Deep learning for perception tasks

Gaisser, F.

DOI
10.4233/uuid:f88ae605-0f72-4638-b7c2-fc5a98996fc2

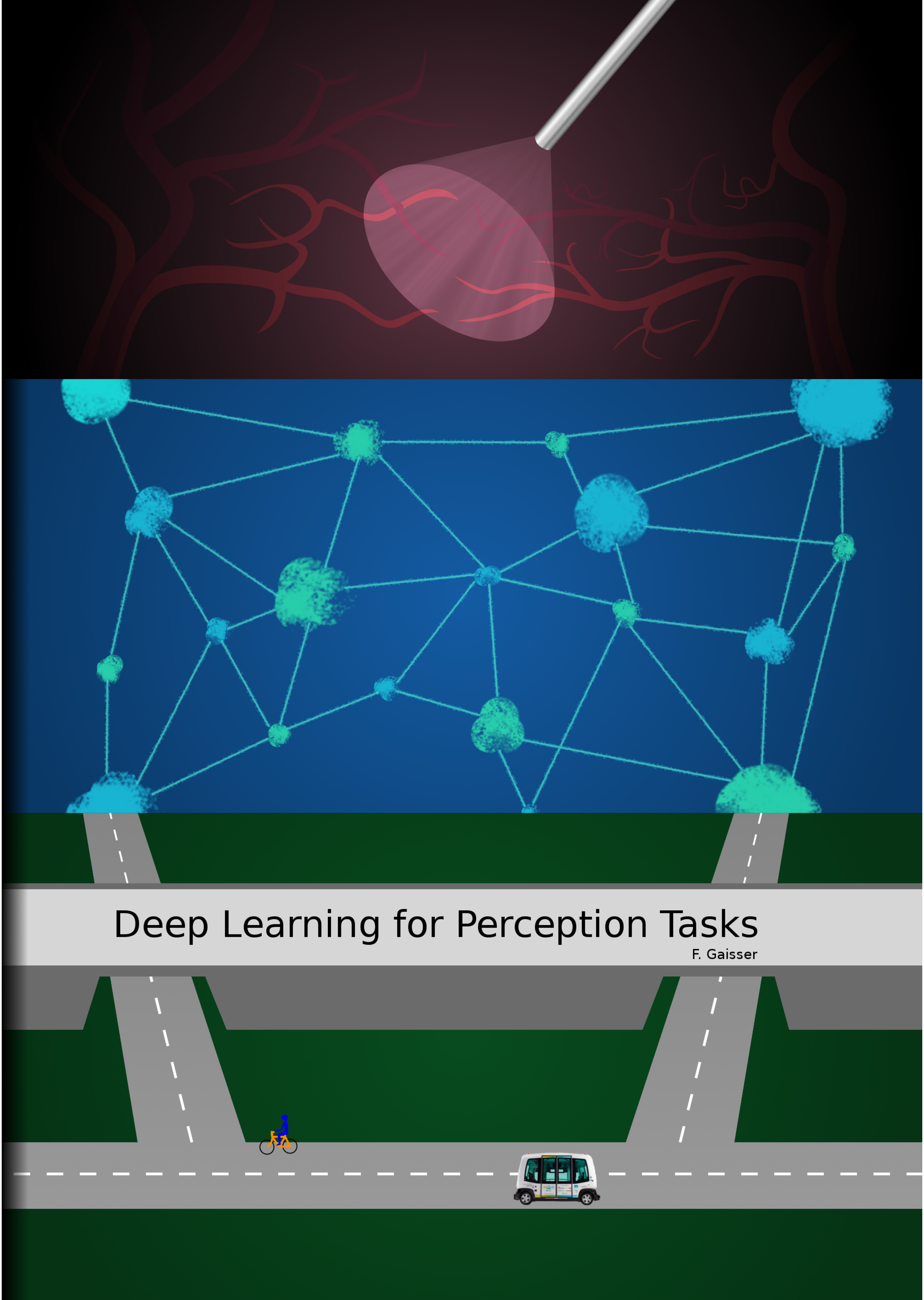
Publication date
2021

Document Version
Final published version

Citation (APA)
Gaisser, F. (2021). *Deep learning for perception tasks*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:f88ae605-0f72-4638-b7c2-fc5a98996fc2

Deep Learning for Perception Tasks

F. Gaisser



Deep Learning for Perception Tasks

F. Gaisser

Deep Learning for Perception Tasks

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, Prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board of Doctorates
to be defended publicly on
Monday 4 January 2021 at 10:00 o'clock

by

Floris GAISSER
Master of Science in Mechanical Engineering, Delft University of Technology,
the Netherlands
born in The Hague, the Netherlands

This dissertation has been approved by the promotors

Composition of the doctoral committee:

Rector Magnificus	chairman
Prof. dr. ir. P.P. Jonker	Delft University of Technology, promotor
Prof. dr. J. Dankelman	Delft University of Technology, promotor
Dr. ir. R. Happee	Delft University of Technology, promotor

Independent members:

Prof. dr. D. Oepkes	Leids Universitair Medisch Centrum
Prof. dr. D. Stoyanov	University College London
Prof. dr. A. Yarovoy	Delft University of Technology
Dr. M. Spaan	Delft University of Technology
Prof. dr. D.M. Gavrila	Delft University of Technology, reserve member

This research was financially supported by:

- **3D Fetoscopy** project as part of Topsector Life Science Health (LSH) / Innovative Medical Devices Initiative (IMDI) / Novel Instruments For Minimally Invasive Techniques (NIMIT).
- **WEpods** project as part of the Food Valley initiative of the Province of Gelderland and the municipalities of Ede and Wageningen.
- **I-AT** the Inter-regional project on Automated Transport of the European Union.

Printed by: Optima

ISBN 978-94-6361-503-7
A digital copy of this thesis can be downloaded from http://repository.tudelft.nl.

Contents

Summary

In recent years large advances have been made in the field of machine learning, driven by novel deep learning methods. Deep learning is a research field that focusses on creating neural networks. This field has seen a rapid advance due to an increase in computational power, availability of large amounts of data and a wide variety of novel methods that allows for more efficient training of neural networks. Deep learning has been applied in various fields to solve many different tasks. Effective training of these neural networks requires selecting the right data, network architecture and learning method. However, thorough understanding of the task for which the neural network is trained is needed to adhere to these requirements. This thesis will illustrate that deep learning methods can effectively be applied to perception tasks by thorough understanding of the task.

Part I: Region Detection and Feature Extraction in Fetoscopy

The Twin-to-Twin Transfusion Syndrome (TTTS) is a condition of unborn twins that requires a complex laproscopic surgical procedure on the placenta that is complicated by the visual conditions inside the uterus. In **Chapter 2** TTTS is described and the surgical procedure is detailed. Only few highly trained surgeons can perform this complex and mentally challenging procedure. Therefore, this procedure can only be performed in a very select number of hospitals. Alternatively, abortion of one of the twins is a method to treat this complication as the mortality rate is extremely high if left untreated, resulting in many abortions or unsuccessful pregnancies on a yearly basis. The preferred procedure is complicated requiring the surgeon to create a map of the placenta before laser coagulation of the anastomoses. Creation of such a map can be done by panorama reconstruction of the fetoscopic images. This method has been successfully been applied to fetoscopic images obtained in an ex-vivo setting.

However, these methods fail in an in-vivo setting as described in **Chapter 3**. In this chapter the challenges encountered and their underlying cause are detailed further. Also, the state-of-the-art methods used in an ex-vivo setting are evaluated for panorama reconstruction in the in-vivo setting. This chapter is concluded with recommendations to solve the challenges encountered in an in-vivo setting, which are adopted in the following chapters.

In **Chapter 4** the challenge of extracting matchable features is tackled by coupling similarity learning to the matching performance. In **Chapter 5** the Single Shot Detection method is adapted to detect stable regions on the veins in order to tackle the challenge of detecting stable keypoints. Also, the matchable features method of the previous chapter is integrated. Furthermore, a qualitative measure of the visibility condition is extracted that is indicative of the registration performance and allows to create a more reliable panorama reconstruction process.

Part II: Road User Perception in Automated Driving

The field of Automated Driving focuses on developing methods and applications that can support the driver in driving a car and ultimately completely take over the role of the driver. The research in this field is fuelled by the goal to reduce the amount of lethal accidents of which most can be attributed to human error. In **Chapter 6** the challenges of this field are detailed and the road user perception task is described. This task consist of three subtasks; road user detection, tracking and prediction. Effective detection of road users has been developed over the years with the use of deep learning algorithms and choice of the right sensors of which an example is given in **Chapter 7**. Traditionally in road user prediction, motion history is used to predict the future trajectory of the road user. However, the motion of a road user are the results of decisions that are based on constraints such as the road geometry and other road users. In **Chapter 8** the motion information is transformed to the road geometry, and combined with a Recurrent Neural Network (RNN), the future trajectory can be predicted much more accurately. In **Chapter 9** the motion information is encoded with an RNN, and the interaction between road users is modelled through social pooling. The road geometry is incorporated through an attention mechanism. This novel approach alleviates the two most important constraints in the motion model to predict the future trajectory of road users.

The applications of deep learning as described in this thesis illustrate that thorough understanding of the task improves the effectiveness of the applied deep learning methods. In the field of TTTS the thorough analysis of the task showed that stable keypoint detection and matching is important in panorama reconstruction. This analysis also showed what challenges are encountered in the in-vivo setting compared to the ex-vivo setting. These challenges posed different requirements to the applied methods and by including these in the design of deep learning based system of algorithms it was possible to overcome these challenges. The deep learning methods presented in this thesis for panorama reconstruction open the road to in-vivo application.

Furthermore, for road user detection analysis of the task showed that combining the strengths of the various sensors would improve the performance significantly. This conclusion has been made independently in other research. Analysis of the road user prediction task showed that there are indicators for the future trajectory, such as road geometry and interaction between road users, that are not effectively incorporated. The deep learning methods presented in this thesis for road user prediction improve accuracy and applicability for automated vehicles.

Samenvatting

Recentelijk zijn grote stappen gemaakt in het veld van *Machine Learning*, gedreven door ontwikkeling van nieuwe *Deep Learning* methodes. Deep learning is een onderzoeks veld dat zich richt op ontwikkeling van neurale netwerken. Dit veld heeft snelle ontwikkelingen ervaren door meer rekenkracht van computers, beschikbaarheid van grote hoeveelheden gegevens en een verscheidenheid aan nieuwe methodes waardoor neurale netwerken efficiënter getraind kunnen worden. Deep learning wordt toegepast in verscheidene toepassingen om veel verschillende taken op te lossen. Effective training van deze neurale netwerken vereist selectie van de juiste data, netwerk architectuur en leermethode. Echter, diepgaand begrip van de taak waar voor het neurale netwerk getraind wordt, is nodig om aan deze vereisten te voldoen. In deze thesis zal beschreven worden dat deep learning methodes effectief toegepast kunnen worden op perceptie taken door diepgaand begrip van de taak.

Deel I: Regio Detectie en Feature Extractie in Fetoscopy

Het *Twin-To-Twin Transfusion Syndrome* (TTTS) is een aandoening bij ongeboren tweelingen welke een complexe laparoscopische chirurgische procedure benodigd op de placenta. Deze procedure wordt bemoeilijkt door het zicht in de uterus. In **hoofdstuk 2** wordt TTTS en de chirurgische procedure in detail beschreven. Enkel een paar goed opgeleide chirurgen kunnen deze complexe en mentaal uitdagende procedure uitvoeren. Daarom wordt deze procedure maar in een paar ziekenhuizen aangeboden. Als alternatief zou abortie van één van de tweeling zijn, aangezien de overlevingskans erg laag is, indien deze aandoening niet behandeld wordt, waardoor jaarlijks veel abortes en mislukte zwangschappen gebeuren.

De voorkeurs procedure is gecompliceerd en vereist dat de chirurg een kaart maakt van de placenta voordat deze met een laser de anastomoses coaguleert. Het maken van zo'n kaart kan ook gedaan worden door panorama reconstructie van de fetoscopische beelden. Deze methode is succesvol toegepast met fetoscopische beelden verkregen in een ex-vivo setting.

Echter, deze aanpak faalt in een in-vivo setting zoals beschreven wordt in **hoofdstuk 3**. In dit hoofdsuk worden de uitdagingen en hun onderliggende oorzaken uitgebreid beschreven. Verder worden de nieuwste methodes in de ex-vivo setting geëvalueerd voor panorama reconstructie in de in-vivo setting. Dit hoofdstuk sluit af met aanbevelingen hoe de beschreven uitdagingen in een in-vivo setting zouden overwonnen kunnen worden.

In **hoofdstuk 4** wordt de uitdaging, om de passende beschrijving van referentie punten te verkrijgen, aangepakt door de overeenkomstigheid van de gekoppelde beschrijvingen van referentie punten aan *similarity learning* te koppelen. In **hoofdstuk 5** wordt

de *Single Shot Detection* (SSD) methode aangepast zodat deze stabiele gebieden op de bloedvaten van de placenta kan detecteren. Dit wordt gebruikt om een oplossing te vinden om stabiele referentie punten te vinden. Verder wordt de methode van het vorige hoofdstuk geïntegreerd. Tot slot wordt een maatstaf van de zicht kwaliteit gebruikt om indicatie te verkrijgen van de registratie prestatie. Dit leidt tot een betrouwbaar panorama reconstructie proces die toegepast zou kunnen worden in een in-vivo setting.

Deel II: Weggebruiker Perceptie in Geautomatiseerd Rijden

Onderzoek naar geautomatiseerde voertuigen richt zich op ontwikkeling van methodes en toepassingen die de bestuurder van een auto ondersteunt en uiteindelijk volledig de taak van bestuurde kan overnemen. Onderzoek hierin is gemotiveerd door het doel om het aantal dodelijke ongelukken te verminderen. Hiervan kunnen de meeste verweten worden aan menselijke fouten. In **hoofdstuk 6** worden de uitdagingen in dit onderzoeksveld gedetailleerd. Daarnaast wordt de weggebruiker perceptie *(road user perception)* taak gedetailleerd welke bestaat uit drie subtaken: detectie, volgen en voorspelling *(recognition, tracking, prediction)*.
Effective detectie van weggebruikers is ontwikkeld in de laatste jaren door gebruik te maken van deep learning algoritmes en de juiste keuzes in sensoren. Hiervan is een toepassing gegeven in **hoofdstuk 7**.
Oorspronkelijk werd in voorspelling van de acties van een weggebruiker de bewegingsgeschiedenis gebruikt. Echter, de acties van een weggebruiker zijn het resultaat van genomen besluiten die gebaseerd zijn op andere informatie zoals de weggeometrie en andere weggebruikers. Daarom wordt in **hoodstuk 8** de bewegingsgeschiedenis getranformeerd naar de weggeometrie en gecomobineerd met een *Recurrent Neural Network* (RNN) om de toekomstige route beter te kunnen voorspellen.
In **hoofdstuk 9** wordt de bewegingsgeschiedenis gecodeerd door een RNN en wordt de interactie tussen weggebruikers gemodelleerd door *social pooling*. De weggeometrie wordt geïntegreerd met een *attention mechanism*. Deze vernieuwende aanpak maakt gebruik van de twee meest belangrijke onderdelen in de besluitvorming van weggebruikers en zijn daarom ook de belangrijkste aspecten om de acties van weggebruikers te voorspellen.

De toepassing van deep learning zoals beschreven in deze thesis toont aan dat diepgaand begrip van de taak de effectiviteit verbeterd waarmee de deep learning methodes wordt toegepast.
Bij TTTS is door grondige analyse van de taak aangetoond dat het vinden van stabiele herkenningspunten en het koppelen van deze van belang zijn voor panorama reconstructie. Deze analyse bracht tevens aan het licht welke uitdagingen men tegen

komt in de in-vivo setting ten opzichte van de ex-vivo setting. Deze vereisen een andere manier van toepassing van de methodes. Door deze te integreren in het ontwerp van het op deep learning gebaseerde systeem, was het mogelijk om deze uitdagingen te overwinnen. De deep learning toepassingen zoals gepresenteerd in deze thesis openen de deur naar toepassing in een chirurgische toepassing.

In de analyse van de weggebruiker detectie taak werd gevonden dat het bundelen van de krachten van verschillende type sensoren de prestatie significant zou verbeteren. Analyse van de taak waarin de acties van weggebruikers voorspeld wordt, toont aan dat er indicatoren zijn, zoals de weggeometrie en interactie tussen de weggebruikers, die het voorspellend vermogen kunnen verbeteren. De deep learning methodes die gepresenteerd worden in deze thesis, verbeteren de toepassingsmogelijkheden voor geautomatiseerde voertuigen.

1

Introduction

Between March 9 and 15, 2016 the world champion in Go, Lee Sedol, was beaten in a five-game match by the computer program *AlphaGo*, made by DeepMind [115]. This is considered one of the groundbraking developments of 2016. But why is this so impressive? In the years before many games have been conquered by computers. For example, *Deep Blue* beating the chess world champion in 1997 is one of the well known examples [20]. So why is *AlphaGo* then so different?

First, look at how these other games are solved. Computer programs rapidly explore (part of) the tree of possible moves and then selecting the best move. Chess offers 20 possible moves in the first turn and after 6 moves there are about 9 million possible positions and flattens out to 10 million possible positions after that. In total chess has a maximum of 10^{43} possible positions [114], though these are never completely explored. *Deep Blue* could win from the world champion by exploring up to 20 moves ahead with evaluating about 200 million positions per second by shear brute-force computation power.

In contrast *Go* is played on a field of 19×19 resulting in 361 possible moves in the first turn. Therefore, it has after two moves 130.000 and after four moves 17 billion possible positions. In total Go has about $2.082 \cdot 10^{170}$ legal positions [134] which is more than the number of atoms in this universe and would take many times longer to compute than this universe has existed. Go is so complex because the search tree expands so fast that exploring it in a brute-force way is out of the question. So how is it done?

AlphaGo has two neural networks. The first finds possible moves that look promising, thus reducing the number of branches that have to be evaluated. The second network evaluates the selected moves by not calculating the complete branch expansion, but by learning the value of a certain position through experience. This is similar to how humans play; we only consider good moves and plan ahead by knowing what positions can give an advantage. Thus *AlphaGo* could win by learning the game similar to how humans play; by learning to recognize good positions and strategies to win.

To make life more comfortable humans try to make machines taking over tasks. One approach would be to make machines learn these tasks by themselves. Because if a human can learn it, shouldn't a machine be able to learn this too with the right methods? However, to successfully achieve this, it is required to understand three things first; What are the tasks at hand and how to structure these tasks. How do humans learn these tasks, and how can machines learn these tasks. Appendix A describes these three aspects in detail, in this introduction a short summary is given.

Many tasks can be described according to the *Sense-Think-Act* paradigm (Appendix A.1.1) [16, 87]. In *Sensing* the environment is observed and the relevant information is extracted. The *Thinking* step uses the relevant information to make an abstraction into a meaning and a decision is made. Finally, the *Acting* transforms the decision into one or more actions. Often tasks can be described as the hierarchical combination of multiple tasks, forming complex tasks. Furthermore, the division of these (complex) tasks into smaller tasks, allows reuse of other tasks and makes it easier to learn a complex task in steps.

Humans have various methods to learn tasks and the methods relevant to machine learning are detailed (Appendix A.1.2). *Deductive learning* is where a rule is explained and its application is learned by the student [77]. For machines this can be translated as programming a rule. Programming a machine is how commonly a machine is made to perform its task, though generally this does not involve learning.
Learning by Example, one of the forms of *inductive learning*, provides the student with many examples of a task and the desired outcome of the task [12]. The student can then learn to extract the important information (Sensing), reason about it (Thinking) and take actions to achieve the desired outcome of the task (Acting). This is one of the most common methods in machine learning also referred to as *supervised learning*.
Learning by Doing is a set of learning methods where the student learns the task by performing it [102]. *Trial and Error* is a method where through failed and successful attempts to achieve the task, the task can be learned [132]. The *Contrastive Loss* method has similarities with this approach [45]. Furthermore, *Exploration* is where the student knows (in approximation) how to perform the task, but by exploring variations of the task he/she learns a more optimal method to perform the task. In machine learning *reinforcement learning* is a method that explores the action or parameter space and optimizes the cost or reward function to improve the performance of the task [18, 105, 123]. *Transfer Learning* is a method where the knowledge of a task is used in learning another task [89, 90]. This approach has found wide application in machine learning as it is fairly easy to reuse knowledge on a machine.

Deep Learning is a field of machine learning that focusses on creating neural networks similar to neurons in the human brain (Appendix A.2.1). Recent advances in deep learning have been fuelled by many factors such as increase in computational power, availability of data and a wide variety of novel methods (Appendix A.2.2). The application of deep learning was not limited to a single field, but has found application in many different tasks, such as image understanding, object recognition, text understanding and translation, speech recognition and many more.

Combining the understanding of how to structure a task, how humans learn these and how to train deep learning neural networks, three requirements can be described to effectively learn perception tasks (Appendix A.2.4). First, the input *data* of a neural network is the source of everything, as it contains the relevant data from which the meaning for the decision making has to be extracted. Second, the *architecture* of a network is of great importance, since it needs to support the abstraction, decision making and output structure that is needed for the task. Third, the *learning method* is of great importance, as it updates the parameters of the network, such that it can abstract the data into the desired output.

1.1 Problem Definition

Effective application of deep learning methods to all type of tasks that adhere to the Sense-Think-Act paradigm is challenging. Therefore, in this thesis a focus is made on perception tasks, which encompass tasks that involve at least the *Sense* step of the Sense-Think-Act paradigm. Furthermore, in this thesis the statement is made that through correct application of deep learning methods it is possible to learn these type of tasks. In order to achieve this it is hypothesised that at least a thorough understanding of the task is needed in order to adhere to the requirements to effectively apply deep learning methods. However, also certain advances are still needed in deep learning to actually achieve this;
First, the human brain consists of about 86 billion neurons which cannot be modelled at the speed the human brain is functioning. Therefore, more advances in computational power and in modelling efficiency is needed.
Second, currently it is not possible to accurately determine how these billions of neurons are connected. Therefore, a better understanding is needed of how to structure neural networks and what architectures are effective.
Third, humans learn non-stop and any number of tasks, whereas currently deep learning is applied to a selective subset of tasks. Therefore, transfer learning is not as well applied as in humans. However, with better understanding of how to transfer knowledge from one task to another this can be improved.

1.1.1 Goal

Through understanding on how to effectively apply deep learning to tasks, advances can be made in the latter two challenges. Therefore, the main goal of this thesis is:

Illustrate that deep learning can effectively be applied to perception tasks through thorough understanding of the task.

In order to achieve this, two projects in different domains have been chosen; Fetoscopic surgery in Twin-to-Twin Transfusion Syndrom (TTTS), and road user perception in autonomous vehicles. In both fields specific complex tasks have been identified and through analysis of these tasks, deep learning methods have been successfully applied to effectively solve these tasks.

1.1.2 Panorama Reconstruction for Twin-to-Twin Transfusion Syndrome

According to the World Health Organization about 211 million pregnancies occur per year [139]. Of these pregnancies about 1% are twin gestations resulting in half a million twins that are in risk of Twin-to-Twin Transfusion Syndrome world wide every year. This complication can be treated with fetocsopic laser coagulation [27, 78, 101, 111]. However, this procedure requires highly trained and skilled surgeons and is thus only available in a very select number of hospitals [83]. Alternatively, abortion of one of the twins is a method to treat this complication as the mortality rate is $80 - 100\%$ if left untreated [49, 94, 111]. Therefore, on a yearly basis close to a million babies die due to abortion or unsuccessful pregnancy [68]. In order to reduce this number of unnecessary deaths the procedure to treat this complication should be made more widely available. In Chapter 2 the challenges of TTTS are described in more detail and it is shown how panorama reconstruction can support the surgeon in the treatment of TTTS. Panorama reconstruction is a perception task that can benefit from effective application of deep learning methods. Therefore, part of this thesis focusses on creating an overview of the placenta by applying of deep learning, which significantly reduces the complexity of the surgery.

1.1.3 Road User Perception in Automated Driving

In a recent report of the World Health Organization on road safety about 1.35 million people die every year in traffic accidents [140]. This motived the United Nations to adopt a goal to reduce this number by 50% by 2030. To achieve this goal vehicles must become more safe by introducing driver assistance systems, as 65% and possibly up to 92.5% of these accidents can be attributed to human error [133]. Advanced driver assistance systems (ADAS) consist of various tasks, of which many involve other road users. In Chapter 6 the road user perception task is detailed further. Therefore, part of this thesis focusses on improving road user perception through effective application of deep learning methods as this is one of the most challenging tasks in automated driving.

1.2 Contributions and Thesis Outline

Based on the two chosen application fields, this thesis is divided into the two aforementioned parts, followed by a discussion and conclusion on how the used deep learning methods were applied across the two domains. The contributions and outline of the thesis can be described as follows:

Part I - Region Detection and Feature Extraction in Fetoscopy

Chapter 2 *Panorama Reconstruction for TTTS*
This chapter gives an introduction to the medical aspects of fetoscopy for TTTS and the challenges encountered in its treatment. It is reasoned that a panorama reconstruction of the placenta will support the surgeon. It is shown how the panorama reconstruction task can be spit into smaller tasks and how deep learning can be applied to obtain a complete overview of the placenta.

Chapter 3 *Requirements for In-Vivo Panorama Reconstruction*
Current state-of-the-art work on panorama reconstruction of the placenta all focus on ex-vivo obtained data. However, to support the surgeon during the procedure, in-vivo data has to be processed in real-time. Therefore, in this chapter the challenges that are posed by in-vivo data are detailed. Furthermore, it is shown that the current methods applied to ex-vivo data fail on in-vivo data. Lastly, recommendations are provided to successfully process in-vivo data.

Chapter 4 *Matchable Feature Extraction for Image Registration*
One of the challenges posed by image registration is the matching of keypoints. Since, most feature extraction methods are designed to be generic to all sources of data, they are less suited for fetoscopic data. In this chapter it is shown that a Convolutional Neural Network can be trained to extract features that are better suited for image registration.

Chapter 5 *Stable Region Detection for Image Registration*
One challenge of fetoscopic image registration is obtaining stable keypoints for matching. By redefining the problem and using the object detection network, *SSD*, a keypoint detection network is trained to find stable keypoints on the placenta. Using the previously introduced feature extraction learning method, the whole image registration up to the transform estimation is defined in a neural network.

Part II - Road User Perception

Chapter 6 *Road User Perception in Automated Driving*
This chapter gives an introduction to the field of automated vehicles and specifically the aspect of Road User Perception. The aspect of road user perception covers three different tasks; Recognition of road users, Tracking of the state over time of these road users, and Predicting the state into the future such that actions of the ego-vehicle can be planned. Deep learning has already found application in these tasks, though can still benefit greatly from more effective application of advances in deep learning.

Chapter 7 *Radar Detection and Camera Classification*
One of the challenges of road user perception is to reliably recognize road users. By leveraging the advantages of different type of sensors, this chapter shows that the detection performance can be improved over vision-only methods and allows for real-time on-vehicle deployment by using deep learning methods.

Chapter 8 *Trajectory Prediction within Infrastructure*
Generally, vehicles will drive only on the road, especially in an urban setting. Therefore, this chapter introduces a change in the input data format inspired by analysis of the task as performed by humans. Furthermore, a neural network architecture originating from natural language processing is used to predict the future trajectory of these vehicles.

Chapter 9 *Trajectory Prediction with Interaction and Road Attention*
Generally, vehicles are not the only vehicle driving on a road. Therefore, this chapter introduces a new architecture to model the interaction between vehicles to improve the predicted trajectory. Furthermore, this new architecture allows to combine both road geometry and interaction, such that most of the features used in trajectory prediction are considered.

Part I

Region Detection and Feature Extraction in Fetoscopy

The Twin-to-Twin Transfusion Syndrome (TTTS) is a condition of unborn twins that requires a complex laproscopic surgical procedure on the placenta that is complicated by the visual conditions inside the uterus. Therefore, this part of the thesis details deep learning algorithms that can support the surgeon in performing this procedure. In Chapter 2 first TTTS and how panorama reconstruction can support the surgeon are described. Next, the challenges posed by this procedure on the panorama reconstruction algorithms are detailed. Finally, is concluded how deep learning methods can improve the panorama reconstruction performance. In Chapter 3 the challenges for in-vivo panorama reconstruction are further detailed, and compared to the existing methods found in literature for ex-vivo panorama reconstruction. In the following chapters possible algorithms to tackle these challenges are detailed. In Chapter 4 a novel approach is proposed using similarity learning that combines *learning by doing* and *trial and error* to increase the performance in extracting matchable features by a factor 3. In Chapter 5 the challenge of finding stable keypoints is redefined by using the Single Shot Detection method to detect stable regions. Furthermore, by extracting a qualitative measure and combining this with the stable region detection and matchable feature extraction, the panorama reconstruction process can be optimized. These novel methods improve the panorama reconstruction of the placenta such that it can support the surgical team in performing the procedure.

2

Panorama Reconstruction for Twin-to-Twin Transfusion Syndrome

Abstract

The Twin-to-Twin Transfusion Syndrome (TTTS) is a condition of unborn twins that requires a complex laproscopic surgical procedure on the placenta that is complicated by the visual conditions inside the uterus. In this chapter TTTS is described in more detail and explained how the laproscopic procedure can be described following the Sense-Think-Act paradigm. Furthermore, it is shown how panorama reconstruction can support the surgeon in performing the procedure. Finally, is described how deep learning methods can improve the panorama reconstruction performance.

2.1 Introduction

About 1% of all pregnancies are twin gestations, of which 30% are identical (monozygotic) twins. These type of twins occur when a single fertilized egg (zygote) splits into two separate embryos. When this happens after the second day (75%), these twins will share a single placenta (monochorionic) and in most cases will have separate amniotic sacs (diamniotic). The differences and timing of twins is shown in Figure 2.1. Whether twins share a placenta (chorionicity), rather than zygosity determines the outcome of the pregnancy. Monochorionic (MC) twins have a twice higher risk of adverse perinatal outcome compared two other diachorionic twins, due to complications of sharing a single placenta [44, 67].

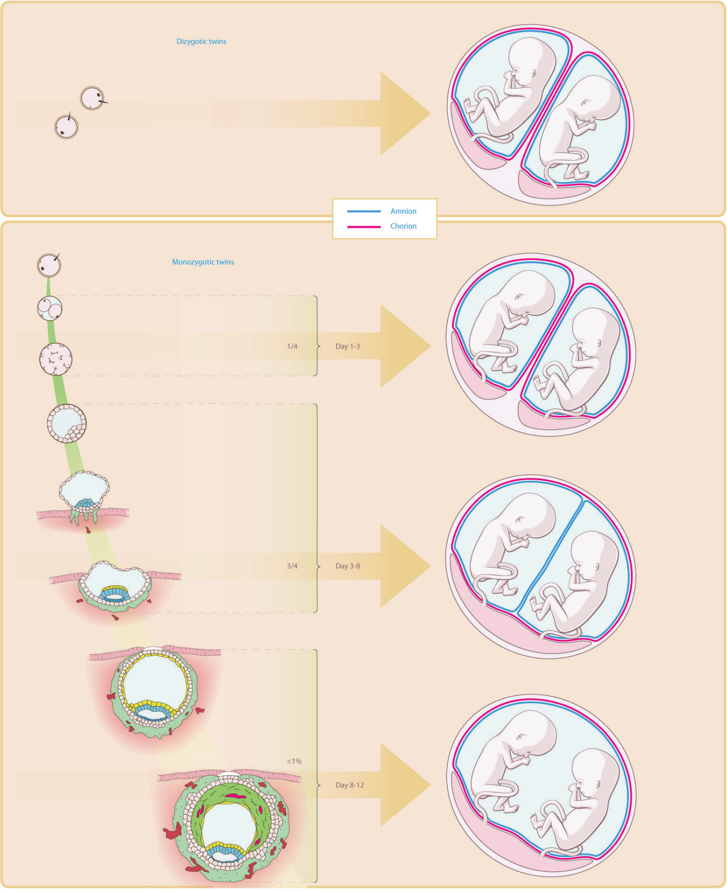


Figure 2.1: Differences in twin pregnancies. (adopted from [83])

2.2 Twin-to-Twin Transfusion Syndrome

One of the complications that only occurs in MC pregnancies is the Twin-To-Twin Transfusion Syndrom (TTTS), occurring in about 10% of pregnancies [67, 68]. This syndrome complicates the pregnancy by an imbalance of blood between the twins. Blood flows from the foetus to the placenta and back to exchange nutrients and by-products with the mother. However, an exchange of blood occurs through connecting vessels between the twins on the placenta (vascular anastomoses). This syndrome occurs when an imbalance is created when an artery is donating blood to a vein of the other twin and is not compensated by other vascular anastomoses. This is also illustrated in Figure 2.2

This syndrome is the direct result of transfusion from the donor to the recipient twin. This causes in the donor twin a decreased blood volume and decreased urinary output. Which leads to a low level of amniotic fluid and kidney problems. While for the recipient twin the increased blood volume results in a higher urinary output and can lead to heart failure. If this syndrome is left untreated it will result in a high risk of intrauterine death, premature birth and miscarriage, with a mortality rate of 80–100% [49, 94, 111].

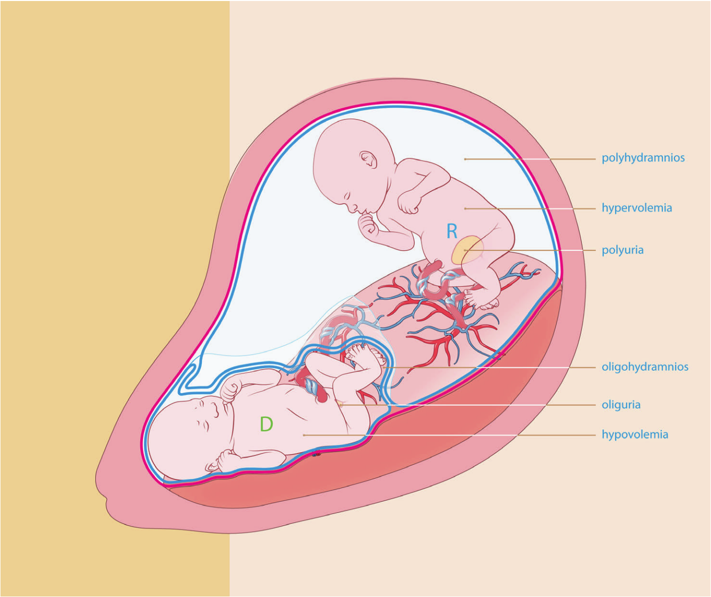


Figure 2.2: Twin-to-Twin Transfusion Syndrome (TTTS). D is donor and R is recipient twin. (adopted from [83])

This syndrome can be treated by abortion of a twin, which increases the survival rate of the remaining twin, but has always an adverse perinatal outcome compared to other therapies. Another method for treatment is to (repeatedly) drain excess amniotic fluid through a needle that is passed into the sac of the recipient (amniodrainage). However, causal treatment with fetoscopic laser coagulation of the vascular anastomoses is the preferred procedure, as it has the highest success rate [27, 78, 101, 111].

2.3 Fetoscopic Laser Coagulation

Fetoscopic laser coagulation is a procedure where the vascular anastomoses are coagulated with a laser in order to separate the blood flow of the twins as shown in Figure 2.3. After this treatment the survival rate of both twins increases to 35–67% [25, 111, 135] although some of the surviving babies show other complications. The success of this therapy depends on coagulation of all vascular anastomoses and this so called Solomon technique has shown significant reduction in remaining vascular connections and resulting complications [118, 119]. Figure 2.4 shows laser coagulation with the Solomon technique.

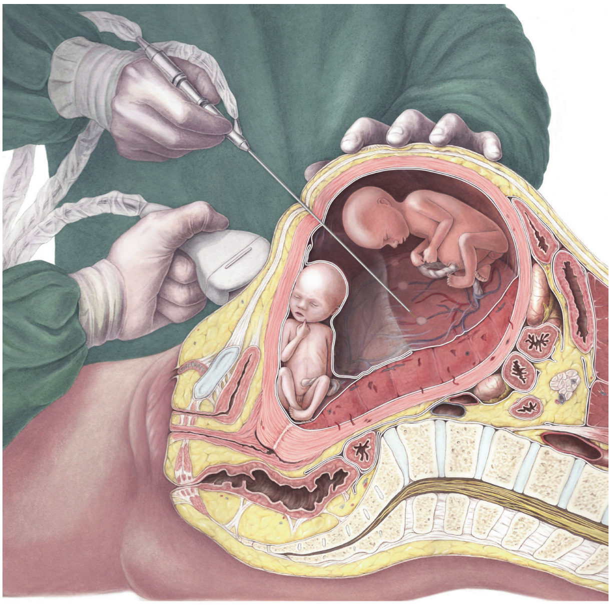


Figure 2.3: Fetoscopic laser coagulation. (adopted from [83])

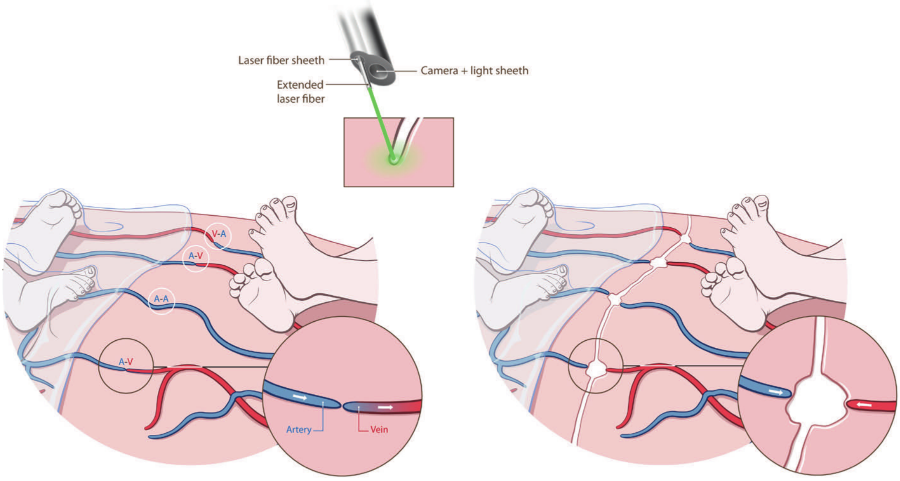


Figure 2.4: Laser coagulation with the Solomon technique. (adopted from [83])

Even though fetoscopic laser coagulation therapy has been around for 25 years, it is still offered in a limited number of highly specialized hospitals. Furthermore, even the most experienced surgeons show a high percentage of adverse perinatal outcome. This can be partly attributed to the complexity and the learning curve of the procedure [83]. To better understand this procedure it can be described as a task in the Sense-Think-Act paradigm as shown in Figure 2.5. On the most abstract level, the surgeon has to find the anastomoses and determine if these are vein-to-vein, vein-to-artery, artery-to-vein or artery-to-artery connections. For the thinking step, the surgeon or its team should determine the right order of coagulating the anastomoses and where to perform the Solomon line. Finally, the surgeon has to guide the fetoscope to these anastomoses and perform the laser coagulation.

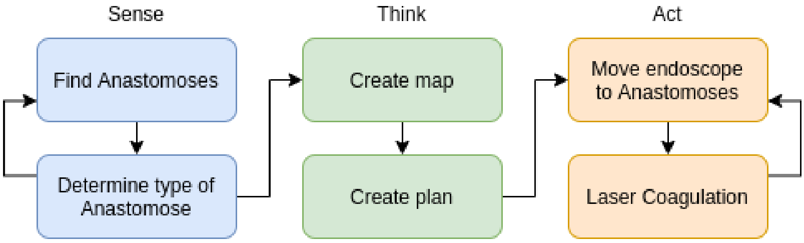


Figure 2.5: Sense-Think-Act description of surgery task.

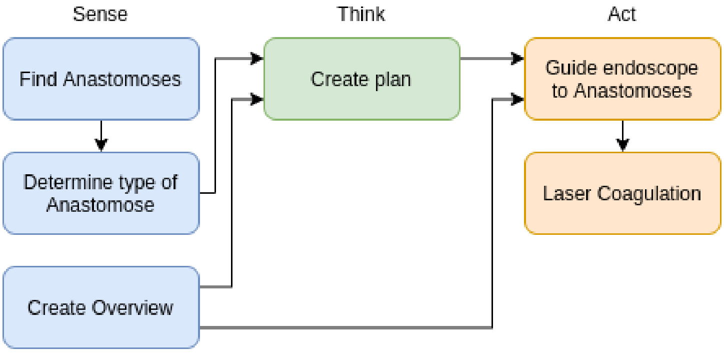


Figure 2.6: Sense-Think-Act description of surgery task with an overview available.

2.3.1 Panorama Reconstruction Supporting Surgeon

Finding these anastomoses initially, as well as finding them again for coagulation is a complex subtask. The surgeon generally starts at the umbilical cord and follows the branching veins and arteries. While doing this, a map has to be created of the veins and the found anastomoses. Sometimes somebody in the surgical team makes notes, though many surgeons memorize this map. This is a highly skilled and taxing task caused by the limited overview of the placenta. It has been suggested that obtaining a complete overview of the placenta will reduce the complexity of the procedure. Obtaining such an overview is limited by the viewing angle of the fetoscope, the in-uterine visibility condition and occlusion of the placenta by the fetus. This overview will simplify finding all vascular anastomoses and the surgery can be described as a hierarchical Sense-Think-Act model involving cooperation of man and machine creating as described in Figure 2.6. The reduced complexity of the surgery will shorten the learning curve, which will allow this procedure to be performed in a larger variety of hospitals and more readily available as less specialized and trained surgeons are required.

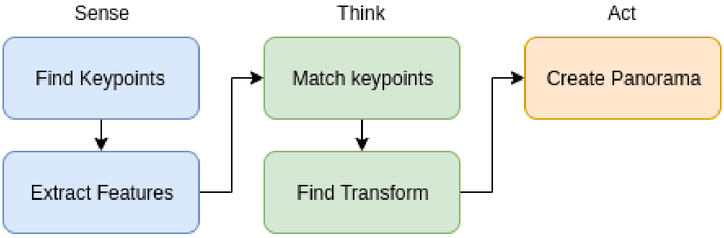


Figure 2.7: Sense-Think-Act description of creating an overview subtask.

2.4 Panorama Reconstruction of Fetoscopic images

Such an complete overview of the placenta can be obtained by panorama reconstruction of the fetoscopic images. This task can be detailed in the following steps:

1. Obtaining keypoints in images
2. Extract a feature describing each of these keypoints
3. Match the keypoints between two successive image
4. Find the transform based on the found keypoint matches
5. Combine all images into one large panorama based on the found transforms

The subtask of sensing consists of step one and two, thinking is step three and four and, step 5 is the acting subtask as shown in Figure 2.7. These steps are generally the same though can use different methods to achieve this task. An example of a reconstructed panorama viewing part of the placenta is shown in Figure 2.8.

Reconstructing large view panoramas of the internal anatomical structures has been a large field of research and found many applications, such as for retina [113], bladder [120], oesophagus [21] as well as ex-vivo fetoscopic [72, 99, 131]. Even though these approaches seem promising, they are not directly applicable to in-vivo fetoscopic panorama reconstruction.

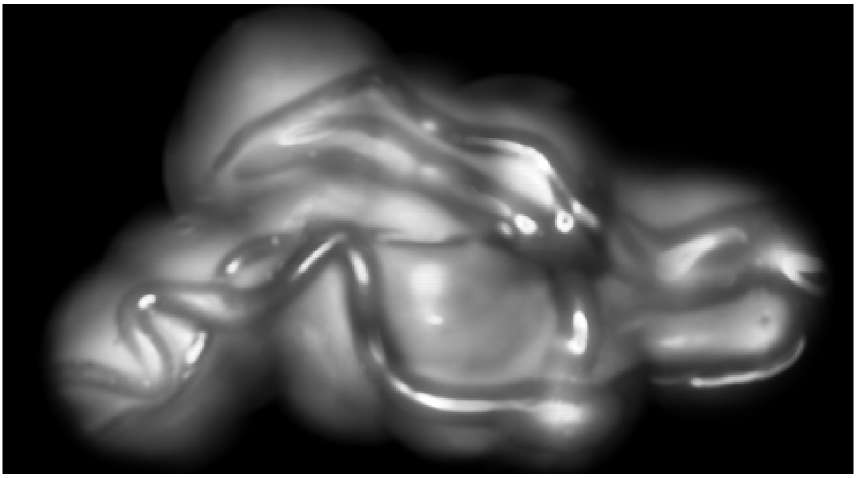


Figure 2.8: Reconstructed panorama viewing part of the placenta, as obtained in this thesis.

First, Reeff et. al. show the reconstruction of a small part of an ex-vivo placenta [99]. The results show that the image registration has a low accuracy and that reconstruction without post-processing contains many artefacts. Furthermore, the images are captured by moving the camera sideways in a structured circular pattern. This cannot be reproduced in an in-vivo setting, as the possible motion is only rotation around the entry point. Also, the used motion results in translational transformations between images which can be robustly estimated with existing methods.
Second, Liao et. al. project endoscope images of a color injected placenta on a 3D ultrasound model, which show accurate results in image registration [72]. However, such a setting with an ex-vivo color injected placenta is not compatible with our goal of in-vivo surgery.
Third, there are promising results in other applications areas such as bladder reconstruction [120], in which an ex-vivo dye injected bladder is reconstructed from a flexible endoscope with image registration, bundle adjustment and spherical projection. However, also here this method is not suited for our setting, as no prior structure is available. Furthermore, the encountered transformations between successive images here and in Seshamani et. al. [113] are also mostly translations.
Last, by Carroll et. al. an accurate reconstruction in oesophagus reconstruction is presented [21]. Pipe projection is used here, which is also not applicable to our setting. Furthermore, spatial consistency is not required for this type of reconstruction.

The ultimate goal is applying fetoscopic panorama reconstruction to TTTS surgery, thus a shift from ex-vivo to in-vivo research is necessarily. However, there are significant differences between the ex-vivo and in-vivo settings which pose challenges that have not yet been resolved:

1. Obtaining in-vivo fetoscopic videos is challenging, since data can only be obtained from living subjects and extending the length of the procedure increases the risks significantly.

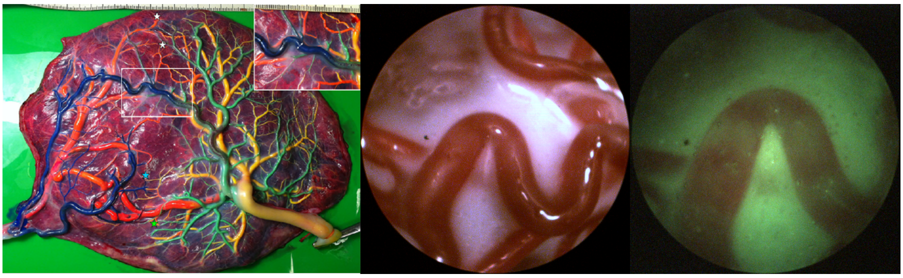


Figure 2.9: Example of a) dye injected placenta, b) ex-vivo, and c) in-vivo fetoscopic view

2. The motion of the fetoscope is limited around the entry point of the body resulting in perspective transformations between images. These are much harder to estimate compared to translation transformations between images which are allowed by free motion.
3. For panorama reconstruction a projection surface is required. However, the placenta is deformable and inside of the body it has no general shape. In contrast to outside the body, the placenta can be placed on a flat surface and a plane projection can be used.
4. The contrast between the veins and the placenta is small. In the ex-vivo setting this has been resolved by dye injection of the placenta.
5. The visibility conditions caused by the in-vivo setting is not considered in the ex-vivo setting and simplifies the challenge of panorama reconstruction significantly.

2.4.1 Requirements for Panorama Reconstruction

To resolve these challenges, the following observations can be made:

- Currently, surgeons are trained using simulators removing the risk to human life [83]. Such a simulator, if visually realistic, can also be used to obtain fetoscopic videos close to in-vivo surgery settings. Furthermore, parameters influencing visibility, illumination and movement can be controlled, resolving the first challenge.
- The challenges of restricted motion (2) and irregular surface projection (3) have been resolved in other fields of research but require more keypoint pairs between two images.
- The last two challenges have not been resolved and also complicate obtaining keypoint pairs between two images. Furthermore, state-of-the-art methods are insufficient to be applicable to the in-vivo setting as will be shown in Chapter 3.

In order to resolve these last two challenges some requirements are posed and detailed in Chapter 3, but summarized here as five points of improvement:

1. Keypoint detection method should be improved such that stable and reproducible keypoints are detected in the unfavourable viewing conditions as encountered in in-vivo fetoscopic images.

2. The feature extraction method of these keypoints have to be improved, so that these can be matched even though there is only very little visual difference between them.

3. The panorama reconstruction process could be optimized. Generally a chain of images is used to reconstruct a panorama. When an image registration in the chain is inaccurate or unavailable, this has a large influence on all following image pairs. However, if this image registration can be detected or even taken out, the quality of the panorama would improve greatly.

4. The motion of the fetoscope can be improved by providing extra information to the surgeon. The distance to the placenta is changed by moving the fetoscope around. By calculating the fetoscopes position relative to the placenta, the change in distance can be predicted. Unfavourable viewing conditions can be predicted and by giving feedback to the surgeon to move the fetoscope fore- or backward, more favourable conditions can be maintained. Moreover, when the image registration performance is low, the system can ask the surgeon to move back to a previous position and obtain new and better images. Overall, the surgeons should be considered an integral part of the panorama reconstruction process.

5. The equipment also plays a role in the image registration performance. The field of view is depended on the viewing angle of the fetoscope, thus choosing a good fetoscope is crucial. The changing illumination condition can be managed using a high dynamic range camera and higher quality images in a larger range of illumination conditions can be obtained. Improving the viewing conditions will result in better keypoint matches and more accurate image registration.

A system applicable to TTTS surgery can be created following the previous recommendations. Most important is to obtain good keypoints and matches, despite the much more difficult viewing conditions. Furthermore, limiting bad viewing conditions will improve the panorama reconstruction performance. The goal of this thesis is to illustrate that deep learning methods can effectively be applied through thorough understanding of the task. This chapter has provided with a detailed overview of the task. The next section describes how deep learning is applied to this task in the next chapters.

2.5 Application of Deep Learning

With the advancements in deep learning methods, the challenges of finding and matching enough keypoint pairs between two images as described in Chapter 3 might be resolvable. Therefore, Chapter 4 redefines the subtask of extracting features as features that are matchable as well as distinctive enough from features describing different points on the placenta. This is achieved through the application of Siamese convolution neural networks combined with the contrastive loss method. This approach shows how a learning method previously applied to object classification and feature-space transformations, can also be used to learn the difference between matchable features and those features that cannot provide robust matches. In a sense a combination between *learning by doing* and *trial and error* to achieve both matchability and distinctiveness.

Chapter 3 shows that traditional methods are not effective in the in-vivo setting. Therefore, by analysing the structure represented in the image data, Chapter 5 redefines the subtask of finding keypoints by finding the points on veins instead of corners and edges. This is achieved by applying the Single Shot Detection object detection method which in a sense is posing the task differently and using *transfer learning* to learn and successfully perform this challenging task through deep learning. Furthermore, in Chapter 5 a performance metric is obtained aiding in finding the transform between two successive images, which greatly improves the applicability of in-vivo panorama reconstruction in the fetoscopic laser coagulation therapy.

3

Requirements for In-Vivo Panorama Reconstruction

Floris Gaisser, Suzanne H.P. Peeters, Boris Lenseigne, Pieter P. Jonker and Dick Oepkes,

Adopted from:
Moving from ex-vivo to in-vivo, Medical Image Understanding and Analysis: 21st Annual Conference, MIUA 2017, pp. 581–593.

Abstract

Current state-of-the-art methods focus on panorama reconstruction in an ex-vivo setting. However, these methods fail in the in-vivo surgical setting. This chapter describes the panorama reconstruction approach, the challenges posed by the in-vivo setting and the influence of these challenges on the panorama reconstruction. With experiments it is shown that the viewing quality in-vivo is greatly reduced compared to ex-vivo research settings. Furthermore, the limited motion of the fetoscope complicates the image registration as this motion requires more correct matches which are lacking. This chapter concludes by identifying the aspects necessary to shift from ex-vivo to in-vivo panorama reconstruction. Following these recommendations it should be possible to develop an approach that can be applied to TTTS surgery as will be shown in the following chapters.

3.1 Introduction

In the previous chapter Twin-to-Twin Transfusion Syndrome (TTTS) and the procedure to treat this have been introduced. It has been explained that creating a panorama of the placenta would be beneficial in successfully performing the procedure. Related research on panorama reconstruction of other anatomical structures and ex-vivo fetoscopy showed that creating a panorama can be achieved with state of the art keypoint methods [72, 99, 131]. However, the ultimate goal is applying fetocsopic panorama reconstruction to in-vivo TTTS surgery.

Unfortunately this goal has not been achieved yet. An important challenge is that generally limited data is available of the in-vivo setting. Especially, because in TTTS surgery extending the length of the procedure increases the risks significantly [83]. In many other applications the in-vivo setting is artificially recreated, assuming that this is representative of the setting used during surgery. For retina, bladder and oesophagus endoscopy this assumption is often valid.

However, in TTTS surgery this is not the case. There are certain differences between the ex- and in-vivo setting, preventing the use of state-of-the-art keypoint methods for image registration in the in-vivo setting Therefore, in this chapter these differences and the impact on the panorama reconstruction performance are described in detail as well as possible steps to tackle these challenges are provided.

This chapter is structured as following; First, the general approach to panorama reconstruction is introduced in Section 3.2. Next, Section 3.3 describes the surgical settings and its effects compared to the state-of-the-art in endoscopic panorama reconstruction. Following, the resulting differences for image processing and the challenges posed are discussed in Section 3.4. Furthermore, Section 3.5 evaluates the applicability of several image processing methods on in-vivo Fetoscopic video. Finally, Section 3.6.1 will conclude by providing the steps (research topics) towards successful in-vivo fetoscopic panorama reconstruction. The challenge of matching keypoints will be covered in the following Chapter 4. Furthermore, the challenge of detecting stable points or areas of interest will be covered in Chapter 5.

3.2 Panorama Reconstruction

Panorama Reconstruction combines multiple images into one larger image [144]. These images individually contain only a part of the panorama. But every image has at least one other partly overlapping image. A chain of images can be created, so that each successive image is a pair with overlapping areas. From this chain the whole panorama can be created. In creating a panorama, it is generally assumed that the visual information of the images is on the same surface. Thus, the panorama is a reconstruction of a plane (placenta) [33][99], cylinder (oesophagus) [21] or sphere (bladder) [9][120]. Furthermore, this means that all points in the overlapping area can be transformed with a rigid transformation to the other image. Key to panorama reconstruction is finding this rigid transformation. This process is generally referred to as image registration. There are two general approaches to this: dense [69] and interest-point based [144][17]. Dense methods use the whole image. The difference between the pixels of both images is minimized. This approach is accurate but also computational intensive. Furthermore, as there are many local minima in the optimization space, finding the correct transform cannot be guaranteed. Therefore, this method is generally used for stereo vision [69] or fine-tuning the panorama. Point-based methods find points in both images that describe the same location or area in both images. These pairs of points are then used to estimate the transformation [17]. This approach has many applications and is generally used in endoscopic panorama reconstruction [9][120][99]. These different steps are described in more detail in the next sections.

For point-based panorama reconstruction methods, only a select set of points are used. These are generally described as *keypoints*, because they describe key locations of the viewed scene [75]. As these keypoints have to be found reliably in multiple images of different conditions, they should be unique, easy to find repeatedly and accurately describe their location. The Harris corner detection method is an example of such a method [46]. Corners are considered to accurately describe their location and easy to find, however not unique. This method uses difference between neighboring pixels to detect a change along the x or y axis. If in both directions the change is large, a corner is found. Similarily edges are found with a change in only one direction. However, this method is not robust to scale and rotation changes. Therefore, methods such as the Difference of Gaussian add scale and rotation invariance to the detection of keypoints [75].

For every two overlapping images, pairs of keypoints have to be obtained from the set of keypoints created for each image. This process is called *matching*. Since there is no prior information on what keypoints are matching, each keypoint is described by its visual information. These descriptions are compared between images and the best matching pairs are chosen. A keypoint description should accurately describe the visual appearance as well as handle changes in orientation, scale, etc. Therefore local methods such as the Histogram of Gradients are generally used as in the SIFT method [75]. Other methods include SURF [8], BRIEF [19] and ORB [103]. Even though, assuming keypoints can be described perfectly, the existence of multiple visually similar keypoints is not taken into account. Therefore, not all matches can be considered correct matches (*mismatches*). To improve the ratio of correct matches, a common practice is to also review the second best match. If the difference between the best and second best match is small, then it is probable that there are multiple visually similar keypoints. Furthermore, the matches can be validated after transform estimation based on the rigid transform assumption.

The final step in image registration is estimating the transformation between the two images. In panorama reconstruction, this transformation is assumed to be a *rigid transformation*. This means that all points in the image keep their spatial relation and the surface is not deforming due to the transformation. Therefore, a point in the first image $[x_1, y_1]$ can be transformed by matrix multiplication to a point in the second image $[x_2, y_2]$. There exist two types of rigid transformations: affine and perspective. In affine transformations the camera has no out-of-plane rotations and the viewing direction is generally considered to be perpendicular to the surface. The camera can move in the image plane and viewing direction. Therefore, there is rotation (θ), translation (t_x, t_y) and scale (s) as described in (3.1). As a side note; Skew is considered to be part of affine transformation as well. However, skew changes the spatial relation and therefore not part of rigid transformations.

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} s + \cos\theta & -\sin\theta & t_x \\ \sin\theta & s + \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \tag{3.1}$$

For perspective transformations also the out-of-plane rotations are considered. This introduces four extra parameters in the transformation matrix ((3.2)), describing the rotation around the x and y axis, (p_x, p_y, s_x, s_y) and makes use of homogeneous coordinates.

$$\begin{bmatrix} x_2' \\ y_2' \\ w_2' \end{bmatrix} = \begin{bmatrix} s + \cos\theta & s_y - \sin\theta & t_x \\ s_x + \sin\theta & s + \cos\theta & t_y \\ p_x & p_y & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \tag{3.2}$$

$$x_2 = x_2'/w_2'$$
$$y_2 = y_2'/w_2'$$

The parameters of the rigid transform can be estimated from the matches. A common approach is least-squares, which minimizes the squared error. However, this method cannot handle large number of mismatches, as they have a large influence on the error. Therefore, another common method is *RANSAC* [31]. This method randomly takes samples and solves for the parameters. Then a confidence is created based on the projection error of the remaining probable pairs. After a given amount of iterations, the transformation with the highest confidence is used. However, small inaccuracies in the keypoint locations have a large influence on the accuracy of the transform. Therefore this method can handle mismatches, but has a sub-optimal accuracy. To both account for mismatches and inaccurate keypoint locations, the method *LmedS* can be used [85]. It uses the median of the error to reject mismatches. Moreover, it can find an optimal solution by minimizing the error. However, with more than 50% of mismatches, an optimal solution can not be guaranteed, due to local minima in the optimization space.

A panorama is reconstructed from a chain of image pairs, based on the assumption that all images view the same surface. However, with perspective transformations it does not mean that the image planes are parallel with the viewed surface. Therefore, a suitable common projection surface has to be chosen, similarly to the physical surface. However, these image registrations describe the relation between images and not to the projection surface. Therefore, a reference image is generally chosen and its image plane as the base for the projection surface. Every following image is transformed to the reference image using the image registrations. However, image registrations generally have small inaccuracies due to mismatches and localization errors of the keypoints. As images are transformed down the chain to the reference image, an error is built up in the overall projection. This error can become large, which results in visual inconsistencies in the panorama.

3.3 Differences in Setting

The differences between an ex-vivo research setting and our in-vivo simulator (Figure 3.1a) will be described in this section. Each paragraph describes one specific aspect and examples of the visibility conditions are shown in Figure 3.3.

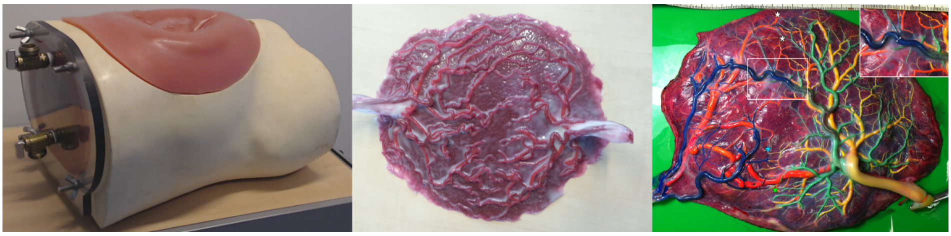


Figure 3.1: a) Our simulator, b) placenta, and c) dye-injected placenta

Ex-vivo research uses a placenta after birth, which is larger compared to an earlier stage of the pregnancy. Furthermore, the blood vessels of the placenta are dye injected for extra contrast with the underlying tissue. Figure 3.1 compares the ex-vivo dye-injected with our simulator placenta. To record data, the placenta is placed on a flat working space, making the placenta also flat. Therefore, in contrast to in-vivo settings the observed surface is a plane. Furthermore, compared to in-vivo settings, the placenta does not move and gaining a good view is possible from every angle.

With ex-vivo research, the impact of the amniotic fluid is overlooked. This fluid is generally far from clear, as the fetus normally pees in the amniotic fluid, giving it a yellow-brownish color. However, if the fetus is in distress, which is often the case with TTTS, the fetus might release some bowel movement, giving the amniotic fluid a green turbid color, reducing the range of visibility.

In the ex-vivo setting, the illumination condition can be completely controlled. There can be ambient lights as well as high intensities of light. With this controllable setting, near optimal illumination conditions can be created at all times. However, with in-vivo settings, there is no ambient light. The only light source is from the endoscope itself creating an uneven distribution of light. There is too much light in the center of the image and it gets too dark towards the edges. Moreover, too strong light might blind the fetus, so that only less ideal illumination conditions can be created.

The motion of the fetoscope is the last difference between an ex- and in-vivo setting, and often not considered. In the work of Reeff et al. [99] and Tella et. al. [131] the motion of the endoscope is described as an outwards spiralling motion. This mostly in-plane translation motion is shown in Figure 3.2a. However, surgeons cannot make an outwards spiralling motion. Instead, they start at the umbilical cord of the recipient and follow veins one by one to the end and back. Furthermore, the motion of the fetoscope is restricted at the point of entry in the in-vivo setting. This restriction only allows a combination of in-plane translation and out-of-plane rotations or forward motion as shown in Figure 3.2c.

The combination of bad visibility and illumination conditions force the surgeon to move the endoscope closer to the placenta. However, this reduces the imaged surface significantly. The sample views of related work contain multiple blood vessels [99][72]. Whereas for in-vivo images, one or two blood vessels, sometimes a crossing or split can be seen.

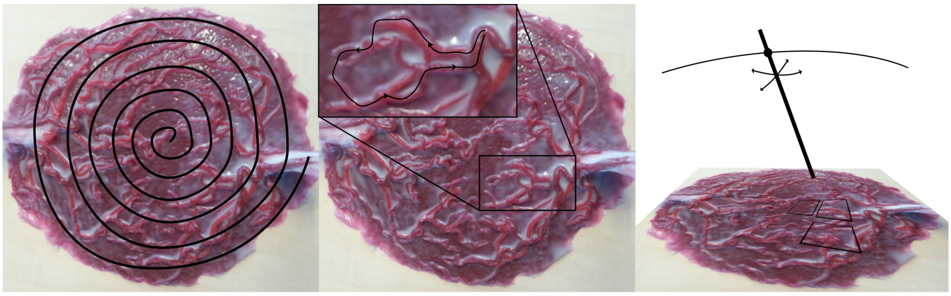


Figure 3.2: Fetoscope movements a) ex-vivo: spiral motion b) in-vivo: followed path c) in-vivo: rotational motion around entry point resulting in perspective transformations

3.4 Influence on Panorama Reconstruction

The image registration performance is influenced by the previously described differences in the setting. This section describes this effect for each of the steps of panorama reconstruction as described in Section 3.2.

Keypoint methods internally create an intensity map of the difference between pixels in *x* and *y* direction. Dye injected placentas have better constrast, while illumination and the visibility condition reduce the amount of observable contrast. Reducing the field of view limits the amount of visible structure even more. Therefore the more complicated viewing conditions reduce the number of detected keypoints.

Generally it is assumed that the strongest points in the intensity map represent corners. However, this assumption is not valid when the image contains only a few or no corners. Because of image noise a point along an edge is seen as stronger than other points along the edge and thus selected as a keypoint. Since noise is random, this keypoint is not reproducible and as shown in Figure 3.3 occurs quite strongly when there is not enough light. Moreover, keypoints have similar appearance along an edge, thus the location is not unique. These aspects make the selected keypoints unreliable, as they are not unique and not reproducible.

Matches are obtained from the visual appearance around each keypoint. However, the underlying tissue of the placenta does not contain a lot of structure, creating weak descriptions. Moreover, most keypoints are on an edge of a blood vessel. Since blood vessels have very similar appearance, their description is also very similar. The weak and similar description of keypoints complicate the matching process, resulting in less reliable matches.

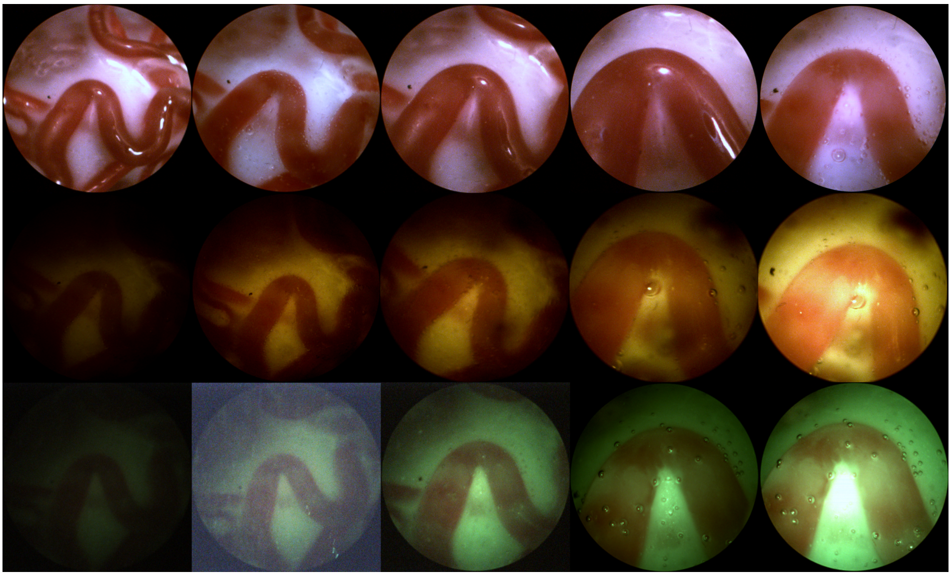


Figure 3.3: Variations in viewing conditions. Top row left to right: ex-vivo - far, ex-vivo with water - far, ex-vivo - nominal, ex-vivo - close, ex-vivo with water - close
Middle row: yellow liquid; bottom row: green turbid liquid; left to right: far - dark, far - nominal, nominal for both, close - nominal, close - bright

The motion of the fetoscope has a direct relation to the transformation between two successive images. The motion of the fetoscope is not restricted in ex-vivo experiments. Therefore, the fetoscope is generally moved sideways, creating translation transformations. However, because of rotation around the entry point, perspective transformations as shown in Figure 3.2c are created. In previous work we showed that estimating translational motion results in a two times lower pixel error compared to other types of motion [33]. Furthermore, rotation around the entry point changes the distance to the placenta. Resulting in a change of light and unfavourable illumination conditions is the result as shown in the two most outer columns of Figure 3.3.

With an outward spiralling motion of the fetoscope, a continuous set of loops can be created between two images between the inner and outer spiral allowing for bundle adjustment or loop-closure. The difference between an incorrect and a feasible panorama reconstruction lies with the refinement step as can be seen in figure 2 of the work of Reeff et. al. [99]. However, this spiralling motion is not possible with in-vivo fetoscopy and combined with all previous mentioned aspects, panorama reconstruction is much more complicated compared to ex-vivo settings.

3.5 Experiments & Results

To investigate the influence of the previously described viewing conditions on image registration experiments are devised. Images are captured for every condition using our placenta (Figure 3.1b) and for in-vivo settings with our simulator (Figure 3.1a). A 5mm solid core endoscope with a medical xeon light source and a GigVision BlackFly camera have been used. The effective area of the endoscope is 1030 × 1030 pixels and after undistortion the inner square has an area of 850 × 850 pixels.

The visibility condition caused by the color and turbidity of the amniotic fluid is varied in three settings; One simulating the ex-vivo setting and 2 in-vivo settings simulating the two different types of amniotic fluid that can be encountered; normal *yellow* fluid and the distressed case of *green turbid* fluid. These settings are created by adding dye to the water and shown in Figure 3.3 middle and bottom row.

The illumination condition is varied in ex-vivo and in-vivo settings. The *ex-vivo* setting is created by placing the placenta on a table with enough ambient light. For the in-vivo settings, the placenta is placed inside the simulator and the illumination is manually adjusted to create a *dark*, *nominal* and *bright* setting.

The field of view is varied by changing the distance to the placenta. Three settings are created: *close*, *nominal* and *far*, respectively at ±1, 2 and 3 cm from the placenta. The far setting is chosen such that the camera is at it limitations for the green turbid fluid and the close setting always gives a clear view.

The different visibility and illumination conditions combined with different viewing distances can be seen in Figure 3.3. A set of sequential images is recorded for all possible variation combinations along the path described in Figure 3.2b. Every following image is taken such that the movement is about $\frac{1}{3}$ of the visible range, resulting in more images for decreasing quality and viewing area.

Experiment 1: Number of Keypoints

With decreasing quality of the viewing condition also the number of keypoints is expected to decrease. In this experiment three different keypoint methods are evaluated; SIFT, SURF and ORB. Figure 3.4 (at end of chapter) shows the number of obtained keypoints for changing illumination, visibility condition and field of view. As expected the results show that with decreasing distance the number of keypoints decrease. For the illumination condition more light gives better contrast and thus more keypoints. However, for the far distance, increase in contrast is lost due to the increased distance. Furthermore, for the green turbid liquid, there is not enough light, resulting in much noise and many keypoints are obtained on the noise itself.

Experiment 2: Reproducibility of Keypoints

Lack of structure in the form of corners reduces the number of reproducible keypoints. By manually establishing the transformation between two successive images, the ratio of reproducible keypoints can be obtained. Figure 3.5 (at end of chapter) shows for the different keypoint methods the number of retained keypoints for varying viewing conditions. The number of reproducible keypoints is about $15 - 25\%$ of the number of detected keypoints. However, looking at the results of individual images pairs the variation is large, sometimes many and sometimes no reproducible keypoints at all are detected.

Experiment 3: Matchability of Keypoints

The matching performance is limited due to similar appearance of the blood vessel and weak description of the keypoints. The previously obtained keypoints and manualy established transformations are used to evaluate the matching performance. For the ex-vivo situation the ratio of correctly matched keypoints is about 10%. However, for the in-vivo situation three groups can be created; First, good visibility condition, with close to ex-vivo matching. Second, low illumination condition, resulting in too much noise and many unreliable keypoints. Third, low structure situation, with too few keypoints to do image registration. The latter two groups have close to zero correctly matchable keypoints and no image registration can be obtained.

3.6 Discussion

With the previous experiments we have shown, that the visibility conditions encountered in in-vivo TTTS fetoscopic surgery complicate the image registration process. The number and the reproducibility of keypoints is reduced to the point that no valid matches can be found. This reduction in performance can be explained by three key aspects; detecting reliable keypoints, matching of keypoints and motion of the fetoscope.

Corners are the source of reliable keypoints, even though they can be found on a placenta, the number of corners in images from ex-vivo settings is limited. Changing to an in-vivo setting reduces the contrast and the field of view. Less corners and with less contrast appear in the image. Furthermore, with reduced illumination the image contains more noise. Since keypoint detection methods adjust themselves to the structure present in the image. Points along the edge of blood vessel are obtained as keypoints, though they cannot be considered reliable keypoints.

Keypoints are matched based on their visual appearance. For corners, where two blood vessels cross or split, there is clear and unique visual information. However, for a keypoint on the edge of a blood vessel, the visual appearance is very similar to any point along the edge of a blood vessel. This is even true for a curved vein, but the feature is rotated. This results in many mismatches and only a few correct matches.

The fetoscope can only change the view in lateral position by rotating around the point of entry. This motion also changes the distance to the placenta, thus creating unfavourable viewing conditions. Both the distance, but also the illumination condition that has to be manually adjusted, create visibility conditions that are unfavourable for image registration. Moreover, this type of motion also complicates the image registration as more parameters have to be estimated, thus requiring more and better matches.

3.6.1 Conclusion

The goal of this chapter is to identify and rise awareness to the challenges encountered with in-vivo fetoscopic panorama reconstruction. In the previous section we have discussed three key aspects that complicate image registration. In Chapter 4 better keypoint matching is achieved with deep-learning by using contrastive loss.

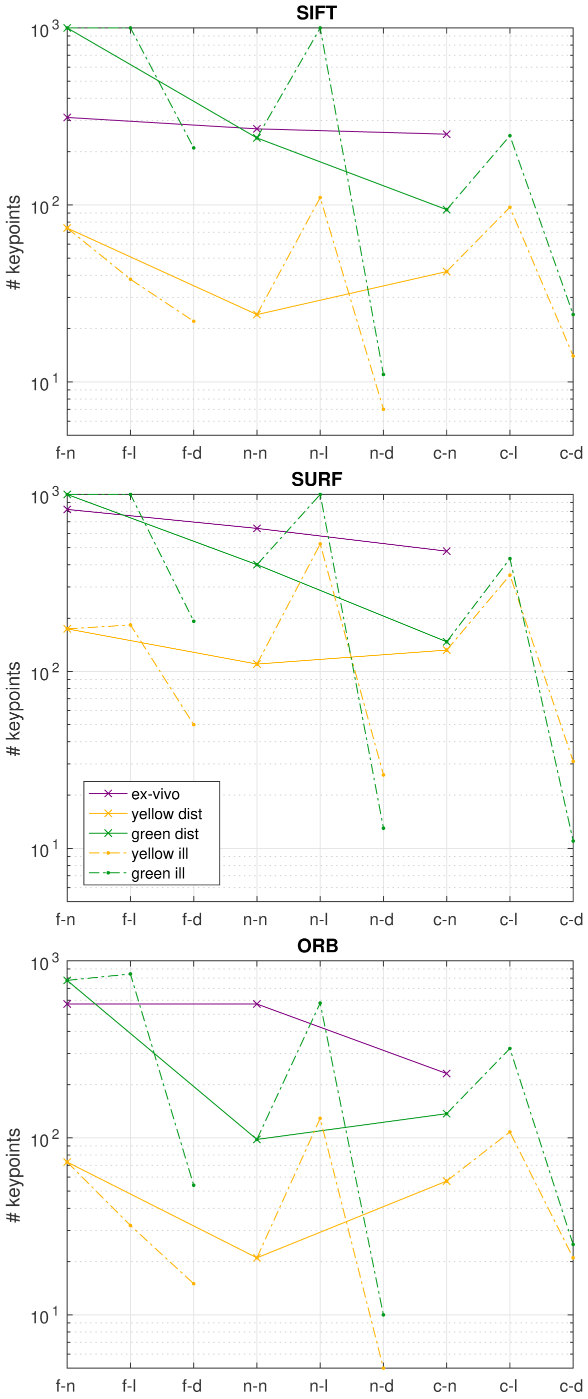


Figure 3.4: Comparison of detected keypoints; x-axis labels are: far (f), nominal (n) and close (c) distance and nominal (n), light (l) and dark (d) illumination. Example: f-l is far and light

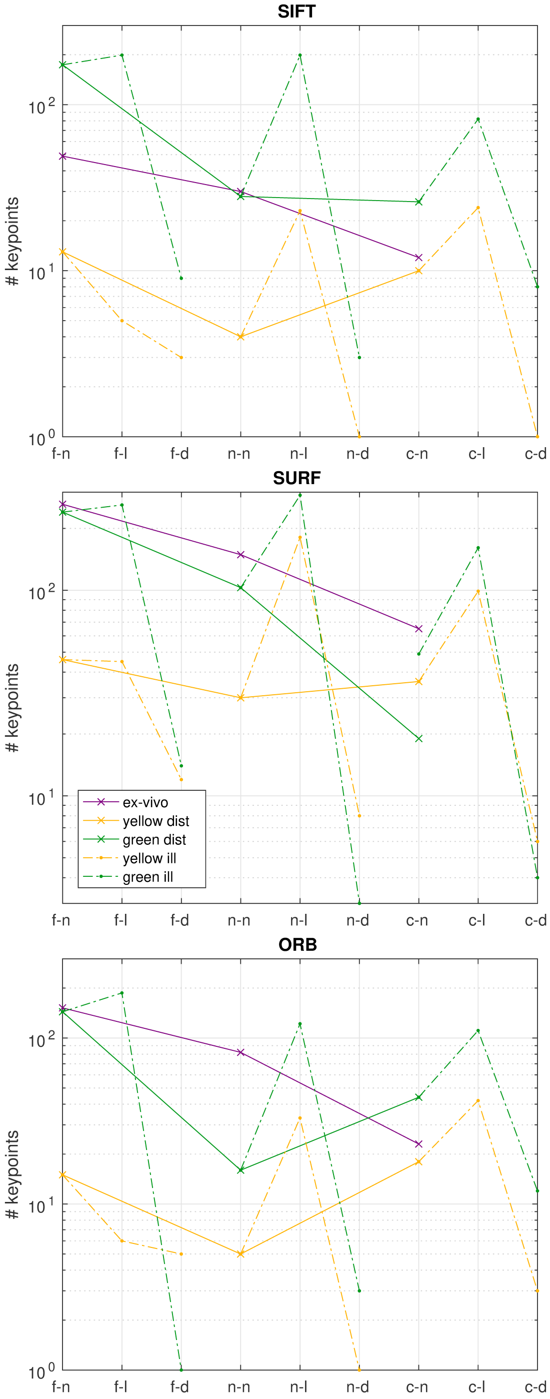


Figure 3.5: Comparison of reproducible keypoints; x-axis labels are: far (f), nominal (n) and close (c) distance and nominal (n), light (l) and dark (d) illumination. Example: f-l is far and light

4

Matchable Feature Extraction for Image Registration

Floris Gaisser, Pieter P. Jonker, Toshio Chiba,

Adopted from:
Image Registration for Placenta Reconstruction, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 7th Workshop on Biomedical Image Registration. 2016 pp. 33–40

Note: This publication was published before the publication of the previous chapter.

Abstract

As described in the last two chapters, image registration for in-vivo placenta reconstruction requires extraction of matchable features between image pairs. This chapter introduces a feature extraction method that can extract more robustly matchable features. This feature extraction method consists of a Convolution Neural Network (CNN) that describes key areas in the image such that it can be matched to similar areas in the image pair. In order to extract robust matchable features a novel approached is proposed using similarity learning in training the CNN. Compared to feature extraction methods used in literature for ex-vivo panorama reconstruction, up to three times more keypoints could be matched in the various image transformations between the image pair. This novel method allows to solve one of the main challenges posed for in-vivo placenta reconstruction.

4.1 Introduction

In the previous chapter it was pointed out that one of the areas to improve the panorama reconstruction process would be to improve the keypoint matching performance such that the image registration is more accurate and robust to mismatches. Standard keypoint methods are designed to be generic enough to describe all type of structures. However, in fetoscopic image registration this is not required. Therefore, extracting features that can better describe the visual appearance of the placenta would be beneficial. Furthermore, since the visual appearance of the veins on the placenta is overall very similar, it is difficult to extract features that accurately describe the small differences between them, which makes it difficult to find correct matches.

Therefore, this chapter will focus on extracting features that allow better matching and is specific to the fetoscopic images. Deep learning allows to learn feature extraction that is specific to the provided data. Furthermore, with the introduction of *Contrastive Loss* the feature extraction can be trained to find better matchable features.

This chapter first describes the image registration process, its challenges and how convolutional neural networks could extract learned features in Section 4.2. Next, the approach to extract matchable features with contrastive loss is described in Section 4.3. Through experiments in Section 4.4, it will be shown that with the proposed method much more correct matches can be obtained. Furthermore, the consistency of matching is greatly improved such that a transform can be obtained. This results in a robust and more accurate image registration.

4.2 Image Registration

Reconstructing large view panoramas of the internal anatomical structures has been a large field of research and found many applications, such as retina [113], bladder [120] and oesophagus [21] reconstructions, as well as in ex-uterin endoscopic mosaicking [72, 99].

First, [99] shows the reconstruction of a small part of an ex-vivo placenta, though the results show that the image registration has a low accuracy and the reconstruction without post-processing contains many artefacts. Furthermore, the images are captured by moving the camera sideways in a structured circular pattern. First of all this cannot be reproduced in an in-vivo setting, but also the transforms between images now only consists of translations.

Second, in [72] they project endoscope images of a color injected placenta on a 3D ultrasound model, which shows accurate results in image registration. However, such a setting with an ex-vivo color injected placenta is not compatible with our goal of in-vivo surgery.

Third, there have been promising results in other applications such as bladder reconstruction [120]. There an ex-vivo dye injected bladder is reconstructed from a flexible endoscope with image registration, bundle adjustment and spherical projection. However, also here this method is not suited for our setting, as no prior structure is available. Furthermore, the encountered transformations here and in [113] are also mostly translations which can be robustly estimated with existing methods.

Last, in oesophagus reconstruction [21] an accurate reconstruction is presented, however here pipe projection is used, which is also not applicable to our setting. Furthermore, spatial consistency is not required for this type of reconstruction.

Although all above methods are not directly applicable in our aimed setting, some successes have been shown.

4.2.1 Image Registration

The previously discussed applications all use image registration methods which try to find the transformation between two images [47]. They try to find corresponding pairs of interesting points in both images by feature matching, whereafter a transform is estimated based on the found matches [144].

To find matching pairs, first interesting keypoints are chosen using methods such as the maximum Difference of Gaussians [75] as used in SIFT. Next, to find the corresponding point in the other image, the selected keypoints are described using a feature extraction method, such that the features are similar regardless of the appearance changes due to the transforms between the images. Obtaining such features has been the source of many invariant methods such as the Scale-Invariant Feature Transform (SIFT) [75] or Binary Robust Independent Elementary Feature (BRIEF) [19].

Though feature extraction methods are designed to be invariant to transformations, there are still challenges in obtaining appropriate matches. To handle incorrect matches, transform estimation methods try to find a best fitting estimation by iteratively fitting on random subsets of the matches and selecting the best fitting subset. RANSAC [31] is robust to mismatches but finds a sub-optimal estimation, where LMedS [85] finds a more accurate estimation but requires at least 50% correct matches.

4.2.2 Problem statement

Our initial as well as other research [99, 120] showed that the state-of-the-art methods have promising results but lack application in a realistic setting, i.e. it cannot be applied in real surgery. Hence, our research focusses on using fetoscopic videos from a more realistic setting which introduces challenges not encountered before.

First of all, there is the loss of contrast of the blood vessels due to the inability to use dye injected placentas. Then, most of the time complex perspective transforms are encountered as the endoscope has a fixed point entering the uterus and the view is mostly changed by rotating about this entry point. Finally, since reconstruction of the placenta has to be done near real-time, long post-processing is not possible and therefore the transform estimation has to be fairly accurate and also consistent.

Our initial research showed that on our fetoscopic images, state-of-the-art keypoint methods fail to extract robust keypoints and features, partly because these methods are designed for natural images and require unique and distinctive structures. But in our case blood vessels on the placenta are very similar and have a very limited structure.

4.2.3 Convolutional Neural Networks

In the fields of Machine Learning and Computer Vision, deep-learning neural networks have found a wide range of applications due to their ability to learn specific concise representations from the raw image data [62, 125]. They outperform many state-of-the-art methods as well as the previously described keypoint description methods. Furthermore, inspired by the neural sciences on how humans learn, a Convolutional Neural Network (CNN) can be trained to extract invariant features by using similarity learning [45, 141]. Consequently, these characteristics motivate us to use CNNs to cope with the challenges encountered in fetoscopic image registration.

4.3 Matchable Feature Extraction

In contrast to keypoint feature extraction methods, convolutional neural networks have to be trained to learn a mapping between the input image data and a feature vector. Our proposed method uses a two staged approach; first a network is trained to extract features that are robust to small perspective transforms. Second, training an extension of this first network is performed to fine-tune the feature extraction, in order to obtain features that can be matched robustly.

To train any neural network, a loss function is used to acquire the feedback for updating the internal state of the network. Our method is described in detail in Sec-

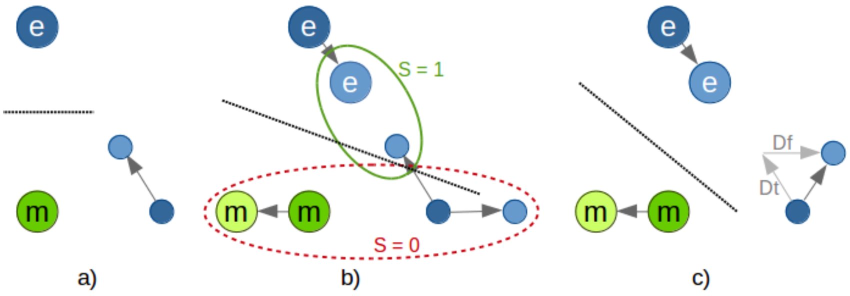


Figure 4.1: a) Label based learning b) contrastive loss c) matching learning; (e) elephant features, (m) mouse features

tion 4.3.1. The network for image registration is described in Section 4.3.2 detailing the feature extraction and the matching and registration parts of the network. As the network is trained using a training set, the creation of the training set is described in Section 4.3.3. Finally, the remaining sections describe the experimental setup (Section 4.3.4), the results (Section 4.4) and a discussion of the results (Section 4.5).

4.3.1 Learning Method

CNNs learn a mapping between the input and required output by updating internal weights based on feedback given to the network. This feedback, also defined as the error or the loss, is obtained by defining a function which generally takes the current and the desired output of the network as inputs. This function tries to minimize the error between output of the network and desired output, thus using only feedback on similarity.

A different approach is to also define feedback on dissimilar inputs. This is achieved with the contrastive loss function [45]. Which defines feedback to decrease the difference between similar pairs and to increase the difference between dissimilar pairs, which results in a more easily separable and more evenly distributed feature space.

To describe the difference in feedback, consider a network trained to classify images containing either a mouse or an elephant. Suppose during training a sample of an elephant is incorrectly described as a mouse. Normally feedback is provided to decrease the difference between the class label from the network and the label from the training sample. This results in making the output more similar to the elephant label, as shown in Figure 4.1a where the feature after learning is still closer to the incorrect mouse label.

For contrastive loss training, a siamese network [15] using two images is used to train the network. Generally, this method is utilized to train a network for feature extraction making the output a feature vector. In the case where a sample of an elephant and a mouse is used, the difference between their outputs is increased up to a defined margin, as shown in Figure 4.1b with the red dashed ellipse. However, in the case two samples of the same label are used, the difference between the two outputs is minimized as shown with the green solid ellipse. Hence improving the feature extraction towards their correct label, as well as making the two features more dissimilar and more easily separable.

Our goal is to train a CNN to extract invariant and robust features to describe key areas. To realise invariance to perspective transforms, the error between different transformations of the same patch has to be minimized, while to extract features that are separable, the error between different patches has to be maximized. This can be achieved with the contrastive loss function as is defined in ((4.1)). Where X_i is the output of the network as feature vectors, m the margin, generally defined as 1, s the similarity of the pair with 1 as similar and 0 as dissimilar. For more details we refer to the original work on contrastive loss [45].

$$L = s\frac{1}{2}(D_w)^2 + (1-s)\frac{1}{2}(max(0, m - D_w))^2 \tag{4.1}$$

$$D_w = \|X_1 - X_2\|_2$$

In the process of image registration, extracted features are matched on their Euclidean norm similarity. To train a network to extract features that can be matched robustly, the contrastive loss function is extended. The ground truth from the training samples is used to select true matches. Next, feedback is defined such that the error between incorrectly matched features is increased and between correctly or supposedly matched features is decreased. This is described by ((4.2)), where $f = 1$ when the feature matching obtained a false match and $f = 0$ when the feature matching was correct. D_f and D_t are respectively the differences between X_1 and the feature vector obtained by feature matching X_f or X_t obtained by the true transform.

$$L = \frac{1}{2}((1-f)D_f + fD_t)^2 + f\frac{1}{2}(max(0, m - D_f))^2 \tag{4.2}$$

$$D_f = \|X_1 - X_f\|_2$$
$$D_t = \|X_1 - X_t\|_2$$

Function 4.2 is inspired by the contrastive loss function, in minimizing the difference between correct matches and increasing the difference between incorrect matches. But it differs by introducing two reference features to match with; the true match X_t and the feature based match X_f. In the case where the feature matching was correct ($f = 0$), these two references are the same, and (Equation 4.2) can be considered similar to the case where $s = 1$ in (Equation 4.1) as the second term is cancelled out. However, in the case where the feature matching obtained an incorrect match ($f = 1$), additional feedback is given based on the incorrect match. This has as effect that not only the correct features are made more similar and the incorrectly matched features more dissimilar, but also that the specific aspects that form the difference between the correct feature and the incorrect feature are improved.

To describe this effect, consider the previous example of training a network describing images of a mouse and an elephant. Imagine the feature vector describing some aspects of the animals including colour and size. Suppose during training an image of an elephant was mistakenly matched with a feature of a mouse. The feedback will increase the difference between these two features, in both colour and size. Furthermore, feedback is given to reduce the difference between the correct feature and the extracted feature. As the size of an elephant is large, the aspect of size is increased even more. But as both animals are grey, the aspect of colour is reduced. Even so, the importance of the colour aspect is reduced over time up to the point that the network will not use colour any more to describe the animals. This is shown in Figure 4.1c with the combination of the difference between the incorrect feature and extracted feature D_f as well as the difference between the correct feature and the correct feature D_t. Resulting in a much better separable feature space as indicated with the black dotted line.

It can be argued that the triplet learning from [141] is very similar to our proposed method. However, there is one key difference in the way how a dissimilar pair is chosen. In [141] this is a fixed pair chosen at the moment the training set is created, whereas our method dynamically obtains a dissimilar pair based on the output of the network. Therefore it is adaptive to what is learned in the network, creating a much better separable feature space. Furthermore no dissimilar pairs have to be selected when creating a training set, reducing the training set size as well as training time significantly.

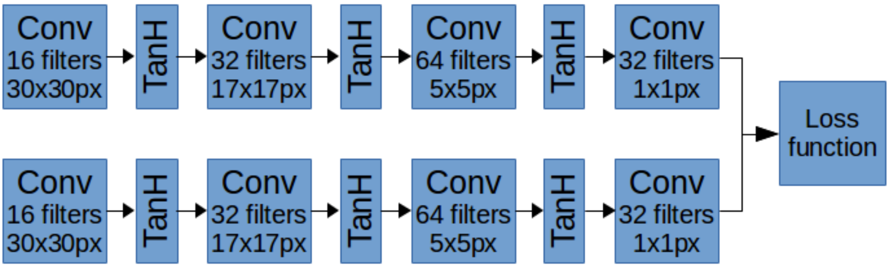


Figure 4.2: CNN architecture.

4.3.2 Network Architecture

As stated before, the network is trained in two stages; feature extraction training and robust matching training. Both stages use a siamese network architecture, where two parallel networks with the same architecture share their internal weights to process two simultaneous inputs [15].

For feature extraction, a network is designed such, that an input image patch of 50×50 pixels is reduced to a feature vector of size 32, by choosing the right number and filter sizes for the convolution layers as shown in Figure 4.2.

For training robust matchable features, the same network is used, but instead of a single image patch, 961 patches of 50×50 pixels are extracted in a 31×31 grid from a 500×500 image. Furthermore the contrastive loss layer is replaced with the matching loss layer as described in the previous section.

For evaluation with image registration, the matching loss layer is replaced with a matching and rigid transform estimation layer. This layer outputs the estimated rigid transform found by RANSAC or LMEDS [31, 85] and the mean projection pixel error between the true transform and the estimated transform.

```
Algorithm 1 Training data
Step 1: Create image patches.
Step 2: Discard similar patches
Step 3: Select only interesting patches
Step 4: Create transformed patches
Step 5: Similarity pairing
```

4.3.3 Training data

To train any CNN, a dataset has to be created that is as small as possible to reduce the training time. As well as a complete and an evenly distributed representation of the variations to be encountered, in order to achieve robustness and avoid overfitting. In Algorithm 1 the steps for creating these training sets are shown and detailed below.

First, a subset of images from the fetoscopic videos are selected to decrease the amount of training data. As the motion within one second is expected to be small, only 5 images each second are selected. Next, for the first training stage, patches of 50 × 50 pixels (Figure 4.3 right bottom) are extracted and for the second stage patches of 500 × 500 pixels (right top) are extracted at an interval of 50 pixels from the valid area of 550 × 550 pixels of the source images.

Steps 2 and 3 are to improve the quality of the extracted patches used in the dataset. First, the absolute pixel difference between all patches is obtained. Patches that are too similar are discarded, such that reoccurring variations are not presented multiple times. As a result, the dataset contains an evenly distributed representation of the variations. To further improve the information density of the dataset, all patches with below average gradient energy are discarded. This results in a set of patches that are above average descriptive and makes sure that non-descriptive patches are excluded.

In order to have invariance to the expected transformations, every patch is rigidly transformed. For the training sets, fixed step sizes are chosen for every component of the perspective transform, related to the observed transforms occurring between two successive frames. Similarly, for the evaluation sets, random transforms are chosen.

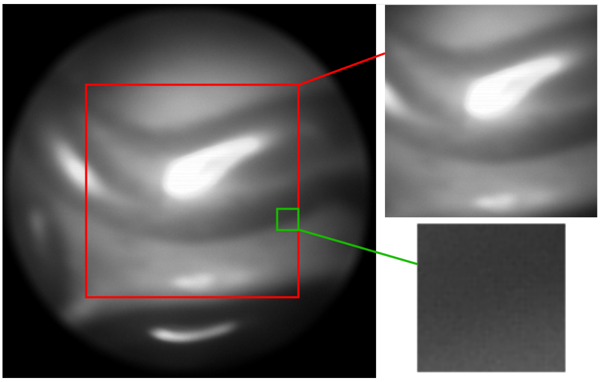


Figure 4.3: Image from fetoscope and crops for learning

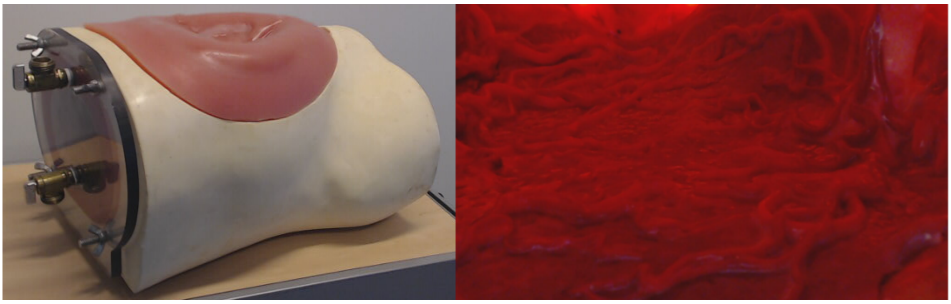


Figure 4.4: Left: Simulator, Right: inside of simulator with placenta

For similarity training, pairs are created in the final step where every patch is paired with their variations. Furthermore for the first training stage also dissimilar pairs have to be selected. Therefore every patch is paired with 25 of their most similar patches based on the absolute pixel difference obtained in step 2.

4.3.4 Experimental Setup

To evaluate the introduced image registration method, fetoscopic videos were utilized from a TTTS surgery simulator used to train surgeons as shown in Figure 4.4 [84].

It has to be noted that the artificial model of the placenta as shown in Figure 4.5, is a close as possible representation of a real placenta. This is unlike the much easier dye injected placentas that are used in the current state-of-the-art. Furthermore, the positioning of the placenta and use of the fetoscope is similar to that of in-vivo surgery (Figure 4.4).

The image registration method has been implemented on a Dell precision M4700

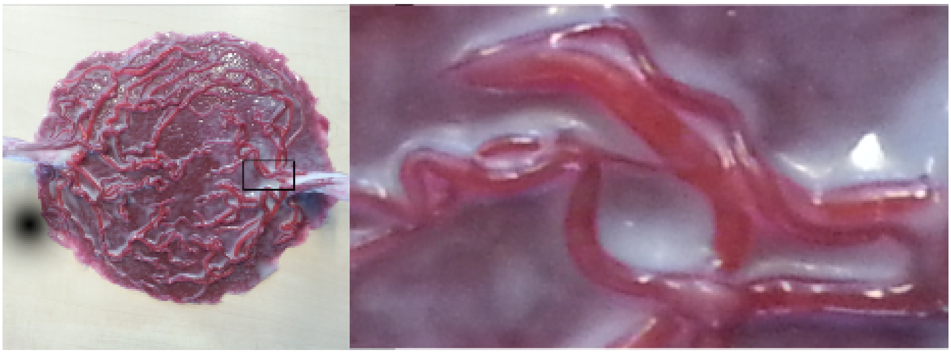


Figure 4.5: Artificial placenta

with the Caffe [54] and OpenCV libraries. The videos have been acquired with a medical camera capturing a circular image of 880 × 880 pixels representing an area of about 8 × 8 mm as shown in Figure 4.3.

4.4 Experiments & Results

For performance evaluation of image registration in a realistic setting, a video taken from the simulator operated by an expert is processed. Three sections of the video have been chosen with similar length of about 75 seconds, representing different areas of the placenta. Training is performed on one of the videos and compared with the other two. Patches are extracted for both the first and second stage of training and evaluation as described in Section 4.3.3. By changing the training set, three combinations of training and evaluation could be obtained.

4.4.1 Experiment 1

First the invariance of the novel feature extraction method is evaluated in respect to the different transformations and compared to the state-of-the-art keypoint descriptors. In Table 4.1, the average performance is shown together with the standard deviation of the correctly matched points out of the total keypoints. CNN1 represents the performance trained only with the first stage, while CNN2 was trained with the novel matching learning method.

Table 4.1: Correctly matched points

Method	SIFT	BRIEF	CNN1	CNN2
Translation	28.2%	29.5%	67.5%	**81.4%**
	±24.0%	±10.1%	±15.6%	**±13.6%**
Rotation	22.1%	31.5%	53.4%	**74.5%**
	±19.0%	±7.8%	±13.1%	**±12.6%**
Scale	21.1%	36.1%	57.8%	**72.9%**
	±16.9%	±7.9%	±11.6%	**±12.2%**
Perspective	13.7%	27.4%	51.4%	**68.4%**
	±9.8%	±7.2%	±7.1%	**±12.2%**
All	13.8%	26.2%	50.9%	**62.8%**
	±4.7%	±6.7%	±3.9%	**±6.1%**

All methods use a fixed grid of 31 × 31 points with a spacing of 10 pixels, therefore always having 961 keypoints for feature extraction. This was also chosen for SIFT and BRIEF to guarantee that keypoints were available that represented the same area in both images. For both SIFT and BRIEF, a match was only accepted if the distance ration to the second best match was below a threshold as shown in [17]. This threshold was adjusted such that only the best matches, but also enough matches could be retained for the next experiment.

4.4.2 Experiment 2

For performance evaluation of image registration in a realistic setting, comparable to in-vivo surgery, Table 4.2 shows the image registration error as the mean pixel error of the estimated transform together with the standard deviation.

For state-of-the-art keypoint description methods, RANSAC is used for transform estimation, whereas for the proposed methods also LMedS is used, as more than 50% of the matches are correct matches.

It should be noted that even by adjusting the threshold, for both SIFT and BRIEF, in 10-25% of the images the matching ratio was so low that less than the required 4 matches were found. Furthermore, for about 15-25%, no reasonable transform estimation could be found. These have all been excluded from this comparison, as they influenced the average pixel error drastically.

Table 4.2: Mean pixel error of estimated transform. $^{1)}$ RANSAC $^{2)}$ LMeDS

Method	SIFT	BRIEF	CNN1 $^{1)}$	CNN1 $^{2)}$	CNN2 $^{1)}$	CNN2 $^{2)}$
Translation	4.0 px	3.5 px	3.2 px	2.6 px	2.6 px	**2.4 px**
Translation	±1.7 px	±1.7 px	±3.4 px	±1.8 px	±1.5 px	**±1.4 px**
Rotation	7.1 px	8.0 px	6.6 px	4.0 px	4.3 px	**2.4 px**
Rotation	±1.9 px	±2.0 px	±5.0 px	±2.6 px	±2.8 px	**±1.7 px**
Scale	7.1 px	8.6 px	5.3 px	3.6 px	4.6 px	**2.6 px**
Scale	±1.8 px	±1.4 px	±3.7 px	±2.0 px	±3.1 px	**±1.6 px**
Perspective	9.9 px	9.8 px	6.6 px	4.2 px	5.7 px	**2.9 px**
Perspective	±3.1 px	±2.8 px	±3.9 px	±2.6 px	±3.2 px	**±1.6 px**
All	8.3 px	8.5 px	7.5 px	5.2 px	6.6 px	**3.0 px**
All	±3.0 px	±2.7 px	±4.0 px	±2.9 px	±3.1 px	**±1.6 px**

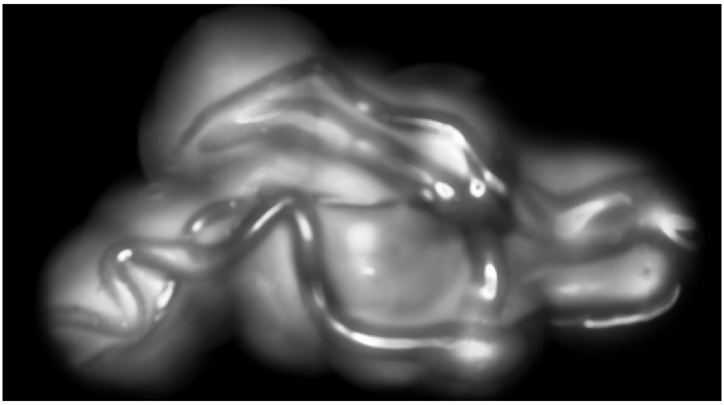


Figure 4.6: Reconstruction of placenta

4.4.3 Experiment 3

Using 26 sequential registered images from the previous experiment, a partial reconstruction of the placenta, as shown in Figure 4.6, has been made of the same area shown in Figure 4.5. In this reconstruction, no post-processing or blending methods were used, but still giving promising results.

4.5 Discussion

In this chapter an image registration method is introduced to handle the challenges posed by fetoscopic videos. The main challenge in image registration is to obtain invariant features that can be matched robustly. With the experiments it was shown that feature extraction with a CNN trained in a novel way, allows for more robust features and improves image registration of fetoscopic images.

The first experiment shows that depending on the applied rigid transformation, for the novel approach of using learned feature extraction, up to 67.5% of the features can be matched. The key behind this, is that the network learns to extract the essential components to describe an area, such that it is still invariant to the applied transforms.

The remarkable low matching performance of state-of-the-art methods can be explained by the ratio between the robustness to variations and the difference between different keypoints. For both SIFT and BRIEF, as they are designed to be invariant to these type of rigid transformations, the difference between extracted features of similar keypoints is small. Thus, for robust keypoint matching, it requires a very different type of keypoint, which is also the reason why it is advised to only accept matches by a distance-to-second-best ratio. However, as having different type of keypoints is not feasible with fetoscopic images, since blood vessels look very similar, the result is a low matching performance.

This also explains the two causes why many of the matching samples for SIFT and BRIEF had to be excluded from the results. This had two causes. First, the distance-to-the-second-best-ratio threshold rejects the majority of matches, resulting in less than 4 matches. Second, the features are too similar and are matched incorrectly.

In contrast to state-of-the-art keypoint descriptors, our novel matching learning method increases the difference between different areas on top of the invariant feature extraction. This is shown in the improvement in matching performance between CNN1 50.9% and CNN2 62.8% for all transforms.

In [99] they showed a matching performance of 68% for SIFT matching. This is quite different from the results presented in this chapter. But, this difference can be explained by three aspects. First, the field of view of their endoscope is larger, showing much more structure. Second, they use a dye injected placenta, which results in much more contrast allowing for better features to be extracted. Third, the motion they used during recording consists mostly of translation. SIFT obtains a 2 times better matching performance in experiment 1 for translation (28.2%) compared to the realistic transforms encountered during surgery which it only matches 13.8%.

The results of experiment 2 show that having more correct matches makes for more robust and precise transform estimation. This is reasonable because of the well known correlation between the amount of matches and the transformation error. It should also be noted that LMedS will give an optimal estimation, where RANSAC will give the best estimation of its iterations. This can be seen from the results of CNN1 with LMedS and CNN2 with RANSAC for all transforms. The latter has more correct matches, 62.8% compared to 50.9%, but also a higher estimation error of 6.6 compared to 5.2 pixels. Furthermore, looking at the individual matching results, it can be seen that RANSAC will sometimes give an estimation that is quite far off.

Another aspect that is often not considered is the consistency of the image registration process. With conventional keypoint matching methods some of the images could not be registered. The same problem has been reported in [99]. With our proposed method, 100% of the test images could be matched, as the features and matches obtained were very robust to the variations in the image data and the perspective transform between two successive images. Therefore, continuous and complete panorama reconstruction should be obtainable with this novel method.

In experiment 3 an attempt is made towards reconstructing large view panoramas, using images from a fourth sequence. Unfortunately, motion blur and lack of structure in small areas, limited the length of the sequence and therefore the area that could be reconstructed. However, the consistency of the obtained transform estimation shows that large view panoramas can be reconstructed. Furthermore shows that the quality of the videos is important as well.

One aspect of keypoint based image registration that is not covered in this chapter is the detection of these keypoints. In this work, a grid of 31×31 is used as keypoints, where generally these are detected, such as in the detection part of SIFT. In future work, this aspect will also be included, but the exclusion of this aspect can be explained.

First, as stated before, it cannot be guaranteed that the detection will obtain keypoints that are matchable between the two images. In a grid of keypoints, this can be guaranteed with an increased distance, where the maximum possible distance between matchable keypoints, excluding the transformation, is half of the interval between the points on the grid.

Second, a placenta, consisting of a network of blood vessels, has very limited unique features. Moreover, the edge between a blood vessel and the underlying tissue of the placenta is very similar along the whole edge. As a result, a keypoint is generally arbitrarily detected along this edge and consistent keypoint detection cannot be guaranteed. For future work, a keypoint or an interesting area should be selected on the structure of this edge and not the gradient around a point on this edge.

4.5.1 Conclusion

In this chapter, a novel method is described for the second and one of the most crucial steps in panorama reconstruction. This method can extract robust matchable features using a Convolution Neural Network, which is trained with a novel matching similarity learning method. Eventhough, keypoints are selected in a grid, the transformation estimation accuracy is improved. In Chapter 5 a method is introduced that can detect stable keypoints.

5

Stable Region Detection for Image Registration

Floris Gaisser, Suzanne H.P. Peeters, Boris A.J. Lenseigne, Pieter P. Jonker and Dick Oepkes,

Adopted from:
Stable Image Registration for In-Vivo Fetoscopic Panorama Reconstruction, Journal of Imaging - Special issue on selected papers of MIUA, 2017.

Abstract

In Chapter 2 and 3 challenges in panorama reconstruction in an in-vivo setting have been detailed. One of these challenges is to obtain keypoints in image pairs that describe the same areas such that these can be matched. This is challenging as state-of-the-art methods have very low performance in these settings. The cause to this challenge is explained in this chapter and is tackled by proposing an innovative approach of applying object detection for stable region detection.

Another challenge is that in the in-vivo setting the visibility condition can vary a lot and that the image registration process can give adverse results from time to time. These unfavourable visibility conditions could be detected and acted upon. Therefore, in this chapter a qualitative measure is obtained to make the panorama reconstruction process more robust to these adverse visibility conditions and the resulting inaccurate image registration.

5.1 Introduction

The previous two chapters showed that the in-vivo setting complicates the image registration process considerably. Chapter 3 investigated the underlying causes and identified some domain specific challenges; First, the visibility condition is complicated by the color and turbidity of the amniotic fluid. Second, the motion of the fetoscope or the body changes the distance to the placenta. Last, the illumination is limited by intensity as it cannot blind the foetus. These aspects result in a very limited range of visibility conditions in which the image quality is such that current keypoint methods can be used.

To handle the challenges of an in-vivo fetoscopic setting, Chapter 3 suggested four points of improvement; Improving the keypoint detection and matching method, such that it can handle some of these complex settings. Next, the panorama reconstruction process could be improved, by discarding inaccurate or unavailable image registrations and not to create image registration chains. Furthermore, the visibility condition can be improved by obtaining an image quality measure and giving some form of feedback to the surgeon. Last, also the equipment plays a role in the performance of the panorama reconstruction. A larger viewing angle improves the field of view and a high dynamic range or low light camera will obtain a larger range of feasible visibility conditions.

This chapter revisits the differences and the resulting challenges of in-vivo fetoscopic panorama reconstruction in Section 5.2. Section 5.3 introduces recent developments in deep-learning and details how a neural network can be used to handle the challenges posed by in-vivo fetoscopic panorama reconstruction. The proposed approach will be evaluated in Section 5.4 according to the given requirements. Finally, a discussion and conclusion will be given.

5.2 Challenges of In-Vivo Setting

In [34] we described key aspects in which an in-vivo setting differs from an ex-vivo setting and we concluded that in contrast with an ex-vivo setting, state-of-the-art keypoint methods have a very limited performance in an in-vivo setting. Therefore other approaches e.g. based on deep learning must be found. In this section we recap the differences in setting and how they influence the image registration between two adjacent fetoscopic images, and we conclude with presenting a set of requirements for a proper image registration in in-vivo settings. The next section then describes the methods we propose to adhere to these requirements.

5.2.1 Differences in Setting

The visibility in fetoscopic images is a key problem that complicates the image registration between two or more images in an in-vivo setting. The first aspect of good visibility is the amount of light as well as an even distribution. In an ex-vivo setting, the amount of light can be completely controlled and positioned. Therefore, an optimal position and an even distribution of light can be obtained. However, in an in-vivo setting this is not the case:

- The amount of light is limited by the light source and cannot be chosen too bright as it might blind the fetus to the point of annoyance such that the fetus becomes restless.
- The amniotic fluid is far from clear as the fetus micurates in it. Moreover, as commonly the case in TTTS, the fetus might release bowel movements due to distress, giving the amniotic fluid a green turbid color. This color and turbidity of the amniotic fluid absorbs light, reducing the distance the light can reach.
- Also, the fetus and particles that float in the amniotic can limit the field of view of the view of the placenta. In Figure 5.1c and Figure 5.2c air bubbles can be observed. However, these are the result of using water in mimicking the in-vivo setting and are not part of the surgical setting.
- The source of light is the fetoscope itself. This results in an uneven distribution of light, which reduces the amount of illumination towards the edge of the view. Furthermore, saturation of the imaging sensor in the center of the image inhibits proper observation of the structure of the placenta

Examples are shown in Figure 5.1. Especially for the green turbid liquid it is difficult for the camera to acquire a proper image, resulting in a large amount of sensor noise.

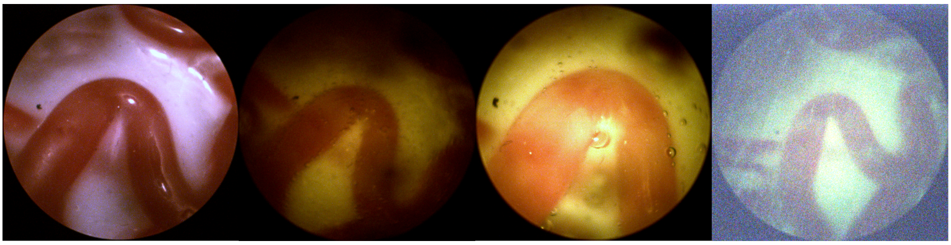


Figure 5.1: (**a**) Ex-vivo view, (**b**) uneven distribution of light, (**c**) too much light saturating the sensor, (**d**) not enough light creating sensor noise

The second aspect of good visibility is the distance to the placenta. With enough distance to the placenta it is possible to observe many different structures on the placenta. In an ex-vivo setting the placenta is generally placed on a flat surface and the fetoscope can be positioned at any distance to the placenta. Furthermore, the fetoscope can be moved laterally with equal distance to the placenta. However, this is not the case in an in-vivo setting:

- The distance to the placenta is limited due to the reduced amount of light.
- The fetoscope is limited in motion at the point of entry. It can only rotate around the point of entry and move forward and backward.
- A lateral movement of the field of view can only be obtained by rotation. Therefore, the lateral change of view also changes the distance to the placenta. This results not only in a change of visible structure, but also a change in illumination.
- The scanning procedure in the in-vivo setting is to follow veins from the umbillical cord and back. Which creates large loops, whereas the ex-vivo setting uses a spiraling motion, which has many small loops.

Figure 5.2a shows an example of an ex-vivo setting with a satisfactory amount of structure. The same area is also show for in-vivo visibility settings. In contrast, showing a nominal example in the in-vivo setting with green turbid liquid, results in Figure 5.2b in a smaller field of view and more pixel noise in the image. To obtain the same field of view the fetoscope can be moved back for a more distant view (Figure 5.2d), but results in an image with too little illumination. To obtain more light and less noise the fetoscope can be moved forward for a closer view (Figure 5.2c), though this results in saturation due to light reflections.

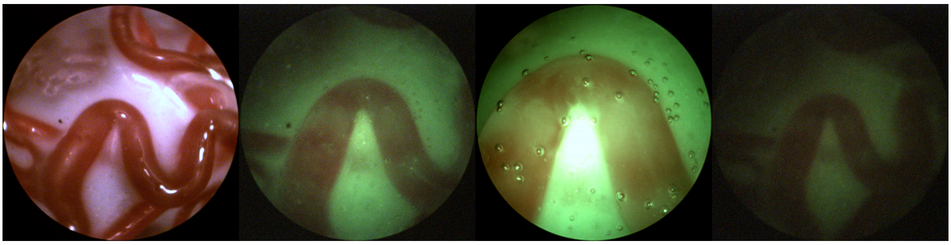


Figure 5.2: *ex-vivo:* (**a**) sufficient structure; *in-vivo:* (**b**) nominal, (**c**) close and bright , (**d**) far and dark

5.2.2 Influence on Image Registration

For panorama reconstruction, it is necessary to correctly find all transformations between adjacent images constituting the panorama. A transformation between two adjacent images can be estimated with a minimum of 4 matches, assuming they are correctly matched and accurately describe the same locations on the placenta. The keypoint matching process assumes that two well matching keypoints describe the same physical point. To find matching keypoints in two images, the area around a keypoint is described with a histogram of gradients. Around corners this generally provides an unique enough description of the keypoint such that it can be matched with a similar keypoint in an adjacent image. Such a corner is dominated by equally strong gradients in two dimensions. In contrast, along edges, such as along a vein, there is a strong gradient perpendicular to the edge and practically no gradient along the edge. Consequently, keypoints selected on an edge are very alike as they have a very similar structure around the point. Moreover, taking sensor noise into account, the histogram of gradients has an additional random component that is often larger than the fine difference between two edges in adjacent images. With a growing variation in the exact location and an increasing number of incorrect matches, the required number of correct matches increases as well. The *LMeDS* transform estimation method is robust to inaccurate locations, but requires at least 50% correct matches to obtain a transformation [85]. Whereas, the *RANSAC* method is sensitive to inaccurate locations, though robust to incorrect matches [31]. Unfortunately there is no method that is robust to both inaccurate locations and incorrect matches.

In an in-vivo setting the limited distance to the placenta reduces the observable structure and the limited amount of light creates sensor noise. Hence, unstable keypoints are detected that are described by similar features and matching keypoints result in many seemingly good, but incorrect matches, describing different points on the placenta, usually along veins. Concluding, in three key aspects traditional keypoint matching methods fail in an in-vivo setting; detecting stable keypoints, reliable matching of keypoints, and obtaining enough matches for a proper estimation of the transform.

5.2.3 Image Registration Requirements

To research other approaches, such as based on deep learning, it is important to specify the requirements for an image registration process that consistently performs its task in an in-vivo setting:

- Keypoints in one image should be reproducible in another image and both should accurately describe the same physical location on the placenta

- The features describing a keypoint in one image should be so unique that the matching keypoint in another image has almost the same unique features
- Keypoints in one image for which no matching keypoint is found in the other image should have such unique features that it is not incorrectly matched to keypoints in that other image at different locations
- The image registration process should be able to detect whether an obtained transformation is incorrect in order to exclude it from the panorama reconstruction.

The section below describes the method we propose to adhere to these requirements.

5.3 Method

In recent years, deep-learning neural networks have been applied in many different fields, tackling various complex problems [64]. This approach is successful because it has the ability to learn any complex task without having knowledge on how to solve the task, as long as the desired output is known and enough training data is available. A deep learned network consists of a pipeline of trainable layers, which makes it possible to train the network to handle compound structures.

Convolutional layers are very suitable to extract relevant data from structured data such as images. It is comparable to convolutional filtering the image, but then with filter coefficients that are trained instead of coefficients determined by a user. A convolutional layer has a set of filters that is moved over the input image extracting relevant structures everywhere in the image. This can be applied in many different applications, notably in image classification [117].

In this work we propose a deep convolution neural network to tackle the challenges stated in the previous section. With it we will:

- Detect stable regions on the veins of the placenta
- Extract matchable features from these regions
- Learn a visibility and matchability measure of an image

These steps are detailed in the next sub-sections.

5.3.1 Stable Region Detector

Soon after the introduction of deep learning, this approach was also applied to the detection of keypoints [116, 136, 143]. These methods are similar to hand-crafted methods such as SIFT and ORB, but have the advantage that the networks can be trained to select keypoints that are more apt for matching and image registration. Although these networks are often trained with keypoints detected by a handcrafted method, this is not very suitable for our case and another way of obtaining a keypoint training set needs to be found.

Image registration requires the detection of stable keypoints, but it is yet unclear what defines a stable keypoint not being a corner. A straight edge (Figure 5.3a) constraints the keypoint in one direction. This is also the case for a circular edge, when rotation is also taken into account, as shown in Figure 5.3b. However, keypoints with the same curvature can be matched. On curved edges, having an additional change in scale, the matching becomes more unique, but not unique enough to do the job (Figure 5.3c). Therefore, any edge alone, albeit curved, cannot be considered a source of stable keypoints. We need additional information to make the keypoint unique.

Consequently, we propose to define stable keypoints being center points on the medial axis of the veins. As both sides of the vein are curves of different curvature they provide independent constraint dimensions making the point more unique. When also the width of the vein is taken into account this constraints the detection also in the dimension of scale, as shown in Figure 5.3d. This makes our proposed method less a keypoint detector but rather a region detector; we use three instead of two independent dimensions.

Since our approach resembles region / object detection rather than keypoint detection, we investigated also Region Convolutional Neural Networks such as RCNN [41], Fast-RCNN [40], Faster-RCNN [100] and SSD [74], which have been developed to detect and classify objects in images. Earlier methods such as RCNN and Fast-RCNN used external region proposal methods, but Faster-CNN and SSD use

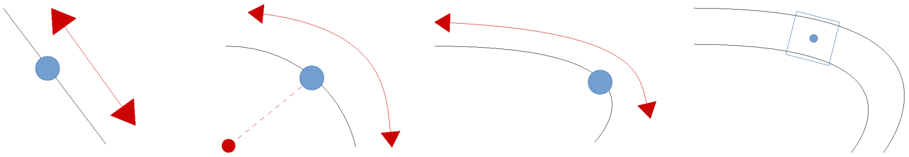


Figure 5.3: Constraints on (**a**) edge, (**b**) circular, (**c**) curve, (**d**) veins

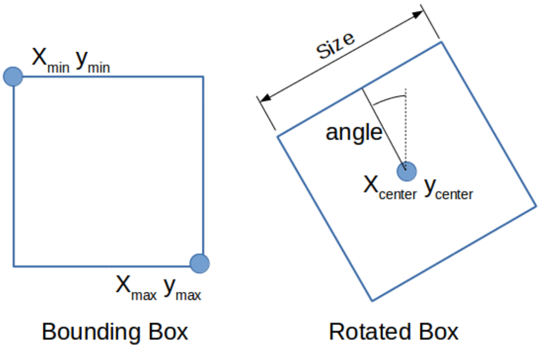


Figure 5.4: (**a**) Definition of Bounding Box (BBox) (**b**) Definition of Rotated Box (RBox)

the same convolutional network for classification as for region proposals, where SSD detects objects at multiple scales. Therefore, this last method was chosen as basis for our stable region detection method.

The SSD method detects regions by defining bounding boxes with their min and max corners as shown in Figure 5.4a. These are learned by training the neural network to output the location of the two corners for each feature cell according to their *default boxes*. An additional classification layer learns the detection probability of each class in every default box. If the classification layer outputs a positive classification, the matching output of the detection layer is used for localization of the classified object. We refer to the Faster-RCNN [100] and SSD [74] papers for more details on the specifics on how to train these detectors.

In order to detect stable regions on the placenta, we propose to detect square areas on the veins. However, the bounding boxes as defined by SSD are not suitable to describe the orientation of the vein. Therefore, we extent SSD and redefine the default boxes by the center, the size, and the angle of the box, as shown in Figure 5.4b.

The ground truth of these detections is obtained by manually annotating the center and the radius of the veins in the images. Taking the gradient of these annotations, also the direction of the vein is defined. An example of such annotation of the veins is shown in Figure 5.5.

It is interesting to note that the definition of our points of stable-regions are similar to that of keypoints. Similar to a keypoint we also extract features around a location of a rotated box. But whereas keypoints are solely defined by a point, a scale and

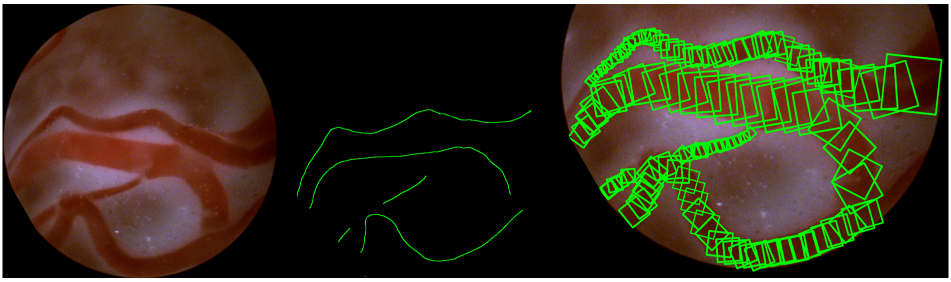


Figure 5.5: (**a**) sample image (**b**) annotated center line (**c**) selection of annotated RBoxes

an orientation of the keypoint, we restrict the possible locations to be at the center of a vein. This makes them stable and within some margin also reproducible.

5.3.2 Stable Matching

The second challenge in the image registration process is to extract features that are descriptive enough for the proper matching of keypoints. In [116], this was achieved by training with positive and negative samples, using Euclidean distance to measure similarity in a Siamese CNN. This is similar to [33] where patches were selected in a grid to extract features that were trained in a Siamese CNN with contrastive loss.

In this paper we extended the SSD architecture similarly to [33]. An additional convolution layer extracts a feature for the detection of Contrastive Loss. Furthermore, every detection is fine-tuned with its matching performance such that detections that are difficult to match are assigned a lower probability to be detected. In this way we remain with matchable features.

5.3.3 Qualitative measures

Our last challenge is to obtain a measure of success for the image registration. This can be used to guide the surgeon or/and his assistant. For this, a qualitative measure is trained by using the matching performance which was used to train matchable features. Since the images registration is highly influenced by the visibility, we define two more outputs to describe this visibility. One describes the amount of illumination and the other describes the distance to the placenta. The visibility is defined as optimal in nominal illumination and distance conditions.

These outputs provide an indication about the performance of the image registration. In case of bad registration the images can be discarded in the process. However, to obtain a sequence of images that is continuous, the surgical team should be included in the process, i.e. the surgeon should be made aware that the panorama reconstruction process has lost position. Furthermore, an assistant controlling the light intensity should be made aware of the illumination condition to actively adjust this.

5.3.4 Network architecture

The above described contributions are implemented based on the VGG-16 network with SSD as a starting point. In order to detect stable regions, we first associate detection scales with the annotated veins of various sizes and select only the first four levels as a scale space pyramid to detect rotated boxes. Each detection scale by default consists of three layers; first the classification layer for determining if there is a positive detection. Second, the location layer describing the location of the detection and third the prior boxes, describing the template detections. Every scale also passes on the features to the next scale.

Next, for stable matching we change two aspects; First, the SSD network was made into two parallel pipelines as shown in Figure 5.6a. These two networks share their weights as a Siamese Neural Network. Second, each detection scale is extended with an additional convolutional layer to extract a feature describing every detection as shown in Figure 5.6b. These, combined with the region detections can be used to find the matches for image registration.

Finally, to extract a measure for visibility and image registration performance, the bottom most detection scale is extended with a convolution layer, a max pooling layer and a convolution layer for classification.

5.4 Experiments

We performed various experiments to show how our method can handle the image registration challenges encountered in an in-vivo setting. For this we used data from our previous work with various visibility conditions. [34]. For training, we selected two sets of data, an ex-vivo setting and an in-vivo setting, including both nominal conditions for yellow and green amniotic fluid. For each setting a minimum of 25 and a maximum of 42 images were obtained for the same trajectory on the placenta. The number of images vary because of the differences in visibility. In total 745 images were used for various settings.

The training data was augmented by rotating the image in steps of 45 degrees and flipping it, such that 16 variations are obtained. For testing, all variations in visibility are used. Therefore, in nominal conditions 20% of the total set is used for testing and the rest is used for training. For all other visibility conditions all data is used for evaluation.

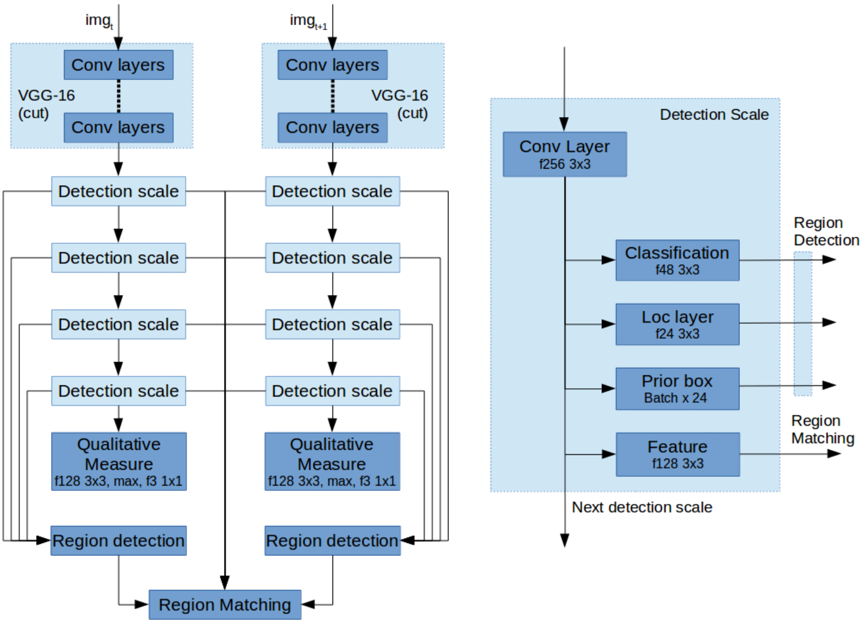


Figure 5.6: (**a**) Left: Detection network architecture (**b**) Right: Architecture of a single detection scale

5.4.1 Experiment 1 - Stable Region Detector

The stable region detector as proposed in Section 5.3.1 should detect the center of the vein. Therefore, we manually annotated the center and the radius of the veins and extracted the direction of the veins. According to the chosen scales and number of cells in the convolutional layers, the closest annotated point is selected as the ground truth and used to train the stable region detection network.

We evaluated the detection performance of these regions as well as their reproducibility for both the bounding boxes (BBox) and rotated boxes (RBox). We applied a confidence threshold of 0.95 and obtained on average 21.0, with a minimum of 11, regions per image in the in-vivo setting. With this high threshold the performance is also very high with 94.4% correctly detected regions. Lowering the threshold provides more regions albeit that the precision goes down very quickly. Below 0.7 only incorrect regions are detected. Therefore, we used this threshold of 0.95 in the rest of our experiments. The results of the BBox detections with more thresholds are shown in Table 5.1.

Table 5.1: Number of detections and precision.

Threshold	ex-vivo		in-vivo	
	BBox	**RBox**	**BBox**	**RBox**
0.95	23.7	25.8	18.1	21.0
	96.6%	97.1%	92.9%	94.4%
0.90	26.9	29.2	21.1	24.8
	89.2%	91.0%	85.5%	86.6%
0.80	34.8	35.9	27.4	30.8
	73.5%	80.1%	71.2%	73.5%
0.70	41.0	43.9	39.0	40.5
	63.4%	66.7%	52.4%	59.1%
0.60	52.8	58.8	50.7	51.8
	49.2%	55.3%	41.7%	50.5%

To determine the reproducibility of the detected regions, the transform between two successive images have been manually established. The ratio of the detections in two adjacent images that describe the same area are obtained by transforming the detections from one image to the other. The reproducible number of detections is on average 81.8% of the detected regions for the ex-vivo, and for the in-vivo settings 76.5% and 73.6%. For all visibility conditions an overview is presented in Table 5.3. It also provides a comparison with the results of the keypoint methods from our previous work.

5.4.2 Experiment 2 - Stable Region Matching

To obtain matchable features we trained the neural network with Contrastive Loss on the matches. To evaluate the matching performance of our approach, the true matches from the previous experiment are used and compared to the number of matched regions. For the nominal ex-vivo setting 73.4% and for the nominal in-vivo settings 69.3% and 58.4% were correctly matched. For these settings all images had enough stable matches to obtain image registration. Furthermore, the mean pixel error was less than 2 pixels using LMeDS as the transform estimation method. Table 5.4 shows the matching performance for the other more challenging settings than nominal. For some visibility conditions an insufficient ratio of correct matches were found to use LMeDS, thus RANSAC was used instead.

5.4.3 Experiment 3 - Qualitative Measure

To obtain a qualitative measure for the matching process as a whole for two adjacent images, the performance of the previous experiment is defined as *bad* if no transformation could be found either by having not enough detected regions or having not enough correctly matched pairs. A *good* performance is defined by more than 50% correct matches and a minimum of 6 correct matches. Which is based upon the requirement of LMeDs of having at least 50% correct matches and having more than 4 matches to handle location inaccuracy. By training an output with these labeled outcomes a measure of matchability could be obtained.

To obtain a qualitative measure of the visibility, a dataset was created containing also the *dark*, *light*, *close* and *far* visibility conditions. For the illumination and distance variation the nominal situation was defined as 0 and the two extremes of the variation as either -1 or 1 and trained with Euclidean loss. Table 5.2 shows the results for the qualitative measures as a ratio of giving a correct indication and an overall correct indication of successful image registration, where these measures are combined for the nominal setting.

Table 5.2: Qualitative Measure Precision

Measure	**Variation**	**Ex-vivo**	**Yellow**	**Green**
Distance	close	65%	60%	42%
	nominal	70%	68%	58%
	far	76%	72%	61%
Illumination	dark	88%	76%	40%
	nominal	90%	82%	60%
	light	93%	87%	83%
Matching		98%	95%	88%
Registration		100%	98%	91%

5.5 Discussion

In this chapter an extension of an SSD network is introduced to detect regions in fetoscopic images with stable matchable features. With the same network architecture also a measure of matchability is obtained for the purpose of obtaining a sufficient set of matchable regions of consistent quality for proper image registration.

In experiment 1 it is shown that it is possible to detect stable regions on the placenta based on the medial axis of veins, under visibility conditions encountered in an in-vivo setting. Compared to keypoint methods this approach only detects a limited number of regions, albeit that the number of reproducible regions is much higher and more consistent over all adjacent images in a trajectory.

The method to learn matchable feature was evaluated in Experiment 2. It showed that in better visibility conditions, a high percentage of correct matches could be obtained and that the number of correct matches is especially in darker settings reduced. Therefore, in the more complicated settings sometimes not enough matches could be found to obtain a transformation. However, in the nominal settings for 100% of the images sufficient matches could be found to obtain a transform.

These results show again that the visibility greatly complicates the in-vivo setting. First, for both the yellow and green-turbid liquid, the darker conditions have not enough contrast to provide the required detail to detect enough regions and extract matchable features. Next, the distance to the placenta also reduces the amount of regions that can be detected, resulting in not enough matches to either use LMeDS or obtain a transform estimation. Last, for the green-turbid settings, many images contain a large amount of sensor noise. These images provide a large number of keypoints, however with our region detection method, almost no stable region could be detected.

The transform estimation precision is not as accurate as expected. It seems that also our region detection method does not describe the same physical location uniquely enough. A more accurate transform estimation should be obtainable with dense optimization. This is anyway required for panorama reconstruction of large sequences without loops, though not implemented in this work.

As stated it will still be very difficult to estimate a correct transform for all different visibility conditions. Therefore, in these cases it is important to be able to detect that the visibility condition is not suited for image registration. Experiment 3 evaluates the three qualitative measures defined and their combination for image registration. In most cases it is possible to detect whether the image is suitable for image registration. Furthermore, as the visibility condition is of great influence on the construction of the panorama image, this visibility should be communicated to the surgical team such that they can adjust the visibility at certain points on the panorama.

5.5.1 Conclusion

The aim of this chapter is to improve the panorama reconstruction process for in-vivo fetoscopic imaging based on the four recommended points of improvement as described in Chapter 3. First, the keypoint detection method is replaced by an extension of the SSD method to detect stable regions defined on the veins on the placenta. Second, SSD is extended with the method of Chapter 4 to extract matchable features. Next, the panorama reconstruction process was improved, by detecting the complicating visibility conditions for the image registration and discarding improperly matched image pairs. Furthermore, a measure of the visibility condition was extracted such that it can be fed back to the surgical team. In this way, fetoscopic images of higher matchability might be obtained by a retry of the surgical team.

The above improvements now achieve a more reliable and accurate sensing and enables panorama reconstruction for in-vivo fetoscopic images. Furthermore, for the task of TTTS laser coagulation, the surgeon is provided with information about the quality of the image data and the reconstruction performance. In this way the surgeon can obtain an overview of the placenta which provides more relevant information and enables him to perform the surgery better.

5.6 Detailed Results

The next two pages show detailed results for the experiments and are compared with results from chapter 4.

Table 5.3: Experiment 1: keypoints / regions detected

Setting	**condition**		**Detected**					**Reproducible**				
			SIFT	SURF	ORB	BBox	RBox	SIFT	SURF	ORB	BBox	RBox
ex-vivo	nominal		269	643	470	23.7	25.8	10.9%	23.3%	17.6%	80.1%	**81.8**%
Yellow	dark	close	14	31	21	8.6	9.1	13.1%	21.5%	27.7%	30.1%	32.5%
	dark	nominal	7	26	5	10.2	10.7	15.4%	31.4%	8.4%	33.0%	31.9%
	dark	far	22	50	15	11.2	12.8	15.4%	26.2%	15.8%	34.4%	37.0%
	nominal	close	42	132	57	12.9	13.2	23.3%	25.4%	34.8%	62.7%	62.8%
	nominal	nominal	24	110	21	20.4	23.4	16.5%	25.0%	21.1%	73.6%	**76.5**%
	nominal	far	74	174	73	20.7	23.8	17.6%	25.2%	23.8%	70.1%	71.2%
	light	close	97	350	108	12.0	11.9	22.4%	27.0%	35.3%	60.4%	60.1%
	light	nominal	110	525	129	16.1	17.3	19.5%	33.8%	23.3%	65.6%	66.7%
	light	far	38	183	32	18.7	18.8	13.2%	24.1%	15.7%	69.6%	70.0%
Green	dark	close	24	11	25	4.6	4.9	27.9%	25.3%	33.4%	5.1%	4.8%
	dark	nominal	11	13	10	4.5	4.6	27.2%	26.1%	15.6%	6.0%	6.1%
	dark	far	210	192	54	2.7	2.8	4.2%	7.3%	0.5%	1.3%	1.3%
	nominal	close	94	147	137	11.7	13.8	26.1%	31.4%	30.9%	57.5%	61.1%
	nominal	nominal	239	401	98	15.8	18.6	11.3%	24.4%	15.5%	68.6%	**73.6**%
	nominal	far	1000	1000	776	3.1	2.8	17.4%	23.0%	18.6%	2.5%	2.6%
	light	close	246	434	320	9.3	9.7	32.6%	36.7%	34.6%	55.9%	60.8%
	light	nominal	1000	1000	578	2.3	2.7	19.9%	18.4%	21.4%	2.6%	2.1%
	light	far	1000	1000	844	1.8	2.4	19.9%	23.2%	22.2%	2.4%	2.5%

Table 5.4: Experiment 2: Matches found for transform with LMeDS or * RANSAC

Setting	**condition**		**correctly matched**		**Sufficient matches**		**pixel error**	
			BBox	**RBox**	**BBox**	**RBox**	**BBox**	**RBox**
ex-vivo	nominal		71.3%	73.4%	100%	100%	2.1 ± 0.8 px	**1.9** ± **0.7 px**
Yellow	dark	close	24.7%	25.7%	0.0%	0.0%	−−	−−
	dark	nominal	25.8%	25.8%	0.0%	0.0%	−−	−−
	dark	far	31.7%	32.4%	0.0%	0.0%	−−	−−
	nominal	close	45.5%	45.8%	39.8%	40.9%	3.1 ± 1.4 px *	3.1 ± 1.3 px *
	nominal	nominal	65.2%	69.3%	100%	100%	2.0 ± 0.9 px	**1.9** ± **0.8 px**
	nominal	far	63.7%	67.1%	100%	100%	2.1 ± 0.8 px	1.9 ± 0.6 px
	light	close	42.6%	42.8%	32.6%	34.9%	3.4 ± 1.3 px *	3.2 ± 1.3 px *
	light	nominal	55.1%	55.9%	96.6%	100%	2.4 ± 1.1 px	1.9 ± 0.7 px
	light	far	58.3%	60.4%	100%	100%	2.1 ± 0.8 px	1.9 ± 0.7 px
Green	dark	close	−−	−−	−−	−−	−−	−−
	dark	nominal	−−	−−	−−	−−	−−	−−
	dark	far	−−	−−	−−	−−	−−	−−
	nominal	close	41.7%	49.3%	2.4%	47.6%	−−	3.2 ± 1.2 px *
	nominal	nominal	55.7%	58.4%	100%	100%	2.5 ± 0.9 px	**2.1** ± **0.8 px**
	nominal	far	−−	−−	−−	−−	−−	−−
	light	close	35.4%	35.5%	0.0%	2.4%	−−	−−
	light	nominal	−−	−−	−−	−−	−−	−−
	light	far	−−	−−	−−	−−	−−	−−

Part II

Road User Perception in Automated Driving

The field of Automated Driving focuses on developing methods and applications that can support the driver in driving a car and ultimately completely taking over the role of the driver. This goal requires many solutions to challenges in various aspects of the driving task as is detailed in Chapter 6. The goal of this thesis is to illustrate that through correct understanding of the task deep learning algorithms can be applied more effectively and used to resolve most tasks. To illustrate this, Chapter 6 analyses the driving task and shows that road user perception can benefit from effective application of deep learning methods to obtain an understanding of the current and future state of the dynamic environment in which automated driving systems drive. In Chapter 7 the advantages of radars, which can detect and localize objects efficiently are combined with the advantages of camera's, which contain the data to effectively classify these objects. A neural network combines the data from these sensors and uses similarity learning to combine *learning by doing* and *trial and error* learning approaches to effectively detect road users in real-time on a real automated vehicle. In Chapter 8 a linguistic sequence-to-sequence Recurrent Neural Network (RNN) is used to model the motion of road users in relation to the road and predict their future trajectories. This work shows that by formatting the data differently, the task can be learned much more effectively. In Chapter 9 the previous work is extended by making the RNN encode the data such that no reformatting of the data is needed to relate the state of the road user to the road structure. Furthermore, this RNN is combined with social pooling to model the interaction between road users to more accurately and reliably predict the future trajectories of road users.

6

Road User Perception in Automated Driving

Abstract

Advanced Drive Assistance Systems (ADAS) aim to support the driver in the driving task, and ultimately achieve Automated Driving where the system completely takes over the driving task. Road user perception is one of the major components to achieve this goal. This task consists of recognizing, tracking and predicting the future motion of the other participants in traffic. In this chapter road user perception is described in more detail and explained how deep learning methods can improve the performance in this task.

6.1 Introduction

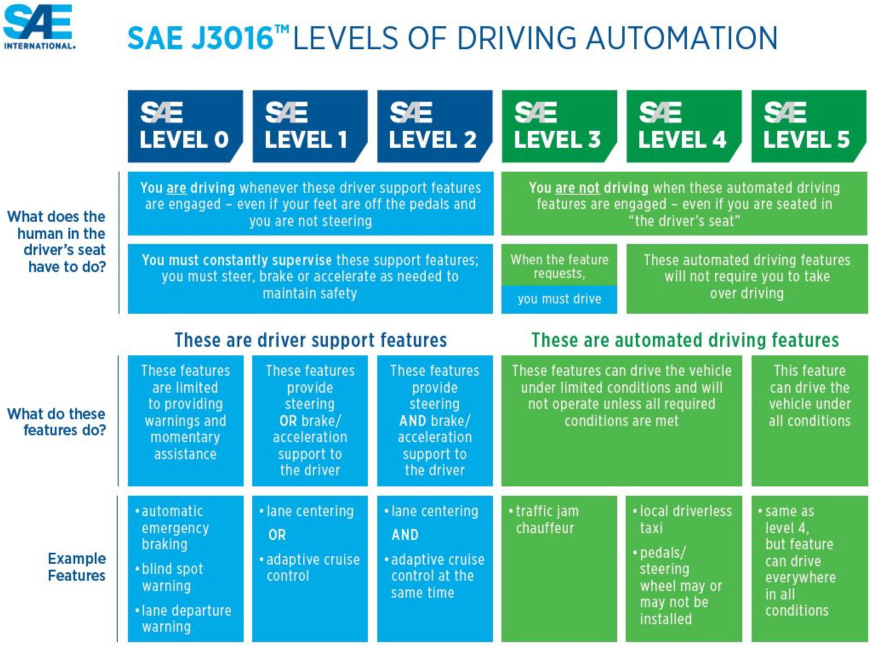
According to the World Health Organization a saturated number of approximately 1.35 million people die each year in traffic accidents and many more are injured. Traffic accidents claim more lives than most diseases, and is the lead cause of deaths among young people aged 15-29 [140]. This is the main cause for the UN to adopt the goal to reduce the amount of lethal accidents with 50% by 2030 in their "2030 Agenda for Sustainable Development" [3]. This goal is adopted by many other organizations and fuels the development of many safety-enhancing technologies. According to a recent report [133] 65%, and even possibly up to 92.6%, of these fatal traffic accidents can be attributed to human error.

Therefore, research is focused on supporting the driver in the driving task in the form of Advanced Driver Assistance Systems (ADAS). Systems such as Night Vision System, Adaptive Front Lights and Surround View Cameras enhance the perception of the human driver, such that accidents caused by oversight are reduced. More advanced systems perceive and take (partial) control of the vehicle; Adaptive Cruise Control (ACC) adapts the speed of the vehicle to the preceding vehicle, Lane-keeping Assist steers the vehicle to stay within the current driving lane and Automated Emergency Braking (AEB) brakes automatically if there is a high possibility of collision. These ADAS systems, control only a small part of the driving task and the driver still has to do most of the driving. Parking Assist is the only system widely spread system that completely takes over a driving task, though only a very simple task. The challenge of these ADAS systems is that the driver can depend too much on these systems or lose focus on the driving task due to over simplification of the task. Therefore, much research is aimed at removing the driver completely from controlling the vehicle, creating automated or fully autonomous vehicles.

Safety is not the only motivation to develop autonomous vehicles. Everyday millions of people commute to work by car and spend a lot of time in the car. This commute is often increased by traffic jams, resulting in the metropolitan areas of the United States a delay about 54 hours per year [51]. Using the commute time more effectively would give up to one day of free time per week. Furthermore, autonomous vehicles could mobilize individuals that are unable to drive by themselves such as youngsters and elderly. But also make public transport more readily available and affordable. The latter could also have a beneficial effect on the traffic congestion by sharing rides.

6.1.1 Automated Vehicles

Development of autonomous vehicles is motivated by the increased safety they can provide, more efficient use of personal time and improved mobility. However, the form of autonomy and its area of application can vary. Therefore, the Society of Automotive Engineers (SAE) has developed a taxonomy for levels of automation [4] as detailed in Figure 6.1.



SAE J3016™ LEVELS OF DRIVING AUTOMATION

	SAE LEVEL 0	SAE LEVEL 1	SAE LEVEL 2	SAE LEVEL 3	SAE LEVEL 4	SAE LEVEL 5
What does the human in the driver's seat have to do?	You are driving whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering			You are not driving when these automated driving features are engaged – even if you are seated in “the driver's seat”		
	You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety			When the feature requests, you must drive	These automated driving features will not require you to take over driving	
	These are driver support features			These are automated driving features		
What do these features do?	These features are limited to providing warnings and momentary assistance	These features provide steering OR brake/ acceleration support to the driver	These features provide steering AND brake/ acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met		This feature can drive the vehicle under all conditions
Example Features	• automatic emergency braking • blind spot warning • lane departure warning	• lane centering OR • adaptive cruise control	• lane centering AND • adaptive cruise control at the same time	• traffic jam chauffeur	• local driverless taxi • pedals/ steering wheel may or may not be installed	• same as level 4, but feature can drive everywhere in all conditions

Figure 6.1: The J3016 SAE levels of automation [4]

The previously described ADAS systems can be categorized as SAE levels 1 and 2. In these cases the driver still has some part in the driving task. For levels 3 and above another categorization can be made. One part of automated vehicles focus on automating highway driving, mainly car manufactures focus on this type of automation, since automating the commute to work is a strong motivation for their customer base. The second group of automated vehicles focus on urban automated driving. This can come in the form of self-driving taxis as developed by companies such as Waymo, Cruise, Uber etc. that will be able to drive anywhere in a large area such as a city. Another form of urban automation comes from (small) self-driving shuttles, that aims to automate public transport on limited routes. In this work, the focus will be on the latter form of urban automation.

These automated vehicles will be able to drive pre-defined routes and perform all aspects of the driving task. This requires a complete set of functionality containing localization, path planning, motion control, object perception and many more. A few of these will be highlighted in order to provide a basic understanding of what is required for an autonomous vehicle.

Localization is the process of finding the position of the ego-vehicle in the environment. This can be done absolute, with for example a gps sensor, describing the position in a global coordinate system. Relative localization describes the position relative the road or lane the vehicle is driving on, by for example lidar map or in-lane localization. And differential localization is describing the current position relative to the previous time step, with for example an imu, odometry or visual odometry. All have the goal to describe the state of the ego-vehicle such that the automated driving systems can make decisions on its next actions.

Road user perception is the process of observing the ego-vehicles environment such that other traffic participants are taken into account. Many different type of sensors are used in literature. A camera is generally used to classify the type of object that is perceived. A lidar and radar are used to localize and detect objects. Or a fused approach of these sensors can be used. More details on road user perception is given in the next section.

Path planning and motion control are the steps of planning and executing a driving motion in its environment. This first requires localization to know where the ego-vehicle is and an understanding of the road infrastructure such that the ego-vehicle is driving in the right position. Also the path planning needs to know what other traffic participants are doing such that in accordance with the traffic rules driving actions can be made.

6.1.2 Challenges of Automated Vehicles

The development of autonomous vehicles is facing many complex challenges of which a few are highlighted;

- All aspects of the driving task have to be precise and reliable. For example localization has to be accurate to a certain degree, for example less than 10 cm otherwise there is a high chance of colliding with something while driving.

- Understanding the infrastructure that is driven in, meaning knowing where the road is and what the static traffic rules are, such as priority lanes and traffic lights.

- Understanding the intention of other traffic participants. Knowing where other road users are is not sufficient, the vehicle needs to know what the future actions of these road users will be.
- The vehicle needs to drive safely and avoid collisions at all time. Making the right decisions in the driving task is quite complex and even humans make mistakes. Furthermore, sometimes these goals can be in conflict with each other.

Part of this thesis focusses on the last challenges by analysing the road user perception task and how to effectively apply deep learning methods to solve these challenges.

6.2 Road User Perception in Automated Driving

The complete task of driving a vehicle is complex and consists of many different tasks that are only used in specific situations, such as parking a car, changing a lane etc. In this thesis only the tasks involving road user perception are considered. Generally these tasks can be described in the Sense-Think-Act paradigm as shown in Figure 6.2. Here road user perception is considered the main sensing step providing the path planning thinking step with sufficient information about the (future) states of the dynamic environment such that a safe path can be obtained, which is subsequently executed.

Road user perception has as its goal to perceive and understand the intention of other traffic participants. Some distinctions can be made with the term object detection as is often used in literature. Object recognition has as the goal of classifying the type of object (object classification) and obtaining its location (object detection). In autonomous driving one is mostly only interested in the traffic participants. This includes pedestrians, cyclists and all type of vehicles moving on or close to the road, but also unclassified objects present on the road driven on. This means that immovable objects that are next to the road and thus not participating in traffic are considered irrelevant. Therefore, the term *road user recognition* is the detection and classification of the relevant objects defined as the objects that are participating in

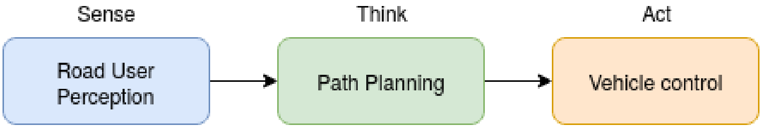


Figure 6.2: Description of driving tasks involving road user perception

traffic, *road users*. Subsequently, road user perception consists of road user recognition followed by tracking of the state of this road user over time and prediction of its intentions. These three components of road user perception are illustrated according the Sense-Think-Act paradigm in Figure 6.3.

The application of deep learning has already found its way into road user perception in recent years [22, 23, 24, 63]. Therefore, the following sections will detail the challenges of the steps of road user perception and the application of deep learning to solve them.

6.2.1 Road User Recognition

Road user recognition is the first subtask of the road user perception task as shown in Figure 6.3. The road user recognition and other subtasks have been detailed according to the Sense-Think-Act paradigm in Figure 6.4 where the top row describes road user recognition. The object detection step perceives the environment and extracts the location, a description of the object and optionally additional information such as size or motion information. The thinking step is object classification that uses the object description to determine the object class and finally only the relevant objects, defined as road users, are kept in the act step. In the next paragraphs more is detailed about the object detection and classification steps and the application of deep learning to solve them.

Road user recognition is the process to find all (relevant) road users in the ego-vehicles environment and obtain the position, orientation, size and type of object. Traditionally object recognition is done with computer vision algorithms on images obtained by cameras as they are readily available and easy to use. Furthermore, image based object recognition is a large field of research and is not limited to road user recognition. Advances in deep learning aimed at object recognition, such as RCNN based neural network architectures like FasterRCNN [100], SSD [74] and YOLO [98] pushed the performance much further. However, the performance increase in road user recognition lagged behind [22, 23, 30].

To obtain sufficient performance for the driving task, the road user recognition task was analysed. A few observations could be made; First, the distance to the objects

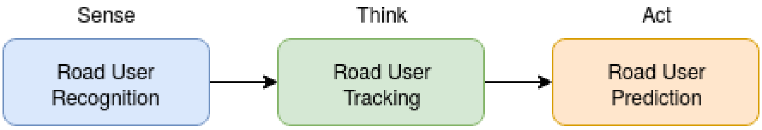


Figure 6.3: Description of the road user perception tasks

varied between a few meters to more than 50 meters. Which results in a large variation in visual size, which is not encountered in other fields of object recognition. Second, the visual difference between relevant objects and the background is limited, for example a car can pass in front of another car resulting in similar visual appearance of a relevant object and background. Also, the visual appearance of the relevant objects varies largely and is generally not sufficiently described in datasets. Lastly, for the driving task the 3D position of a road user is desired, though images cannot accurately describe the distance to an object. Therefore, in road user recognition other type of sensors are considered. Below a short overview of the main type of sensors used in automated vehicles is given:

- Cameras are very cheap and easy to use. They contain data similar to the human vision system. This allows an algorithm to extract information on the type of object and with some accuracy the position and orientation [22, 23]. The downside of camera's is that they are similar to human eyes, and thus greatly impacted by adverse weather conditions such as low standing sun, darkness, rain, snow etc. Furthermore, camera's are not the best sensor for detecting the position and orientation.

- Lidars are often used on automated vehicles, initially for localization. As these sensors provide ranging data, mostly all around the vehicle, it can effectively be used to detect objects. Since, most lidars have a much lower resolution compared to camera's, they lack the details to reliably classify the type of object especially at a larger distances.

- Radars are often used on production vehicles in Adaptive Cruise Control and Automated Emergency Braking systems, because they are much cheaper than lidars and have a very high precision in detecting objects and their relative position. In more modern radars, currently entering the market it is also possible to classify the type of object, although this is challenging as radars work with the approach to classify objects based on their relative motion profiles. Therefore, bikes or pedestrians standing still, cannot be classified as such.

These different type of sensors each have their advantages and disadvantages, thus combining these in an efficient way would beneficial. In more recent years with the rise of lidar and radar technologies, detection technologies started to include these as well [24, 63, 70, 71, 93]. This and the application of deep learning resulted in near human like performance on some datasets. However, various challenges remain such as the applicability in automated vehicles, that requires a short processing time with limited

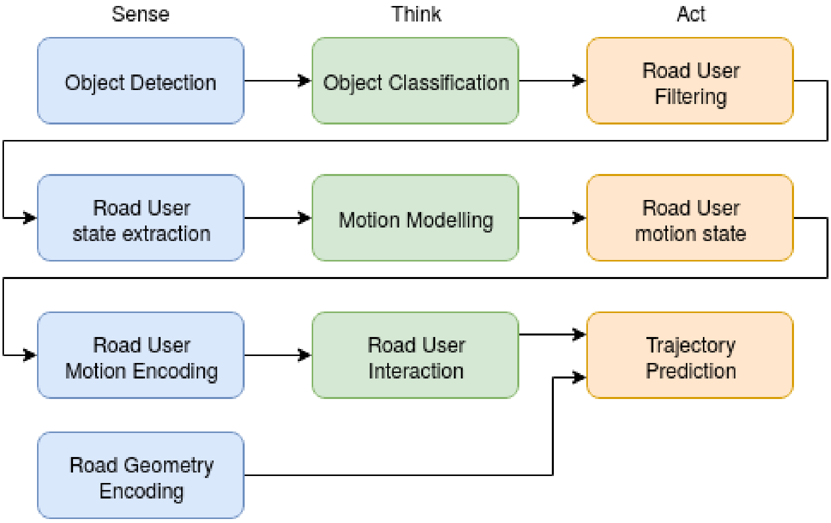


Figure 6.4: Description of the road user perception subtasks

processing power as well as the other steps in the object perception process. To improve the applicability of these methods in automated vehicles Chapter 7 uses radar information to reliably detect objects and uses a deep learned network to efficiently classify these objects as relevant road users.

6.2.2 Road User Tracking

The road user recognition step as described in the previous section, uses data from a single time instance. The goal of road user tracking is to combine the information from multiple time instances to obtain information such as speed and acceleration. On the second row of Figure 6.4 this is described in the Sense-Think-Act paradigm. First, the information describing the state of the road user is extracted. This can be done from different sources, For example, camera object detections combined with lidar object detections each providing part of the state. Next, the motion of these objects is modelled and finally the motion state is obtained. The specific methods to do road user tracking are not detailed as deep learning has found sufficient application and are not relevant for the discussed topics of this thesis.

6.2.3 Road User Prediction

Road user prediction is the last step of the road user perception process with the goal to obtain the intentions of the road users such that the automated vehicle can take these intentions into account when planning its own actions. The field of road user prediction for a long time was directly related to that of road user tracking, as in both steps, a motion model was created of past motion states of the road user as shown in Figure 6.4 with the motion modelling and motion encoding steps. In these methods the motion model was used to create a trajectory by predicting future states [109]. However, as the previous motion is only partly indicative of the intention and the resulting future actions, other indications should be included.

To improve the road user prediction task an analysis was made to the factors that influence the decisions made by a human driver in planning their driving actions. These actions are influenced by various indicators such as the (motion) state of the ego-vehicle, road geometry, road infrastructure (traffic lights, signs etc.) and other road users [66, 95]. These indicators can be categorized into *manoeuvre-aware* and *interaction-aware* methods according to [66]. Manoeuvre-aware methods are dependent on the intended manoeuvre of the road user, often based on prototypical manoeuvres combined with a motion model [7, 80]. Interaction-aware methods are generally based on dynamic Bayesian networks that model the interaction between road users and can initiate interactive manoeuvres such as a lane change, overtaking [130] and car following [73].

The introduction of deep learning methods opens certain possibilities to more effectively incorporate the various indicators of the road users intentions in order to obtain more accurate future trajectories. Chapter 8 focusses on incorporating the road geometry information more effectively as prototypical manoeuvres have limited flexibility and through deep learning this can be made more widely applicable. Following the reasoning from [66] in Chapter 9 a deep learning architecture is used to incorporate both indicator categories such that road and interaction aware trajectory prediction can be achieved.

7

Radar Detection and Camera Classification of Road Users

Floris Gaisser and Pieter. P Jonker,

Adopted from:
Road User Detection on the Autonomous Shuttle WEpod, Machine Vision Applications (MVA), 2017 Fifteenth IAPR International Conference on, pp. 101–104.

Abstract

Over a million fatal accidents occur every year with road vehicles. Road user detection for Advanced Driver Assistance Systems and Autonomous Vehicles could significantly reduce the number of accidents. Despite the research focus on road user detection and such systems, there is a surprising lack of research in real-world applications. In this work, radar and camera data are combined on an autonomous shuttle called '*WEpod*', driving on the public road in Wageningen, The Netherlands. With experiments we show that our method reduces the candidate region margin to $0.2m$ and reduces the miss rate significantly. Furthermore, our specifically trained Convolutional Neural Network improves the performance by 1.4% over vision-based road user detection, and combined with radars we improve by 7.6%. Finally, with our approach we show a performance of 95.1% on the WEpod while driving on the public road.

7.1 Introduction

In the previous chapter reliable recognition of road users was pointed out as one of the main challenges for autonomous vehicles. The challenging aspect of road user recognition is that no object can be missed, because this can have fatal consequences. However only objects that are relevant; road users, should be detected as well as the number of false detections should be limited, as the vehicle should not react to non-existent objects, resulting in unexpected and dangerous behaviour.

In existing methods that only use visual input often false detections are caused by visual appearances in the background or objects at different scales. Therefore, this chapter combines the strong points of a radar sensor, detecting objects and a camera, providing the visual information to classify said object.
To achieve this, first a region proposal method is proposed, that projects the radar detection to an area in the image that is to real world scale of the object. Furthermore, by also incorporating the roll and pitch motion of the vehicle this method is much more accurate compared to other region proposal methods [41].
Next, these proposed regions are then classified using a convolution neural network. Furthermore, *Contrastive Loss* is used to improve the classification performance of the neural network, such that detections that are irrelevant can be classified as such. Lastly, a convolutional layer is used instead of a fully connected layer to classify specific areas within the region of interest such that a more accurate position of the object can be determined.

This chapter is organized as follows. First, Section 7.1.1 gives a short description on related work of fusion-based detection and classification using Convolutional Neural Networks. Section 7.2 gives background on our approach, followed by the experiments in Section 7.3. The results are discussed and a conclusion is given in Section 7.4.1.

7.1.1 Related work

Detection can generally be split in two parts; first, detecting candidate regions of interest and second, classifying these as relevant or irrelevant. In general, two different sensors are used in fusion-based detection. Laser scanners / Lidars are often used for road user detection, however, they depend on light and are obstructed by fog and rain, making them unreliable in many real-world situations. [81, 91, 92, 107].

Radars detect objects with lower frequency electromagnetic wave reflections and are not much influenced by weather conditions. Literature has shown that smaller objects, such as pedestrians and bicyclists, can also be detected [10, 142] and hence using radars is a common choice in real-world applications, although they are seldom combined with visual data [56, 79].

Since all road users are visually distinguishable, a camera is generally well suited for classification. However, an abstraction from raw pixel data into classes is needed, which is generally described as a vector of probabilities for each of the classes.

ConvNets are the state-of-the-art method to classify multi-class visual problems [42]. Multiple convolutional, pooling and rectification layers are combined, so that the visual input is abstracted into lower dimensional data. This data describes the differences and unique visual components of each class. Multiple fully connected layers classify this data into probabilities for each of the classes [42].

These ConvNets have to be trained; thus many images with known classifications are fed into the network and a loss layer provides feedback of the performance to the network [42]. This approach puts an emphasis on learning a general visual description of the class. However, in road-user detection, the difference between a relevant and non-relevant detection also needs to be learned. A Siamese network with contrastive loss is an approach to learn this difference [45] and has shown better results than the traditional class-based training [128]. Therefore, we apply this approach in our system.

7.2 Road User Detection Method

For our fusion-based road user detection method we combine radar detections with classification of visual data. Other work [56, 79] reported similar approaches, however, we improved two aspects of their approach. Firstly, the dynamic candidate regions method fuses radar and image data more accurately. Secondly, the contrastive loss function used in training our ConvNet improves the precision and recall of the classification. In the next two sections we give a detailed description of these two aspects.

7.2.1 Dynamic Candidate Region

In our approach the radar detections are transformed into the camera image as regions of interests, which are then fed into a classifier. A dynamic projection of the detection location to the image plane combined with the detection distance and camera calibration allow us to generate candidate regions of interest at real-world scale in real-time. The method is detailed in the next paragraphs.

Detections in the radar plane are provided by the radar in the form of distance and angle (d_r, θ_r). Assuming that all objects are standing on the ground, they can be transformed into the vehicle coordinate system. This is detailed in Equation 7.1, with the sensor location (x_r, y_r) and orientation (α_r) with respect to the vehicle (X_w, Y_w, Z_w).

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = \begin{bmatrix} \cos\alpha_r & -\sin\alpha_r & x_r \\ \sin\alpha_r & \cos\alpha_r & y_r \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} d_r * \cos\theta_r \\ d_r * \sin\theta_r \\ 1 \end{bmatrix} \tag{7.1}$$

In contrast to the work of Milch and Behrens [79] and Premebida and Nunes [92], we do not consider the road to be flat. Therefore, we incorporate the roll (β) and pitch (γ) of the vehicle, measured by the vehicle's inertia measurement unit. These values are obtained from the gravitational direction and the angles are defined to the horizontal coordinate system and hence they are not Euler angles, which can be seen from the rotation matrix in Equation 7.2.

$$\begin{bmatrix} X_{rp} \\ Y_{rp} \\ Z_{rp} \end{bmatrix} = \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & \cos\beta & \sin\beta \\ \sin\gamma & -\sin\beta & \cos\beta * \cos\gamma \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \tag{7.2}$$

As the detections are rotated with the vehicle's motion, they can be transformed to the camera coordinate system (X_c, Y_c, Z_c). This is described in Equation 7.3, with the camera position (x_c, y_c, z_c) and orientation (α_c). These coordinates can be further projected to image coordinates (u, v), e.g. with the OpenCV [53] *projectPoints* function, also taking lens distortion into account.

$$\begin{bmatrix} Z_c \\ -X_c \\ -Y_c \end{bmatrix} = \begin{bmatrix} \cos\alpha_c & -\sin\alpha_c & 0 & -x_c \\ \sin\alpha_c & \cos\alpha_c & 0 & -y_c \\ 0 & 0 & 1 & -z_c \end{bmatrix} \cdot \begin{bmatrix} X_{rp} \\ Y_{rp} \\ Z_{rp} \\ 1 \end{bmatrix} \tag{7.3}$$

As the distance to the detection is available and ConvNets need a fixed sized input, every candidate region can be created in acordance with its real-world size. To allow pedestrians, cyclists and cars with a maximum height of 2 m to fit, crops of 2.4×2.4 m are created with a 0.2 m margin to compensate for variations. This margin is chosen based on the results of experiment 1 (Section 7.3.1). However, this is not wide enough for vehicles seen from the side. Fortunately, the radar also provides a width measure of the detection so additional crops to both sides can be created.

7.2.2 Classification

Convolutional Neural Networks (ConvNet) have been highly effective in image detection and classification and found their way to fusion-based pedestrian detection

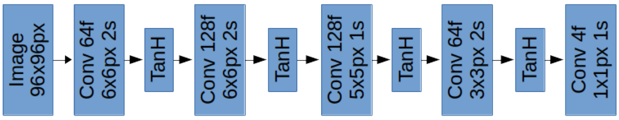


Figure 7.1: Our neural network architecture

[107]. ConvNets learn a representation of the input images with different levels of abstraction. In contrast to the general approach to increase the network's size and complexity to improve the classification performance, we are bound by the available processing capacity. All candidate regions have to be classified within a 66 ms cycle time. In the following paragraphs, we describe our approach.

Neural Networks learn a lower dimensional representation of the input data through various convolutional and rectification layers. These are followed by fully connected layers that can learn the relation between the more abstract representation and the desired label output. Our approach is similar to this and is shown in Figure 7.1, where 64f stands for 64 filters, 6×6 px for the filter size and 2s for a step of 2 px.

The real-world size of the different road-users differ, a pedestrian is about 1×2 m, while a car is about 2×1.5 m seen from behind up to 6×1.5 m seen from the side. However, all these types have to be recognized from a $2.4 \times 2.4m$ crop, thus unrelated information is also present in each input. Therefore, the output is not a single prediction for each class, but rather a grid of 8×8 predictions. Each grid cell represents an area of 1×1 m and can be used to extract the smallest area in which a road-user might be present or not. Since fully connected layers cannot give such an output, this layer is replaced by a convolutional layer, which has a filter size of 1×1, creating a *convolutional fully connected* layer.

The general approach is to learn an abstraction toward the class label with *softmax*. However, two visually similar classes, such as a pedestrian and bicyclist, will often be classified wrongly. The underlying cause can be explained by visualizing the output of the last convolutional layer with t-sne. The outputs of pedestrians and bicyclists overlap as displayed in Figure 7.2. The contrastive loss method is used to increase the separation between these classes and reduce the difference between two similar inputs [45].

Our learning approach is to first train the network normally to obtain basic abstraction. Next, we obtain a set of pairs that have similar abstraction but belong to different classes as well as pairs that belong to the same class but have different abstractions. Except for the convolutional fully connected layer, the network is trained with contrastive loss.

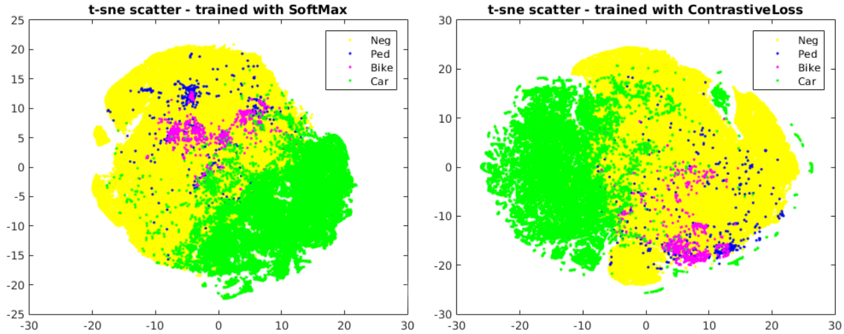


Figure 7.2: Class separation with t-sne

7.3 Experiments

The goal of this chapter is to show road user detection in a real-world application. For this we use the autonomous shuttle WEpod which is driving on the public road of the Wageningen University's campus in the Netherlands. In this paper we use the three front camera-radar pairs of the nine pairs around the vehicle.

In the sections below we report experiments to evaluate our method. However, we also use the KITTI dataset to have a comparable benchmark [36].

7.3.1 Experiment 1 - Dynamic Candidate Regions

The dynamic projection as described in Section 7.2.1, should make the candidate regions more accurate, since we do not assume the road to be flat. To evaluate this, three different types of road sections on the WEpods route are chosen: a straight road, a curve and a speed bump. Three recordings of 20 seconds are taken for the flat and curved road types, while for the speed bumps only 2-3 seconds could be taken as bumps are short.

The horizon is manually annotated in each frame. For the static projection method, the roll and pitch values are set to zero. The pitch accuracy is calculated from the vertical pixel distance in the center, and the roll accuracy from the angle difference between the annotated and projected horizon. Table 7.1 details the accuracy and variation for both projection methods of the roll and pitch in each type of road section.

Furthermore, a margin is calculated from the roll and pitch variations, so that a detection of 2 × 2 m would fit in the candidate region. The first value represents a 2σ variation on a detection (d_r, θ_r) at 10 m distance and 28 deg angle, and the

Table 7.1: Comparison of traditional static and our proposed dynamic candidate regions method

Road	Straight	Curve	Bumps
Static			
Roll [deg]	1.64 ±0.34	1.00 ±0.43	1.68 ±0.67
Pitch [px]	4.2 ±1.3	2.0 ±1.3	9.6 ±10.6
Margin [m]	*0.30–0.75*	*0.23–0.57*	*0.65–1.62*
Dynamic			
Roll [deg]	0.27 ±0.18	0.17 ±0.18	0.12±0.12
Pitch [px]	1.1 ±0.7	0.9 ±0.7	1.3 ±1.4
Margin [m]	**0.09–0.22**	**0.07–0.17**	**0.08–0.21**

second at 25 m and 28 deg. From these results, a margin of 0.2 m is chosen for the candidate regions, resulting in a crop size of 2.4 × 2.4 m.

7.3.2 Experiment 2 - Learning

To compare the classification performance of our contrastive loss training with conventional learning, a training set of images of fully visible pedestrians, bicyclists and cars at a maximum distance of 25 m was created from the KITTI database. For

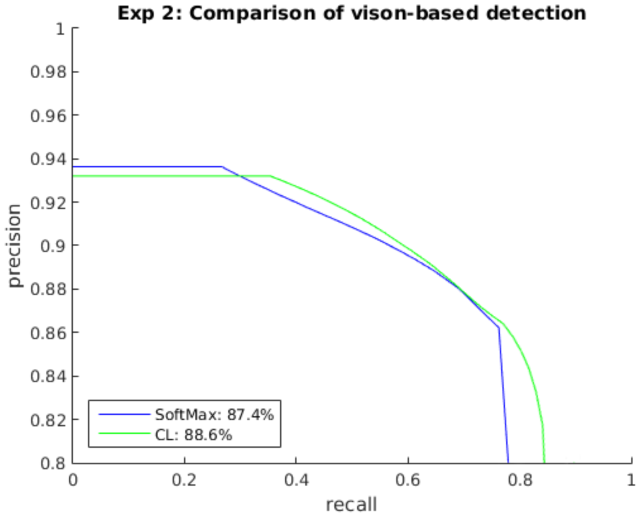


Figure 7.3: Results: Vision-based performance

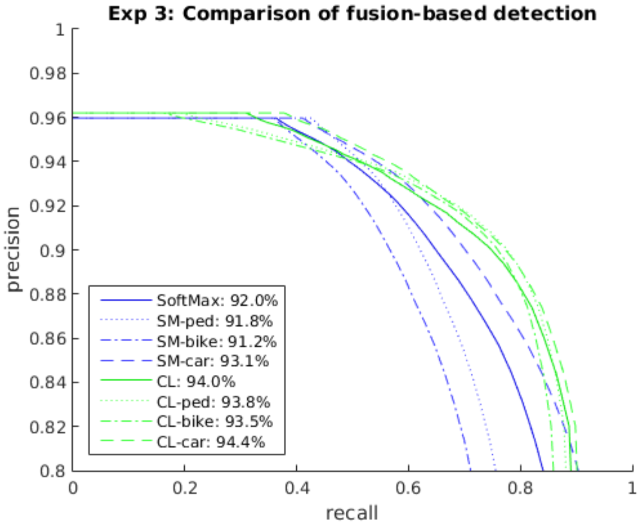


Figure 7.4: Results: Fusion-based performance

evaluation, a sliding projected window approach creates image crops at different distances from the test images. Figure 7.3 details the recall and precision of the the different methods. Figure 7.2 shows the separation of the different classes from the fourth convolutional layer outputs.

7.3.3 Experiment 3 - Radar Fusion

Fusing image classification with radar candidate regions should improve the recall performance. The positive evaluation set is created as 2.4×2.4 m candidate regions from the 3D position of the ground truth and the camera calibration from the KITTI database. A total of 9 crops is created for each true detection, by adding random variation of max $0.3m$ in x and y direction to simulate the radar detection inaccuracy and the proposal accuracy. The negative evaluation set is created from random projected candidate regions. Detections are considered correct if the Intersection over Union (IoU) ≥ 0.5. The same networks as were used in experiment 2 are evaluated, and also per type of road user. The results in Figure 7.4 show that fusion-based detection improves vision-based detection.

7.3.4 Experiment 4 - Real-world Application

While driving on the campus, recordings from the front three sensor pairs were obtained. In total, 423 pedestrians, 864 bicyclists and 1329 cars were manually annotated. The radar detections were used to generate dynamic candidate regions

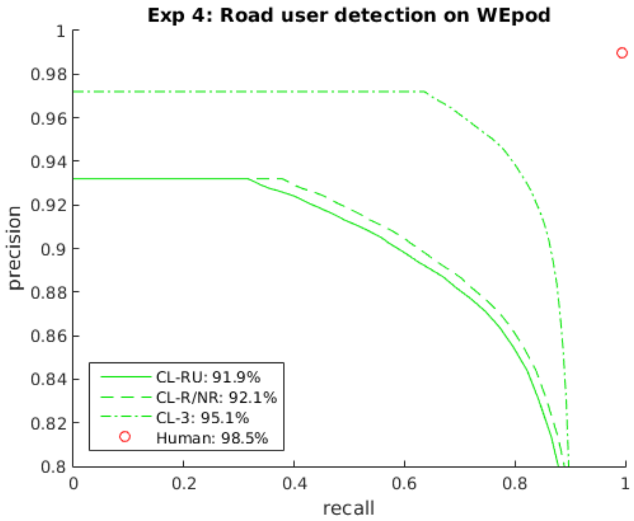


Figure 7.5: Results: Real-world performance

which were classified with our ConvNet. We distinguished between relevant and non-relevant classification (*CL-R/NR*) and classification as the correct type of road user (*CL-RU*). Furthermore, we combined the detection over three successive images and accepted the classification if two are the same (*CL-3*). The results are shown in Figure 7.5.

7.4 Discussion

With experiment 1 we showed that the road cannot assumed to be flat. On a flat road, the candidate region would have an offset up to 0.75 m, missing half of most road users. Moreover, in the case of speed bumps, the candidate region would miss a whole car or most of pedestrians and bicyclists, thus increasing the miss rate. With our dynamic candidate regions method, the offset is reduced to 0.2 m and added as a margin to the candidate region. We reduced processing time by having smaller and more effective candidate regions.

Our ConvNet with contrastive loss improved the performance with 1.4% over the conventional approach, as shown in experiment 2. Experiment 3 combined the dynamic candidate regions with our ConvNet which increased the performance with 7.6%.

The road user detection was benchmarked on an existing dataset, but in future we will present our own dataset based on the WEpod recordings also containing more road user types. Moreover, it will contain more data and different variations compared to other datasets. With more data we expect the performance gap to be closed even further.

Experiment 4 showed that we obtained a performance of 91.9% on road user detection for our WEpod vehicle driving on the public road. This performance is still below the human benchmark of 99% precision and 99.5% recall for a single image. However, by combining the classification of three successive images the performance is increased to 95.1%. Furthermore, we are much closer to the human benchmark and hence the WEpod can drive safely on the public road.

7.4.1 Conclusion

This chapter has shown successful application of road user detection with radar and camera sensors on an autonomous shuttle driving on the public road. This has shown that through fusing sensors the advantages of each type of sensor can efficiently used to create road user detection methods. With significant advances in the state-of-the-art research, such as FasterRCNN and Single Shot Detection in road user detection, the focus of this part of the thesis shifted to road user prediction research in the next chapters.

8

Trajectory Prediction within Infrastructure

Geetank Raipuria, Floris Gaisser and Pieter P Jonker,

Adopted from:
Road Infrastructure Indicators for Trajectory Prediction, 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 537–543.

Abstract

As described in Chapter 6, safe and comfortable path planning in a dynamic urban environment is essential for autonomous driving. This requires the future trajectories of all other road users in the environment of the vehicle. These trajectories are predicted through modelling the motion and behaviour of these road users. For efficient trajectory prediction only motion indicators are not sufficient. Therefore, in this chapter a novel motion modelling and trajectory prediction is introduced. This model uses a sequence-to-sequence RNN in a curvilinear coordinate system with curvature. With experiments it is shown that performance of road user trajectory prediction benefits from this approach of transforming the motion data into a different coordinate system.

8.1 Introduction

A key aspect in autonomous driving is path planning through a dynamic environment. This environment, such as urban roads, also contain other human driven vehicles. For safe and comfortable driving, it is essential for an autonomous vehicle to ensure timely detection of possible collisions, while avoiding false collision warnings.

Such path planning can be done by considering the future trajectories of the vehicles driving in the vicinity of an autonomous vehicle. The trajectories of these vehicles are unknown and need to be estimated. Human drivers do this intuitively by considering various indicators such as past motion, road structure, turn or braking lights etc. These indicators can also be obtained by the autonomous vehicle and should be considered in the prediction of the future trajectory. However, in the state-of-the-art methods for trajectory prediction, mostly only the past motion is considered in prediction. In a few methods these indicators are used to estimate the intention of the road user and adjust the model for trajectory prediction accordingly.

Therefore, our aim in this chapter is to integrate such indicators in the process of trajectory prediction itself. First, Section 8.2 discusses the state of the art methods on indicators and modelling required for trajectory prediction. Next, a novel method on the infrastructure indicators is introduced in Section 8.3. Furthermore, the adaptation of the sequence-to-sequence RNN for trajectory prediction is detailed. Through experiments in Section 8.4 the applicability of these methods are evaluated for an autonomous vehicle. Finally, the results are discussed and a conclusion is given in Section 8.5.

8.2 Related Work

To predict the future trajectory of other road users, generally a model is created from the available relevant past information. Therefore, this section first discusses various methods to model the state of the object. Second, the relevant data required for the model is discussed.

8.2.1 Modelling

Modelling the state of a road user from continuous information is generally referred to as tracking. An extensive amount of research exists on this topic. Motion models are essential to the task of tracking [13] and trajectory prediction [108]. This section first discusses work on tracking methods and next how these can be used to predict the trajectory of objects.

Modelling the motion of other road users often uses multiple noisy and partial observations of the latent state. By expressing the motion as a linear transformation with added Gaussian noise, this can be modelled as a linear dynamic system. A Kalman Filter (KF) [13] or extensions such as an extended KF (EKF) [108] or Unscented KF (UKF) [129] can be used to model these linear dynamic systems.

Generally, the motion observed from road users can be modelled with a linear dynamic system, though not with only a single model. A pedestrian walking along the road has a constant speed and therefore the model assumes the acceleration to have no effect (Constant Speed model). However a pedestrian standing still has no speed, thus the model would require a different linear representation where the speed also has no effect (Constant Position model). Therefore, a tracking method that can model different behaviour of road users should contain different types of models and switch between them [13, 108].

Such a change in motion is often instantiated by the intention of the road user or other environmental causes. However, these are difficult to directly observe. One approach is to switch between models by fitting all models and determining the best fit or mixture of models such as in IMMs [13]. Another approach is by estimating the intention or modelling the switching directly by Bayesian filters [108]. This allows tracking and motion modelling of objects with changing behaviours.

Three types of methods for trajectory prediction are described in [66]; A *Physics* method is solely based on motion properties such as one of the motion models described above. The *Manoeuvre* method predicts a trajectory using a motion model selected by the intention of the object. The *Interaction* method also includes the influence of other road users.

For manoeuvre methods the intention and the resulting behaviour are estimated, such as *following road* or *changing lane* [38, 55, 66]. Each of these manoeuvres have a corresponding motion model that can be used to predict the trajectory of the object. However, an alternative model can only be selected with additional information indicating a different intention.

Many manoeuvres include another road user such as *following vehicle* or *overtaking vehicle*. Therefore, [66] described the interaction method as one of the trajectory prediction methods. This interaction allows to select different motion models based on the interaction. For example the speed of the preceding vehicle limits the speed of the tracked vehicle. However, the speed of this vehicle is unknown for future moments in time. Thus the current speed of the vehicle is generally used for trajectory prediction and any change in speed of the preceding vehicle is not incorporated.

Therefore, solely switching of a motion model is not sufficient to incorporate intention and interaction of vehicles. We state that inputs which influence the motion model should be directly incorporated into the model. In the example of following a preceding vehicle, the speed of the preceding vehicle should be incorporated into the linear dynamics of the motion model. However, modelling external influence as part of the linear dynamic systems is in many cases very difficult or impractical.

Furthermore, consider a vehicle simply following the road. With a constant velocity motion model the predicted trajectory is a straight line. While for a constant turning rate or acceleration model the trajectory is making a curve (Figure 8.1). However, the vehicle is not only driving in a straight line or only making a (single) turn. The road consists of various straight and curved parts, and thus influencing the vehicles direction and limiting the position. Therefore, we argue that all variations in a vehicles motion are caused by interaction with various aspects of the environment. Fortunately, the environment contains indicators that shows how the environment is influencing the motion. These indicators will be discussed in the next section.

8.2.2 Indicators

In this section we discuss different types of information that can indicate how the future trajectory of an object is influenced. Therefore, we categorize this data into different type of indicators:

- *motion indicators* describe the kinematics of the vehicle, directly used in motion models
- *object indicators* is information displayed by the object.
- *infrastructure indicators* is how the road, traffic signs etc. influence the object.
- *interaction indicators* is how objects influence each other.

The most used and important indicator is that of motion information. This information is described by position, velocity etc. at multiple instances and form the past trajectory of the vehicle. This trajectory is the direct result of the intention of the object. Therefore, it is used by many motion model methods to extract the intention of the object. In turn the intention can be used to select a corresponding motion model that can be used to predict the future trajectory [35, 121].

However, a change in motion is generally the result of a change in manoeuvre. In most cases, knowledge about an intended change of manoeuvre is preferred before it is being executed. Humans can predict a change in manoeuvre quite easily, because

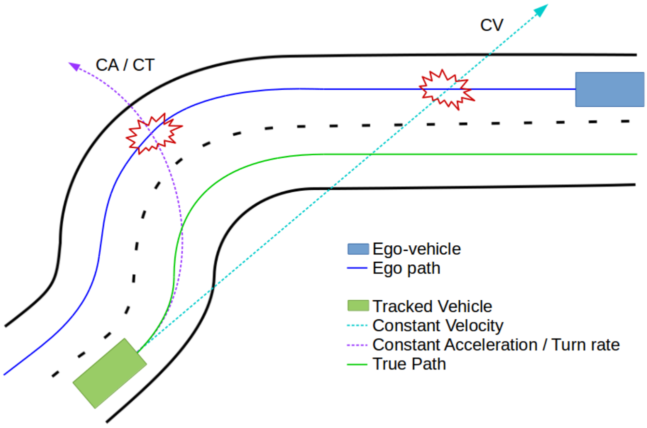


Figure 8.1: Trajectory Prediction using motion model

they also use many other indicators. Therefore, current research is focussed at including other information in the intention extraction [29, 60, 61, 88] .

As described before objects influence each other. For example when entering the highway [29] or following another vehicle [66]. A vehicle changes its trajectory based on the interaction with the other object. A vehicle on the highway may slow down or accelerate to create space to allow another vehicle to enter the highway. The vehicle that wants to enter the highway will adjust its trajectory accordingly.

To support this interaction between objects, many road users indicate their intention before hand to others. For example a vehicle intending to change the lane is by law required to use the *turn indicator light* before hand. Also pedestrians often indicate their intention, though more indirectly. In [32, 61] the body pose and head orientation is used to estimate the persons orientation and intention whether the person is going to cross the road.

Also the structure of the road and other infrastructure components such as traffic lights and signs influence the trajectory of road users. In [88] the structure of intersections is used to extract the intention of a cyclist and select a corresponding motion model for trajectory prediction.

Selecting a specific motion model for every intention will require many different motion models, while they only differ in minor aspects. Therefore, we propose that the motion model should be extended to integrate the specific differences into the model, such as the velocity of the preceding vehicle or the curvature of the road.

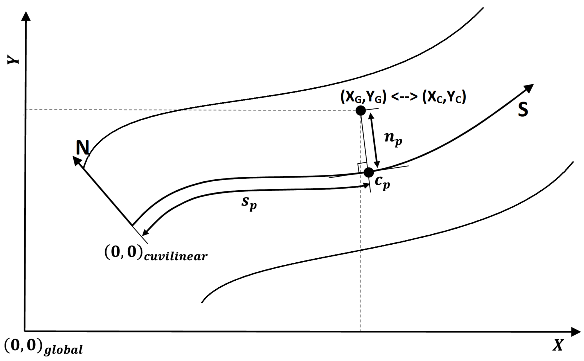


Figure 8.2: The Curvilinear Coordinate system, image adopted from [55]

To achieve this we limit current work to predicting the trajectory of cars following the road. These roads can have various shapes, though have no intersections. Furthermore, we propose a method to effectively include the structure of the road as an integral part of the modelling and prediction. Also, this approach should be easily extendable with other type of indicators.

8.3 Method

The contributions of this work consist of three parts; First, describing the motion information as a function of the road shape in order to integrate the road structure. Second, extract the curvature of the road for better trajectory prediction in curved sections. Last, we propose a sequence-to-sequence RNN to model vehicles and predict their trajectory with multiple indicators.

8.3.1 Curvilinear Coordinate System

In [55] the longitudinal and lateral position as well as velocity of the vehicle with respect to the road is extracted for intention prediction. To achieve this, a non-linear coordinate system is defined as a function of the shape of the road. This section will describe how to obtain this coordinate system and model the motion of a vehicle as a function of the road shape.

Figure 8.2 shows a curved road section, with $[X^G, Y^G]$ in the *Global Cartesian Coordinate System* (GCCS) and $[X^C, Y^C]$ in the *Curvilinear Coordinate System* (CCS). The road geometry is defined as a piecewise cubic spline as defined in Equation 8.1. Where X^G and Y^G is the position in GCCS, s the parametric variable in the range of $[0...k]$ and $a_x, b_x, c_x, d_x, a_y, b_y, c_y, d_y$ constants of the spline.

$$X^G = a_x * s^3 + b_x * s^2 + c_x * s + d_x \tag{8.1}$$

$$Y^G = a_y * s^3 + b_y * s^2 + c_y * s + d_y$$

The position $[X^C, Y^C]$ of the vehicle in CCS is defined by the projection of the position to point c_p on the s-axis and the lateral distance n_p as shown in Figure 8.2. To find c_p the function f in Equation 8.2 is minimized with a non-linear optimization. Where s is the unknown parameter and a_x, b_x, c_x, d_x, a_y, b_y, c_y, d_y are the constants of the spline.

$$f = (a_x * s^3 + b_x * s^2 + c_x * s + d_x - X^G)^2 +$$

$$(a_y * s^3 + b_y * s^2 + c_y * s + d_y - Y^G)^2 \tag{8.2}$$

Note that s is the parametric variable of the spline and is not the distance along the spline to point c_p. To obtain this distance l_p Equation 8.3 can be used.

$$l = \int_0^{s=s_p} dl \cdot ds = \int_0^{s=s_p} \sqrt{\left(\frac{dx}{ds}\right)^2 + \left(\frac{dy}{ds}\right)^2} ds \tag{8.3}$$

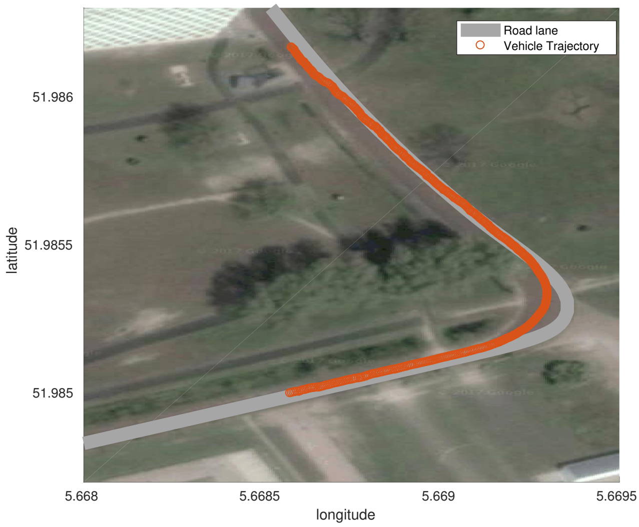


Figure 8.3: Trajectory in GCCS on map and lane

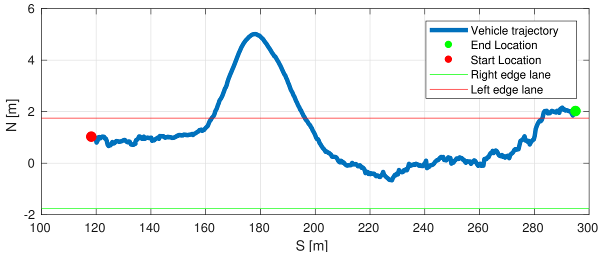


Figure 8.4: Trajectory in Curvilinear Coordinate System

In contrast to [55] a CCS is defined for each direction of the road, such that the driving direction is always positive and the curvature of the inner and outer curves match the road structure. For more details on how the Curvilinear Coordinate System is used we refer to the work [96].

Generally, it can be assumed that vehicles follow the road and don't go off-road intentionally. Therefore, we propose to use the CCS to model the motion of the vehicle as a function of the road structure. To illustrate the benefit of this, Figure 8.1 shows a vehicle driving on a road section. This section of road first makes a slight left turn and then a sharper right turn. Both constant velocity and acceleration models show that the vehicle will go off road, but also intersect with the path of the ego-vehicle. However, the actual trajectory of the vehicle will follow is that of the road and can be modelled with the CCS.

The effect of modelling the motion in CCS is illustrated with a real world example of a vehicle taking a left curve (Figure 8.3). The motion in CCS as shown in Figure 8.4 is similar to how humans think, more left or right of the center of the lane and further down the road. From the described position within the lane in CCS it is also clear that the driver cut the corner. This is can also be noticed in Figure 8.3 when the lane is drawn.

Also for the velocity there is a large difference: Before and after the turn the x and y-velocity change significantly in GCCS as shown in Figure 8.5. However, in CCS the velocity is related to the motion along the road. The s-velocity is the speed along the road and the n-velocity is lateral to the road. When making a perfect turn the s-velocity is the true speed shown on the speedometer. Also, the n-velocity describes the change of position within the lane. Again, the cutting of the corner is observable from the lateral n-velocity in Figure 8.5.

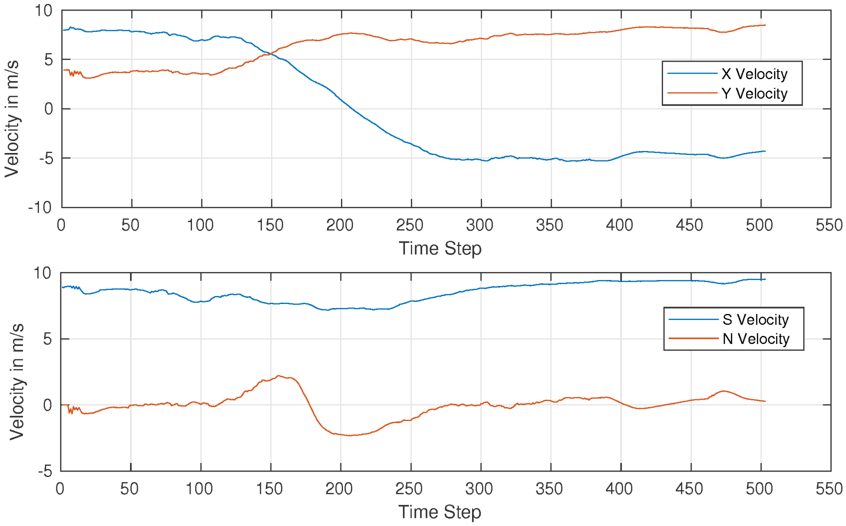


Figure 8.5: Velocity in GCCS (top) and CCS (bottom)

8.3.2 Curvature

The motion in CCS is described as longitudinal and lateral movement along the road. Unfortunately, this also removes information of a change in direction of the road, thus information about turns. Since, drivers generally reduce the velocity because there is a curve, an additional feature is required that reintroduces this relevant information of the road.

Therefore, we define a feature describing the curvature of the road obtained by taking the change in the direction with respect to the curve length as defined in Equation 8.4 [138].

$$k = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\left(\dot{x}^2 + \dot{y}^2\right)^{(3/2)}} \tag{8.4}$$

$$\text{with } \dot{x} = \frac{dx}{ds}, \dot{y} = \frac{dy}{ds}, \ddot{x} = \frac{d^2x}{ds^2}, \ddot{y} = \frac{d^2y}{ds^2}$$

Generally, a vehicle slows down before the curve, due to safety and control of the vehicle, which can be observed from Figure 8.6. However, information about the curvature at the position of the vehicle is not informative, because a driver slows down before the curve and speeds up in towards the end of the curve. Furthermore,

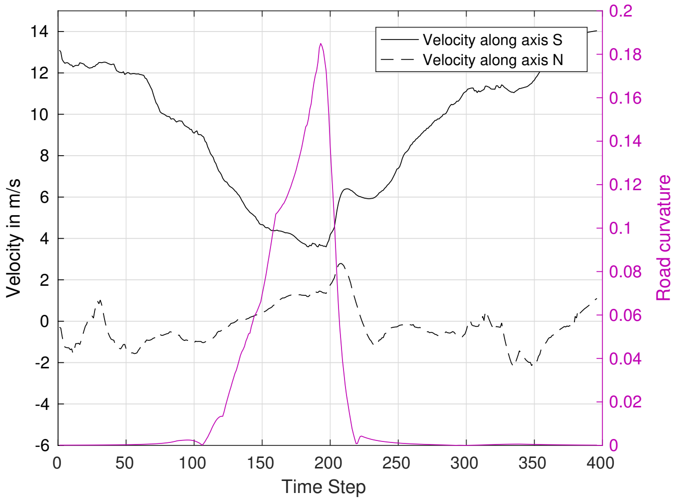


Figure 8.6: Change in Velocity due to Curvature

a fast driving vehicle has to look further ahead as it has to slow down more than a slow driving vehicle. Therefore, the curvature indicator is defined as the curvature 2 seconds ahead of the vehicle. By modelling the slowing-down behaviour as a dynamic system, this look-ahead-time can be estimated more accurately, though more information about the driving style of the vehicle is also required. Therefore, the value of 2 seconds was approximated by observing the behaviour of the recorded vehicles.

8.3.3 Sequence-to-Sequence model

In [122] a sequence-to-sequence RNN model is used to encode a sentence in one language and decode it in a different language. We propose to use the encoder part of this type of network to model the state of the vehicle and use the decoder part for prediction of the vehicles future trajectory. In this section we describe how this network is adapted for vehicle trajectory modelling and prediction with motion and infrastructure indicators.

Figure 8.7 shows our network design, with Long Short Term Memory (LSTM) units [50] as RNN units. The encoder part is fed with the past information, in the form of position and velocity in CCS x_{t-n} as well as the curvature k_{t-n}. The LSTM units don't output any information, and only pass forward the hidden state to the next time step.

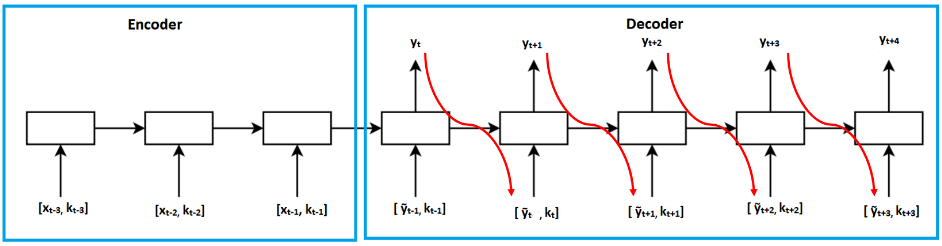


Figure 8.7: Sequence to Sequence RNN model

The decoder part is used for trajectory prediction. Every LSTM unit is fed with the vehicle's position and velocity of the current time step $\widetilde{y}_{t+n}$ combined with the additional indicators such as the curvature k_{t+n}. For training $\widetilde{y}_{t+n}$ is known as x_{t+n}, though for deployment this is not known. As the output of an unit is the predicted state at the next time step, which can be used as the input of the next unit. Therefore, we pass the predicted state $\widetilde{y}_{t+n-1}$ to the next unit in the sequence-to-sequence RNN.

However, this causes a discrepancy in training and inference, which leads to poor performance [11]. A scheduled sampling during training can be used, where x_{t+n} is selected with a probability η or alternatively $\widetilde{y}_{t+n-1}$. At the start of training $\eta = 1$, selecting the training data. As training progresses η is reduced such that the network is trained with the same settings as during inference.

In contrast to translation, for trajectory prediction, the inputs and outputs are a continuous sequence of trajectory data. As a result we do not reverse the order of the input sentence. Also, we do not use the special end-of-sentence symbol in our model, as there is no specific end of the sequence.

8.4 Experiments

The contributions of this work consists of two parts; First, the use of road structure and curvature as infrastructure indicators in trajectory prediction. Second, a novel modelling approach for trajectory prediction using these infrastructure indicators along with motion indicators. With these contributions we aim to improve trajectory prediction on curved road sections. We perform experiments to evaluate our proposed approach against a conventional motion based prediction method on real data.

For our experiments trajectories of human driven vehicles under natural driving condition were recorded along with road infrastructure information. The data was collected with a test vehicle, the WEpod, in the region of Wageningen, The Netherlands. The test vehicle is equipped with 6 IBEO LUX LIDAR sensors running at 25Hz. Each LIDAR has a 110 degree (horizontal) FOV and 4 vertical planes. Data from these LIDAR sensors is used to detect and record vehicles moving around the test vehicle.

The IBEO system provides vehicle position $[X, Y]$, velocity $[\dot{X}, \dot{Y}]$ and heading angle θ in the WEpods (ego-vehicle) reference frame. The vehicle states were converted to the global, Universal Transverse Mercator (UTM) coordinate system (GCCS), using GPS localization.

Figure 8.8 shows the road sections on which the vehicle trajectories were recorded. These were selected to record vehicle trajectories on roads of different curvature, while avoiding the influence of features like pedestrian crossings, complex road design like roundabouts and intersections with traffic lights.

We obtained 285 unique vehicle trajectories, with a minimum length of 7 seconds containing 175 time steps. The trajectory data was segregated with a 4:1 ratio into a training and test set. For trajectories longer than 10 seconds, the trajectory was split into multiple parts at an interval of 3 seconds. This resulted in 496 training trajectories and 143 trajectories used for testing.

Additionally, an artificial dataset was created to pre-train the RNN model. This dataset consists of simulated vehicle trajectories with different driving behaviours in CCS along the longitudinal direction. We simulated vehicle trajectories on straight roads including constant velocity and de/acceleration followed by constant velocity. Vehicle motion on curved roads was simulated considering different radii. Based on the curve radius the maximum safe speed was calculated, and vehicles were simulated

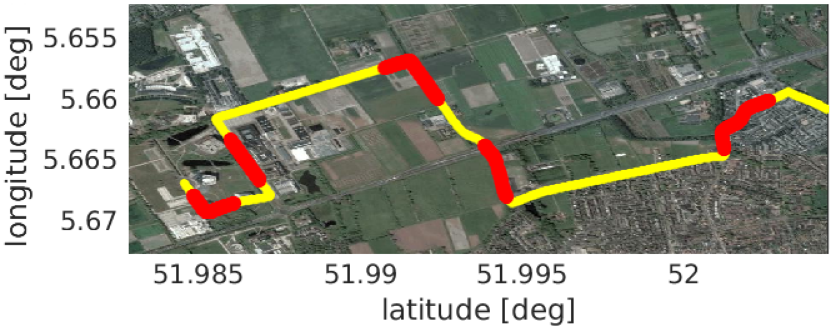


Figure 8.8: Recorded road sections (in red)

to decelerate to a speed below this value. Gaussian noise was added to all states, with mean and variance obtained from LIDAR measurements. The dataset consists of 1/5th simulated trajectories on road with no curvature, and the rest randomly generated for curves of different road curvatures. Training the network with the transfer learning regime gave an 8% improvement in performance and was subsequently used in all experiments.

To compare the trajectory prediction in CCS with GCCS a baseline Interactive Multiple Model filter [13] with constant velocity and acceleration models was used. To compare the performance of the RNN and the curvature indicator, we trained two RNN models one with only motion features as input, and another with both motion features as well as road curvature.

We determined experimentally the best LSTM network architecture, by varying the number of hidden layers between and the number of cells in each layer. This resulted in an architecture of two hidden layers of 275 and 160 cells. The output layer uses basic RNN cells with Rectified Linear Unit (ReLu) non-linearity. Furthermore, we used the Adam [58] optimization algorithm to train the network. The loss is calculated using as Mean Squared Error over the four output states $(X, Y, \dot{X}, \dot{Y})$. The network is trained with a constant learning rate of 10^{-3}.

8.4.1 Experiment 1: Sequence-To-Sequence model

This experiment is designed to establish that the RNN model can perform regression in a non-linear space and model an internal state over multiple samples. Sine-wave prediction is chosen as the regression task. The space is single dimensional, and is described by the function $x = a * sin(2\pi f t + \phi)$. Where a is the amplitude, f the frequency, ϕ the phase and t the independent variable.

The training dataset consists of randomly generated sine-waves with random amplitude, frequency and phase. The first 25 samples of the wave are provided as input to the model, which then predicts the next 25 samples. To make this prediction, the three variables a, f, ϕ need to be modelled internally by the encoder and the value x predicted.

For sin-wave prediction, the RNN model is found to perform best with one hidden layer of 40 LSTM cells. The neural network is trained using Adams optimization [58], with Mean Squared Error loss. Figure 8.9 shows some predicted sin wave from test set samples.

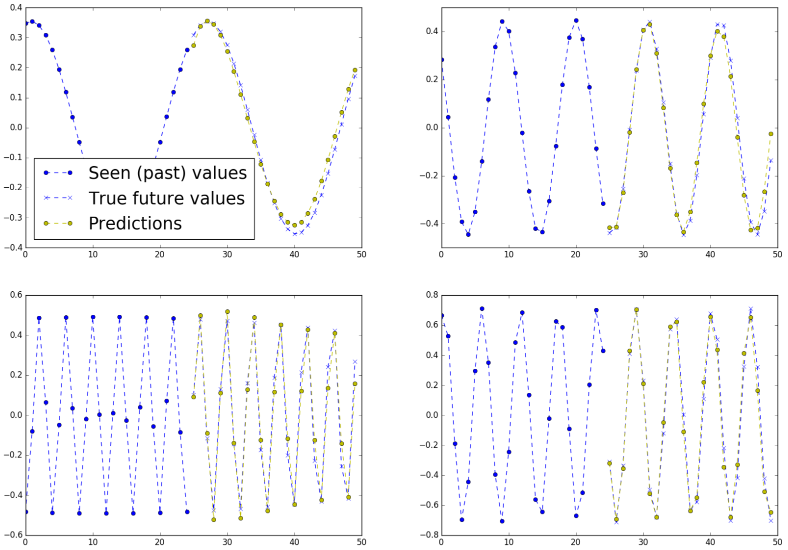


Figure 8.9: Sine-wave prediction with Sequence-to-Sequence RNN

8.4.2 Experiment 2: Curvilinear Coordinate System

To compare trajectories predicted in CCS with those predicted in GCCS, we plot them both in GPS coordinates along with the ground truth trajectory. The IMM prediction model is provided with 2 seconds (50 time steps) of vehicle states as input

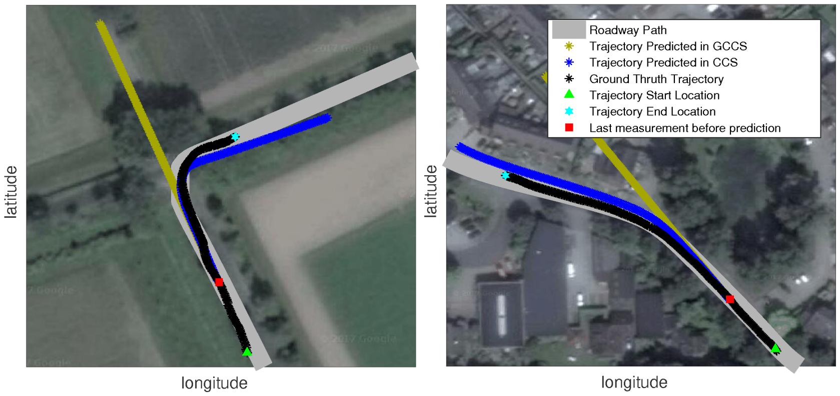

Roadway Path
Trajectory Predicted in GCCS
Trajectory Predicted in CCS
Ground Thruth Trajectory
Trajectory Start Location
Trajectory End Location
Last measurement before prediction
latitude
longitude
latitude
longitude

Figure 8.10: Vehicle trajectories predicted in GCCS and CCS

	IMM		**Motion RNN**		**Curvature RNN**	
axis:	X	Y	X	Y	X	Y
1 sec	0.95	0.70	**0.82**	**0.73**	0.87	0.80
2 sec	1.90	1.39	1.78	**0.86**	**1.55**	0.90
4 sec	4.87	2.91	3.98	**1.24**	**3.19**	1.30
6 sec	9.47	4.48	7.61	**1.33**	**6.20**	1.38

Table 8.1: Mean error [m] for all test trajectories

data and predicts the trajectory for next 8 seconds (200 time steps). Figure 8.10 shows the position plots of two (real) examples of vehicle trajectories predicted in GCCS and CCS. In the left figure makes the road a sharp 90 degree turn, and in the right figure makes the roadway two consecutive turns of about 30 and 45 degrees.

8.4.3 Experiment 3: Curvature Indicator

To make the performance comparison between the three models, IMM, Motion RNN, and Curvature RNN, the models are provided with input data for 25 time steps (1 second), and the error is reported for the predicted vehicle position at 25, 50, 100 and 150 time steps in meters. Error in velocity would be reflected in position, as a result the velocity predictions are not used as a separate metrics to compare the three models.

We segregate the test trajectories into three groups based on the curvature. Figure 8.11 shows the performance of the three models for each type of test trajectory. Table 8.1 gives the mean error over all trajectories for the three models.

8.5 Discussion

In this chapter the Curvilinear Coordinate System was used to incorporate the structure of the road into modelling vehicle motion. Experiment 2 aimed to show that CCS improves the trajectory prediction over GCCS. Since in CCS the motion is a function of the road structure, modelling this information is directly integrated. This was then also clearly showed with two examples of the results in Figure 8.10, where trajectory prediction in GCCS has no knowledge about the curve and predicted a straight line. Note here that no single motion model would be successful as has been shown in Figure 8.1.

Since information about any change in direction of the road is eliminated by CCS, the velocity prediction does not correspond to that of normal driving behaviour. Generally

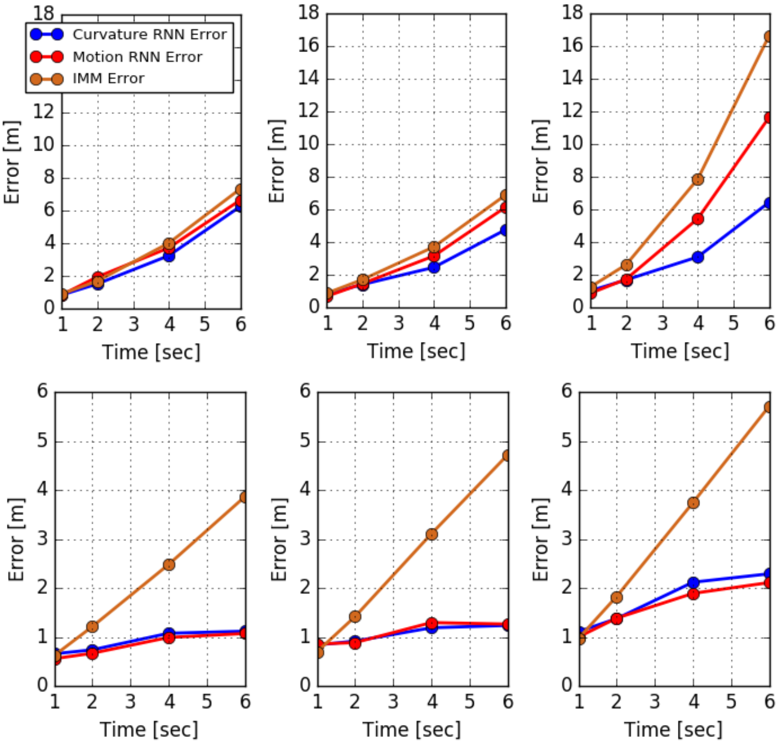


Figure 8.11: Prediction Error (in m) on the *X* (top) and *Y* (bottom) axis for test trajectories over prediction time. Left: trajectories with almost no curvature; Center: slight curvature; Right: large curvature.

a vehicle slows down before a curve. Therefore, an indicator was introduced in the form of curvature. Experiment 3 shows with Figure 8.11c that for sharp curves the longitudinal prediction error is much reduced. For 4 seconds prediction the mean error is 3 meters, which is less than the length of a car and can be used for path planning. For longer periods the error is increasing as for 6 seconds prediction the mean error is 6 meters. This can be explained by the fact that for longer periods of time more unmodelled factors influence the vehicle and its trajectory.

It is interesting to note that for the lateral position the curvature provides no improvement in prediction, though this was to be expected. The lateral position within the lane is generally not a result of a curve, but of other factors. However, it is interesting to point out that the RNN seems to learn that vehicles stay within their lane. Which was one of the goals of this work, but was not explicitly defined in training the RNN.

In order to model the motion and predict the trajectory of the vehicle, while directly integrating additional indicators such a curvature, we introduced the sequence-to-sequence RNN. This RNN was used for language translation, but hasn't been used in trajectory prediction (at the moment of writing). Therefore, we devised a toy-example in Experiment 1 to predict sine-wave patterns. A simple LSMT with only a few cell could already model and predict these non-linear patterns. In Experiment 3 the RNN model has shown to clearly outperform IMM in trajectory prediction as it was able to model some level of driver behaviour such as keeping its lane.

By further studying the results of the sequence-to-sequence RNN model, we noticed something peculiar. The predicted position and velocities did not correspond. Differentiating the positions gave different velocities and integrating the velocities gave different positions. This means that the RNN has no understanding of the laws of physics and does not adhere to the kinematic rules of motion. Therefore, in future work this constraint will be added to the loss such that $v_t = (x_t - x_{t-1})/dt$.

One of the aims of this work was to find a modelling method that could be used to incorporated additional indicators in the motion modelling and trajectory prediction. We proposed the sequence-to-sequence RNN and showed with the curvature indicator this was attainable. When studying the results some unmodelled factors became apparent, which we intent to include as indicators in future work.

Some recorded trajectories had to be excluded as a vehicle or bicycle was preceding the vehicle and limiting the velocity of the tracked vehicle. By modelling this interaction with an indicator describing the speed of a preceding road user, this can included in the motion model.

When observing the predicted velocities of vehicles we noticed that most vehicles would drive about 40 km/h, though some roads had a maximum speed of 30 km/h and others had 50 km/h. Therefore, we also intend to include an maximum speed indicator, such that the trajectory prediction will reflect normal driving behaviour on various type of roads.

8.5.1 Conclusion

One of the limitations in this chapter was that only road sections without intersections were used. This was done to avoid the possibility of multiple paths and that of interaction between road users. In order to include these added challenges a few observations can be made; First, multiple paths can be modelled by adding a node in the piece-wise spline where multiple splines branch off. Second, intersections have road users interacting with each other governed by traffic rules. Chapter 9 will include interaction between road users.

9

Trajectory Prediction with Interaction and Road Attention

Tim Resink, Floris Gaisser and Pieter P. Jonker,

Adopted from submission:
Road attention: map-based vehicle trajectory prediction for interaction models.

Abstract

In the previous chapter a RNN model was introduced to predict future trajectories of road users more efficiently by taking the road geometry into account. Next to road geometry also the interaction between road users influences the future trajectories of these road users. To model this interaction Social Pooling is a suitable method. However, this requires that the spatial relationship between the interacting road users is retained. Therefore, transformation of the road users states to a Curvilinear Coordinate System, as introduced in the previous chapter, is not applicable. This chapter introduces a novel approach to encode the road geometry and select the relevant part of the road through an attention mechanism. This road attention method can be combined with social pooling in order to create a interaction and road geometry aware RNN model.

9.1 Introduction

Automated Vehicles in our society drives a large field of research. One key task of automated vehicles is to plan a safe and comfortable path in a dynamic environment with other road users. This requires information about the road as well as the other road users with high reliability, as false information can lead to collisions. One major challenge in this task is that of object trajectory prediction. This field focuses on predicting the intentions and the resulting future trajectory of a road user such that the ego-vehicle can plan its actions accordingly. To accurately plan a path at multiple time instances, the states of the surrounding vehicles should be known over time. Therefore, it is desired to obtain sequences of states such as the position, velocity and heading over time, rather than just a classified intention or predicted end state.

This trajectory of future states is obtained by a trajectory prediction method. This task is challenging as it combines different types of information; Previous motion is the most commonly used information source. Through motion modelling a future path can be predicted. However, the past driven path of a vehicle is insufficient information to predict its future behaviour. The future states are constrained by the the geometry of the driven road. Therefore, the road structure needs to be incorporated in trajectory prediction. Furthermore, interacting road users influence each other's future path, thus for a vehicles trajectory prediction its surrounding vehicles have to be taken into account. Even for humans this task is challenging and is one of major causes of dangerous situations when unexpected situations occur.

Therefore, the goal of this chapter is to describe the information structure of these data sources and propose a novel method on how to combine these data sources to predict the future trajectory of a road user. First, the relevant work to motion prediction is detailed in Section 9.2. Next, in Section 9.3 we introduce a novel RNN structure that can efficiently extract motion information of other road users, incorporate interaction between them through social pooling and then predict a trajectory with an attention mechanism related to the road geometry. Through experiments, in Section 9.4, the applicability of our proposed RNN network is evaluated for an autonomous vehicle. Finally, the results are discussed and a conclusion is given in Section 9.5.

9.2 Related work

Classically, vehicle motion prediction is achieved either by modelling and simulating the vehicle as a dynamic system (motion models), or by classifying driving behaviour in a set of manoeuvres and obtaining some prototypical trajectory for each class [7, 66, 80]. In the latter case, a classifier is used on constructed features that describe the observed trajectory. The prototypical trajectory is generated by a manoeuvre-specific motion model or (stochastic) representations of past classified trajectories. Both of these approaches have limitations; Motion models have poorer long-term prediction accuracy, because some states that relate to the driver input (e.g. acceleration and steering rate) cannot be modelled. Manoeuvre classification approaches attempt to capture all driving behaviour in a fixed set of manoeuvre classes, resulting in generalized predictions for each manoeuvre class, omitting case-specific details.

Recently, machine learning approaches are proposed in an attempt to overcome these limitations. Generally, deep learning models for sequential data, called Recurrent Neural Networks (RNNs) are used. An RNN is a parametrized regression model that learns relations between sequential vehicles states. These models can find latent factors, such as driving style, which are difficult to define manually. Modern RNNs such as the Long-Short Term Memory (LSTM) [50] or the Gated Recurrent Unit (GRU) [26] are enhanced RNNs with internal memory for improved accuracy over longer sequences. These models overcome the limitation of motion models by modelling the latent factors in driving behaviour, as well as the limitations of prototypical trajectories by not classifying all driving behaviour as a fixed set of manoeuvres. For sequence prediction tasks with RNNs, the encoder-decoder architecture [122] has shown to be a suitable choice. This architecture originates from the field of Natural Language Processing, but has shown to be valuable for vehicle motion prediction as well [28, 57, 65, 82, 97]. However, these approaches remain limited as they only use motion information. Therefore, models that incorporate either road geometry or interaction between vehicles are discussed in the next sections.

Road geometry

One crucial aspect that influences human driving behaviour is the geometry of the road. Incorporating the geometry of the road in prediction has long been a subject of research. In [86], a semantic map is used to classify lane changes with the distance to the centrelines of all present lanes. In [59], this map-based prediction is taken a step further by extracting features related to the position and orientation with respect to entrances and exits of intersections. By describing the road layout ahead of the vehicle, more accurate velocity profiles can be obtained that are typical to such layouts. In [88], the road topology is used to transform cyclist trajectories to a more general coordinate system where all trajectories are initially aligned and motion models according to the turn direction and sharpness can be deployed. A similar philosophy is used in [55], where such models are used on trajectories that are defined in a Curvilinear Coordinate System (CCS) on the road centrelines, making all lane-following motions approximately linear. To incorporate the road geometry [97] defines trajectories in CCS with the road curvature as feature and uses an encoder-decoder model to obtain more appropriate velocity profiles in turns. All these methods define the trajectories in some local coordinate system, where the global configuration among different dynamic agents is lost.

Various methods avoid defining a local coordinate system by using visual information from the scene to include road geometry. In [65], road geometry is included with a Convolutional Neural Network (CNN) feature extractor on front view images. In [106], top view images of the road topology are used with a CNN and attention mechanisms to extract relevant features of the road. The drawback of such approaches lies in the information source, rather than the modelling method. The road information from the front-view images in [65] is prone to occlusion, whereas the top-view images of the road topology in [106] are an information resource that is mostly unavailable in practical applications.

Interaction

Modelling interaction among dynamic agents has been an active challenge for motion prediction. Interaction couples the behaviour of several dynamic agents, making predictions of these agents' behaviour mutually dependent. Classically, interaction-aware prediction is attempted by coupling graph-based models [14, 66]. Such models are computationally expensive and suffer from a growth in uncertainty due to the coupling of uncertain future states of multiple agents, also known as the freezing robot problem [110]. The complexity of the modelling problem can be relaxed by assuming that all interaction is unilateral, or by limiting the number of interacting vehicles [39].

Recently, interaction methods were designed based on RNN encoder-decoder models, focused on the field of pedestrian motion prediction. Social LSTMs are first proposed in [2], modelling dynamic pedestrian behaviour in dense crowds. This method, also known as social pooling, uses individual encoded trajectories in a grid-based pooling layer to obtain interaction-aware predictions from the decoder. Variants on social pooling have been proposed since, improving the effectiveness of the pooling layer. In [28], a CNN is included in the pooling module to better account for the spatial configuration of the scene. In [43] the grid-based method is replaced with a relative position embedding method to reduce computational cost without a decrease in model accuracy. Currently, convolutional social pooling is the only interaction-aware RNN method that has been applied on vehicles.

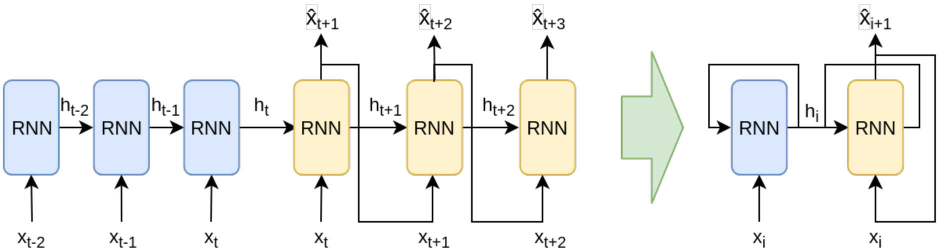


Figure 9.1: a) Expanded Encoder-Decoder model. b) Actual Encoder-Decoder model

9.3 Model

This work focuses on creating a deep-learning based model that can incorporate both the road geometry and interaction between vehicles in trajectory prediction. To incorporate the interaction between vehicles, social pooling has show great promise and is detailed more in Section 9.3.2. However, social pooling requires that the spatial relationship between vehicles is retained. Therefore, a novel method is proposed in Section 9.3.3 that can encode the road geometry and supply this to the decoder component of this model. First, the architecture of such a model is detailed in Section 9.3.1.

9.3.1 Encoder-Decoder Model

The encoder-decoder model originates from the field of Natural Language Processing and is also referred to as the sequence-to-sequence model, as a sequence of words is processed into another sequence of words. In trajectory prediction, a sequence of vehicle states is encoded by a set of (stacked) RNN units. Each unit encodes the input states x_i of the vehicle over time into a hidden state h_i. This hidden state serves as a memory block where relevant sequence information is contained. Figure 9.1a shows an unenrolled RNN model, where at different time steps an RNN unit is used. Whereas, in the actual model a single RNN unit is iteratively used to encode the sequence as shown in Figure 9.1b.

At the end of a sequence, h_t represents an encoding of the vehicle trajectory in the form of a fixed length feature vector. A decoder, constructed with a similar RNN, can use this feature vector h_t as its own memory block and decode it into future vehicle states, generating a trajectory prediction. The input of the decoder x_i are its own predictions from the previous time step $\hat{x}_{i-1}$, allowing it to keep track of its previous predictions. This encoder-decoder architecture is detailed in Figure 9.1b and adopted in Figure 9.2a with our proposed modules.

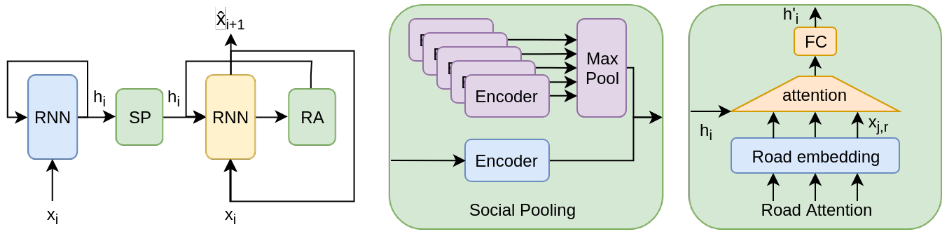


Figure 9.2: a) Encoder-Decoder model with Social Pooling and Road Attention modules b) Social Pooling module with other embeddings in purple. c) Road Attention module with in orange the attention mechanism.

9.3.2 Social Pooling

The goal of social pooling is to pool the embeddings of multiple surrounding vehicles into a single feature vector. In [2] these embeddings were combined in a spatial grid, which are converted into a single vector again by a fully connected layer. Also, with a grid based approach multiple objects in a grid cell as well as empty cells pose challenges.

In [43] the spatial grid is replaced by a relative position embedding added to the hidden states, which is then pooled with the symmetric max-function. In this work the latter approach is adopted, where we select up to 5 vehicles around the ego-vehicle. After creating the pooled states of the surrounding vehicles, this is concatenated with the ego-vehicles embedding and supplied to the decoder part as shown in Figure 9.2b.

9.3.3 Road Attention

The RNN decoder needs to know about the road geometry. In [97] this was done by transforming the ego-vehicle to a curvilinear coordinate system that describes the road geometry. However, in order to use social pooling, the global spatial relationship between vehicles needs to be retained. Therefore, in this work an attention mechanism is used to select the relevant information for the embedded road geometry as shown in Figure 9.2c.

The road geometry can be described by j way-points on the centerline of the lane. Since these way-points are also a sequence, they can be encoded similarly to the encoder used to encode the vehicle states. However, the output states $x_{r,j}$ of all RNN units in the encoder RNN are used to create a sequence of the encoded road geometry ahead of the vehicle. Next, the Bahdanau attention method [6] is used to obtain attention weights α_t to select the right part of the encoded road geometry based on the hidden state h_i and the encoded road geometry $x_{r,j}$. With these weights the context state c_i can be calculated:

$$a_{t,j} = W_b tanh(W_a[h_i; x_{r,j}]) \tag{9.1}$$

$$\alpha_{t,j} = \frac{e^{a_{t,j}}}{\sum\limits_j e^{a_{t,j}}} \tag{9.2}$$

$$c_i = \sum_j \alpha_{t,j} x_{r,j} \tag{9.3}$$

Here W_a and W_b are learned weights of the attention mechanism. The context vector is concatenated with the hidden state at the previous time step and then passed through a fully connected (FC) layer to obtain a context aware GRU hidden state $h_i^{'}$.

9.3.4 Training and Implementation Details

The input to the model is a past observed trajectory consisting of a sequence of 30 normalized Cartesian coordinates over time, sampled at 10 Hz. This input is first embedded through a 1D temporal convolution layer, which allows the model to extract state information over time related to velocity and acceleration, as stated in [65]. The convolution layer consists of a kernel size of 3, and 16 convolution filters, and zero padding is used to generate an embedded input of equal length to the original input. The RNN encoder-decoder architecture that generates predicted trajectories based on the embedded input consists of Gated Recurrent Units. The encoder consists of a single GRU unit of 48 units, whereas the decoder consists of 2 stacked GRU units each with 48 parameters. The decoder generates an output of 48 parameters at every time step, which are mapped to 2 Cartesian coordinates with a linear projection layer. The road encoder consists of a smaller GRU unit of 36 parameters. For the social pooling, the relative position between the subject vehicle and surrounding vehicle is embedded in the surrounding vehicle hidden state with a fully connected layer. The reduced hidden state from the social pooling and the subject vehicle hidden state are concatenated and compressed to 48 parameters, the size of the decoder hidden state.

For training, all 5583 vehicles are used to randomly extract a sequence from their entire recorded trajectory. The trajectory is split up in 30 states (3s) as the past observed trajectory, and 60 states (6s) as the ground truth trajectory that is to be predicted. 75% of the sequences is used for training, and the rest for validation. The training data is fed to the model in batches of 16 trajectories, and trained for 200 epochs. For social pooling, up to 5 vehicles in the surrounding are taken into account, consisting of the preceding vehicle and up to 2 vehicles in the adjacent lanes if they are in the proximity of the subject vehicle.

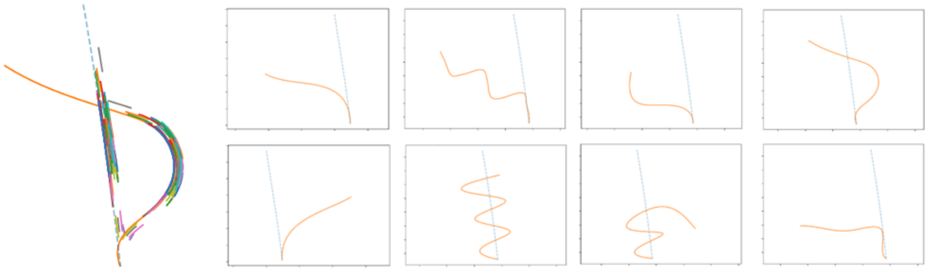


Figure 9.3: a) trajectories before and after curvilinear transformation b) 8 additionally introduced road geometries.

9.4 Experiments

In this section, the experiments are highlighted. The performance of the proposed Road Attention is examined on the public Next Generation Simulation Dataset (NGSIM). This is achieved by comparing multiple models that are enhanced with Road Attention and/or Social Pooling, to clearly distinguish the impact of the different model components.

9.4.1 Data

The Next Generation Simulation (NGSIM) Interstate 80 (i80) provides recordings of approximately 6000 vehicles in the San Francisco Bay Area. These are a mixture of low density traffic and peak hour congestion, with varying intensity of vehicle interaction. Furthermore, accurate map information is available, providing road geometry. One downside of the i80 dataset is that the variation in road geometry is limited and only reflects a small part of the (urban) driving environments. To this end, we propose an augmented version of the i80 dataset, which we will call the i80 curved (i80c) dataset. The augmentation is inspired by the Curvilinear Coordinate System (CCS) approach used in [97].

In this augmentation, we convert the positions of the vehicle x_i from Cartesian (X, Y) coordinates to curvilinear (L, N) coordinates. A cubic spline is fitted on the centreline of the original road shape. For all vehicles positions the longitudinal (L) distance and lateral (N) distance is obtained. By generating a different cubic spline, and converting the curvilinear coordinates back to Cartesian coordinates, the vehicle positions are converted to a different type of road shape. These transformations are detailed in [96]. In total, 8 different road shapes have been created and added to the original road geometry data as shown in Figure 9.3b. The main goal of these road geometries is to introduce strongly varied road curvature. By introducing varied and sudden curves, extrapolation from previous time instances by motion modelling will not be effective.

9.4.2 Evaluation metrics

In the experiments, the model performances are assessed on the i80 and i80c data. The models will obtain an input trajectory recording of 3 seconds (30 states), and predict over a prediction horizon of 6 seconds (60 states). The 6 second prediction horizon is needed for the autonomous vehicle to anticipate the driving behaviour of other vehicles sufficiently well.

The Mean Absolute Error (MAE) of the longitudinal and lateral predictions and ground truths will be used for evaluation. These MAE_{lon} and MAE_{lat} are obtained by converting the predictions to the curvilinear coordinates: $(x,\ y) \to (s,\ n)$, and computing the mean absolute error between these prediction coordinates and the ground truth with:

$$\varepsilon_{l_1-lon} = \frac{1}{N}\sum_{i=0}^{N}|s_i - \hat{s}_i| \tag{9.4}$$

$$\varepsilon_{l_1-lat} = \frac{1}{N}\sum_{i=0}^{N}|n_i - \hat{n}_i|. \tag{9.5}$$

This metric dissects the error into a longitudinal and lateral component, which gives much more valuable information compared to an Euclidean error measure. The lateral error describes how well the trajectory can predict the vehicle's position within its lane. Whereas the longitudinal error describes how well the driving actions and the congruent speed of a vehicle are predicted.

Table 9.1: Prediction performance for all models on i80 dataset.

MAE [m]	RNN		SP		RA		SPRA	
time[s]	lon	lat	lon	lat	lon	lat	lon	lat
1	0.71	0.13	0.88	0.17	0.69	0.14	0.72	0.18
2	1.50	0.23	1.47	0.24	1.33	0.20	1.41	0.25
3	2.44	0.32	2.17	0.30	2.31	0.27	2.20	0.30
4	3.63	0.42	3.09	0.35	3.40	0.34	3.06	0.35
5	5.04	0.49	4.13	0.41	4.65	0.40	4.10	0.41
6	6.51	0.56	**5.32**	0.48	6.11	**0.47**	5.34	0.48

Table 9.2: Prediction performance for all models on i80c dataset.

MAE [m]	RNN		SP		RA		SPRA	
time[s]	lon	lat	lon	lat	lon	lat	lon	lat
1	0.84	0.56	1.00	0.57	0.83	0.51	0.95	0.80
2	1.74	0.96	1.73	0.80	1.54	0.58	1.55	0.81
3	2.82	1.37	2.68	1.16	2.51	0.69	2.27	0.83
4	4.11	1.98	3.77	1.67	3.55	0.76	3.10	0.89
5	5.67	2.77	5.06	2.44	4.88	0.87	4.21	0.99
6	7.60	3.69	6.64	3.06	6.32	**1.04**	**5.43**	1.23

9.4.3 Results

The proposed method is evaluated both on the i80 and i80c dataset extended with Road Attention and/or Social Pooling. The first model is the basic encoder-decoder model (RNN), to obtain a baseline performance. Then, the Road Attention (RA) and Social Pooling (SP) modules are added to the basic model to obtain two more models. Finally, both the Road Attention as well as the Social Pooling are added to the basic model to obtain the full model (SPRA). In Table 9.1 the results on the i80 and in Table 9.2 the results on the i80c dataset are shown.

Table 9.3 compares the convolution social pooling results of [28] as they also use data of the i80 dataset combined with the us-101 dataset, which consists of a straight highway section.

Table 9.3: Euclidean Mean Distance performance comparison with state-of-the-art.

time [s]	CS-LSTM [28]	SP
1	**0.61**	0.81
2	**1.27**	1.33
3	2.09	**2.01**
4	3.10	**2.82**
5	4.37	**3.81**

9.5 Discussion

Table 9.3 show that the results of our proposed social pooling method has similar performance compared to that of convolution based social pooling. Our method embeds the position of the other vehicles in the hidden state instead of creating a spatial grid. This approach is much more computational efficient as in the spatial grid many cells will be empty, but still are used in the calculation of its convolution.

The long term prediction in Table 9.1 on the i80 dataset show the importance of Social Pooling. Both models that include the social pooling module greatly increase the prediction accuracy. The road attention module improves performance over the baseline RNN in the section where road information is valuable, namely the on-ramp of the i80 highway. Furthermore, it can be seen that short term prediction performance is very similar. The road curvature is very limited in the i80 dataset where the road attention module provides redundant information to the prediction performance, explaining similar performances between the SP and SPRA models.

On the i80c datasset long term prediction performance shows a major increase in prediction performance with the road attention module, especially in lateral performance. The social pooling module shows similar poor performance as the baseline model on the curved roads. Though, social pooling still improves the longitudinal prediction performance, similar to the i80 dataset. The combined SPRA model combines the advantages of both the social pooling and road attention modules showing improved longitudinal and lateral prediction performance.

For short term prediction, generally the interaction is limited or already indicated through an already initiated manoeuvre, thus a normal RNN can predict this behaviour quite well. Therefore, the SP model diminishes the prediction performance on the short term compared to the baseline RNN, but improves on the long term as can be seen on both the i80 and i80c results.

9.5.1 Conclusion

The aim of this chapter is to improve object trajectory prediction by combining road infrastructure information with the interaction of the surrounding road users. Social pooling has shown in state-of-the-art research that if can efficiently combine the information of multiple road users. However, this method also imposes a constraint of the inability to use a local coordinate system for each road user. Fortunately, the road attention method is able to encode the road infrastructure method efficiently without defining a local coordinate system. Therefore, the combination of these methods achieves a large step in the direction of correct object trajectory prediction.

10

Discussion & Conclussion

Applications of deep learning have expanded drastically over the last decade. This phenomenon has been driven by the increase in computational power as well as availability of large amounts of data. However, not in all type of applications deep learning has been successfully adopted. This can mostly be attributed to limited understanding of how to apply deep learning effectively, as well as the complexity of deep learning network architectures. To provide insight in how to apply deep learning methods efficiently in many types of applications, in this thesis it is stated that it is important to have a thorough understanding of the tasks it needs to perform. Furthermore, to successfully execute these tasks using deep learning methods various aspect should be taken into account. These are summed up, with examples, in the next paragraphs.

Many tasks can be described using the Sense-Think-Act paradigm. In this paradigm the *Sense* step is to perceive the environment and filter data on relevance for the systems task at hand. The *Think* step uses the perceived information as well as it own built up / stored knowledge and based on the goal, it selects useful actions from a (possibly built up) repertoire that when executed alter the state of the environment, making the system progress towards the required goal. This action influences the systems environment and hence new Sense-Think-Act cycles should be performed until the goal of the task is achieved. This definition adequately describes for example tasks performed by a mobile robot finding its way towards a goal or manipulating objects in its environment. However, the paradigm can also be extended to tasks performed by humans.

For a human example, in Chapter 2 a surgeons task of endoscopic laser coagulation therapy for TTTS has been described as shown in Figure 2.5. In this procedure the surgeon has to find and map anastomoses and determine its type (Sense). When all are found a plan is made to coagulated these in the right order (Think), which is then subsequently executed (Act). Note that when we zoom in on this *Act* step, it again can be decomposed into a Sense-Think-Act loop, where the *Sense* step is finding the next anastomosis as depicted in the map, *Think* is how to manipulate the tool and *Act* is performing the coagulation.

For a technical system example, in Chapter 6 the object prediction task has been described in the Sense-Think-Act paradigm as shown in the bottom rows of Figure 6.4. In the sensing step the motion information of the road users is obtained and encoded. In the thinking step reasoning takes place on the interaction between these road users and their actions. Also road information is encoded and the relevant information is extracted. The reasoned actions and road information are then combined in an *Act* step describing the future states of the road user. As reasoned above a way to describes systems is to describe them as nested Sense-Think-Act cycles. In the example above, the object prediction task can be seen as a subtask of an object perception task which in its turn is part of a highly complex Automated Driving task.

Correctly understanding a task and decomposing it into less complex, more achievable subtasks is necessary to solve these tasks using deep learning algorithms. Deep learning is widely applied in perception tasks, this is the *Sense* step in the Sense-Think-Act paradigm, though other steps as well.

In order to solve such tasks with deep learning methods it is important to consider the following three specific aspects for such a perception task; data quality, network architecture and learning method.

First of all, the statistical features, variation and quality of the perceived data highly determine the performance of the successive steps. It is not only important to understand what information needs to be gathered in the *Sense* step, but also in what format the subsequent thinking step needs this information. Therefore, the sensing step is often also divided into a subtask to perceive information and a subtask to filter, abstract and fuse the data into the format needed.

Second, a neural network architecture is used in deep learning to process information to a desired output. This network architecture is set-up by choosing the right type of layers and then how the data will be processed. For instance, if we consider the difference between the FasterRCNN and SSD network architectures, both object recognition algorithms on these networks use convolutional layers although each with a different approach to achieve these convolutions. This can be seen in Chapter 8 where the object motion information was transformed to match the road infrastructure. Whereas in Chapter 9 the network was designed to extract the road infrastructure information to match the object motion information.

Third, the learning method used can influence the performance of the deep learning method. Chapter 4 showed that learning to extract features based on the feature matching performance, improves the performance of the matching process but changes nothing in the performance of feature extraction itself.

The goal of this thesis was to gain insight in how to effectively apply deep learning methods in a variety of perception tasks. Understanding the task at hand is crucial. This was demonstrated in this thesis based on two applications, which conclusions can be found in the next sections.

10.1 Part 1 - Fetoscopy

In Twin-To-Twin Transfusion Syndrome an imbalance in blood flow exists between the twins. In order to resolve the imbalance, a surgeon tries to find connecting blood vessels (shunts) on the placenta and uses laser coagulation to prevent further blood flow. To make sure that the imbalance is equalized, the order of coagulation of the shunts is important. Furthermore, since the procedure entails a certain amount of risk, it is also important that all shunt are found. To achieve this, an overview of the placenta would be beneficial to the successful outcome of the procedure as described in Chapter 2.

Therefore, the goal of the research in Part 1 of this thesis was to reconstruct the placenta from fetoscopic images obtained during the exploratory phase of the procedure and mark locations of the shunts as found by the surgeon. By thoroughly analysing the process of panorama reconstruction important subtasks have been found and detailed in Chapter 2. They are shown in Figure 2.7. In Chapter 3 the challenges to reconstruct a panorama of the placenta encountered by mutual registration of images have been described in detail. These subtasks and challenges have been solved by successful application of deep learning methods and are discussed in the next sections.

10.1.1 Stable Keypoint Detection

Finding stable keypoints in images is the first subtask and traditionally this is done by methods such as SIFT, SURF and ORB. These methods detect corners and edges as keypoints, based on gradient changes in the image. However, Chapter 3 showed that for in-vivo visibility conditions these methods have a limited performance. This lack of performance was explained by limited contrast and an increased noise level due to ill illumination conditions. Furthermore, the structure of the placenta shows a very limited number of corners and mostly consists of edges, which are only stable in one dimension. The experiments in Chapter 3 showed that these methods are not sufficient to effectively find keypoints to do image registration for his task.

In Chapter 5 the Single Shot Detection (SSD) algorithm was introduced to detect stable regions on veins on the placenta. This method finds regions spanning the two edges of veins on the placenta and defines as keypoint the middle of the vein as shown in Figure 5.3d. This method also includes the orientation of the vein, thus defines a keypoint that is constraint in all dimensions. Furthermore, they can be reliably detected in multiple images and can thus be considered as stable keypoints.

Through correct understanding of the stable keypoint detection task, insight had been gained in what was needed to perform that task effectively. The subsequent adoption of the SSD algorithm, designed for object recognition, is a perfect example of the importance of understanding a task correctly in order to successfully apply deep learning methods to solve that task. The SSD algorithm was only altered in changing the definition of the bounding box to that of a rotated box as shown in Figure 5.4. The rest of the algorithm remained the same to achieve stable keypoint detection. This shows that through correct selection of the network architecture and learning method it is possible to apply deep learning methods to solve many tasks successfully.

10.1.2 Matchable Feature Extraction

The next subtask was to extract matchable features to obtain robust matches. In state-of-the-art methods such as SIFT this is done by defining a grid based histogram of gradients to obtain a descriptive feature. However, in Chapter 3 it has been shown that the visual structure on the placenta is everywhere very similar, resulting in insufficiently distinctive features. Combined with the reduced keypoint detection performance, this results in too few correctly matched keypoints for image registration in the in-vivo setting.

In Chapter 4 contrastive loss has been introduced from the field of object recognition to resolve the challenge of very similar appearance of places on the placenta. The selection of this learning method is inspired by the needs of the matching task, hence the name: Matchable Feature Extraction. The keypoint matching task needs to find keypoints that describe the same location on the placenta in two overlapping images. Therefore, the contrastive loss method has been used to increase the similarity between features describing the same locations, while increasing the difference with all other features. In most deep learning methods the learning method is a form of supervised learning, which is a form of learning by example. Whereas the use of contrastive loss is a form of learning by doing. Learning the difference between correct and incorrect matches can be considered learning by trial and error, with simultaneously learning the incorrect and successful execution of the task.

In Chapter 5 contrastive loss learning has been combined with the SSD algorithm to detect stable keypoints and extract matchable features at the same time. This results in a significant increase in correctly matched keypoints between two overlapping images of the placenta. This solved the image registration subtask of the panorama reconstruction task of placentas for in-vivo laser coagulation therapy, by effectively solving its two major sensing subtasks using deep learning.

10.1.3 Keypoint matching and Transform Estimation

Chapter 5 showed that next to the subtask of detecting stable regions, also the quality of the image and the detected regions can be learned. These qualitative measures are not directly part of the overall task, but do provide an important indication on how well the task can be performed with the current information. Furthermore, it provides valuable information to obtain correct transformations between images, in the form of illumination and view-point information. If this information is provided to the surgeon, such that the illumination as well as the distance to the placenta can be adjusted, then the visibility condition can be improved and a correct image registration can be obtained. The inclusion of the surgeon in the machine's task of panorama reconstruction is generally not considered, as tasks performed by machines and humans are often strictly separated. But our approach will ensure that a more optimal and constant image registration performance can be maintained resulting in an overview of the placenta supporting the surgeon.

10.2 Part 2 - Road User Perception

One of the major tasks in automated driving vehicles on the public road is the perception of other road users as described in Chapter 6. The goal of this task is to recognize other road users and estimate what their (possible) future actions will be. This information is then used in planning the actions of the ego-vehicle. The description of this road user perception task is applicable to how humans do it, as well as how automated vehicles do it and has been described using the Sense-Think-Act paradigm in Figure 6.3. Each of the steps are detailed in the next sections.

10.2.1 Road User Recognition

The road user recognition step consists of two tasks; First, perceiving the environment and detecting the location of objects surrounding the ego-vehicle. Second, classifying these objects as relevant road users, defined as those objects that participate in traffic. That are objects that are either part of, moving on, or are close to the road where the ego-vehicle is driving on.

To achieve this task, various types of sensors and methods can be used as described in Chapter 6.2.1. Through analysis of the object recognition task and the available sensors, an effective combination of sensors and algorithms had been selected in Chapter 7. The road user recognition task first of all consists of the object detection subtask, for which radars are very effective and have shown high performance and reliability. Then, for object classification, the use of visual information has shown great benefits. Therefore, camera information had been selected in combination with a convolutional neural network. This network is trained with contrastive loss to emphasis the differences between relevant road users and irrelevant objects.

Although state-of-the-art deep learning based methods have shown better performance in the years following the research in Chapter 6.2.1, some of the methods in this chapter follow similar patterns of thought. The RCNN method also separates the task of object detection and classification, but these methods still used only camera information for the subtasks. Moreover, methods such as MV3D, that make efficiently use of other types of sensors, also follow the separation of the detection and recognition task. For example, MV3D uses a top-view approach on laser data to extract 3D locations of objects which are then fused with camera information to obtain road user recognition.

The rapid advances in the state-of-the-art methods for road user recognition fuelled the decision to focus on the much less resolved road user prediction subtask of road user perception.

10.2.2 Road User Prediction

In order to plan safe actions humans as well as automated vehicles need to determine the intentions of the other road users. Since the intention is generally not directly observable but determine the actions of those road users, many state-of-the-art methods use the motion information of those road users to infere their intentions. However, the actions of road users are the result of their intentions and constrained by the actions the road user can safely take. Therefore, to effectively predict the future states of other road users it is important to acquire - by perception - the information that drives their considerations and predict their decisions.

As normally a road user is driving on the road, the road geometry is one of the constraints that is considered in determining the possible actions of a road user. Human drivers often take the road geometry into account in deciding their driving actions. For example, a driver decides to follow a road, resulting in steering actions that will keep the vehicle within the boundaries of that road and make a turn when the road makes a curve. Therefore, in Chapter 8 the motion information of a road user has been transformed to the road geometry, such that longitudinal motion is following the road and lateral motion is changing the distance to the boundaries of the road. This transformation encodes the road geometry in the motion of the vehicle and allows the application of state-of-the-art motion prediction methods without alteration. This approach shows that with thorough analysis of a task, as performed by humans, the used information is often more descriptive.

Driving actions are often influenced by other road users, as one wants to avoid collisions. These interactions are governed by traffic rules and drivers will determine their actions in accordance. Through analysis of the road user prediction task, it became apparent that in order to model interaction between road users, their geometric relation had to be maintained. Transforming this motion information to road geometry greatly complicates this. Therefore, in Chapter 9 first the motion information has been encoded and social pooling was used to model the interaction. This showed that information, as required to solve a task, need the right format before it can effectively be used.

In order to incorporate road geometry, first this information has been encoded by an RNN similar to the encoding of the motion information. Next, an attention mechanism was used to select the right section of the encoded road geometry information and subsequently combined with the encoded interaction and motion information. This approach showed that a method, originally designed for natural language processing, can be applied to modelling motion information as well as road geometry. Furthermore, it showed that the use of an attention method, which also found its origin in natural language processing, can be used to obtain relevant information. These two applications illustrated how transfer learning can improve the applicability of deep learning to various types of sequential tasks.

10.3 Conclusion

The goal of this thesis was to illustrate that deep learning methods can be applied effectively in solving various tasks in realistic applications (only) by thorough understanding and analysis of the task at hand.

In part 1 of this thesis, the task of a surgeon to perform laser coagulation surgery on a placenta using fetoscopic images of twins, showed that the thorough analysis of that task considerably reduced the complexity of the technical task of panorama reconstruction of the placenta. The application of traditional state-of-the-art methods has shown limited success for an in-vivo setting. Therefore the panorama reconstruction task was analysed in the aspects where the in-vivo setting forms a challenge for state-of-the-art methods. The analysis was then translated into requirements for each of the subtasks needed to create the panorama. Following these requirements the thesis showed that the success of coping with the challenges was due to thorough understanding of which information was actually required, followed by the effective application of deep learning networks and selected learning methods.

The second part of this thesis focused on improving road user perception for automated driving vehicles. By thorough analysis of available sensors and their data and application of promising learning methods, an effective application of a deep learned neural network was obtained that improved road user recognition performance for automated driving. For road user prediction, classical methods mostly used motion information to predict future trajectories. By analysing how humans perform the driving task, it became apparent that other factors such as road geometry and road user interaction considerably influence their actions. Therefore, these factors have been incorporated in the prediction process using various deep learning methods.

In this thesis in two domains of perception tasks I showed that deep learning can be fruitfully used. However, this can only be achieved by a deep understanding of the task itself and its decomposition into subtasks, the data that is needed in every subtask, the selection and modelling of adequate network architectures and the selection and modelling of the proper learning algorithms. This is not trivial.

A

Appendix

A.1 How do humans learn tasks

To understand how humans can learn a given task, two aspects of this process are described; First, what are tasks and how to structure them and second, what methods are used by humans to learn these tasks.

A.1.1 Tasks

If there is a desired change that requires an action of the human, they can chose to do many different things, which can be described as tasks. For example, in driving a car, a driver wants to go somewhere and to achieve this he has to operate the car. Currently this is done by controlling the car with the steering wheel to change direction and the accelerator and braking pedals to control the speed. Also, the driver observes the environment around the vehicle. With this information the driver continuously makes decisions on what to do. For example, the driver observes a traffic light and notices that it is red. This means that the driver is not allowed to continue. Therefore the driver decides to stop and controls the car to stop in front of the traffic light.

Taking the example above, performing a task can be split into three distinct steps according to the *sense-thing-act* paradigm [16, 87] as detailed below:
Sensing is where the environment is observed and the relevant information is extracted. In the example, a traffic light is observed and the driver extracts the relevant information by noticing it is red.
The *Thinking* step uses the relevant information to make an abstraction into a meaning and a decision is made. For a red traffic light, the driver knows that he cannot continue to drive and should decide to stop.
Acting transforms the decision into one or more actions that effectuates this decision. For example the decision of stopping is done by pressing the brake pedal.

The above described task is to stop in front of the red traffic light. However, this stopping action can also be split into a smaller task; First, the driver observes the vehicle's speed and the traffic light or the stopping line. Second, the speed is abstracted into a distance to stop and compared with the distance to where the driver has to stop. This is translated into a decision of waiting and/or slowing down. Last, the decision is transformed into an action of doing nothing, moving the foot to the brake pedal and/or pressing it.

Just as the above example of the stopping task is part of the task reacting on a traffic light, many tasks can be described as the hierarchical combination of multiple tasks, forming complex tasks. Furthermore, the division of these (complex) tasks into smaller tasks, allows these tasks to be reused as parts of other complex tasks. For example, the subtask of determining if no traffic is approaching when crossing the street while walking, can also be used to determine if it is safe to cross an intersection when driving a car. Therefore, it is important to understand how to structure tasks.

A.1.2 How humans learn

Even though the human brain is a complex machine and is capable of learning many things, humans split complex tasks into manageable parts. These parts are tasks that can again consist of multiple smaller tasks. Learning these tasks is challenging and humans have various ways of learning tasks of different levels of complexity. However, in the context of machine learning the three most relevant learning methods are described here:

Learning by Example is where an example of the task is presented. An example is how a parent teaches a child the word 'apple'. The parent shows an apple to the child and speaks the word 'apple'. The child will look at the apple and will learn the appearance of the object shown (Sensing). At the same time the child will hear the word (Sensing) and learns that the shown object is an 'apple' (Thinking). By repeating the word 'apple' the parent is trying to make the child also say the word 'apple'. After a while, the child will be able to pronounce the word correctly (Acting) and has learned what an apple is and is able to communicate with other humans about this object.

Learning by Doing is a learning approach with a set of approaches to learn a task by performing the task. In the first approach, called *Trial and Error*, part of the Sense-Think-Act steps is known, but not all. For example a child is trying to put a block into the box with different shaped holes (Figure A.1). The child knows it has to get the block into the box and has learned how to handle blocks and how to put a block through a hole (Acting). However, it doesn't understand yet that a specific shaped block only fits through the inverse shaped hole. By trying many different possibilities, it explores the possibilities up to the point it has found the right hole. After being successful putting blocks in the box, the child will try to find the relevant differences between the failing trials and the successful ones. After a while it has learned that the shape is important (Sensing). Next, the child will start to understand the meaning of the shape and its inverse hole, such that it will allow the block to pass (Thinking).

Another approach is *Exploration* which is a learning method that fine-tunes what has already been learned. This can be for all three steps of a task. For example adults that passed their driving exam, have learned the basics of driving. While driving,

Figure A.1: Example of shape sorting toy.

more and more experience is build up, obtaining a better understanding of all the variations of the situations that can be encountered. The driver often will use a form of self-supervised learning, on their own driving performance, trying to improve their driving skills. Such as reacting more efficiently to an unexpected situation to drive more safely (Thinking) or taking a curve more smoothly in order to drive more comfortably (Acting).

Transfer Learning is a learning method where knowledge of one task is used in learning of performing another task. For example recognizing a red light can be used in many different tasks, such as understanding the state of a traffic light, a electronic door lock or if a device is on or off. Another example is when a child learns to recognize a toy car, the same knowledge can be used to learn to recognize real cars and with the help of *Exploration* a person can learn to recognize all kind of cars, and is able to use this when learning on how to drive a car.

With the understanding of tasks and of human learning, only one required aspect is missing in order to transfer learning of complex tasks. The information that is required to learn a task is not easily described, as humans have many different senses, of which the data is unconsciously abstracted. Therefore, it is very difficult to describe exactly what information is used. Furthermore, as humans are very complex systems that have learned many different tasks, it is difficult to segregate what type of experience is used in order to learn a new task.

A.2 Machine Learning

To make life better, humans try to make machines take over tasks, such as driving a car. To make a machine perform a task, they generally need to be programmed to do so. These tasks can be programmed by logic or mathematical constructs, like functions to convert the sensor input to a more meaningful format (Sensing), rules what to do when encountered a certain input (Thinking) and how to control a certain output (Acting).

Also machines are often programmed in the same *Sense-Think-Act* steps to perform tasks. In the example of stopping for a red traffic light, one could define a method that detects traffic lights by defining their shape and trying to match different areas of a camera image to this shape (Sensing). Furthermore, it is a convention that the top light is red and has meaning to stop. Thus, a rule can be defined to check for the color in three areas of the traffic light to extract the state. In the case the traffic light is red, the vehicle has to stop (Thinking). In that case, also the distance to the traffic light is obtained and a deceleration profile is calculated such that it is efficient and comfortable (Thinking). Then this deceleration profile is used by the vehicle controller to slow down the vehicle and stop at the right position (Acting).

Simple tasks are easily programmed, however complex tasks often consist of many different smaller tasks, which require a lot more work. Also, a programmed task is often created for that specific task and thus less easily reused in another complex task. For example, a piece of code recognizing a traffic light, will focus on the fact that the top light is red, middle one yellow and the bottom light green. This piece of code is not easily transferred to a task of recognizing the state of an electronic door lock with only a red and green light, that has no specific position relationship.

In contrast, *machine learning* focusses on making a machine learn a task, instead of programming it. This is achieved by creating a (code) construct that can transform an input to an desired output by learning how to do this. The code of recognizing the state of a traffic light can also be used to learn to recognize the state of an electronic door lock. The only difference is what is contained in the input, namely a picture of a traffic light or that of an electronic door lock. Therefore, the ability to learn a task, allows to generalize a task and to transfer this knowledge to another complex task (Transfer Learning).

A.2.1 Neural Networks

Since the 1980's many researchers have tried to apply machine learning similar to how humans learn tasks by (partially) artificially recreating the human brain. The human brain consists of neurons of which one is shown in Figure A.2. Each neuron receives signals from other neurons through its dendrites. Accordingly to the received inputs the neuron activates and subsequently sends a signal to other neurons through it axon. The human brain is formed by interconnecting about 86 billion of these neurons into a network, allowing it to learn complex tasks.

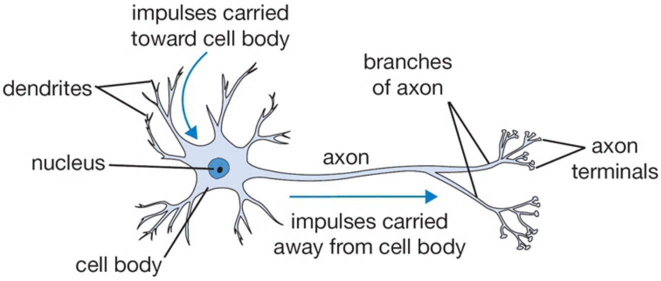

impulses carried toward cell body
dendrites
branches of axon
nucleus
axon
axon terminals
impulses carried away from cell body
cell body

Figure A.2: Description of human neuron (adopted from [137])

These neurons have been artificially recreated by researchers on computers as shown in Figure A.3. Inputs of other neurons x_i are weighted and summed. The neuron activates according to an activation function such as a *sigmoid* or *tanh* function. This output is then shared to other neurons. Connecting multiple of these neurons creates an artificial neural network.

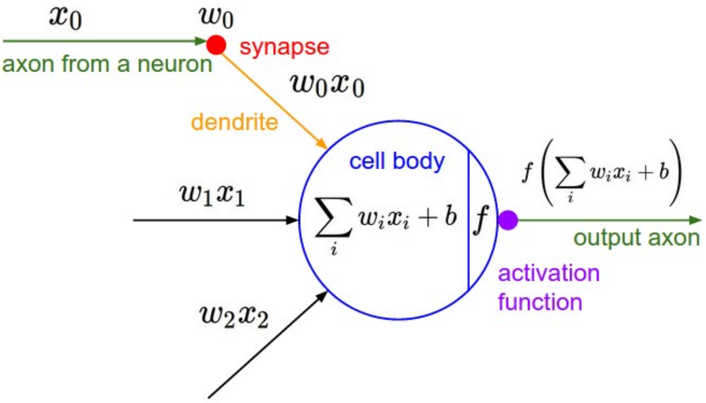


Figure A.3: Artificial model of a neuron (adopted from [137])

A simple example of such a neural network is depicted in Figure A.4. These neurons are often arranged in the form of layers describing the steps from in- to output. The input layer provides inputs to the dendrites of the first layer of neurons. These neurons are then all connected to a second layer of neurons. Finally the output layer is a (single) neuron describing the output of the network. Since the two middle layers are not accessible, they are called hidden layers of the network.

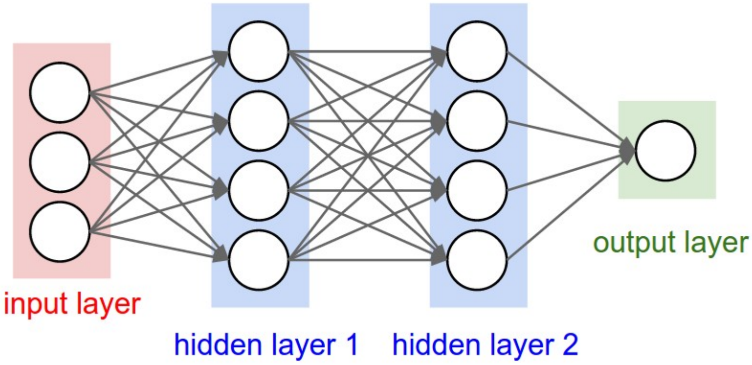


Figure A.4: Artificial Neural Network (adopted from [137])

These networks could learn complex abstractions between input and output and therefore easily learn simple tasks. Though this approach seemed promising, it lacked the ability to be extended to complex tasks. To understand why these type of neural networks lack the ability to learn complex tasks, a better understanding is needed on how those networks work and are trained.

The network described in Figure A.4 has three inputs, two hidden layers of each four nodes and one output layer. This network thus has 32 weights and 9 biases, making the number of parameters to be learned 41. This could be solved with linear algebra or methods such as least squares fitting, though it would require 41 input-output pairs that precisely describe all variations of the input.

Unfortunately, obtaining such an input set is near impossible. Therefore, to find the values of these parameters, it is learned through feedback of an error function. The network is initialized with random parameters. The network is fed an input and calculates an output. The difference between this output and the desired output is calculated as the error of the network. This error is then in the form of a gradient propagated backward through the network, updating the parameters of each neuron. By repeating this process, preferable with multiple iterations of all possible combinations of the input-output pairs, the parameters are learned.

However, such neural networks failed learning complex tasks due to three reasons:
First, for complex tasks, the required abstraction requires multiple hidden layers with many neurons, increasing the number of simulated neurons drastically to the point they could not be simulated any more. Thus, the abstraction capabilities of a neural network are limited.
Second, for complex tasks, the input is larger and contains more data. For example, consider a very small image of $16 \times 16 = 256$ pixels, each with only 10 different possible values. Then there are 10^{256} possible variations of this image, making it infeasible to try to learn all possible variations. Thus, in learning a task, a generalization of the information has to be learned as well.
Last, complex tasks generally consist of multiple smaller tasks. Humans instinctively know how to split this up and what part of the task to change. In a sense humans can correctly update the right parameters in one go, while with neural networks all parameters are updated in small steps. Thus, the learning process is inaccurate and slow.

Therefore, interest in neural networks was limited to learning more simple tasks, for which the amount and variation of required information was limited.

A.2.2 Deep Learning

However, recently the scientific community has revisited the concept of neural networks with deep-learning that allows to create more complex (deeper) neural networks. This was instantiated by a few advances, of which four stand out;
First, the internet and the availability of data rose to the concept of big data, giving access to much more data to learn from. An example is the large number of images that is available in the ImageNet Challenge (ILSVRC) [104]
Second, with advances in GPU's the available computing power increased drastically and allowed to create much more complex network designs.
Third, the introduction of more complex learning methods such as batch learning and optimized descent methods, increased the learning performance and speed.
Last, the ease of creating and using neural networks. This was fuelled by a few good deep learning frameworks such as Caffe [54] and Tensorflow [1].

A good example of where the introduction of deep-learning found success is the ImageNet Challenge where images have to be classified and objects localized. This challenge contains a previously unseen large amount of data with 10 million images labelled into 1000 different object classes. Deep learned Convolutional Neural Networks such as AlexNet [62], OverFeat [112], VGG [117], Inception [52, 124, 126, 127] and ResNet [48] showed great successes. These networks all have shown a drastic increase in the number of hidden layers. This increase can be explained due to three advances; First, the use convolutional layers, that contain many times less neurons compared to fully connected layers. Furthermore, through the advances in learning methods the weights remain stable even while increasing the number of hidden layers. Lastly, also the huge increased in computational power made this possible.

This increase in layers allowed to learn previously unseen visual complexity, resulting in networks learning visual patterns such as edges, corners, but also shapes, that are spatial independent, which was not previously possible in traditional neural networks. Which brings the last major advancement, that of *Transfer Learning*, where the weights and neural network structure trained on e.g. ImageNet data is used to train on a different set of data and even on data from a complete different domain, such as detection of cancer in medical images. This allows a huge increase in development speed of neural networks.

A.2.3 Human Learning in Deep Learning

Human learning methods are used in deep learning as well. With the example of how *AlphaGo* learned to play, it can be shown how the human learning methods are used to train neural networks similar to how humans learn.

Learning by Example is one of the most used learning methods in machine learning, referred to as supervised learning, where the input and desired output are provided. In AlphaGo this approach is used to initialize the policy network by learning from moves by expert human players. The input of the network is defined as the current state of the board and the output is a softmax probability distribution over all legal moves in each state. In a sense the system learns which moves are allowed and which moves are good according to games previously played by experts .

The supervised learned policy network is fine-tuned through games of self-play, thus *Learning by Doing*. This achieves that the policy network learns policies; combination of positions and moves, that win the game rather then mimicking human moves through *Exploration*. Similarly to how humans improve their playing and find new tactics by gaining experience through playing games against themselves.

Finally a value network is created, providing a success value for a given state. This network is trained by exploring the input space, all the possible positions on the board, and feeding back the outcome of this game. Through training enough state-outcome pairs, the value of a state is learned, resulting in learning the meaning with respect to the possible action, such as in learning through *Trial and Error*.

As described in the previous section, the ability to reuse previously learned neural networks in learning different unrelated tasks covers the last component of *Transfer Learning* in the human learning methods. With these learning method humans can learn any task, which can be reproduced in deep learning. Therefore, the next section describes what is needed to for deep learning to achieve the same.

A.2.4 Requirements for Deep Learning

With the understanding of how humans structure tasks and learn these, combined with an understanding of how to train neural networks through deep learning, it should be possible to better describe the requirements to transfer human learning to machine learning of tasks.

The input *data* of a neural network is the source of everything, as it contains the relevant data from which the meaning for the decision making has to be extracted. In selecting the data, some things have to be considered;
First, the format of the data should represent the relevant information clear enough that it can be extracted. For example, learning to recognize different fruits requires color information above every other feature, as the difference between round fruits like apples, oranges, peaches etc. is mostly determined by the color information.
Second, The variations of the relevant information should be represented well. For example, apples have different shapes as well different colors. These should be contained within the input data.
Third, for the amount of data a balance has to be found. An increase in data requires longer training. But also the balance within the variations of the data should be found, a neural network will not be able to learn the difference between apples and oranges if it is fed 1000 images of apples and only 10 images of oranges. Furthermore, these 1000 images of apples, probably contain redundant images of the variations that can be encountered.
Fourth, defining what is irrelevant information is just as important as defining what is relevant, as not all possible combinations of the input data can be used for training. For example, learning the color variations of red, yellow and green apples, compared to oranges, will probably result in the network learning that all non-orange colored objects are apples. But this invertedly also includes blue apples, so training the network with a blue object as 'not-apples' would prevent this.
Last, also the desired output (labels) of the input data, should be considered. In the example of different coloured apples, it could be beneficial to define different labels for red, yellow and green apples, allowing to learn what type of apples are there and exclude the case of 'blue apples'. Therefore, the data used has to be carefully chosen and can be transformed to generate more samples and small variations.

The *architecture* of a network is of great importance as well. On one hand the network needs a certain level of complexity to enable the network to abstract the relevant information and transform this to the desired output. While, on the other

hand, the network should not be too complex, such that the parameters in the network cannot be learned from the available data. Also, too many parameters will make the network overfit on the data, by learning the fine difference of each training sample and loose the ability to generalize over the variations within the data.
With the rise of deep learning, the convolutional layer was (re)introduced. This layer moves a kernel with only a few parameters over the image. By combining multiple kernels, the convolutional layer learns to recognize patterns in the input data. This not only reduces the number of parameters but also can recognize these patterns independent on their location in the data, allowing for a strong generalization over the data. Then in turn allows to create structures with convolutional layers such as Inception [126] and ResNet [48] architectures, introducing the ability to abstract even more complex tasks and find correlations in these abstractions.
Furthermore, the introduction of Recurrent Neural Networks (RNN) [76] enables a neural network of reusing abstractions from previous iterations to influence the current abstractions. The outcome of a previous abstraction is combined with the current abstraction. This is then further abstracted and shared to the next iteration. This allows the network to remember the state. As such memory can be formed, like Long Short Term Memory (LSTM) [37] which can remember recent information through passing and has learned how to extract the relevance (long term memory). Therefore, it is key to choose the right architecture and complexity to be able to learn the given task.

Also, the *learning method* is of great importance, as it updates the parameters of the network, such that it can abstract the data into the desired output.
For example, batch learning uses multiple inputs simultaneously and calculates a more generalized and stable gradient. This makes the gradient update of the network less susceptible to small (unwanted) variations in the input data. And reduces the chance of over-fitting and getting stuck in local minima of the parametrization space.
Furthermore, the learning method in the form of the loss or error function should be considered carefully. Generally the error is calculated between the networks and the desired output and an update is obtained based on this difference. Which reflects the human learning method of *Learning by Example* which is often revered to as *supervised learning* in machine learning.
There are also different approaches on how to obtain the updates of the network. For example, in contrastive loss [45] the error is defined between two outputs of the network. For example, two inputs of apples are fed to the network, the output of the network should be the same, namely 'apples'. Thus, the gradient is defined such that the difference is minimized. But for an input of an apple and an orange, the output should be different, making the gradient maximizing this difference. Redefining the

learned task in the form of (dis)similarity, can improve the abstraction that is learned. Which is very similar to humans learning from mistakes, whom will define a strong learning update to not repeat the same mistake like in *Trial and Error*.
In the previous approaches the desired output is defined as a discrete label. Although this is one of the easiest types of describing desired outputs, it is certainly not the only form. Other approaches define other forms of outputs such as image segmentation in SegNet [5] where every value in the input is classified. Or, the translation of a sentence from one to another language as done by sequence-to-sequence networks in Google translate [122]. Also, obtaining the presence and location of an object as used in the Single Shot Detector (SSD) [74]. Therefore, defining different type of outputs, will allow for different type of tasks to be learned as these are directly related.
Lastly, other human learning methods are represented in machine learning as well. In *semi-supervised learning*, the task is described and defined such that no desired output has to be defined with the input data. Often, another method will approximate the desired output. Another approach in machine learning is *reinforcement learning* where a scoring or cost function is defined to describe the success of a task [18, 105, 123]. These approaches reflect the human learning methods of *trial-and-error* and *learning-by-doing*

The three key aspects that have to be considered in deep learning a complex tasks; the task, the data and the learning construct have been described together with methods that can be used. However, there is no clear guideline what methods can be used to learn a specific complex task. As well as when a chosen setup for deep learning a complex task will be successful. Effectively, the task of learning a machine a specific task is still not understood in detail by humans. Therefore, for researchers deep learning is often a process of *trial-and-error* of different methods and varying setups and finally choosing the best performing one.

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). *"Tensorflow: Large-scale machine learning on heterogeneous systems"*. URL http://tensorflow.org/. Software available from tensorflow.org.

[2] Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., and Savarese, S. (2016). *"Social LSTM: Human Trajectory Prediction in Crowded Spaces"*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 961–971.

[3] Assembly, U.G. (2015). *"Transforming our world: the 2030 agenda for sustainable development"*. URL https://www.refworld.org/docid/57b6e3e44.html.

[4] Automotive Engineers SAE, S. of (2018). *"Levels of automation j3016"*. URL https://www.sae.org/standards/content/j3016_201806/.

[5] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). *"Segnet: A deep convolutional encoder-decoder architecture for image segmentation"*. IEEE transactions on pattern analysis and machine intelligence, 39(12), pp. 2481–2495.

[6] Bahdanau, D., Cho, K., and Bengio, Y. (2014). *"Neural machine translation by jointly learning to align and translate"*. arXiv preprint arXiv:1409.0473.

[7] Barber, D. (2004). *"A stable switching kalman smoother"*.

[8] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). *"Speeded-up robust features (surf)"*. Computer vision and image understanding, 110(3), pp. 346–359.

[9] Behrens, A., Stehle, T., Gross, S., and Aach, T. (2009). *"Local and global panoramic imaging for fluorescence bladder endoscopy"*. 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 6990–6993.

[10] Belgiovane, D. and Chen, C.C. (2016). *"Bicycles and human riders backscat-*

tering at 77 ghz for automotive radar”. 2016 10th European Conference on Antennas and Propagation (EuCAP), pp. 1–5.

[11] Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. (2015). “*Scheduled sampling for sequence prediction with recurrent neural networks*”. Advances in Neural Information Processing Systems, pp. 1171–1179.

[12] Bishop, C.M. (2006). “*Pattern recognition and machine learning*”. springer.

[13] Blackman, S. and Popoli, R. (1999). “*Design and analysis of modern tracking systems(book)*”. Norwood, MA: Artech House, 1999.

[14] Brand, M., Oliver, N., and Pentland, A. (1997). “*Coupled hidden Markov models for complex action recognition*”. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1997), pp. 994–999.

[15] Bromley, J., Bentz, J.W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Säckinger, E., and Shah, R. (1993). “*Signature verification using a âĂIJsiameseâĂİ time delay neural network*”. International Journal of Pattern Recognition and Artificial Intelligence, 7(04), pp. 669–688.

[16] Brooks, R.A. et al. (1991). “*Intelligence without reason*”. Artificial intelligence: critical concepts, 3, pp. 107–63.

[17] Brown, M. and Lowe, D.G. (2007). “*Automatic panoramic image stitching using invariant features*”. International journal of computer vision, 74(1), pp. 59–73.

[18] Busoniu, L., Babuska, R., De Schutter, B., and Ernst, D. (2010). “*Reinforcement learning and dynamic programming using function approximators*”. CRC press.

[19] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). “*Brief: Binary robust independent elementary features*”. Computer Vision–ECCV 2010, pp. 778–792.

[20] Campbell, M., Hoane, Jr., A.J., and Hsu, F.h. (2002). “*Deep blue*”. Artif. Intell., 134(1-2), pp. 57–83.

[21] Carroll, R.E. and Seitz, S.M. (2009). “*Rectified surface mosaics*”. International journal of computer vision, 85(3), pp. 307–315.

[22] Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., and Urtasun, R. (2016). “*Monocular 3d object detection for autonomous driving*”. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2147–

2156.

[23] Chen, X., Kundu, K., Zhu, Y., Berneshawi, A.G., Ma, H., Fidler, S., and Urtasun, R. (2015). *"3d object proposals for accurate object class detection"*. Advances in Neural Information Processing Systems, pp. 424–432.

[24] Chen, X., Ma, H., Wan, J., Li, B., and Xia, T. (2017). *"Multi-view 3d object detection network for autonomous driving"*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1907–1915.

[25] Chmait, R.H., Kontopoulos, E.V., Korst, L.M., Llanes, A., Petisco, I., and Quintero, R.A. (2011). *"Stage-based outcomes of 682 consecutive cases of twin–twin transfusion syndrome treated with laser surgery: the usfetus experience"*. American journal of obstetrics and gynecology, 204(5), pp. 393–e1.

[26] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). *"Learning phrase representations using rnn encoder-decoder for statistical machine translation"*. arXiv preprint arXiv:1406.1078.

[27] Crombleholme, T.M., Shera, D., Lee, H., Johnson, M., DâĂŹAlton, M., Porter, F., Chyu, J., Silver, R., Abuhamad, A., Saade, G., et al. (2007). *"A prospective, randomized, multicenter trial of amnioreduction vs selective fetoscopic laser photocoagulation for the treatment of severe twin-twin transfusion syndrome"*. American Journal of Obstetrics & Gynecology, 197(4), pp. 396–e1.

[28] Deo, N. and Trivedi, M.M. (2018). *"Convolutional Social Pooling for Vehicle Trajectory Prediction"*.

[29] Dong, C., Dolan, J.M., and Litkouhi, B. (2017). *"Intention estimation for ramp merging control in autonomous driving"*. Intelligent Vehicles Symposium (IV), 2017 IEEE, pp. 1584–1589.

[30] Enzweiler, M. and Gavrila, D.M. (2009). *"Monocular pedestrian detection: Survey and experiments"*. IEEE transactions on pattern analysis and machine intelligence, 31(12), pp. 2179–2195.

[31] Fischler, M.A. and Bolles, R.C. (1981). *"Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography"*. Communications of the ACM, 24(6), pp. 381–395.

[32] Flohr, F., Dumitru-Guzu, M., Kooij, J.F., and Gavrila, D.M. (2015). *"A probabilistic framework for joint pedestrian head and body orientation estimation"*.

IEEE Transactions on Intelligent Transportation Systems, 16(4), pp. 1872–1882.

[33] Gaisser, F., Jonker, P.P., and Chiba, T. (2016). *"Image registration for placenta reconstruction"*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 33–40.

[34] Gaisser, F., Peeters, S.H., Lenseigne, B., Jonker, P.P., and Oepkes, D. (2017). *"Fetoscopic panorama reconstruction: Moving from ex-vivo to in-vivo"*. Annual Conference on Medical Image Understanding and Analysis, pp. 581–593.

[35] Galceran, E., Cunningham, A.G., Eustice, R.M., and Olson, E. (2017). *"Multi-policy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment"*. Autonomous Robots, pp. 1–16.

[36] Geiger, A., Lenz, P., and Urtasun, R. (2012). *"Are we ready for autonomous driving? the kitti vision benchmark suite"*. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 3354–3361.

[37] Gers, F.A., Schmidhuber, J., and Cummins, F. (1999). *"Learning to forget: Continual prediction with lstm"*.

[38] Gindele, T., Brechtel, S., and Dillmann, R. (2010). *"A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments"*. Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, pp. 1625–1631.

[39] Gindele, T., Brechtel, S., and Dillmann, R. (2013). *"Learning context sensitive behavior models from observations for predicting traffic situations"*. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, Itsc, pp. 1764–1771.

[40] Girshick, R. (2015). *"Fast r-cnn"*. Proceedings of the IEEE international conference on computer vision, pp. 1440–1448.

[41] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). *"Rich feature hierarchies for accurate object detection and semantic segmentation"*. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587.

[42] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *"Deep learning"*. URL http://www.deeplearningbook.org. Book in preparation for MIT Press.

[43] Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., and Alahi, A. (2018). *"Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks"*.

[44] Hack, K., Derks, J., Elias, S., Franx, A., Roos, E., Voerman, S., Bode, C., Koopman-Esseboom, C., and Visser, G. (2008). *"Increased perinatal mortality and morbidity in monochorionic versus dichorionic twin pregnancies: clinical implications of a large dutch cohort study"*. BJOG: An International Journal of Obstetrics & Gynaecology, 115(1), pp. 58–67.

[45] Hadsell, R., Chopra, S., and LeCun, Y. (2006). *"Dimensionality reduction by learning an invariant mapping"*. Computer vision and pattern recognition, 2006 IEEE computer society conference on, volume 2, pp. 1735–1742.

[46] Harris, C. and Stephens, M. (1988). *"A combined corner and edge detector."* Alvey vision conference, volume 15, p. 50.

[47] Hartley, R. and Zisserman, A. (2003). *"Multiple view geometry in computer vision"*. Cambridge university press.

[48] He, K., Zhang, X., Ren, S., and Sun, J. (2016). *"Deep residual learning for image recognition"*. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

[49] Hecher, K., Diehl, W., Zikulnig, L., Vetter, M., and Hackelöer, B.J. (2000). *"Endoscopic laser coagulation of placental anastomoses in 200 pregnancies with severe mid-trimester twin-to-twin transfusion syndrome"*. European Journal of Obstetrics & Gynecology and Reproductive Biology, 92(1), pp. 135–139.

[50] Hochreiter, S. and Schmidhuber, J. (1997). *"Long short-term memory"*. Neural computation, 9(8), pp. 1735–1780.

[51] Institute, T.A.T. (2019). *"Urban mobility report"*. URL https://static.tti.tamu.edu/tti.tamu.edu/documents/mobility-report-2019.pdf.

[52] Ioffe, S. and Szegedy, C. (2015). *"Batch normalization: Accelerating deep network training by reducing internal covariate shift"*. arXiv preprint arXiv:1502.03167.

[53] Itseez (2015). *"Open source computer vision library"*. URL https://github.com/itseez/opencv.

[54] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). *"Caffe: Convolutional architecture for fast feature embedding"*. Proceedings of the ACM International Conference on Multimedia, MM '14, pp. 675–678. URL http://doi.acm.org/10.1145/2647868.2654889.

[55] Jo, K., Lee, M., Kim, J., and Sunwoo, M. (2017). *"Tracking and behavior*

reasoning of moving vehicles based on roadway geometry constraints". IEEE Transactions on Intelligent Transportation Systems, 18(2), pp. 460–476.

[56] Kato, T., Ninomiya, Y., and Masaki, I. (2002). *"An obstacle detection method by fusion of radar and motion stereo"*. IEEE Transactions on Intelligent Transportation Systems, 3(3), pp. 182–188.

[57] Kim, B., Kang, C.M., Lee, S.H., Chae, H., Kim, J., Chung, C.C., and Choi, J.W. (2017). *"Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network"*.

[58] Kingma, D. and Ba, J. (2014). *"Adam: A method for stochastic optimization"*. arXiv preprint arXiv:1412.6980.

[59] Klingelschmitt, S. and Eggert, J. (2015). *"Using Context Information and Probabilistic Classification for Making Extended Long-Term Trajectory Predictions"*.

[60] Klingelschmitt, S., Platho, M., Groß, H.M., Willert, V., and Eggert, J. (2014). *"Combining behavior and situation information for reliably estimating multiple intentions"*. Intelligent Vehicles Symposium Proceedings, 2014 IEEE, pp. 388–393.

[61] Kooij, J.F.P., Schneider, N., Flohr, F., and Gavrila, D.M. (2014). *"Context-based pedestrian path prediction"*. European Conference on Computer Vision, pp. 618–633.

[62] Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012). *"Imagenet classification with deep convolutional neural networks"*. Advances in neural information processing systems, pp. 1097–1105.

[63] Ku, J., Mozifian, M., Lee, J., Harakeh, A., and Waslander, S.L. (2018). *"Joint 3d proposal generation and object detection from view aggregation"*. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1–8.

[64] LeCun, Y., Bengio, Y., and Hinton, G. (2015). *"Deep learning"*. Nature, 521(7553), pp. 436–444.

[65] Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H.S., and Chandraker, M. (2017). *"DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents"*.

[66] Lefèvre, S., Vasquez, D., and Laugier, C. (2014). *"A survey on motion prediction and risk assessment for intelligent vehicles"*. Robomech Journal, 1(1),

p. 1.

[67] Lewi, L., Deprest, J., and Hecher, K. (2013). *"The vascular anastomoses in monochorionic twin pregnancies and their clinical consequences"*. American journal of obstetrics and gynecology, 208(1), pp. 19–30.

[68] Lewi, L., Jani, J., Blickstein, I., Huber, A., Gucciardo, L., Van Mieghem, T., Doné, E., Boes, A.S., Hecher, K., Gratacós, E., et al. (2008). *"The outcome of monochorionic diamniotic twin gestations in the era of invasive fetal therapy: a prospective cohort study"*. American Journal of Obstetrics & Gynecology, 199(5), pp. 514–e1.

[69] Li, Y., Shum, H.Y., Tang, C.K., and Szeliski, R. (2004). *"Stereo reconstruction from multiperspective panoramas"*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(1), pp. 45–62.

[70] Liang, M., Yang, B., Chen, Y., Hu, R., and Urtasun, R. (2019). *"Multi-task multi-sensor fusion for 3d object detection"*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7345–7353.

[71] Liang, M., Yang, B., Wang, S., and Urtasun, R. (2018). *"Deep continuous fusion for multi-sensor 3d object detection"*. Proceedings of the European Conference on Computer Vision (ECCV), pp. 641–656.

[72] Liao, H., Tsuzuki, M., Kobayashi, E., Dohi, T., Chiba, T., Mochizuki, T., and Sakuma, I. (2008). *"Fast image mapping of endoscopic image mosaics with three-dimensional ultrasound image for intrauterine treatment of twin-to-twin transfusion syndrome"*. Medical Imaging and Augmented Reality, pp. 329–338.

[73] Liebner, M., Baumann, M., Klanner, F., and Stiller, C. (2012). *"Driver intent inference at urban intersections using the intelligent driver model"*. 2012 IEEE Intelligent Vehicles Symposium, pp. 1162–1167.

[74] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., and Berg, A.C. (2016). *"Ssd: Single shot multibox detector"*. European conference on computer vision, pp. 21–37.

[75] Lowe, D.G. (2004). *"Distinctive image features from scale-invariant keypoints"*. International journal of computer vision, 60(2), pp. 91–110.

[76] Medsker, L. and Jain, L. (2001). *"Recurrent neural networks"*. Design and Applications, 5.

[77] Michalski, R.S. (1986). *"Understanding the nature of learning: Issues and research directions"*. Machine learning: An artificial intelligence approach,

2(1), pp. 3–25.

[78] Middeldorp, J.M., Sueters, M., Lopriore, E., Klumper, F.J., Oepkes, D., Devlieger, R., Kanhai, H.H., and Vandenbussche, F.P. (2007). *"Fetoscopic laser surgery in 100 pregnancies with severe twin-to-twin transfusion syndrome in the netherlands"*. Fetal diagnosis and therapy, 22(3), pp. 190–194.

[79] Milch, S. and Behrens, M. (2001). *"Pedestrian detection with radar and computer vision"*.

[80] Murphy, K.P. (1998). *"Switching kalman filters 1 introduction"*. Dynamical Systems, 1(August), pp. 1–16.

[81] Oliveira, L. and Nunes, U. (2013). *"Pedestrian detection based on lidar-driven sliding window and relational parts-based detection"*. Intelligent Vehicles Symposium (IV), 2013 IEEE, pp. 328–333.

[82] Park, S., Kim, B., Kang, C.M., Chung, C.C., and Choi, J.W. (2018). *"Sequence-to-Sequence Prediction of Vehicle Trajectory via LSTM Encoder-Decoder Architecture"*.

[83] Peeters, S. (2015). *"Training and teaching fetoscopic laser therapy: assessment of a high fidelity simulator based curriculum"*. Ph.D. thesis, Leiden University Medical Center.

[84] Peeters, S. et al. (2015). *"Simulator training in fetoscopic laser surgery for twin–twin transfusion syndrome: a pilot randomized controlled trial"*. Ultrasound in Obstetrics & Gynecology, 46(3), pp. 319–326.

[85] Peter J. Rousseeuw (1984). *"Least median of squares regression"*. Journal of the American Statistical Association, 79(388), pp. 871, , 880.

[86] Petrich, D., Dang, T., Kasper, D., Breuel, G., and Stiller, C. (2013). *"Map-based long term motion prediction for vehicles in traffic environments"*. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, (Itsc), pp. 2166–2172.

[87] Pfeifer, R. and Scheier, C. (2001). *"Understanding intelligence"*. MIT press.

[88] Pool, E.A., Kooij, J.F., and Gavrila, D.M. (2017). *"Using road topology to improve cyclist path prediction"*. Intelligent Vehicles Symposium (IV), 2017 IEEE, pp. 289–296.

[89] Pratt, L. and Jennings, B. (1996). *"A survey of transfer between connectionist networks"*. Connection Science, 8(2), pp. 163–184.

[90] Pratt, L.Y. (1993). *"Discriminability-based transfer between neural networks"*. Advances in neural information processing systems, pp. 204–211.

[91] Premebida, C., Carreira, J., Batista, J., and Nunes, U. (2014). *"Pedestrian detection combining rgb and dense lidar data"*. 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4112–4117.

[92] Premebida, C. and Nunes, U.J.C. (2013). *"Fusing lidar, camera and semantic information: A context-based approach for pedestrian detection"*. The International Journal of Robotics Research, p. 0278364912470012.

[93] Qi, C.R., Liu, W., Wu, C., Su, H., and Guibas, L.J. (2018). *"Frustum pointnets for 3d object detection from rgb-d data"*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 918–927.

[94] Quintero, R.A., Ishii, K., Chmait, R.H., Bornick, P.W., Allen, M.H., and Kontopoulos, E.V. (2007). *"Sequential selective laser photocoagulation of communicating vessels in twin–twin transfusion syndrome"*. The Journal of Maternal-Fetal & Neonatal Medicine, 20(10), pp. 763–768.

[95] Raipuria, G. (2017). *"Situational awareness in intelligent vehicles"*. Master's thesis, Delft University of Technology - Mechanical Engineering.

[96] Raipuria, G. (2017). *"Vehicle trajectory prediction using road structure"*. Master's thesis, Delft University of Technology - Mechanical Engineering. URL http://resolver.tudelft.nl/uuid:6cae1b47-f44e-4b74-8bfd-9098ce843e68.

[97] Raipuria, G., Gaisser, F., and Jonker, P.P. (2018). *"Road infrastructure indicators for trajectory prediction"*. 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 537–543.

[98] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). *"You only look once: Unified, real-time object detection"*. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788.

[99] Reeff, M., Gerhard, F., Cattin, P.C., and Székely, G. (2006). *"Mosaicing of endoscopic placenta images"*. Informatik fÃijr Menschen, volume 1.

[100] Ren, S., He, K., Girshick, R., and Sun, J. (2015). *"Faster r-cnn: Towards real-time object detection with region proposal networks"*. Advances in neural information processing systems, pp. 91–99.

[101] Roberts, D., Neilson, J.P., Kilby, M., and Gates, S. (2008). *"Interventions for the treatment of twin-twin transfusion syndrome"*. Cochrane Database Syst Rev, 1.

[102] Roberts, J.W. (2002). *"Beyond learning by doing: The brain compatible approach"*. Journal of Experiential Education, 25(2), pp. 281–285.

[103] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). *"Orb: An efficient alternative to sift or surf"*. Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 2564–2571.

[104] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., and Fei-Fei, L. (2015). *"ImageNet Large Scale Visual Recognition Challenge"*. International Journal of Computer Vision (IJCV), 115(3), pp. 211–252.

[105] Russell, S.J. and Norvig, P. (2016). *"Artificial intelligence: a modern approach"*. Malaysia; Pearson Education Limited,.

[106] Sadeghian, A., Legros, F., Voisin, M., Vesel, R., Alahi, A., and Savarese, S. (2017). *"CAR-Net: Clairvoyant Attentive Recurrent Network"*.

[107] Schlosser, J., Chow, C.K., and Kira, Z. (2016). *"Fusing lidar and images for pedestrian detection using convolutional neural networks"*. 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 2198–2205.

[108] Schneider, N. and Gavrila, D.M. (2013). *"Pedestrian path prediction with recursive bayesian filters: A comparative study"*. German Conference on Pattern Recognition, pp. 174–183.

[109] Schubert, R., Richter, E., and Wanielik, G. (2008). *"Comparison and evaluation of advanced motion models for vehicle tracking"*. 2008 11th international conference on information fusion, pp. 1–6.

[110] Schwarting, W., Alonso-mora, J., and Rus, D. (2018). *"Survey on Planning and Decision-Making for Autonomous Vehicles"*. (January), pp. 1–26.

[111] Senat, M.V., Deprest, J., Boulvain, M., Paupe, A., Winer, N., and Ville, Y. (2004). *"Endoscopic laser surgery versus serial amnioreduction for severe twin-to-twin transfusion syndrome"*. New England Journal of Medicine, 351(2), pp. 136–144.

[112] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2013). *"Overfeat: Integrated recognition, localization and detection using convolutional networks"*. arXiv preprint arXiv:1312.6229.

[113] Seshamani, S., Lau, W., and Hager, G. (2006). *"Real-time endoscopic mosaicking"*. Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006, pp. 355–363.

[114] Shannon, C.E. (1950). *"Xxii. programming a computer for playing chess"*. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 41(314), pp. 256–275.

[115] Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). *"Mastering the game of go with deep neural networks and tree search"*. nature, 529(7587), pp. 484–489.

[116] Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P., and Moreno-Noguer, F. (2015). *"Discriminative learning of deep convolutional feature point descriptors"*. Proceedings of the IEEE International Conference on Computer Vision, pp. 118–126.

[117] Simonyan, K. and Zisserman, A. (2014). *"Very deep convolutional networks for large-scale image recognition"*. CoRR, abs/1409.1556.

[118] Slaghekke, F., Lewi, L., Middeldorp, J.M., Weingertner, A.S., Klumper, F.J., Dekoninck, P., Devlieger, R., Lanna, M.M., Deprest, J., Favre, R., et al. (2014). *"Residual anastomoses in twin-twin transfusion syndrome after laser: the solomon randomized trial"*. American Journal of Obstetrics & Gynecology, 211(3), pp. 285–e1.

[119] Slaghekke, F., Lopriore, E., Lewi, L., Middeldorp, J.M., Zwet, E.W. van, Weingertner, A.S., Klumper, F.J., DeKoninck, P., Devlieger, R., Kilby, M.D., et al. (2014). *"Fetoscopic laser coagulation of the vascular equator versus selective coagulation for twin-to-twin transfusion syndrome: an open-label randomised controlled trial"*. The Lancet, 383(9935), pp. 2144–2151.

[120] Soper, T.D., Porter, M.P., and Seibel, E.J. (2012). *"Surface mosaics of the bladder reconstructed from endoscopic video for automated surveillance"*. Biomedical Engineering, IEEE Transactions on, 59(6), pp. 1670–1680.

[121] Streubel, T. and Hoffmann, K.H. (2014). *"Prediction of driver intended path at intersections"*. Intelligent Vehicles Symposium Proceedings, 2014 IEEE, pp. 134–139.

[122] Sutskever, I., Vinyals, O., and Le, Q.V. (2014). *"Sequence to sequence learning with neural networks"*. Advances in neural information processing systems, pp. 3104–3112.

[123] Sutton, R.S., Barto, A.G., et al. (1998). *"Reinforcement learning: An introduction"*. MIT press.

[124] Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A.A. (2017). *"Inception-v4, inception-resnet and the impact of residual connections on learning."* AAAI, volume 4, p. 12.

[125] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). *"Going deeper with convolutions"*. arXiv preprint arXiv:1409.4842.

[126] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). *"Going deeper with convolutions"*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.

[127] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). *"Rethinking the inception architecture for computer vision"*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826.

[128] Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). *"Deepface: Closing the gap to human-level performance in face verification"*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701–1708.

[129] Tao, J. and Klette, R. (2012). *"Tracking of 2d or 3d irregular movement by a family of unscented kalman filters"*. Journal of information and communication convergence engineering, 10(3), pp. 307–314.

[130] Tay, C. (2009). *"Analysis of dynamic scenes: application to driving assistance"*. Theses, Institut National Polytechnique de Grenoble-INPG.

[131] Tella-Amo, M., Daga, P., Chadebecq, F., Thompson, S., Shakir, D.I., Dwyer, G., Wimalasundera, R., Deprest, J., Stoyanov, D., Vercauteren, T., et al. (2016). *"A combined em and visual tracking probabilistic model for robust mosaicking: Application to fetoscopy"*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 84–92.

[132] Thorpe, W.H. et al. (1979). *"Origins and rise of ethology"*. Heinemann Educational Books.

[133] Treat, J.R., Tumbas, N., McDonald, S., Shinar, D., Hume, R.D., Mayer, R., Stansifer, R., and Castellan, N. (1979). *"Tri-level study of the causes of traffic accidents: final report. executive summary."*

[134] Tromp, J. and Farnebäck, G. (2006). *"Combinatorics of go"*. International

Conference on Computers and Games, pp. 84–99.

[135] Valsky, D.V., Eixarch, E., Martinez-Crespo, J.M., Acosta, E.R., Lewi, L., Deprest, J., and Gratacós, E. (2012). *"Fetoscopic laser surgery for twin-to-twin transfusion syndrome after 26 weeks of gestation"*. Fetal diagnosis and therapy, 31(1), pp. 30–34.

[136] Verdie, Y., Yi, K., Fua, P., and Lepetit, V. (2015). *"Tilde: a temporally invariant learned detector"*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5279–5288.

[137] Weisstein, E.W. (2017). *"Cs231n: Convolutional neural networks for visual recognition."* URL http://cs231n.github.io/.

[138] Weisstein, E.W. (2017). *""Curvature" from mathworld–a wolfram web resource"*. URL http://mathworld.wolfram.com/Curvature.html.

[139] (WHO), W.H.O. (2005). *"The world health report - make every mother and child count"*. URL https://www.who.int/whr/2005/chapter3/en/index3.html.

[140] (WHO), W.H.O. (2018). *"Global status report on road safety 2018"*. URL https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/.

[141] Wohlhart, P. and Lepetit, V. (2015). *"Learning descriptors for object recognition and 3d pose estimation"*. arXiv preprint arXiv:1502.05908.

[142] Yamada, N., Tanaka, Y., and Nishikawa, K. (2005). *"Radar cross section for pedestrian in 76ghz band"*. 2005 European Microwave Conference, volume 2, pp. 4–pp.

[143] Yi, K.M., Trulls, E., Lepetit, V., and Fua, P. (2016). *"Lift: Learned invariant feature transform"*. European Conference on Computer Vision, pp. 467–483.

[144] Zitova, B. and Flusser, J. (2003). *"Image registration methods: a survey"*. Image and vision computing, 21(11), pp. 977–1000.

Acknowledgements

The road to obtain a PhD is not merely an academic process but one where one grows in many facets. This road intertwines and touches the life of many other people. Therefore, I'd like to express my gratitude to all those people:

Firstly, I'd like to thank my promotor prof. Pieter Jonker for his unrelenting support in reviewing and refining my publications and thesis over the years. Thank you for giving me the freedom and autonomy in finding my way and giving guidance when needed most. You balanced out this autodidact with stacks of books, theses and publications, while giving me the support to grow into an independent researcher and insightful engineer. Your warm and friendly nature makes people listening to you, open their minds and grow accordingly, come with ease. I hope to continue to learn from you and follow in your foot steps in becoming a great man in so may ways.
Also, I would like to express my gratitude to prof. Jenny Dankelman for her time and effort in reviewing and refining my thesis. Even though our interactions were limited over the years, I've always respected and valued your opinion as you often offered a different view on things and allowed me to widen my own view. I would like to thank Dr. Riender Happee for your support in reviewing my thesis and for giving me the opportunity to work on the WEpods project when I was in dire need of a different subject. But also in the discussions about the academic life and process. You might have not noticed, but this has helped me a lot.
I would like to thank all the members of the doctoral committee for their time and effort reviewing my thesis.

I would have never finished this PhD without the support of my close colleagues and friends. Aswin, you have been my compatriot in obtaining a PhD and an amazing friend and colleague for nearly a decade. I value the many hours in scientific discussion and your courage in your steadfast pointing out my errors despite my stubbornness. Your motivation, commitment and the many times you gave me the well needed kick in the proverbial behind, made it possible to finish this thesis.
Machiel, you have been a great friend and have shown how to endure the hardships that are encountered in ones life. Together with Aswin you've made the many long days and nights we worked on Robby and Lea for the Robocup enjoyable.
Maja, you are a wonderful friend with your caring and friendliness you have made me grow to become more than a mere Sheldon. Thank you for opening your home when I needed due to my insufficient planning skills. The many wonderful experiences and memories we four made will always be cherished.
Many thanks go out to dr. Boris Lenseigne, your humour made it fun to work next

to you all those years. The mathematical discussions we had are valued very much. Special gratitude go to Martijn Wisse for giving me the opportunity to become the coordinator for the Minor Robotics. This has given me a lot of experience, self-knowledge and joy over the years.
I would like to thank all the members of the Delft Biorobotics laboratory and the Vision Based Robotics department. You have created a productive and wonderful work atmosphere. I've enjoyed the many Friday afternoon drinks, coffee moments and talks I've had.
Thanks go out to all the other colleagues at the university. Over the years I've enjoyed the many events, courses and interactions I've had.
Many thanks go out to the partners in the 3D Fetoscopy project, but foremost Suzanne Peeters who showed me the intricacies of TTTS and introduced me to Prof. Chiba who kindly allowed me to do an internship for which I'm very grateful.
Special words of gratitude go out to my students. I had a great time teaching and working with you. Especially Geetank and Tim, without your hard work this thesis would not have seen the day of light in this form.
Thanks go out to Wouter Caarls and Marcel de Vries for having the patience to teach me C++ which I've been using non-stop for so many years now.
Many words of gratitude go also out to all the colleagues in RCS, RRC and JMR over the years. It has always been a pleasure to work with you.
Bas, thank you for being the best friend one can wish for. I have so many good memories of the good times we've had and I appreciate all the moments where we could share our experiences.
All my other friends, you are just too many to name all, but you'll never be forgotten!

Last, I want to say many warm words to my family. Dad, your scientific mind has endorsed me to pursue a PhD and grow into the scientific person I'm now. Mum, your love and understanding of people want me not to be a mere engineer or scientist, but become a valuable member of society and take an active role in that. Joachim, I'm so lucky with you as my brother. Even though we differ so much, we understand and support each other infinitely. Eriko, you are the love of my life, I can't image it without you. I'm so grateful you gave me the opportunity to pursue my dreams. I hope that I can show you for the rest of my life what you mean to me. Hiroko, Chikara, Rebecca, Hayato, and Eri, thank you for accepting me in your family. I've always felt welcome and appreciate all you've done for me.

Finally, some last words go out to Squad for developing Kerbal Space Program. Without this game the hard moments of my PhD would have been twice as hard, although you've also been the reason why some of my thesis has been delayed.

About the author

Floris Gaisser was born in The Hague, The Netherlands on February 18, 1984. In 2003 He obtained his Atheneum diploma at the Openbaar Lyceum de Amersfoortse Berg in Amersfoort. In the same year, he began his study in Industrial Design at the Delft University of Technology. After completing his bachelor degree in 2007, he started in 2008 at Mechanical Engineering in the master track Intelligent Mechanical Systems. After completing an internship of four months in 2012 at NEC in Japan, he received his M.Sc degree in Mechanical Engineering in 2013. He conducted his M.Sc. thesis on the topic of *Face recognition for cognitive robots* at the Vision Based Robotics department, supervised by Prof. dr. ir. P. P. Jonker. Following his M.Sc. graduation, he was a research assistant at the Vision Based Robotics department for 8 months where he worked on sub-millimeter localization of EEG sensors on a sensor cap, as well as self localization for Augmented Reality.

Continuing in image processing, he started his doctoral studies in October 2013 at the BioMechanical Engineering department on in-vivo placenta reconstruction with fetoscopic video to support laser coagulation treatment of TTTS. Supervised by dr. ing. Maja Rudinac, dr. ir. Boris Leseigne and Prof. dr. ir. Pieter Jonker. This research topic involved the development of panorama reconstruction algorithms suitable for in-vivo fetoscopic videos. Between May and August 2015, he was a visiting researcher at the Nihon University, Tokyo Japan, at the National Center for Child Health and Development where he worked with Prof. Dr. T. Chiba on 4K fetoscopic panorama reconstruction. He also obtained interest in the WEpods project and assisted in the development of Object Recognition, Tracking and Prediction algorithms. During the course of his doctoral studies he has supervised multiple M.Sc thesis projects, was coordinator of the minor Robotics, gave multiple B.Sc courses as part of the minor Robotics and also assisted in teaching activities for the M.Sc courses 3D Robot Vision and Robot Practicals.

In 2017 he started as a perception engineer in the Automated Driving group of Robot Engineering Systems, Delft. Where he is currently working as Product Lead Automated Driving and is developing various perception algorithms for Automated Driving Systems, such as the WEpod and the Mission vehicle.

List of publications

Journal papers

Road attention: map-based vehicle trajectory prediction for interaction models
Tim Reesink, Floris Gaisser, Pieter P. Jonker
Submitted to: *IEEE Intelligent Vehicles Transactions*

Stable Image Registration for In-Vivo Fetoscopic Panorama Reconstruction
Floris Gaisser, Suzanne Petters, Boris Lenseigne, Pieter P. Jonker, Dick Oepkes
Appeared in: *Journal of Imaging, Jan 2018*

Conference papers

Road Infrastructure Indicators for Trajectory Prediction
Geetank Raipuria, Floris Gaisser, Pieter P. Jonker
IEEE Intelligent Vehicles Symposium (IV) 2018

Fetoscopic Panorama Reconstruction: Moving from Ex-vivo to In-vivo
Floris Gaisser, Suzanne Petters, Boris Lenseigne, Pieter P. Jonker, Dick Oepkes
Annual Conference on Medical Image Understanding and Analysis (MIUA) 2017

Road User Detection with Convolutional Neural Networks: An Application to the Autonomous Shuttle WEpod
Floris Gaisser, Pieter P. Jonker
Annual Conference on Machine Vision Applications (MVA) 2017

Image Registration for Placenta Reconstruction
Floris Gaisser, Pieter P. Jonker, Toshio Chiba
IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW) 2016

Online face recognition and learning for cognitive robots
Floris Gaisser, Maja Rudinac, Pieter P. Jonker, David Tax
International Conference on Advanced Robotics (ICAR) 2013

Propositions

%section*Propositions

1. Progress in the field of deep learning is hindered by our limited understanding of how we learn ourselves. (This thesis)

2. Constraining the problem to get the research done, does not necessarily bring us closer to resolving the problem. (Pt. 1 of this thesis)

3. Knowing why it does not work is just as valuable as knowing what works. (Ch. 3 of this thesis)

4. The ease in which humans can learn a task is not indicative of how well a machine can learn that task. (This thesis)

5. A Machine Learning researcher's dream is to experience growing up again.

6. Without reality, in the sense-think-act paradigm only the thinking (AI) remains, i.e. building castles in the sky.

7. There is no absolute understanding of the universe, only how we perceive and interpret it.

8. When the answer is staring you in the face but cannot be seen, you should not only open your eyes but your mind as well.

9. Understanding what and how to learn is the first step to (artificial) intelligence.

10. For humans the accumulation of experience is one's life, whereas for artificial intelligence it is only to perform a task.

These propositions are regarded as opposable and defendable, and have been approved as such by the promotors Prof. dr. ir P.P. Jonker, Prof. dr. J. Dankelman and Dr. ir. R. Happee.

Stellingen

1. Vooruitgang in Deep Learning onderzoek wordt gehinderd door ons beperkte inzicht in over hoe we zelf leren. (Dit proefschrift)

2. Inperken van het probleem om het onderzoek gedaan te krijgen, helpt ons niet dichterbij een oplossing voor het probleem. (Deel 1 van dit proefschrift)

3. Weten waarom iets niet werkt is net zo waardevol als weten wat wel werkt. (H. 3 van dit proefschrift)

4. Het gemak waarmee mensen een taak kunnen leren, is niet indicatief voor hoe goed een machine die taak kan leren. (Dit proefschrift)

5. Iedere Machine Learning onderzoeker's droom is om nogmaals opgroeien te kunnen ervaren.

6. Zonder de werkelijkheid in het "sense-think-act" paradigm blijft alleen het denken ('AI') over, oftewel het bouwen luchtkastelen.

7. Absoluut begrip van het universum bestaat niet, alleen hoe we iets waarnemen en interpreteren.

8. Als het antwoord je toelacht, maar je ziet het nog steeds niet, moet je niet alleen je ogen te openen, maar ook je geest.

9. Begrip van wat en hoe te leren is de eerste stap naar (kunstmatige) intelligentie.

10. Voor mensen is het vergaren van ervaringen ons leven, maar voor kunstmatige intelligentie dient het slechts om een taak te kunnen uitvoeren.

Deze stellingen worden opponeerbaar en verdedigbaar geacht en zijn als zodanig goedgekeurd door de promotoren Prof. dr. ir P.P. Jonker, Prof. dr. J. Dankelman en Dr. ir. R. Happee.