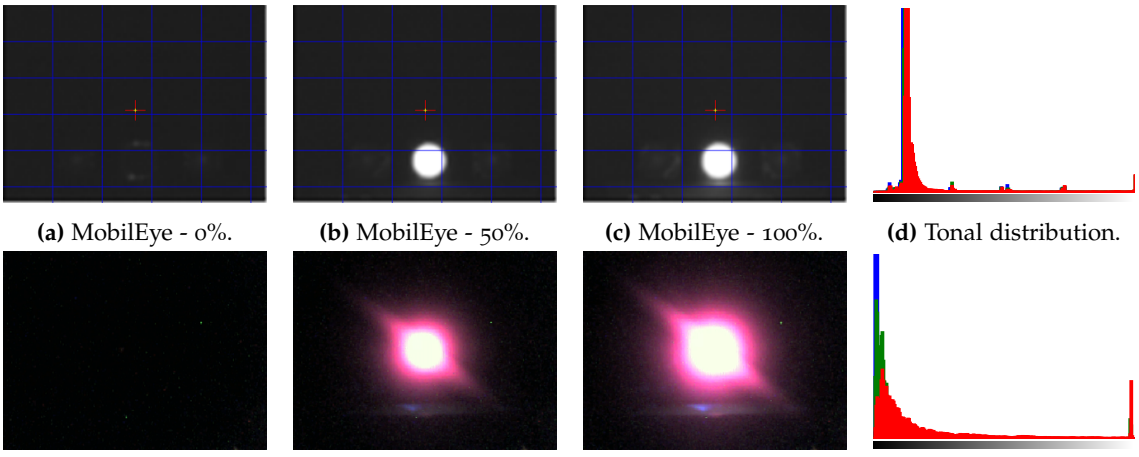
(a) MobilEye - Off **(b)** MobilEye - On **(c)** Tonal distribution.

(d) Webcam - Off **(e)** Webcam - On **(f)** Tonal distribution.

Fig. 86: 940 nm 5x5 LED matrix @ 50 cm. Blue is off-state, red is on-state.

C.2 BLINDING THE CAMERA

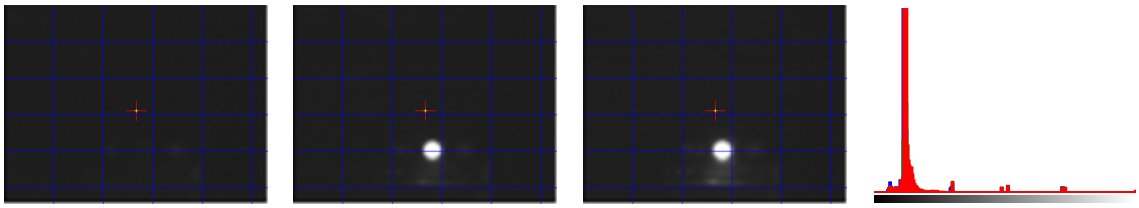
Refer to Section 4.1.3 for an explanation and interpretation of the results. In short, each group of eight images consists of three images for the MobilEye C2-270, three images for the reference webcam and two tonal distributions (blue is 0%, green is 50% and red is 100%).



(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

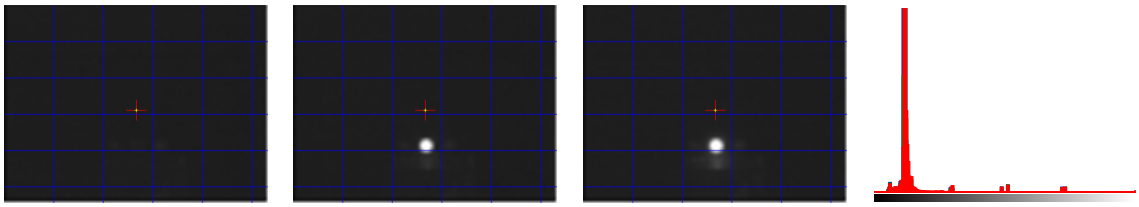
Fig. 87: 365 nm spot in dark @ 50 cm. Blue is 0%, green is 50%, red is 100%.



(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

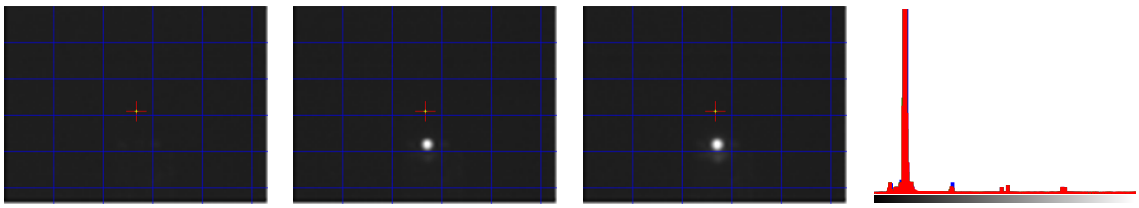
Fig. 88: 365 nm spot in dark @ 100 cm. Blue is 0%, green is 50%, red is 100%.



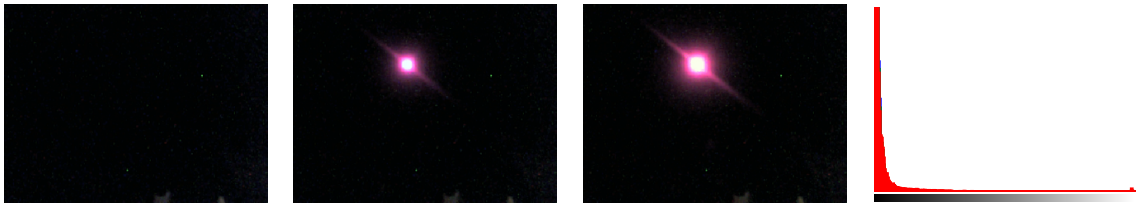
(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 89: 365 nm spot in dark @ 150 cm. Blue is 0%, green is 50%, red is 100%.

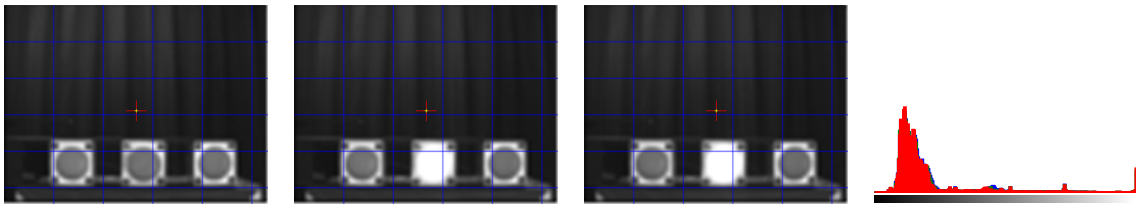


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

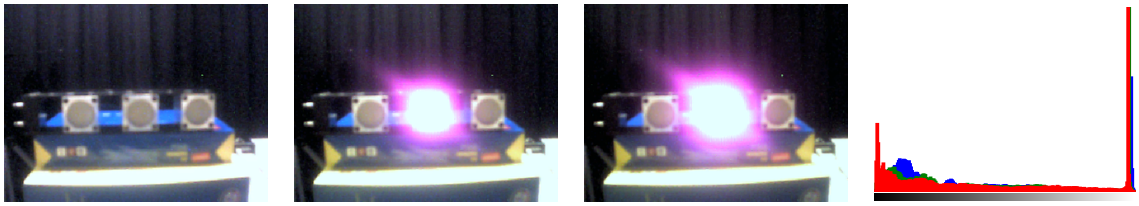


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 90: 365 nm spot in dark @ 200 cm. Blue is 0%, green is 50%, red is 100%.

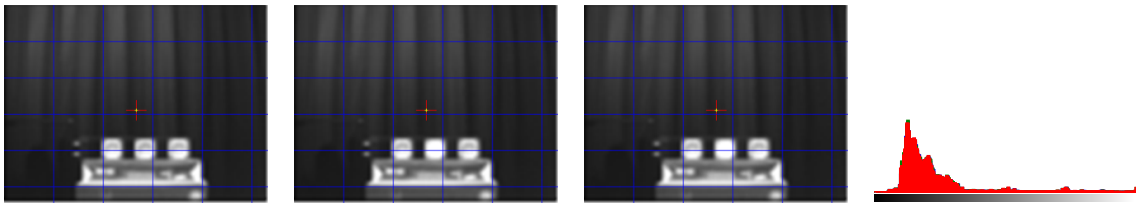


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

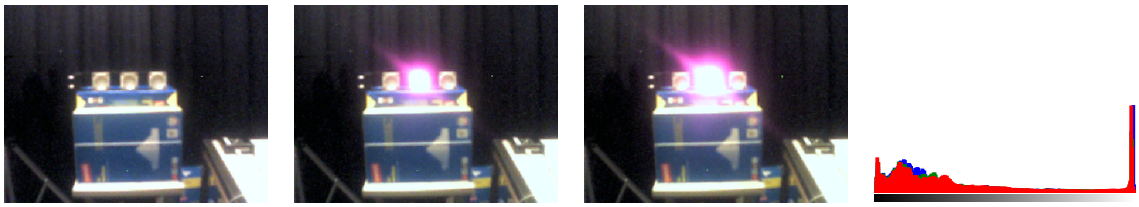


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 91: 365 nm spot in light @ 50 cm. Blue is 0%, green is 50%, red is 100%.

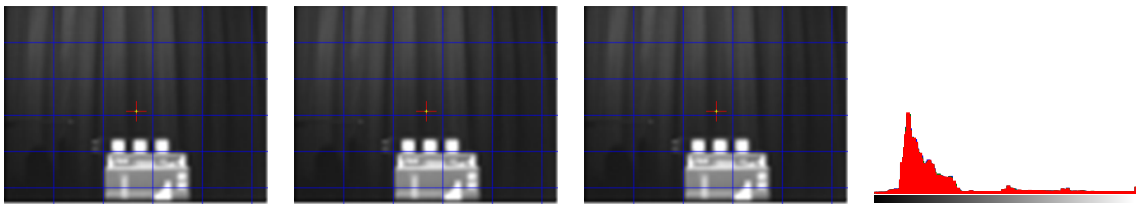


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

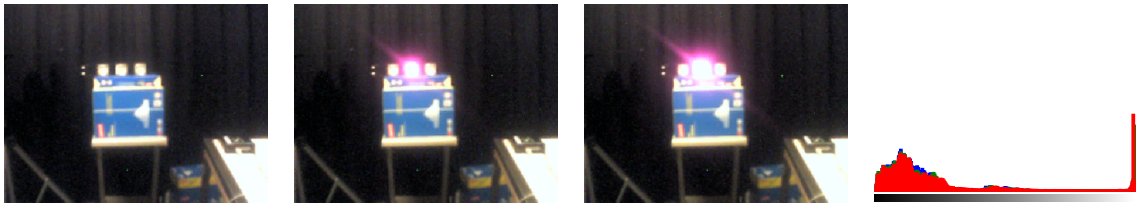


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 92: 365 nm spot in light @ 100 cm. Blue is 0%, green is 50%, red is 100%.

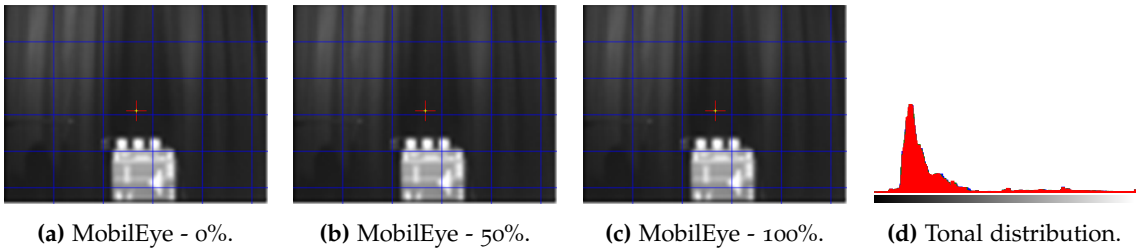


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

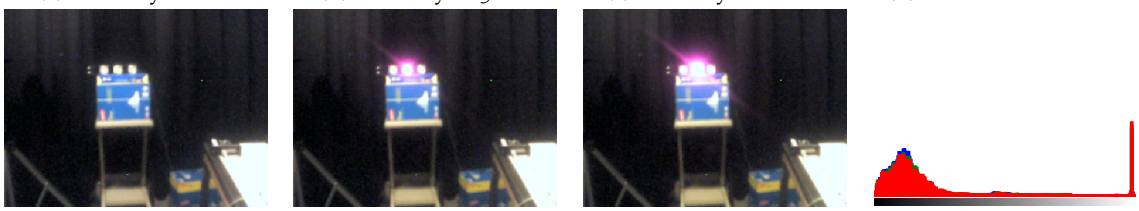


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 93: 365 nm spot in light @ 150 cm. Blue is 0%, green is 50%, red is 100%.

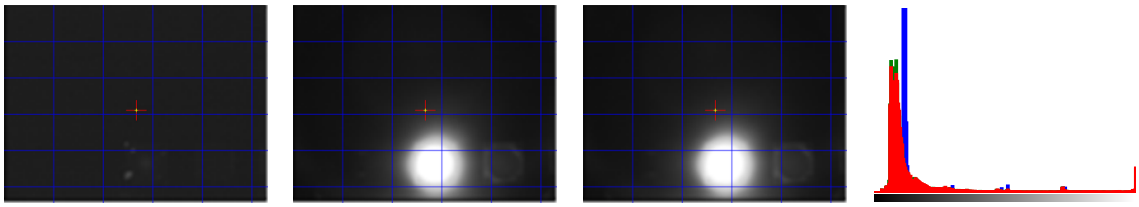


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

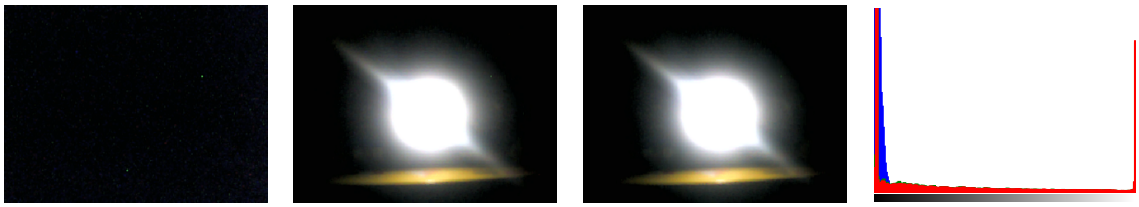


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 94: 365 nm spot in light @ 200 cm. Blue is 0%, green is 50%, red is 100%.

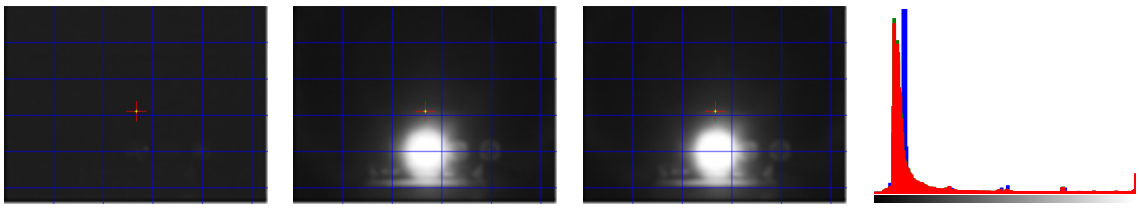


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.



(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

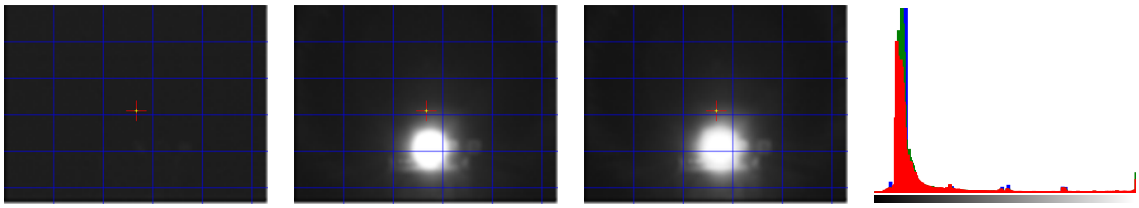
Fig. 95: White spot in dark @ 50 cm. Blue is 0%, green is 50%, red is 100%.



(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

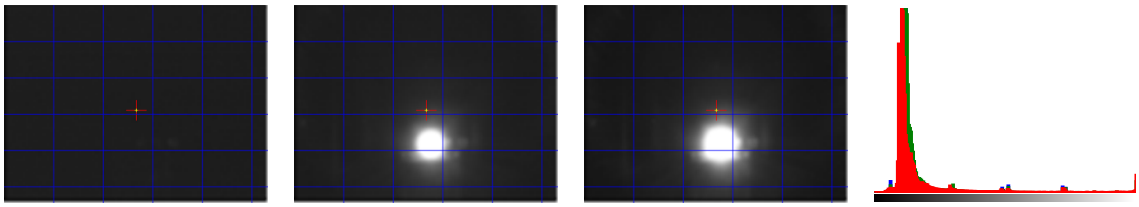
Fig. 96: White spot in dark @ 100 cm. Blue is 0%, green is 50%, red is 100%.



(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

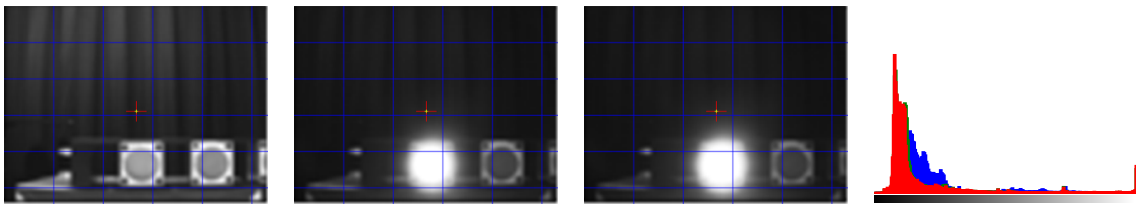
Fig. 97: White spot in dark @ 150 cm. Blue is 0%, green is 50%, red is 100%.



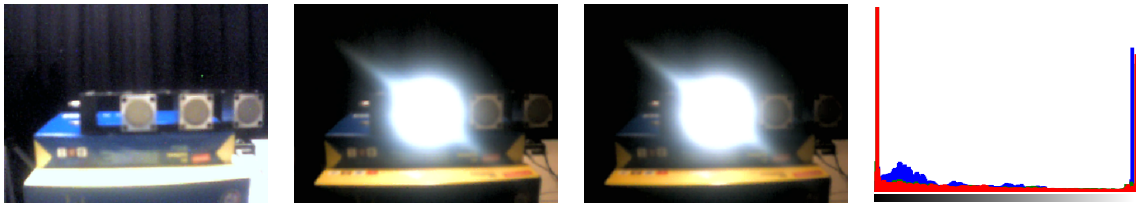
(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 98: White spot in dark @ 200 cm. Blue is 0%, green is 50%, red is 100%.

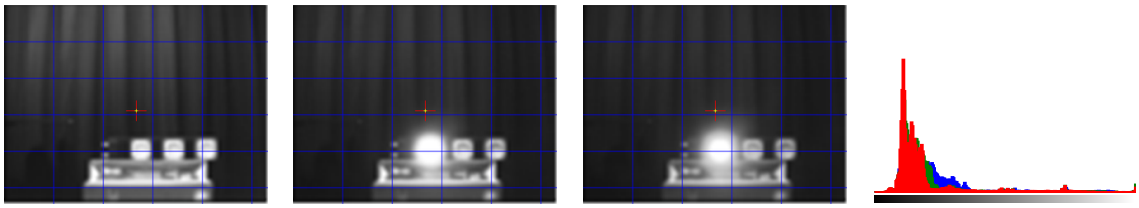


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

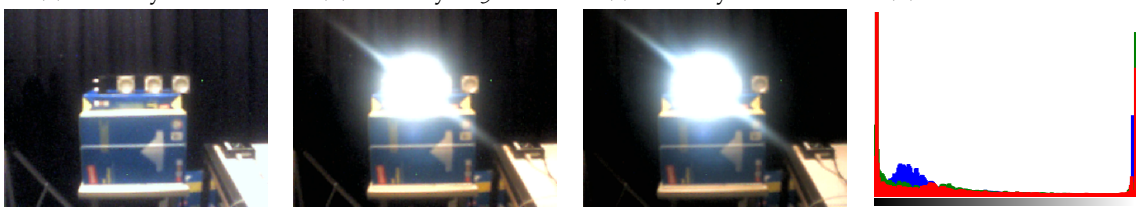


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 99: White spot in light @ 50 cm. Blue is 0%, green is 50%, red is 100%.

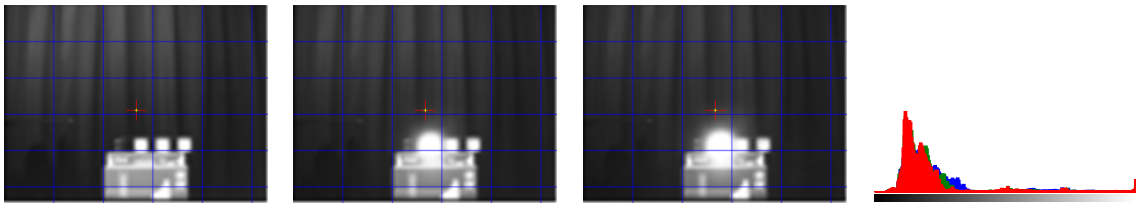


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

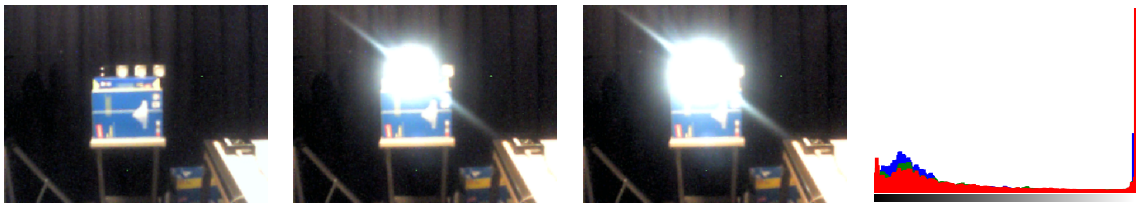


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 100: White spot in light @ 100 cm. Blue is 0%, green is 50%, red is 100%.

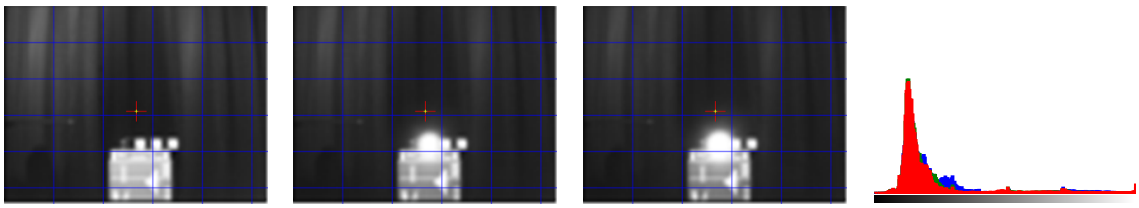


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

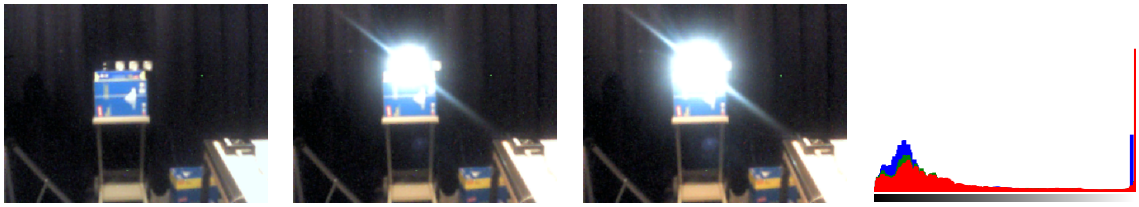


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 101: White spot in light @ 150 cm. Blue is 0%, green is 50%, red is 100%.

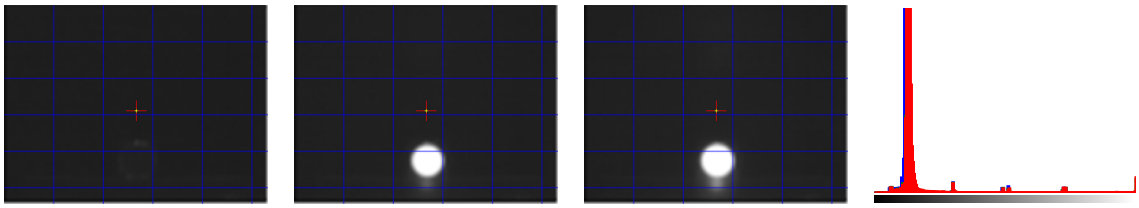


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

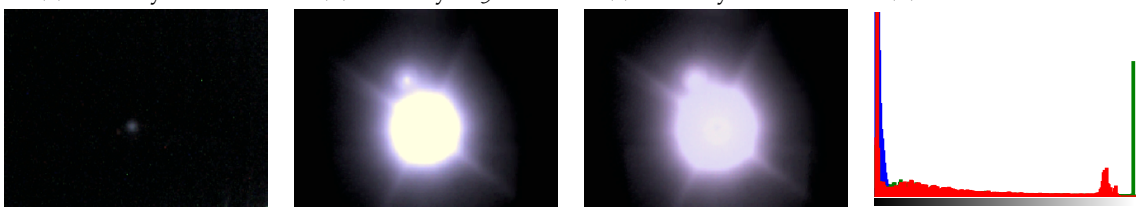


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 102: White spot in light @ 200 cm. Blue is 0%, green is 50%, red is 100%.

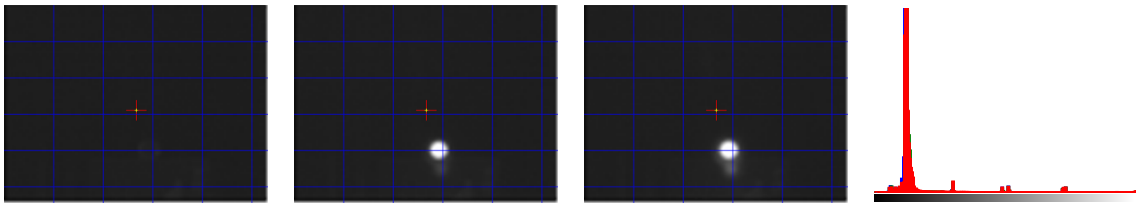


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

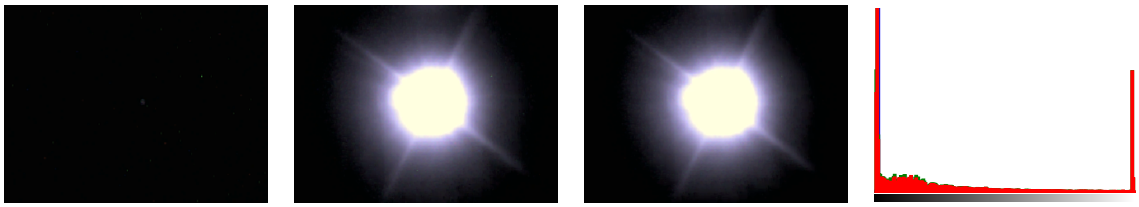


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 103: 850 nm spot in dark @ 50 cm. Blue is 0%, green is 50%, red is 100%.

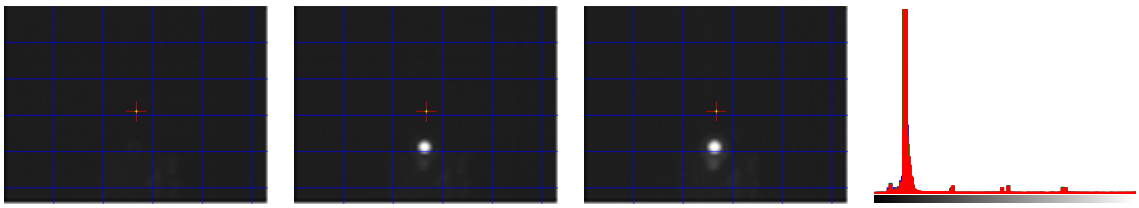


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

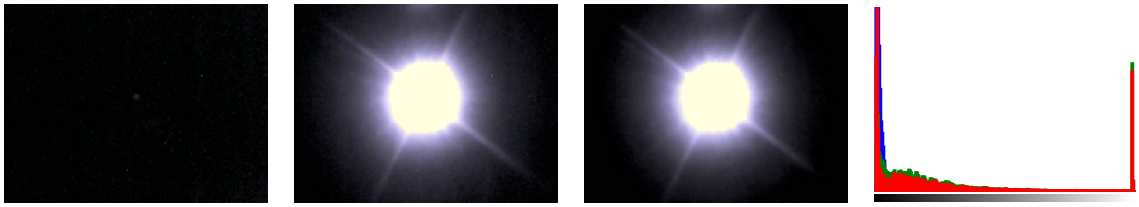


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 104: 850 nm spot in dark @ 100 cm. Blue is 0%, green is 50%, red is 100%.

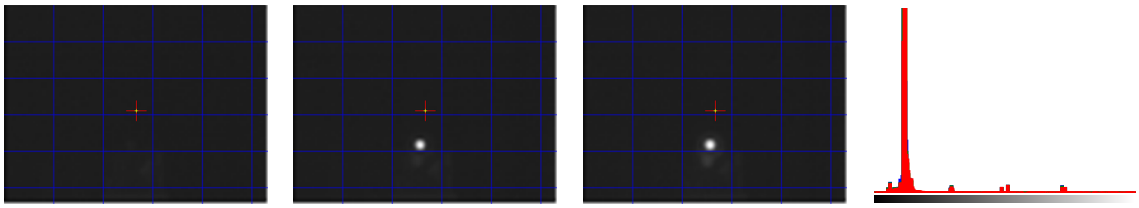


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

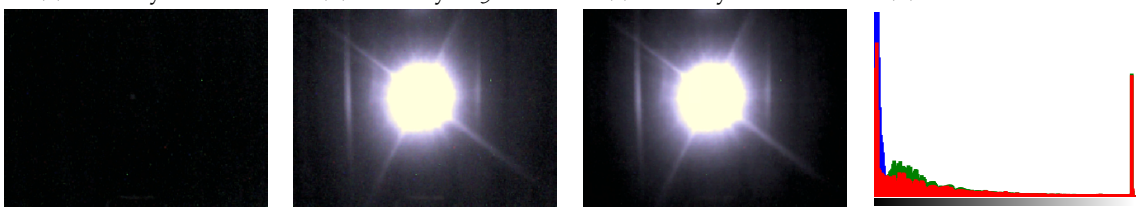


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 105: 850 nm spot in dark @ 150 cm. Blue is 0%, green is 50%, red is 100%.

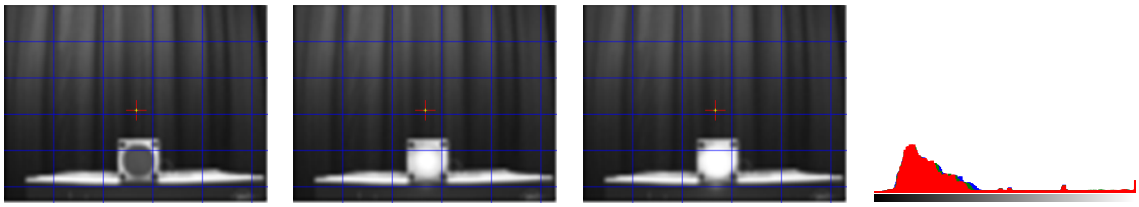


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

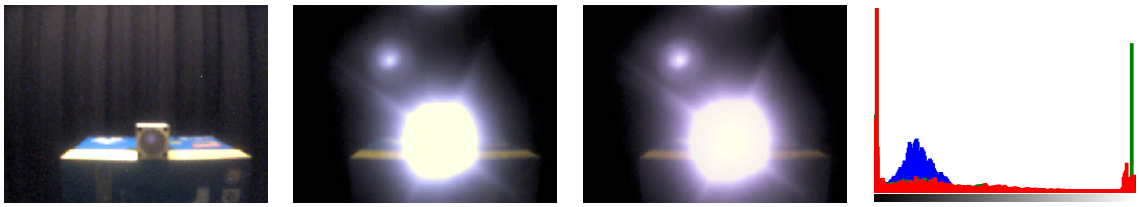


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 106: 850 nm spot in dark @ 200 cm. Blue is 0%, green is 50%, red is 100%.

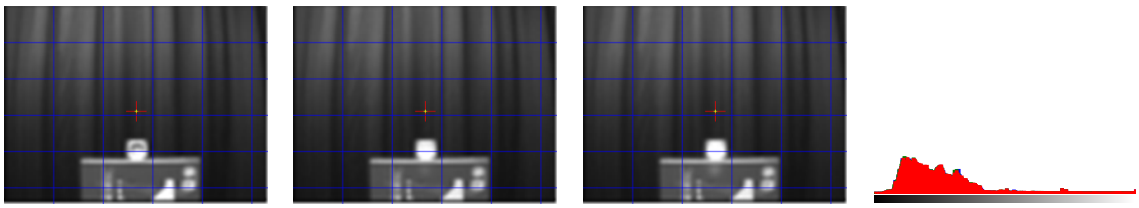


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

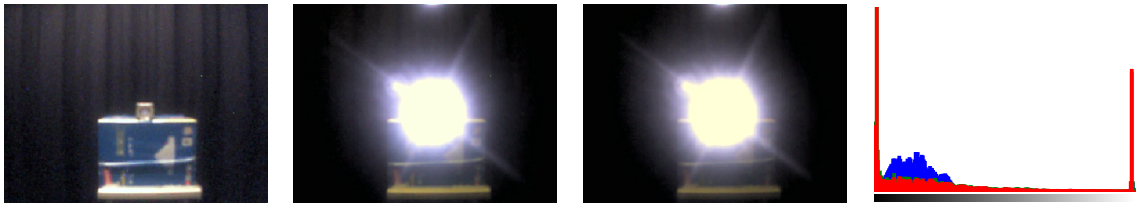


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 107: 850 nm spot in light @ 50 cm. Blue is 0%, green is 50%, red is 100%.

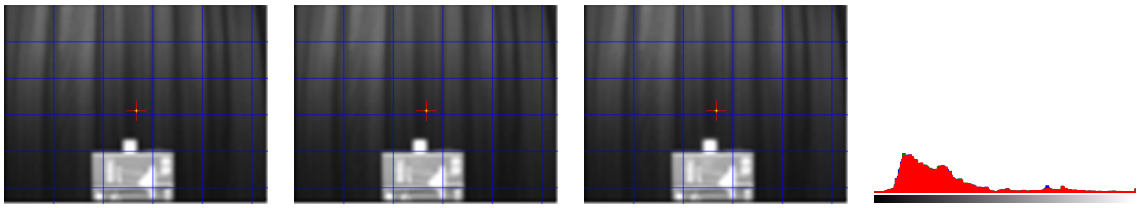


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

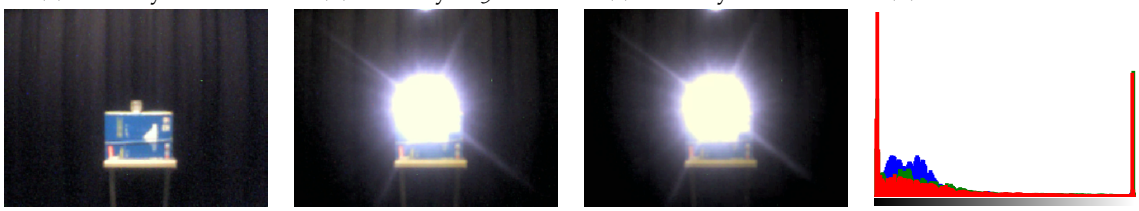


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 108: 850 nm spot in light @ 100 cm. Blue is 0%, green is 50%, red is 100%.

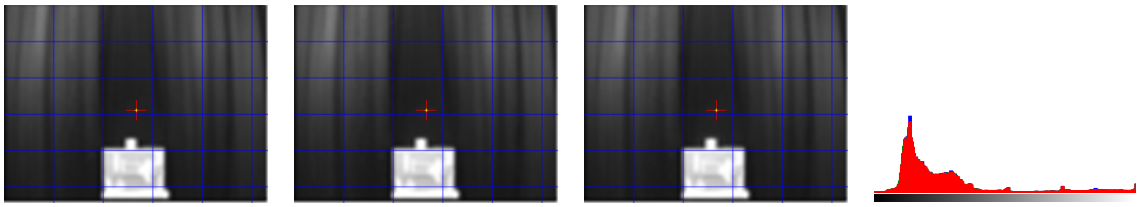


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

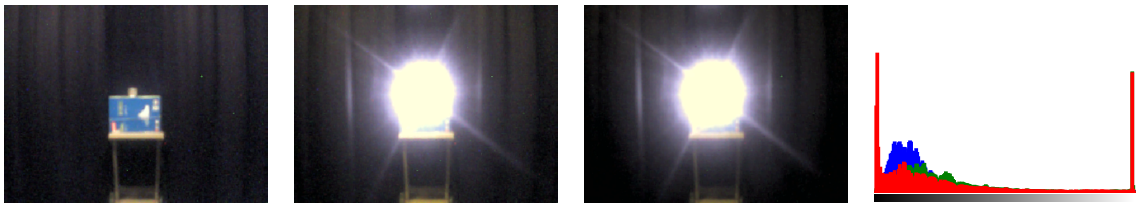


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 109: 850 nm spot in light @ 150 cm. Blue is 0%, green is 50%, red is 100%.

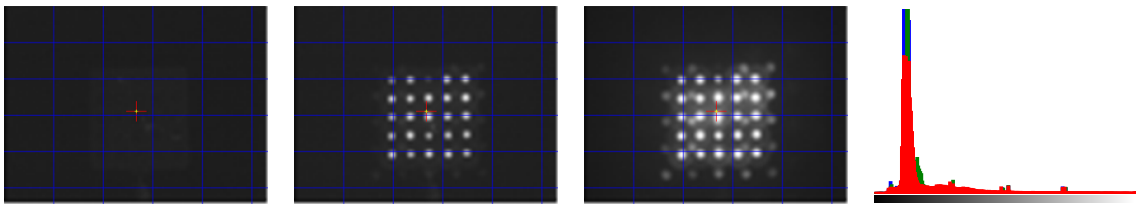


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

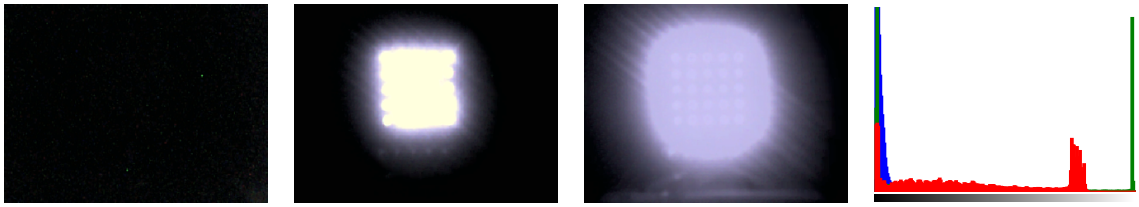


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 110: 850 nm spot in light @ 200 cm. Blue is 0%, green is 50%, red is 100%.

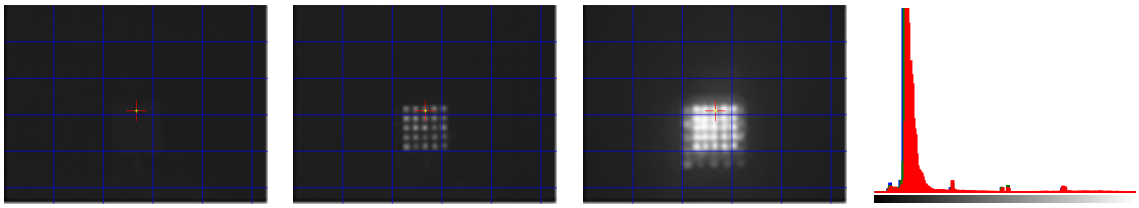


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

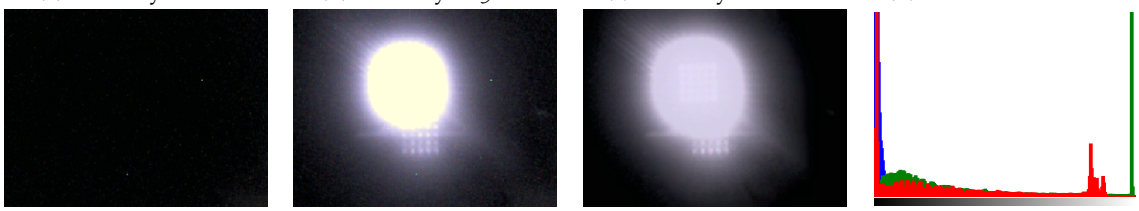


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 111: 940 nm 5x5 LED matrix in dark @ 50 cm. Blue is 0%, green is 50%, red is 100%.

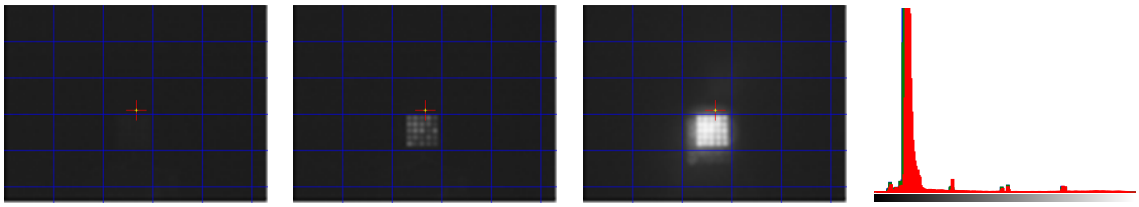


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.



(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

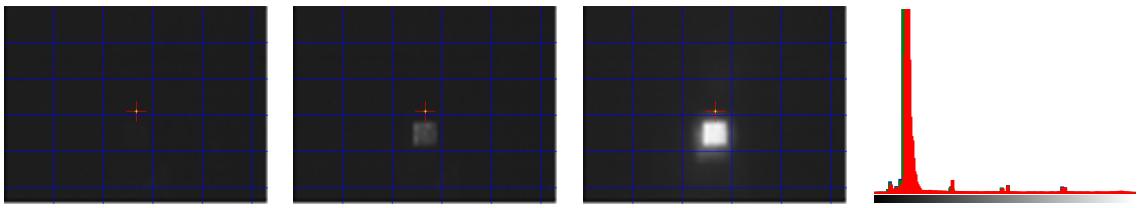
Fig. 112: 940 nm 5x5 LED matrix in dark @ 100 cm. Blue is 0%, green is 50%, red is 100%.



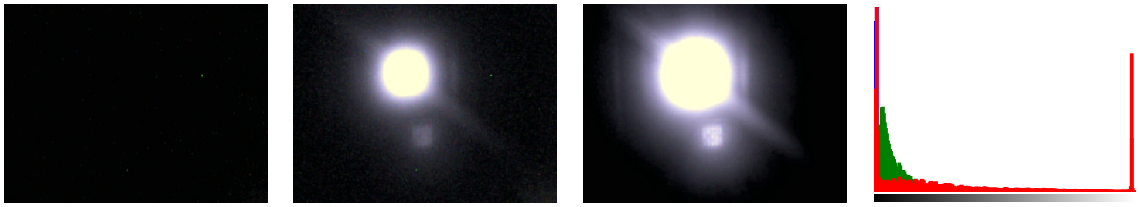
(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 113: 940 nm 5x5 LED matrix in dark @ 150 cm. Blue is 0%, green is 50%, red is 100%.

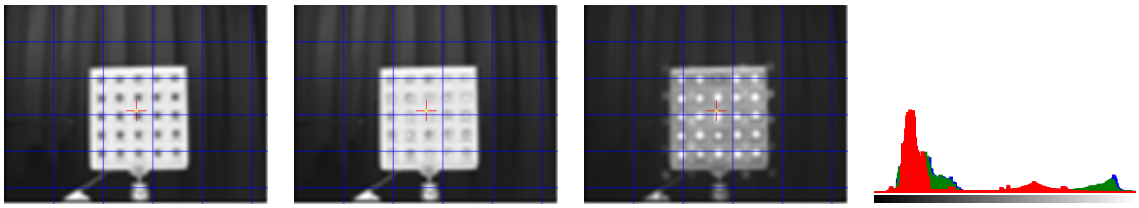


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

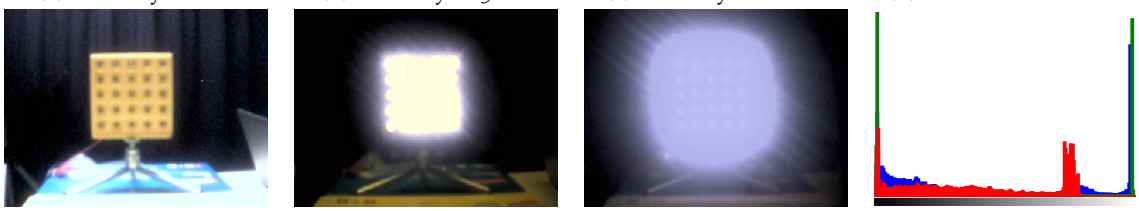


(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 114: 940 nm 5x5 LED matrix in dark @ 200 cm. Blue is 0%, green is 50%, red is 100%.

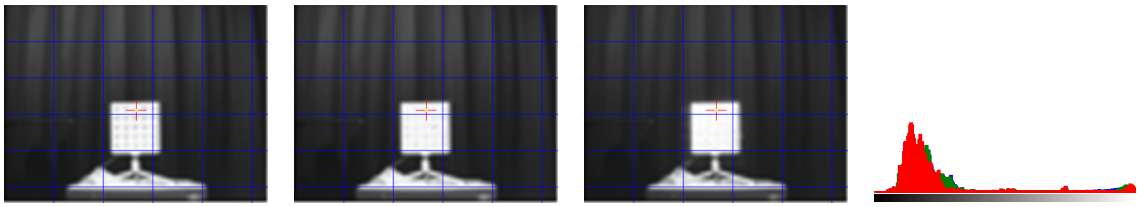


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.



(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

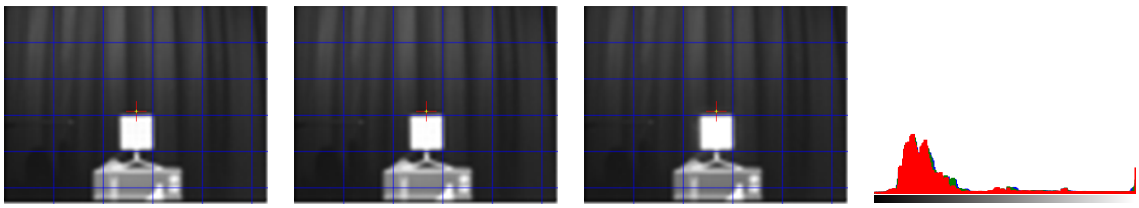
Fig. 115: 940 nm 5x5 LED matrix in Light @ 50 cm. Blue is 0%, green is 50%, red is 100%.



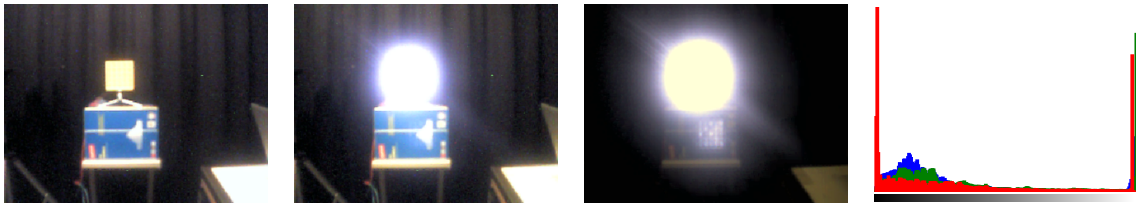
(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 116: 940 nm 5x5 LED matrix in Light @ 100 cm. Blue is 0%, green is 50%, red is 100%.

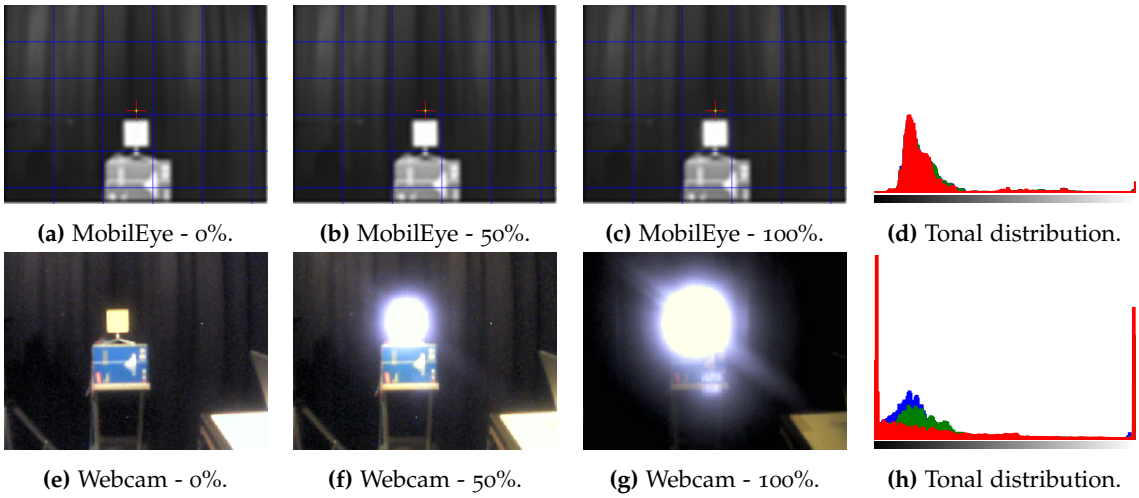


(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.



(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 117: 940 nm 5x5 LED matrix in Light @ 150 cm. Blue is 0%, green is 50%, red is 100%.



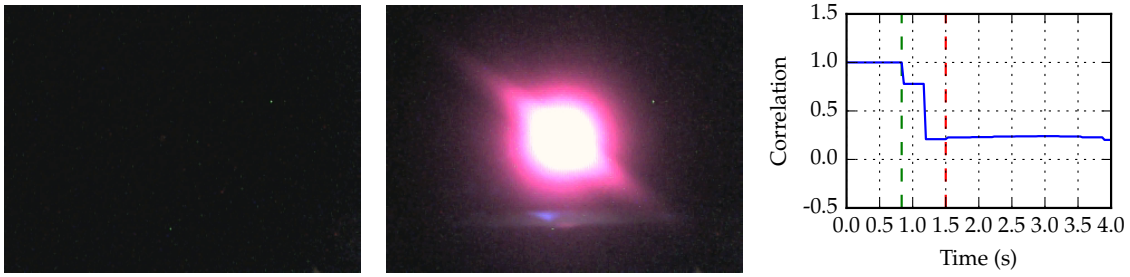
(a) MobilEye - 0%. **(b)** MobilEye - 50%. **(c)** MobilEye - 100%. **(d)** Tonal distribution.

(e) Webcam - 0%. **(f)** Webcam - 50%. **(g)** Webcam - 100%. **(h)** Tonal distribution.

Fig. 118: 940 nm 5x5 LED matrix in Light @ 200 cm. Blue is 0%, green is 50%, red is 100%.

C.3 CONFUSING THE AUTO CONTROLS

Refer to Section 4.1.4 for an explanation and interpretation of the results. In short, each group of three images consists of two frames from the video, the start frame and the least correlated frame. The graph is a correlation over time, between the first frame and all subsequent frames. The blinding interval is indicated between the two vertical bars.



(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 119: 365 nm spot in dark @ 50 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

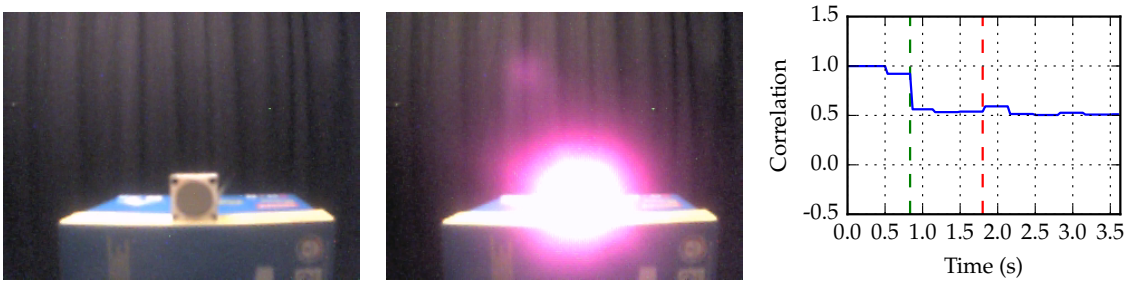
Fig. 120: 365 nm spot in dark @ 100 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 121: 365 nm spot in dark @ 150 cm. Green line is time of start, red line is time of stop.

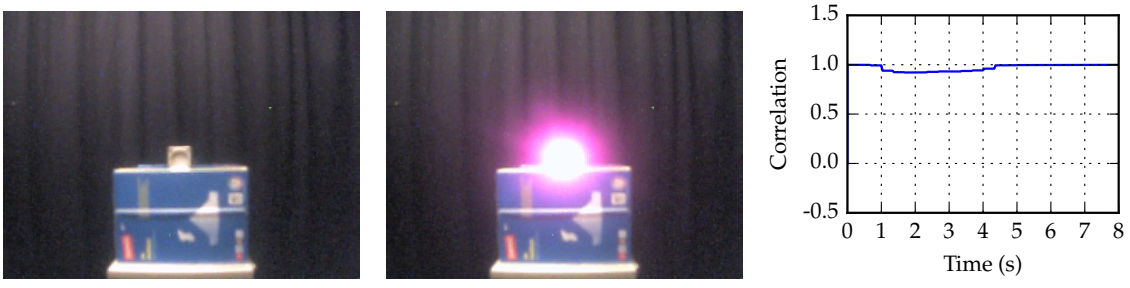
(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 122: 365 nm spot in dark @ 200 cm. Green line is time of start, red line is time of stop.



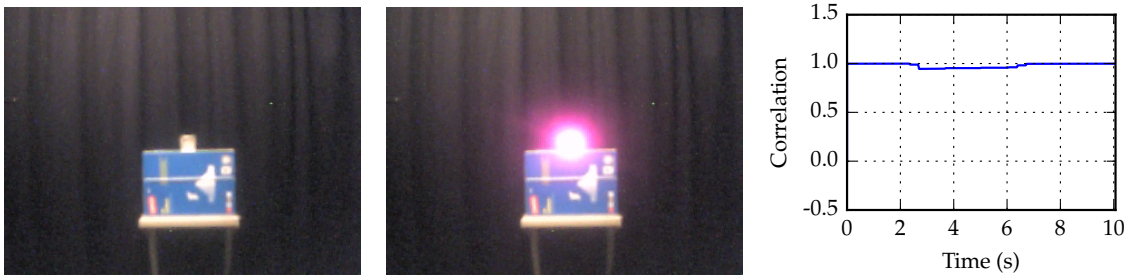
(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 123: 365 nm spot in light @ 50 cm. Green line is time of start, red line is time of stop.



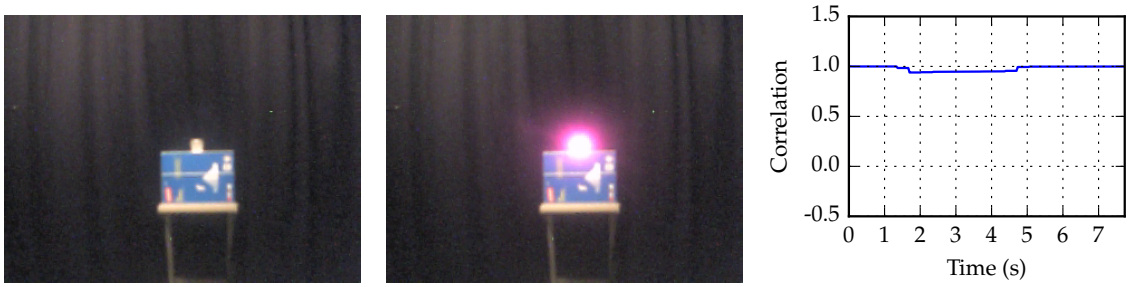
(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 124: 365 nm spot in light @ 100 cm. Green line is time of start, red line is time of stop.



(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 125: 365 nm spot in light @ 150 cm. Green line is time of start, red line is time of stop.



(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 126: 365 nm spot in light @ 200 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 127: White spot in dark @ 50 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

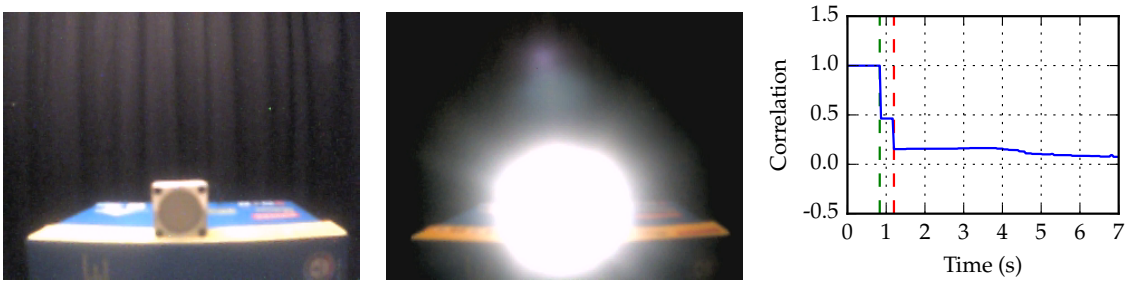
Fig. 128: White spot in dark @ 100 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 129: White spot in dark @ 150 cm. Green line is time of start, red line is time of stop.

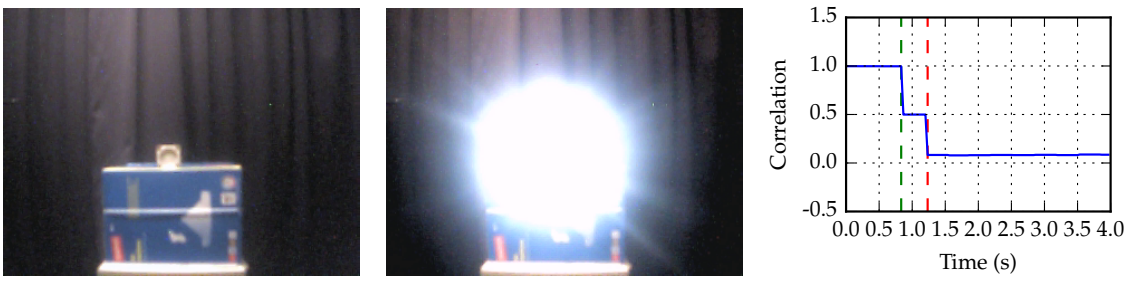
(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 130: White spot in dark @ 200 cm. Green line is time of start, red line is time of stop.



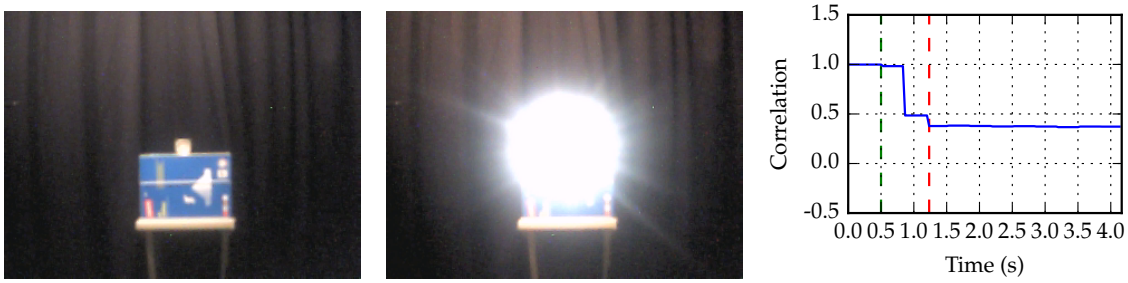
(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 131: White spot in light @ 50 cm. Green line is time of start, red line is time of stop.



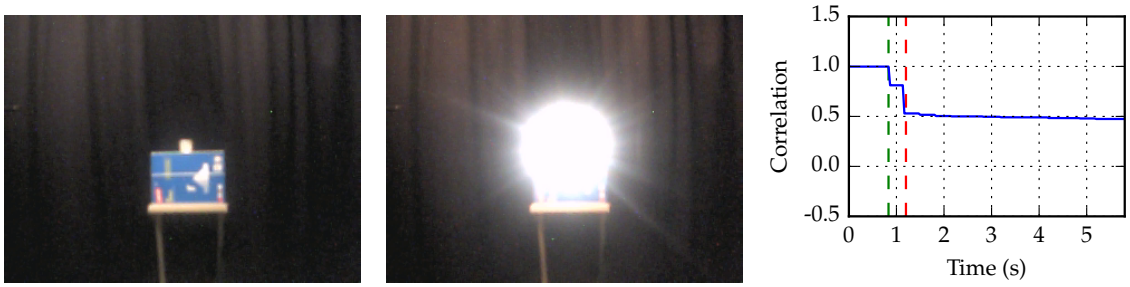
(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 132: White spot in light @ 100 cm. Green line is time of start, red line is time of stop.



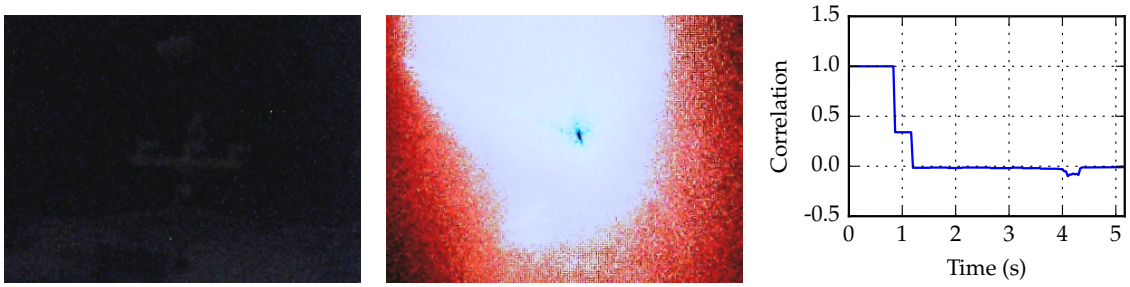
(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 133: White spot in light @ 150 cm. Green line is time of start, red line is time of stop.



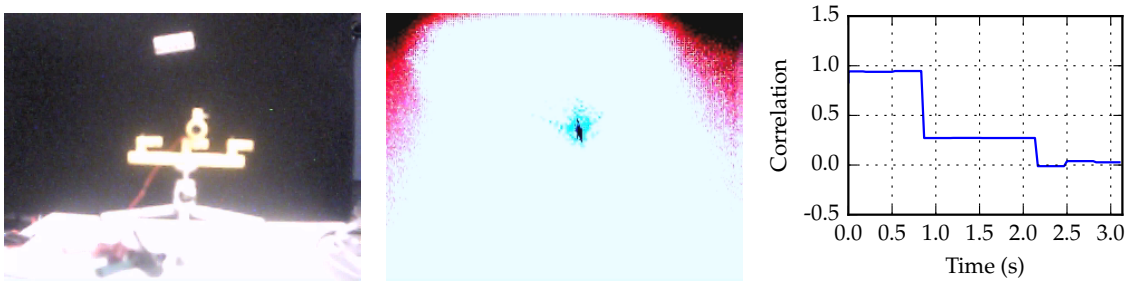
(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 134: White spot in light @ 200 cm. Green line is time of start, red line is time of stop.



(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 135: 650 nm Laser in dark @ 50cm. Green line is time of start, red line is time of stop.



(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 136: 650 nm Laser in Light @ 50cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 137: 850 nm spot in dark @ 50 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 138: 850 nm spot in dark @ 100 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 139: 850 nm spot in dark @ 150 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 140: 850 nm spot in dark @ 200 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

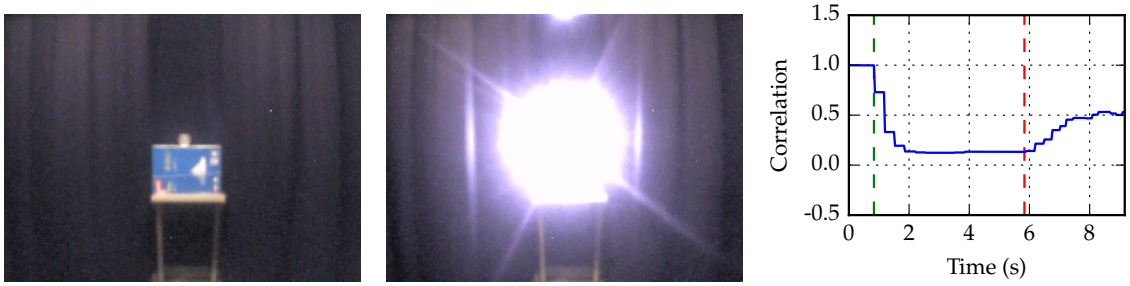
Fig. 141: 850 nm spot in light @ 50 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 142: 850 nm spot in light @ 100 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 143: 850 nm spot in light @ 150 cm. Green line is time of start, red line is time of stop.



(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 144: 850 nm spot in light @ 200 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 145: 940 nm 5x5 LED matrix in dark @ 50 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 146: 940 nm 5x5 LED matrix in dark @ 100 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 147: 940 nm 5x5 LED matrix in dark @ 150 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 148: 940 nm 5x5 LED matrix in dark @ 200 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 149: 940 nm 5x5 LED matrix in Light @ 50 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 150: 940 nm 5x5 LED matrix in Light @ 100 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 151: 940 nm 5x5 LED matrix in Light @ 150 cm. Green line is time of start, red line is time of stop.

(a) Start frame. **(b)** Lowest correlated frame. **(c)** Correlation over time.

Fig. 152: 940 nm 5x5 LED matrix in Light @ 200 cm. Green line is time of start, red line is time of stop.

D OVERVIEW OF HARDWARE

This chapter lists all the hardware used during the experiments conducted for this work.

D.1 MOBILEYE C2-270

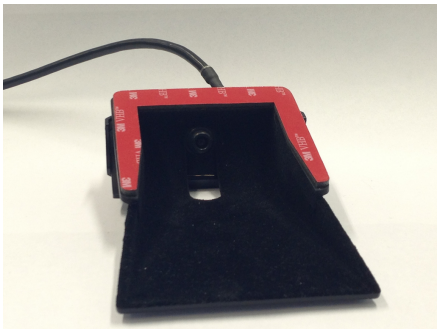


Fig. 153: Windshield Camera. This camera is installed in a car directly under the windshield. It is responsible for processing the data. The camera is an Aptina MT9V024 CMOS Red/Clear camera, capable of providing an image size of 752x480 at 60 FPS.

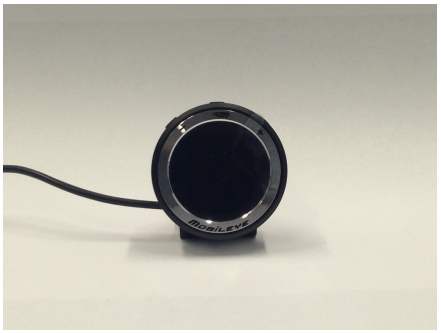


Fig. 154: Information Display. Informs the driver about dangerous events, such as incoming traffic or pedestrians.

Fig. 155: Connection Box. The camera, display, power and car signals (CAN or digital) are connected to this box.

Fig. 156: PC Connection box. This box is required for configuring and calibrating the MobilEye system. It is not included in the package, and only meant for car dealers.

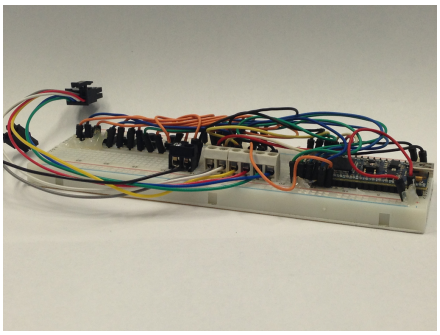


Fig. 157: Custom built simulator. Signal simulator to make the MobilEye C2-270 believe it was installed in a car. Supports CAN signals and digital signals. This simulator is connected to a PC via USB.

D.2 IBEO LUX 3

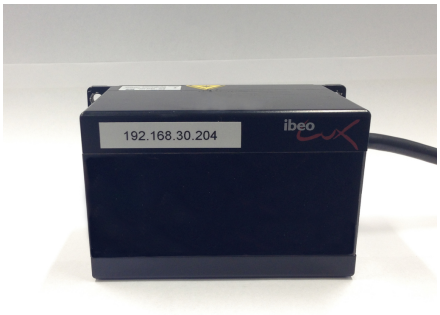


Fig. 158: ibeo LUX 3 Lidar. Can measure distance up to 200 m with a viewing angle of 110 °. It has four layers and can sense up to three echoes to see through bad weather conditions. Scanning speeds are 12.5 Hz, 25 Hz or 50 Hz. The angular resolution is up to 1/8 ° horizontal, the distance resolution is 4 cm.

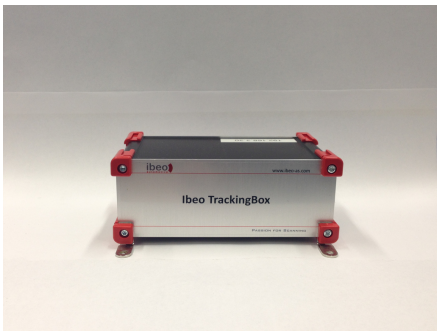


Fig. 159: ibeo Tracking Box. Can track dynamic objects such as pedestrians and cars from Lidar data.

Fig. 160: ibeo Box. Connects the Lidar to the Ethernet bus or CAN bus.

D.3 LIGHT SOURCES

D.3.1 Infrared

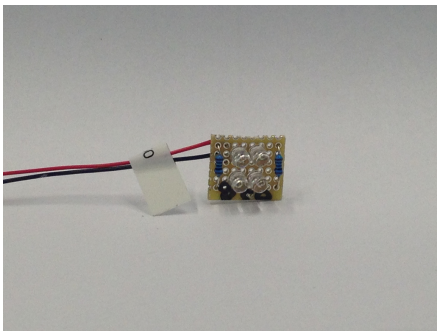


Fig. 161: Osram SFH4550 IR 850 nm LED. Rated for 100 mA per LED. Viewing angle ±3 °.

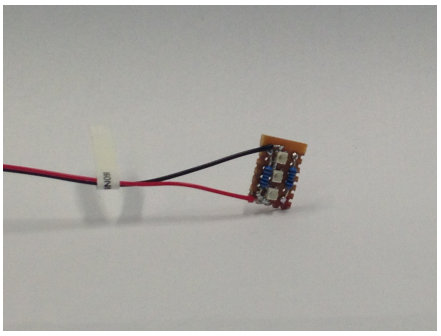


Fig. 162: Osram SFH4258 IR 860 nm LED. Rated for 100 mA per LED. Viewing angle ±15 °.

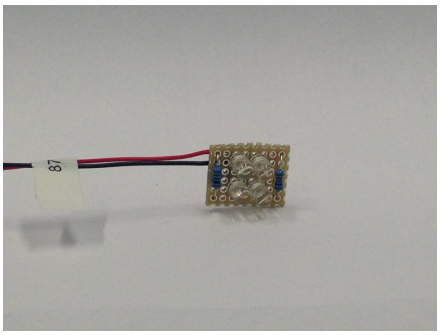


Fig. 163: Ledsee IR 875 nm LED. Rated for 100 mA per LED. Viewing angle ±10 °.

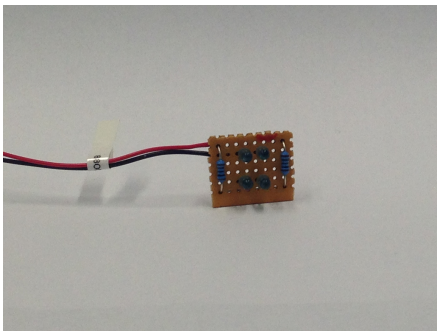


Fig. 164: Honeywell SEP8705-3 880 nm LED. Rated for 50 mA per LED. Viewing angle ±15 °.

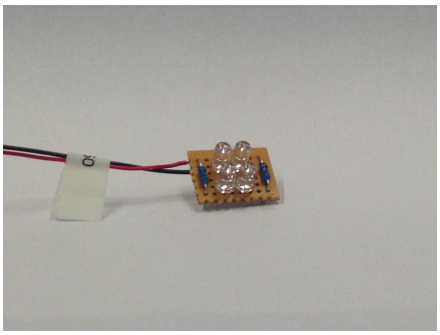


Fig. 165: Ledsee IR 940 nm LED. Rated for 20 mA per LED. Viewing angle ±15 °.

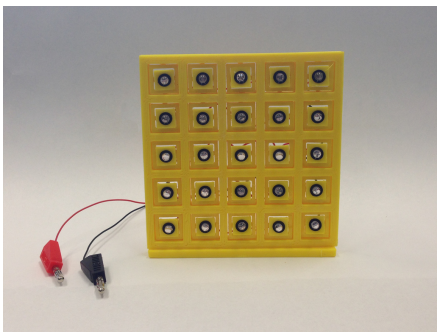


Fig. 166: Ledsee IR 940 nm 5x5 LED Matrix. Each LED is rotatable in horizontal and vertical direction for beam positioning. Rated for 20 mA per LED. Viewing angle ±10 °.

D.3.2 Spots

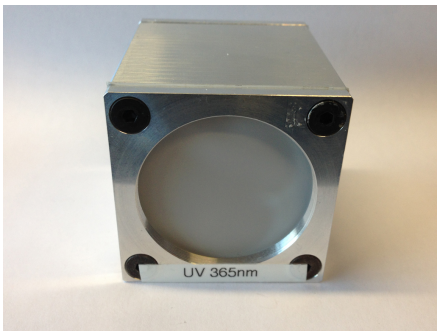


Fig. 167: UV 365 nm LED spot with diffuser. Inside is a LEDENGIN LZ1-00U600 Rated for 5 W.

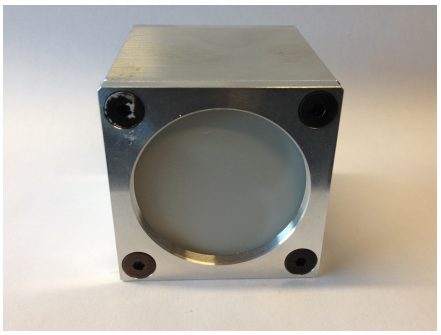


Fig. 168: White LED spot with diffuser. Inside is a CML INNOVATIVE TECHNOLOGIES ILL3A0001E. Rated for 3 W. The color is cold-white with three peaks in the spectrum at a wavelength of 450 nm, 550 nm and 610 nm.

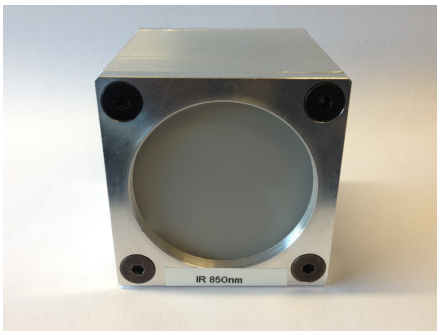


Fig. 169: IR 850 nm LED spot with diffuser. Inside is a LEDENGIN LZ1-00R400. Rated for 2 W.

D.3.3 Lasers

Fig. 170: Picotronic LE650-5-3 650 nm diode line laser with a focusable lens. Maximal output is 5 mW.

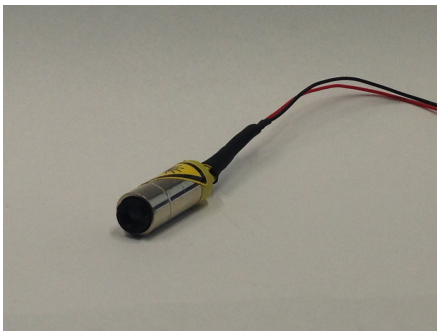


Fig. 171: Ledsee 650 nm diode point laser with a focusable lens. Maximal output is 5 mW.

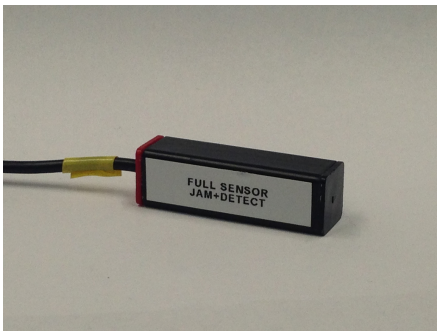


Fig. 172: Osram SPL-PL90 905 nm diode laser and an Osram SFH-213 infrared sensitive photodetector. Maximal output is 25 W for 100 ns. The viewing angle is 9 °.

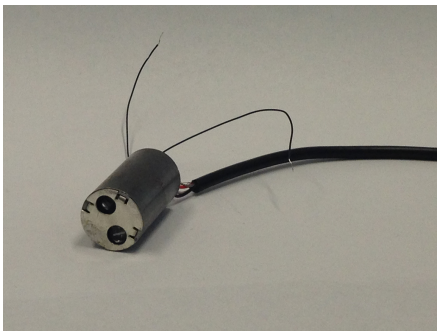


Fig. 173: Osram SPL-PL90 905 nm diode laser and an Osram SFH-213 infrared sensitive photodetector. Maximal output is 25 W for 100 ns. This is the same as above, but without the plastic enclosure/filter.

D.4 CAMERAS

Fig. 174: C-Cam BCI5-1394-M-40 with IR-pass filter. Monochrome Firewire camera with advanced controls over shutter speed and exposure time. Maximal image size is 1280x1024 at 25 FPS.

Fig. 175: Basler acA2040-25gm NIR. Monochrome Ethernet Near-Infrared camera with advanced controls over shutter speed and exposure. Maximal image size is 2048 x 2048 at 25 FPS.

Fig. 176: Trust WB-3400T USB Complementary Metal Oxide Semiconductor (CMOS) webcam with IR-filter removed. Used as reference camera for experiment validation. Maximal image size is 640x480 pixels at 30 FPS.

D.5 MEASUREMENT TOOLS

Fig. 177: Tenma 72-6693 Lux Meter. Capable of measuring light intensity, as an absolute value.

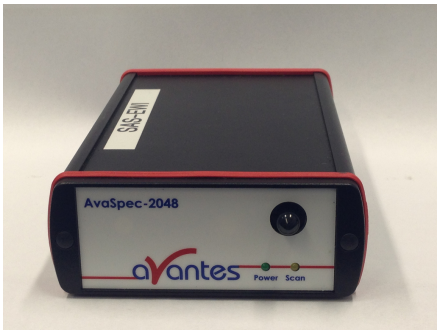


Fig. 178: AvaSpect-2048 USB Spectrometer. Capable of measuring light between 200 nm and 1100 nm, producing relative measurements.

Fig. 179: HP 8011A Pulse Generator. Can generate square wave pulses of a certain length. Can be externally triggered.

Fig. 180: Philips PM 5715 Pulse Generator. Pulse generator with a delay function (10 ns - 10 ms). Can be externally triggered.

D.6 OTHER

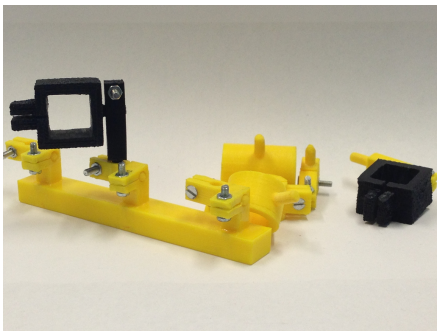


Fig. 181: Several custom made 3D-printed mounts to hold the light sources and lasers. Can be mounted on tripod.

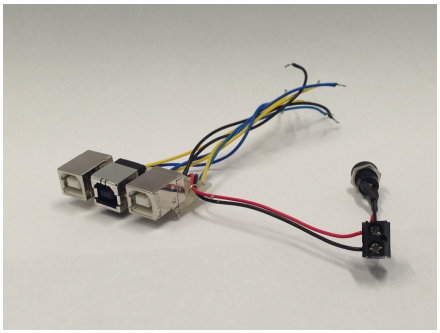


Fig. 182: Several custom made 3D-printed mounts to hold the light sources and lasers. Can be mounted on tripod.

Fig. 183: Test subject number one. His name is Henk.

ACRONYMS

ACC	Adaptive Cruise Control. 5, 15, 17
ADAS	Advanced Driver Assistance System. i, 48, 101
AGC	Automatic Gain Correction. 24, 25
ALS	Autocovariance Least-Squares. 109
ASK	Amplitude Shift Keying. 34, 35
AV	Autonomous Vehicle. i, 1–3, 5–9, 11, 12, 14, 15, 21–23, 26–28, 34, 47, 84, 88, 90, 96, 97, 101, 103
BASt	Bundesanstalt für Straßenwesen. 5
C/A	Coarse/Acquisition. 17, 24, 26
CAN	Controller Area Network. 10, 14, 49, 97
CAS	Collision Avoidance System. 3, 5, 15–17, 87, 103
CCD	Charge-Coupled Device. 27, 28, 32, 50, 51, 93, 113
CDMA	Code Division Multiple Access. 17
CMOS	Complementary Metal Oxide Semiconductor. vii, 27, 28, 32, 33, 50, 51, 54, 93, 141, 147
CRC	Cyclic Redundancy Check. 35
CUSUM	Cumulative Sum. 46
CVSS	Common Vulnerability Scoring System. 8, 9
CWI	Continuous Wave Inference. vii, 24, 26
DARPA	Defense Advanced Research Projects Agency. 1
DAVI	Dutch Automated Vehicle Initiative. 48
DSLR	Digital Single-lens Reflex. 28, 32
DSP	Digital Signal Processor. 85, 104, 105
ECDSA	Elliptic Curve Digital Signature Algorithm. 27
ECU	Electronic Control Unit. 33, 35
EGNOS	European Geostationary Navigation Overlay Service. 18
Emap	Enhanced Map. vii, 21–23
EMC	Electromagnetic compatibility. 17
EMP	Electromagnetic Pulse. 11
ESA	European Space Agency. 18
FMEA	Failure Mode and Effect Analysis. 8, 9
FPGA	Field-Programmable Gate Array. 85, 104, 105
FSK	Frequency Shift Keying. 34, 35
GIS	Geographical Information System. 22
GLONASS	Global Navigation Satellite System. 18, 20, 21
GNSS	Global Navigation Satellite System. i, vii, ix, 2, 17–19, 21–26, 36, 47, 97, 98, 103
GPS	Global Positioning System. vii, 7, 17, 18, 20–22, 24–26, 36, 39, 46, 89, 106

HMI	Human-machine Interaction. 10
HMM	Hidden Markov Model. 41
IMU	Inertial Measurement Unit. 7, 21, 23, 25, 36, 39, 46, 89, 98, 106
KF	Kalman Filter. vii, 36–41, 45, 46, 67, 89, 97, 106–110
Lidar	Laser Image Detection and Ranging. i, iii, vii, viii, 2, 3, 7, 14–17, 21, 29, 47, 67–90, 95–99, 101–106, 113, 142, 143
MCU	Microcontroller Unit. 35, 36, 79, 85, 104, 105
MSAS	Multi-functional Satellite Augmentation System. 18
NAVSTAR	Navigation Satellite Time And Ranging. 17
NHTSA	National Highway Traffic Safety Administration. 5
NMA	Navigation Message Authentication. 26, 27
PDF	Particle Density Function. 42–44
PF	Particle Filter. vii, viii, 16, 36, 39–45, 89–92, 102, 107, 109–111
PRN	Pseudorandom Noise. 17, 18, 24
QE	Quantum Efficiency. 27, 52
Radar	Radio Detection and Ranging. 2, 14, 15, 17, 47, 103
SBAS	Satellite-based Augmentation System. 18
SDCM	Wide-area System of Differential Corrections and Monitoring. 18
SDR	Software Defined Radio. 34, 35
SIR	Sequential Importance Resampling. 42
SIS	Sequential Importance Sampling. 42
Sonar	Sound Navigation and Ranging. 14
TESLA	Timed Efficient Stream Loss-Tolerant Authentication. 27
TPMS	Tire-pressure Monitoring System. i, vii, 33–35, 47
V2I	Vehicle-to-Infrastructure. 6
V2V	Vehicle-to-Vehicle. 6, 96
V2x	Vehicle-to-X. 23
WAAS	Wide Area Augmentation System. 18

BIBLIOGRAPHY

[1] Agentschap Telecom. *Vergunningsvrije radiotoepassingen*. 2014. URL: http://www.agentschaptelecom.nl/sites/default/files/brochure-vergunningsvrije-radiotoepassingen.pdf (visited on 08/29/2014) (cit. on p. 14).

[2] Air Force Link. *PHaSR gun*. 2005. URL: http://www.af.mil/news/story%5C_media.asp?storyID=123012699 (visited on 10/29/2006) (cit. on p. 32).

[3] Bernt M Åkesson, Bagterp Jørgensen, and Kjølstad Poulsen. "A Tool for Kalman Filter Tuning". In: (2007) (cit. on p. 109).

[4] Assad Alam, Sagar Behere, and MA Khan. "The Development of a Cooperative Heavy-Duty Vehicle for the GCDC 2011: Team Scoop". In: *IEEE Transactions on Intelligent Transportation Systems* 13.3 (Sept. 2012), pp. 1033–1049. ISSN: 1524-9050. DOI: 10.1109/TITS.2012.2204876. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6236179%20http://trid.trb.org/view.aspx?id=1218005 (cit. on pp. 1, 47, 97, 101).

[5] José M. Álvarez Alvarez and Antonio M. Lopez. "Road Detection Based on Illuminant Invariance". In: *IEEE Transactions on Intelligent Transportation Systems* 12.1 (Mar. 2011), pp. 184–193. ISSN: 1524-9050. DOI: 10.1109/TITS.2010.2076349. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5594640 (cit. on p. 29).

[6] Nayegandhi Amar. "LIDAR technology overview". In: *ETI–US Geological Survey. Retrieved August* 17 (2006), p. 2008 (cit. on p. 17).

[7] Aptina. *Aptina MT9V024 Datasheet*. 2008. URL: https://aptina.atlassian.net/wiki/download/attachments/10551342/MT9V024%5C_DS.pdf (visited on 12/18/2014) (cit. on pp. 54, 62, 94).

[8] Armlaser. *Laser Dazzlers*. 2015. URL: http://www.armlaser.com/laser-dazzlers-c-30.html (visited on 01/12/2015) (cit. on pp. 32, 93).

[9] C Arora et al. "Seam Reconstruct: Dynamic scene stitching with Large exposure difference". In: *Applications of Digital Information and Web Technologies, 2009. ICADIWT '09. Second International Conference on the*. Aug. 2009, pp. 574–578. DOI: 10.1109/ICADIWT.2009.5273843 (cit. on p. 93).

[10] Ibeo a.s. *ibeo Lux 2010 - Technical Facts*. 2010. URL: http://www.autonomoustuff.com/uploads/9/6/0/5/9605198/ibeo%5C_lux%5C_as.pdf (visited on 09/23/2014) (cit. on pp. 16, 17, 69).

[11] Ibeo a.s. *ibeo Object Tracking*. 2010. URL: http://www.autonomoustuff.com/uploads/9/6/0/5/9605198/ibeo%5C_object%5C_tracking%5C_as.pdf (visited on 02/12/2015) (cit. on pp. 67, 89).

[12] Ibeo a.s. *Operating Manual ibeo LUX 2010 Laserscanner*. 2010 (cit. on p. 70).

[13] C. Bahlmann et al. "A system for traffic sign detection, tracking, and recognition using color, shape, and motion information". In: *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.* (2005), pp. 255–260. DOI: 10.1109/IVS.2005.1505111. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1505111 (cit. on p. 29).

[14] Li Bai and Yan Wang. "A Sensor Fusion Framework Using Multiple Particle Filters for Video-Based Navigation". In: *IEEE Transactions on Intelligent Transportation Systems* 11.2 (2010), pp. 348–358. URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=5437329 (cit. on pp. 22, 23, 29).

[15] Arne Bartels et al. *Legal Consequences of an Increase in Vehicle Automation*. Tech. rep. 2013 (cit. on p. 5).

[16] Jan Becker et al. "Junior: The Stanford Entry in the Urban Challenge". In: 25.9 (2008), pp. 569–597. DOI: 10.1002/rob (cit. on pp. 1, 47, 97, 101).

[17] David Betaille and Rafael Toledo-Moreo. "Creating Enhanced Maps for Lane-Level Vehicle Navigation". In: *IEEE Transactions on Intelligent Transportation Systems* 11.4 (Dec. 2010), pp. 786–798. ISSN: 1524-9050. DOI: 10.1109/TITS.2010.2050689. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5499153 (cit. on p. 21).

[18] Krysta Boccuzzi. "Investigating the Causes of and Possible Remedies for Sensor Damage in Digital Cameras Used on the OMEGA Laser Systems". In: August (2008). URL: http://beta5.lle.rochester.edu/media/publications/high%5C_school%5C_reports/documents/hs%5C_reports/2008/Boccuzzi%5C_Krysta.pdf (cit. on pp. 32, 33).

[19] Jared Boone. *Tire Pressure Monitoring System decoding tools*. 2012. URL: https://github.com/jboone/tpms (visited on 08/14/2014) (cit. on p. 35).

[20] Daniele Borio, Cillian O'Driscoll, and Joaquim Fortuny. "GNSS Jammers: Effects and Countermeasures". In: *2012 6th ESA Workshop on Satellite Navigation Technologies (Navitec 2012) & European Workshop on GNSS Signals and Signal Processing* (Dec. 2012), pp. 1–7. DOI: 10.1109/NAVITEC.2012.6423048. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6423048 (cit. on pp. 24, 26).

[21] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008 (cit. on p. 51).

[22] Alberto Broggi et al. "Extensive Tests of Autonomous Driving Technologies". In: *IEEE Transactions on Intelligent Transportation Systems* 14.3 (2013), pp. 1403–1415. URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=6522193 (cit. on pp. 1, 28, 47, 97, 101).

[23] Zoz Brooks. *Hacking Driverless Vehicles*. 2013. URL: https://www.youtube.com/watch?v=k5E28fp4oc0 (cit. on pp. 2, 46).

[24] Stephen Checkoway and D McCoy. "Comprehensive Experimental Analyses of Automotive Attack Surfaces." In: *USENIX Security ...* (2011). URL: http://static.usenix.org/events/sec11/tech/full%5C_papers/Checkoway.pdf (cit. on pp. 3, 11, 14, 34).

[25] Hsu-Yung Cheng et al. "Lane Detection With Moving Vehicles in the Traffic Scenes". In: *IEEE Transactions on Intelligent Transportation Systems* 7.4 (Dec. 2006), pp. 571–582. ISSN: 1524-9050. DOI: 10.1109/TITS.2006.883940. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4019429 (cit. on p. 29).

[26] Andrew Dempster. "How Vulnerable is GPS?" In: (2001), pp. 1–6 (cit. on p. 24).

[27] Ernst-Dieter Dickmanns. "Vehicle guidance by computer vision". In: *High Precision Navigation*. Springer, 1989, pp. 86–96 (cit. on p. 1).

[28] Danny Dolev and Andrew C Yao. "On the security of public key protocols". In: *Information Theory, IEEE Transactions on* 29.2 (1983), pp. 198–208 (cit. on p. 9).

[29] Jie Du, Jason Masters, and Matthew Barth. "Lane-Level Positioning for In-Vehicle Navigation and Automated Vehicle Location (AVL) Systems". In: *2004 IEEE Intelligent Transportation Systems Conference* (2004), pp. 1–3. URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=1398868 (cit. on pp. 21, 22).

[30] Isao Echizen. "Privacy Protection Techniques Using Differences in Human and Device Sensitivity". In: (2012), pp. 1–5 (cit. on pp. 33, 57).

[31] Sean R Eddy. "What is a hidden Markov model?" In: *Nature biotechnology* 22.10 (Oct. 2004), pp. 1315–6. ISSN: 1087-0156. DOI: 10.1038/nbt1004-1315. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2931519%5C&tool=pmcentrez%5C&rendertype=abstract (cit. on p. 44).

[32] Golnaz Elahi and Eric Yu. "Modeling and analysis of security trade-offs – A goal oriented approach". In: *Data & Knowledge Engineering* 68 (2009), pp. 579–598. ISSN: 0169023X. DOI: 10.1016/j.datak.2009.02.004 (cit. on p. 9).

[33] S Eum and HG Jung. "Enhancing Light Blob Detection for Intelligent Headlight Control Using Lane Detection". In: *Intelligent Transportation Systems, IEEE ...* 14.2 (2013), pp. 1003–1011. URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=6408201 (cit. on p. 29).

[34] European Commission. *Cars safer from 1 November 2012*. 2012. URL: http://europa.eu/rapid/press-release%5C_IP-12-1169%5C_en.htm (visited on 04/23/2014) (cit. on p. 34).

[35] Extreme Tech. *Frankfurt Auto Show: Mercedes shows off fully autonomous S-Class, production cars coming by 2020*. Sept. 2013. URL: http://www.extremetech.com/extreme/166598-frankfurt-auto-show-mercedes-shows-off-fully-autonomous-s-class-production-cars-coming-by-2020 (visited on 06/02/2014) (cit. on p. 1).

[36] KB Faizal. "Adaptive Exposure Control Algorithm For Day Night Video Surveillance". In: *vvdntech.com* (2013). URL: http://www.vvdntech.com/downloads/CCTV%5C_Camera%5C_Day%5C_Night%5C_Video%5C_Surveillance.pdf (cit. on pp. 33, 57).

[37] FAQs.org. *Geographic Information Systems FAQ*. 1997. URL: http://www.faqs.org/faqs/geography/infosystems-faq/ (visited on 08/19/2014) (cit. on p. 20).

[38] Alberto Faro, Daniela Giordano, and Concetto Spampinato. "Adaptive Background Modeling Integrated With Luminosity Sensors and Occlusion Processing for Reliable Vehicle Detection". In: *IEEE Transactions on Intelligent Transportation Systems* 12.4 (Dec. 2011), pp. 1398–1412. ISSN: 1524-9050. DOI: 10.1109/TITS.2011.2159266. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5941065 (cit. on pp. 29, 30).

[39] M Farsi, K Ratcliff, and Manuel Barbosa. "An overview of controller area network". In: *Computing & Control Engineering Journal* 10.3 (1999), pp. 113–120 (cit. on p. 14).

[40] Federal Space Agency Information- Analytical Centre. *GLONASS Constellation Status*. 2014. URL: https://glonass-iac.ru/en/GLONASS/ (visited on 10/12/2014) (cit. on p. 18).

[41] Federal Space Agency of the Russian Federation. *Developments of the GLONASS system and GLONASS Service*. 2004. URL: http://www.oosa.unvienna.org/pdf/sap/2004/vienna/presentations/wednesday/pm/revnivyk.pdf (visited on 04/24/2014) (cit. on p. 20).

[42] W.J. J Fleming. "Overview of automotive sensors". In: *IEEE Sensors Journal* 1.4 (2001), pp. 296–308. ISSN: 1530437X. DOI: 10.1109/7361.983469. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=983469 (cit. on pp. 5, 13, 14).

[43] Forbes. *Digital Carjackers Show Off New Attacks*. 2013. URL: https://www.youtube.com/watch?v=oqe6S6m73Zw (cit. on p. 14).

[44] D Fox, Jeffrey Hightower, and L Liao. "Bayesian filtering for location estimation". In: (2003). URL: http://www.computer.org/csdl/mags/pc/2003/03/b3024.pdf (cit. on pp. 42, 44).

[45] G. Galanis et al. "Applications of Kalman filters based on non-linear functions to numerical weather predictions". In: *Annales Geophysicae* 24.10 (Oct. 2006), pp. 2451–2460. ISSN: 1432-0576. DOI: 10.5194/angeo-24-2451-2006. URL: http://www.ann-geophys.net/24/2451/2006/ (cit. on p. 36).

[46] Andreas Geiger et al. "Team AnnieWAY's Entry to the 2011 Grand Cooperative Driving Challenge". In: *IEEE Transactions on Intelligent Transportation Systems* 13.3 (Sept. 2012), pp. 1008–1017. ISSN: 1524-9050. DOI: 10.1109/TITS.2012.2189882. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6178014 (cit. on pp. 1, 47, 97, 101).

[47] Chad Gibbons. *Hacking the Jeep Interior CAN-Bus*. 2013. URL: http://chadgibbons.com/2013/12/29/hacking-the-jeep-interior-can-bus/ (visited on 09/18/2014) (cit. on p. 14).

[48] Lee Gomes. *Hidden Obstacles for Google's Self-Driving Cars*. 2014. URL: http://www.technologyreview.com/news/530276/hidden-obstacles-for-googles-self-driving-cars/ (visited on 08/31/2014) (cit. on pp. 17, 33).

[49] NJ Gordon, DJ Salmond, and AFM Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". In: *IEE Proceedings F (Radar and Signal ...* 140 (1993), pp. 107–113. URL: http://digital-library.theiet.org/content/journals/10.1049/ip-f-2.1993.0015 (cit. on p. 39).

[50] GPS World. *A Comparison of Lidar and Camera-Based Lane Detection Systems*. 2012. URL: http://gpsworld.com/a-comparison-of-lidar-and-camera-based-lane-detection-systems/ (visited on 09/23/2014) (cit. on pp. 16, 29).

[51] MS Grewal and AP Andrews. "Applications of kalman filtering in aerospace 1960 to the present". In: *Control Systems, IEEE* (2010), pp. 69–78. URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=5466132 (cit. on p. 36).

[52] Paolo Grisleri and Isabella Fedriga. "The braive autonomous ground vehicle platform". In: *... Symposium on intelligent autonomous vehicles* (2010). URL: http://www.ce.unipr.it/people/bertozzi/pap/cr/iav2010-braive.pdf (cit. on pp. 28, 29, 92).

[53] Francis X III Grovers. *Kalman Filter Tutorial*. 2013. URL: https://www.youtube.com/watch?v=18TKA-YWhX0 (visited on 05/01/2014) (cit. on p. 39).

[54] D P Grubor, D M Šulić, and Vida Žigman. "Classification of X-ray solar flares regarding their effects on the lower ionosphere electron density profile". In: *Annales Geophysicae*. Vol. 26. 7. Copernicus GmbH. 2008, pp. 1731–1740 (cit. on p. 19).

[55] Kristen Hall-Geisler. *How Automatic Braking Systems Work*. 2014. URL: http://auto.howstuffworks.com/under-the-hood/trends-innovations/automatic-braking-system.htm (visited on 09/23/2014) (cit. on p. 16).

[56] A Harvey. *CV Dazzle: Camouflage from Computer Vision'*. 2012 (cit. on p. 33).

[57] Oona A Hathaway et al. "The law of cyber-attack". In: (2012) (cit. on p. 6).

[58] Richard Hawkins et al. "Using a software safety argument pattern catalogue: Two case studies". In: *Computer Safety, Reliability, and Security*. Springer, 2011, pp. 185–198 (cit. on p. 7).

[59] R Andrew Hicks and Christopher Croke. "Designing coupled free-form surfaces." In: *Journal of the Optical Society of America. A, Optics, image science, and vision* 27 (2010), pp. 2132–2137. ISSN: 1084-7529. DOI: 10.1364/JOSAA.27.002132 (cit. on p. 105).

[60] SA Hirani. "Energy consumption of encryption schemes in wireless devices". In: (2003). URL: http://d-scholarship.pitt.edu/7620/ (cit. on p. 35).

[61] JD Hol, TB Schon, and F Gustafsson. "On resampling algorithms for particle filters". In: *Nonlinear Statistical Signal ...* (2006). URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=4378824 (cit. on p. 42).

[62] Paul Horowitz, Winfield Hill, and Thomas C Hayes. *The art of electronics*. Vol. 2. Cambridge university press Cambridge, 1989 (cit. on p. 85).

[63] Todd E. Humphreys. "Detection Strategy for Cryptographic GNSS Anti-Spoofing". In: *IEEE Transactions on Aerospace and Electronic Systems* 49.2 (Apr. 2013), pp. 1073–1090. ISSN: 0018-9251. DOI: 10.1109/TAES.2013.6494400. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6494400 (cit. on p. 26).

[64] Radoslav Ivanov, Miroslav Pajic, and Insup Lee. "Attack-resilient sensor fusion". In: *Proceedings of the conference on Design, Automation & Test in Europe*. European Design and Automation Association. 2014, p. 54 (cit. on p. 46).

[65] Imad Jawhar, Nader Mohamed, and Hafsa Usmani. "An Overview of Inter-Vehicular Communication Systems, Protocols and Middleware". In: *Journal of Networks* 8.12 (Dec. 2013), pp. 2749–2761. ISSN: 1796-2056. DOI: 10.4304/jnw.8.12.2749-2761. URL: http://ojs.academypublisher.com/index.php/jnw/article/view/9396 (cit. on p. 11).

[66] RE Kalman. "A new approach to linear filtering and prediction problems". In: *Journal of basic Engineering* 82.Series D (1960), pp. 35–45. URL: http://fluidsengineering.asmedigitalcollection.asme.org/article.aspx?articleid=1430402 (cit. on p. 36).

[67] Yousun Kang et al. "Multiband Image Segmentation and Object Recognition for Understanding Road Scenes". In: *IEEE Transactions on Intelligent Transportation Systems* 12.4 (Dec. 2011), pp. 1423–1433. ISSN: 1524-9050. DOI: 10.1109/TITS.2011.2160539. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5959984 (cit. on pp. 28–30).

[68] Robert Kastner and Thomas Michalke. "Attention-based traffic sign recognition with an array of weak classifiers". In: *... (IV), 2010 IEEE* (June 2010), pp. 333–339. DOI: 10.1109/IVS.2010.5548143. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5548143%20http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=5548143 (cit. on pp. 29, 30).

[69] CG Keller and Markus Enzweiler. "The benefits of dense stereo for pedestrian detection". In: *IEEE Transactions on Intelligent Transportation Systems* 12.4 (2011), pp. 1096–1106. URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=5765690 (cit. on pp. 29, 31).

[70] Andrew J. Kerns, Kyle D. Wesson, and Todd E. Humphreys. "A blueprint for civil GPS navigation message authentication". In: *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014* (May 2014), pp. 262–269. DOI: 10.1109/PLANS.2014.6851385. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6851385 (cit. on p. 27).

[71] B Ravi Kiran and G V N A Harsha Vardhan. "A Fast Auto Exposure Algorithm for Industrial Applications Based on False-Position Method". In: Advances in Intelligent Systems and Computing 247 (2014). Ed. by Suresh Chandra Satapathy, Siba K Udgata, and Bhabendra Narayan Biswal, pp. 509–515. DOI: 10.1007/978-3-319-02931-3. URL: http://link.springer.com/10.1007/978-3-319-02931-3 (cit. on p. 33).

[72] Alois Knoll. *Environmental Sensing and Data Processing*. 2014 (cit. on pp. 1, 2, 28, 47, 101).

[73] Hui Kong, SE Sarma, and Feng Tang. "Generalizing Laplacian of Gaussian filters for vanishing-point detection". In: *IEEE Transactions on Intelligent Transportation Systems* 14.1 (2013), pp. 408–418. URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=6313912 (cit. on p. 29).

[74] Herbert J Kramer. *Observation of the Earth and its Environment: Survey of Missions and Sensors*. Springer, 2002, p. 772 (cit. on p. 17).

[75] Stefan Laible et al. "3d lidar-and camera-based terrain classification under different lighting conditions". In: *Autonomous Mobile Systems ...* c (2012). URL: http://link.springer.com/chapter/10.1007/978-3-642-32217-4%5C_3 (cit. on p. 16).

[76] RB Langley. "GPS, the Ionosphere, and the Solar Maximum". In: *GPS world* (2000). URL: http://gauss.gge.unb.ca/gpsworld/gpsworld.july00.pdf (cit. on p. 19).

[77] JH Lim, Omer Tsimhoni, and Yili Liu. "Investigation of Driver Performance With Night Vision and Pedestrian Detection Systems—Part I: Empirical Study on Visual Clutter and Glance Behavior". In: *IEEE Transactions on Intelligent Transportation Systems* 11.3 (Sept. 2010), pp. 670–677. ISSN: 1524-9050. DOI: 10.1109/TITS.2010.2049843. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5477182 (cit. on p. 28).

[78] Kevin Lim et al. "LiDAR remote sensing of forest structure". In: *Progress in Physical Geography* 27 (2003), pp. 88–106. ISSN: 03091333. DOI: 10.1191/0309133303pp360ra (cit. on p. 69).

[79] Herbert S Lin, Kenneth W Dam, William A Owens, et al. *Technology, policy, law, and ethics regarding US acquisition and use of cyberattack capabilities*. National Academies Press, 2009 (cit. on p. 6).

[80] Dave Litwiller. "CCD vs CMOS - Facts and Fiction". In: January (2001) (cit. on p. 27).

[81] Huaping Liu, Yulong Liu, and Fuchun Sun. "Traffic sign recognition using group sparse coding". In: *Information Sciences* (Jan. 2014), pp. 1–15. ISSN: 00200255. DOI: 10.1016/j.ins.2014.01.010. URL: http://linkinghub.elsevier.com/retrieve/pii/S002002551400022X (cit. on p. 29).

[82] DF Llorca and Vicente Milanés. "Autonomous Pedestrian Collision Avoidance Using a Fuzzy Steering Controller". In: *IEEE Transactions on Intelligent Transportation Systems* 12.2 (2011), pp. 390–401. URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=5715879 (cit. on p. 29).

[83] Lockheed Martin. *GPS III: The Next Generation Global Positioning System*. 2011. URL: http://www.lockheedmartin.com/content/dam/lockheed/data/space/documents/gps/GPSIII%5C_FactSheetFINAL1.pdf (visited on 09/23/2014) (cit. on p. 27).

[84] George Loukas, Diane Gan, and Tuan Vuong. "A Review of Cyber Threats and Defence Approaches in Emergency Management". In: *Future Internet* 5.2 (May 2013), pp. 205–236. ISSN: 1999-5903. DOI: 10.3390/fi5020205. URL: http://www.mdpi.com/1999-5903/5/2/205/ (cit. on p. 3).

[85] J. L. Lovell et al. "Using airborne and ground-based ranging lidar to measure canopy structure in Australian forests". In: *Canadian Journal of Remote Sensing* 29.5 (2003), pp. 607–622. ISSN: 07038992. DOI: 10.5589/m03-026 (cit. on p. 69).

[86] Kebina Manandhar and Yao Liu. "Combating False Data Injection Attacks in Smart Grid using Kalman Filter". In: *2014 International Conference on Computing, Networking and Communications (ICNC)* (Feb. 2014), pp. 16–20. DOI: 10.1109/ICCNC.2014.6785297. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6785297 (cit. on p. 46).

[87] Xuesong Mao and Daisuke Inoue. "Demonstration of In-Car Doppler Laser Radar at 1.55 μm for Range and Speed Measurement". In: *IEEE Transactions on Intelligent Transportation Systems* 14.2 (2013), pp. 599–607. URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=6387600 (cit. on pp. 14, 15, 95).

[88] Katty McCarron. *Battery exhaustion a looming issue with TPMS sensor units*. 2012. URL: http://www.tirebusiness.com/article/20120227/ISSUE/302279954/battery-exhaustion-a-looming-issue-with-tpms-sensor-units (visited on 08/14/2014) (cit. on p. 34).

[89] James P McDermott. "Attack net penetration testing". In: *Proceedings of the 2000 workshop on New security paradigms*. ACM. 2001, pp. 15–21 (cit. on p. 8).

[90] Richard J Meinhold and Nozer D Singpurwalla. "Understanding the Kalman filter". In: *The American Statistician* 37.2 (1983), pp. 123–127 (cit. on p. 107).

[91] Rob Merchant. "Analysis of Laser Light Threat to CCTV". In: (2012) (cit. on p. 32).

[92] MobilEye. *About MobilEye*. 2014. URL: http://www.mobileye.com/about/ (visited on 12/15/2014) (cit. on p. 48).

[93] A Mogelmose. "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey". In: *IEEE Transactions on Intelligent Transportation Systems* 13.4 (2012), pp. 1484–1497. URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=6335478 (cit. on pp. 29, 30).

[94] BA Moghaddam, H Haleh, and S Ebrahimijam. "Forecasting Trend and Stock Price with Adaptive Extended Kalman Filter Data Fusion." In: *International Proceedings of Economics ...* 4 (2011), pp. 119–123. URL: http://scholar.google.com/scholar?hl=en%5C&btnG=Search%5C&q=intitle:Forecasting+Trend+and+Stock+Price+with+Adaptive+Extended+Kalman+Filter+Data+Fusion%5C#0 (cit. on p. 36).

[95] Mohammad Sohrab Hossan Monsi. *Laser Radar for Precise Vehicle Velocity Measurement*. kassel university press GmbH, 2009 (cit. on p. 69).

[96] PY Montgomery, TE Humphreys, and BM Ledvina. "A multi-Antenna Defense Receiver-Autonomous GPS Spoofing Detection". In: *Inside GNSS* (2009). URL: http://scholar.google.com/scholar?hl=en%5C&btnG=Search%5C&q=intitle:A+multi-Antenna+Defense+Receiver-Autonomous+GPS+Spoofing+Detection%5C#0 (cit. on p. 26).

[97] National Highway Traffic Safety Administration. *U.S. Department of Transportation Releases Policy on Automated Vehicle Development*. 2013. URL: http://www.nhtsa.gov/About+NHTSA/Press+Releases/U.S.+Department+of+Transportation+Releases+Policy+on+Automated+Vehicle+Development (visited on 05/01/2014) (cit. on p. 5).

[98] Navipedia. *GLONASS Future and Evolutions*. 2011. URL: http://www.navipedia.net/index.php/GLONASS%5C_Future%5C_and%5C_Evolutions (visited on 10/12/2014) (cit. on p. 18).

[99] Navipedia. *GLONASS Performance*. 2011. URL: http://www.navipedia.net/index.php/GLONASS%5C_Performances%5C#GLONASS%5C_Accuracy%5C_Comparison (visited on 09/01/2014) (cit. on p. 21).

[100] Tyler Nighswander and Brent Ledvina. "GPS software attacks". In: *Proceedings of the ...* (2012). URL: http://dl.acm.org/citation.cfm?id=2382245 (cit. on p. 25).

[101] NIST. *NVD Common Vulnerability Scoring System Support v2*. 2007. URL: http://nvd.nist.gov/cvss.cfm (visited on 08/26/2014) (cit. on p. 8).

[102] Alan Ohnsman. *Nissan Sets Goal of Introducing First Self-Driving Cars by 2020*. 2013. URL: http://www.bloomberg.com/news/2013-08-27/nissan-sets-goal-of-bringing-first-self-driving-cars-by-2020.html (visited on 04/23/2014) (cit. on p. 1).

[103] Hiro Onishi. "Paradigm change of vehicle cyber security". In: *Cyber Conflict (CYCON), 2012 4th International ...* (2012). URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=6243987 (cit. on pp. 3, 9).

[104] OSRAM. *SPL PL90 Datasheet*. 2014. URL: http://www.osram-os.com/Graphics/XPic8/00149644%5C_0.pdf/SPL%20PL90.pdf (visited on 02/15/2015) (cit. on p. 74).

[105] Edgar Osuna, Robert Freund, and Federico Girosi. "Training support vector machines: an application to face detection". In: *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE. 1997, pp. 130–136 (cit. on p. 29).

[106] Andriy Panchenko and Lexi Pimenidis. "Towards practical attacker classification for risk analysis in anonymous communication". In: *Communications and Multimedia Security* (2006). URL: http://link.springer.com/chapter/10.1007/11909033%5C_22 (cit. on p. 9).

[107] Gaurav Pandey, James R McBride, and Ryan M Eustice. "Ford campus vision and lidar data set". In: *The International Journal of Robotics Research* 30.13 (2011), pp. 1543–1552 (cit. on p. 16).

[108] Panagiotis Papadimitratos and A Jovanovic. "GNSS-based Positioning: Attacks and countermeasures". In: *... Conference, 2008. MILCOM ...* iii (2008). URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=4753512 (cit. on p. 27).

[109] Ron Patton. *Software testing*. Sams Pub., 2006 (cit. on p. 47).

[110] P Perez, Jaco Vermaak, and Andrew Blake. "Data fusion for visual tracking with particles". In: *Proceedings of the IEEE* 92.3 (2004), pp. 1–18. URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=1271403 (cit. on p. 40).

[111] Jonathan Petit, Michael Feiri, and Frank Kargl. "Revisiting attacker model for smart vehicles". In: *Wireless Vehicular Communications (WiVeC), 2014 IEEE 6th International Symposium on*. IEEE. 2014, pp. 1–5 (cit. on p. 11).

[112] Jonathan Petit and Steven E. Shladover. "Potential Cyberattacks on Automated Vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* (2014) (cit. on pp. 2, 3, 6, 10).

[113] Walter Preiss. *WP's SloMo CCD and CMOS Sensor Info*. 2014. URL: http://www.fen-net.de/walter.preiss/e/slomoinf.html (visited on 12/18/2014) (cit. on p. 51).

[114] ML Psiaki and BW O'Hanlon. "GPS spoofing detection via dual-receiver correlation of military signals". In: *IEEE Transactions on Aerospace and Electronic Systems* 49.4 (2013). URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=6621814 (cit. on pp. 24, 26).

[115] MR Rajamani and JB Rawlings. "Estimation of the Disturbance Structure from Data using Semidefinite Programming and Optimal Weighting". In: *Automatica* (2009). URL: http://www.sciencedirect.com/science/article/pii/S000510980800366X (cit. on p. 109).

[116] Duminda I B Randeniya, Sudeep Sarkar, and Manjriker Gunaratne. "Vision–IMU Integration Using a Slow-Frame-Rate Monocular Vision System in an Actual Roadway Setting". In: *IEEE Transactions on Intelligent Transportation Systems* 11.2 (June 2010), pp. 256–266. ISSN: 1524-9050. DOI: 10.1109/TITS.2009.2038276. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5373840 (cit. on p. 29).

[117] Eric Rescorla and Brian Korver. "Guidelines for writing RFC text on security considerations". In: (2003) (cit. on p. 7).

[118] Rijksoverheid. *Besluit van tot wijziging van het Reglement verkeersregel en verkeerstekens 1990*. Tech. rep. 2014 (cit. on p. 23).

[119] Ishtiaq Rouf et al. "Security and Privacy Vulnerabilities of In-car Wireless Networks: A Tire Pressure Monitoring System Case Study". In: *Proceedings of the 19th USENIX Conference on Security*. USENIX Security'10. Berkeley, CA, USA: USENIX Association, 2010, p. 21. ISBN: 888-7-6666-5555-4. URL: http://dl.acm.org/citation.cfm?id=1929820.1929848 (cit. on pp. 34, 35).

[120] Joanna Rutkowska and Alexander Tereshkin. "Evil maid goes after TrueCrypt". In: *Invisible Things Lab's Blog* (2009) (cit. on p. 11).

[121] SAE International. "Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems". In: *Surface Vehicle Information Report J3016* (2014) (cit. on p. 5).

[122] A. Samman et al. "Potential use of near, mid and far infrared laser diodes in automotive LIDAR applications". In: *Vehicular Technology Conference* 5 (2000), pp. 2084–2089. URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=883239 (cit. on pp. 16, 71, 95).

[123] Scott Sanders. "Wavelength-agile lasers". In: *Optics and photonics news* 16.5 (2005), pp. 36–41 (cit. on p. 95).

[124] R. Sandhu. "Good-enough security". In: *IEEE Internet Computing* 7 (2003). ISSN: 1089-7801. DOI: 10.1109/MIC.2003.1167341 (cit. on p. 8).

[125] Christoph Schmittner et al. "Security Application of Failure Mode and Effect Analysis". In: *Computer Safety, Reliability, and Security* (2014), pp. 310–325 (cit. on pp. 8, 9).

[126] Byron Schmuland. *Example of a stochastic process which does not have the Markov property*. 2011. URL: http://math.stackexchange.com/a/89414 (visited on 08/31/2014) (cit. on p. 44).

[127] Bruce Schneier. "Attack trees". In: *Dr. Dobb's journal* 24.12 (1999), pp. 21–29 (cit. on p. 8).

[128] Robert Shirey. "RFC 2828: Internet security glossary". In: *The Internet Society* (2000) (cit. on p. 7).

[129] Désiré Sidibé. "Particle Filters and Applications in Computer Vision". In: (2011). URL: http://www2.lirmm.fr/~strauss/PageImage3/sidibe%5C_module%5C_image%5C_2011.pdf (cit. on p. 40).

[130] Meital Ben Sinai et al. "Exploiting Social Navigation". In: *arXiv preprint arXiv:1410.0151* (Oct. 2014). arXiv: 1410.0151. URL: http://arxiv.org/abs/1410.0151v1 (cit. on p. 25).

[131] S Singh, K Kingsley, and CL Chen. "Tire Pressure Maintenance - A Statistical Investigation". In: April (2009). URL: http://trid.trb.org/view.aspx?id=889178 (cit. on p. 34).

[132] Dean H Stamatis. *Failure mode and effect analysis: FMEA from theory to execution*. Asq Press, 2003 (cit. on p. 9).

[133] Talon. *The role of non-lethal effectors in port and harbour protection.* 2013. URL: http://www.talon2013.org/index.php?PAGE=features-non-lethal-effectors (visited on 02/16/2015) (cit. on p. 32).

[134] Isabelle Tang and Toby P. Breckon. "Automatic Road Environment Classification". In: *IEEE Transactions on Intelligent Transportation Systems* 12.2 (June 2011), pp. 476–484. ISSN: 1524-9050. DOI: 10.1109/TITS.2010.2095499. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5671488 (cit. on p. 29).

[135] M Thuy and F León. "Lane Detection and Tracking Based on Lidar Data". In: *Metrology and Measurement Systems* XVII.3 (2010). URL: http://www.degruyter.com/view/j/mms.2010.xvii.issue-3/v10178-010-0027-3/v10178-010-0027-3.xml (cit. on p. 16).

[136] R Toledo-Moreo. "Lane-level integrity provision for navigation and map matching with GNSS, dead reckoning, and enhanced maps". In: *IEEE Transactions on Intelligent Transportation Systems* 11.1 (2010), pp. 100–112. URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=5286855 (cit. on pp. 21–23).

[137] TomTom. *TomTom and Volkswagen partner to shape the future of Highly Automated Driving*. 2014. URL: http://corporate.tomtom.com/releasedetail.cfm?ReleaseID=874192 (visited on 10/13/2014) (cit. on p. 21).

[138] Trimble. *Using Pseudo Random Code as an Amplifier*. 2014. URL: http://www.trimble.com/gps%5C_tutorial/sub%5C_amplify.aspx (visited on 08/22/2014) (cit. on p. 24).

[139] Tweakers.net. *Russisch GPS-alternatief GLONASS heeft Wereldwijde Dekking*. 2011. URL: http://tweakers.net/nieuws/78663/russisch-gps-alternatief-glonass-heeft-wereldwijde-dekking.html (visited on 08/01/2014) (cit. on p. 21).

[140] Udacity. *Particle Filter - Final Position Estimation*. 2012. URL: http://forums.udacity.com/questions/1021237/particle-filter-final-position-estimation (visited on 06/24/2014) (cit. on pp. 40, 42).

[141] United Nations. "10 years of achievement of the United Nations on Global Navigation Satellite Systems". In: (2011). URL: http://www.sciencedirect.com/science/article/pii/S0168169999000563 (cit. on p. 24).

[142] University of Texas. *UT Austin Researchers Successfully Spoof an $80 million Yacht at Sea*. 2013. URL: http://www.utexas.edu/news/2013/07/29/ut-austin-researchers-successfully-spoof-an-80-million-yacht-at-sea/ (visited on 10/12/2014) (cit. on p. 25).

[143] U.S. Department of Defence. "Global Positioning System Standard Positioning Service Performance Standard". In: *Assistant secretary of defense for command, control, ...* September (2001). URL: http://scholar.google.com/scholar?hl=en%5C&btnG=Search%5C&q=intitle:Global+Positioning+System+Standard+Positioning+Service+Performance+Standard%5C#2 (cit. on p. 20).

[144] U.S. Department of Defense. "DoD Permanently Discontinues Procurement Of Global Positioning System Selective Availability". 2007. URL: http://www.defense.gov/releases/release.aspx?releaseid=11335 (cit. on p. 20).

[145] U.S. Department of Defense. *GPS Accuracy*. 2014. URL: http://www.gps.gov/systems/gps/performance/accuracy/ (visited on 08/15/2014) (cit. on p. 20).

[146] U.S. NVD. *CVE-2014-0160*. 2014. URL: http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-0160 (visited on 02/11/2015) (cit. on p. 9).

[147] QK Vuong, SH Yun, and Suki Kim. "A new auto exposure and auto white-balance algorithm to detect high dynamic range conditions using CMOS technology". In: *Proceedings of the World Congress on ...* (2008). URL: http://www.iaeng.org/publication/WCECS2008/WCECS2008%5C_pp1204-1208.pdf (cit. on p. 33).

[148] George Wald. "The Receptors of Human Color Vision Action spectra of three visual pigments in human cones account for normal color vision and color-blindness". In: *Science* 145.3636 (1964), pp. 1007–1016 (cit. on p. 50).

[149] Yue Wang, Eam Khwang Teoh, and Dinggang Shen. "Lane detection and tracking using B-Snake". In: *Image and Vision Computing* 22.4 (Apr. 2004), pp. 269–280. ISSN: 02628856. DOI: 10.1016/j.imavis.2003.10.003. URL: http://linkinghub.elsevier.com/retrieve/pii/S0262885603002105 (cit. on p. 29).

[150] JS Warner and RG Johnston. "GPS spoofing countermeasures". In: *Homeland Security Journal* (2003). URL: http://72.52.208.92/~gbpprorg/mil/gps4/GPS-Vulnerability-LosAlamos.pdf (cit. on p. 25).

[151] GR Widmann, MK Daniels, and Lisa Hamilton. *Comparison of Lidar-Based and Radar-Based Adaptive Cruise Control Systems*. 724. 2000. URL: http://connectedvehicle.itsa.wikispaces.net/file/view/Comparison+of+Lidar-Based+and+Radar-Based+2000-01-0345.pdf/456154696/Comparison%20of%20Lidar-Based%20and%20Radar-Based%202000-01-0345.pdf (cit. on p. 15).

[152] Wikipedia. *Dilution of Precision*. 2014. URL: http://en.wikipedia.org/wiki/Dilution%5C_of%5C_precision%5C_(GPS) (visited on 08/19/2014) (cit. on p. 20).

[153] Wikipedia. *Rolling Shutter*. 2010. URL: http://en.wikipedia.org/wiki/Rolling%5C_shutter (visited on 08/28/2014) (cit. on p. 28).

[154] PI Wilson and John Fernandez. "Facial feature detection using Haar classifiers". In: *Journal of Computing Sciences in Colleges* (2006), pp. 127–133. URL: http://dl.acm.org/citation.cfm?id=1127416 (cit. on pp. 30, 31).

[155] Lingyun Xiao and Feng Gao. "A comprehensive review of the development of adaptive cruise control systems". In: *Vehicle System Dynamics* 48.10 (Oct. 2010), pp. 1167–1192. ISSN: 0042-3114. DOI: 10.1080/00423110903365910. URL: http://www.tandfonline.com/doi/abs/10.1080/00423110903365910 (cit. on p. 15).

[156] Peng Yang et al. "Face recognition using ada-boosted gabor features". In: *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*. IEEE. 2004, pp. 356–361 (cit. on p. 29).

[157] Qingyu Yang, Liguo Chang, and Wei Yu. "On false data injection attacks against Kalman filtering in power system dynamic state estimation". In: *Security and Communication Networks* (Aug. 2013), n/a–n/a. ISSN: 19390114. DOI: 10.1002/sec.835. URL: http://doi.wiley.com/10.1002/sec.835 (cit. on pp. 45, 46).

[158] Wei Zhang and QMJ Wu. "Tracking and pairing vehicle headlight in night scenes". In: *IEEE Transactions on Intelligent Transportation Systems* 13.1 (2012), pp. 140–153. URL: http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=6018308 (cit. on p. 29).

[159] Mehrdad Ziari et al. "Photorefractivity in vanadium-doped ZnTe". In: *Applied physics letters* 60.9 (1992), pp. 1052–1054 (cit. on p. 93).