Computer vision for advanced driver assistance systems

Computer Vision for Advanced Driver Assistance Systems

Computer Vision for Advanced Driver Assistance Systems

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op donderdag 29 oktober 2020 om 13:30 uur

door

Willem Pieter Sanberg

geboren te Tilburg

Dit proefschrift is goedgekeurd door de promotor en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.ir. A.M.J. Koonen
1e promotor:	prof.dr.ir. P.H.N. de With
copromotor:	dr. G. Dubbelman
leden:	prof.dr. B. Leibe (RWTH Aachen University)
	prof.dr. D.M. Gavrila (Technische Universiteit Delft)
	prof.dr.ir. G. de Haan
adviseurs:	dr.ir. J. Elfring
	dr. M. Nieto (Vicomtech)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Voor mijn ouders,
geen twijfel mogelijk

Computer Vision for Advanced Driver Assistance Systems

W. P. Sanberg

Cover photo: W. P. Sanberg (cycling in Zuid-Limburg with Jos and Robert)
Cover editing and design: W. P. Sanberg
Printed by: Gildeprint - The Netherlands

ISBN 978-90-386-5128-6
NUR-code 959

Summary

Computer Vision for Advanced Driver Assistance Systems

Mobility of persons and transportation of goods involve multiple aspects and play a central role in society. This is particularly important when facing important challenges like infrastructure and safety, but also concerning environmental issues and health-related mobility restrictions. Mobility and transportation facilitate human interaction, have a tremendous impact on economy and tend to positively contribute to societal equality. Since both the world population and the rate of urbanization are growing, mobility is an essential factor in envisioning the future of society, cities and the environment. The demand for safe, comfortable, accessible and sustainable urban mobility will thus remain a challenging topic for the coming years, requiring advances in technology, policy making and both entrepreneurial and societal innovations. This motivates the development of Advanced Driver Assistance Systems (ADAS), aiming at supporting or automating driving tasks. This thesis contributes to this field, both by offering additional functionality and by extending the operational domain. The focus is on affordable systems without external communication or HD-maps and solely relies on stereo camera vision, since it provides information on both appearance and geometry of the vehicle's surroundings in a cost-effective way. The reported research entails three specific topics: freespace segmentation, geometric 3D scene modeling and collision warning.

Freespace segmentation: The first part of the thesis concerns freespace segmentation, which finds the available region within a traffic scene where the car can drive. The first contribution in this topic forms a color-based extension of the disparity Stixel World algorithm (Chapter 1). The algorithm relies on fusing an efficient histogram-based color analysis into the optimization process of the original disparity-only Stixel World algorithm. This reduces the detection of false obstacles that originally occurred at erroneous disparity data, caused by difficult imaging conditions such as bright sun or rainy weather. By updating the color modeling with a self-supervised learning scheme while the car drives, the freespace segmentation score improves from 0.86 to 0.97 on newly recorded public data. As a second contribution, the system latency is improved by introducing the color-based Stixel World algorithm (Chapter 2). Its online self-supervised color modeling considers real-world surfaces, instead of pixel counts when building the color histograms, thereby making the modeling distance-aware. This allows

the freespace segmentation to operate without the most recent disparity signal, so that the disparity estimation can be removed from the critical system path, while preserving the quality of the results. This is achieved by using previous disparity measurements to translate pixel colors into region class probabilities, which are then employed in the cost-optimization function of the Stixel World algorithm. On a new dataset with a focus on adverse, rainy conditions, the false obstacle detection rate decreases from 17 % to 13 %, while doubling the throughput rate. The third chapter on freespace segmentation presents an online self-supervised convolutional neural network (Chapter 4). Experiments show that it is feasible to train this neural network with automatically generated training masks, even when they contain errors due to adverse imaging conditions. This reduces the need for manual labeling, and facilitates updating the neural network during driving, so that the system can handle changing environments while using a small efficient neural network only. More specifically, the proposed algorithms with online training outperform the offline reference methods with 5 %, both for $F_{\max}$ and AP. More importantly, the original FCPN (without online training) that was successful on KITTI data performs worse than the baseline on the new data set. This indicates that the online training strategy is a good and efficient proposal to enable the use of a small neural network under varying conditions. As an additional result, pretraining the network speeds up the proposed online training with a factor of five. Overall, this strategy boosts performance with 4.2 % in adverse imaging conditions compared to the Stixel World baseline, thereby increasing the system robustness.

Geometric 3D scene modeling: The second topic that is addressed is efficient geometric 3D scene modeling. The context is military surveillance capturing image data from a surveillance vehicle, where live and historic images are registered for scene analysis (Chapter 6 A. To this end, the disparity Stixel World representation is extended by (1) rotating stixels along their vertical axes so that they are aligned to the orientation of the modeled surface, (2) adding interpolated stixels that cover vertical gaps, and (3) applying pixel masking within stixel rectangles to remove potential small background areas. These additions allow generating a textured 3D model of the scene, which is then used to render a synthetic view as a registered image for change detection. The work was implemented in CUDA for real-time execution on HD+ images in a prototype vehicle. The proposed additions together increase the pixel-level registration accuracy with 6 % on new, manually annotated data. When the lateral viewpoint offset between images is low, 97 % of the registrations by the enhanced system are within 5 pixels, and 79 % of the pixels fall even within a 1-pixel margin. When the live recording is from an offset more than a regular lane width, the system still achieves reasonable accuracy of about 70 %.

Collision warning: The final topic of this thesis is collision warning, for which two systems are presented. ASTEROIDS is a class-agnostic, probabilistic processing pipeline, which generates warnings from disparity and optical pixel flow data (Chapter 5). The method is not limited to specific classes or scenarios, and does not require external HD maps, vehicle communication or high-level semantic information. The system tracks stixels over time and samples so-called asteroid particles based on an uncertainty analysis of the measurement process. This is enclosed in a Bayesian histogram filter around a time-to-collision versus angle-of-impact state space, which is analyzed further with a peak-alarm detector (CFAR). The presented probabilistic approach is robust against low-quality input data for disparity and optical flow. As a result, performance quality is not hampered by employing disparity and flow methods that are less compute-intensive, reducing resource requirements. The evaluation shows successful performance on three different datasets. Namely, the ASTEROIDS approach does not generate any false warnings on the well-known KITTI tracking data set, it detects all but one collisions on newly simulated data, and performs reliably without errors on newly recorded data with many near-collisions. The ASTEROID system generates warnings typically around 1.2 seconds ahead of impact in nighttime and around 2 seconds upfront during the day. The second approach, SSCOD, fuses pixel-level semantic segmentation from a neural network with disparity stixels, which are then clustered using a customized DBSCAN process (Chapter 6 B). This is validated with a Forward Collision Warning module running in a prototype vehicle. Typically, SSCOD detects vehicles in front of the car at least 40 meters ahead in daytime and at least 35 meters ahead in nighttime conditions. This method has been successfully demonstrated live at the ITS European Congress in 2019 using a real vehicle with an integrated system.

In conclusion, this thesis contributes to freespace segmentation, 3D geometry modeling and collision warning for ADAS. The contributions are characterized by robustness under adverse imaging conditions, which are typically less considered in datasets and published research work. This explains why the color-extended stixel world algorithm and the associated online training schemes have drawn clear interest in the research community. Furthermore, the real-time stixel implementation with orientation alignment, the elegant small neural network design for scene analysis, and the hybrid collision warning approach where traditional techniques enable a small neural network implementation are contributions of interest for the industry, and the former was already integrated in an industrial prototype for professional applications. The followed path in research leads to systems of feasible complexity, employing realistic AI solutions, which readily contribute to increasingly safe and automated driving.

Samenvatting

Computervisie voor geavanceerde ondersteuningssystemen ten behoeve van voertuigbesturing

Mobiliteit van personen en goederentransport spelen een centrale rol in de maatschappij. Dit levert vooral relevante uitdagingen op voor infrastructuur en veiligheid en beïnvloedt ook milieuaspecten en gezondheidsgerelateerde beperkingen in mobiliteit. Mobiliteit en transport faciliteren menselijke interactie hebben een grote impact op de economie en aanwijsbare bijdrages aan sociale gelijkheid. Aangezien de wereldbevolking en de verstedelijking beide toenemen, is mobiliteit een essentiële factor in toekomstvisies op de maatschappij, steden en het milieu. Hierbij blijft de vraag naar veilig, comfortabel, toegankelijk en duurzaam stedelijk vervoer een uitdagend onderwerp voor de komende tijd en vergt innovaties in technologie en regelgeving en aanpassingen in zowel ondernemerschap als maatschappij. Het bovenstaande motiveert de ontwikkeling van geavanceerde systemen om voertuigbestuurders bij hun taken te ondersteunen of deze deels te automatiseren (*Advanced Driver Assistance Systems*, ADAS). Dit proefschrift draagt bij aan dit onderzoeksveld door middel van het uitbreiden van zowel de functionaliteit als de condities voor de operationele inzet van deze systemen. De focus ligt op betaalbare systemen die onafhankelijk zijn van externe communicatie en/of zeer gedetailleerde digitale kaarten, waarbij de systemen enkel gebruik maken van stereocamera's, omdat deze op efficiënte wijze visuele informatie geven over de omgeving en zijn geometrie.

Het gepresenteerde onderzoek omvat drie onderwerpen: het detecteren van vrije ruimte (*freespace*), het in 3D geometrisch modelleren van de omgeving en het waarschuwen voor aanrijdingen (*collision warning*).

Detectie van vrije ruimte: Het detecteren van de vrije ruimte komt neer op het bepalen van de beschikbare ruimte waar het voertuig kan rijden in de actuele verkeerssituatie. De eerste bijdrage op dit onderwerp is een kleurgebaseerde uitbreiding van het zogeheten *Stixel World* algoritme (Hoofdstuk 2). Deze uitbreiding is gebaseerd op het integreren van een efficiënte histogramanalyse van de beeldkleuren, in het optimalisatieproces van het oorspronkelijke dispariteitsalgoritme. Deze uitbreiding reduceert de detectie van valse obstakels die eerst voorkwamen bij verkeerde dispariteitschattingen door ongunstige opnamecondities zoals fel zonlicht of regenachtig weer. Door de kleurenanalyse uit te breiden met een zelflerend mechanisme voor gebruik tijdens de autorit, verbetert de *freespace* detectiescore van 0.86 naar 0.97 met nieuwe opnames, die ook publiekelijk toegankelijk zijn. De tweede bijdrage betreft het verminderen van de systeemvertraging van de detectie door het introduceren van een op kleuren gebaseerd *Stixel World* algoritme (Hoofdstuk 3). De zelflerende kleurenanalyse maakt histogrammen van de voorkomende oppervlaktes in plaats van beeldpixels te tellen, zodat het systeem daarmee afstandsbewust wordt. Hierdoor heeft het systeem niet meer de meest actuele dispariteitsmetingen nodig, zodat deze meting niet meer in het kritieke systeempad uitgevoerd hoeft te worden met behoud van de kwaliteit. Dit wordt bereikt door pixelkleuren aan de hand van oudere dispariteitsmetingen te vertalen naar waarschijnlijkheden voor klasses, welke worden toegepast in de kostenoptimalisatie van het kleurgebaseerde *Stixel World* algoritme. Het gebruik van een nieuwe dataset met een ongunstige opnamecondities toont aan dat de detectie van valse obstakels reduceert van 17 % naar 13 %, terwijl de systeemdoorvoer wordt verdubbeld. Het derde hoofdstuk over *freespace* detectie presenteert een methode die gebruik maakt van een zelflerend convolutioneel neuraal netwerk (Hoofdstuk 4). De experimenten wijzen uit dat het haalbaar is om dit netwerk te trainen met automatisch gegenereerde datalabels, zelfs als deze labels fouten bevatten vanwege ongunstige opnamecondities. Dit reduceert de afhankelijkheid van handmatig geannoteerde datasets en faciliteert bovendien dat het netwerk kan worden geactualiseerd terwijl het voertuig in gebruik is. Hierdoor kan het systeem omgaan met veranderingen in de omgeving terwijl het slechts van een klein efficiënt neuraal netwerk gebruik maakt. Het nieuwe zelflerende systeem scoort 5 % beter in $F_{\max}$ en AP dan de standaardaanpak. Nog belangrijker is dat de standaardaanpak (die niet wordt geactualiseerd tijdens het rijden) goed scoort op de veelgebruikte publieke KITTI dataset, maar ondermaats presteert op nieuwe data. Dit impliceert dat de zelflerende aanpak een goede en efficiënte manier is om te kunnen vertrouwen op een klein neuraal netwerk onder variabele condities. Een extra resultaat is dat het actualiseren van het netwerk tijdens gebruik vijf keer sneller gaat als het vooraf al getraind is met relevante data. In zijn geheel verbetert het algoritme de kwaliteit van de resultaten met 4.2 % vergeleken met de dispariteitsmethode, waardoor de robuustheid van het systeem toeneemt.

Geometrische 3D scènemodellering: Het tweede onderzoek betreft efficiënte geometrische 3D scènemodellering. De context van dit onderzoek is militaire inspectie die gebruikt maakt van beeldmateriaal dat is opgenomen vanuit een patrouillevoertuig, waarbij huidige en historische beelden over elkaar worden gelegd voor automatische veranderingsdetectie (Hoofdstuk 6 A). Om dit te realiseren, wordt de *Stixel World* representatie uitgebreid met: (1) rotatie om de verticale as om *stixels* uit te lijnen langs het oppervlak dat ze modelleren, (2) geïnterpoleerde *stixels* om verticale openingen in het model op te vullen en (3) het wegfilteren van achtergrondpixels binnen stixelrechthoeken. Met deze toevoegingen kan de stixelrepresentatie worden gebruikt om een 3D model met textuur te genereren, en vervolgens een geregistreerd beeld te synthetiseren voor de veranderingsdetectie. Het systeem is geïmplementeerd in CUDA om beelden van hoge resolutie in *real-time* te kunnen analyseren in het prototypevoertuig. De toevoegingen aan het model verbeteren de nauwkeurigheid van de beeldregistratie met 6 % op nieuwe, manueel geannoteerde data. Bij een beperkte afstandsverschuiving tussen de huidige en de historische beelden valt na registratie 97 % van de geannoteerde punten binnen 5 pixels, en 79 % zelfs binnen 1 pixel van de annotatie. Bij een laterale afstand van meer dan een reguliere rijbaanbreedte haalt het systeem nog steeds een acceptabele registratiescore van ongeveer 70 %.

Collision warning: Het derde en laatste onderwerp is waarschuwen voor aanrijdingen, waarvoor twee systemen worden gepresenteerd. Het is eerste is ASTEROIDS, dat een klasse-agnostische, probabilistisch verwerkingsproces is, dat gebruik maakt van dispariteit en beeldbewegingsdata (Hoofdstuk 5). De analyse is niet beperkt tot specifieke obstakeltypes of scenarios en is onafhankelijk van externe digitale kaarten, voertuigcommunicatie en semantische informatie. Het systeem volgt *stixels* over tijd en genereert zogenoemde *asteroid*-deeltjes, waarbij de onzekerheden in het meetproces worden meegenomen. Deze deeltjes worden eerst bewerkt met een Bayesiaans histogramfilter op een toestandsruimte met impacttijd en impacthoek als dimensies, waarna een analyse volgt met een piekalarmdetector (CFAR). Deze probabilistische aanpak is robuust tegen invoerdata van lage kwaliteit. Hierdoor blijft de waarschuwingsfunctie betrouwbaar wanneer minder complexe dispariteits- en bewegingsschatters worden gebruikt met minder rekenkracht. De evaluatie op drie verschillende datasets toont goede prestaties. ASTEROIDS genereert geen valse waarschuwingen op de bekende KITTI dataset, het detecteert alle aanrijdingen op één na in een nieuwe gesimuleerde dataset, en het toont betrouwbare kwalitatieve resultaten op een nieuwe dataset met opzettelijke bijna-aanrijdingen. Het ASTEROID-systeem waarschuwt ’s nachts ongeveer 1.2 s en overdag ongeveer 2 s vooraf aan een mogelijke aanrijding. Het tweede waarschuwingssysteem, SSCOD, combineert semantische informatie op pixelniveau verkregen met een neuraal netwerk, met dispariteitsstixels, welke vervolgens worden gegroepeerd met een aangepast DBSCAN algoritme (Hoofdstuk 6 B). Dit systeem is gevalideerd in een module voor waarschuwingen bij frontaal aanrijdingsgevaar. Over het algemeen detecteert SSCOD voertuigen die de weg blokkeren overdag minstens 40 m van tevoren en ’s nachts op ongeveer 35 m. De module is succesvol gedemonstreerd als prototype in een echt voertuig op het Europees ITS Congres in 2019.

Concluderend, dit proefschrift draagt bij aan de detectie van *freespace*, 3D geometrische scènemodellering en *collision warning*, waarbij alle aspecten ingezet worden voor ADAS. De bijdrages worden gekenmerkt door hun robuustheid tegen ongunstige openamecondities, wat weinig aandacht krijgt in wetenschappelijk onderzoek en datasets, zodat de gepresenteerde zelflerende kleurenanalyses goed zijn ontvangen. Bovendien zijn er drie relevante bijdrages gepresenteerd voor de industrie: het *real-time* stixelsysteem met orientatiespecificatie, het elegante ontwerp met kleine neurale netwerken voor scène-analyse en de inzet van traditionele methodes om de vereiste rekenkracht van neurale netwerken te verminderen. De eerstgenoemde is al gevalideerd in een industrieel prototypevoertuig voor professionele toepassingen. Het onderzoek in dit proefschrift leidt tot systemen met acceptabele complexiteit die gebruik maken van realistische AI oplossingen die direct nuttig bijdragen aan veilig en geautomatiseerd rijden.

Contents

Chapter 1 Introduction

1.1 Smart vehicles and the future of personal transport

Mobility of persons and transportation of goods involves multiple aspects and plays a central role in society, particularly at the time of writing this thesis, when society faces important challenges around environmental issues and health-related mobility restrictions. According to the Global Agenda Council on the Future of Automotive & Personal Transport, mobility has an influence on multiple societal dimensions, like that it facilitates human interaction, it has a tremendous impact on economy and it even tends to positively contribute to societal equality [1]. Since both the world population and the rate of urbanization are expected to grow, mobility is an essential factor in envisioning the future of society, cities and the environment. The demand for safe, comfortable, accessible and sustainable urban mobility will thus remain a challenging topic for the coming years, requiring advances in technology, policy making and both entrepreneurial and societal innovations. Considering the various nature of the previous aspects, these challenges also bring interesting opportunities. If policy makers can be open to mobility innovations and incorporate them into their long-term societal planning, new mobility models are expected to bring economical and societal benefits [1].

On the aspects of accessible and comfortable mobility for personal transportation, most people travel for work on a daily basis, as a commuter or as part of their job. In addition, in their spare time people travel for trips, holidays and visiting friends and family. A large part of this need for mobility is currently being fulfilled with the use of a personal car [1]–[3].

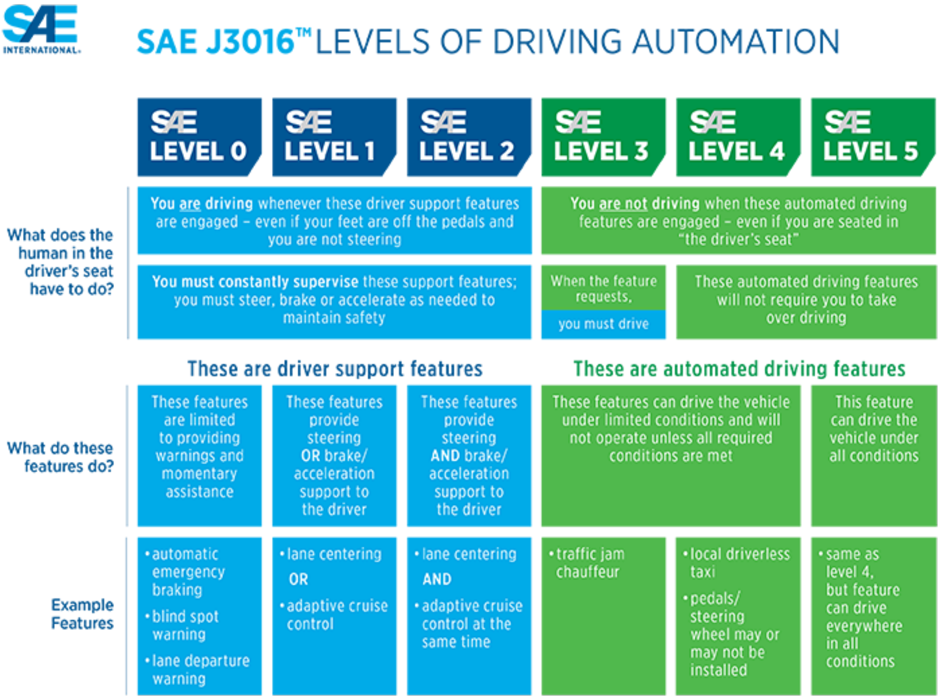
Considering road mobility from a safety perspective, the main challenge is to resolve traffic accidents. These accidents are mainly caused by human factors. USA-based research projects show that human errors play a critical role in 90% of accidents or even more [4], [5]. Advances in vehicle automation and intelligent transportation systems are expected to have a positive influence on these numbers [6].

Therefore, both academic and industrial efforts have been made in the past years to address the above-mentioned challenges on safe, comfortable and accessible mobility to safeguard its future. One primary objective is to increase the automation of vehicle control. This reduces the load on the human driver by increasingly taking care of aspects for driving the vehicle. This is expected to improve traffic safety and increase the comfort during traveling time, given the aforementioned reasons on traffic accidents and involved risk factors [6].

Considering the foreseen improvements of automated driving on the longer term, additional benefits can be realized. For instance, automated driving could lead to an increase in access to mobility, for example, for people currently not capable of driving such as physically handicapped or visually impaired people. Additionally, fully automated driving could also contribute to a reduction of occupied space that is currently reserved for cars in urban areas, either by optimized parking or car sharing services. Unfortunately, the predictions on the effect of automated driving on the environment are uncertain and may have positive or negative impact on energy usage (the predictions vary either way up to a factor two) [6].

A widely used system to categorize the development steps in the road map towards automated driving is introduced by the international Society of Automotive Engineers, the SAE. The scheme distinguishes six levels, numbered 0 to 5, ranging from *no automation* towards *full automation, anytime and anywhere* [7]. The levels are depicted and described in Figure 1.1. For example, the first steps consist of developing Advanced Driver Assistance Systems (ADAS), which support the human driver in certain driving tasks. This can be informative, such as measuring the distance towards the preceding vehicle, or active, such as keeping a safe distance at all times with an Adaptive Cruise Control (ACC) system.

These depicted SAE levels of automation in mobility will be used as a central guideline for a further and more detailed discussion on the underlying technological requirements and system strategies. This detailing is addressed in the next section.



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 1.1 — *Levels of automation, as distinguished by the SAE (adopted from [8]).*

1.2 Strategic and technological landscape of future mobility solutions

This section will sketch the landscape of seven core strategies in system architectures that are currently under debate in the mobility community. Different routes are possible towards realizing full autonomous vehicles, where the overall problem has not yet been solved, given the immense societal implications.

These seven aspects are all closely linked, but broadly speaking, two clusters can be discerned. The first cluster is more related to societal questions and business models, and concerns (1) different development strategies to increase automation, (2) ownership models and (3) vehicle connectivity. The second cluster contains the aspects that are more focused on the technical questions, namely (4) digital maps, (5) sensor selection, (6) computational resources and (7) data and data analysis. These aspects are discussed next in individual subsections, after which we elaborate upon how the work in this thesis is positioned in relation to this discussion.

1.2.1 Development strategies of automation

Increasing the extent of automated functionality in mobility typically appears in a gradual fashion, since it requires both law adaptation and acceptation from vehicle drivers in society.

For an L0 automated car, two different potential trajectories of improvement are generally recognized, based on either extending functionality first or extending the operational domain first. This is often referred to as *everything somewhere*, with gradual increase of the operational domain, versus *something everywhere*, with gradual increase of functionality. As an example of the latter, companies such as Tesla, Daimler and MobilEye first build lower SAE-level ADAS (FCW, ACC, LKA, etc.)[1], which are immediately released on the general market, after which the makers then aim at gradually increasing the functionality [9]. An argument against this approach is that drivers tend to have difficulties with adapting to a partially automated car (L2 or L3) that requires monitoring by the human to overcome potential imperfections in the system [10]. Humans constantly have to assess the performance of the system and be ready at all times to take over with manual control. In reality, humans are prone to be distracted by performing secondary tasks such as reading or interacting with their phones or touch screens, which hampers their readiness to take control [11], [12].

To avoid this transition stage while still ultimately providing an L5 automated car offering full functionality under all circumstances, companies such as Waymo, Uber, Delphi (via the acquisition of nuTonomy) and General Motors (via the acquisition of Cruise) follow an alternative strategy. For instance, they aim at developing a prototype car of at least L4, followed by medium-scale market penetration of such cars that are made available within a certain geographical area and climate, after which they then aim at alleviating those limitations for mass deployment [6], [9]. The various difficulties with this approach are the high initial cost of development, the large time-to-deployment, and the uncertainty in long-term acceptance and legalization.

1.2.2 Private ownership versus shared services

Closely related to the development strategies for automation is the choice between development for privately owned cars versus Mobility as a Service (MaaS) without personal ownership.

Privately owned cars need to be affordable for mass adoption, impeding the use of an expensive sensor suite [13], [14]. At the same time, the scope and the business infrastructure (like maintenance outlets and, if required, digital maps) should be available everywhere from the start. This topic has a societal aspect as well, where some parties predict that future generations will not value car ownership [1], while others are reluctant to lose the freedom and flexibility that an own car offers and/or expect that shared cars will not meet high levels in availability, maintenance and hygiene [6], [14].

In contrast, MaaS covers all concepts that offer mobility and transportation to customers, for example, via shared cars or robotic taxis, instead of people owning a private car [1]. The projected benefits of MaaS for developing autonomous cars are both cost and feasibility. Proponents expect to amortize the costs of development,

[1] FCW: Forward Collision Warning; LKA: Lane-Keeping Assist(ance)

deployment, and infrastructure for service and maintenance over a fleet of cars that is owned by the MaaS provider[9], [15]. Additionally, MaaS allows a gradual, self-paced increase of automated functionality and its operational domain [9], [15]. However, a potential issue with MaaS is that it requires a high adoption rate to be successful [6], while research indicates that not all available systems currently perform well enough for user acceptance [16] and that designers should particularly focus on providing easily accessible systems [17]. The aspect of a high adoption rate is equally important in the aspect of the next subsection, which concerns vehicle connectivity.

1.2.3 Connected versus stand-alone

The dependency on a high adoption rate of automated vehicles holds for another fundamental architectural choice as well: developing vehicles that function stand-alone versus vehicles that communicate with each other and/or with the infrastructure for smart information and/or control.

Cars with vehicle-to-vehicle (V2V) communication exchange information on velocities and trajectories (and potentially much more) with cars around them. This communicated information is more accurate and available at a lower latency than when it should be measured individually, which can make the difference between a safe evasion maneuver and a fatal crash. However, cars can only safely rely on this functionality if enough traffic participants are connected and sharing information [6], and even then, fallback systems need to be in place for situations where communication is not available. Ideally, this connectivity would not be limited to vehicles, but also include vulnerable road users such as pedestrians and cyclists. Another type of communication is vehicle-to-infrastructure (V2I) communication, for example, when a centralized system sends out congestion information and speed advise to improve traffic flow and density. The impact of these systems is also larger when more vehicles participate, but this requires a substantial initial investment in infrastructure for roadside equipment [1], [6].

1.2.4 Digital maps

The fourth key architectural choice concerns the integrated use and dependency on digital maps. More specifically, the level of detail required in such a map is crucial and addresses the fundamental question of balancing the intelligence between the car and the digital map. Several real-world demonstrations of highly automated driving, such as the tests of Waymo and Daimler's Intelligent Drive, heavily relied on high-definition digital maps (HD maps) that contain detailed information on the layout of the infrastructure, such as centimeter-accurate annotations of road markings and traffic signs [18]. In these approaches, the car effectively drives on a digital track though the digital world. This reduces the amount of processing that is required in the car, since it 'merely' needs to recognize its location with respect to the map and avoid collisions with dynamic traffic participants. However, generating such a map requires advanced technology in sensing, registration and automation, and is even deemed infeasible by several parties [19]. Crucial issues are continuous updating the map at the required level of detail in all relevant areas, and having such data available in the car at all times in a timely cost-effective manner. Additionally, the system should still function safely when the map data contains errors or is unavailable.

Relying on digital maps requires a clear definition of what can be considered static information to be available in the map and dynamic information of the surroundings to be sensed and interpreted by the car itself. Some static information content can change, requiring an update of the map (new speed limit, new roundabout), while some objects in the surroundings can change and need to be flagged and repaired in the real world (road damage, worn markings, stolen traffic signs). Additionally, some scenes change only temporarily (road works), whereas some object changes are irrelevant (garbage bins, parked cars, blossoming trees). Several parties propose to address these issues with a crowd-sourcing approach, where each traffic participant uploads information to a cloud that updates HD maps continuously where necessary and sends those updates back towards the users. Especially as part of a MaaS application, this could be an interesting solution, provided that it has sufficient participants and a proper V2I infrastructure at its disposal.

However, many benefits of an HD map are lost if the vehicle cannot determine its own position accurately. This generally sets requirements on the sensor equipment, which is discussed in the next subsection.

1.2.5 Sensors for environment perception and awareness

With a growing degree of automation in the vehicle and the associated decision making, the vehicle should be increasingly aware of its dynamic surroundings, which cannot be fully derived from digital maps. This implies that complementary sensors are applied in the vehicle, which aim at providing information of the actual situation. The related architectural design choice then concerns the selection of an appropriate sensor suite for the vehicle, matching with the aforementioned trend. As mentioned earlier, sensors can have a considerable impact on the cost of the automation system[13], [14], while they simultaneously are key in environment perception.

This section briefly presents the most important options in sensor selection in the current status of developments. First of all, centimeter-accurate localization to exploit the full potential of HD maps typically requires either an expensive RTK-GPS (Real-Time Kinematic Global Positioning System) combined with an IMU (Inertial Measurement Unit) module, or advanced vision-based localization algorithms and the corresponding, accurate reference data in the map. Second, there are generally three modalities for environment perception operational in practice: cameras (for the visible light spectrum), Lidar (light detection and ranging, using infrared light) and Radar (radio detection and ranging, using relatively large wavelengths) [20]. Several ADAS or automated driving-related systems and deployment experiments have been presented that leverage a combination of multiple sensor modalities [18], [21], [22]. In contrast to the generally accepted benefits of camera and Radar sensors, the use and necessity of Lidar is currently still under discussion for automated cars. Lidar provides highly accurate distance measurements, but is currently by far the most expensive sensing modality [20]. Virtually all car manufacturers and suppliers rely on Lidar sensors and expect them to become less costly in mass production and through the development of solid-state Lidar [20], [23]. In contrast, an experimental autonomous driving event along the Bertha Benz Memorial Route was successfully performed in 2013 without using a Lidar sensor [18]. Also specific state-of-the-art fully electric cars offering partial self-driving functionality, are not equipped with Lidar and the manufacturing company claims to never apply it in future versions. They motivate this statement by observing that human drivers do not need laser light to drive, so exploiting a visible-light camera as a supplementary sensor should be sufficient. In the manufacturer's opinion, Lidar for automated driving should be considered a short-term solution that sets researchers on a costly detour and hampers them in building a reliable camera-based system [19]. While the vehicles in the previous examples combined camera sensing with Radar sensing, other parties have even presented a camera-only demonstration of self-driving. Even though the functionality has not yet been deployed commercially at a large scale, it still showed successful and safe automated driving through a crowded city center with complex traffic scenarios, while using only a single sensor modality [24].

In our research opinion, it is too early to draw a definitive conclusion on the final application of Lidar in the context of automated driving. A motivation for this lack of a final conclusion is that there has not yet been a large-scale deployment of systems with either functionality, so that a fair comparison cannot be made.

1.2.6 Computing requirements for automated driving

One of the enabling factors of the developments in automation is the increasing accessibility to compute resources. Especially the rapid expansion of the capabilities of algorithms that exploit Artificial Intelligence (AI) is tightly coupled to the increased mapping capacity of new silicon processor designs. These new designs make compute resources widely available at a reasonable cost, most notably via Graphical Processor Units (GPUs) for the personal computer market. These GPUs originally offered processing power for more advanced computer games, but were quickly additionally exploited for AI research, thereby facilitating parallel compute-intensive tasks on batches of data for demanding experiments on both desktop PCs and also in larger clusters or server setups.

However, although examples of research on practical deployment exist [25], [26], most of the above-mentioned capabilities serve at best as research platforms or prototype configurations, and are not suited for real-world deployment in embedded systems, requiring a high level of hardware robustness and having severe constraints on power consumption and physical packaging dimensions. Moreover, many state-of-the-art research efforts had a focus predominantly on task accuracy instead of ease of deployment, often leveraging increasingly advanced and complex AI models without regarding real-world feasibility constraints, especially in the case of deep learning (DL) and neural networks (NN). In contrast, deploying a DL system on an embedded computed device requires also serious considerations on latency, memory capacity, robustness in functionality, operational reliability and overall system cost. In practice, this means redesigning NN architectures or applying NN optimization techniques (compression, pruning, etc.) to fit the advanced designs within the constraints of embedded devices. Simultaneously, the hardware industry is developing new hardware compute nodes to extend resources for more advanced on-chip AI models. In this regard, there is easily a mismatch between common state-of-the-art research algorithms of a factor of 10 in both power and price.

Due to the increased awareness of DL inference cost, neural network designers have come up with very interesting new proposals which tend to lead to much more efficient designs than the existing state-of-the-art networks, while preserving or even boosting performance. An example is in the newly emerging research field of Neural Architecture Search (NAS). Recently, several first NAS examples have been presented that are hardware-aware, in the sense that the algorithms optimize the neural network architecture jointly for task accuracy and resource efficiency [27]–[29]. This optimization is part of the neural network training, which is the broader discussion topic of the next subsection.

1.2.7 Data and analysis for automation

Systems that rely on artificial intelligence require machine learning procedures that extract relevant information from data. It is generally accepted in the machine learning community that having more data leads to improved results, and even that robust results cannot be achieved without it. The major benefit of having a large and well-balanced dataset is that the system can learn beforehand about as many as possible situations that it can encounter during deployment, limiting the risk of incorrect behavior in unknown situations.

The data is utilized to train the algorithm to capture and extract the information that is relevant to the application at hand, for which different broad categories are distinguished. In general, the most widely adopted strategy of machine learning exploits *supervised learning*, which relies on training examples accompanied by labels with desired corresponding responses of the system (for instance, a picture of a car with a tag *car* and a bounding box of its location). This is in contrast with *unsupervised learning*, which solely uses the training examples (pictures of cars) without label information. Acquiring the proper labeling is costly, especially considering the extensive amount of samples that is required, since labeling often consists of labor-intensive manual annotations made by domain experts. Therefore, when designing machine learning-based systems, important problems to solve are the quality and acquisition of suitable data and its relevant labels. Many different strategies are used, of which several are interesting for this application field. One example is *weak supervision*, where the labels are of a lower level of detail than the desired output. Examples of *weak supervision* are providing only an object tag when a bounding box is desired, or providing only the bounding box when a pixel-wise segmentation mask is the final objective [30]. The rationale is that lower-level annotations are easier to obtain in large quantities, and that the training procedure should be able to exploit the large dataset for generalization of the model behavior, so that the model is capable to extract the desired labels itself.

A related broad category of training for machine learning is *natural* or *self-supervision*, which uses the natural structure of the data to generate labels for training in an automated way [31]. Examples of data aspects that one can exploit are temporal correlation within one signal and the mutual correlation between multi-modal signals. For instance, information between frames within a video sequence is temporally correlated. If the first frame is known to contain a car, then it is likely that the next frame contains that car as well, so that this assumption then could serve as a weak label. Similarly, when different sensors capture the same scene in different modalities, those data streams are then correlated by their contents. For instance, if an object location is known in a Radar measurement, and that measurement can be registered to a video stream, then the location of the car in the video can be calculated as well. This strategy exploits the transfer of information from one sensor domain to another and builds upon the strength of one sensor to interpret information from a complimentary sensor source [19].

Data sensing for mobility applications can also be implemented via crowd-sourcing, so that individual automated actors can learn from the collective experiences of their peers, which share knowledge of new, unseen scenarios with the community, such as applied in the open-source RoboEarth project [32]. Similarly, in a closed commercial example, car drivers in a fleet of vehicles sense their environments and communicate this information to a central data server for analysis. This allows to gather information at a large scale on new uncommon situations for the system and to update the other members within the fleet with the analyzed information [19]. A potential drawback of such a strategy is the privacy concern of drivers and other traffic participants and the legal implications of systems that can be updated after certification.

The next subsection draws conclusions from the strategic aspects from the above discussions, which results in the objectives and research questions addressed in this thesis.

1.3 Scope and objectives of this thesis

Given the previous discussions on both societal and technical aspects of automated driving systems, it can be concluded that all scenarios require the vehicle to have proper perception of its surroundings. This perception is obtained by a suitable sensor suite and by other means, such as advanced digital maps and connectivity.

Another aspect coming to the foreground is that the automated system has to deal with many uncommon situations in which the decision making should have a robust behavior. From a research point of view, contributing towards full automation seems to be most feasible by expanding circumstances and increasing functionality alternately. For instance, extending obstacle detection with obstacle-path prediction (functionality) versus improving robustness against weather conditions or environment content (circumstances) as a next step. Both improvements contain valuable considerations for the development towards full automation.

The focus of our research is on affordable and robust ADAS that can work in standalone vehicles. With this approach, we avoid dependencies on large-scale deployment or expensive sensory equipment and facilitate gradual growth of automation. For instance, we avoid depending on V2I or V2V communication, HD maps, RTK-GPS or Lidar equipment. Instead, we will focus on camera video processing alone. Additionally, we will aim at lowering dependencies on large-scale human-made data annotations. One strategy is using real-world or even physical-world modeling where possible (such as using depth/distance), thereby avoiding training to a large extent and alternatively relying on self-supervised training.

Note that these strategies do not prohibit the use of our work in vehicles with connectivity, or as part of a MaaS solution, or can exploit an available HD map. In fact, we aim at keeping our contributions generic and potentially offering either new or strengthened environment perception for supplementary redundancy.

ADAS requirements and approach motivation

From the perspective of the application domain, we focus mainly on two core driver-assistance tasks at the perception side, namely (1) static scene modeling and interpretation, predominantly for freespace detection, and (2) collision warning within the larger area of dynamic scene analysis. All research presented in this thesis is constrained to several general system requirements:

- hardware system is utilizing an affordable, camera-based setup;
- computation efficiency is reasonable so that real-world applicability is feasible using resource-constrained platforms;
- algorithms can be deployed in standalone operation and facilitate integration into larger systems;
- robustness should be considered with respect to difficult and varying conditions of everyday traffic;
- safety is optimized such that decision making prevents missing obstacles and avoids detecting too many of them.

Underneath these topics and constraints, we have several technological objectives that we aim to address, as follows. The general objective of this thesis is to investigate scene modeling for ADAS applications, such as freespace detection or collision warning. This scene modeling can be developed along several directions, e.g. scene geometry and dynamic scene analysis, which both support the previous themes. Smart decision making will evidently rely on AI algorithms, for which quality and quantity of input data is of paramount importance. This leads to the following objectives.

Improving geometric scene modeling by increasing the representation accuracy with extended analysis. Depending on the target application, geometric modeling should have a certain richness in its representation. For freespace segmentation, modeling flat fronto-parallel surfaces of obstacles blocking a trajectory is sufficient. However, this is insufficient for producing fully textured 3D models, which are useful in a surveillance context with live change detection.

Improving dynamic scene modeling by offering collision warnings for any class of obstacles. Current solutions are often limited to specific object classes or collision scenarios, so that alleviating these constraints can greatly extend the application domain.

Reducing data-annotation effort for training input to develop machine learning strategies. For current and upcoming vehicle automation systems, the level of annotation detail is at pixel level. For instance, panoptic segmentation requires pixel-precise annotations of (parts of) actors, objects and scene areas. Therefore, the human-expert annotation becomes intractably expensive. This readily implies the development of alternative ways of labeling or learning.

Exploring hybrid architectures, which are combinations of traditional computer vision and deep learning-based approaches. AI systems can only be successfully deployed in situations for which large amounts of training data are available. This inherently makes the application less robust for rare or unexpected situations. It is known that such situations can be better handled by more conventional computer vision models, not relying on heavy training, but leveraging world knowledge instead. Therefore, a joint combination of those techniques seems attractive and important for systems research towards robust and reliable operation.

1.4 Contributions

This section provides an overview of the scientific contributions presented in this thesis.

1.4.1 Contributions to freespace segmentation

We have developed and evaluated three novel variations of freespace segmentation, namely the color-extended Stixel World algorithm (Chapter 2), the color-only Stixel World algorithm (Chapter 3) and a color-only algorithm with a Fully Convolutional Network (FCN, Chapter 4). All three methods rely on disparity analysis for self-supervision with online training. Our first algorithm relies on strong data fusion for artifact reduction and has served as a breakthrough enabling online learning. The second algorithm is designed to reduce system latency by more sophisticated color modeling that omits strong fusion, while retaining the improved accuracy. The third version provides a similarly reduced system latency and yields improved performance due to more specialized color modeling. These methods show that online self-supervised color modeling facilitates efficient algorithms that can adapt to varying environment conditions. Specifically the latter of the three algorithms, which embeds our self-supervised online learning framework into neural-network based analysis, is well received by the scientific community.

Additionally, we contribute to the availability of public benchmarks for freespace segmentation by releasing a dataset called *EHV-road*, which concerns freespace segmentation. The dataset was released in three batches (2014, 2015 and

2017). It is a relatively small dataset, but it is focused on relevant uncommon cases with difficult imaging conditions or structures, making it relevant to test performance on specific pitfalls. It consists of stereo color data and also contains the 10 or even 30 preceding frames for each annotation to facilitate online training.

1.4.2 Contributions to collision warning

For general collision mitigation, it is important to embed redundancy into an application system. For this reason, we have explored stereo-based collision analysis in this context, by attempting to extract relevant and reliable collision information, starting from efficient disparity stixel processing. The evaluation of our framework has shown that this is indeed possible. A key aspect of the proposed method is that it avoids specific object detection or pretraining on certain scenarios. Instead, it provides warnings on collisions with any obstacle, using simple but effective newly introduced *asteroids* as a model for generic dynamic analysis for stereo vision input. This novel concept allows to directly model obstacles and their trajectories. Additionally, we incorporate the measurements into a newly designed state space, containing time and direction of impact, to directly assess potential collisions.

Next to developing and analyzing a collision warning system, we have also created new datasets for the required evaluation process. The first one is a set with simulated data, the PreScanStereoCollision Dataset, where the ego-vehicle drives on collision trajectories with various kinds of other traffic participants in different scenery. Using simulated data is unavoidable for quantitative experiments, since real-world datasets have no true collisions. The color stereo data is generated within the PreScan simulation environment. It contains ego-vehicle data and object annotations in the same format as the widely used KITTI-tracking benchmark for optimal usability. Additionally, real-world data from near-collisions was also recorded in different environments. This data is not annotated, but it can provide qualitative support of the concept.

1.4.3 Contributions to deep learning for ADAS

With respect to the field of convolutional neural networks, our contributions are in self-supervised training, exploiting small neural network architectures, and exploiting the combination of neural networks with traditional methods.

The first aspect contributes to reducing manual labeling effort of annotation experts, in turn allowing to increase the amount of training data easily. The second aspect, exploiting small neural networks, reduces the computational load of the in-vehicle system to facilitate embedded deployment. The third aspect shows the benefit of hybrid architectures, where traditional algorithms facilitate in reducing the complexity of the employed AI system, which in turn strengthens the applicability and benefits of AI in ADAS. All these aspects have been investigated within the context of freespace detection and collision warning.

1.4.4 Real-world system integration

We have explored two different cases of integrating our work in real-world prototypes for extended validation testing. First, a prototype with an integration of our modeling-extended Stixel World (slanting, interpolation, masking) for 3D scene registration was successfully tested on a moving vehicle under difficult real-world conditions in a military surveillance application. Second, stixel-based object detection was integrated in the VI-DAS system of the equally named European H2020 project, through fusion of stixels with a semantic scene segmentation calculated by a deep neural network. The functionality of our module has been verified via interfacing it with a newly developed forward-collision-warning module and associated tracking modules. This has resulted in successful forward collision warning during day and nighttime conditions within a context of partially automated driving.

1.4.5 Research Questions

The key problems addressed in this thesis are freespace segmentation and collision warning. The tools exploited for solving these problems are computer vision techniques such as disparity estimation, machine learning and probabilistic modeling. The following questions define the essential issues that are investigated in the succeeding chapters. The final contributions of this thesis include validating a selected set of algorithms in a real practical setting with moving prototype vehicles and real-time constraints. This validation part is reflected in a separate research question.

RQ1 Improvement of the performance of freespace segmentation systems under adverse imaging conditions and their robustness towards changing conditions and environments.

- RQ1a: *What are the common artifacts in current freespace algorithms and what is their root cause?*
- RQ1b: *In what way can different data modalities from a stereo camera be jointly leveraged in this context?*
- RQ1c: *How can color models be extended for freespace segmentation, while retaining a low complexity?*
- RQ1d: *What is the added value of self-supervised online learning for increasing robustness?*

RQ2 Leverage of computer vision to improve dynamic collision-warning functionality.

- RQ2a: *How can stereo disparity imaging be exploited for collision warnings?*

- RQ2b: *Is it possible to prepare an ADAS module such that future sensor fusion can be exploited beneficially?*
- RQ2c: *How should dynamic measurement data be represented efficiently for direct support to collision warning?*

RQ3 Exploration of real-time applications of AI and 3D geometry in traffic-scene and road-scene analysis.

- RQ3a: *What methods can reduce the computational requirements of neural networks to facilitate deployment in real-world systems?*
- RQ3b: *How can 3D scene geometry be modeled efficiently and accurately, to make it suitable for real-time synthetic view rendering?*

1.5 Thesis outline

This section provides an overview of the contents of this thesis. It describes how the different chapters are connected, which problems and approaches are addressed per chapter. Figure 1.2 portrays a graphical representation of the structure of this thesis. Besides the indicated topics per chapter, the figure also highlights (in yellow and red colored blocks) two research directions regarding the contributions of this thesis across different chapters.

Chapter 2 presents the color-extended Stixel World algorithm as our first new contribution. The original Stixel World algorithm provides a compact medium-level geometric representation of traffic scenes, which is calculated efficiently for real-world applicability. In order to reduce the number of false stixels, the approach in this chapter is to fuse color into the algorithm to improve the system robustness. To preserve the efficiency of the method, the proposed extension is based on efficient color modeling. Simultaneously, the system needs to operate robustly under both normal and adverse imaging conditions. We propose to rely on online modeling to comply with both goals, and evaluate this strategy on EHV-road (2014), a newly recorded and annotated dataset. The contributions of this chapter were presented at IEEE ITSC 2014 [33].

Chapter 3 introduces a self-supervised color-based Stixel World algorithm. The objective of the research in this chapter is to reduce the latency of the free-space segmentation system, while maintaining similar performance compared to the color-extended Stixel World algorithm. The added value of the proposed method, specifically under poor imaging conditions, is evaluated on a new subset of EHV-road (2015), containing only dark and rainy frames. The contributions of this chapter were presented at IEEE ITSC 2015 [34], IEEE/RSJ IROS-PPNIV 2015 [35] and NCCV 2015 [36].

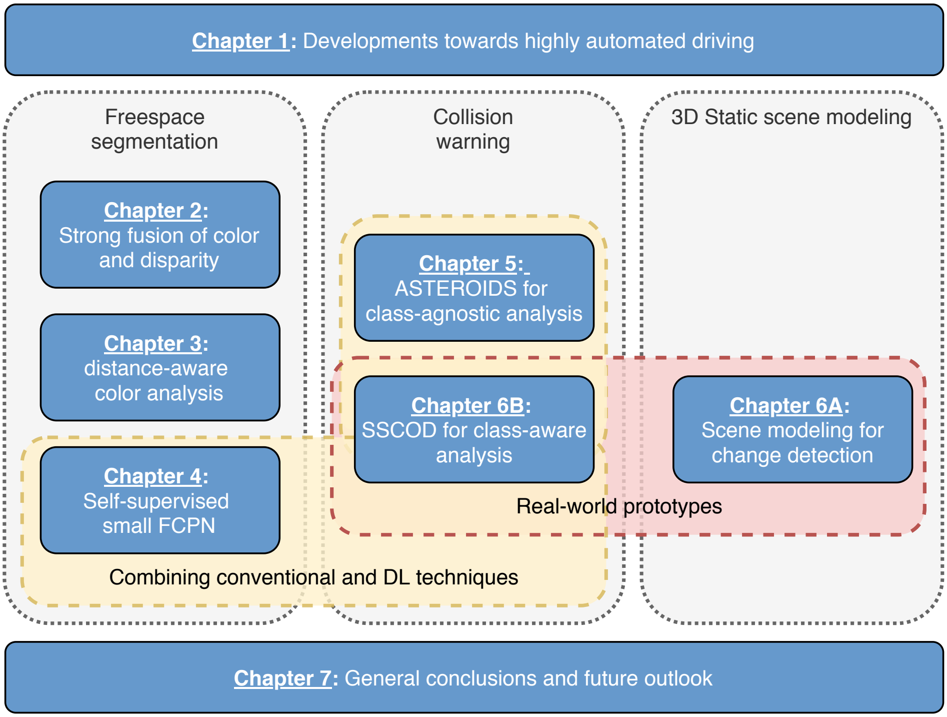

Chapter 1: Developments towards highly automated driving
Freespace segmentation
Collision warning
3D Static scene modeling
Chapter 2: Strong fusion of color and disparity
Chapter 3: distance-aware color analysis
Chapter 4: Self-supervised small FCPN
Chapter 5: ASTEROIDS for class-agnostic analysis
Chapter 6B: SSCOD for class-aware analysis
Chapter 6A: Scene modeling for change detection
Real-world prototypes
Combining conventional and DL techniques
Chapter 7: General conclusions and future outlook

Figure 1.2 — *Schematic overview of the structure of this thesis.*

Chapter 4 continues and completes the research on freespace segmentation using stereo vision. An in-depth analysis of the evaluated color models and algorithm setting used in the previous chapter shows that the performance of the system can benefit from a more adaptive color modeling method. As a solution, this chapter exploits the power of convolutional neural networks to increase the adaptivity of the color modeling in the freespace segmentation framework. The proposed method relies on the EHV-road 2014 and 2015 subsets for all training activity, and is evaluated on a new subset of EHV-road (2017). The contributions of this chapter were presented at IEEE IVS-DD [37], NCCV 2016 [38] and IS&T EI-AVM 2017 [39].

Chapter 5 presents research on stereo-based collision analysis. Since traffic is a highly dynamic environment, this chapter extends the analysis to include the dynamic aspects of traffic, and discusses a generic collision warning system that does not rely on pretraining on object classes or traffic scenarios. To this end, a probabilistic framework is proposed that combines the efficient stixel representation with flow measurements from a neural network and then generates newly introduced asteroid particles to model potential collisions in a specialized state space. The contributions of this chapter were presented at IS&T EI-AVM 2019 [40] (receiving a best-paper award) and were published in IEEE Trans.-IV [41].

Chapter 6 discusses the integration of the work into two different real-world prototypes (Change Detection 2.0 and the demonstrator of the EU H2020 project VI-DAS), for which two different extended versions of the Stixel World algorithm were developed. The first demonstrator consists of the change detection system for the Netherlands Ministry of Defence, which is used in a military surveillance context and requires pixel-accurate image registration with large viewpoint differences. The proposed solution extends the analysis of the scene content to include obstacle regions. To this end, the Stixel World representation is enriched with stixel slanting around the vertical axis, stixel interpolation and pixel masking, to enhance image registration. This work was presented in a joint effort with dr. D.W.J.M. van de Wouw at VISAPP 2018 [42]. The VI-DAS demonstrator concerns regular, yet dynamic traffic scenarios in which specific types of traffic participants need to be detected, classified, localized and tracked. This information can then be used both in a high-level risk-assessment system and for new, light-weight forward-collision warning. This pipeline uses disparity stixels for geometry assessment, a neural network for semantic information and clustering for object-level analysis.

Chapter 7 summarizes the main conclusions from all chapters, provides a discussion on the posed research questions and presents a future outlook.

Chapter 2 Freespace segmentation with the Color-extended Stixel World algorithm

2.1 Introduction

The technology overview in Chapter 1 discusses that ADAS are receiving an increasing amount of attention in research and that several systems are being employed in commercially available vehicles. ADAS offer situational awareness in a wide range of sub-functionalities, such as lane-departure warning, lane-keep assist, pedestrian detection, traffic-sign recognition and many others. The first part of this thesis focuses solely on the fundamental issue of determining where the car can or cannot drive, by splitting the scene into freespace/traversable regions versus obstacle/non-traversable regions. This distinction can facilitate subsequent processes, *i.e.*, full semantic scene parsing, lane detection and short-term vehicle routing. Moreover, this work concentrates on stereo vision-based systems for freespace detection. Stereo cameras provide dense scene information in front of the vehicle in a cost-effective way, concerning both appearance (by means of color) and 3D scene geometry (by means of disparity). Both color [43]–[45], disparity [46], [47] and their combination [48]–[50] have been employed in related work on freespace segmentation. This active line of research has still several open challenges, of which the most important ones are listed below.

- *Issues with disparity signals*: Disparity measurements generally suffer from errors such as noise, strong outliers and holes due to occlusions, or due to little texture information in large image regions. Although these issues can be addressed to a certain extent by using high-quality cameras and more advanced disparity estimation, they can never be fully resolved, since traffic scenes will often contain image areas with, for instance, low illumination, shadows, sunny reflections or motion blur.
- *Signal fusion*: It is evident that the fusion of the estimated disparity with other modalities is advantageous for obtaining more reliable information in case of difficult imaging conditions. Optical flow and texture analysis typically suffer from the same challenges as disparity estimation, as all require well-textured image regions. More orthogonal and complementary image modalities are therefore color, shape, and appearance.

The work in this chapter has been published at IEEE ITSC 2014 [33].

As with all ADAS, freespace segmentation should be able to execute as a real-time function in a vehicle. Making algorithmic contributions suited for a practical application leads to several design constraints. For instance, the system should be:

- of low complexity to allow for real-time execution on an embedded platform with limited computation and memory budget;
- able to handle the varying imaging conditions mentioned above;
- safe in the sense that it neither misses obstacles nor detects too many.

An established and efficient work that provides freespace segmentation is the so-called disparity Stixel World algorithm [51], which performs a stereo-based scene-geometry analysis. It efficiently generates a compact 3D representation of a scene and distinguishes ground from obstacles. Since the original method relies on disparity alone, the output generally suffers from created artifacts on noisy disparity data. However, an interesting property of the Stixel World method is that it allows for strong fusion of different modalities in one probabilistic framework. More specifically, instead of analyzing each modality separately and then combining their results [48], [50] (*i.e. weak* fusion), the Stixel World method can be extended to efficiently analyze multiple modalities simultaneously (*i.e. strong* fusion). To our knowledge, this data fusion property of the Stixel World algorithm has received little attention in literature, despite its apparent advantage. The work presented in this chapter aims to address and contribute to these fusion properties.

More broadly, advantages of the combination of using depth information with color information has been shown in other frameworks [48]–[50]. A particular interesting strategy is to use the dense disparity-based depth information, to learn a color-based road-versus-obstacle model online (while driving) and thus, in a self-supervised manner [50], [52]. This model is used to classify image regions as either road or obstacle, of which the result is then combined with the disparity-based analysis. This combination is typically performed with rather straightforward fusion methods, and can be as simple as using the depth analysis up to a distance from the vehicle and the color-based analysis after this distance [53].

Considering this compact overview of challenges, constraints and alternative methods, the work in this chapter will contribute to the field of freespace segmentation in the following three ways.

- A novel Color-extended Stixel World algorithm is introduced, that allows to exploit strong fusion of disparity and color modalities for freespace-versus-obstacle image segmentation. The main objective of this strategy is reducing the impact of disparity artifacts caused by difficult lighting conditions.

- A simple and efficient color model is proposed for fusion to enable a low-complexity system. To ensure that the simple color model still provides relevant information to the analysis, our system updates the class color models online (while driving) in a self-supervised mode.

- A new public dataset is released with a combination of aspects that did not exist earlier: it has (1) stereo color-video sequences, (2) freespace annotations, (3) both good and adverse imaging conditions.

The remainder of this chapter is structured as follows. First, a short description of the disparity Stixel World algorithm is provided in Section 2.2, since it serves as a basis of our work. The main contribution is put forward in Section 2.3, where the strategy to fuse color information into an extended stixel framework is described. Section 2.4 elaborates on the evaluation approach, including the publicly available dataset and experiments, the results of which are provided in Section 2.5. Lastly, conclusions are presented in Section 2.6.

2.2 Related work: the disparity Stixel World algorithm

This section gives a short overview of the Stixel World framework as presented in [51], which is used as a basis of the research work in this chapter. The main objective of stixel segmentation is to find the optimal labeling L^* of vertically stacked, piecewise planar ground or obstacle segments for the input disparity data $\mathbb{D}$. The concept is illustrated in Figure 2.1. Formally, finding L^* can be formulated as a MAP estimation problem, as specified in Equation (2.1):

$$L^* = \arg\max_{L \in \mathbb{L}} P(L|\mathbb{D}), \tag{2.1}$$

which can be solved efficiently using Dynamic Programming. Using Bayes' theorem and assuming that (a) columns are independent, (b) disparity measurements $d_{u,v} \in \mathbb{D}$ at individual pixels (u, v) are statistically independent and (c) data within disparity column D_u is independent from the labeling in other columns, the posterior probability can be written as in Equation (2.2):

$$P(L|\mathbb{D}) \sim \prod_{u=0}^{w-1} P(D_u|L_u) \cdot P(L_u). \tag{2.2}$$

Here, u is the column index and w the image width. The probability $P(L_u)$ models a-priori world knowledge to constrain the labeling to avoid dispensable segments and physically unlikely situations. This world model offers a way to regularize the results for semi-global optimality (namely, at the level of image columns). Since this world model is unrelated to the way the data analysis within the Stixel World method is handled, our environment model is based on the same conditions as laid out in [51], which contains the full details concerning $P(L)$. Finally, the likelihood of the data given a certain labeling, can be written as

$$P(D_u|L_u) \sim \prod_{n=1}^{N_u} \prod_{v=v_n^b}^{v_n^t} P(d_v|s_n, v), \tag{2.3}$$

where n is the segment index, N_u the number of segments in L_u, and v_n^b and v_n^t the bottom and top row-index of stixel segment s_n that has a binary label $l_n \in \{g, o\}$, representing the ground and obstacle classes, respectively.

The next step is accounting for invalid disparity measurements $d_v \notin [d_{\text{min}}, d_{\text{max}}]$. For example, these will occur if the estimator cannot find a match in the stereo frames. To this end, a probability of encountering a non-valid measurement is defined, p_{invalid}, as well as the probabilities that such a pixel will represent either ground or obstacle, $p(l_n|\text{inv.disp.})$. With this, the probability of invalid data for each class can be calculated using Bayes' rule: $p_{\text{invalid}}^{l_n} = p(\text{inv.disp.}|l_n) = p(l_n|\text{inv.disp.}) \cdot p_{\text{invalid}}/p(l_n)$. This then leads to

$$P(d_v|s_n, v) = \begin{cases} P_D(d_v|s_n, v) \cdot (1 - p_{\text{invalid}}^{l_n}) & \text{for valid } d_v, \\ p_{\text{invalid}}^{l_n} & \text{otherwise.} \end{cases} \tag{2.4}$$

Here, $P_D(d_v|s_n, v)$ represents the probability of a single valid disparity measurement d_v at a certain row v, assuming that it would belong to a potential segment s_n. The distribution $P_D(d_v|s_n, v)$ is modeled as a mixture model that consists of a uniform distribution to handle outliers and a Gaussian distribution to model how well the measurement fits the potential segment:

$$P_D(d_v|s_n, v) = \frac{p_{\text{out}}}{d_{\text{max}} - d_{\text{min}}} + \frac{1 - p_{\text{out}}}{A_{\text{norm}}} e^{-\frac{1}{2}\left(\frac{d_v - f_n(v)}{\sigma^{l_n}(f_n, v)}\right)^2}. \tag{2.5}$$

In Equation (2.5), p_{out} is the fixed probability of encountering an outlier. The normalization term A_{norm} and the modeled standard deviation σ^{l_n} are defined in [51]. The remaining term, $f_n(v)$, models the expected disparity within a segment for ground and object segments. For objects, $f_n^o(v) = \mu_n$ is adopted, assuming a fronto-parallel object surface at the mean disparity of the segment. For ground segments, $f_n^g(v) = \alpha \cdot (v_{\text{horizon}} - v)$ is used, assuming a linear groundplane surface with a slope α.

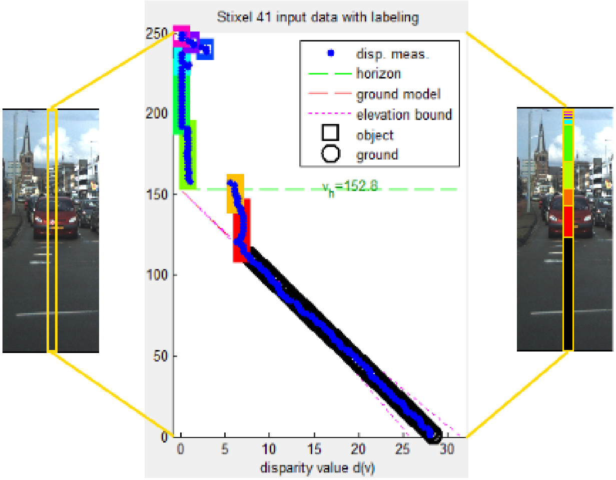


Figure 2.1 — *Illustration of the stixel representation of a traffic scene. The left image is an example of a cropped input image (of the left camera from a stereo pair) with a box indicating a stixel column to be processed. The graph in the middle shows the corresponding measured disparity in blue dots (being the input to the Stixel World algorithm), and the resulting stixel output plotted on top of that, using a unique color per segment. The right figure visualizes the same result as an overlay on the input image for clarification. The region marked black is correctly detected ground, obstacle stixels are colored by distance from the ego-vehicle (red is closeby, green far away). These results show some small inaccuracies (the car is divided into two segments and false disparity measurements in the sky lead to small stixels at the top region of the image).*

2.3 Method: the Color-extended Stixel World algorithm

This section describes the proposed method, called the Color-extended Stixel World. The first sub-section addresses the general algorithmic design. Then, two consecutive sub-sections describe the applied color representation and the online-training strategy in more detail.

2.3.1 Extending the Stixel World with color analysis

As a key contribution, we incorporate a color signal $\mathbb{C}$ in the original Stixel World model. To this end, the data term of Equations (2.1) and (2.4) should now reflect both color and disparity information. Starting from $P(L|\mathbb{D}, \mathbb{C})$, a derivation can be made analogous to the description in Section 2.2. With the additional assumption that disparity and color modalities are independent, the data term of the likelihood can be rewritten as:

$$P(D_u, C_u|L_u) \sim \prod_{n=1}^{N_u} \prod_{v=v_n^b}^{v_n^t} P(d_v|s_n, v) \cdot P(c_v|s_n, v), \tag{2.6}$$

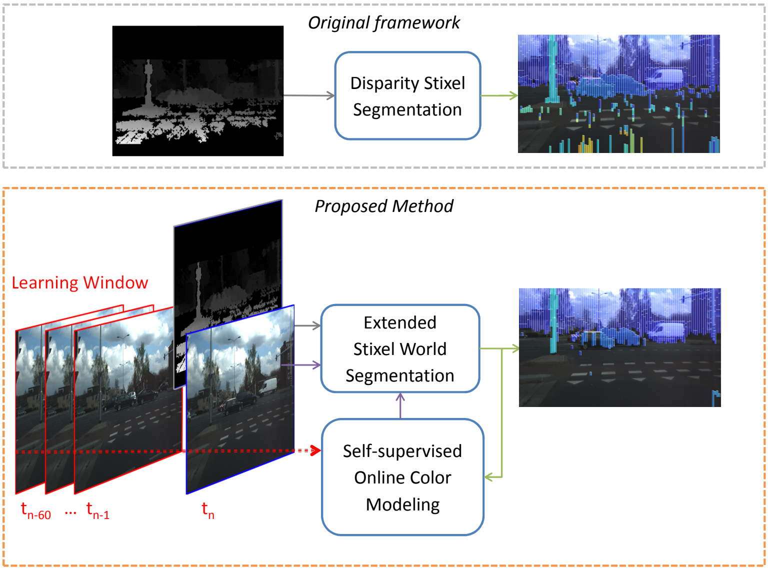

Original framework
Disparity Stixel Segmentation
Proposed Method
Learning Window
Extended Stixel World Segmentation
Self-supervised Online Color Modeling
t_{n-60} ... t_{n-1}
t_n

Figure 2.2 — *Original stixel framework (top), relying on disparity images alone, versus the proposed extension (bottom), which exploits both disparity and color information.*

where segment s_n has a label $l_n \in \{g, o\}$, as before. Note that the term $P(d_v|s_n, v)$ also incorporates the construction for invalid disparity measurements as in Equation (2.4) but is left out here for compactness. Furthermore, we do not alter the definition of the world model $P(L)$, but focus on defining a suitable color model within $P(c_v|s_n, v)$. This term should capture the probability of a certain color measurement given a certain segment label. We define this to be independent of the position v of the segment and merely consider the label of a segment, so that $P(c_v|s_n, v) = P(c_v|l_n)$. This is a reasonable simplification, since $P(L)$ already constrains physically unlikely segmentations and we can assume that the color of the road surface is approximately constant within the image.

The Color-extended Stixel World optimization analyzes a log-likelihood cost function based on $P(L|\mathbb{D})$. In that cost function, we impose weighting factors between the cost of the disparity term $P(d_v|s_n, v)$ and that of the color term $P(c_v|s_n, v)$. These weighting factors are selected to set a balancing ratio of $1 : \lambda$ between disparity-based cost and color-based cost. This balancing is required to compensate for the oversimplification in the modeling of their theoretical joint probability density in two disjoint terms, which use different probabilistic methods (i.e. Gaussian distributions and histograms). The joint probability density function of color and disparity cannot be computed in practice, and the ratio provides a convenient means for tuning the normalization of the simplified model.

The main steps of our processing framework are conceptually depicted in

Fig. 2.2 and Algorithm 1. It comprises of two main steps: learning color models for freespace and obstacle areas (Algorithm 1: [Learn Color Models]) and segmenting the current frame using the Color-extended Stixel World method (Algorithm 1: [Process Current Frame]), both addressed in the following subsections.

Algorithm 1 Segmentation in the Color-extended Stixel World

```
Input: image $I_{t_n}$; disparity $\mathbb{D}$; learning window $LW$;

  [Learn Color Models]
  for each $t \in LW$ do
      $C_t \leftarrow$ TransformRGB2Color($I_t$)
      for $l \in \{ground, obstacle\}$ do
          $TM_t^l \leftarrow$ GenerateTrainingMask($L_t^*, TM_{\text{prior}}^l$)
          $X_t^l \leftarrow$ ExtractSamples($C_t, TM_t^l$)
          $H_{t_0}^l \leftarrow$ AddToHistogram($H_{t_0}^l, X_t^l$)
      end for
  end for
  $P(C|l) \leftarrow$ NormalizeHistogram($H_{t_0}^l$)

  [Process Current Frame]
  $\mathbb{C} \leftarrow$ TransformRGB2Color($I_{t_0}$)
  $L^* \leftarrow$ StixelSegmentation($\mathbb{D}, \mathbb{C}, P(C|l)$)
Output: Optimal Labeling $L^*$
```

2.3.2 Color representation with an adaptive palette

A common approach in color analysis is employing color histograms as dense area descriptors. Color histograms can be defined with linear, non-linear or adaptive binning strategies. We apply the adaptive binning strategy *minimum-variance quantization*, also known as *median-cut quantization*, as described in [54], which is referred to as an *indexed* color space. Indexing the color space with an adapted color palette ensures that the borders of the histogram bins fit optimally to the color signal of a certain traffic scene in an efficient way, see Figure 2.3. This strategy will be compared to relying on unadapted, linearly spaced bins in our experiments.

Our analysis will mainly focus on the RGB color space, since it yields good results in numerous color-based experiments on video analysis. This approach will be compared to an HS-based approach in several tests as well (since we do not use the intensity signal, the commonly known HSI space becomes HS in our case). The function that transforms each RGB image frame I_t to its desired color representation in $\mathbb{C}_t$, is indicated with 'TransformRGB2Color' in Algorithm 1.

2.3.3 Self-supervised online learning of color models

Since the aim of this work is to develop a system that is highly adaptive to different traffic environments, the color models $P(c|l)$ are learned online, which means

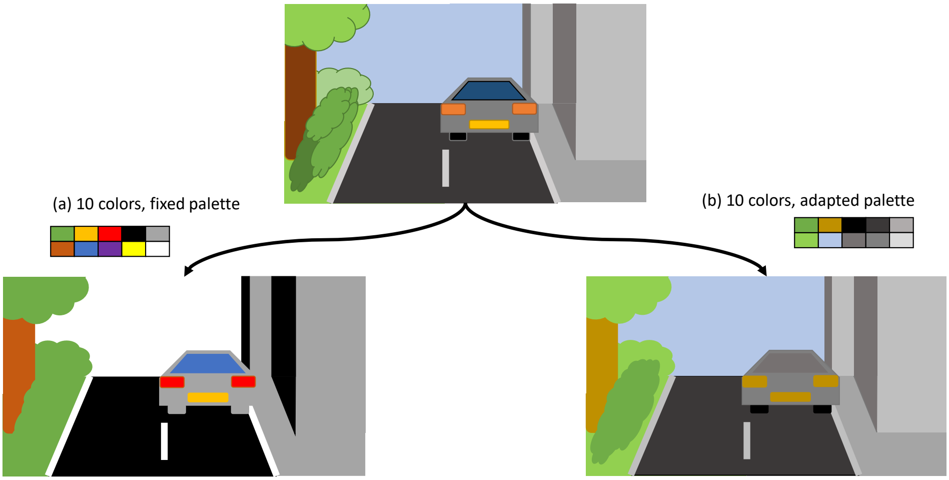


Figure 2.3 — *Conceptual illustration of the benefits of using an adaptive color palette. Since the input image (top) contains many gray tones, quantizing the image with a fixed palette (a) hampers discerning important different image regions (e.g. the car, buildings and sidewalk are merged, grass and bushes are merged, etc.). By adapting the palette towards different gray and green tones (b), relevant information is retained, while a similar color space reduction is achieved.*

that they are updated while driving. This is an intuitive approach, since an offline learning strategy would require a single color model that is both general enough to be applicable to all potential ground appearances and simultaneously discriminative enough to always separate the ground from its surroundings. Our online learning approach is indicated in Algorithm 1 under the labeling text 'Learn Color Models'. The training process contains two key selection strategies: first selecting appropriate frames and within those, selecting appropriate pixels. These steps are defined in the next paragraphs below.

A. Defining the learning window over frames

In our framework, a learning window LW is defined, containing one or more frames that precede the current frame at $t = t_n$ with a maximum range of 60 frames back in time, denoted by t_{n-60}. These frames are transformed to the indexed color space. From this signal, training samples are selected that are considered to be representative of either the road or the obstacle class. These samples are then used to fill and normalize a color histogram for each class, providing the required $P(c_v|l_n)$. Note that the color model $P(c|l)$ is not yet learned at the start of a sequence and hence, assumed to be a uniform distribution. As a consequence, the first frames are effectively segmented using only their disparity signal.

B. Defining the training mask over pixels

Selecting ground and obstacle training samples from the preceding frames I_t within learning window LW requires a *training mask*, denoted by TM_t^l, for each frame and each class $l \in \{g, o\}$. To this end, we exploit the fact that each previous

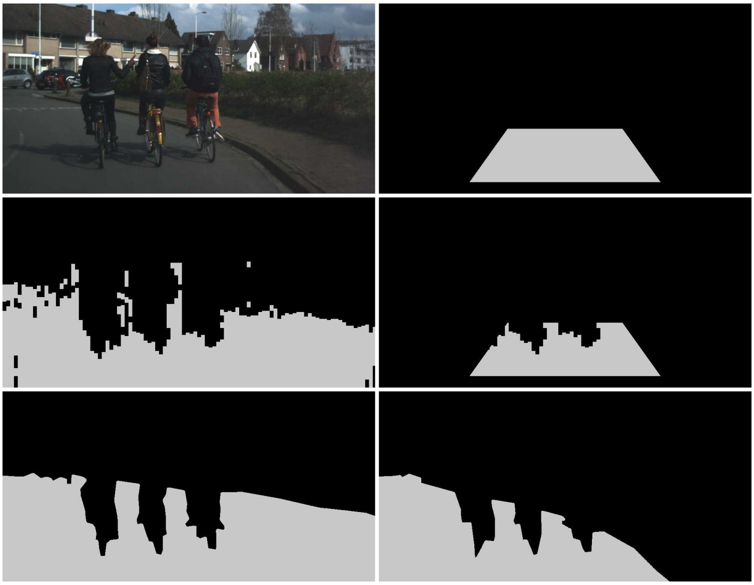


Figure 2.4 — *Different training masks and ground-truth annotations. Top: example image (left) and TM^g_{prior} (right); middle: ground segmentation mask (left) and its intersection with the prior. Bottom row: annotation of* drivable surface *(left) and* road *(right).*

frame was already analyzed and segmented by our system at its corresponding time t. This process results in an estimate of obstacle and ground areas in each frame, which we refer to as segmentation masks (*SegmMask*).

We explore two strategies to create training masks, namely using the full mask (*SegmMask*), or intersecting the mask with a prior mask (*Intersect*). For ground samples, this is illustrated in Fig. 2.4 with an example image (top left) and the corresponding estimation of the ground area, as provided by its disparity-based segmentation result (Fig. 2.4, middle left). We define the prior mask (TM^g_{prior}) as a fixed trapezoid at the bottom center of the image mask (Fig. 2.4, top right). Intersecting this prior mask with the disparity segmentation result, leads to a mask that contains the road area directly in front of the car, excluding detected obstacles (Fig. 2.4, middle right).

A comparable strategy is employed to generate a *training mask* to extract obstacle training samples. The *SegmMask* is the inverted version of the mask for ground (i.e., the black regions in Fig. 2.4 middle-left). For the *Intersect*, we apply a mask containing the area below the horizon. This makes the color modeling of obstacles more balanced towards obstacles that are on the road, which are more relevant to distinguish than, for instance, tree leafs or rooftops.

2.4 Evaluation approach

The objective of the evaluation is to measure the quality of the proposed freespace segmentation algorithm and the impact of three crucial design choices. These choices are: (a) the color-space specifications, (b) the strategy of selecting training samples with training masks TM_t^g and TM_t^o, and (c) the position and range of the learning window LW. All our results will be compared to results that are acquired with the our implementation of the disparity Stixel World algorithm [51], which was upgraded with several enhancements, as will be explained in Section 2.4.3.

2.4.1 Dataset

To evaluate the proposed system, a dataset was acquired in an urban environment, using a BumbleBee2 camera, mounted behind the windshield of a car, just below the rear-view mirror. The camera has a baseline of 12 cm, a resolution of 1024×768 pixels and a frame rate of 20 fps. This EHV-road14 dataset is publicly available online [1].

From the recorded data, 74 representative frames have been selected and manually annotated with both road and drivable surface areas, as illustrated in the bottom row of Fig. 2.4. The frames contain a large variety of relevant traffic situations, such as small, crowded streets with cyclists, road repair sites, large crossings and highways. The set contains asphalt and paved roads of several colors (black, gray, red), frames with low illumination due to heavily clouded skies or trees, and frames with high illumination from clear sky with sunny reflections. Several example frames are provided in Fig. 2.5.

Unfortunately, our algorithm cannot be executed on benchmarks such as the KITTI dataset [55], since they do not contain the preceding frames of annotated road images that are required for our online training strategy.

2.4.2 Disparity estimation

To obtain disparity measurements, a multi-threaded version of the OpenCV implementation of the Semi Global Block Matching algorithm of [56] was applied. Due to the many low-texture image regions in our dataset, we have empirically found that a matching window size of 7×7 pixels and smoothing parameters $p_1 = 16 \cdot (7 \times 7)$ and $p_2 = 8 \cdot p_1$ provide the most acceptable results. Additionally, a *winner margin* measure was exploited, to force the algorithm to have a higher precision at the cost of recall. This is beneficiary for the baseline Stixel World method, since it can handle missing values better than erroneous ones. This can be seen as a simplification of the work presented in [57], in which disparity estimates are accompanied by a confidence measure to adaptively set an outlier probability. In our approach, this confidence is binary with a relatively strict threshold, so that the *winner margin* is at least 20.

[1] The data can be found at `http://tue-mps.org`.

2.4.3 Stixel World parameters

As described, our camera has a lower resolution and a smaller baseline than, for example, the camera used for the KITTI benchmark dataset [58], potentially resulting in a lower quality of the disparity estimates in our dataset. To compensate for this deficiency and to obtain more favorable results for the baseline method, we have made several improvements to the baseline framework. For instance, the geometric groundplane model $f_n^g(v)$ is updated while driving, instead of using a single fixed model as done by the authors of the original work. To this end, we exploit a v-disparity representation such as in [47], for several vertical slices of each frame, making our system more robust against groundplane deviations over time and non-horizontal groundplanes. Moreover, we have tuned the label-based transition probabilities defined in $P(L)$ to boost the performance of the baseline method even further. Finally, we have added an artificial ground segment to the bottom of each stixel column, denoted with *Seg0*. This segment represents the area below the camera view, which can safely be assumed to be road in this context, while it reduces false detections in the lower image regions, due to noisy disparity estimates. To show the value of these additions, the performance is reported of each of these three disparity baselines (original settings, tuned transition probabilities and with the artificial *Seg0* as ground).

The relevant Stixel World parameters, as described in Section 2.2, are set as follows throughout all experiments: $p_{\text{out}} = 0.25$; $p_{\text{invalid}} = 0.25$; $p_g^{\text{invalid}} = 0.55$; $p_o^{\text{invalid}} = 0.45$; $p_g = p_o = 0.5$; $d_{\text{min}} = 1$; $d_{\text{max}} = 32$. Furthermore, a stixel width of 10 image columns is adopted, and the disparity and color signals are subsampled vertically with a factor of 3, prior to segmentation. Note that since the full-resolution image data are exploited to compute look-up tables and color models, the subsampling strategy is comparable to the approach in [51]. The research version of the Color-extended Stixel World method is a MATLAB-based implementation. The additional complexity of the color processing is quite small, compared to the complexity of the disparity-analysis baseline. Therefore, it is safe to assume that the proposed extension can operate as a real-time system, similar to the original system [51].

2.4.4 Metrics

The performance of the Color-extended Stixel World algorithm is measured in two distinct aspects to provide a balanced view between data and application. These aspects are based on raw pixel count and on the measured real-world drivable corridor. Both aspects are discussed separately in the next subsections.

A. Pixel-level metric

Inspired by the work of Fritsch *et al.* on performance metrics for road detection algorithms [55], our road detection algorithm is evaluated in a Bird's Eye View (BEV) representation of the scene. A BEV representation is corrected for geometric distortion to avoid that pixels near the car outweigh pixels farther away in the segmentation score. In this representation, we have employed several metrics to assess the performance of our algorithm. First of all, the recall and precision of the road area are measured, with the road annotation as a reference. However, pixels that are drivable but not belong to the road are ignored in the evaluation, such as curbs and grass. The purpose of this strategy is to assess in which areas improvements or deteriorations of the results appear. More specifically, we focus our evaluation on increasing the recall of the road regions, since that is the most relevant ground area for ADAS. As a side-effect, it is acceptable to reduce the precision as long as that occurs mostly in drivable surfaces that are not road. In this evaluation, an area up to 30 meters in front of the vehicle is considered.

Since the stixel representation approximates area contours with rectangular shapes, it is not possible to achieve a pixel-accurate segmentation. Consequently, achieving perfect recall and precision is also not realistic. To determine how close the performance of the tested methods come to the maximum attainable performance, we estimate realistic optimal recall and precision scores by eroding and dilating the ground-truth segmentation masks with a square kernel with dimensions similar to the stixel width.

B. Drivable corridor metric

Next to the pixel-level metric which is vision-inspired, the added value of our method is assessed bearing a practical application in mind, namely actually measuring where the ego-vehicle can drive. This can also be measured in the BEV representation of our road-segmentation results. In that representation, we derive how far a vehicle can drive by calculating where the first object is that a vehicle of average width would drive into. Using the ground-truth annotations, it is possible to define recall and precision scores, which incorporate a 5-meter safety margin around obstacles. Recall indicates how much of the ground-truth drivable distance is detected correctly. The recall will be lower than unity when a false obstacle is detected in front of the first real obstacle. The precision represents how much of the detected drivable distance is correct. If the real obstacle is missed, the precision will be lower than unity. Consequently, for each frame, either the recall or precision of the drivable distance is always equal to unity. Namely, since the metric on the binary segmentation mask starts from the detected freespace in front of the vehicle and evaluates an uninterrupted distance towards the first detected obstacle, it can only result in a distance that is (1) a certain amount too short, (2) perfect or (3) a certain amount too large. The drivable-corridor evaluation is performed over a range of up to 50 meters. For these metrics, the resulting F-score is provided as well, which is defined as the harmonic mean of recall and precision.

2.4.5 Experiment design

This section provides a brief overview of the experiments that were performed to assess the critical design choices of our system. The three choices of interest are (1) selection of the color representation, (2) selection of the training mask strategy, and (3) specification of the learning window.

A. Evaluating color representations

Several different color spaces and settings were evaluated. Specifically, we vary the number of clusters k that is used to approximate the colors in the current learning window and the color-data weighing factor λ, for both indexed RGB and linearly binned HS representations. For linear binning, this is translated into k bins per dimension, resulting in k^2 bins for HS.

B. Evaluating training masks

To evaluate our choice of *training mask* (TM_t^l) for the selection of training samples, the two approaches described in Section 2.3.3 are compared. This involves using the segmentation mask alone, or intersecting it with a fixed, a-priori defined trapezoid in front of the vehicle (for ground samples), or the area below the horizon (for obstacle samples).

C. Evaluating learning windows

The most relevant settings of the learning window are its range and position with respect to the new frame. First, the time-length of the learning window is varied by extending it further into the past with a maximum of 60 frames earlier than the current frame (equivalent to 3 seconds in the past). This sub-experiment validates if the added complexity of taking more frames into account translates into more robustness. Next, it is analyzed whether it is possible to leave a gap between the frames in the learning window and the frame currently analyzed. This is an important sub-experiment, since if there is more time available to analyze the frames of the learning window, either the constraints on execution time can be relaxed, or more complex algorithms can be employed. As a third sub-experiment on the learning window parameters, the duration of the LW is limited to a single frame and then its position is varied. Effectively, this combines the extreme cases of the first experiment (varying the time-length) with the idea of the second (leaving a gap between LW and the current frame). In the most extreme case of this third experiment, a color model is learned on a single frame, 60 frames earlier in time (corresponding to 3 seconds).

2.5 Results

The upcoming sections contain the results of the evaluation of our Color-extended Stixel World method. Fig. 2.5 shows four representative positive qualitative results and illustrates the pixel-level road-segmentation scoring metric.

A. Performance of color representations

The most relevant results of the experiments comparing different color spaces and settings are provided in Fig. 2.6 and Table 2.1. Based on the results illustrated in Fig. 2.6, the HS color representation (bright green pluses) performs better at increasing the recall, while RGB (cyan pluses) tends to improve the precision of

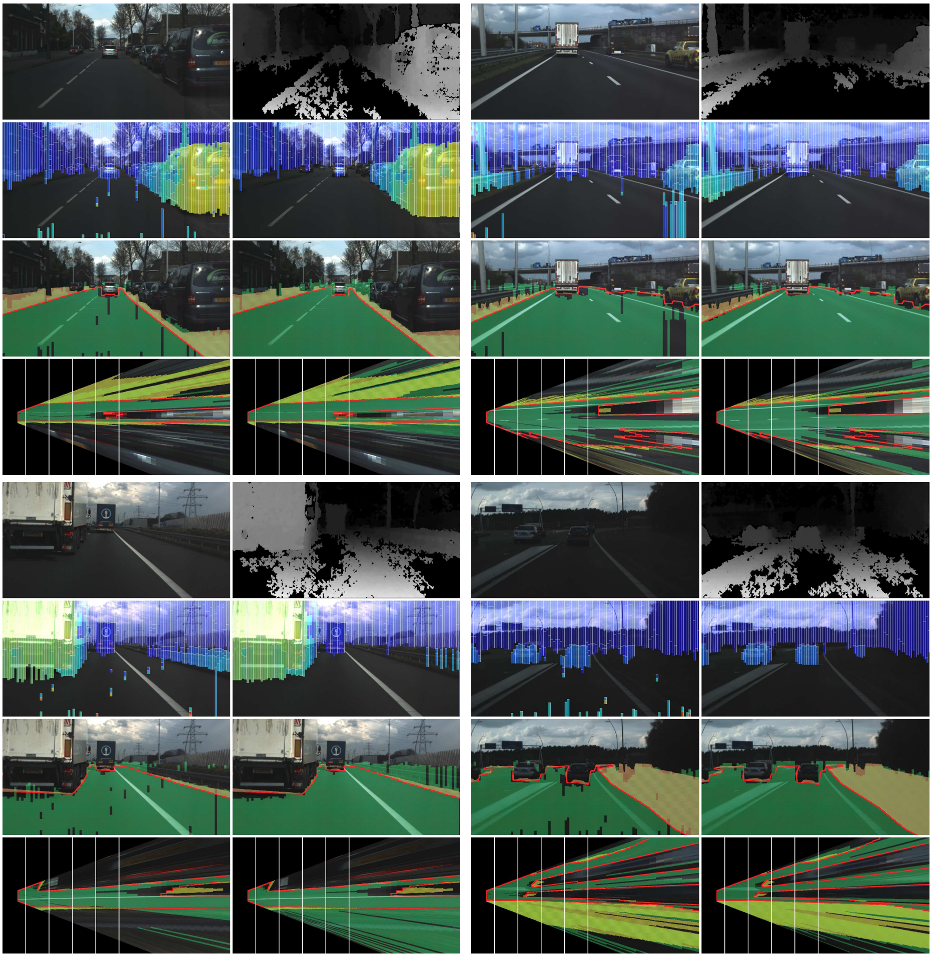


Figure 2.5 — *Qualitative results illustrated with eight images per frame. Top row: left camera image (rectified and cropped) and corresponding disparity image. Other rows: baseline result (left) and ours (right), in three different visualizations. First, the Stixel overlay (color depicts distance: red (close) to blue (far)). Second, an overlay of the ground mask (in green) with the road ground-truth outlined in red, and ignored pixels that are drivable but not road in orange. Third, the same masks in their BEV representation with white lines at 10, 20, 30, 40 and 50 meters.*

the road segmentation. The experiments with a high λ and a low k generally result in deterioration of the (tuned) baseline results. This is plausible, since the color representation cannot contain much discriminating information, due to the low k, while the confidence is over-increased due to the high λ, leading to erroneous results. We have obtained the best results with an indexed RGB color model, using $k = 64$ and $\lambda = 4$ (yielding an F_1-score of 0.968). The closest HS-based score (with an F_1-score of 0.967) is obtained using $k = 16$ and $\lambda = 2$, where we ignored experiments with worse precision than the baseline, since the precision in road regions is critical for safe ADAS. Although these pixel-based scores are very similar, the RGB experiment outperforms the HS experiment with 0.919 to 0.861 in the recall of the drivable distance (see Table 2.1). This means that more false obstacles are detected with the HS color space, even though the HS model uses 16×16 bins and the RGB only 64. In the top graph of Fig. 2.7, two experiments with $\lambda = 1$ are shown. The results of these are similar or worse than the (tuned) baseline method, illustrating the importance of correct normalization when fusing different signal modalities.

B. Performance of training masks

To evaluate our choice of *training mask* (TM_t^l) for the selection of training samples, we compare the two approaches described in Section 2.3.3. This involves using the segmentation mask alone, or intersecting it with a fixed, a-priori defined trapezoid in front of the vehicle (for ground samples) or the area below the horizon (for obstacle samples). In the bottom-left graph of Fig. 2.7, the results of these strategies are shown, using the best settings for RGB and HS, as found in Section 2.3.2. For the recall of the drivable distance alone, the selection of the training mask has little influence, since all blue and green graphs nearly overlap. However, in the two-dimensional recall-precision plot of Fig. 2.6, the influence of the training mask selection is clearly visible, as the the dotted green (HS) and blue (RGB) markers separate from each other in the two-dimensional field (mostly in terms of range). The central markers in the blue and green series are the experiments with one full segmentation mask and one intersection, of which the scores are similar for both color spaces. The scores with high recall are obtained using the segmentation masks for both ground and obstacles, while the scores with high precision are obtained using the intersection masks for both classes. Hence, the system becomes more conservative (higher precision) when the color models are focused on relevant areas (the trapezoid in front of the car for road class and the area below the horizon for the obstacle class). The subsequent experiments below rely on the use of the full segmentation mask for ground samples and the intersect method for obstacle samples, since it results in the highest F_1 score and forms a natural, good compromise.

C. Performance of learning windows

The most relevant settings of the learning window are its size in frames and its position with respect to the new frame. We have found that the effects of the *LW* settings are low for both metrics. In other words, our algorithm performs similarly under different *LW* settings. In Fig. 2.6, the results are marked with magenta circles. They mostly overlap, even though the tests included several extreme cases. In the bottom-right graph of Fig. 2.7, the experiments with the 3-sec.-old frame are marked explicitly. They perform slightly worse, but still outperform the baseline with more than 25%. This signifies that the system is flexible in the selection of *LW* frames, since a single frame, selected within the last 3 seconds, can provide sufficient information to learn a reliable color model to improve the segmentation of the road area. Evidently, this is based on the assumption that the scene appearance does not change drastically in this time interval.

D. Discussion on the drivable-corridor metric

As discussed, our evaluation relied on both a pixel-level and a drivable-corridor metric. Comparing the qualitative results of the baseline and our method in Figure 2.5, clearly shows the reason why these metrics are different and are relevant. The disparity-only baseline method tends to generate false stixels at the bottom of the image, *i.e.* close to the car. These stixels are often small, so that the pixel-level metric over the entire area will not be much affected. However, the drivable corridor is cut off immediately at these false obstacles, leading to a very low recall. In a practical application, this would have the undesired effect of unnecessarily braking by the ego-vehicle. Our proposed color-extended version resolves these false detections to a great extent, thereby improving the recall of the drivable corridor (and hence, the applicability of the system). On top of this, the average precision scores of our method on the corridor metric lie between 0.993 and 1.0 over all frames, experiments and ranges. This means that our method, hardly influenced by the used settings, very rarely misses a true obstacle within 50 meters over the entire width of the corridor. Although not quantified in the current experiments, we deem it infeasible that our system would ever *not* brake for a true obstacle, since the missed detections are still at a safe distance, so that objects can and will still be detected at a later moment.

Table 2.1 — *Obtained results for road segmentation and drivable distance (expressed by F_1 score, recall and precision) for various disparity baselines and involved parameters of our method. Highest score per metric printed in bold.*

						Road Segmentation			Drivable distance ($\leq$ 50m)		
						F_1 score	Recall	Precision	F_1 score	Recall	Precision
Disparity baseline (original settings)						0.910	0.863	0.964	0.489	0.323	**1.000**
Disparity baseline (optimized settings)						0.961	0.961	0.961	0.717	0.559	**1.000**
Disparity baseline (optimized & Seg0 is ground)						0.963	0.965	0.961	0.858	0.755	0.993
Color space	LW	TMg	TMo	k	λ						
indexed RGB	10:0	SegmMask	Intersect	64	4	**0.968**	0.969	0.967	0.956	0.919	0.995
indexed RGB	10:0	SegmMask	Intersect	32	8	0.963	0.964	0.963	**0.968**	**0.946**	0.992
linear HS	10:0	SegmMask	Intersect	16x16	2	0.967	0.973	0.961	0.923	0.861	0.995
indexed RGB	10:0	SegmMask	SegmMask	64	4	0.965	**0.975**	0.956	0.958	0.925	0.995
indexed RGB	10:0	Intersect	Intersect	64	4	**0.968**	0.960	**0.976**	0.956	0.915	**1.000**
indexed RGB	60:60	SegmMask	Intersect	64	4	0.965	0.961	0.968	0.950	0.905	0.999
linear HS	60:60	SegmMask	Intersect	16x16	2	0.967	0.973	0.961	0.909	0.835	0.997

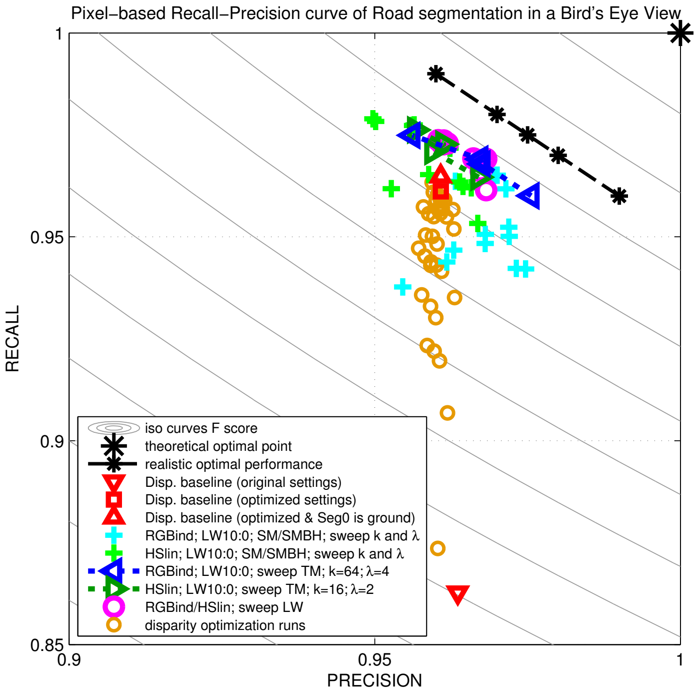


Figure 2.6 — *ROC plot of the pixel-level metric of* road, *ignoring other pixels that are drivable, such as grass and pavement (zoomed in, best viewed in color). In the legend: LW [start:end]: Learning Window with range* $[t_0 - t_{start}, t_0 - t_{end}]$*; TM: training mask, which can be* SegmMask *or* Intersect*.*

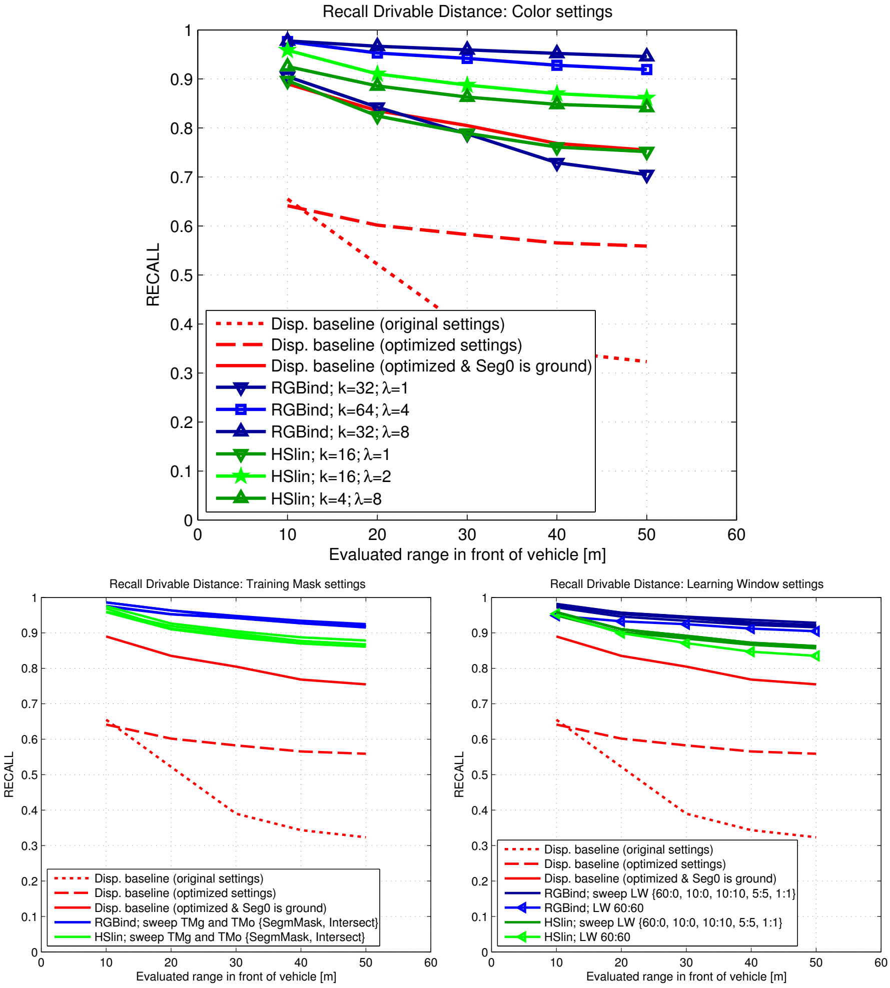


Figure 2.7 — *Recall of the drivable distance over increasing ranges. In the legend: LW [start:end]: Learning Window with range $[t_0 - t_{start}, t_0 - t_{end}]$; TM: training mask, which can be* SegmMask *or* Intersect*; k: the number of bins (for HS: k bins per dimension).*

2.6 Conclusions

In this chapter, we have presented a color extension to the disparity-based Stixel World algorithm, in order to more robustly segment available freespace versus occurring obstacles in traffic scenes, using online learned color models in a self-supervised way. This extension particularly improves the robustness of the segmentation against erroneous disparity estimates, which inevitably occur during challenging low-texture imaging situations, regardless of the quality of the stereo camera being used. An improved, safe and reliable segmentation of freespace and obstacles is obtained by fusing the disparity information with other (image) modalities, so that joint exploitation of the information is facilitated in the overall system.

The proposed method contains the following novel aspects:

- *strong data fusion* by extending the original optimization criterion with a color-based cost term;
- *color analysis* for incorporating the most suitable color feature in the cost function;
- *self-supervised online training stage* for using a simple color model which is kept representative during operational driving through different scenes;
- *sample-selection analysis* to optimally select training frames and training samples for the online training stage.

It can be concluded that the method is enhanced by adding color data with an informative and efficient representation and the self-supervised online learning of the color models. Besides these algorithmic additions, we have contributed in two other aspects. First, we present a *newly recorded dataset* to evaluate this disparity and color-based freespace segmentation systems with online learning. Second, a *new evaluation metric* is employed that measures improvements in drivable-corridor estimation, which is directly useful for practical deployment.

Experiments and evaluation of the Color-extended Stixel World method have shown the following key results.

A. Color representation: The indexed RGB color model ($k = 64$ and $\lambda = 4$; $F_1 = 0.968$) slightly outperforms the best HS-based method ($k = 16$ and $\lambda = 2$; $F_1 = 0.967$) in the pixel-based metric. Interestingly, the RGB experiment outperforms the HS-experiment with 0.919 to 0.861 in the recall of the drivable distance.

B. Balancing the fusion: Experiments with $\lambda = 1$, or high λ combined with a low k, typically perform worse than the (tuned) baseline method, illustrating the importance of appropriate normalization when fusing different signal modalities.

C. Learning window selection: A single frame, selected within the last 3 seconds, can still provide sufficient information to improve the segmentation performance with 25% compared to the disparity baseline.

D. Training mask design: The influence of the used training mask is small on the drivable-corridor metric. However, the pixel-based metric shows that the system typically becomes more conservative (higher precision) when the color models are focused on relevant image areas using the intersected masks.

The combination of these aspects results in an increased pixel-based F_1 score on road segmentation from 0.96 to 0.97, compared to a heavily optimized baseline method. Without our additional optimization efforts, the baseline method scored 0.91. More importantly, in detecting drivable distance (the novel application-inspired metric), the proposed method increases the F_1 score from 0.86 to 0.97. These results clearly indicate that the Color-extended Stixel World method, based on strong fusion of disparity and color modalities, is an accurate and robust method for road versus obstacle segmentation.

The discussed aspects in this chapter are only elements within a larger design framework for ADAS. This chapter has dominantly addressed robustness and performance quality as indicators for system improvement. For example, another crucial aspect is processing speed, since the results should become available by real-time processing in the car. However, the use of strong fusion in the work of this chapter implies that both the color and the disparity signal have to be available prior to initializing further processing. In turn, this implies a bound on the minimal latency that the system as a whole can achieve. The next two chapters aim at redesigning the data flow of our system pipeline to reduce the system latency without degrading the quality of its performance.

Chapter 3 Distance-aware color modeling for freespace segmentation

3.1 Introduction

This chapter continues further with the use of color and disparity for freespace segmentation, but without relying on a strong fusion strategy. Whereas the previous chapter has introduced the use of color into the Stixel World modeling, this chapter concentrates on optimizing the involved computational effort with the aim of reducing the latency of the complete freespace segmentation task.

Similar as in Chapter 2, the research scope is limited to stereo camera-based systems. Thereby, other issues are still a point of interest, most notably difficult but realistic imaging conditions that degrade the stereo-disparity signal (such as low light, bad weather, or a low-quality sensing system). These artifacts typically lead to the detection of false obstacles, which lower the systems applicability under real-world conditions. The previous chapter has introduced the Color-extended Stixel World algorithm to address this issue. This solution resolves many erroneous results of the disparity analysis at a low additional computational cost, in contrast to alternative solutions, such as high-quality cameras or more advanced disparity estimation techniques. However, another key property of ADAS is the system latency, *i.e.* the time delay between the moment of data acquisition and the moment of the output response. Since ADAS preferably function at high vehicle speeds, the latency of such systems should be as small as possible. Hence, any delay that can be removed from the critical path of the analysis is beneficial to the value and applicability of the system, provided that this removal does not degrade the acuracy of the results.

Naturally, there are different potential strategies to reduce the freespace segmentation latency, such as reducing the algorithmic complexity, increasing computational capacity with improved hardware, or subsampling the input data to reduce the computational load. Another option is to redesign the data flow, so that the processing time of the critical path is reduced. For example, the Stixel World pipeline that our research builds upon, consists of two main components: the disparity estimation and the Stixel World computation. Although fast disparity

The work in this chapter has been presented at IEEE ITSC 2015 [34] and IEEE/RSJ IROS-PPNIV 2015 [35].

estimation methods exist [59], [60], this typically requires either relying on suboptimal algorithms, or processing at a low resolution, or employing customized hardware that is not commonly available. Even in the state-of-the-art system presented in [51], the disparity estimation takes 40 ms per frame on a dedicated FPGA platform, whereas the stixel analysis of the data takes 30 ms, when executed on a high-quality multi-core CPU. This is a demanding requirement when considering that the overall latency of the practical system should not exceed 100-150 ms as an order of magnitude.

Theoretically, executing these steps in parallel could reduce the system latency close to a factor of two. However, the Color-extended Stixel World algorithm relies on strong fusion, which requires the disparity of the input color frame to be available at the same time for the analysis. This results in the following challenges:

- ADAS with a large system latency can limit the speed that the equipped vehicle can be safely controlled at well below an order of magnitude of 100 km/h;
- disparity estimation roughly takes up half of the computation time within the critical path of the reference system, while the system latency is upper-bounded as discussed above;
- strong fusion of disparity and color improves the systems performance, but imposes a bottleneck in reducing the system latency, since the current color modeling is not informative enough to rely on without exploiting strong fusion.

Therefore, the problem statement in this chapter concerns developing a more advanced color modeling with the aim of removing the disparity analysis from the critical system path to reduce the system latency. Besides this problem statement, the solution should satisfy the generic system constraint on robustness, safety and complexity, as described in Chapter 1.

The remainder of this chapter is structured as follows. Section 3.2 starts with motivating the solution architecture and presenting the framework, after which our main contributions on new color modeling strategies are established, building upon the previous chapter. Section 3.3 elaborates on the evaluation approach, including the new publicly available dataset and the design of the experiments. The results of this are presented in Section 3.4. Lastly, conclusions are provided in Section 3.5.

3.2 Method: the Color-based Stixel World algorithm

3.2.1 Background motivation

The system in the previous chapter relied on simple color models that were kept up-to-date in an automated, self-supervised way, while the car was driving. This online updating required disparity estimation for the self-supervising process. An alternative strategy to *online* color modeling is *offline* modeling [44], which would completely remove the need for online disparity estimation in the vehicle. However, since the online strategy functions successfully to address the challenging nature of traffic environments, the work in this chapter continues with the same principle of online learning. Traffic-scene appearance varies highly with weather conditions, geographical location, time of the day and the complexity of the scene. For instance, urban traffic scenes tend to predominantly contain gray-tones in low-light situations, in contrast to bright and open rural or highway scenes. We consider it more feasible to build a robust, yet discriminating color model which is adapted to that specific time and place, rather than designing a generic color model that holds for every environment and weather condition and is still of low complexity. On top of this, having the disparity data available online facilitates distance-aware color modeling. As a result, the disparity estimation cannot be completely removed from the system, but the key aspect is that it is no longer required to be executed at the same frame rate, and hence, it does not anymore negatively impact the latency of the critical path of the system.

Based on the above considerations, we now refine the solution direction for the problem statement of this chapter, which leads to the following aspects:

- the Stixel World concept is now re-defined to a Color-based Stixel World algorithm that does not require disparity estimation in its critical path, in contrast to the previous versions;
- the new color modeling and processing is made distance-aware (which is the key contribution of this chapter), still utilizing disparity but at a reduced frame rate;
- a novel annotated dataset is provided to validate the effect of our contributions under adverse imaging conditions.

3.2.2 Framework architecture

The research in this chapter addresses two subproblems: (a) defining an efficient and robust color-based cost term for within the stixel framework, and (b) finding a color representation that is informative enough to separate freespace from obstacles and yet is sufficiently suited for online processing.

Figure 3.1 shows an overview of the designed framework. The basis of our approach is the online self-supervised learning method as described in Section 2.3.3. This method processes preceding stereo frames and generates a (noisy) freespace-versus-obstacle labeling based on disparity. Consecutively, this labeling is exploited as self-supervised training masks for the color representation for these two classes.

These previously mentioned sub-problems are addressed in four following subsections, describing the Color-based Stixel World objective, the distance-aware processing, choices on color representation and color-space selection.

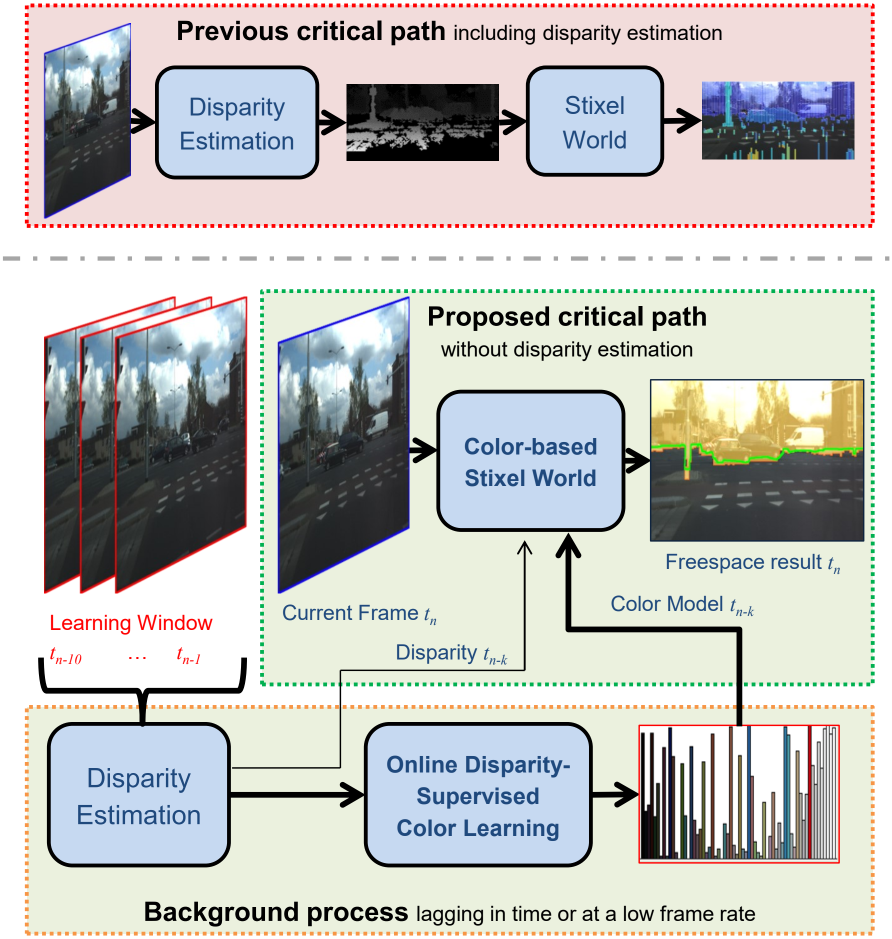


Figure 3.1 — *Comparing the critical paths of previous methods [33], [51] and the proposed Color-based Stixel World segmentation system. The latter one has no direct dependency on disparity, since the disparity-supervised color modeling in the lower part of the scheme can operate on selected intervals or executed at a lower frame rate than the freespace segmentation, by varying the range of the learning window (the possible time delay is indicated with the offset* k*).*

3.2.3 Color-based cost function

This subsection describes the structure of the Color-based Stixel World for which the distance-aware color processing is presented in the next subsection.

At its core, the stixel optimization process relies on probability distributions, as described in Section 2.2. Therefore, the Color-based Stixel World algorithm requires a color-based likelihood function, specified by

$$P(C_u|L_u) \sim \prod_{n=1}^{N_u} \prod_{v=v_n^b}^{v_n^t} P(c_v|s_n, v), \tag{3.1}$$

which is analogous to the specifications of Equations (2.3) and (2.6). As discussed in Section 2.2, parameter n is the segment index, N_u the number of segments in L_u, and v_n^b and v_n^t the bottom- and top-row index of segment s_n. Each segment contains a certain color at every row (c_v), and has a label $l_n \in \{g, o\}$, representing the ground and obstacle classes, respectively.

The term $P(c_v|s_n, v)$ should capture the probability of a certain color measurement given a potential freespace or obstacle segment. In the strong fusion approach of the previous chapter, this color-based term is simplified to $P(c_v|s_n, v) = P(c_v|l_n)$, under the assumption that the probability only depends on the label of the segment under evaluation, and not on its position v. In the new model of this chapter, this simplifying assumption is refined to facilitate the distance-aware modeling. Therefore, the result is a mixture model with a uniform distribution with probability p_{out} to model outliers and a normalized histogram-based distribution per class over all colors with probability $P_{\text{DA}}(c|l)$, formally written as:

$$P_{\text{mix}}(c_v|s_n, v) = p_{\text{out}} + (1 - p_{\text{out}}) \cdot P_{\text{DA}}(c_v|l_n, v), \tag{3.2}$$

where the subscript of P_{DA} indicates that this distribution will be distance-aware. The following subsection describes our strategy to compute the corresponding distribution.

3.2.4 Distance-aware color analysis

Including the distance-awareness into the histogram-based color modeling is motivated by the basic phenomenon that camera images naturally suffer from geometric, perspective distortion. Effectively, pixels representing areas close to the camera contain a smaller real-world surface than pixels representing areas at a large distance. Therefore, surfaces that are close to the camera are dominant in regular histograms, which contain only basic pixel counts. This imbalance can result in inaccurate color modeling of far-away obstacles.

To address this issue, our distance-aware color modeling consists of three elements: a weighted and an unweighted version of the color histogram and a leveraging function. The first element is the weighted color histogram $P_{wh}(c|l)$. This is generated by weighting each pixel with its corresponding real-world surface during computation. As a result, this histogram is more balanced towards obstacles at a large distance. However, stereo-based distance measurements are less certain at large distances, potentially leading to false obstacle detections close to the camera. Therefore, as a second element, the algorithm also builds the regular, unweighted histogram $P_h(c|l)$. Both histograms are integrated into the distance-aware posterior distribution P_{DA} using a balancing factor as follows:

$$P_{\text{DA}}(l_n|c_v, d, v) = (1 - \alpha_w(v, d)) \cdot P_h(l_n|c_v) + \alpha_w(v, d) \cdot P_{wh}(l_n|c_v). \tag{3.3}$$

Here, $\alpha_w(v, d)$ is a factor in the unity interval $[0, 1]$ to linearly balance the regular and the distance-weighted color posteriors, which are calculated from the corresponding histograms using Bayes' rule. We define the factor $\alpha_w(v, d)$ empirically after having discovered that a linear relation is useful for incorporating the perspective nature of our images with earth at the bottom, and a surface-based component to exploit that nearby objects cover a larger image area. The former aspect is covered with $\alpha_{\text{linear}}(v)$ and the latter with $\alpha_{\text{surface}}(v)$. These factors for a pixel on row v and having disparity d are specified by the following set of equations:

$$\alpha_w(v) \quad = (\alpha_{\text{linear}}(v) + \alpha_{\text{surface}}(v))/2, \tag{3.4}$$

$$\alpha_{\text{linear}}(v) = \begin{cases} v/v_{\text{horizon}}, & \text{if } v \leq v_{\text{horizon}} \\ 1, & \text{otherwise} \end{cases}, \tag{3.5}$$

$$\alpha_{\text{surface}}(d) \quad = (1/\zeta) \cdot \sqrt{|A_{\text{surface}}|(d)}. \tag{3.6}$$

In these equations, v_{horizon} is the row-index of the horizon, and ζ is the maximal pixel size that is present in the current disparity data, which ensures that $\alpha_{\text{surface}} \leq 1$. In our stereo-camera framework, the real-world surface area $|A_{\text{surface}}|$ that is represented by a pixel with disparity value d, can be determined from the disparity signal and the camera parameters $\mathcal{C}$. More specifically, using the camera focal length in pixels ($\mathcal{C}_{\text{foc,pix}}$) and in millimeters ($\mathcal{C}_{\text{foc,mm}}$), the size of a pixel in the sensor in millimeters ($\mathcal{C}_{\text{pix,mm}}$) and the stereo baseline ($\mathcal{C}_{\text{b}}$), the real-world surface can be computed from the following geometric relation:

$$|A_{\text{surface}}|(d) = \left(\frac{\mathcal{C}_{\text{foc,pix}} \cdot \mathcal{C}_{\text{b}}}{d} \cdot \frac{\mathcal{C}_{\text{pix,mm}}}{\mathcal{C}_{\text{foc,mm}}} \right)^2. \tag{3.7}$$

Since we have removed the disparity estimation from the critical path of the color-based freespace segmentation, the system has to rely on a disparity signal from at least one frame earlier. Fortunately, the differences between consecutive frames are small and, on top of that, they are smoothed by the probabilistic nature of our processing. The algorithm relies on a fixed linear groundplane model to fill any holes in the disparity map, prior to determining pixel surfaces. Fig. 3.2 illustrates these steps and their effect on the posterior distribution.

Note that the strategy of making the processing distance-aware cannot be achieved by simply computing histograms using a Birds Eye View (BEV) representation of the image. A BEV representation can work for the groundplane, but it heavily distorts the area of obstacles, since it projects all image pixels onto the

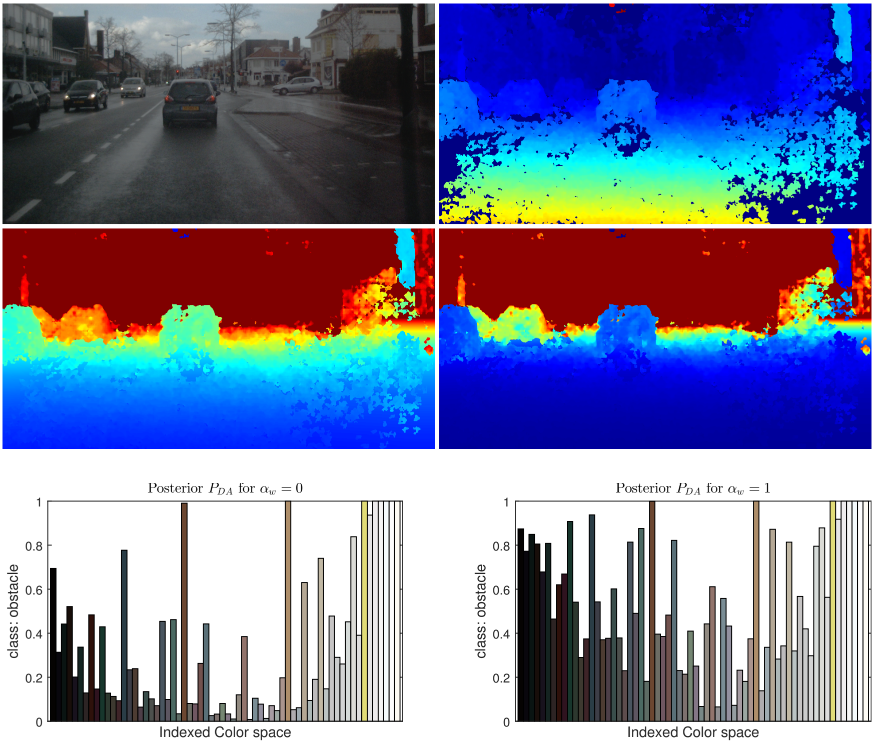


Figure 3.2 — *Illustration of the proposed depth-aware processing (DA). Top row: the brightened camera image (left) with its disparity signal (right). Middle row, left: the distance image, where holes are filled in using a static, linear, groundplane assumption and distance saturates at 35 m. Middle row, right: real-world surface of each pixel where colorization illustrates quadratic course of pixel surfaces. Bottom row: the posterior for the obstacle class without (left) and with DA (right).*

same flat plane. In contrast, our approach models each pixel surface individually, leading to a more accurate representation of obstacles in the histograms.

The steps of our online learning strategy, which builds the color models from the stereo images in the learning window, are more formally presented in Algorithm 2. As discussed before, this process runs in parallel and potentially on a lower frame rate than the freespace segmentation process. The steps of the segmentation, which is executed on every new frame, are presented in Algorithm 3. Since the aim of the work in this chapter is to create a separation between the core functionality of determining the freespace and the background process of updating the color models, we explicitly present the core process of segmentation as a separate element in Algorithm 3 and keep the background color modeling process in Algorithm 2. The next two subsections complete the description of our method by discussing our use of color spaces and color representations.

Algorithm 2 Distance-Aware Online Color Modeling

```
Input: stereo images $I_{L,t}$ and $I_{R,t}$ of the learning window $LW$

for each $\{I_{L,t}, I_{R,t}\} \in LW$ do
    $D_t \leftarrow$ EstimateDisparity($I_{L,t}, I_{R,t}$)
    $L_t^* \leftarrow$ EstimateDisparityStixels($D_t$)
    $C_t \leftarrow$ TransformRGB2Color($I_t^L$)
    $W_t \leftarrow$ CalculateDistanceAwareWeights($D_t$)
    for $l \in \{ground, obstacle\}$ do
        $TM_t^l \leftarrow$ GenerateTrainingMask($L_t^*, TM_{\text{prior}}^l$)
        $X_t^l \leftarrow$ ExtractSamples($C_t, TM_t^l$)
        $H^l \leftarrow$ AddToRegularHistogram($H^l, X_t^l$)
        $H_w^l \leftarrow$ AddToWeightedHistogram($H_w^l, X_t^l, W_t$)
    end for
end for

for $l \in \{ground, obstacle\}$ do
    $P_h(C|l) \leftarrow$ NormalizeHistogram($H_h^l$)
    $P_{wh}(C|l) \leftarrow$ NormalizeHistogram($H_{wh}^l$)
end for
Output: regular and weighted color models $P_h(C|l)$ and $P_{wh}(C|l)$ for $l \in \{ground, obstacle\}$
```

Algorithm 3 Segmentation with the Color-based Stixel World

```
Input: image $I_t$, color models $P_h$, $P_{wh}$ and disparity $D_{t-n}$
  $\alpha_w \leftarrow$ CalculateBalancingFactors($D_{t-n}$)
  $C \leftarrow$ TransformRGB2Color($I_t$)
  $L^* \leftarrow$ ColorStixelSegmentation($C_t, \alpha_t, P_h, P_{wh}$)
Output: Optimal Labeling $L^*$
```

3.2.5 Color-space selection

Since the input of the proposed processing pipeline contains color images, a key preprocessing step is the selection of the color space. Various color spaces exist, to accommodate for different purposes in color processing. In this chapter, we restrict ourselves to the most potentially beneficial alternatives, selected by either a broad acceptance and availability, or a potential complexity reduction.

Besides the color spaces, the involved preprocessing step is indicated as well. In our case, the applied preprocessing is histogram equalization (HEQ), which is helpful, especially since the work is aimed at handling dark, low-light frames. It is performed on the raw RGB images (individually on each color plane) prior to converting it to a different color space. Our experiments will test and compare the selection of color representations listed below.

RGB RGB is employed as the main full-color reference color space.

HS Derived from the HSV color space, the Hue and Saturation component are explored to increase the robustness against varying lighting conditions. The V component is not used. HSV can be derived from the RGB color space by a fixed transformation [61].

IllumInv The Illuminant Invariant color space, presented in [62], is a more elaborate method for robust handling of changing lighting conditions and even shadows. It requires an automated offline camera-calibration method to find a parameter θ, which can then be used to transform each new image into an illuminant-invariant single-component image. We have adopted the proposed robust entropy-based calibration method and found that $\theta = 90 \pm 0.5^{\circ}$ for our camera. We refer explicitly to [62] for more details on this color space and calibration method.

Gray Executing the segmentation on a grayscale image representation serves as a baseline for extreme cases of monochrome lighting conditions. This model is investigated for complexity reasons, since it would significantly reduce the constraints on the camera hardware and the corresponding data bandwidth when the grayscale analysis is successful.

3.2.6 Relative color representation

In contrast to the color-space selection described in the subsection above, several aspects of the color representation are kept constant throughout this chapter, in line with the findings presented in Chapter 2. First of all, the proposed system always employs the median-cut algorithm on the frames in the learning window [54], as discussed in Section 2.3.2. This ensures an adaptive color representation that has both a sufficiently low complexity for fast processing and is still suitable for the current traffic scene, as the color reduction is performed online. Second, the data is further reduced by employing stixels that span 11 image columns. This increases the robustness and decreases the computational load at the cost of horizontal resolution in the labeling.

Since a stixel data column spans several input data columns (11 in our experiments), these input columns need to be condensed into a single column vector for each stixel. The work in this chapter presents and evaluates the following three methods for doing this.

Method 1 (take the mode): The first method is taking the mode (the most occurring value in the set) over indexed RGB values for each image row in a $[w_{stix} \times w_{stix}]$ window, located at the central image column of each stixel. Since colors are represented with indexes, that have no direct mathematical or functional relationship to each other, taking the mode is the most straightforward method of obtaining a robust, representative value.

Method 2 (mode and edge strength): The second method is to add color variation to the modeling, instead of only considering absolute color. The reason is that relative information may also be descriptive in this case, since freespace areas tend to be more homogeneous than obstacles such as cars, pedestrians and houses. Therefore, the color mode is combined with local edge strength, to assess both absolute and relative color information. The local edge strength is calculated using

Sobel filter responses, averaged over a $[w_{stix} \times w_{stix}]$ window, which is shifted vertically along the individual stixel columns.

Method 3 (first and second mode): As a third alternative, the relative color representation is extended with explicitly modeling color pairs, instead of measuring color variation. Specifically, this consists of both the first and the second mode in a $[w_{stix} \times w_{stix}]$ window (in homogeneous areas, the first and second modes are taken equal). This makes the relative color modeling more informative and more discriminative, since it considers both local color homogeneity and specific color pairs. The latter aspect is not accounted for when using local edge strengths.

Together with the other settings, all three color representations will be tested and compared in experiments, as described in Section 3.3.5.

3.3 Evaluation approach

This section presents the approach to evaluate the performance of the proposed color-based freespace scene segmentation, including the dataset and executed experiments.

3.3.1 Dataset

Two datasets, called EHV-road14 [33] and EHV-road15 [34], are employed to evaluate the different configurations of the proposed system. Both datasets are acquired in an urban environment, using a BumbleBee2 camera, mounted behind the windshield of a car just below the rear-view mirror. The camera has a baseline of 12 cm, a resolution of 1024×768 pixels and a frame rate of 20 Hz. Both datasets are publicly available[1].

Whereas EHV-road14 contains both frames with bright and frames with dim-light conditions, EHV-road15 is solely focused on dark, clouded, low-light and rainy frames. The first dataset was already described in the previous chapter. The second, EHV-road15, consists of 114 frames that have a road and a freespace annotation (road including pavement, grass, etc.). For each annotated frame, the 10 preceding frames are also available to facilitate the online color modeling. The sequences are selected in such a way that they neither contain windshield wipers nor obstacles directly in front of the ego-vehicle (e.g. within 1-2 m), since those would hamper appropriate disparity estimation. The two datasets combined contain a large variety of relevant traffic situations, such as crowded streets, road repair sites, large crossings and highways. They contain asphalt as well as paved roads of several colors (black, gray, red).

Unfortunately, our algorithm cannot be executed on benchmarks such as the KITTI dataset [55], since such sets do not contain the preceding frames of annotated road images, which are required for our online training strategy.

[1] The data can be found via `http://tue-mps.org`.

3.3.2 Configuration of disparity estimation

Our system employs a multi-threaded version of the OpenCV implementation of the Semi Global Block Matching algorithm of [56] to obtain disparity measurements, with similar settings as in [33] (see Section 2.4.2). The pixel range of the disparity estimator is set to $d_{\text{min}} = 1$ and $d_{\text{max}} = 48$. Additionally, we exploit the *winner margin* measure to force the algorithm to provide only measurements with a high confidence at the cost of a reduced density in the disparity signal. This is beneficial for the baseline disparity Stixel World method, since it can handle missing values better than erroneous ones. This can be seen as a simplification of the work from [57], in which disparity estimates are accompanied by a confidence measure to adaptively set an outlier probability. In our approach, this confidence is binary with a relatively strict threshold, so that the *winner margin* is at least 20.

3.3.3 Configuration of Stixel World parameters

As described, our camera has lower resolution and a smaller stereo baseline than e.g. the camera used for the original Stixel World algorithm [51], resulting in lower quality disparity estimates. To compensate for this deficiency and to obtain more favorable results for the baseline method, we have made improvements to the baseline framework, as presented in [33] and discussed in Section 2.4.3. For example, our implementation estimates the groundplane model online, instead of using a single fixed model, and exploits tuned label-based transition probabilities defined in $P(L)$ to boost the performance of the baseline method even further.

Our experiments adopt a stixel width of 11 image columns and subsample the disparity and color signals vertically with a factor of 3, prior to segmentation. Note that the full-image data is exploited to compute look-up tables and color models, which is comparable to the approach in [51]. The stixel width and subsampling settings have been selected empirically and provide a decent tradeoff between execution time and freespace modeling accuracy in the range of interest with the employed camera (visible in the results presented in Fig. 3.5). At present, the version of the proposed Color-based Stixel World method is a MATLAB-based implementation. Core parts of the added functionality have also been implemented in C++ for analysis, showing a reasonable complexity and promising execution times. For instance, generating the look-up table with color modes takes 11 ms on a consumer-grade notebook [63], so that real-time executions are facilitated as in [51].

3.3.4 Evaluation metric: freespace per stixel column

As a key quantitative analysis method, the *detected freespace per stixel column* is evaluated. This main metric is denoted with F_{stxcol}, *i.e.*, the percentage of stixel columns for which the freespace is correctly determined. As a ground-truth reference, the true freespace is calculated from the manual drivable-surface annotation for each stixel column. The image-mask annotations are translated into real-world distances using a static linear model of a flat groundplane surface, which effectively translates a stereo-image row index into a distance.

Returning to the evaluation of the metric, we calculate the quality of the detection masks by measuring the deviation between detected and annotated freespace, which is expressed as a percentage for each column. Furthermore, for the metric calculation we define certain margins around this ground truth, because objects are detected with the quantized stixel resolution, and the reference masks can be inaccurate at pixel level. Therefore, freespace detections are counted as correct when they are within the range of 30% too short or 15% too long. This asymmetrical range reflects the fact that missing an obstacle is more dangerous than detecting one too close. For the same reason, we distinguish the incorrect stixels into obstacle misses (freespace too long) and false obstacle detections (freespace too short). Note that, although a deviation of 30% may seem large, it corresponds to only a couple of pixels after several meters and only to a few centimeters in regions close to the ego-vehicle.

Metric discussion: In essence, this metric is conceptually comparable to our corridor-level drivable-distance metric in Chapter 2, but is computed at a higher horizontal resolution, since it uses the width of a stixel instead of the width of a car. The latter resolution is too coarse for in-depth analysis of the freespace segmentation performance. It should be noted that evaluating at a corridor level has two pitfalls. Namely, a single false nearby obstacle heavily degrades the achieved recall, and additionally, detecting a large obstacle only in a single stixel column already results in a high precision of the drivable distance. In contrast, our new stixel-resolution metric provides a more detailed insight about the best settings for obtaining reliable results. Moreover, our current evaluation analysis considers the complete freespace region (road and non-road), while the evaluation in the previous chapter disregards potential errors in the non-road freespace. Therefore, the new metric is still designed with our specific ADAS application in mind, while offering more detailed insights and being more strict.

3.3.5 Experiment design

This section briefly presents the experiments assessing the color representation, the distance-aware processing and the online-learning aspect of our Color-based Stixel World framework. The evaluation consists of three categories.

The first category discusses the most important comparison of adding our RGB analysis and comparing that with the reference baseline methods. The second category involves evaluation of the influence of the most critical parameters like the distance-aware color modeling, adding histogram equalization and configuring the learning window. The third category presents two tests at a higher level. More specifically, these experiments look at different datasets and perform an oracle-analysis of optimal settings per frame, to identify potential bottlenecks of the current system design.

All our results are compared to our implementation of the disparity-only baseline (reference) approach of [51], as well as to the color-extended method presented in Chapter 2.

3.4 Results

This section commences with illustrating the added value of our distance-aware Color-based Stixel World framework with a first qualitative result. Fig. 3.3 shows qualitative results of our best method in comparison with the results of the disparity baseline. It can be observed that our Color-based Stixel World algorithm typically provides similar as or even better freespace results than the disparity method. The bottom-right case presents a problematic case, since the artifacts in the disparity segmentation are consistent throughout the full learning window for this sample. This causes the image areas with light reflections to be modeled as obstacles in the color model, leading to false obstacle detections in the color analysis.

The next subsections address the aforementioned three categories of more detailed experiments. The first category involves comparison to the baseline.

3.4.1 Quantitative results with RGB compared with baseline methods

The quantitative results of our first set of experiments, using the RGB color space only, are shown in Fig. 3.4. First the experiments were conducted with the combined dataset, plotted at the left. For a refined view, the datasets are also evaluated individually, plotted in the middle and right graphs. In the graphs, each green-orange-red bar triplet represents the quantitative results of one execution with a fixed, defined configuration. In the following discussion, this is referred to as a *run*. Each run uses a specific configuration, of which the most influential aspects are indicated in the tag line. For example, run *j*, the fourth triplet from the bottom, uses histogram equalization (HEQ), with color representation Method 2 (Mode 1 & Edge), and distance-aware color modeling (DistAw).

With respect to the presented results, it is noteworthy that the Color-extended Stixel World of the previous chapter (run *b*; $F_{\text{stxcol}} = 66.3\%$), obtains a lower score than the baseline system (run *a*; $F_{\text{stxcol}} = 77.3\%$). This is the effect of our new metric, which is more strict due to its higher horizontal resolution and considers a larger range towards the horizon. Our earlier algorithm of Chapter 2 was mainly tuned to reduce the number of false nearby detections. This is confirmed by the graph in Fig. 3.4, which shows that the percentage of stixels with false obstacle detections is reduced (from 8.9% to 3.2%). However, the number of stixel columns with missed obstacles increases (from 13.8% to 30.5%), resulting in a lower number of correct stixel columns. This can be explained by the fact that the first algorithm did not consider distance-aware color modeling, whereas the new metric has an extended range, so that the distance-awareness becomes more important. With our new method, which considers color pairs and is distance-aware (run *k*), the amount of correct stixels increases to 77.6%, which is 11.3% better in comparison with the strong-fusion method (run *b*). It is also an improvement over the disparity baseline (run *a*), although to a much smaller extent of only 0.3%. The main accuracy gain of the new algorithm compared to the disparity baseline, is the reduction of stixel columns in which the freespace is overestimated (obstacles are missed) from 13.8%

to 11.8%. In general, it is worth noticing that our method achieves the highest results on the most difficult subset of the data with very low-light conditions (right graph in Fig. 3.4). This shows that color is a relevant signal modality that can be exploited in situations that are difficult to handle with disparity alone. A more detailed discussion on the different configurations follows in the next subsections.

The second category of experiments evaluates the most important configuration aspects of our contribution, *i.e.* distance awareness (Subsection 3.4.2), color spaces with histogram equalization (Subsection 3.4.3) and learning windows (Subsection 3.4.4).

3.4.2 Evaluation of distance-aware color processing

Fig. 3.5 provides a detailed comparison between the results of the Color-extended Stixel World that uses strong fusion of disparity and color and our new color-based method. It presents color-based results *without* and *with* distance-aware processing. The figure clearly shows that the strong fusion method tends to miss parts of obstacles, specifically at large distances, where the uncertainty in the disparity signal is high and the color contrast is typically low. The proposed algorithm reduces these errors to a large extent with the more informative color modeling. On top of this, the distance-aware processing (DA) gives a further improvement and makes the results more consistent. The added value of DA is also quantitatively visible in Fig. 3.4, by comparing runs i, j and k to f, g and h, respectively, within the left and right graphs, by the larger green bars.

3.4.3 Grouped analysis of histogram equalization and color space

The previous subsection addressed the evaluation of the relative color representation and the distance-aware processing. The other settings of the system concern the selection of the color space, the use of histogram equalization and the choice of the learning window for the online color modeling. The quantitative results of the previous subsections are reused to derive the overall influence of these individual settings, for example, all runs with histogram equalization (HEQ) versus all runs without.

All combinations of the selected color and learning window settings have been tested, resulting in 24 different configurations (runs). The effect of the individual color and learning window settings is illustrated in Fig. 3.6. It presents box plot analyses on F_{stxcol} (the percentage of stixel columns for which the detected freespace is correct), over all data, and over runs grouped by the specific settings of interest.

Using a paired t-test, applying HEQ provides a significant improvement over not using equalization ($p = 3.04 \times 10^{-8}$), and it increases the median F_{stxcol} from 67.4% to 74.1%. Likewise, the RGB color space, scoring a median of $F_{\text{stxcol}} = 76.4\%$, outperforms the hue saturation HS (68.5%), IllumInv (69.7%) and Gray (65.2%)

representations (with p-values of $p = 1.12 \times 10^{-18}$, $p = 1.93 \times 10^{-4}$ and $p = 1.85 \times 10^{-29}$, respectively). In conclusion, histogram equalization and the RGB color space are clearly attractive settings for the system configuration.

The third graph in this figure considers the configuration of the learning window, which will be discussed in the next subsection.

3.4.4 Performance of learning windows with all color settings

This subsection first continues with the results of the learning window experiments, as presented in the right graph of Figure 3.6. The paired t-tests of the statistical analysis shows that a full learning window (LW10:1:1) is better than a shorter, lagging one (LW10:1:3) with $p = 3.25 \times 10^{-3}$. No significant difference was found between the results of using the full or low frame-rate learning window (LW9:3:3) ($p = 8.33 \times 10^{-1}$). However, note that the actual scores are rather similar: the full, lagging and low frame-rate learning windows are achieving a median F_{stxcol} of 71.9%, 70.8% and 69.7%, respectively. In conclusion, the system performance is virtually not hampered by relying on a shorter learning window or one that executes at a lower frame rate.

More quantitative results of experiments with the configuration of the learning window are provided in Fig. 3.4. The best freespace segmentation results using the RGB color space are achieved by using a combination of histogram equalization, color pairs, and distance-aware color processing (run *k*). For this combination, several learning window parameters have been evaluated. Two exemplary results are provided in Fig. 3.4, using only frames $t-10$ to $t-3$ (disregarding the two most recent frames, run *l*), or using the full range ($t-10$ to $t-1$), but at a lower frame rate by skipping two of every three frames (run *m*). Both achieve very similar results to run *k* ($F_{\text{stxcol}} = 77.6\%$): *l* scores $F_{\text{stxcol}} = 76.1$ and *m* scores $F_{\text{stxcol}} = 77.5\%$. This illustrates the robustness of our method with respect to the online training strategy. Our algorithm does not require all preceding frames and also does not require the adjacent preceding frame or frames for the current evaluation. This is an important result, since it shows that we can remove the disparity estimation from the critical processing path to lower the computational requirements for real-time execution of the freespace segmentation function.

The last category of experiments involves the higher-level tests, namely the influence of the dataset (Subsection 3.4.5) and the oracle analysis (Subsection 3.4.6).

3.4.5 Comparing performance on different data subsets

This and the following subsection describe more general tests at a somewhat higher level. This subsection discusses the influence of the dataset, since the value of our contribution can be more pronounced for certain datasets.

To this end, additional quantitative results are provided in Fig. 3.8. In this figure, all stixel columns over all frames are evaluated together for each individual run. The results are shown on the combined data as well as on the individual datasets. The rightmost graph in Fig. 3.8 clearly shows that the added value of color processing is more pronounced for the EHV-road15 data. This can be explained by the fact that the EHV-road14 contains frames with bright as well as frames with dim lighting conditions, whereas EHV-road15 is solely focused on dark, clouded, low-light and rainy frames. These situations are specifically difficult for disparity-based methods, rendering our use of color data more valuable. Of all color settings, run f results in the highest percentage of correctly detected freespace (77.6%, averaged over all data), which is similar to the disparity-only (reference) method (77.3%). For the EHV-road15 data, the improvement is higher: 78.0% compared to 74.4%, respectively. When specifically focusing on reducing the number of missed obstacles in difficult imaging conditions, run h reduces the percentage of erroneous stixels from 17.2% to 12.9%, compared to the disparity-only method. On the combined data, the stixel-error fraction reduces from 13.8% to 11.5%.

3.4.6 Oracle analysis on optimal settings per frame

The second higher-level test involves a different way of testing, concerning an oracle test to identify possible performance bottlenecks for potential improvement.

To this end, the five theoretical experiments at the bottom of Fig. 3.8 provide a relevant analysis at meta-level. These scores are generated by selecting the optimal setting for each frame out of a (subset of) the available runs, to assess the added value of the processing choices and to provide insights in where the most gain is to be expected in future research. First of all, it is noteworthy that for *every* setting, there are frames in the dataset on which it performs best. When the optimal score is selected from all possible runs (including the baseline that relies solely on disparity), the highest theoretical score can be achieved (86% correct), as could be expected. However, also with the color data alone there is room for improvement, compared to using the same color space and preprocessing step for every frame. This means that even with our adaptive median-cut color indexing, the system can extract more information from different color representations in different situations (fourth bar from below in Fig. 3.8; 83.3% correct). Also, it should be noted that even with the simplest learning window (LW9:3:3), the color-based Stixel World can outperform the disparity-only version with a more sophisticated color representation (the bottom bar in Fig. 3.8; 80.2% correct), although using more frames is still better (third bar from below in Fig. 3.8; 81.9% correct).

The aforementioned observations are illustrated with the qualitative visual results in Fig. 3.7, where the disparity-only results are compared to three of our color-based strategies. The figure shows (1) the setting performing the best (RGB+HEQ, LW10:1:1), (2) one of the experiments relying on the color space that was specifically designed for this context (IllumInv+HEQ, LW10:1:1), and (3) the experiment with the lowest computational complexity, since it uses grayscale images and only three LW frames (GRAY+HEQ, LW9:3:3). The top-two examples show that our methods are all capable of delivering similar or better results compared to the disparity-only framework in several situations. The bottom-two subfigures of

Fig. 3.7 illustrate that any variation of settings can always be coupled to a set of frames, where they perform best. Therefore, the system performance is expected to increase by adapting the color modeling in more ways than what has been explored in this chapter. For example, color spaces may be combined or selected online, or the most informative frames within the learning window could be selected adaptively. This fuels the insight that our system could benefit from more adaptivity than what is achieved so far. The results suggest that the flexibility of our model based on the online changes appears to be too restricted for optimal performance.

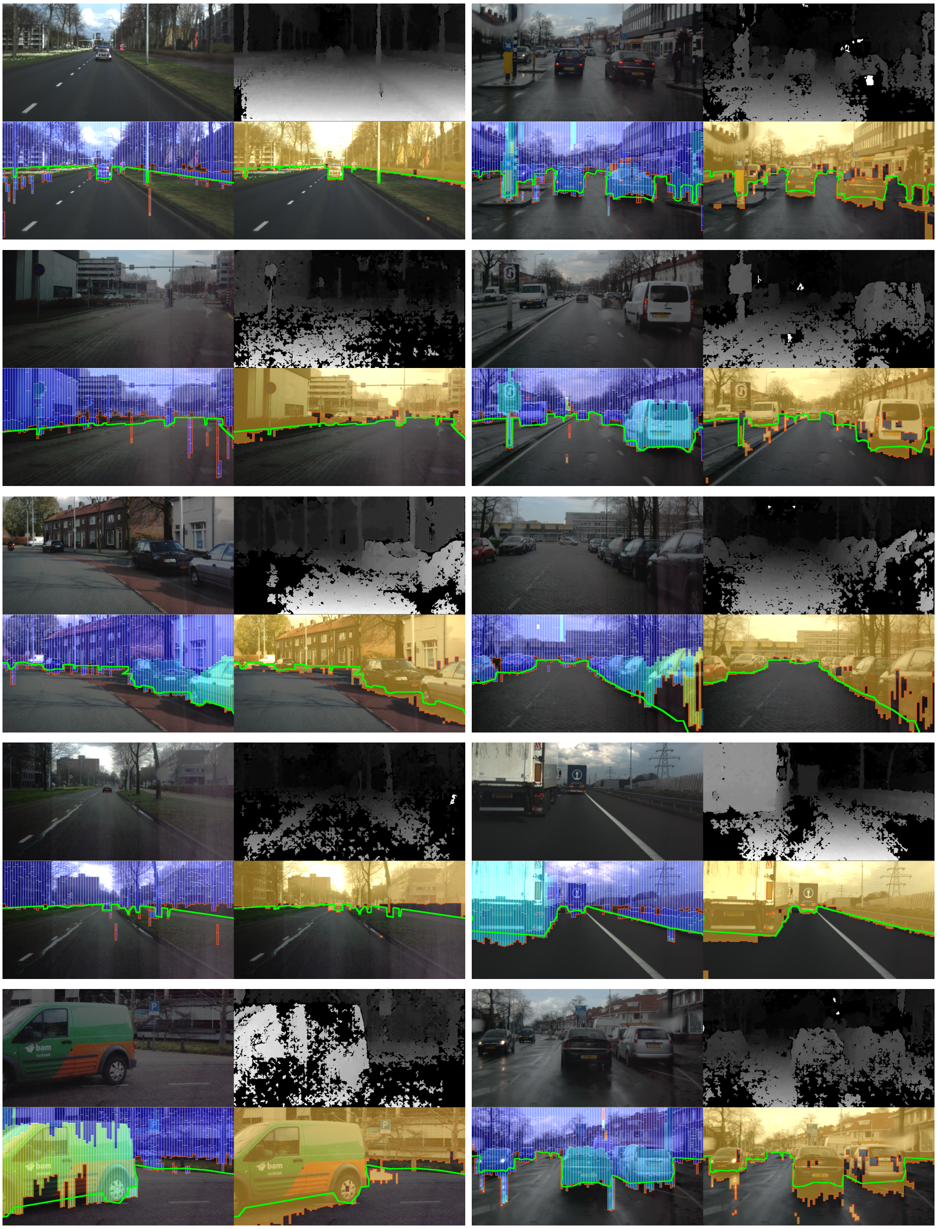


Figure 3.3 — *Qualitative results of our proposed method (HEQ; Mode1&2; DistAw). Each subfigure shows the left camera image (top left) and its disparity (top right); result of the disparity baseline result (bottom left) and our new improved result (bottom right). Green line: border of the ground-truth annotation of the drivable surface. In the disparity-based result, the stixels are colored by their depth (red (close) to blue (far)). In the color-based results, a homogeneous overlay of the detected obstacle region is visualized. The bottom-right example illustrates a case were our color-modeling cannot resolve all artifacts in the disparity-based learning window.*

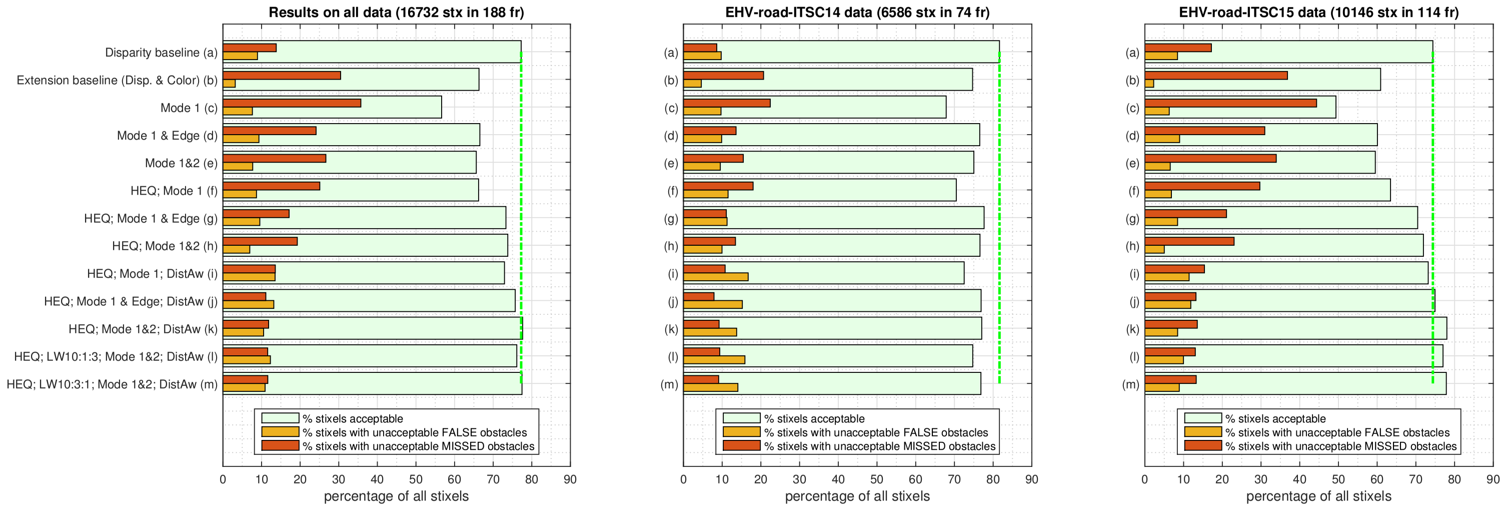


Figure 3.4 — *Quantitative results of correct, missed or over-estimated freespace detection for the baselines and several of our experiments. Color experiments all rely on indexed RGB without or with histogram equalization (HEQ). Learning window parameters: LW start:step:end; default setting is LW10:1:1, meaning that it it uses all 10 preceding frames. Run 'l' (LW10:1:3) disregards the two most recent frames, run 'm' (LW10:3:1) skips two frames out of three. Runs 'i' to 'm' exploit our distance-aware histogram processing (DistAw).*

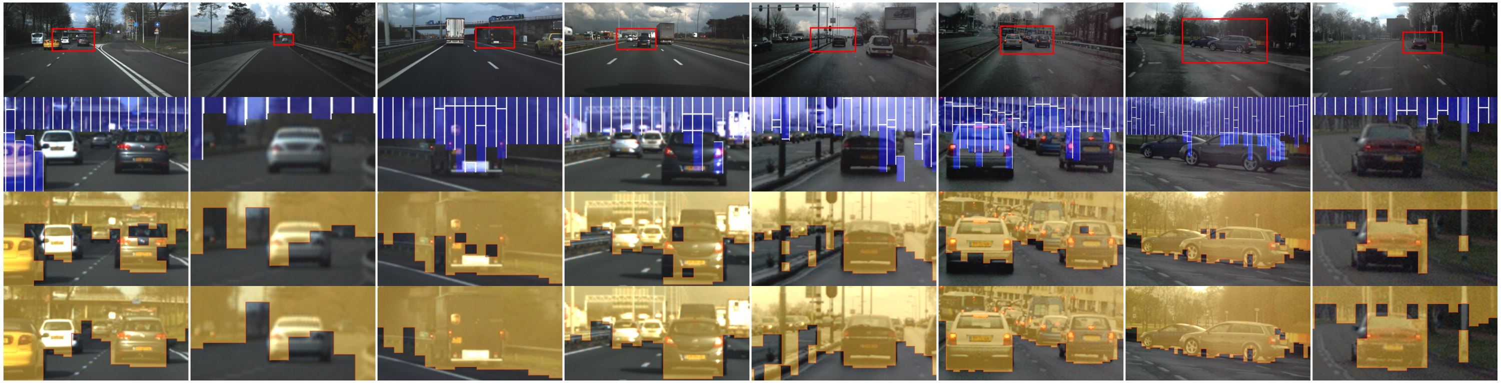


Figure 3.5 — *Qualitative results of freespace segmentation at a large distance to compare the strong fusion method of Chapter 2 with our proposed DA color-based method. The figure contains original images with red boxes on areas of interest (top row); examples from EHV-road14 (first four) and EHV-road15 (second four, brightened). The strong fusion method misses obstacles due to high uncertainty in the disparity signal, combined with low contrast in the color signal at large distances (second row). This is improved by using histogram equalization and color pairs (third row), and, additionally, by DA processing for increased consistency (bottom row).*

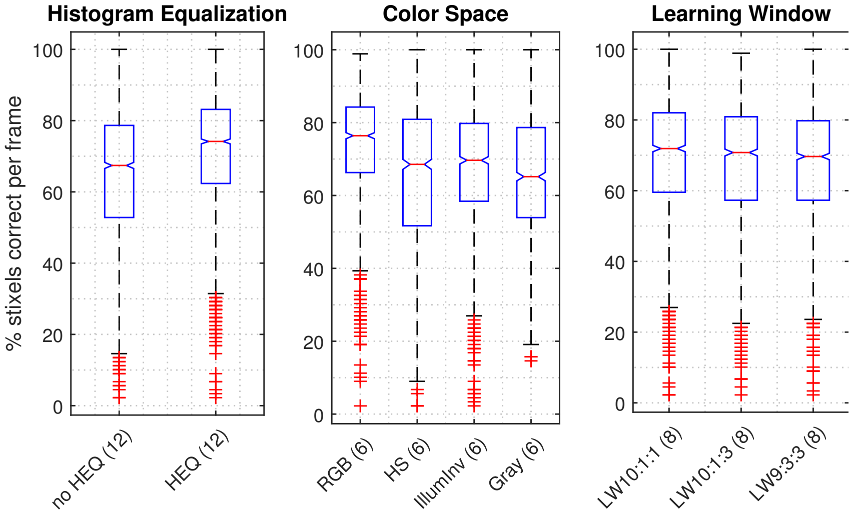


Figure 3.6 — *Box plots comparing the different settings for histogram equalization, color space and learning windows combined over all experiments (runs). For each frame in each experiment, the percentage of stixels with a correct freespace estimate is calculated, which is visualized as a box plot per setting. Hence, each box contains 188 data points per run. The number of runs per box is denoted in brackets in each label.*

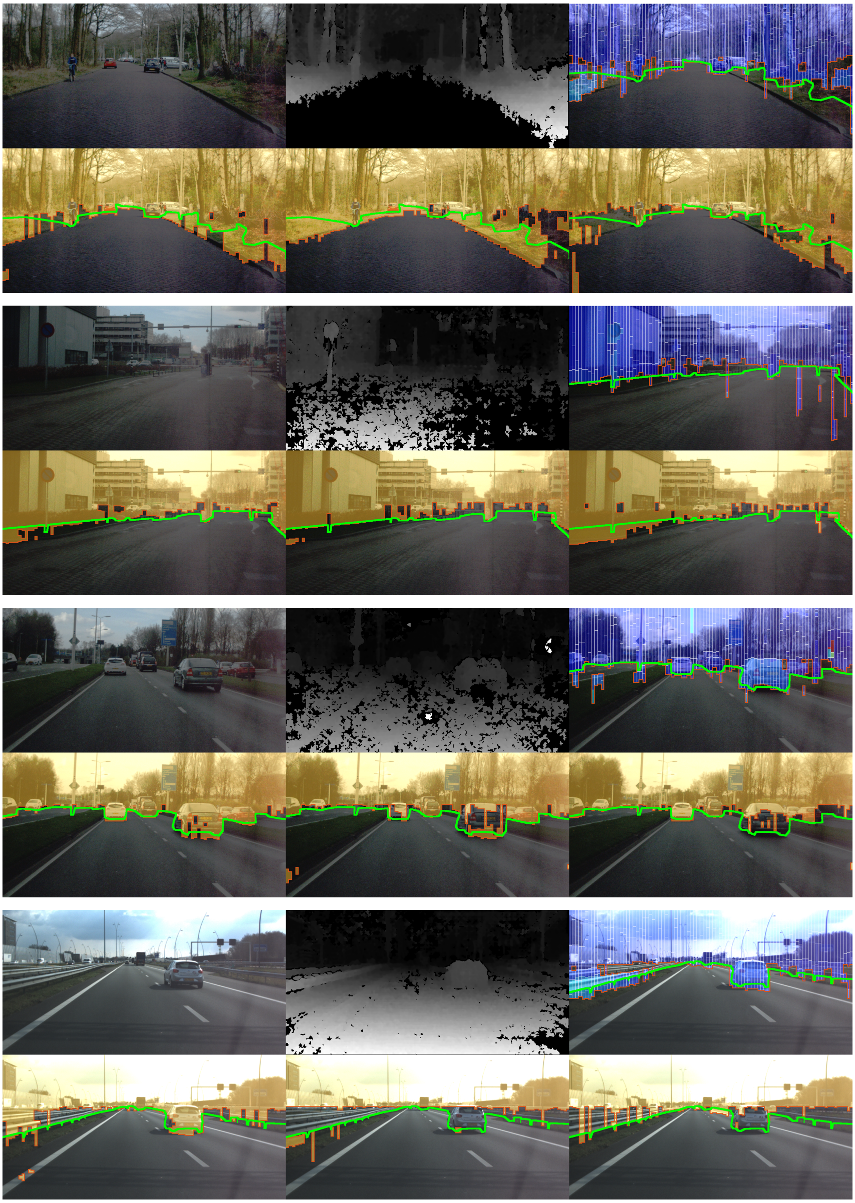


Figure 3.7 — *Qualitative results of the baseline versus three of our runs; each subfigure contains two input and four result images. Top row: left camera (left), its disparity image (middle), and the disparity baseline result (right, stixels colored by distance: red (close) to blue (far)). Our result at the bottom row: RGB+HEQ with LW10:1:1 (left), IllumInv+HEQ with LW10:1:1 (middle) and Gray+HEQ with LW9:3:3 (right). Our results show detected obstacle region (orange) and hand-annotated reference (green). Similar or better color-based results in the top two subfigures; the bottom two subfigures show examples of scenes where not all color settings provide equally acceptable results.*

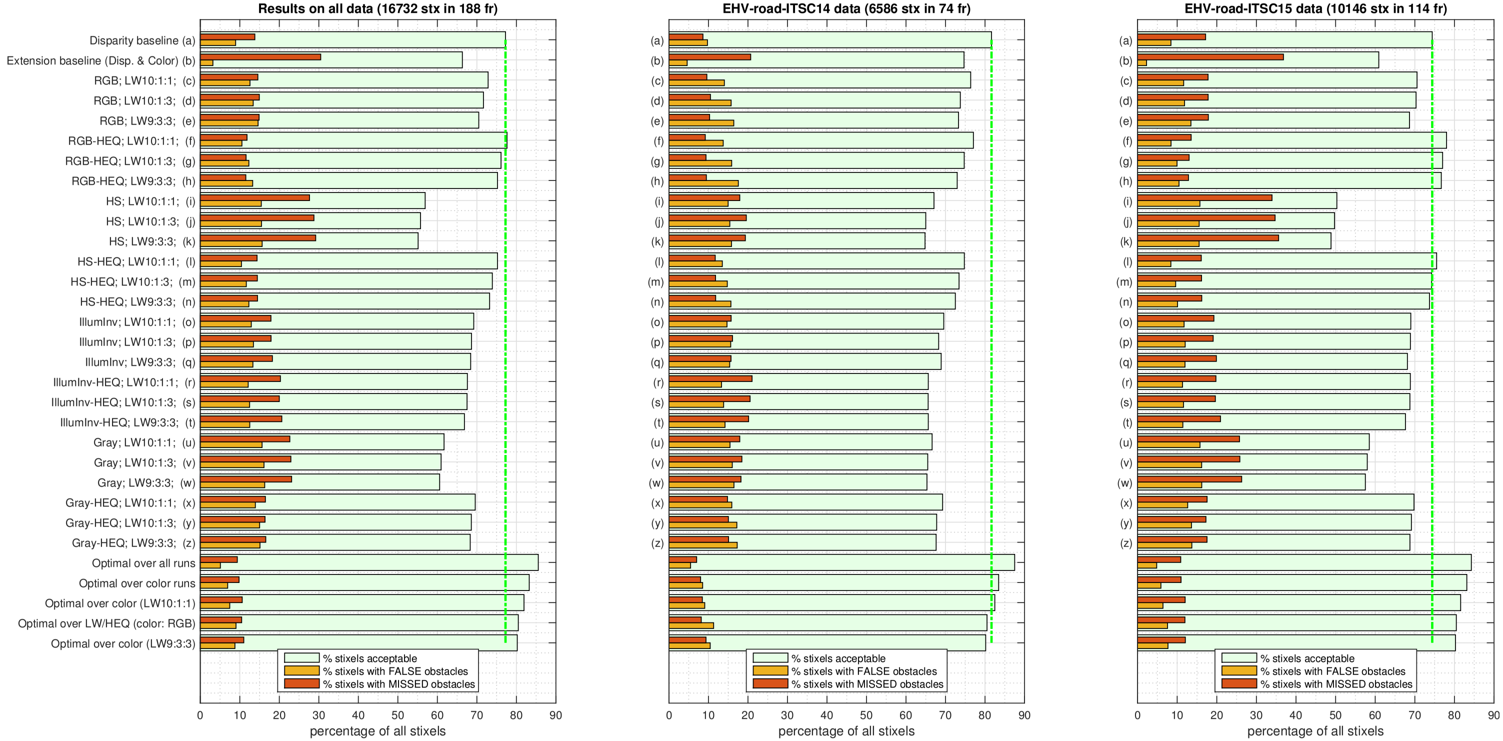


Figure 3.8 — *Quantitative results comparing the correct, missed and over-estimated freespace detections in our experiments with the results of reference methods. Reference systems are (a) the disparity-only Stixel World algorithm [51] and (b) the Color-extended Stixel World algorithm that uses strongly fused color and disparity [33]. Our experiments are listed (c) and further, tagged with labels that indicate use of histogram equalization (HEQ), color space and learning window (LW start:step:end). Final four entries: the theoretical oracle experiments that select optimal settings for each frame out of all runs (a-z, including disparity), out of all color-based runs alone (c-z), out of all runs with LW10:1:1 (c, f, ..., x), out of all runs with RGB (c-h) and out of all runs with LW9:3:3 (e, h, ..., z).*

3.5 Conclusions

This chapter has introduced a stixel-based probabilistic framework for color-based freespace and obstacle segmentation as complementary masks. Our system learns color appearance models for freespace and obstacle classes in an online and self-supervised fashion. To this end, it applies a disparity-based segmentation, which can execute in the background of the critical processing system path and at a lower frame rate than the color-based algorithm. This approach enables operation without requiring a real-time disparity estimate. Consequently, the current road scene can be analyzed without the extra latency of disparity estimation. This feature results into a reduced response time from data acquisition to data analysis, which is a critical property for high-speed ADAS.

Our proposed Distance-aware Color-based Stixel World algorithm contains the following novel aspects:

- *color-based cost function* for the Stixel World optimization process;
- *color processing* that generates the corresponding color representation with a balance between complexity and informativeness;
- *distance awareness* in the color modeling and cost function to address perspective camera distortion.

Besides these novel algorithmic aspects, we have contributed with our evaluation strategy as follows:

- *online-learning settings* have been again included in the exploration to analyze bottlenecks for system latency and potential future improvements;
- *new public dataset* with road annotations in scenes that emphasize difficult imaging conditions such as low-light and rainy scenes;
- *new evaluation metric* that measures improvements in drivable-corridor estimation, which is more detailed and more strict than the metric of Chapter 2.

The evaluation of the design of our Color-based Stixel World leads to several key results, which are summarized below.

A. Distance-aware processing: Our incorporation of distance-awareness in the color modeling improves the results with all color representations, for example, from 73.8 % to 77.6 % (+3.8 %) with the RGB-HEQ-Mode1&2 representation; and even from 66.2 % to 72.9 % (+6.7 %) for the RGB-HEQ-Mode1 representation on the combined datasets.

B. Color-space selection: When using RGB for the input, our framework outperforms the specifically designed Illuminant Invariant color space (median F_{stxcol} of 76.4 % versus 69.7 %).

C. Learning window design: The aggregated results over learning window settings show that using a learning window that is more favorable to latency reduction (namely, lagging and at lower frame rate) does not lead to a large drop in performance (median F_{stxcol} of 76.5 % versus 77.6 %), which illustrates the power of the online-learning scheme.

D. Dataset subset analysis: The new data shows that the added value of our algorithmic contributions is most pronounced in the data recorded under adverse conditions, where the disparity alone is not reliable enough for robust processing.

Summarizing, our algorithm is based on two key contributions: (i) an informative color-pair representation using the first and second mode of an online-adapted indexed color space, and (ii) distance-aware color-histogram processing, based on real-world metric pixel surfaces.

The evaluation on new, publicly available data shows that the color-based analysis can achieve similar or even better results in difficult imaging conditions, compared to the state-of-the-art disparity-only method. As an illustrative example, our color processing detects the correct freespace for 77.6 % of all stixels, compared to the disparity-only score of 77.3 %. Furthermore, the color-based method overestimates the freespace in only 13.5 % of the stixel columns in the most challenging dataset, thereby outperforming the disparity baseline (17.2 %) and the strong-fusion method (36.8 %) using the new, more strict evaluation process. This shows that the proposed system improves the quality of the freespace analysis, while simultaneously facilitating lowering the latency (compared to both disparity-based methods) and the computational load of the freespace segmentation algorithm (compared to the *strong*-fusion approach).

With respect to latency, the elegant strategy to reposition a dominant processing block outside the critical execution path, clearly contributes to reduction of the latency and makes the system independent of immediate, online disparity estimation. Although we did not quantify the latency reduction on our machine, literature on an industrial, state-of-the-art pipeline reports an execution time of 30 ms for the Stixel World algorithm on CPU and 40 ms for the disparity estimation on a dedicated FPGA. With these numbers, the latency could be reduced with more than 50 % of the initial latency value.

Besides the previous system aspects, the provided meta-analysis of the results shows that the proposed approach of online color modeling is beneficial and can be extended for further improvements, with potential scores of up to 82 % within the currently assessed parameter-setting space. The next chapter will further look into this strategy by exploiting convolutional neural nets (CNNs). Neural networks, and the broader field of deep learning, represent an interesting family of algorithms that recently became popular within the field of computer vision. In recent years, research has shown that CNNs are powerful in encoding color-appearance information and can be successfully deployed in a wide variety of image analysis tasks.

Chapter 4 Freespace segmentation with an online-tuned FCPN

4.1 Introduction

The previous chapter has explored the combination of distance-aware processing with color-based histogram analysis for freespace segmentation. This chapter extends this strategy by introducing and exploiting a neural network within the segmentation process. The research work presented in the previous chapter shows that (1) decoupling disparity and color (in other words, not longer relying on strong fusion) facilitates a reduction of the critical paths' latency by design; (2) the distance-aware color-based Stixel World algorithm can still provide good freespace segmentation; (3) our search space does not entail a single optimal color representation, and hence, a more adaptive color modeling can further improve the freespace segmentation.

A potential method for more advanced color modeling is using neural network-based approaches. Neural networks, the building blocks within the field of deep learning, are becoming increasingly successful and popular for image analysis. In the field of intelligent vehicles, many of the recent state-of-the-art algorithms rely on neural networks, mostly on Convolutional Neural Networks (CNNs). They excel in a wide variety of ADAS applications, such as stereo disparity estimation [64], object detection for cars and pedestrians [65] and road estimation [66], [67]. However, for an effective deployment of CNNs, supervised training typically requires thousands of manually annotated ground-truth labels. Even though this stage is performed offline, the manual labeling effort does not scale well to including all possible situations under all possible conditions. Moreover, to deal with a large number of classes and situations, neural networks typically contain millions of parameters, leading to substantial memory, power and bandwidth requirements. Considering the desired ADAS application, such a network should still fit within the hardware and timing constraints of the platform. Therefore, ADAS could potentially benefit from a small neural network, since it would (1) offer fast inference execution and hence a low-latency system, (2) yield a small memory footprint, and (3) lead to a reduced power consumption, all to comply

The work in this chapter has been presented at IEEE IVS-DD 2016 [37], NCCV 2016 [38] and IS&T EI-AVM 2017 [39].

with generic constraints for embedded computing.

In summary, the above system aspects on deploying neural networks in ADAS lead to several challenges for this chapter:

- encoding all required information for freespace segmentation in a wide variety of possible scenes into a network with low inference time and low memory footprint;
- further reducing the dependency on disparity for the freespace segmentation path (preferably completely);
- decreasing the dependency on manual labeling for any training stages.

The approach presented in this chapter addresses these challenges in a combined way. Following the approach of the previous chapter, the proposed system adopts the decoupled data flow and adheres to the online self-supervised training strategy, while employing more advanced color modeling by means of a small CNN. To realize this, we first show the possibility of training a CNN for freespace segmentation without the use of manual annotations, so in a self-supervised fashion. By employing the CNN, the freespace segmentation becomes independent from the disparity signal, so that it does not require disparity estimation in its critical path.

The remainder of this chapter is structured as follows. A compact discussion about related work is provided in Section 4.2. Our freespace segmentation method with self-supervised and online training strategies are described in more detail in Section 4.3. The accompanying validation procedures are provided in Section 4.4, with the corresponding results in Section 4.5. Finally, Section 4.6 summarizes the research findings.

4.2 Related Work

This section provides a short overview of several aspects of deep learning that are relevant to the research carried out in this chapter: semantic scene parsing, supervision strategies and transfer learning. Additionally, it provides a comparison of the current work with two specific closely related approaches. The position of our work within the current field is summarized in the last subsection.

4.2.1 Deep learning for semantic scene parsing

Freespace segmentation is a pixel-level classification problem in which the algorithm should decide whether a pixel represents freespace for every pixel in the input image.

The breakthrough successes of neural networks started with image-level classification [68]. Consecutively, several strategies were developed to go from global classification to fine-grained detection or full-scene segmentation, such as classification of sliding windows [69], or image region proposals [70]. Later, Fully Convolutional Networks (FCNs) have been introduced for pixel-level segmentation [71].

An FCN is a Convolutional Neural Network, where all fully connected layers are replaced by their (mathematically equivalent) convolutional counterparts. This adaptation transforms the network into a deep filter that preserves spatial information, since it only consists of filtering layers that are invariant to translation. A challenge with this approach for pixel-level segmentation is that FCNs also typically contain several subsampling layers. These increase the effective receptive field of the filters, which improves the quality of the results. However, it simultaneously reduces the size or resolution of the generated output, compared to the input image. To address this issue and generate true pixel-level labeling, the authors of [71] have introduced skip layers that exploit early processing layers, which have a higher resolution, to refine the spatially coarse information in the final layer. In this way, the output resolution matches that of the input.

FCNs have several attractive properties in comparison to the earlier methods for scene parsing. For example, FCNs have no constraints on the size of their input data and execute inference in a single pass efficiently per image, instead of a single pass per superpixel, window or region. Consequently, they do not require concepts like superpixel, region, sliding window or multi-scale pre- or post-processing [71]. Due to these attractive properties, the system in this chapter is based on an FCN.

4.2.2 Supervision strategies for deep learning

In literature, training a neural network typically requires many data samples for successful convergence of the large amount of parameters and proper generalization of the classifier. To facilitate this, different supervision strategies for this training process are adopted throughout the field.

Most commonly, training is performed in a supervised manner, which relies on the availability of corresponding target labels for each training sample [68]. Since acquiring the required labeling is cumbersome for the amount of data samples that is needed, alternative strategies are semi-supervised training, e.g. where an object-segmentation mask is provided only for the first frame of a video [72], and unsupervised training[73], which does not rely on any labeling at all. In the field of image segmentation, an additional interesting approach of training is weak supervision, where a limited set of related annotations are exploited for training. For example, the authors of [74] train CNNs for pixel-level segmentation on image-level labels, since labels for the latter are more abundant than for the former.

Even though weakly- or unsupervised training methods of CNNs are improving, they are currently still outperformed by fully supervised methods [73], [74]. Therefore, the work in this chapter relies upon an intermediate method: self-supervised training, which exploits automatically generated training labels. If training labels can be generated automatically, the amount of supervised training data becomes practically unlimited. However, this leads to a paradox, since it requires an algorithm that can generate the labeling, which is exactly the issue that needs to be solved. To this end, we propose to rely on an algorithm based on traditional (non-deep learning) computer vision methods. This algorithm needs not to be perfect, but at least sufficiently good to generate reasonable training labels. The goal is then that the FCN, trained with these imperfect labels, outperforms the conventional segmentation algorithm.

4.2.3 Transfer learning: adapting to new environments

The supervision strategies described in the previous subsection address the source of the labels of the training data, if indeed labels are required. Another aspect is the source of the data samples themselves. If properly labeled data samples exist in a different domain, one can try to transfer that knowledge to the domain of interest, instead of starting the training procedure from scratch. This is known as *transfer learning* or *domain adaptation*.

In image recognition and object detection problems in natural environments, a common method is to start with a network that is trained on a large and generic dataset [68], [73], [75], [76]. To apply this network to a new task, one can remove the last layer of the network, which provides class confidences, and train a new one for the problem at hand. This exploits the observation that these pre-trained networks are a compact and yet rich representation of the images in general, since they are trained extensively on a broad visual dataset [77]–[79]. An extension of this concept is not just retraining the last classification layer of a pre-trained network, but to also fine-tune a larger part or even the complete network with task-specific data [70], [76], [80], [81].

The work presented in this chapter also exploits domain adaptation. The general layout of a traffic scene does not change, so that base knowledge should be useful to transfer. In contrast, locations, times of day, imaging conditions and object constellations or viewing angles can differ highly, which requires adaptation. When this adaptation is performed during operation of the system, this adaptation strategy is called *online training*, which is slowly being adopted for video segmentation tasks successfully [72].

4.2.4 Online self-supervised adaptive road segmentation

Considering the overall approach in the context of freespace segmentation, our work is also related to both [43] and [82]. This section shortly highlights the similarities and differences.

The system presented in [43], automatically generates labels that are exploited to train a CNN for road detection, which is applied as a sliding-window classifier. The method also has an online component, which analyzes a small rectangular area at the bottom of the image (assumed road) and calculates a color transform to boost the uniformity of road appearance. The results of offline and online classifications are combined with Bayesian fusion. Our proposed work differs in several aspects. First, we do not need to assume that the bottom part of an image is road in the online training step, which is often an invalid assumption in stop-and-go traffic, since we exploit the stereo disparity as an additional signal. Second, the offline supervision masks used in [43] are generated using a surface classification algorithm [83] that is (re-)trained on manually labeled data, so that the method is not completely self-supervised. Third, the applied offline and online method in [43] is a hybrid combination of supervised and handcrafted features, whereas our freespace segmentation network can be trained and tuned independently, using a single FCN, while avoiding an additional fusion step. Lastly, we do not require a sliding window in our inference step by using an FCN and not a CNN.

The system presented in [82] also segments drivable regions in images, and adapts its modeling during driving. It relies on stereo disparity data for short-range analysis, and uses the generated labels of that process to extend the range of the algorithm. The authors call the real-time training 'dramatically' sensitive to small changes or imperfections in the labeling, so that the online supervision system must be error-free. The authors conclude that the method is a complicated process involving multiple ground-plane estimations, heuristics, etc., in order to generate training labels with little noise [82]. In contrast, our supervision process consists of a holistic, probabilistic model. Moreover, our training process can handle the noise in the labeling that results from adverse imaging conditions. The domain-adaptation strategy presented in [82] is limited to retraining the last layer of their neural network, whereas our system can perform a full retraining in real time, making it more flexible in handling different scenes. Besides, the system of [82] was validated in forest and grass scenes, using a small robot car with a low point of view towards the scene. These scenes typically contain large regions with a low intraclass variance (e.g., ground is all green grass or all brown leafs). In contrast, our dataset contains a wide variety of everyday traffic scenes.

4.2.5 Conclusions on the related work

From the glossary of related work in the above subsections, the following considerations can be distilled. Building upon latest successes in similar computer vision problems, we will utilize the power of FCNs for pixel-level scene segmentation with fast inference.

In addition, we continue with the Stixel World approach of the previous chapters, since it is an efficient method and inherently suitable for generating (imperfect/noisy) training masks, while making only generic non-limiting assumptions in its prior world model. These labels facilitate our self-supervised training strategy with the FCN. Furthermore, we make use of transfer learning in our online training approach to speed up our adaptation process.

The distinctive element of this chapter is that the system can rely on a small, efficient FCN, which is embedded in the design of a self-supervised online learning framework. Our online training is designed in such a way that it (1) offers more adaptive power than previous work, so that the system can handle more variation in the appearance of traffic scenes, and (2) is simultaneously able to handle more noise and imperfections in the automatically generated labels. This approach optimizes the trade-off between efficient execution and information capacity to handle uncommon but realistic, difficult cases. The next section presents our strategy to realize these points.

4.3 Method

This section describes the applied methodology for improved stereo-based freespace segmentation. Figure 4.1 provides a schematic overview of the proposed framework, which will be described in detail in the following subsections. First, the baseline FCN algorithm for image segmentation is described. After this, the self-supervised and the corresponding online training strategies of the FCN are explained in more detail.

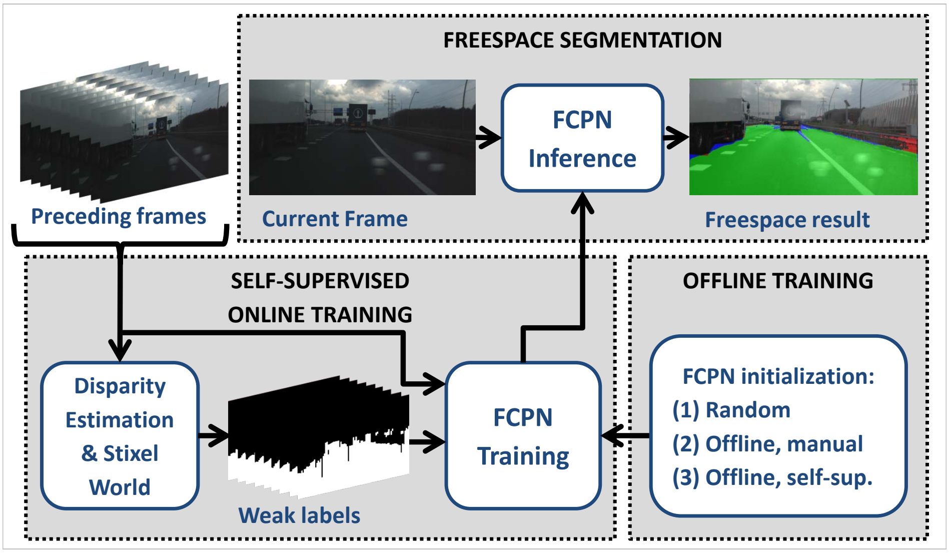


Figure 4.1 — *Schematic overview of our freespace segmentation method with online self-supervised training.*

4.3.1 Fully Convolutional Patch Network

The color-based segmentation algorithm used as a basis of our work is a Convolutional Patch Network (CPN) [67]. A Convolutional Patch Network can have the network structure of any CNN, as it differs only in the training strategy. Typically, CNNs are trained on full images, which means that the gradients of the backpropagation process are combined over all images in a batch. However, training on image patches instead of full images, was shown to both speed up the training process and lead to better results in this research context [67].

Provided that the context (road detection) and data (images captured from within a vehicle [55]) are comparable to our research, we adopt the FCPN architecture and the recommendations about the optimal training strategy, as presented in [67]. The network consists of several convolutional, max pooling and non-linear layers, as defined in Table 4.1.

Table 4.1 — *Specification of the layers within the FCPN architecture.*

Layer	Layer type	Size$^{(*)}$	Output area
No.		[pixels]	[data points]
	Input image patch	$28 \times 28 \times 3$	28×28
1	Convolutional	$7 \times 7 \times 12$	22×22
	+ maximum pooling + ReLU	$2 \times 2 \mid 2$	11×11
2	Convolutional + ReLU	$5 \times 5 \times 6$	7×7
3	Convolutional$^{(+)}$ + ReLU	$4 \times 4 \times 48$	1×1
4	Convolutional$^{(+)}$	$1 \times 1 \times 194$	1×1
	+ spatial prior$^{(\#)}$ + ReLU		
5	Convolutional$^{(+)}$ + tanh	$1 \times 1 \times 1$	1×1

$^{(*)}$ Definition of sizes:
- input image patch width and height with 3 color channels;
- conv. layer size is denoted as $w \times h \times n$, representing filter width, filter height and the number of filters;
- max. pooling layer size indicates kernel width, kernel height and stride.

$^{(\#)}$ The spatial prior adds two 1×1 filters that contain the normalized absolute position of the current patch within the input image.

$^{(+)}$ These are transformed fully connected layers.

The special feature of this network is the spatial prior, which is trained in the learning process as an integral part of the network, using the normalized positions of the training patches. This spatial prior exploits the spatial bias that is naturally present in road or freespace segmentation in traffic scenes.

As described in Section 4.2.1, CNNs can be transformed into FCNs [71] that are mathematically equivalent, but can be evaluated more efficiently during their inference stage. Since a patch network has a regular CNN architecture, the same holds for the network employed in this chapter, as described in [84]. Therefore, our system will rely on this Fully Convolutional Patch Network (FCPN) for color-based freespace segmentation. The freespace segmentation will rely on RGB data only, and not on RGB-D data (color combined with disparity). Section 4.3.3 provides the argumentation for taking this approach.

Hyperparameter selection

This subsection briefly presents the hyperparameters that are used in the training of the FCPN. Note that our current work is neither meant to offer an exhaustive test on optimizing the network architecture nor to optimize the hyperparameters of the training process. We acknowledge the fact that the results may be improved by investigating these aspects more properly, but the focus of this work is to show the feasibility of self-supervised training and the additional benefits of the proposed online tuning in the context of freespace segmentation. For completeness, the training parameters are included in Table 4.2. These parameters have been optimized with a tenfold cross-validation on the KITTI dataset for road detection

Table 4.2 — *Specification of the FCPN hyperparameters used during training.*

Data & loss		Regularization		Learning rate		Other	
Patches	28×28	L1	0.001	λ_0	0.004	Momentum	0.9
Batch size	48	L2	0.0005	γ	0.0005	Dropout	0
Loss	Quadratic			β	0.75		

by Brust *et al*. [67]. Amongst others, the table presents the values for the learning rate scheduling, of which the curve is specified by

$$\lambda(n) = \lambda_0 \cdot (1 + \gamma \cdot n)^{-\beta}, \tag{4.1}$$

where n is the value of the current training iteration count. Additionally, the training stage exploits *momentum*. This means that in each iteration, a certain fraction of the numerical gradient values in the previous iterations are added to the gradient values of the current iteration. This typically leads to faster convergence in the training stage.

4.3.2 Self-Supervised Training

The objective of employing a self-supervised training strategy for the FCPN is twofold: (1) alleviating the burden of manually labeling data samples, which is especially cumbersome for pixel-level segmentation tasks, and (2) facilitating online training to adapt the system while driving, in which case manual labeling is not possible.

Self-supervised training can be realized by using an algorithm that generates imperfect, yet reasonable training labels for the RGB images. The algorithm that generates these labels is chosen to be independent of the freespace segmentation process, exploiting an additional signal modality, namely stereo disparity. This disparity-based algorithm generates masks that indicate flat, drivable surface. Since these masks are estimates themselves and are not perfect, we consider them to represent *noisy* training labels.

Stereo disparity is an attractive modality, since it is computationally inexpensive and yet provides relevant information in the context of freespace detection. We propose to analyze the disparity signal with the disparity Stixel World algorithm. This algorithm is described in Section 2.2, and summarized briefly here for completeness.

The Stixel World algorithm is a probabilistic framework, which segments traffic scenes into vertically stacked, rectangular patches that are labeled as either ground or obstacle. The regularization within the Stixel World algorithm is mostly a-priori designed, exploiting the fact that disparity measurements facilitate metric/real-world reasoning, such as metric margins and gravity-model assumptions. By simplifying the representation of a scene into piecewise planar segments (flat for

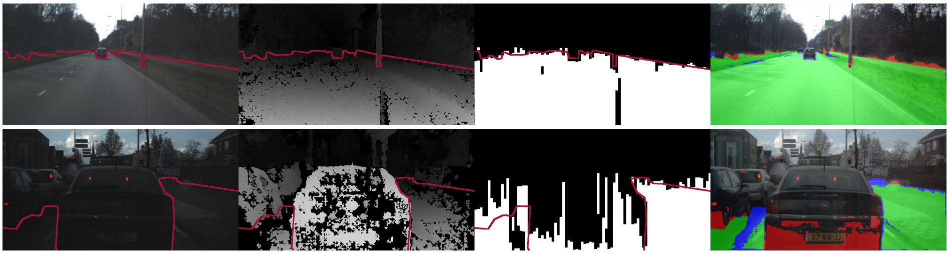


Figure 4.2 — *Illustrative challenges in our self-supervised training: RGB images (taken under adverse but realistic circumstances), their disparity estimates and the stixel-based ground masks (all with ground-truth reference). Errors in the disparity lead to imperfect masks, so that these should be considered as noisy labeling. The preview of our results (rightmost column) shows that we can adequately cope with the errors to a high degree.*

ground and fronto-parallel for obstacles), the segmentation can be formulated as a MAP estimation problem and can be solved efficiently using dynamic programming over columns with disparity measurements. The algorithm is highly parallel and can be executed in real time [51]. Our system adopts the improvements to the disparity Stixel World algorithm from [33] (namely: tuned transition probabilities, dynamic ground-plane expectation modeling and assuming a ground region below the field of view of the image). The disparity-based ground and obstacle masks subsequently serve as the noisy labels for the corresponding color image in our self-supervised training process.

Since this process can operate automatically, it can generate (noisy) labels for many frames. This facilitates training on images for which manual annotations are not available, so that the training set can be enlarged without much effort. This is an attractive property, since many related deep learning experiments have shown that training on more data can be beneficial (as briefly discussed in Chapter 1 and Section 4.2.2). The challenge is that the generated noisy labels will contain errors, as illustrated in Figure 4.2, potentially hampering the training process. Our system relies on the generalization power of the FCPN training process to adequately handle these inconsistencies in the labeling, which can be validated by comparing the results of our self-supervised training with the results of training on manually annotated frames. The latter annotations may have small imperfections on obstacle boundaries, but are overall significantly less noisy.

4.3.3 Strong versus weak versus indirect fusion

This subsection presents the considerations on the fusion strategy applied in this work. Similar to the previous two chapters, the work presented here concerns data fusion, namely fusion of color and disparity data streams. The method of Chapter 2 relied on strong fusion, using an algorithm that simultaneously analyzed color and disparity. The framework presented in Chapter 3 used weak fusion, where color models were learned from disparity-based masks and also balanced using disparity-based distance-aware processing. The work in this chapter relies on an FCPN that can be specifically designed to handle any kind of input. For instance, this facilitates strong fusion by simply feeding the network with RGB+D data, instead of RGB only. However, instead of relying on strong fusion, there are multiple reasons to generate the noisy training labeling on a different data modality than the signal modality that is analyzed by the FCPN for the freespace segmentation. These reasons are explained below.

First of all, strong fusion would re-introduce latency to the system in the inference step, since the FCPN would need to wait for the disparity signal to be estimated prior to analyzing the current image. This should be avoided to comply with the goals of this research, as stated in the introduction of this chapter. Although the inference time is not crucial during offline training stages, it is critical for ADAS operating in the real world.

Secondly, the current strategy increases the likelihood that the trained algorithm can deal with the unavoidable errors in the noisy labels, instead of inadvertently making the same faults in difficult situations. If the disparity data has artifacts that are labeled erroneously as obstacles, and that same disparity data is fed into the FCPN, then the FCPN would be at risk of replicating the false detections instead of repairing them. The reason is that typically the evidence in color images is less pronounced. In the worst case, we suspect that the FCPN would be able to mimic the noisy labels based on disparity and ignore color altogether, since they have conflicting information and the erroneous labels will match the disparity better than the color signal. However, this hypothesis has not been experimentally verified.

Thirdly, our current approach is more generic, since it could use different sources of self-supervised labeling instead of stereo disparity (like Lidar or monocular cues) without redesigning the network.

In conclusion, we opt for neither *strong* nor *weak* fusion. Instead, our framework uses *loose* or *indirect* fusion, since the data modalities are only linked indirectly via the generated training masks.

4.3.4 Online Training

This section briefly explains how our algorithm exploits the self-supervised learning strategy for online learning and its relation to transfer learning, as discussed in Section 4.2.3.

Since traffic scenes occur in a wide variety (urban versus rural, highway versus city-center), and with varying imaging conditions (good versus poor weather, day versus night), ADAS have to be both flexible and robust. A potential strategy is to train many different classifiers and to select the one that is most relevant at the moment (for instance, based on time and geographical location), or to train a complex single classifier to handle all cases. In contrast, the work in this chapter shows that it is feasible to fine-tune a relatively simple, single classifier in an online fashion. This is obtained by using the same self-supervised strategy as for offline learning, namely, based on generally correct segmentation by the disparity

Stixel World algorithm. This results in automatically improved robustness of the freespace detection, because the algorithm is adapted while driving.

The current framework adopts the same training strategy as presented in Chapter 3: the disparity signal is analyzed for several frames in a learning window (LW), and the resulting segmentation labels are exploited as training masks to train or fine-tune the FCPN.

A schematic overview of our experimental framework for freespace detection is shown in Figure 4.1. To show the effect of (within-context) transfer learning, an FCPN is trained from scratch (with random initialization), or the process starts with one of the offline trained models and tune the entire model with online data. Comparing these online strategies with results from solely offline training, shows the importance and added value of adapting the classifier online to the changing environment. Since this adaptation can be realized in a reliable and realistic way, the freespace segmentation system improves without putting extra effort and computational power into training and executing a larger, more advanced FCN.

Algorithm 4 summarizes and presents the steps of our online-training process for freespace segmentation, which can operate in the background of the segmentation process. The segmentation step itself consists solely of executing one forward/inference pass of the FCPN on the most recent image. In comparison to the Algorithms 1, 2 and 3 presented in the previous chapters, the current solution (1) does not depend on prior masks for the training step, (2) involves less conceptual steps such as distance-based weighting, (3) requires less preprocessing such as color-space transformations, and (4) has a more strictly decoupled dataflow, since the freespace segmentation uses color only, and hence, does not require disparity anymore.

Algorithm 4 Online Training of the FCPN

```
Input: Stereo images $I_{L,t}$ and $I_{R,t}$ of the learning window $LW$, initialized FCPN$_{\text{init}}$

  for each $\{I_{L,t}, I_{R,t}\} \in LW$ do
      $D_t \leftarrow$ EstimateDisparity$(I_{L,t}, I_{R,t})$
      $L^*_t \leftarrow$ EstimateDisparityStixels$(D_t)$
      $TM^{\text{f}}_t \leftarrow$ GenerateFreespaceTrainingMask$(L^*_t)$
  end for

  FPCN$_t \leftarrow$ FPCN$_{\text{init}}$
  for $n : 1 \ldots N_{\text{iterations}}$ do
      $X_{\text{patch}} \leftarrow$ SelectLabeledImagePatch$(LW, TM)$
      FPCN$_t \leftarrow$ PerformTrainingStep(FPCN$_t, X_{\text{patch}}$)
  end for
Output: Fine-tuned network FPCN$_t$
```

Although our system does not yet operate in a real-time prototype, we deem this to be feasible in the near future, as (1) the Stixel World system can execute in

Table 4.3 — *Comparison of sizes of commonly used neural networks*

name	layers	parameters$^{(*)}$
AlexNet [68]	5	60M
VGG-19 [85]	19	138M
Inception v1 [86]	22	5M
ResNet-50 [87]	50	25M
FCPN [67]	5	27k

$^{(*)}$ M represents millions, k indicates thousands.

real time, (2) our FCPN is small, which allows for fast inference and fast training, and (3) relies on patch-based training, thereby facilitating fast training as well. A compact overview of commonly used neural network architectures is provided in Table 4.3, indicating their size in the number of parameters. The used FCPN is three orders of magnitude smaller than these commonly used examples, supporting our expectation that real-time execution is feasible.

4.4 Evaluation strategy

This section presents the data and strategy for evaluating the proposed algorithm.

4.4.1 Datasets

The publicly available data EHV-road14 [33] and EHV-road15 [34] are utilized as the training set for our offline training of the FCPN. When combined, the total training data consists of 188 frames with manual annotations of freespace. Additionally, the 10 preceding frames of each annotated frame are available, albeit without annotations. The current work comes with a newly annotated dataset that is employed as (unseen) test set. It was captured in the same configuration and context as in [33], [34] (different parts of the same data-gathering drive), and is publicly available online[1]. This test data consists of 265 manually annotated frames of urban and highway traffic scenes, both under good and adverse imaging conditions. There is a large variety in scenes, covering crowded city centers, small streets, large road crossings, road-repair sites, parking lots, roundabouts, highway ramps/exits, and overpasses. To facilitate the online learning process, the 10 preceding frames of each annotated frame are provided as well (without manual labeling). The RGB frames were captured with a Bumblebee2 stereo camera mounted behind the windshield of a car. Both raw and rectified frames are available, accompanied with our disparity images. These were estimated using OpenCV's Semi-Global Block-Matcher algorithm with the same settings as used in Chapter 3. To the best of our knowledge at the time of writing, these publicly avail-

[1] The test data can be found at `http://www.tue-mps.org`

able and annotated datasets are unique in the aspects that they (1) readily provide preceding frames that are required to perform and evaluate online learning, (2) consist of *color* stereo frames which facilitate our self-supervised training methods, and (3) contain example scenes which are captured under adverse and/or under favorable imaging conditions.

4.4.2 FCPN implementation setup

All our experiments regarding the FCPN are executed using the software library CN24 [67]. The library has been made available by the Computer Vision Group at Jena University, and can be found online[2]. It comes with example scripts that were used for the reported KITTI and LabelMe-Facade experiments. Internally, the CN24 library transforms any fully connected layer within a CNN architecture into its convolutional equivalent counterpart [84]. The library applies a similar upsampling strategy as the skip layers of [71] to produce full-resolution pixel-level confidence or segmentation masks.

4.4.3 Scoring metrics

The quality of the freespace segmentation is assessed using the pixel metrics as employed for the KITTI dataset [55], since this benchmark and its metrics are widely used in the community and are briefly described here for completeness. The segmentation performance is measured by calculating the recall, precision and F_1 score (the harmonic mean of recall and precision) in a Birds-Eye-View projection (BEV) of the image. Additionally, since our FCPNs provide confidence maps, the maps need to be thresholded prior to comparing them to the binary ground-truth annotations. This is resolved by selecting the threshold that maximizes the F_1 score, giving F_{max}. The metric F_{max} indicates the optimal performance setting of the algorithm. Additionally, the average precision AP is calculated, which captures the precision score over the full range of recall. The combination of AP and F_{max} provides a balanced view on the algorithm performance.

4.4.4 Experiments

This section presents our experiments conducted for measuring specific aspects. Firstly, the key experiment involves comparing supervised and self-supervised learning of the FCPN and secondly, the comparison of online versus offline training. The experiments additionally provide a reference with general 3D modeling methods that do not rely on deep learning. To this end, the results are compared against both the disparity Stixel World algorithm as presented in [51] (including the improvements mentioned in [33]), and the distance-aware color Stixel World algorithm as presented in the previous chapter. The latter also applies online color modeling, but relies on traditional computer vision methods instead of deep learning.

[2] The CN24 code is available at `https://github.com/cvjena/cn24`

A. Experiment 1: supervised versus self-supervised training

To validate the feasibility of the self-supervised FCPN training method, three FCPNs are compared, which have an equal architecture but are trained with different data and/or labels. The first model FCPN$_{\text{off,man}}$ is trained offline on manually annotated labels, as a reference result for offline, supervised training. This model replicates the approach of Brust *et al.* [67]. They successfully validated their road-segmentation algorithm on the KITTI-road dataset [55], so it is interesting to evaluate if its performance on the EHV-road dataset is of similar quality. The second model FCPN$_{\text{off,self}}$ is trained offline on the same frames as FCPN$_{\text{off,man}}$, but now using automatically generated noisy labels instead of the clean, error-free manual versions. This model serves as a demonstration of offline, self-supervised training. Thirdly, an FCPN model is trained in a self-supervised fashion on *all* available frames in the training set, including the earlier mentioned preceding frames for which no manual labels are available (FCPN$_{\text{off,self-all}}$). This experiment tests the added value of training on additional data in our framework, which is realized efficiently because of the initial choice of fully self-supervised training.

B. Experiment 2: offline versus online training

Two key experiments are performed to test the benefits of online training for the FCPN-based freespace segmentation and are compared to the offline experiments of the above subsection. Similar to Experiment 1, FCPNs are trained on different data, while their architectures are kept identical. Firstly, an FCPN is trained from scratch (with random initialization) on the noisy labeled preceding frames of each test frame, resulting in an FCPN$_{\text{onl,scr}}$ for each test sequence. Additional experiments validate the benefits of online tuning. To this end, the network is initialized for each training sequence with one of the offline trained models (trained on either manual or self-supervised labels), resulting in an FCPN$_{\text{onl,tun-man}}$ and an FCPN$_{\text{onl,tun-self}}$ for each test sequence. Note that the labels for the online training itself are always self-supervised, since the preceding frames of each sequence are not manually annotated. This constraint follows naturally from the fact that the system is designed to operate during driving.

Moreover, an experiment is performed to further analyze our online training method. Specifically, the power and benefit of '*over-tuning*' our framework is illustrated. To this end, the online trained FCPNs are evaluated on test frames of different, unseen sequences. These experiments investigate the extent to which the online trained FCPNs are tuned to their specific sequence. If the FCPNs are over-tuned, they are expected to perform well if the training sequence and the test frame are aligned, but simultaneously they are expected to perform poorly when they are misaligned. To validate this, three different misalignment experiments are conducted: (1) shift one training sequence ahead, (2) shift one training sequence back, and (3) carry out a random permutation of all training sequences. Note that our data sequences are ordered in time, therefore, there can still be correlation between training sequences and test frames when shifting backward or forward with a single training sequence. Moreover, the shifting experiment will affect some frames more than others, since the intervals between the different sequences are different. These correlations are reduced as much as possible by the full permutation experiment.

In total, the evaluation contains eight different methods and the miss-alignment experiment on the challenging, real-world data set, that has been described in Section 4.4.1.

4.5 Results

This section describes and discusses the results of the experiments described above. The main findings are first presented qualitatively, which is followed by several subsections providing more detail on specific aspects.

4.5.1 Qualitative results

Figure 4.3 shows qualitative results of the experiments. The first column contains the input color image (with ground-truth freespace annotation) and the second column the results of the disparity Stixel World baseline. The third and fourth columns show results of our offline and online FCPN-based methods, respectively. In the top-three images, the offline-trained FCPN detects less false obstacles than the Stixel World baseline. However, it performs worse in several important cases: it misses obstacles (such as the poles in the fourth row, the cyclist in the fifth row) and also classifies a canal as drivable area (sixth row). In contrast, all-but-one images show that our online trained FCPN outperforms both the Stixel World baseline and the offline training strategy. It segments the scene with raindrops on the car windshield robustly, while also the other results are more accurate. The image in the fourth row visualizes an erroneous case: the online trained FCPN does not detect the concrete poles, although they are present in the training masks.

4.5.2 Main quantitative results

For a more in-depth analysis, quantitative results are provided as well. The recall-precision curves of the main experiments are provided by Figure 4.4, where the left graph shows the full range and the right graph provides an enlarged view of the top-right region (closest to the point of optimal performance).

A. Results on offline learning

The experiments with offline learning offer several interesting insights. First, the results of supervised (manual labels) and self-supervised (disparity-based labels) are nearly identical. This confirms the feasibility of self-supervised learning, since relying on noisy labels does not hamper the performance of our system. Self-supervised training on more data did not lead to a clear improvement of the results in our experiments, as illustrated by the graph (yellow performs similarly to red). This may indicate either that our network is too small to exploit the additional data, or that the correlation within the new samples is too high to be

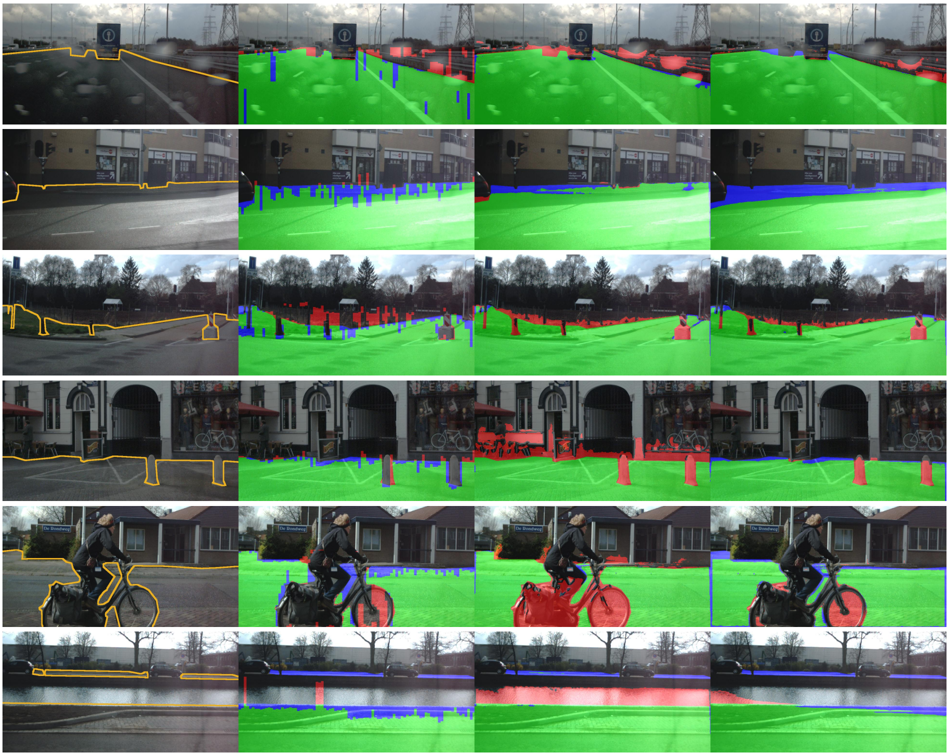


Figure 4.3 — *Qualitative results on the frames shown in the leftmost column with hand-annotated freespace boundary. In the next columns, from left to right: disparity Stixel World baseline; our FCPN result with offline training on manual labels; our FCPN result with online tuning. Colors indicate the freespace segmentation true positives (green), false negatives (blue) and false positives (red). Best viewed in color.*

informative. Another noteworthy result is that the offline-trained FCPN, even with the error-free manual annotations, does not outperform the traditional Stixel World methods. This experiment in fact replicates the method of [67], which was successful on the KITTI-road dataset, while retrained on our data. This observation shows that our EHV-road dataset is indeed more difficult than the KITTI-road dataset. This is likely due to our inclusion of images under adverse conditions and the larger variety in road curvature. The latter specifically has a negative impact on the consistency in the spatial prior in the FCPN.

B. Results on online learning

Considering our online training strategies, Figure 4.4 clearly shows that these outperform the offline training strategies over the full range of recall, thereby confirming the qualitative results of Figure 4.3. Most importantly, note that the online tuned FCPNs now outperform the Stixel World baseline methods, which is not the case for the FCPNs that relied solely on offline training. This highlights

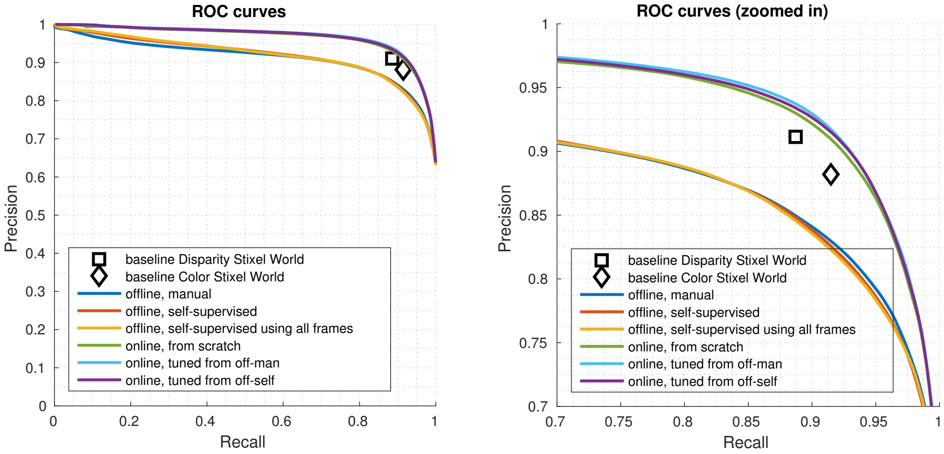


Figure 4.4 — *Recall-precision curves of our experiments after 10,000 training iterations. This figure is best viewed in color. The ROC curves happen to be clustered into two groups, since several graphs closely overlap. One cluster consists of the blue, red and yellow graphs, the other one contains the green, cyan and purple graphs.*

the added value and effectiveness of our online tuning framework. The online method is sufficiently robust under difficult yet realistic environmental conditions, thereby enabling the use of a small and efficient method (the small network of [67]). Sections 4.5.3 through 4.5.5 discuss several aspects of the online training process in more depth.

C. Results on different data subsets

The employed test set contains both sequences captured under favorabe and under adverse imaging conditions. This subsection explicitly emphasizes the effect of these conditions briefly.

Segmenting the regular images is mostly solved by the disparity Stixel World method, so that there is little to gain with additional processing. These images are included in the set to avoid over-fitting on e.g. rainy conditions, and to show that also the regular conditions can be handled by our algorithm. To provide this balanced view, the scores in all graphs and tables are calculated on the complete test set. As a result, the gains of several reported experiments in this chapter may seem small from a quantitative point of view. However, the overview in Table 4.4 shows that the gain of our processing is larger on the subset of rainy images (4.2%). Moreover, note that these scores are evaluated over raw pixels, using the metrics from the KITTI benchmark. Although the errors due to rain and other poor conditions are limited in pixel count, they are important robustness improvements for real-world applicability. The qualitative examples of Figure 4.3 illustrate this claim.

Table 4.4 — *Performance on different subsets (F_{max} scores averaged over frames)*

Method	Regular frames	Rainy frames
Disparity Stixel World	0.887 ± 0.073	0.867 ± 0.069
FCPN$_{\text{onl,tun-self}}$	0.896 ± 0.088	0.909 ± 0.050

4.5.3 Analysis of online training convergence

The trends of our quantitative results over the number of training iterations are shown in Figure 4.5. The training converges after 5,000 to 10,000 iterations and the visible trends are consistent with the ROC curves in Figure 4.4. Specifically, offline training is outperformed by online training and online tuning performs slightly yet consistently better than online training from scratch. An important conclusion of the experiments is that the contribution of online-tuned training is most significant in the speed of convergence, and less relevant for the final result after convergence. More specifically, the models that exploit online tuning outperform offline methods and the baseline already after 100 iterations of training (which takes less than half a second on a GeForce GTX970 graphics card), whereas models trained from scratch need at least 500 iterations to match the offline FCPN and more than 2000 to exceed the Stixel World algorithm.

Figure 4.6 portrays the convergence of the freespace-confidence maps of these two methods. This exhibits the same trend, where the confidence maps produced after 100 and 500 training iterations by FCPN$_{\text{onl,scr}}$ are overall more gray, showing that there is not yet distinction between free or occupied space. Oppositely, the confidence maps of FCPN$_{\text{onl,tun-self}}$ are already more discriminative at an early training stage. This is visible from the higher contrast in the bottom-left image when compared to the top-left image, and from the smaller changes from left to right in the bottom row, when compared to the top row. Note that both final freespace-confidence maps are not perfect in this case, since, ideally, the whole car directly at the center of the image should be black (as it is not freespace).

4.5.4 Analysis of online training settings

Two different experiments have been conducted to asses the robustness of our online training strategy around the FCPN. The first test analyzes the drop in performance as a function of the delay between the frames on which the online tuning is performed and the frame under analysis. The left graph in Figure 4.7 shows that the score drops only about 2% with a delay of 2.5 seconds.

The second test validates the influence of the number of FCPN layers that are tuned online on the freespace segmentation result. The network has 5 layers with parameters, so a comparison is made between tuning all layers (regular online tuning) and tuning only the last 4, 3, 2 or 1 layer(s). Keeping all layers static is the

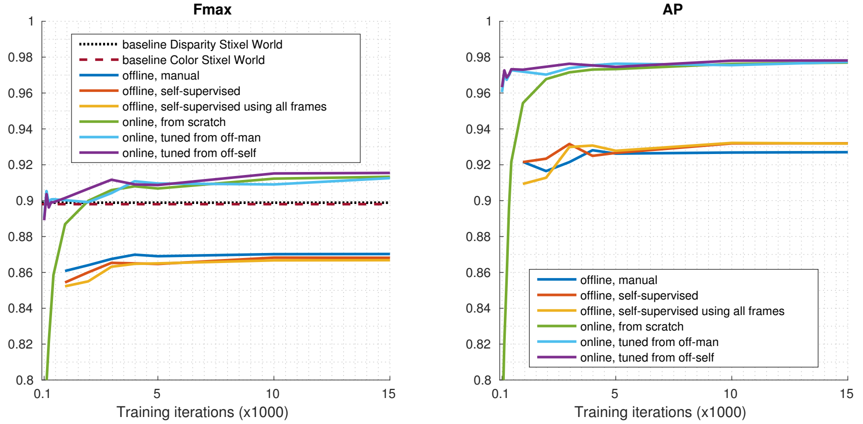


Figure 4.5 — *Convergence of scoring metrics F_{max} and AP over FCPN training iterations. This figure is best viewed in color.*

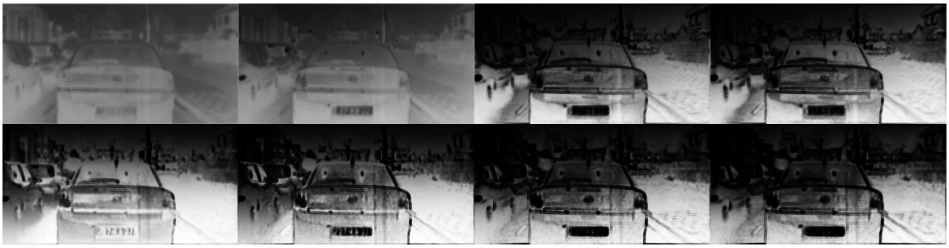


Figure 4.6 — *Illustration of convergence speed for online FCPN training methods. Top row: $\text{FCPN}_{onl,scr}$; bottom row: $\text{FCPN}_{onl,tun\text{-}self}$; both after 100, 500, 5,000 and 10,000 iterations. Although both networks end up with similar confidence maps, initialization with a pre-trained network clearly speeds up the convergence. RGB input image is shown in the second row of Fig. 4.2.*

offline training reference. The right graph in Figure 4.7 shows that tuning only the final layer provides results within 1.5% of the full-tuning approach. Both these experiments provide tradeoffs between tuning time and performance quality.

4.5.5 Analysis of online training with over-tuning

The results of the misalignment of the training sequences and the test frames with the online-trained FCPNs are provided in Table 4.5. It is clear that the misalignment has a negative impact on the performance of the online training approach, as was to be expected. The scores drop even below the scores of the offline-trained models, also for the FCPNs that were initialized with offline pre-training. Since the online-trained FCPNs outperform all other methods when their training sequence and test frame are aligned, this validates our claim that the online training is giving the system flexibility to adapt to new circumstances, and that over-tuning can be exploited beneficially in the context of freespace segmentation for ADAS.

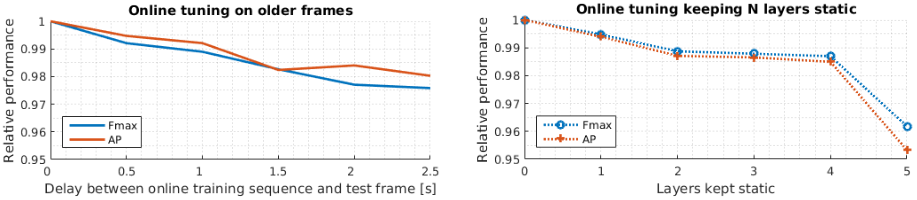


Figure 4.7 — *Two different experiments on robustness of our online tuning strategy. Left: tuning on older frames. Right: tuning only the last couple of layers, with the first N static (N=0: tune all layers, N=5: no tuning; same as offline training only).*

Table 4.5 — *Results of training FCPNs online on different training sequences: aligned (normal), one sequence back or ahead in the (ordered) dataset (+1/-1), or with random permutation. The drop in performance illustrates the adaptive power of the online tuning.*

	Offline	Online training FCPN$_{\text{scratch}}$			Online tuning FCPN$_{\text{off-self}}$		
	(manual)	Normal	+1/-1	Random	Normal	+1/-1	Random
F_{max}	0.87	0.91	0.83	0.79	0.92	0.83	0.80
AP	0.93	0.98	0.91	0.84	0.98	0.92	0.86

While the offline FCPNs give a good average generalization over the complete training set, the online FCPNs specialize on a particular subset of the training set. This hampers their performance on the complete set, however, this is actually beneficial for the task of freespace segmentation for online adaptive ADAS.

4.6 Conclusions

The work in this chapter shows that Fully Convolutional Patch Networks can be trained in a self-supervised fashion in the context of freespace segmentation for ADAS. The segmentation results are comparable to a conventional supervised strategy that relies on manually annotated training samples. Furthermore, we have extended this result to accomplish that it also facilitates *online* training of a segmentation algorithm. As a result of this approach, the freespace analysis becomes highly adaptive to any traffic scene that the vehicle encounters. This provides an increased robustness, while the size of the implemented network is orders of magnitude smaller than previous systems.

With respect to the applied method of this chapter, we leverage the benefits of applying a neural network for exploiting color modeling. The power of this new approach is that the algorithm now directly feeds the color image signal into the segmentation network, without the previously additionally required depth-aware color modeling. The network architecture includes a spatial prior, referring to the distribution of class locations over the image, to exploit knowledge on the varying scene layout. In conclusion, the network directly learns the nature of the perspective camera imaging and is able to correctly segment the freespace.

The evaluation of our system provides the following key results:

A. Self-supervised training: Self-supervised training for freespace segmentation is feasible when relying on imperfect, noisy labels generated with the disparity Stixel World algorithm. Quantitatively, it leads to similar results ($F_{\text{max}} = 0.865$ and $AP = 0.932$) as when relying on manual annotations ($F_{\text{max}} = 0.870$ and $AP = 0.928$), so that our method can operate successfully without manual labels.

B. Online training: All presented online methods ($F_{\text{max}} \simeq 0.92$ and $AP \simeq 0.98$) outperform the offline-only methods mentioned above. More importantly, without online training the FCPN performs worse than the baseline ($F_{\text{max}} = 0.90$) on our data. This indicates that our strategy of online training is a good and efficient proposal to leverage a small neural network by quickly adapting it to varying imaging conditions.

C. Imaging conditions: The added value of our online framework is most pronounced in the rainy-images subset of the data, where it outperforms the baseline with 4.2%.

D. Pre-training/online tuning: Online training an FCPN from scratch or online tuning from a pre-trained FCPN leads to the same segmentation performance. However, the pre-trained network converges 5 times faster, thereby making it feasible for real-time deployment.

Overall, the experiments show that the online training enhances the performance with 5% when compared to offline training, both for F_{max} and AP. This can be explained by the fact that, due to our adaptation strategy, the system is not required to generalize to a large amount of traffic scenes with a single detector. Hence, the detector can -and should- be 'over-tuned' on currently relevant data. In turn, this benefits the use of a small FCPN whose training converges fast enough to facilitate real-time operation. In conclusion, we have presented a system with a small memory footprint and short inference time, that is still able to handle a broad variety of scenes, without the need of manual labeling and without requiring disparity in its critical path.

This concludes our work on freespace segmentation, which is a fundamental low-level processing step for ADAS, which aims at supporting safe navigation through everyday traffic. It provides a drivable ground region of the traffic scene in front of the vehicle. In other words, it has a focus on the static part of this problem. However, traffic scenes are a highly dynamic environment and imaging setting. By incorporating dynamics, and extending the analysis only from the ground to include objects as well, ADAS can potentially provide complimentary relevant information. For instance, this relates to system questions, such as what space is free in the next couple of seconds, or, what object lies on a potential collision course with the ego-vehicle. Timely answers on such questions would allow for live warning and even future path planning. Therefore, to extend the freespace analysis to a higher level of system operation, the next chapter will continue with investigating a collision warning algorithm.

Chapter 5 ASTEROIDS for collision warning

5.1 Introduction

The previous three chapters of this thesis have addressed strategies for freespace versus obstacle segmentation in color images captured with a stereo camera. In the presented approaches, each frame is treated as a static scene. The adaptive freespace segmentation methods consider the continuously changing scenery, but not the dynamics of individual, moving objects. However, traffic forms highly dynamic surroundings. When the current freespace is known, the next step is to predict which space is free in the future, and, more importantly, where to avoid potential collisions. The general objective of ADAS is to reduce traffic accidents, predominantly by avoiding or mitigating collisions. This requires detecting potential collisions accurately and timely, irrespective of whether the avoidance will be executed by a human driver or automatically by a follow-up system.

The discussion on collision warning systems for ADAS technology in Chapter 1, which will be extended in this chapter, can be summarized as follows. Current collision prediction systems operate generally under one or several of the following limitations:

- restricting analysis to the ego-lane in highway scenarios only, thereby ignoring intersections and crossing or oncoming traffic [88];
- relying on pretrained pattern recognition, so that the system can only handle traffic situations [89] or object classes [90], [91] that were a-priori known and already available during training;
- leveraging high-level knowledge, which improves prediction quality when it is reliably available. However, it puts difficult requirements on the infrastructure [21] and/or other participants. Examples of infrastructure requirements are having up-to-date HD maps and exploiting V2X communication and/or centralized roadside compute [92]–[94]. Alternatively, a requirement

The work in this chapter has been presented at IS&T EI-AVM 2019 [40] (receiving a best-paper award) and was published in IEEE Trans.-IV [41].

on other participants is e.g., V2V communication about navigation intent. These constraints do not scale to all possible types of obstacles.

A key takeaway from the current state of the art is the importance of a multi-sensor setup to (1) provide redundancy on safety-critical system aspects to reduce the effect of sensor malfunctioning, or to remove blind spots in the perception of the surroundings, and (2) exploit different data modalities that, when combined, provide all relevant information [21], [22]. These important aspects will be briefly further elaborated upon below.

Providing redundancy: The current work should alleviate the limitations of existing systems, which have been described above. This leads to some system requirements, referring to redundancy and safety. The system should be independent of high-level knowledge, for instance from HD maps or V2X communication, which is not part of the ego-vehicle and not always available in the infrastructure. Moreover, the system should be independent of the obstacle type or collision scenario. This implies a generic solution for collision warning with ADAS.

Exploiting data modalities: We develop a generic collision warning system using a stereo video camera. Stereo cameras are increasingly employed in cars with ADAS, mainly for high-level semantic reasoning and scene-geometry estimation. This makes it relevant to investigate the level of feasibility of stereo vision to generate collision warnings in a generic way. For that reason, this work explores stereo imaging, while aiming at an algorithm that facilitates sensor fusion in the near future.

To further detail the analysis of stereo video for our purpose, our research has considered the strong gain in momentum of the Stixel World algorithm for efficient automotive vision analysis. Originally, it addressed representing scene geometry efficiently from disparity data [95], [96] (see Section 2.2). Additionally, it has a proven value for different subdomains, such as semantic scene segmentation [97], object detection [98] and online training supervision [34], [39]. Given this broad promising range of Stixel World applications, the work in this chapter extends this concept even further and explores how to extract relevant collision warning information, starting from the bare disparity stixels. We aim at designing a generic method, so that it can always benefit from the more advanced versions of the Stixel World proposals under development, e.g. that realize object clustering or find semantic labels.

The above discussion leads to the three following problem statements for this chapter.

- *Extending the static stixels with dynamic trajectory information:* The approach that is presented in this chapter relies on an external motion estimation method, the results of which are fused into the stixel presentation. Additionally, stixels need to be tracked over time, so that current stixels need to be associated to stixels of previous time steps. This concept is referred to as dynamic stixels.

- *Incorporating measurement and modeling uncertainties:* data measurements are performed to account for uncertainties in the total decision making. The aim is to provide confidences throughout the system pipeline. In this way, the final decision obtains a nuanced nature. Since we build upon the stixel representation, a key question is the appropriate modeling of the related confidences.

- *Performing a probabilistic collision data analysis:* Essentially, our algorithm generates probabilistic collision data by sampling so-called asteroids from these dynamic stixels. The result of this is then fed into the measurement update step of a probabilistic Bayesian histogram-filter, which resolves around a specifically designed state-space representation.

The remainder of this chapter is organized as follows. Section 5.2 presents an overview of related work in the context of ADAS for collision warning, providing similarities or differences with the proposed work. Section 5.3 describes the design choices and the corresponding high-level system architecture of our method, which is followed by an in-depth view of the most relevant processing blocks in Section 5.4. The evaluation strategy and results are presented in Section 5.5 and 5.6, respectively. Section 5.7 concludes the presented research.

5.2 Related work

The key elements of the research in this chapter are exploiting stixels for generic collision warning, the use of non-static video cameras for ADAS, and video analysis for moving object detection, tracking and path prediction and subsequent modeling in a state space. These objects can involve many and multiple types of traffic participants (such as cars, pedestrians, cyclists, buses, etc.) which can pass closeby to the ego-vehicle (e.g. the ADAS-equipped car), at maximum speeds of around 50 km/h. This section incorporates these key elements.

A. The Stixel World: Stixels are vertical superpixels with fixed pixel width, which are produced by analyzing disparity data with the Stixel World algorithm [96]. This algorithm processes the data in a column-based manner and divides the scene into either ground or fronto-parallel, rectangular obstacle patches, which are assigned a single disparity value. This forms an efficient representation of the scene geometry and has a proven value for different subdomains. For instance, the disparity Stixel World has been fused with deep neural nets for both semantic scene segmentation [97] and instance segmentation [99], where stixels have been also clustered to detect and recognize objects [98]. Additionally, the Stixel World analysis can provide a supervisory function in an online training setup for free-space segmentation [34], [39]. The work in this chapter extends this broad promising range of applications even further by exploring the strengths and weaknesses of a stixel-based approach to extract collision warning information.

B. Collision warning systems: In related work on collision warning systems, several limitations can be observed that our strategy mitigates or avoids altogether. First of all, most current systems are limited to highway scenarios [88], [92], [93]. Although those can operate at higher vehicle speeds, the systems will not be able to deal with street crossings, non-vehicle traffic or oncoming traffic, which is not a fundamental limitation in our method.

Second, most collision warning systems rely on vision with trained pattern recognition. For instance, a MobilEye system will only recognize cars, trucks, motorcycles, cyclists and pedestrians, with the additional limitation to fully visible rear-ends for vehicle detection [91]. Similarly, the system of Cherng *et al.* classifies situations into five pre-defined dangerous motions that are limited to the ego-direction (such as cut-ins). Moreover, it can handle only regularly-sized cars, just one of which may be in view in a scenario [89]. Both these approaches rule out handling crossing, oncoming, and passing traffic, in contrast to our more generic algorithm.

The mono-camera based system of Ess *et al.* deploys several class-specific detectors, for instance for cars and pedestrians. Subsequently, they rely on class-specific motion models to predict object trajectories for enhanced accuracy [90]. Similarly, in research towards protecting Vulnerable Road Users (VRUs), successful systems rely on modeling pedestrians and their context in high detail, such as analyzing head poses to estimate navigational intent [100]. In contrast, the algorithm in [101] tracks generic object proposals, and only afterwards tries to infer a semantic labels, if desired and possible. The rational behind this strategy is that is is infeasible to train a specific detector for all potential classes one can encounter. From a similar starting point, our system can handle any tangible object, without knowing its type. Additionally, our algorithm uses super pixels to operate at a medium-level representation, not on an object level, since we purely aim at generating collision warnings. This aspect makes the system more robust and more widely applicable than most alternative approaches, since it is not limited to the set of objects for which it was trained.

Thirdly, other previous work addresses free-space detection (the area in front of the vehicle where it can drive) [33], [34], [39], [47], which is a related or even the dual problem of collision warning. The currently proposed method explicitly incorporates motion estimation, motion prediction and timing in the system and analyzes the *obstacle* part of the scene instead of the *ground* part. This extends the analysis to dynamic data instead of using only static data.

C. Motion modeling and tracking: Since our framework concerns tracking elements over time and predicting their future path, a motion model and a data-association strategy should be selected. Models for motion are available at different levels of complexity, varying by the incorporation of steering angles, yaw rate, acceleration and velocity [102]. These can also be employed in parallel and fused afterwards to handle cluttered measurements in highly dynamic urban environments [103], and are commonly leveraged for intent prediction for vulnerable road users such as pedestrians [104], [105]. Since we aim at a class-agnostic analysis and execute at a medium-level stixel representation, and not on object level, we do not model model high-order motion dynamics or navigation intent. Instead in this work, we use simple constant-velocity kinematics without any rotational component and rely on the strength of having multiple stixels per object.

Regarding the problem of data association for tracking, we propose a strategy similar to the extended SORT algorithm [106], which is a box-overlap analysis, enhanced with appearance modeling. In contrast to [106], we simplify the appearance encoding into a color histogram and calculate similarity based on the Bhattacharyya coefficient, an approach used more often in multi-object tracking algorithms [107]. The benefit of our low-level histogram approach is that it does neither require training on class-specific examples nor has to execute a neural network during the association process.

D. State space: Systems in literature typically use geometry-oriented state spaces, for instance by storing locations in occupancy grids that are updated and refined over time [108]–[110], and then for collision analysis derive motion as a secondary signal [108], [109]. In contrast, our design of the state space directly stores the relevant information for or use-case, namely time-to-collision and angle-of-impact. This is in line with the objective of designing a collision warning system. For further reading on state spaces we refer to the book of Thrun *et al.* [111].

Summarizing, we focus our design on an urban setting with medium driving speeds, with nearby traffic and obstacles. In contrast to collision warning systems presented in related work, we do not limit ourselves to specific classes of objects or types of scenarios and aim at generic collision cases and broad usage. To further generalize, we avoid relying on semantic information on traffic layout or participant intentions and restrict ourselves in this work to affordable sensor hardware without V2V or V2I communication infrastructure. Our algorithm exploits the basic disparity stixels. However, it can always benefit from the more advanced versions of the Stixel World proposals under development, e.g. with object clustering or semantic labels, since it is designed as a generic method. Finally, we design a new state space that directly models the quantities of interest, namely angle and time of impact.

5.3 High-level system architecture

This section explains the key concepts and design choices that are underlying the high-level system architecture. First of all, a main challenge when working with stereo disparity data is that it tends to be noisy in general, and missing or erroneous on low-texture image regions, such as surfaces of smooth road or shiny cars. The stixel representation addresses some of these aspects, but at the cost of spatial quantization, due to the limited disparity resolution and fixed horizontal grid. This, in turn, conflicts with smooth, fine-grained tracking of obstacles over time. Given these kinds of challenges, a typical approach is to employ a proba-

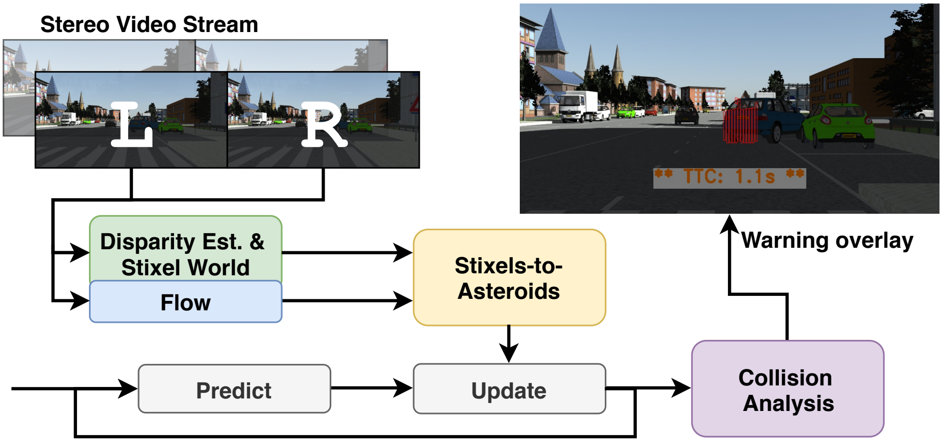


Figure 5.1 — *High-level schematic overview of our collision warning system. It extracts flow and disparity from stereo video and generates asteroids from stixels to analyze potential collisions.*

bilistic processing pipeline. This facilitates maintaining any information in the system, no matter how uncertain, for as long as possible. The proposed probabilistic processing contains various elements. The core element is a Bayesian filter, which is divided in a predict and an update stage. This filter operates within the boundaries of a state space, which contains the probability of a collision with the ego-vehicle from a certain angle at a certain time-to-collision. The use of stixels is translated into a probabilistic representation to provide the measurement input to the update stage of the Bayesian filter. This data flow is depicted in Figure 5.1, also showing a Collision Analysis module that interprets the state and generates warnings accordingly.

The state space and the three high-level processing blocks are described in the following subsections.

5.3.1 State-space representation of collision data

Since the goal of our system is to provide collision warnings, we introduce a state-space design over multiple dimensions that is directly suited to address such warning information. A histogram offers an efficient yet flexible representation of data with multiple dimensions relevant to collision warning, like angle of impact and time to collision. It can represent multi-modal distributions directly without enforcing high-level assumptions on the modeled data, which suits our aim of providing class-agnostic collision warnings. For implementation, we define a three-dimensional state space, the axes of which are *time-to-collision* (*ttc*), *angle-of-impact* (*aoi*) and *collision* (*col*). Figure 5.2 shows a schematic visualization.

The system, as described in the figure, monitors such a state space for the five sides-of-impact (s.o.i.) of the vehicle. The current work has a focus on the frontal view, since that is within the field of view of the sensor setup. The time

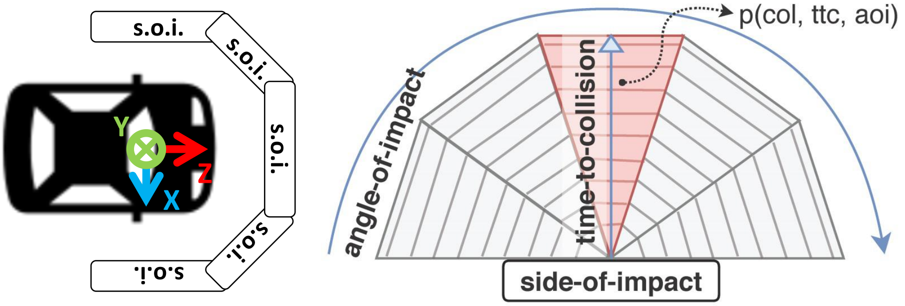


Figure 5.2 — *Schematic visualization of the sides of impact (s.o.i.) around the ego-vehicle with the employed coordinate system (left) and the state space with discretized angle-of-impact, time-to-collision and $p(col, ttc, aoi)$ versus $p(\neg col, ttc, aoi)$ for a single side of impact (right). Our evaluation is focused on the area highlighted in red for the front side of the ego-vehicle.*

axis is discretized with steps of half the sample time of the input data stream (this equals 0.05 seconds) to a maximum of 5 seconds, and the angle-of-impact is split uniformly in five non-overlapping ranges of $36°$ each. To obtain a complete joint probability distribution, the algorithm calculates the belief in *no collision* $p(\neg col, ttc, aoi)$ and the *collision belief* $p(col, ttc, aoi)$ for each angle and time pair.

5.3.2 Bayesian filter: prediction

The prediction step of the Bayesian histogram filter in our system is straightforward due to the design of the state space: the entire space can be shifted over the amount of bins along the time-to-collision axis, corresponding with the sampling rate of the camera. Additionally, a normalized box-averaging filter is applied with the same aperture as the shift. This filter introduces a dispersion of the belief to reflect the uncertainty in the prediction step, *i.e.* the process noise.

5.3.3 Bayesian filter: measurement update

The principal stage of our Bayesian histogram filter is the measurement update and consists of several steps, which will be detailed in the next section. Here, first the high-level concept is presented. The aim is to convert the stereo video data at the input via stixel and asteroid processing into a likelihood, which is notated as $p(measurement|col, aoi, ttc)$. First, the stereo image pair and the previous left camera image are used to estimate the disparity and flow. The disparity is processed with the Stixel World algorithm to build fronto-parallel rectangular superpixels.

The next step is to convert these superpixels to probabilistic measurement data. To this end, we propose a motion particle sampling method that is specifically designed to capture typical noise in our stixel-based approach. In short, the process

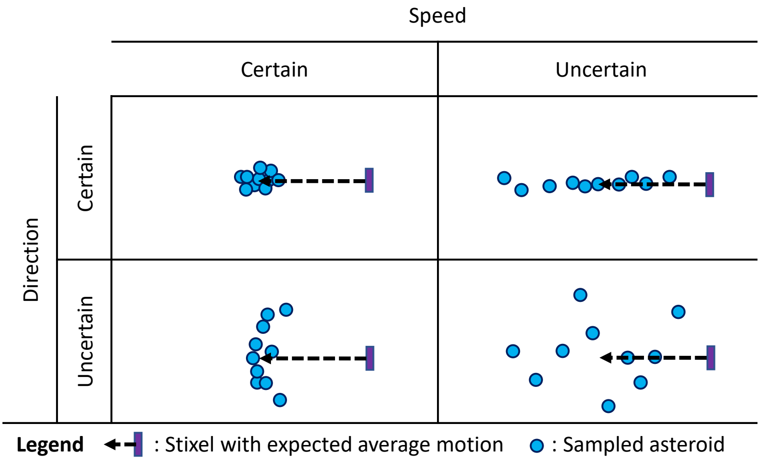


Figure 5.3 — *Conceptual illustration of how asteroid clouds reflect the result of the error propagation in the measurement process.*

consists of the following steps. First, in the process of estimating the velocity of a stixel, the corresponding uncertainty in the form of a probability density function. Second, moving particles, called *asteroids*, are generated according that probability density function. These particles propagate the measurement uncertainty into the collision warning process. The model captures uncertainty both in speed and in direction, as illustrated in Figure 5.3 and described in the following. If both speed and direction are measured with a high confidence, this will generate a very dense ball of asteroids, traversing through space. However, a stixel with a confident direction but with an uncertain speed will generate a laser-beam like stripe of asteroids: they might hit the car all at the same point, but will arrive in a time interval. Alternatively, a stixel with a confident speed but with an uncertain direction will generate a set of asteroids in a wave-front, potentially hitting the car from different directions at similar times. When both direction and speed are uncertain, a dispersed cloud of asteroids can be expected. Concluding, this modeling fluently combines accurate and uncertain measurements of noisy, dynamic data, so that a realistic collision warning analysis can be performed.

5.4 Measurement update and collision analysis

This section presents the details of our key algorithmic contributions, depicted by the three yellow processing blocks in Figure 5.4. The block diagram presents a more detailed view of the top-left block of Figure 5.1. Besides the three yellow processing blocks, the last subsection discusses our collision analysis, which is the light purple block in Figure 5.1.

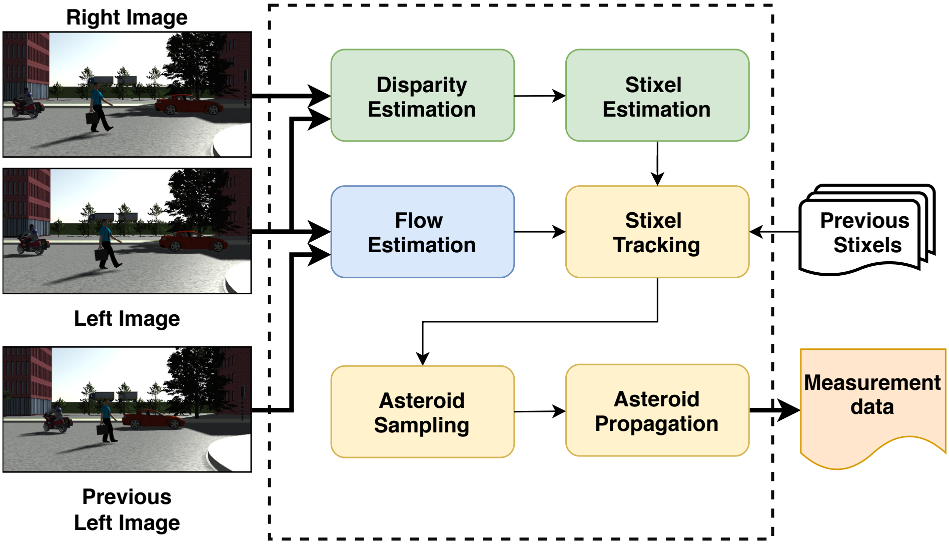


Figure 5.4 — *Schematic overview of the measurement update stage, generating the data likelihood for the Bayesian filter of our collision warning system. Note that this a more detailed view of the top-left block of Figure 5.1.*

5.4.1 Stixel tracking

The *Stixel Tracking* block assigns motion to individual stixels and associates current stixels with those of the previous time step. This facilitates modeling the dynamics of the traffic scene in metric 3D world coordinates, which is crucial for predicting potential collisions.

This Stixel Tracking block is provided with three inputs, namely (1) the set of stixels of the current frame, (2) the optical pixel flow from the previous frame to the current frame, and (3) the set of stixels of the previous time step. The first step of this block is to translate the optical pixel flow to the flow in the 2D image plane for each stixel. To this end, the median of the optical pixel flow is calculated over all pixels that are contained in the stixel area, resulting in the stixel 2D image flow (note that the optical flow corresponds to a 2D displacement measurement for each pixel). Next, the 2D flow is translated to 3D-world motion by trying to match each stixel to the corresponding stixel of the previous set. This matching process first moves a current stixel to its previous position in the image plane according to its 2D image flow. Then, it analyzes the overlapping stixels of the previous set at that location. The stixel is disregarded if less then 75% of its moved area falls within the image region (not traceability), or if less than 50% has overlap with previous obstacle stixels (lack of correspondence). The overlap analysis is conceptually illustrated in Figure 5.5 and further explained below. If there is only one overlapping stixel, this is considered the match. If there are multiple overlapping stixels, these candidates are analyzed in a small selection process. First, candidates that have an overlap ratio of less then $1/(N_{oa}+1)$ are disregarded, where N_{oa}

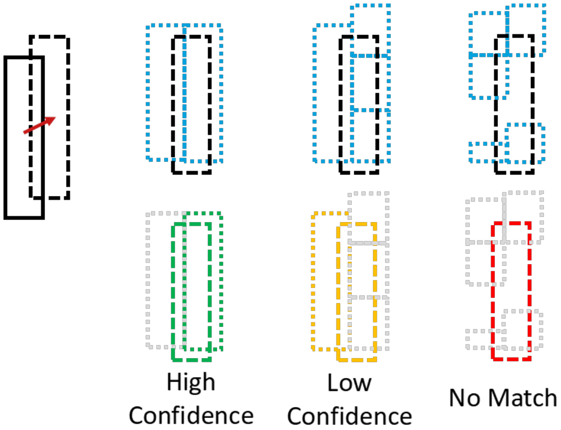


Figure 5.5 — *Conceptual illustration of the stixel tracking process by overlap analysis. The stixel is moved according to its optical motion flow (top, black striped) and then the overlapped stixels are analyzed of the previous set (top, dotted light blue). This results into a match with a certain confidence (bottom middle, green/yellow), or no match at all (red, right).*

is the number of candidates. If this still leaves multiple candidates, the stixel is matched to the candidates by computing the Bhattacharyya coefficient, and then selecting the candidate with the highest coefficient value. This involves comparing the stixel texture-wise via a normalized one-dimensional color histograms (10 bins per channel). If no such candidate exists, no match is made. Each resulting match is assigned a corresponding confidence from this overlap analysis, c_{oa}, which is defined as follows:

$$c_{oa}(\boldsymbol{s}) = \begin{cases} (\boldsymbol{A}(\boldsymbol{s}_{\text{cur}}) \ \cap \ \boldsymbol{A}(\boldsymbol{s}_{\text{match}}))/\boldsymbol{A}(\boldsymbol{s}_{\text{cur}}), & \text{if } N_{oa} = 1; \\ 1 - (d_{oa}^{\text{max}} - d_{oa}^{\text{min}})/d_{oa}^{\text{max}}, & \text{if } N_{oa} > 1; \end{cases} \tag{5.1}$$

where $\boldsymbol{A}(\boldsymbol{s})$ denotes a stixel area counted in pixels and d_{oa}^{max} and d_{oa}^{min} represent the largest and the smallest disparity value of the candidate stixels, respectively. Using this normalized disparity range as a confidence metric in the case of multiple candidates, ensures that c_{oa} is not too conservative, especially if there is over-segmentation in the previous set of stixels. More specifically, if a stixel overlaps with multiple previous stixels that all have a similar disparity value, this should not lead to a low confidence in the previous stixel position.

After the matching process, both stixels that could not be matched and stixels that are clear outliers are removed, to avoid cluttering the subsequent process, while still facilitating a high inclusion of measurements. The tracked stixel should have a confidence of more than 0.5, it should be within relevant range of the ego-vehicle (at most 30 m to the left or right, 2.5 m up or down; up to 60 m in front) and it should have a relative speed below 150 km/h, considering that the maximum allowed absolute speed is around 50 km/h within our urban context. The colored tails of the stixels in Figure 5.6 illustrate the result of the tracking process.

5.4.2 Asteroid sampling

The tracked stixels are supplied to the subsequent *Asteroid Sampling* block, which generates so-called asteroids for each stixel. This step translates the dynamic stixels into a probabilistic measurement distribution, which is an input to the measurement-update stage of the Bayesian histogram filter. To this end, we first define an *asteroid* as a particle with a trajectory sampled from two one-dimensional Gaussian distributions, one for the x- and one for the z-velocities, so that

$$v_x \sim \mathcal{N}_x(\overline{v_x}, \sigma^2_{v_x}) \quad \text{and} \quad v_z \sim \mathcal{N}_z(\overline{v_z}, \sigma^2_{v_z}). \tag{5.2}$$

Note that we have chosen to exclude the y-dimension at this stage. This is in agreement with the design of the state space, which does not differentiate between vertical angles of impact, while it also fits with the aspect that the stixel tracking step already removes stixels that are situated too high or too low. To compensate for this simplification, we assume that the ego-vehicle's height spans this entire horizontal range, where nothing can pass over or under. This assumption is over-cautious, but it simplifies the estimation to a two-dimensional problem.

The average velocity in each axis is calculated from the N_{track} previous positions available in the stixel track, hence, for the x-direction we compute the average speed $\overline{v_x}$ as:

$$\overline{v_x} = \tfrac{1}{T_s \cdot N_{\text{track}}} \sum_{k=0}^{N_{\text{track}}-1} x_{t-k} - x_{t-k-1}, \tag{5.3}$$

and analogous for the z-direction. In the equation above, T_s represents the sample time of the data measurements, and t is the discrete time index of the current frame.

The variances of distributions in Equation (5.2) are derived by extending the standard uncertainty propagation in disparity estimation, using a camera pinhole model with the stixel estimation and our matching process. First of all, the error propagation for the velocity estimate, using standard calculation rules from probability theory [112], results in

$$\sigma^2_{v_x}(\boldsymbol{s}) = \frac{\sigma^2_{x_t} + \sigma^2_{x_{t-1}}}{{T_s}^2}, \tag{5.4}$$

and a similar propagation for the z-direction. Second, the stixel-position variances can be defined from applying two camera pinhole models. These cameras have a stereo camera baseline b, the u-coordinate of the left-camera's principal point u_{pp} and the left camera's focal length f_u. The obtained disparity estimation process comes with uncertainty σ^2_{disp}, which is fixed at 0.5 pixels, in agreement with the general rule that the camera resolution provides a bound on the disparity accuracy.

Starting from the stereo pinhole camera model that translates image coordinates and disparity to world coordinates x and z, we can use the widely adopted variance formula [113], which describes the relation between the variance of an input variable a to an output function variable $Q(a)$ as

$$\sigma_Q = \left| \frac{\partial Q(a)}{\partial a} \right| \cdot \sigma_a, \tag{5.5}$$

to compute the error propagation to the output. This leads to the following equations for the standard deviation of the disparity-dependent coordinates $x(d)$ and $z(d)$:

$$z(d) = \frac{f_u \cdot b}{d} \quad \Longrightarrow \quad \sigma_z = \frac{f_u \cdot b}{d^2} \cdot \sigma_{disp}, \tag{5.6}$$

$$x(d) = \frac{z \cdot (u - u_{pp})}{f_u} = \frac{b \cdot (u - u_{pp})}{d} \quad \Longrightarrow \quad \sigma_x = \frac{b \cdot (u - u_{pp})}{d^2} \cdot \sigma_{disp}. \tag{5.7}$$

Additionally, we incorporate two aspects of our stixel-based processing, namely to divide by stixel height h and, in case of the previous position, to scale with the confidence in the overlap analysis c_{oa}. Therefore, we define the variances in x and z for the current and previous stixel positions in accordance with these aspects and the Eqns. (5.6) and (5.7) as follows:

$$\sigma^2_{x_t}(\boldsymbol{s}) = \frac{\sigma^2_{disp}}{h} \cdot \left(\frac{b \cdot (u_{c,t} - u_{pp})}{d_t^2} \right)^2, \tag{5.8}$$

$$\sigma^2_{x_{t-1}}(\boldsymbol{s}) = \frac{\sigma^2_{disp}}{c_{oa} \cdot h} \cdot \left(\frac{b \cdot (u_{c,t-1} - u_{pp})}{d_{t-1}^2} \right)^2, \tag{5.9}$$

$$\sigma^2_{z_t}(\boldsymbol{s}) = \frac{\sigma^2_{disp}}{h} \cdot \left(\frac{b \cdot f_u}{d_t^2} \right)^2, \tag{5.10}$$

$$\sigma^2_{z_{t-1}}(\boldsymbol{s}) = \frac{\sigma^2_{disp}}{c_{oa} \cdot h} \cdot \left(\frac{b \cdot f_u}{d_{t-1}^2} \right)^2, \tag{5.11}$$

where the variables from the stixel under analysis h, u_c, d and c_{oa} represent height, central u-coordinate, disparity, and overlap-analysis confidence, respectively. Intuitively, stixels that have a larger height also have a more certain x- and z-position, since each row of the stixel can be considered an additional measurement. Moreover, we have introduced a metric for the confidence in the overlap analysis that ranges between zero and unity, hence $0 < c_{oa} \leq 1$. As a result, a sub-optimal confidence of the overlap analysis will increase the uncertainty in the estimate of the previous position. For example, when stixels can be matched clearly to a predecessor, c_{oa} is close to unity, having little impact on the position estimate. However, if the overlap analysis provides a low-confident match, the uncertainty in the previous position will be enlarged. This reduces the impact of noisy tracks into the collision analysis later in the pipeline, since they lead to dispersed asteroid clouds, spreading their collisions over time and location.

The third aspect, after determining the stixel velocities and their uncertainties, is to calculate the amount of asteroids that should be generated for each stixel. This amount depends on several aspects, such as the stixel area and confidence, which are included in the following equation:

$$N_{\text{ast}}(\boldsymbol{s}) = \mathcal{A}(\boldsymbol{s}) \cdot \rho_{\text{ast}} \cdot c_{\text{fit}}(\boldsymbol{s}) \cdot c_{\sigma_d^2}(\boldsymbol{s}). \tag{5.12}$$

The core value in this equation is $\mathcal{A}(\boldsymbol{s})$, which is the stixel surface in m^2 (and not in pixels, as was used before), calculated by translating all four stixel u,v,d corner points to 3D world coordinates. This surface is multiplied with the asteroid density (ρ_{ast}), a system parameter, to come to an initial number of asteroids. However, the equation also incorporates two confidence values, that both can reduce the number of generated asteroids. The first one, $c_{\text{fit}}(\boldsymbol{s})$, is adapted from [114] and defined by:

$$c_{\text{fit}}(\boldsymbol{s}) = 1/(1 + \exp{(e_{\text{obstacle}}(\boldsymbol{s}) - e_{\text{ground}}(\boldsymbol{s}))}), \tag{5.13}$$

where the values e_{obstacle} and e_{ground} model energies, given by

$$e_{\text{obstacle}}(\boldsymbol{s}) \quad = \quad \tfrac{1}{h} \sum_{v \in v_c \pm h/2} |d_v - d|, \tag{5.14}$$

$$e_{\text{ground}}(\boldsymbol{s}) \quad = \quad \tfrac{1}{h} \sum_{v \in v_c \pm h/2} |d_v - a_{\text{gnd}}(v_c - v) - d|, \tag{5.15}$$

with stixel values h (height), v_c (center row) and d (disparity). Additionally, these energies are summed over the rows v spanned by the stixel, and use d_v as the disparity data in the stixel area at row v and a_{gnd} as the expected slope of disparity data representing flat ground in the image plane. This slope can be calculated from the camera setup using $a_{\text{gnd}} = b/h_{\text{cam}}$, where h_{cam} is the height of the camera above the ground surface.

To derive this formula for a_{gnd}, take two points p_a and p_b on the ground surface, each with their own real-world coordinates (x, y, z), image coordinates (u, v) and disparity d. Furthermore, denote the v-coordinate of the camera's principal point as v_{pp}, the stereo baseline as b and its focal length as f. The d, v-slope coefficient a_{gnd} of the groundplane is then $a_{\text{gnd}} = (d_a - d_b)/(v_a - v_b)$. Stereo disparity geometry provides that $y_a = z_a \cdot (v_a - v_{pp})/f = b \cdot f/d_a \cdot (v_a - v_{pp})/f$. Additionally, it holds that $y_a = y_b = h_{\text{cam}}$, since both points are on the ground surface. Together, these equations imply that $d_a = (v_a - v_{pp}) \cdot b/h_{\text{cam}}$, and analogous for d_b. Combined, this results in $a_{\text{gnd}} = b/h_{\text{cam}}$.

The confidence $c_{\text{fit}}(\boldsymbol{s})$ expresses how well the stixel model fits the raw disparity it covers, knowing that the optimization process explores two options (ground or obstacle). It compares fitting either a fronto-parallel surface or a sloped surface to the condensed single disparity column in the stixel. The other confidence, $c_{\sigma_d^2}(\boldsymbol{s})$, is the normalized inverted variance of the disparity within the stixel region. This

Chapter 5

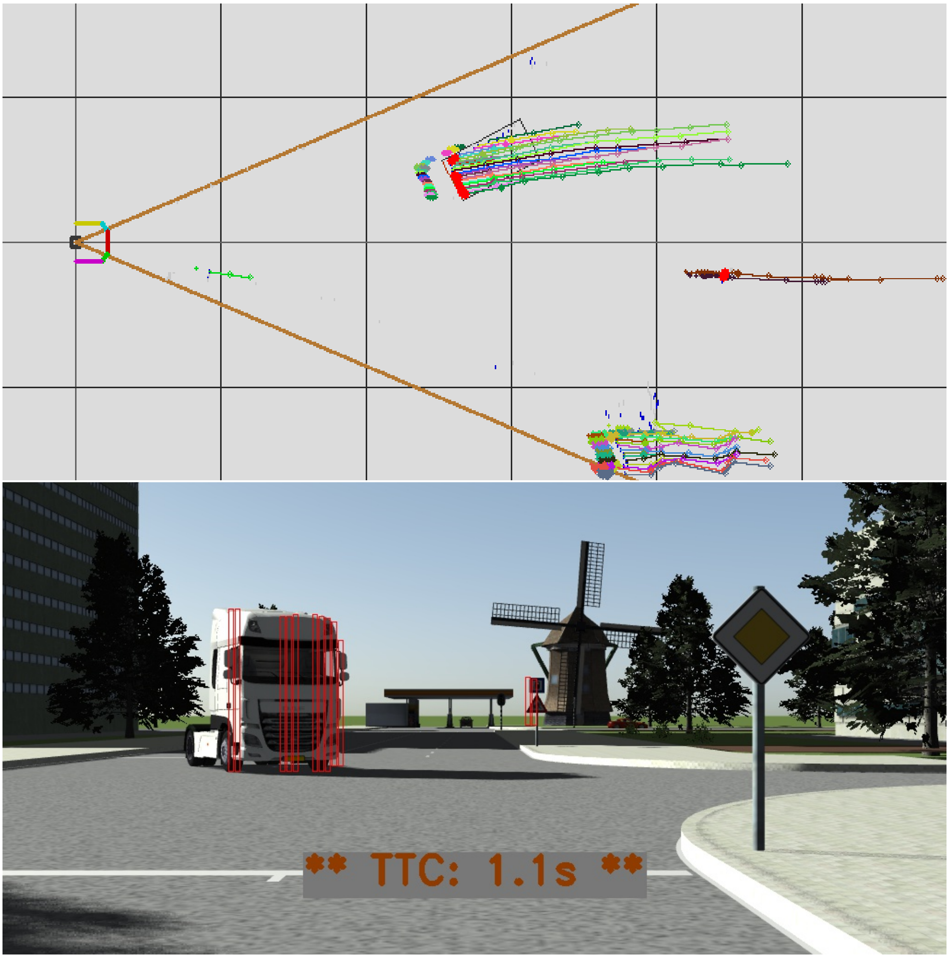


Figure 5.6 — *Left: our processing visualized in a top-down view with the ego-vehicle at the center left (moving to the right), the camera field-of-view lines in dark orange; five colored sides-of-impact and grid lines at every 10 m. Furthermore, the figure shows stixel tracks, sampled asteroid clouds and detected colliding asteroids (in bright red). Right: corresponding camera image with the collision warning overlay.*

is more generic than the previous fitting-based confidence, since it also considers the fact that a stixel spans multiple columns. However, both confidence values aim to decrease the chance of generating spurious false asteroids from stixels in noisy disparity data, by reducing the output number of asteroids in Eq. (5.12).

The top-down view in Figure 5.6 shows the sampled asteroid clouds as colored blobs at the end of stixel tracks. The asteroid clouds from the trees (at the right of the ego-vehicle) are larger, showing more uncertainty in those measurements.

5.4.3 Asteroid propagation

The third block, *Asteroid Propagation*, takes the cloud of asteroids, propagates them along their generated trajectories and monitors which of those are going to impact a safety bubble around the ego-vehicle and their corresponding times to impact.

The propagation process relies on constant-velocity kinematics without any rotational component. Currently, no advanced dynamic models are applied. The constant-velocity model is a reasonable choice given the objective of offering a generic, class-agnostic analysis. Moreover, the impact of this simplification is reduced by having multiple stixels per object and generating multiple asteroids per stixel in a probabilistic fashion. Nonetheless, this constraint will limit the time horizon for which our predictions are reliable. The goal is to explore these boundaries and to identify the strengths and weaknesses of the stixel-based approach, rather than providing a stand-alone all-encompassing collision warning solution. However, note that our method is capable to utilize additional information due to its probabilistic design, if such information would be available to the system.

Performing the collision assessment based on a linear trajectory extrapolation, can be solved efficiently as a standard geometric line-segment intersection problem, as presented in [115]. We model both the trajectory of the asteroid and the side-of-impact lines with an origin point ($\underline{\alpha}_o$ and $\underline{\mathrm{soi}}_o$) and a vector ($\underline{\alpha}_v$ and $\underline{\mathrm{soi}}_v$), and find τ and ζ such that

$$\underline{\alpha}_o + \tau \cdot \underline{\alpha}_v = \underline{\mathrm{soi}}_o + \zeta \cdot \underline{\mathrm{soi}}_v. \tag{5.16}$$

By using this representation, τ directly provides the time to an impact, while ζ indicates the location of the impact (as the distance from $\underline{\mathrm{soi}}_o$). Therefore, an asteroid collides with the side-of-impact if and only if

$$(0 < \zeta < |\mathrm{soi}|) \quad \wedge \quad (0 < \tau < \infty), \tag{5.17}$$

where $|\mathrm{soi}|$ represents the length of the side-of-impact.

For the truck at the left of the scene in Figure 5.6, the asteroids are clearly projected in front of the object (marked in bright red) from analyzing the corresponding tracks of the stixels.

5.4.4 From histogram to probability distribution

The results of the asteroid propagation process are represented in a 2D histogram, matching the configuration of the state space. Each bin contains the amount of colliding asteroids m_{ast} for its corresponding angle-of-impact and time-to-collision. This histogram is then translated into the likelihood with a linear model that depends on the asteroid density parameter ρ_{ast} by

$$\begin{aligned} p(m_{\text{ast}}|col, aoi, ttc) &= 2/\rho_{\text{ast}} \cdot m_{\text{ast}}/\rho_{\text{ast}}, & (5.18)\\ p(m_{\text{ast}}|\neg col, aoi, ttc) &= 2/\rho_{\text{ast}} \cdot (1 - m_{\text{ast}}/\rho_{\text{ast}}). & (5.19) \end{aligned}$$

When $m_{\text{ast}} \geq \rho_{\text{ast}}$, we enforce saturation by setting $p(m_{\text{ast}}|col, aoi, ttc) = 2/\rho_{\text{ast}}$ and $p(m_{\text{ast}}|\neg col, aoi, ttc) = 0$. This means that a fully confident surface of 1 m^2 will generate enough asteroids to saturate a histogram bin, independent of the density parameter. The factor 2 ensures that the probability distribution is normalized. Next, the likelihood is fed into the Bayesian filter-update stage. Additionally, the collision probabilities are further processed in the *Collision Analysis* block, described below.

5.4.5 Collision analysis on the state space

The collision analysis block (see Figure 5.1) processes the state and generates warnings if necessary. This module completes the system processing chain and facilitates assessing the reliability of the analysis in a tangible way.

First, this block extracts a collision probability for each state cell from the joint probability, by marginalizing over the collision axis, hence it calculates the probability $p(col|ttc, aoi)$ by the following equation:

$$p(col|ttc, aoi) = \frac{p(col, ttc, aoi)}{p(col, ttc, aoi) + p(\neg col, ttc, aoi)}. \tag{5.20}$$

Second, it adds robustness by employing a CFAR algorithm, which performs peak detection and tracking in the probability distribution, as discussed in the following subsection.

A. CFAR: peak detection

The next step is to identify peaks in the probability distribution that correspond to potential collisions. This process addresses the fact that the asteroids in the histogram are sampled from the noisy tracked stixel data, and hence, they travel towards the car as a dispersed cloud. Since this shows similarities to detecting objects in noisy Radar data, we propose to employ a well-established method from that field and use a Constant False-Alarm-Rate (CFAR) detection scheme [116], [117]. CFAR is an adaptive thresholding technique to find relevant peaks against noisy background clutter. In theory, it provides the desired detections at the cost of a pre-defined false-alarm rate, which explains the name. This is based on assumptions on the distribution of background clutter. We now briefly describe this, adapted to our context.

A CFAR detector checks if the probability in a cell is a local maximum and higher than a certain threshold. This threshold is derived from the neighboring cells to adapt it to the local noise caused by outlying measurements. We treat each *angle-of-impact* (aoi) as an independent sequence of measurements, which means that our CFAR neighborhood is one dimensional, along the *time-to-collision*-axis (ttc) only, and not depending on the value of aoi. Therefore, we omit aoi in the argument list of the probability distribution for brevity in the following equations. Formally, our CFAR collision-peak detection $C_{\text{cfar}}(ttc)$ is defined with the following set of equations:

$$C_{\text{cfar}}(ttc) \quad = \quad (p(col|ttc) > T_{\text{cfar}}(ttc)) \quad \wedge \quad \left(ttc \equiv \underset{\tau \in \Theta_{\text{cfar}}}{arg\, max}\, p(col|\tau)\right), \tag{5.21}$$

$$T_{\text{cfar}}(ttc) \quad = \quad \frac{\alpha_{\text{cfar}}}{N_{\text{cfar}}} \sum_{\tau \in \Theta_{\text{cfar}}} p(col|\tau), \tag{5.22}$$

$$\alpha_{\text{cfar}} \quad = \quad N_{\text{cfar}}(p_{\text{fa}}^{-1/N_{\text{cfar}}} - 1), \tag{5.23}$$

using several system parameters, where p_{fa} is the theoretically desired probability of false-alarm, Θ_{cfar} the definition of the neighborhood of the ttc-bin under analysis, and N_{cfar} the corresponding amount of training cells (bins) in that neighborhood. The neighborhood consists of training cells, both in front and behind the cell under test. To suppress spurious detections, typically one or more guard cells are defined, in between the cell under test and the training cells. Our Θ_{cfar} is configured empirically as two front-training cells, two front-guard cells, six after-guard cells and six after-training cells. Consequently, Θ_{cfar} spans 17 ttc-bins and has $N_{\text{cfar}} = 8$.

B. CFAR: peak tracking

The CFAR peak detector provides the most critical time-to-collision and does not handle any data association between multiple potential collision blobs in the state in itself. In our CFAR peak-tracking step, we focus on detecting the most critical collision smoothly, thereby addressing the majority of the situations, and leave handling of multiple targets for future work.

Our CFAR peak tracker consists of a sliding-window buffer with a length of seven frames for each angle-of-impact. Within that buffer, lines are fit through every pair of collision-peak ttcs. This again assumes a constant-velocity model, which would lead to a linearly decreasing ttc in more recent measurements. For every line, the number of measurements in the buffer is found that are within three ttc steps of the fitted line. If there are at least four of these inliers, the line is considered to represent a collision event. The event with the highest number of inliers is selected to generate a warning with its corresponding extrapolated ttc. When multiple lines have equal support, the one with the most urgent time-to-collision is given priority. This sliding-window strategy suppresses spurious detections and simultaneously, it resolves missed peaks in the detection step of the CFAR process.

Figure 5.6 presents an example result where the bright red stixels on the front of the truck are stixels that cause the generated collision warning, visible in the top-down view and also in the overlay on the camera image.

5.5 Evaluation approach

This section explains the validation of the proposed system by addressing the selection of data sets, performance metrics and the performed experiments. Even though the provided evaluation cannot serve as an automotive-compliant end-to-end validation of the system, it demonstrates the feasibility of our stixel-based collision warning system through both simulated and real-world experiments.

5.5.1 Datasets

The strengths and weaknesses of the system will be analyzed using real-world and simulated data. More specifically, we employ the well-known KITTI tracking dataset, although this set contains a limited number of (near) collisions. We are not aware of the existence of a real-world stereo-vision dataset that contains collisions, and lack access to e.g. test sites with dummy objects such as used in [118]. Therefore, we analyze our system on a newly simulated PreScanStereoCollision dataset, and a newly recorded real-world TUE&ACNL dataset with intentional near-collisions. All datasets have a focus on urban environments and will be discussed below.

The simulated PreScanStereoCollision dataset (PSSC) is newly made for this research with the PreScan software package [119] and exported in KITTI dataset format for compatibility. This simulated data is included in our evaluation to test actual collisions and easily evaluate different relevant scenarios. We have created 5 sequences in the PreScan environment: *Straight*, *Figure-8*, *Y-crossings-Fast*, *Y-crossings-Slow* and *Mixed*. Figure 5.7 shows example frames of each sequence. Sequence *Straight* is a large rectangular trajectory containing head-on collisions with static objects of decreasing sizes (e.g. from truck to car, down to kids) and an empty road with common side objects such as trees and buildings. Sequence *Figure-8* contains similar obstacles, but now on a curved road, so that the ego-vehicle is constantly changing its heading. The *Y-crossings* sequences contain a straight trajectory for the ego-vehicle, with different objects approaching on collision course from the right, appearing at consecutive y-crossings. In the *Fast* version, each participant moves at its own nominal speed, while in the *Slow* one, speeds are decreased such that the maximum relative collision speed is similar to that of the collision speed in the *Straight* sequence. Finally, the *Mixed* sequence is a busy, fully dressed city center with multiple traffic participants approaching from various directions. It contains all kinds of vehicles and pedestrians that are either on a safe or on a collision course, straight and from different angles. The simulated stereo camera has a baseline of 30 cm, a resolution of 1024×512 pixels and a field of view of 46.2×24.1 degrees.

Additionally, we evaluate our system on the KITTI-tracking dataset. This KITTI data has no collisions and only a handful of near-collisions, but a crucial aspect is to quantify the amount of false alarms on real-world data. The evaluation requires ego-motion as well as the true object positions. Hence, the evaluation is limited to the training set of KITTI-tracking. This is the only part of the dataset for which

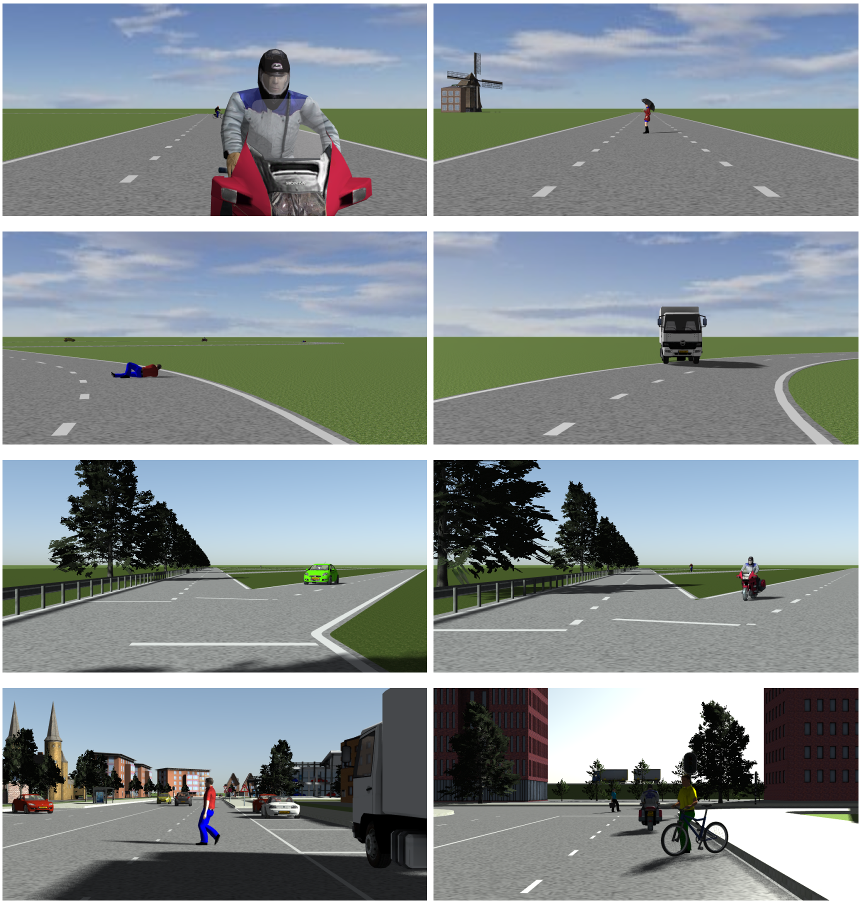


Figure 5.7 — *Examples of our PreScanStereoCollision (PSSC) sequences, two frames per sequence; from top to bottom:* Straight, Figure-8, Y-crossings, Mixed.

the annotated object bounding boxes and positions are available, which can be exploited to generate ground-truth collision warnings. We generate stixels on the surface of object bounding boxes, and set their motion according to the annotated motion of the object. These stixels are then used to generate a single asteroid each, with fixed motion, which is extrapolated to produce the reference ttc labels.

Besides these two datasets, we have recorded a real-world dataset, *TUE&ACNL*, at the Eindhoven University of Technology campus (TUE), the Automotive Campus Netherlands (ACNL) at Helmond and the roads in between. During these recordings, the car is driven around the TUE campus in normal traffic for 30 minutes and is also steered towards near-collisions with other traffic or static obstacles. The recordings at and towards ACNL are partially in normal traffic and partially on temporarily closed roads. More importantly, several sequences were recorded during nighttime. This dataset has no annotations of true obstacle positions and

Table 5.1 — *Evaluation data overview*

Dataset	Camera[1]		#Pos	#Neg
PSSC	30 cm; 46°;	Frames	880	1,914
(4.7 minutes)	1024 pixels; 10 Hz	Events	40	n.a.
KITTI	54 cm; 80°;	Frames	155	7,811
(13.3 minutes)	1242 pixels; 10 Hz	Events	40	n.a.
TUE&ACNL[2]	30 cm; 44°;	Frames	23,000 [3]	
(63 minutes)	480 pixels; 6 Hz	Events	±100	n.a.

[1] Provided are: baseline, horizontal field-of-view, image width and frame rate.

[2] TUE&ACNL has no ground-truth annotations.

[3] Roughly 5,000 of these frames have been recorded at night.

motion for a full quantitative evaluation. However, it offers a valuable qualitative insight into the collision warnings that the system generates in real-world conditions, since we have used a regular automotive-grade stereo camera. Table 5.1 summarizes the properties of the employed annotated data, regarding the duration and the number of frames and collision events.

5.5.2 Metrics

The performance of our collision warning system will be quantified at two positions in the processing chain, namely prior to and after CFAR peak tracking. Ultimately, the goal is to design a system that handles the complete events properly. Hence, it is acceptable that the system misses a collision peak in some frames, if it still detects the corresponding event relying on other frames. The evaluation on a per-peak basis gives an idea on the intermediate level of performance. It contains more samples,which increases the reliability of the analysis, while it also provides insights into strengths or weaknesses in the processing. This performance is also relevant for the described case where the collision analysis would be fused in a larger system. The performance will be quantified by calculating the recall, the precision and their harmonic mean (F_1 score), both prior to and after peak tracking.

5.5.3 Time range

As explained in Section 5.4.3, the time horizon in which our system can reasonably operate is inherently limited by our modeling assumptions. The most dominant limitation originates from our use of a constant-velocity motion model, which is especially uncertain in our context of operation: urban areas with nearby traffic from any direction.

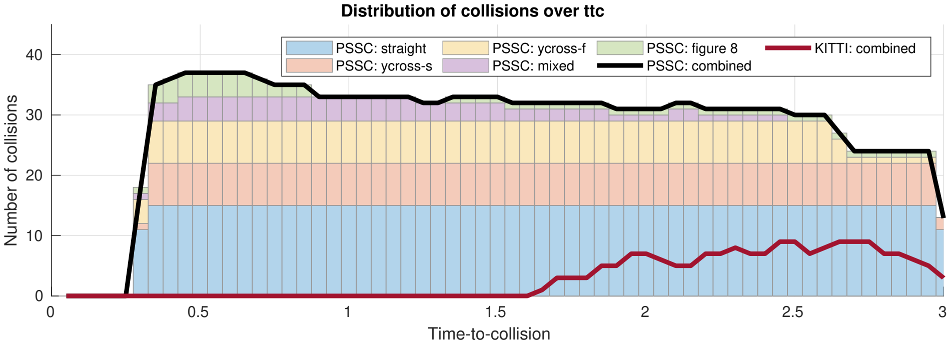


Figure 5.8 — *Distribution of collisions over ttcs for the PSSC and KITTI datasets. The stacked histogram shows the different PSSC subsequences. It illustrates that the KITTI recordings contain very few potential collisions, motivating the need for the complementary simulated PSSC data.*

To link our evaluation to real-world conditions, we rely on the stopping-distance guidelines that are used by the NACTO [120] and NHTSA [121]. They provide ballpark figures for feasible de-acceleration, that are said to range from 6 m/s^2 for a reasonably skilled driver to 9.8 m/s^2 for a professional driver under good conditions. Since our system is designed for urban scenery, the ego-vehicle speed is around 50 km/h. In this case, it would require somewhere between 1.4-2.3 s to fully stop the ego-vehicle, depending on driver skills and conditions. Therefore, if the collision-detection module is integrated tightly into the car control system (e.g. with automated emergency braking as in [118]), it should operate reliably at least up to 1.4 s. However, if the module purely generates warnings to assist an active human driver, it should operate reliably to up to 2.3 s.

Figure 5.8 shows the distribution of collision events in our data over time-to-collision. The events in our simulated data are distributed rather homogeneously between 0.3 and 3.0 s. However, the graph of the KITTI-event distribution confirms that this data was recorded during a clean drive. Namely, the handful of short, potential collision events are never closer than 1.5 s. Therefore, this part of the experimental validation is focused on avoiding false warnings on the KITTI data within the above-mentioned time intervals, while obtaining a high F_1 score on the PSSC in those same time intervals.

As mentioned, the objective of this work is to explore the operational boundaries and identify the strengths and weaknesses of the stixel-based approach, rather than providing a stand-alone all-encompassing collision warning solution.

5.5.4 Experiments

The objective of the validation is already covered by the selection of the datasets, i.e. simulated data with several relevant scenarios and real-world data to test practical feasibility. To further explore the system robustness, we have evaluated the performance over different settings of the core system parameters, being the asteroid density, the maximum tracking length and the parameter p_{fa} for the CFAR module within the *Collision Analysis* block. These experiments focus on validating the newly designed functional blocks of the system.

Additionally, the influence of the quality of the input data is evaluated by selecting different algorithms or settings to generate the disparity and optical flow data. For disparity, a comparison is made between the traditional, widely adopted Semi-Global Block Matching (SGBM) algorithm [56] and a newer, state-of-the-art deep learning-based method, namely DispNet [122]. For flow estimation, the FlowNet2 [123] method is employed, also based on deep learning. The authors of FlowNet2 have presented several neural network architectures and made them publicly available. The different available versions vary highly in inference speed, with a trade-off against pixel-level performance quality. This can be exploited to experimentally quantify the robustness of our data processing against degraded input data. In turn, this will offer relevant insights on the trade-off between system latency and performance quality. Since our our system is stixel-based rather than pixel-based, we aim at being robust to these lower quality, yet faster versions of flow estimation.

5.6 Results

First impressions of the visual results are provided by Figure 5.9. These illustrate typical performance on low-quality flow and noisy disparity, whereas our probabilistic approach is still able to extract relevant information.

5.6.1 Quantitative evaluation on KITTI and PSSC

As a first quantitative evaluation, we present the performance with respect to the ego-vehicle stopping time under different conditions, as discussed in Section 5.5.3. Figure 5.10 portrays the performance of three system configurations that do not detect any false events on the KITTI dataset for the use case of an integrated system (no false positives with ttc <1.4 s, top graph), the use case of a human-in-the-loop (no false positives with ttc <2.3 s, bottom graph) and an intermediate case (no false positives with ttc <1.8 s, middle graph). Within the subset of configurations that comply with that constraint, we present the one with the highest F_1 on the simulated PSSC data. From the graphs in Figure 5.10, we can conclude that the majority of all collisions in all different contexts are detected correctly, showing the strength of our method.

It is interesting to also discuss a failure case. On PSSC, the settings presented above suffer from up to three false negatives, all in the *Figure 8* sequence. The main cause of these misses is the curved ego-motion, which (1) makes the potential collisions short, barely being the minimum required for the event detection module, and (2) does not match well with the prediction step, which only considers straight motion. On top of the curved-motion complications, one collision is with a person lying down on the road (second picture in Figure 5.7), which is so low positioned that it is barely represented with stixels. CFAR correctly detects a

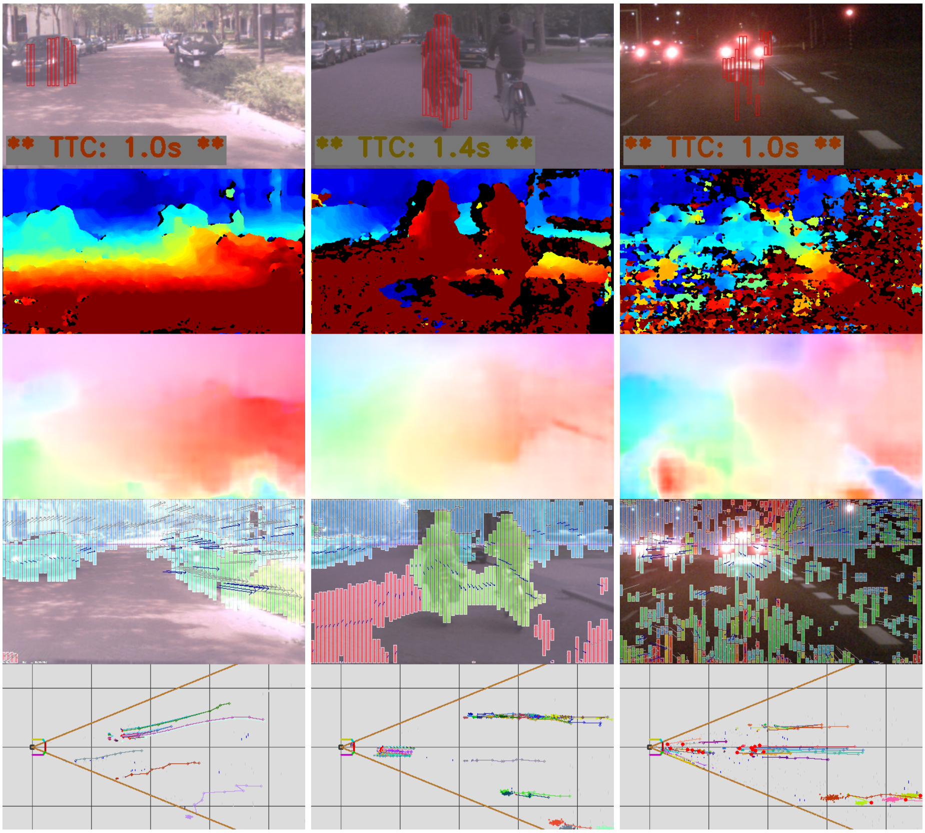


Figure 5.9 — *Three typical collision warnings illustrating clean results from good (left), medium (center) or noisy (right) input, from top to bottom: left camera image with warning overlay, disparity data, color-coded flow data, stixels with flow vectors, top-down scene view. Note that the ego-vehicle is cornering in the first example, so the warning seems false but is actually correct.*

peak at ttc =0.6 s, which is too late for the peak tracking to activate. This shows that our current system is vulnerable for objects lower than around 0.4 m, and could be better adapted to curved ego-motion.

A second set of quantitative results is presented in Figure 5.11, which provides an analysis on how different system parameters influence the detection performance using the PSSC data. Each row shows a different parameter: the method of disparity estimation, the method of flow estimation, the maximal length of stixel tracks, the asteroid density and the CFAR parameter p_{fa}. We have accumulated the results of all parameter combinations and have generated the surfaces by averaging all sub-experiments with a specific value of the parameter under test. The top set of graphs show the peak-detection results, the bottom set those after peak tracking. Both present the recall, precision and F_1 scores. Note that the color maps represent a value surface showing the scores depending on two parameters.

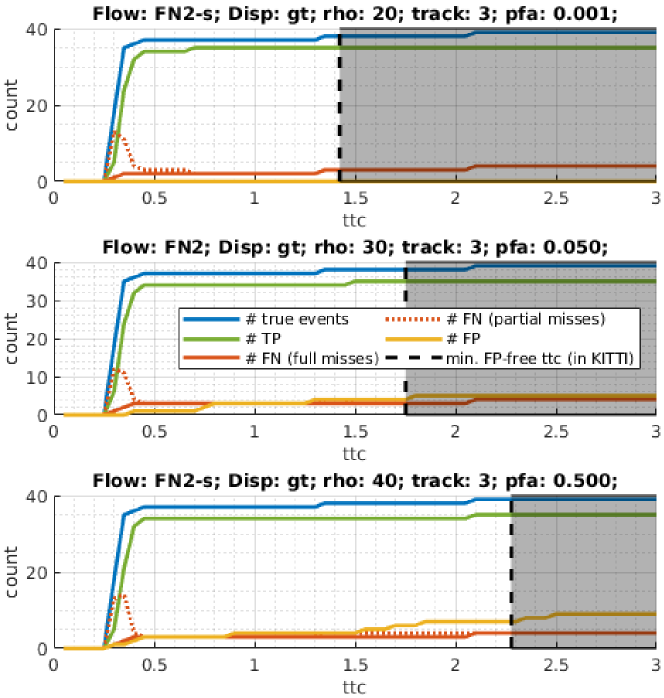


Figure 5.10 — *Best performance on PSSC using settings that produce no false positives on the KITTI data set with ttc <1.4 s (top), ttc <1.75 s (middle) and ttc <2.3 s (bottom).*

Overall, the graphs show better performance at a smaller *ttc*. This makes sense, since it is mostly closer to the ego-vehicle, so that the obstacle is clearer in view and probably sufficiently long in view, such that it could be tracked better.

A noteworthy observation is how little the system performance is impacted by the choice of the flow method. Using the smallest FlowNet2 version (*FN2-s*) yields practically identical performance to using the full version, although the smallest can be executed roughly 17 times faster than the full one (7 ms on an GTX-1080 GPU), at the cost of a drop in pixel-wise performance of up to a factor of 2 [123]. This shows the potential in robustness of combining a superpixel strategy with probabilistic sampling and filtering, as discussed in Section 5.3.

Additionally, the surfaces show that allowing for longer stixel tracks improves the recall and hence the F_1 score of the system, both prior to and after CFAR peak tracking. Other than that, there is a slight preference towards a smaller asteroid density and a high p_{fa} for the peak detection.

A similar analysis for the impact of system configuration on the results on KITTI data is provided in Figure 5.12. Since there are so few actual potential collisions, we only discuss the number of false positives here, which should be low preferably. The surface plots show that most false CFAR peaks (prior to tracking) occur at a large ttc values. However, no false *events* occur at large ttc values. This can be explained by the fact that the measurements at large ttc values are more uncertain and tracking is not yet able to support the estimation, which leads to inconsistent peak detections within the CFAR module. Subsequently, the peak-tracking step filters these out, thereby improving the system robustness.

Other important observations are that reducing either the flow quality, or the use of a small asteroid density, or a large p_{fa} value, all have a slight negative impact on the results. A striking graph is that of the maximum tracking length: shorter tracks or long tracks are better than medium tracks. We hypothesize that short tracks lead to noisy data that is filtered out later more easily, while long tracks lead to more accurate estimations that do not need to be removed.

In the design of our state space and by the structure of the ego-vehicle's impact bubble, the system is able to handle collisions from all directions and at different sides of impact. The reader should note that this is by design, rather than by the presented experiments. However, the evaluation has been limited to the detection of collisions at the front of the vehicle. The cause of this constraint is that the annotated real-world data has been recorded with a single, forward looking stereo camera, so that it is currently not feasible to validate this functionality in practice. More specifically, the horizontal field of view of the sensor setup does not cover many collisions from wide angles-of-impact, but in principle the system facilitates this when the sensor coverage would be enhanced.

5.6.2 Timing

The algorithmic contributions are implemented in C++ and tested on a desktop PC (Xeon E5-1660 0 CPU executing at 3.30 GHz with 12 cores and 15.6 GB RAM). On the KITTI data, the Stixel World algorithm takes roughly 20 ms. The bottleneck within our proposed blocks is the Stixel Tracking module, requiring 35-45 ms with the current implementation and platform. It is faster on the PSSC data (15-20 ms), since the stixel segmentation is much cleaner, which indicates that removing clutter stixels prior to the matching process can speed up processing. Both Asteroid Sampling, and their propagation including the collision check and the appended histogram filter need 1-4 ms each, while the CFAR detection requires only up to 1 ms. Together, this results in a image throughput speed of 15-17 fps, which is sufficiently fast for real-time operation on the 10-fps datasets.

5.6.3 Qualitative evaluation on TUE&ACNL

This section presents the qualitative results on the TUE&ACNL data, which consists of real-world recordings of 63 minutes in normal traffic and on closed roads, partially during the night, with several intentional near-collisions. Figure 5.13 presents four examples of typical ASTEROIDS performance on near-collisions. The first three examples depict frames that have been captured 0.5 s apart, whereas the snapshots of the rightmost example (with the small pole) are 0.16 s apart, since it was only briefly on a collision trajectory. Even though we cannot quantify the estimated ttc values, the warnings generated by the system seem natural and plausible to the driver. Moreover, there was not a single false warning during the whole recording. Three-quarters of the data was recorded in bright sunny weather, causing sharp shadows, high contrast, temporal flicker, direct sunlight and reflections, which all can be handled correctly by our system. The rest of the data was captured during nighttime. The system is still able to generate warnings for near-collisions in the dark, although they typically occur later (earliest at $ttc \approx 1.5\,\mathrm{s}$). In conclusion, this experiment supports our proof-of-concept evaluation with the findings on the KITTI and PSSC datasets and shows promising real-world applicability.

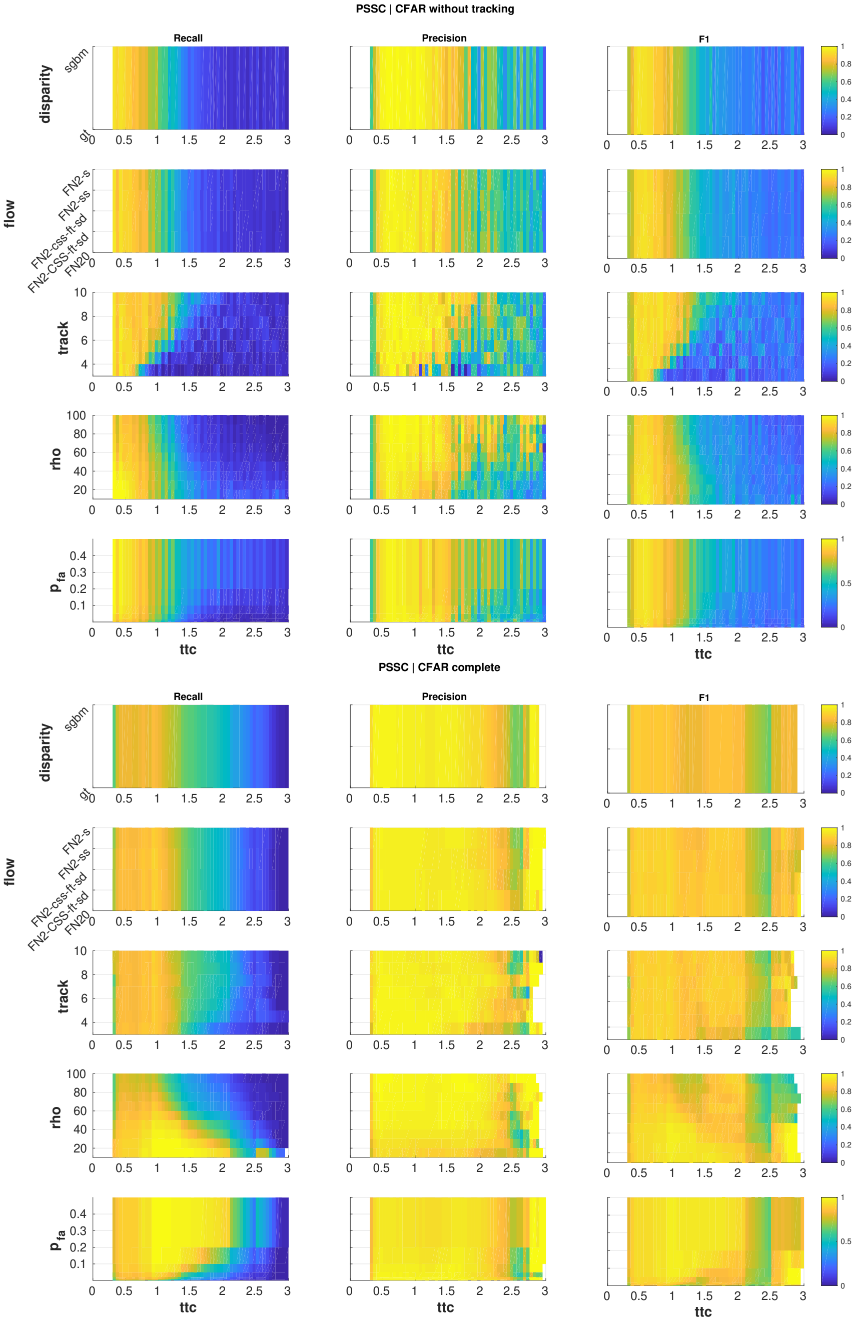


Figure 5.11 — *Impact analysis of system configuration on the system performance using PSSC data, split over different parameters and plotted over time-to-collision. The color yellow represents a desired high score. Note that reducing the quality of the flow and/or disparity has little impact on the performance (top 2 rows).*

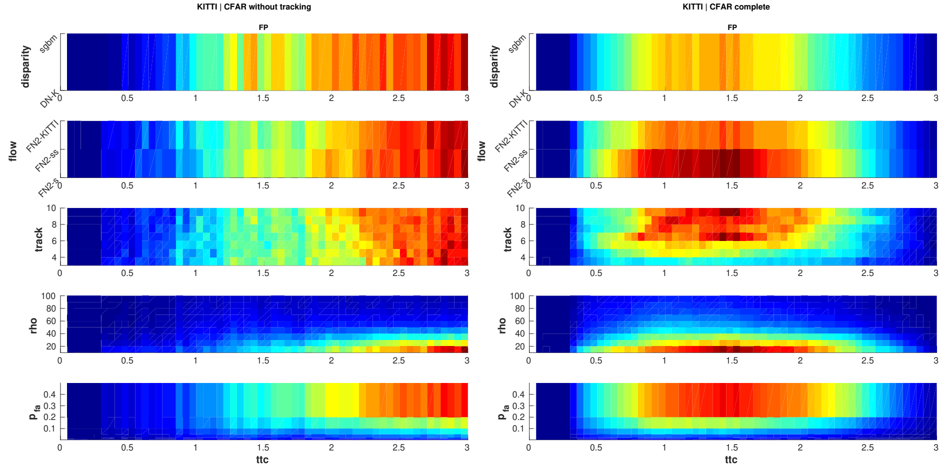


Figure 5.12 — *Analysis of the* relative *impact of system configuration on performance with KITTI data, split over different parameters and plotted over time-to-collision. Because of the low actual potential collisions, we show only false positives. Color towards red represents an undesired (relatively) high FP count. Note that the color ranges are stretched individually to emphasize relative performance within each parameter. Hence, comparing results between parameters or with and without peak tracking in an absolute sense is not the objective here and also not represented.*

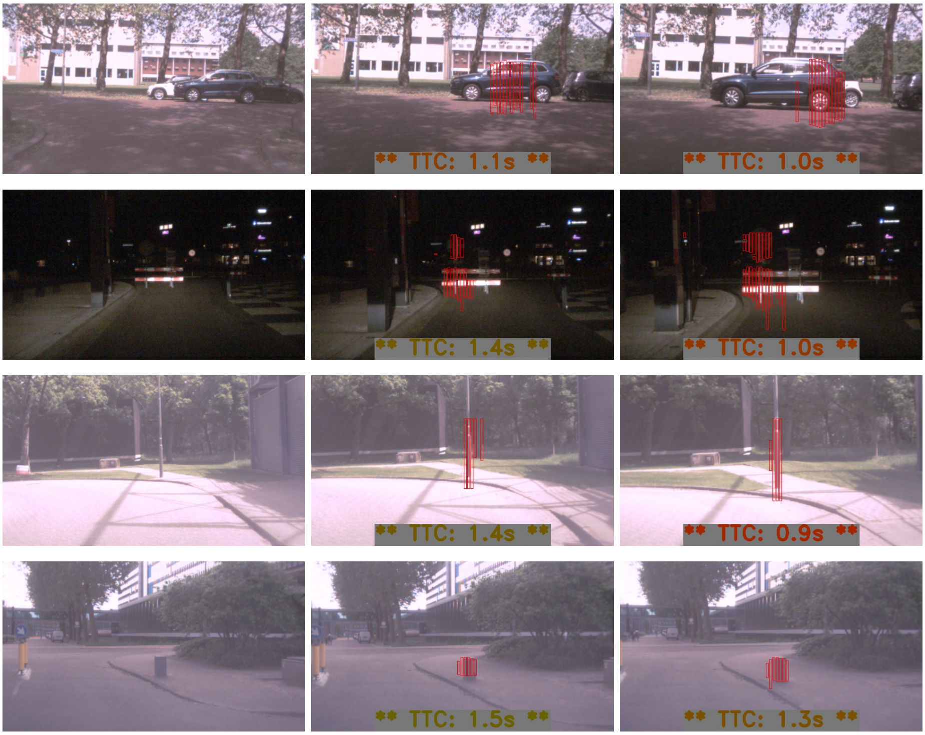


Figure 5.13 — *Examples of our system in action on TUE&ACNL data, in each row three snapshots per collision event. From top to bottom: a car crossing and slowing down; a road works fence at night; a slim lamppost and a low pole at the side of the road.*

5.7 Conclusions

This chapter has presented a vision-based collision warning system for ADAS in intelligent vehicles. The approach is class-agnostic, since it detects general obstacles that lay on a collision trajectory with the ego-vehicle without relying on semantic information. This is in contrast with most current systems, which rely on pre-trained pattern recognition and are limited to predefined object classes or situations. The proposed framework estimates disparity and optical flow from a stereo video stream, extracts stixels, and samples so-called asteroids, based on an uncertainty analysis of the measurement process to model potential collisions. This is all modeled as a Bayesian histogram filter with a time-to-collision versus angle-of-impact state space.

The key contributions of the work in this chapter can be summarized as follows. First, the algorithm is a probabilistic method with a newly introduced particle sampling (asteroids) method. These asteroids are then applied to leverage the efficient and well-known disparity stixels in a generic collision warning system. Second, the fully probabilistic and specialized asteroid approach is not hampered by noisy input data, thereby facilitating a reduction in computational effort for calculating dense optical flow in the larger system. Third, the asteroid system employs a state space that is newly designed and specific for collision warning, based on axes over impact time and angle. These two physical quantities directly offer insight in the relevant collision dynamics of the surrounding objects in the scene, in contrast to commonly used static occupancy grids.

The evaluation provided the following key results:

A. Quantitative performance: The analysis on the KITTI and PreScanStereoCollision datasets has shown that our ASTEROID system detects all potential collisions with obstacles higher than 0.40 m and does not generate false warnings with ttc <2.3 s on the KITTI data.

B. Performance gain by probabilistic design: The proposed probabilistic approach can handle relatively low-quality input, such as noisy disparity and/or flow data. Specifically, using the smallest FlowNet2 version (*FN2-s*) yields practically identical collision warning performance when compared to using the full version, while the former one can be executed roughly 17 times faster than the latter (7 ms on an GTX-1080 GPU), at the cost of a reduction in the pixel-wise optical-flow performance of up to a factor of 2 [123].

C. System robustness: The system did not generate any false warnings on the TUE&ACNL data, showing capabilities to handle bright sunny weather with sharp shadows, high contrast, temporal flicker, direct sunlight and reflections. Moreover, the system was still able to generate warnings for near-collisions in the dark on nighttime data, although they typically occurred shorter ahead of the collision (earliest at $ttc \approx$1.5 s).

D. Timing: The algorithmic blocks as presented in this chapter can be executed to process data at 15-17 fps on a desktop PC. The bottleneck is the stixel-overlap analysis (15-45 ms), while the other blocks of the proposed algorithm require maximally 4 ms.

Summarizing, the research in this chapter addresses collision warning that successfully leverages probabilistic modeling of uncertain disparity and optical flow measurements, where the representation facilitates fusion with other ADAS processes. The algorithm can be employed on affordable hardware and has no requirements whatsoever on car connectivity or HD maps. The validation utilizes both known and new public data, featuring both real-world and simulated data and includes scenarios at nighttime.

The previous chapters have presented research in stand-alone settings, showing the step-by-step development of algorithms for freespace segmentation and for generic collision warning, validated in isolated test environments. The next chapter presents two different projects where newly developed stixel-based research is integrated into larger systems. More specifically, it is practically embedded in a larger context of scene modeling for military surveillance and for safety monitoring in semi-automated driving. Both examples in that chapter are supported by the development of live prototypes, showcasing how the presented strategies capture scene information that can be leveraged in real-world applications.

Chapter 6 Application prototypes

6.1 Introduction

The previous chapters of this thesis have presented our stixel-based research for freespace segmentation and generic collision warning. Additionally, we have developed specialized extended versions of the Stixel World algorithm that have been integrated into two different real-world prototypes: Change Detection 2.0 (CD2.0) and the demonstrator of Vision-Inspired Driver-assistance Systems (VI-DAS), a European Horizon 2020 project. These prototypes represent two different aspects in a broad range of applications, namely 3D modeling of static scenes and dynamic analysis at object-level. The different objectives of these applications evidently lead to different system constraints and requirements.

Static 3D scene modeling: The change detection system for the Netherlands Ministry of Defence concerns accurate 3D modeling of static scenes around the ego-vehicle, so that the current state of the environment can be compared to the historic state which was captured earlier and stored in a database. The change analysis requires pixel-level image registration, which is hampered by the fact that the images typically are captured with large viewpoint differences. Dealing with these viewpoint changes is best performed via a 3D model instead of relying on 2D images.

Dynamic analysis at object-level: The VI-DAS demonstrator concerns regular, yet dynamic traffic scenarios in which specific types of traffic participants need to be detected, classified, localized and tracked. This information can be registered in a local dynamic map for high-level risk analysis. Additionally, it can be used directly to generate forward collision warnings, whenever the forward path of the ego-vehicle is obstructed. To this end, our modules provide 3D obstacle detection and classification by fusing the results of a deep neural network for semantic segmentation with the 3D stixels and adding a low-weight clustering step. This provides a generic object-level representation to analyze collision dynamics.

Together, these prototypes illustrate the versatility and flexibility of the applied modeling in this thesis with regard to its operational domain. Both problem settings and our prototyped solutions will be addressed individually in the following sections. Each section will first provide a more detailed explanation of the system context and the resulting technological challenges and constraints, followed by a discussion of related work. Consecutively, the proposed method and our evaluation approach are presented, completed with the presentation and discussion of the results and conclusion.

6.2 Project A: Change Detection 2.0

When driving the same route twice with a vehicle that has an on-board camera, it is interesting to find similarities or differences with respect to previously gathered information about the scenery. Detecting similarities is relevant for map registration purposes, *i.e.*, localizing the ego-vehicle into a pre-recorded high-definition digital map. Identifying differences has a related purpose, for instance to keep the aforementioned digital map up-to-date automatically, or to detect anomalies during safety inspection.

A relevant use case of the safety scenario is the semi-automated surveillance in a military context. More specifically, it concerns the problem of identifying road-side bombs, so-called Improvised Explosive Devices (IEDs), during military and peace-keeping missions. These bombs have been the most important cause of casualties for NAVO troops in the last few years. Therefore, the Netherlands Ministry of Defence, amongst others, are interested in research for Countering Improvised Explosive Devices (CIED). The work in this section has been developed specifically for that purpose and was supported by the Defence Expertise Centre Counter-IED of the Netherlands Ministry of Defence, receiving funding within the National Technology Program under the project title ”Change detection 2.0 for countering IEDs”. The next subsection will discuss the situational context in more detail.

6.2.1 Context of military surveillance

When military personnel has to keep a hostile area safe, a common strategy is to send out patrol vehicles that drive around to both show their presence and to inspect and identify potential safety hazards. For these kind of patrols, a concrete threat is that opponents use Improvised Explosive Devices (IEDs): bombs that are hidden at the roadside or buried below it, and that are meant to be detonated if the patrol car is nearby. In general, the detonation of an IED is not automated, but done manually, both for the lack of advanced equipment and to avoid causing local casualties. A common strategy is that the location of the IED is marked with a simple object (such as an empty soda bottle), which looks mundane to the unsuspecting eye, but can be spotted from far away with binoculars by the hostile party that set up the IED. Then, the IED can be detonated from a distance when the patrol car passes the marker.

It is not feasible to detect the explosive device itself when relying on a camera-based computer vision system, since the device is typically hidden in a container, a bush or even buried in the ground. However, the corresponding marker object is

The work in this section has been presented at VISAPP 2018 [42].

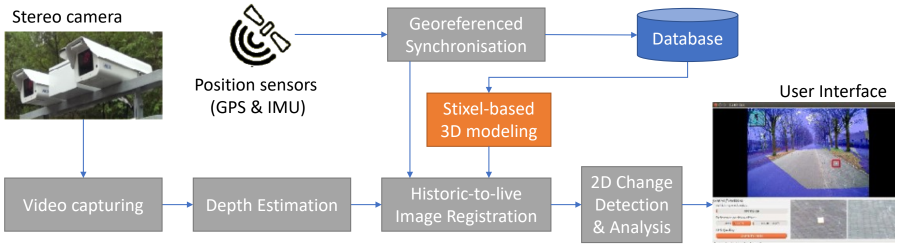

Stereo camera
Position sensors (GPS & IMU)
Georeferenced Synchronisation
Database
Stixel-based 3D modeling
User Interface
Video capturing
Depth Estimation
Historic-to-live Image Registration
2D Change Detection & Analysis

Figure 6.1 — *Conceptual diagram of the full Change Detection 2.0 system, adapted from [124].*

clearly visible. By design, it can be spotted in visible light and even found from a considerable distance. Moreover, the marker will only be present after a bomb has been hidden, so that it will cause a change in the scene appearance if an area has been visited regularly. This makes these markers important and relevant objects for an automated detection system that assists the military personnel during their surveillance.

The layout of the project was put forth by the military itself, naturally driven by the desire to support soldiers under difficult operational conditions, by flagging small changes that can indicate potentially life-threatening situations. Any kind of support of an automated change detection system that complements the limited concentration and memory capacity of humans is highly appreciated.

Additionally, separate military-funded projects address sensor modalities other than camera vision, like ground-penetrating radar and light-polarization sensors. The long-term goal is to develop a multi-modal detection system consisting of the systems that are most promising individually and could complement each other in an integrated setting. However, this is all beyond the scope of the work presented here, which is purely focused on utilizing visual-light cameras. This part is considered the basis for a whole class of CIED systems.

6.2.2 Technological goals, challenges and constraints

The high-level objective of the system is to automatically detect changes in scenes under surveillance using camera footage. To this end, images that are captured during a surveillance task should be compared to historic images that were captured from the same environment on previous inspections and stored in a database. The conceptual diagram of this system is presented in Figure 6.1, to illustrate the overall processing stages in which the work of this chapter has to operate and will be embedded. The figure shows that the system relies on GPS and IMU sensors to accurately measure and store the position and viewing direction of each captured image. This allows to rapidly retrieve the position-relevant historic image from the database, by finding the image that was captured from the closest distance to the viewpoint position of the actual live image.

Since the relevant changes for the system in this context cannot be fully a-

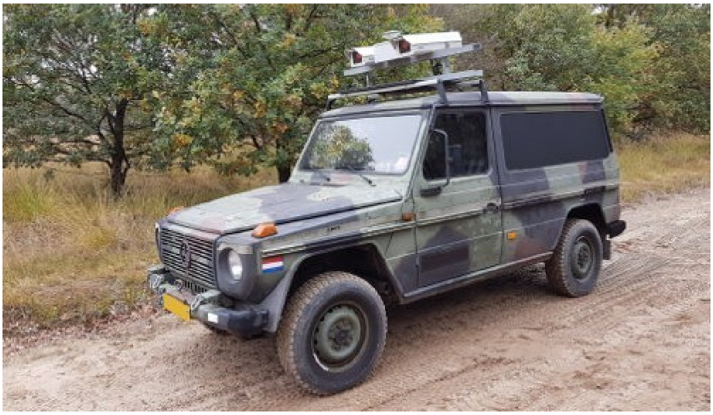


Figure 6.2 — *Prototype vehicle with the stereo camera system on top.*

priori defined, the system cannot be limited to specific classes of objects. Hence, it cannot rely on pretrained object detection, but instead it should be able to detect generic changes. This requires a generic change analysis and will be executed in 2D images based on the analysis of pixels and their neighborhoods.

To this end, an image registration method should process the historic image in such a way that it is aligned with the live image. Moreover, this image registration should be pixel-accurate for the change detection process. However, surveillance routes in a dangerous military context present several special operational conditions for which the system needs high robustness.

Large viewpoint differences between historic and live images are unavoidable, since the vehicle needs to drive different routes through rough terrain. This creates a parallax effect during image capturing, such that objects in a 2D image of the scene can change their relative position. Figure 6.3 provides an example where this phenomenon occurs. As a consequence, it is impossible to use common 2D image registration processes.

Natural changes in the scene appearance (parked cars that have disappeared, varying weather conditions, shadows at different times of the day) impede the use of methods based on pixel-flow (optical displacements of pixels from one to another frame), since not all content is present in both the live and the historic recording.

Life-threatening situations that occur when the vehicle has stopped too late, ask for high sensitivity to small details at a large distance when operating the vehicle at a reasonable speed. The design considerations are explained in more detail in [124], the resulting prototype vehicle (Figure 6.2) is equipped with a pair of ultra-high resolution (exceeding digital 4K television), high dynamic-range cameras in a predefined stereo setup, capturing 5 stereo frames per second.

These operational conditions lead to the following technological constraints.

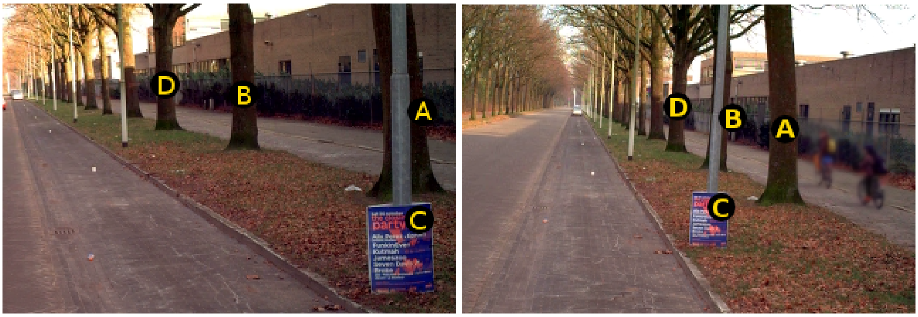


Figure 6.3 — *Illustration of the parallax effect, caused by large viewpoint differences: object C appears either in front of object A or B in the 2D images, although none of them have moved in reality.*

- The large viewpoint difference, up to several meters, requires 3D image modeling using the stereo camera.
- Small changes of interest require processing at a high resolution and locally-accurate scene modeling, allowing change detection also above the ground plane.
- 3D Modeling should be computationally efficient to allow sufficient computational budget for the other system modules and ensure real-time execution.

Therefore, the objective of the work in this chapter is to present an efficient yet accurate 3D scene model, allowing to render the scene from a viewpoint with a large displacement, e.g. several meters, to facilitate a pixel-accurate image registration process for pixel-level image comparison.

The next section will briefly discuss strategies from related literature in the light of the above-mentioned challenges and constraints.

6.2.3 Related work

The specified problem, expressed in computer vision terms, is a combination of image registration and 3D scene modeling. This section briefly discusses related work from both fields.

Image registration is required to map live images to the historic images that were captured at an earlier time. Besides this time difference, there are also viewpoint differences because the driven trajectories of the two inspections also differ in geographic positions of the vehicle expressed in world coordinates. Registration under viewpoint differences has to address vehicle (and thus camera) displacement and perspective camera distortion, for which various strategies exist in literature. One broad category performs all the processing in the 2D image plane. To handle the perspective distortion, it is possible to divide the 2D image into sufficiently small parts such that an affine homography transform can be found for each part individually. For example, the smaller parts can originate from a fixed grid [125],

[126], or from segmenting the scene into planar regions [127]. Even though these strategies can in theory handle the parallax issues as discussed in Section 6.2.2, the available methods cannot be performed fast enough for real-time application within our context.

The second category of registration makes use of 3D processing. This avoids the parallax issue, since a change in camera viewpoint has no influence on the relative positions of the objects in the scene. Hence, change of viewpoint corresponds to a single rigid transformation of the 3D scene. Consecutively, the transformed 3D scene is projected back to 2D in an image rendering step. An example of such an approach is the hierarchical alignment presented in [128]. This system builds a textured polygon model of (parts of) a scene, transforms it to the desired viewpoint and renders a registered 2D image. The approach exploits a polygon model, circumvents any parallax issues and can be executed in real time. This has motivated us to adopt this approach as the inspiration for this research, and extend the part of the scene that the system can analyze.

3D modeling of a scene can be performed with several different strategies. The principal choice is between processing raw 3D point clouds or mesh-like 3D models. Although it takes additional processing to compute the 3D model initially, mesh-based models bring the benefit of (1) generally improving efficiency because of their locality in the processing pipeline, and (2) avoiding noise and holes that will typically arise when processing a raw 3D point cloud. Even though advanced methods exist to enhance raw depth data, they typically rely on a filtering stage that exploits the temporal coherence of a video stream with a high frame rate [129], which is unfeasible in our pipeline given its relatively low frame rate. For these reasons, we opt for building a 3D mesh in the modeling task. A similar strategy was applied in [128].

Different approaches exist to build a 3D model from a point cloud. For instance, highly accurate models are generated from nearby captured data in [130], [131]. However, their processing time is in the order of seconds [131] or minutes [130] and their level of detail is not required in our context.

Many alternative modeling strategies have been developed from super-pixel methods, which have been predominantly designed for image segmentation instead of 3D modeling [132]. In general, they have been designed to process 2D color images, such as LV [133], SLIC [134] and SEOF [135]. Each of these algorithms has its own various extensions to incorporate disparity or 3D point-cloud data. For example, extensions for LV involve LVPCS [136], MLVS [137] and GBIS+D [138], while SLIC is extended in StereoSLIC [139] and SEOF is modified into SEOF+D [138].

Although all of these methods can provide relevant super-pixel segmentations, the resulting super-pixels are shaped irregularly, yielding an inefficient representation. Moreover, they need to be calculated on the whole image at once. For trading-off modeling flexibility against optimality and computational complexity, the Stixel World algorithm was introduced [96]. This probabilistic super-pixel method was designed specifically for the context of intelligent vehicles, aiming at providing a compact yet robust representation of traffic scenes in front of a vehicle, which can be generated efficiently in real time. The Stixel World algorithm relies on disparity data to partition scenes into vertically stacked, rectangular patches, each of them with a certain height and 3D position with respect to the camera sensor. These rectangular patches are labeled as either *ground* or *obstacle* during the segmentation process, thereby providing a two-class semantic segmentation as well as a 3D representation. On top of that, the Stixel World algorithm can be computed efficiently by casting it as a column-wise dynamic programming process, which facilitates parallel execution [96].

Another specialized scene modeling method for intelligent vehicles relies on 3D voxels [140]. It generates and removes cubic voxels to handle the dynamic aspect of a traffic scene and stores them efficiently in an octree-based fashion. The method relies on tracking the voxels over time and does not employ any real-world regularization. However, our camera operates at a low frame rate (±6 FPS) and at a much higher resolution (above HD instead of VGA). Therefore, the method presented in [140] is likely to provide noisy and spurious false detections on our system platform.

With these constraints, we propose to avoid overly-detailed modeling and time-filtered approaches, and instead rely on efficient modeling that incorporates real-world prior knowledge for within-frame regularization. Our system will extend the work of [128] such that the analysis can cover the full scene instead of the ground only. This is realized by customizing the efficient and robust disparity Stixel World algorithm, as explained in the following.

6.2.4 Method overview

The system diagram involving 3D scene modeling and image registration is presented in Figure 6.4, which is a subset of the complete system depicted in Figure 6.1. The historic stereo images, stored in a database, are modeled in 3D using their disparity measurements. This process consists of two submodules, namely one that estimates a potentially curved ground model and one that models erect obstacles. Consecutively, the corresponding RGB texture is projected on the 3D modeled mesh. In parallel, the 3D transformation between the historic and the live image is calculated, using RGB and disparity data. The textured historic 3D model and the live 3D transformation are used to generate a synthetic 2D view of the historic image, as if it would have been captured from the live viewpoint. Next, this synthetic historic image and the live image can be compared and analyzed in the pixel-level change detection block.

The work presented here concentrates on the 3D obstacle modeling to facilitate generating a synthetic image from the historic data that has a high degree of pixel-level correspondences to the live images. The spline-based ground-plane modeling strategy and a change detection strategy are presented in [128]. The system in [141] addresses estimating the 3D transformation and the synthetic rendering of the historic image.

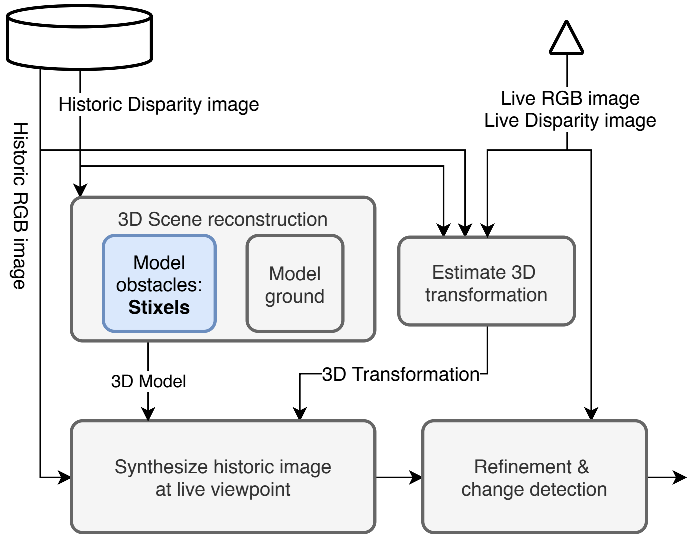


Figure 6.4 — *Conceptual diagram of the Change Detection subsystem. The focus of the current work is the block that models obstacles in 3D via a customized stixel representation (depicted in blue).*

6.2.5 Customized Stixel World Model

The previous version of this Change Detection system could only register the ground-plane region and not obstacles, thereby solely limiting the change analysis to the ground plane. This work presents a strategy to incorporate the obstacles within the same analysis, extending the system scope to the full scene. To this end, we propose a modified version of the Stixel World algorithm. This algorithm has been originally developed to produce efficient medium-level geometric representations, specifically for automotive applications [96]. It takes the disparity image as an input, and generates rectangular patches that model obstacles as fronto-parallel surfaces, as described earlier in Section 2.2.

To exploit stixels to generate a 3D mesh for the change detection application using image registration under large viewpoint differences, we extend the algorithm with slanting, interpolation and masking, successively described in the following subsections. Although these extensions require additional computations compared to the original Stixel World algorithm, our design inherently limits the impact of that on the computational load. Namely, these calculations are performed at the level of stixels. As a result, the processing operates on a couple of hundred well-defined rectangular superpixels, instead of on more than a million raw, noisy pixel-level measurements. Our additions therefore improve the representation accuracy without a severe influence on the computational efficiency.

A. Slanting

The original Stixel World algorithm provides a model for the scene geometry as a collection of fronto-parallel surfaces, called stixels. This is a sufficient approximation for ADAS applications such as freespace or object detection. However, the current goal is to create a textured 3D mesh to generate images from different viewpoints. As a consequence, the flat fronto-parallel surfaces are insufficient, since they are invisible when viewed sidewards (perpendicular to the driving direction). Therefore, we propose to calculate stixels that are slanted over their vertical axis, for an improved fit to the actual surface that they are representing. A top-down view of this process is illustrated in Figure 6.5 A and B.

Stixel slanting is achieved by fitting a line through all (u, d) pairs in a stixel, where u is the column index and d the disparity value of all valid disparity measurements within a stixel rectangle. We have chosen to ignore the row index v in this process for three reasons. Firstly, the horizontal axis is the dominant direction in the viewpoint change, since the camera of the patrol car will be always mounted at the same height and the car will be on the same ground level. Secondly, the Stixel World optimization process already addresses geometric changes in the vertical direction by introducing separate stixels. Thirdly, this strategy reduces the problem to fitting a 2D line instead of a 3D plane.

The line fit itself is achieved by performing a singular value decomposition (SVD) on the collection of (u, d) points within a stixel, solving an over-determined least-squares problem. The result of the line-fit process provides each stixel with new disparity values for its left and right sides, whereas first it had solely one representative disparity value. The stixel surface is still considered rigid, and hence not curved (solely rotated), so that intermediate disparity values can be interpolated linearly between d_{left} and d_{right}, depending on the u-coordinate of interest. To increase the robustness of the slanting process, the algorithm requires that (1) a stixel has a certain minimal height ($h(\boldsymbol{s})$), (2) a stixel contains a minimal share of valid disparity measurements ($N_{\text{valid}}(\boldsymbol{s})$) within its total pixel count ($N_{\text{all}}(\boldsymbol{s})$), and (3) the resulting slanting slope ($\alpha_{\text{slant}}(\boldsymbol{s})$) is within practical boundaries. The actual boundary values depend on the system configuration regarding applied resolution and stixel width. For the work in this section, the relevant values were determined empirically and are provided in Table 6.1. If these criteria are not satisfied, the stixel is kept fronto-parallel to largely prevent the introduction of artifacts into the 3D model.

Table 6.1 — *Criteria for the slanting procedure in the experiments in this section.*

Parameter	Bound
Stixel height (in pixels)	$h(\boldsymbol{s}) \geq 45$
Share of valid disparity measurements in $\boldsymbol{s}$	$N_{\text{valid}}(\boldsymbol{s})/N_{\text{all}}(\boldsymbol{s}) \geq 0.6$
Slanting slope (in pixel-disparity per column)	$\lvert\alpha_{\text{slant}}(\boldsymbol{s})\rvert \leq 1.0$

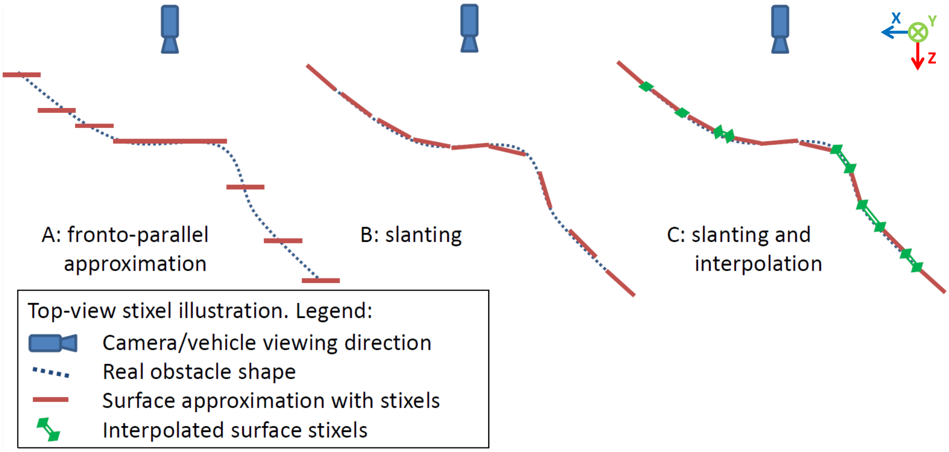

X
Y
Z
A: fronto-parallel approximation
B: slanting
C: slanting and interpolation
Top-view stixel illustration. Legend:
Camera/vehicle viewing direction
Real obstacle shape
Surface approximation with stixels
Interpolated surface stixels

Figure 6.5 — *Schematic illustration of stixel slanting and horizontal interpolation. Note the vehicle-coordinate system at the top right: the z-axis is pointing forward along the camera viewing direction (downwards in the drawing), the x-axis is pointing to the right from the vehicle (left in the drawing) and the y-axis is pointing downwards for the vehicle, which corresponds to pointing into the page in the drawing.*

B. Interpolation

Stixels are calculated as rectangular patches that are adjacently positioned in image columns. However, neighboring stixels may not be adjacent when they are projected to 3D. The reason is that adjacent pixels can represent points that are far apart in the real world, due to diverging camera imaging rays. Also, stixels are straight surfaces of fixed pixel width, thereby limiting the shape that they can model in the real world. The slanting process does not ensure connected stixels, since it only changes the orientation of an otherwise rigid straight approximation of the actual local obstacle surface. Therefore, an interpolation step is performed after the slanting process. In this step, additional stixels are generated to horizontally connect adjacent stixels if they are within reasonable bounds, as illustrated in Figure 6.5 C. The bounds that should be satisfied depend on the positions of the stixels in 3D, which are specified as follows. For two stixels s_1 and s_2 with their distance to the camera in meters as z_1 and z_2 that are adjacent in the 2D image, interpolation is applied if and only if $|z_1 - z_2|/z_1 \; < \; 0.02$. The interpolation process greatly reduces the amount of holes in the 3D mesh, which in turn facilitates projecting a larger textured area, thus improving the image synthesis. Figure 6.6 illustrates the effect of this interpolation in practice.

C. Masking

Stixels are calculated within a fixed horizontal grid, consisting of stixel columns that are 7 pixels wide in the input image. This means that vertical stixel boundaries are not necessarily aligned with real-world surface boundaries: a single stixel

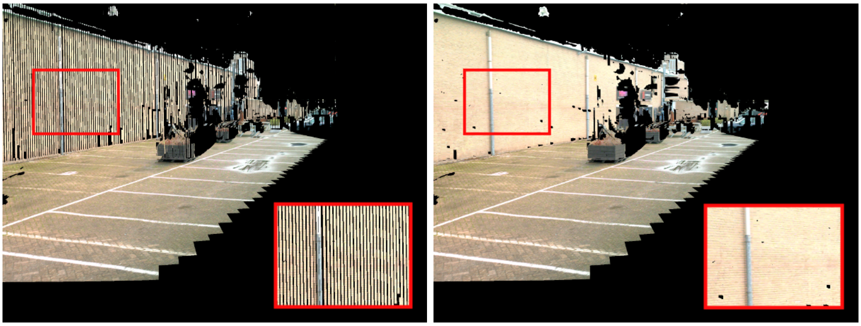


Figure 6.6 — *Illustration of stixel interpolation in practice and corresponding synthesis improvement.*

rectangle can contain part of an object and part of the background. This may not be relevant in the originally envisioned application such as obstacle detection. However, it can cause crucial artifacts for the intended actual texture mapping, which should facilitate pixel-level image comparison. Therefore, we apply a mask to the texture on the stixel surface prior to image rendering from a new viewpoint. All pixels within a stixel rectangle whose disparity values do not match the (possibly slanted) stixel surface, are considered invalid and are ignored in the texture mapping, as illustrated in Figure 6.7. More specifically, a pixel $p_{u,v}$ is considered valid for texture projection if and only if $(d_{u,v} - \hat{d}_u)/\hat{d}_u \leq 0.20$, where $d_{u,v}$ is the original disparity value at location (u, v) and $\hat{d}_u$ the approximated disparity value in the slanted stixel. Effectively, this constraint removes outliers with respect to the line-fitting process.

An alternative strategy would be to use stixels of a smaller width. However, a smaller stixel width (a) reduces the robustness of the stixel calculation process (since fewer disparity data points are available to fit a surface on), (b) harms the efficiency (since more columns have to be analyzed and more stixels will be generated) and (c) still does not address the core of the problem (since the horizontal stixel grid remains fixed). Additionally, making the horizontal grid adaptive would reduce the elegant representation efficiency of the Stixel World approach, thereby harming the parallel computation. In conclusion, we deem it a reasonable trade-off to remove some of the background texture.

6.2.6 Validation data, metrics and results

A. Datasets

The proposed system has been evaluated on two datasets, each consisting of several sequences. The first set has a focus on slanted surfaces, such as depicted in Figure 6.6. The videos in this dataset are used to verify the added value of our extensions to the stixel representation. The other dataset contains structured variations in lateral displacements of the driven trajectory, to explore the range of viewpoint differences that our system can handle.

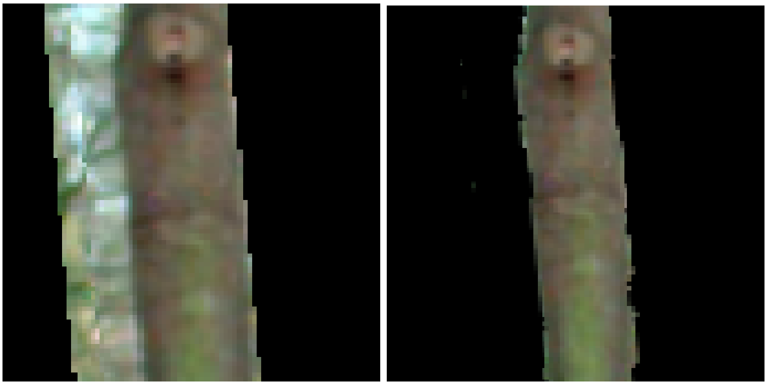


Figure 6.7 — *Illustration of stixel masking in practice, showing a stixel that contains part of a tree and part of the background (left). The pixels representing the background are removed in the masking process, cleaning up the projection (right).*

The data has been recorded with the prototype vehicle shown in Figure 6.2, which, next to the stereo camera, is also equipped with a GPS/IMU device that serves to provide a geo-reference for the frames of the live recording to those in the database. The recordings took place in urban and industrial environments and the sequences were captured at different days and daytimes to include natural and realistic changes in appearance. Note that the context of this research is specialized, and that, to the best of our knowledge, no suitable public benchmarks are available.

B. Quantification metric

The objective of this system is to facilitate pixel-level image comparisons for detecting changes. Hence, we quantify the performance of the system by annotating points of interest and measuring the offset in pixels in the live and synthesized historic data. To this end, we have manually annotated key points in the live data, and marked those same points in the processed historic data. The first dataset contains about 1,800 annotations in total, the second set about 650.

For each annotation, the Euclidean distance in pixels is calculated between the corresponding marked points in the live and the synthesized historic data (denoted as δp). With those results, the registration accuracy $Acc\,(\delta p \leq \zeta)$ is computed, which is defined as the percentage of annotations that have a pixel offset of maximally ζ pixels. In general and after empirical findings, $\delta p \leq 5$ is considered acceptable for the consecutive processing blocks, given that the input image data has a high-definition resolution of 1920×1440 pixels.

C. Results: ablation study on slanted surfaces

The main quantitative results on the first dataset (which has a focus on slanted surfaces) are depicted in the graphs in Figure 6.8. It presents an ablation study on our proposed geometrical additions to the stixel representation: slanting and

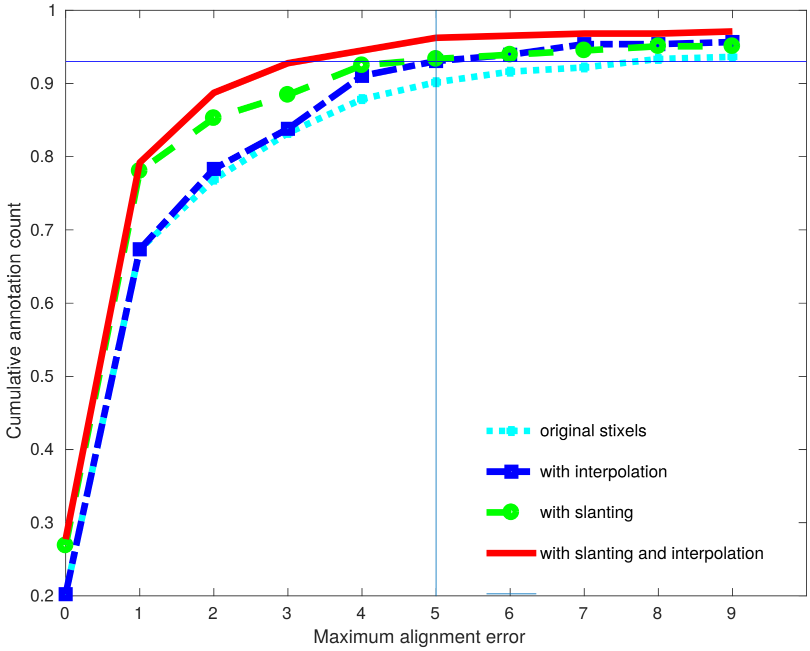


Figure 6.8 — *Quantified ablation study of the proposed additions to the algorithm.*

interpolation. The original Stixel World representation serves as a baseline, and already achieves $Acc\,(\delta p \leq 5)$ of 90 %. Adding either slanting or interpolation results in an improved result of $Acc\,(\delta p \leq 5) = 93\,\%$. When slanting and interpolation are jointly exploited, the performance increases further to $Acc\,(\delta p \leq 5) = 96\,\%$. When measuring more strictly with a smaller margin, this method achieves $Acc\,(\delta p \leq 1) = 79\,\%$, meaning that 79 % of the points is rendered to within 1 pixel of its annotation.

D. Results: effect of lateral displacement

This section presents the results of the tests to quantify how well the proposed system can handle increasingly large viewpoint differences between live and historic data. These experiments are executed on the second dataset, based on driving the same trajectory at different lateral offsets, perpendicular to the viewing direction. The offsets vary from 0 to 700 cm and provide an insight into the robustness and operational range of the system in practice with respect to e.g. driving on different lanes.

The effect of this offset on the alignment accuracy $Acc\,(\delta p \leq 5)$ is summarized in Table 6.2. Additionally, Table 6.3 provides an overview of illustrative examples. Table 6.2 shows that the system achieves a registration score of 97 % under ideal conditions, *i.e.*, if the live and historic data are captured without a lateral offset. It is not surprising that this score is below 100 %, since even without a lateral offset, the system still has to handle changes in appearance (due to changed lighting and weather conditions), while handling inaccuracies that occur in the 3D modeling and texture mapping. Despite these difficulties, the system can still correctly register 90 % of the annotated points even at a viewpoint offset of 160 cm.

The images in Table 6.3 illustrate the degradation of the results for larger offsets. Naturally, the core issue is that the overlapping area in the fields of view of the live and historic data becomes small for large lateral offsets, so that the region decreases in which pixels can be registered. This is clear from the increasing amount of unavailable texture data (shown black) in the registered images (see the third row of Table 6.3). This results in more unmatched points, that all end up in the 15+ error bin in the histograms shown at the bottom row of Table 6.3. Additionally, the effects of inaccuracies in the 3D modeling, such as disparity quantization and the approximation with rectangular patches, become larger at an increased capturing offset.

Table 6.2 — *Registration accuracy for different lateral displacements of the vehicle trajectory, using annotations up to a distance of 50 meters from the ego-vehicle.*

Lateral offset	0 cm	160 cm	350 cm	530 cm	700 cm
Acc $(\delta p \leq 5)$	97 %	90 %	71 %	53 %	20 %

6.2.7 Prototype deployment

The full system of Figure 6.4 was employed on a platform that was mounted in the prototype vehicle shown in Figure 6.2. This included the disparity estimation stage (not shown in the diagram) and the change detection modules beyond the scope of the experiments reported in the current section. The modules of the registration process are divided over three stages that are executed in parallel in a pipelined fashion to maximize the system throughput. The obtained execution times are presented in Table 6.4. If the scheduling overhead is taken into account as well, the full system operates at 4 fps. This throughput rate is near real-time operation and sufficient for proving the concept in a live demonstration, since the UHD-resolution stereo camera operates at 6 fps.

Table 6.3 — *Example results of the proposed registration while increasing lateral displacement. First row: live images (to which historic data should be registered); second and third row: the closest historic images and their disparity; fourth row: resulting synthesized images, where black regions denote areas outside the field of view, missing disparity data or rejected texture during the masking process. Last row: alignment error histograms for a specific lateral displacement using all images in the dataset with that displacement.*

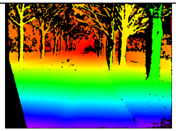
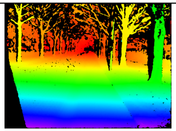
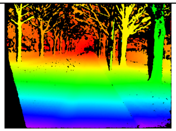
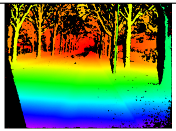
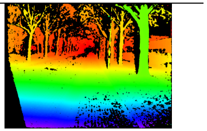
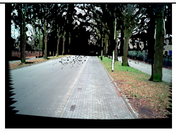
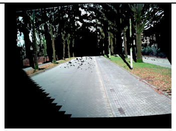
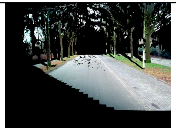
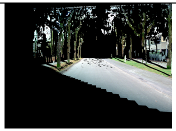
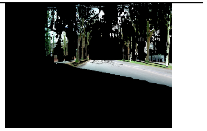
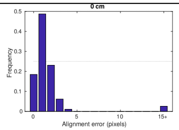
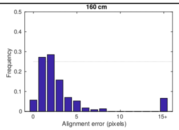
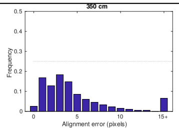
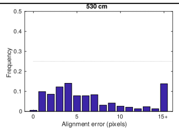
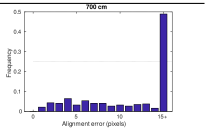
Lateral displacement	0 cm	160 cm	350 cm	530 cm	700 cm
Live image					
Historic image					
Historic disparity					
Registered image (result)					
Histogram with alignment errors					

Table 6.4 — *Execution times of the proposed registration approach, both when stages are executed in isolation and when all are running simultaneously (full load). The (*)-tasks in Stage 2 are executed in parallel on CPU and GPU.*

Execution time	t [ms] Isolated	t [ms] Full load
Stage 1		
GPU: Disparity estimation	90	153
Stage 2		
CPU: 3D ground model (splines)*	130	200
GPU: 3D obstacle model (stixels)*	125	160
Stage 3		
CPU: 3D pose estimation	100	120
GPU: View synthesis	30	46

6.2.8 Conclusion

We have presented a system that constructs a 3D textured model of scenes captured with a high-resolution stereo camera from within a vehicle. This model is highly customized for the specific goal of facilitating pixel-level change detection between live and historic images. The context of this larger system is in military surveillance, where patrol vehicles drive over similar trajectories, however, while capturing images with large viewpoint differences, which nonetheless should be exploited to automatically detect relevant changes in the scene. Our system addresses this problem by modeling the scenes in 3D and generating synthetic images with the historic data, as if they were seen from the viewpoint of the live camera.

To this end, 3D models are created for both the ground surface and the obstacle regions in the images, both exploiting stereo disparity data. Previous work addressed the ground-surface modeling. This work extends that system by adding a customized Stixel World representation, so that the change analysis is no longer limited to the ground surface alone. The customized stixel representation contains slanted surfaces, interpolated stixels and masked texture regions. The combination of these additions maintain the efficiency of the original Stixel World model, while simultaneously improving the modeling accuracy to be more suited and more robust for the varying conditions of the change detection application.

The proposed additions together increase the pixel-level registration accuracy with 6 % on new, manually annotated data. With low lateral offset between live and historic data, our enhanced system scores $Acc\,(\delta p \leq 5) = 97\,\%$ and even achieves $Acc\,(\delta p \leq 1) = 79\,\%$. When the live recording is from an offset of 530 cm, corresponding to more than a regular lane width, the system still obtains a score of $Acc\,(\delta p \leq 5) = 71\,\%$.

In conclusion, the proposed modeling increases the robustness and operational range of the complete change detection system, since it can now analyze the full scene instead of the ground surface alone. In addition to the quantitative evaluation, the system shows promising qualitative results when actively used in a real-time prototype mounted on a vehicle.

6.3 Project B: Vision-Inspired Driver Assistance Systems

This section presents the prototyping of our work in the Vision-Inspired Driver Assistance Systems project[1] (VI-DAS), which was funded within the Horizon 2020 Research and Innovation Programme of the European Commision under Grant Agreement number 690772.

6.3.1 Context of the VI-DAS project

As discussed in Chapter 1, the SAE defined six levels of automation, where Level 0 (L0) corresponds to *no automation* and Level 5 (L5) to *full automation, anytime and everywhere*. At the intermediate levels, different aspects of control of the car are shared between the human driver and the automation system. For example, an automation system with SAE-Level 2 functionality (L2, *occasional automated driving*) can take control of both the vehicle's speed ((de)-accelerating) and its lateral position on the road (steering), provided that certain specific conditions are satisfied. Examples of such conditions can be that automated driving is only supported on clearly lane-marked highways and under favorable weather conditions. The driver can release the operation of the pedals and steering wheel for a while, but should monitor the systems functioning at all times and be able to engage immediately when required for safety. A Level 3 system offers extended functionality. Under certain conditions, it can be in control of speed and steering, it is able to monitor the traffic around the ego-vehicle, while it will warn the driver if the system detects a complicated situation that it cannot handle. In that case, the human driver should be ready to take over control of the vehicle, but it reduces the need for the human driver to be alert at all times.

However, research suggests that drivers may have difficulties adapting to these partially automated stages [10]. Ideally, they should assess the systems performance and be ready to take over manual control. In reality, the drivers are prone to be distracted by performing secondary tasks such as reading or interacting with their phone, which hampers their readiness to take back control at any time [11], [12], so that driver monitoring is required in this context [94].

The VI-DAS project addresses this issue by investigating methods for handling *takeover* (from manual to automated driving) and *handover* (from automated to manual driving) situations for L2 and L3 automation levels. Figure 6.9 shows the full human-machine-transition cycle and the desired high-level system functions being active within each stage. It requires both monitoring the dynamic 3D traffic situation outside and the human driver inside - to asses his or hers readiness for takeover, and alerting if the outside situation requires this.

The remainder of this section is structured as follows. First, we will provide a short explanation of the elements in the VI-DAS system architecture that are relevant to our contribution by causing specific challenges and objectives (Section 6.3.2). The actual proposed solution is presented in Section 6.3.4, while we

[1] The project webpage can be found at `http://www.vi-das.eu/`.

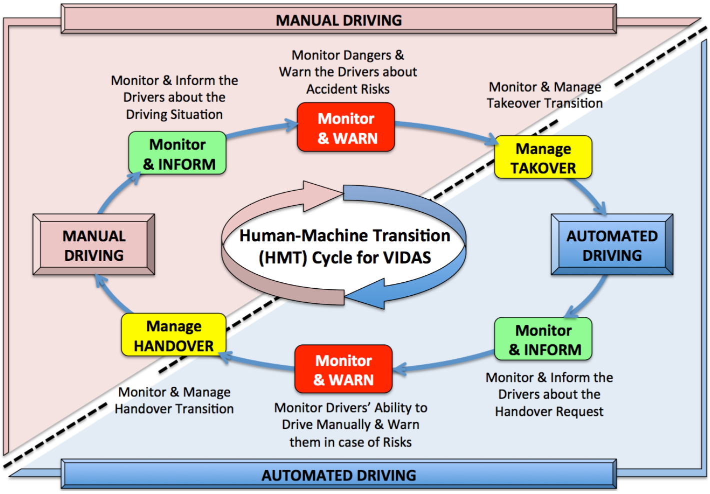

MANUAL DRIVING
Monitor Dangers & Warn the Drivers about Accident Risks
Monitor & Inform the Drivers about the Driving Situation
Monitor & Manage Takeover Transition
Monitor & WARN
Monitor & INFORM
Manage TAKOVER
MANUAL DRIVING
Human-Machine Transition (HMT) Cycle for VIDAS
AUTOMATED DRIVING
Manage HANDOVER
Monitor & WARN
Monitor & INFORM
Monitor & Manage Handover Transition
Monitor Drivers' Ability to Drive Manually & Warn them in case of Risks
Monitor & Inform the Drivers about the Handover Request
AUTOMATED DRIVING

Figure 6.9 — *Human-Machine-Transition cycle as defined for VI-DAS, illustration from [142].*

discuss the scenario, setup and data for evaluation in Section 6.3.5. The results of this qualitative evaluation of our modules are presented in Section 6.3.6, which is followed by a conclusion in Section 6.3.7.

6.3.2 Objectives and challenges within the VI-DAS architecture

Prior to explaining the objectives and challenges addressed in this section, first the conceptual system architecture of VI-DAS is discussed briefly, which is shown in Figure 6.10. Our contribution is a module in the *Outside Sensing* part, to detect obstacles in 3D in the traffic scene outside the car.

Several system modules utilize the output of this sensing part. To facilitate L3 automation, the ego-vehicle should be able to perform path planning in traffic. To this end, the VI-DAS architecture includes a risk-assessment module that predicts likely trajectories of other traffic participants (top right of the *understand*-section in Figure 6.10). This relies upon an HD map with lane-level road-layout information. Dynamic objects in the live scene are registered to that road layout to improve the accuracy of path prediction and the related risk assessment using a local dynamic map (top left of the *understand*-section in Figure 6.10) [143]. Additionally, information on both driver status and scene perception can be send to the cloud (*Connect & Cooperate* section; center bottom region in Figure 6.10) to improve both traffic flow and the safety of other traffic participants via V2X communication [94].

The modules described above require that the *Outside Sensing* part provides object detection, with a 3D bounding box (location and dimension), and also classifies it with semantic labels such as *car* or *pedestrian*. This is fundamentally different from our work presented in the previous chapters, which only distinguishes the coarse levels *freespace*, *obstacles* and *obstacles on a collision trajectory* (to be even more

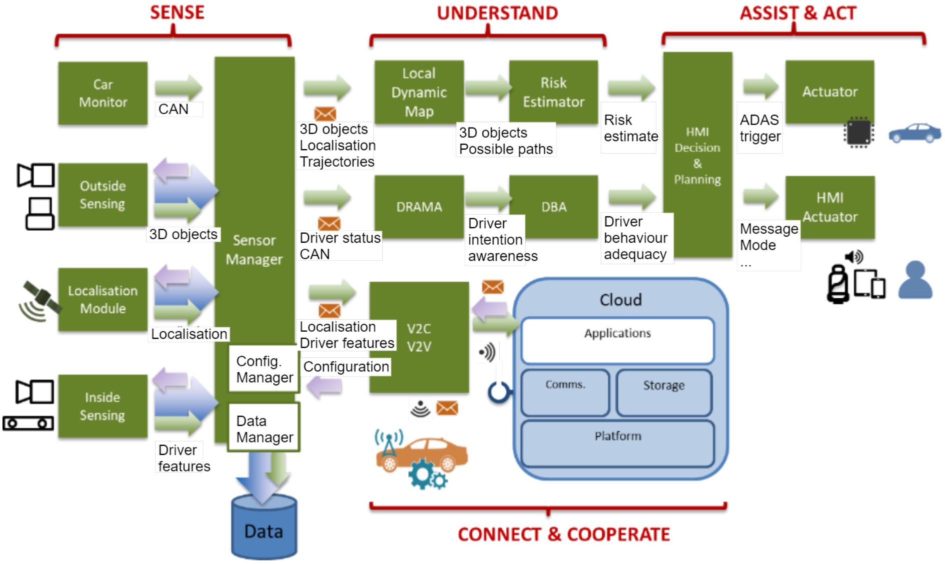


Figure 6.10 — *Conceptual architecture of the VI-DAS project, illustration from [144].*

precise: *matter* on collision course, without any explicit clustering into coherent obstacles, let alone providing a semantic class for them). Nonetheless, we have seen in this thesis and in other related work that the Stixel World algorithm provides an efficient geometric representation of traffic scenes and has shown potential in extended applications in the automotive perception context. Hence, the starting point of the strategy in this section is again the disparity Stixel World algorithm, which however presents a technical challenge. Namely, the Stixel World algorithm yields medium-level, over-segmented and under-classified results, since the stixel superpixels are not clustered into objects and only classified as obstacle or freespace. Similarly, in the work presented in the previous chapters, our analysis was done at either super-object level or at sub-object level. The super-object level has involved separating the region of freespace versus the whole region of obstacles, where all stixels are combined. The sub-object level has been targeting modeling or tracking individual stixels, where multiple stixels can actually belong to the same object. For the current application, stixels of the same object should be clustered, so that the 3D bounding box can be estimated in real-world metric values, and the clusters should be assigned a class label such as *car*, *cyclist* or *pedestrian*.

Next to these functional requirements, the perception model has to adhere to several system constraints for proper integration and operational performance in the VI-DAS architecture. These constraints are listed below.

- Scene-geometry analysis should include object bounding boxes with semantic class labels and real-world metric dimensions.
- Video data is captured with a compact, inexpensive yet automotive-grade

stereo camera.
- Outside-sensing module should offer interfaces in the exploited middleware software system to ensure integration.
- System should execute at near real-time speed, with a frame rate of approximately 6 fps, to provide operation at the targeted vehicle speed.

The next subsection provides a brief discussion on the related work in this context.

6.3.3 Related Work

This subsection briefly presents the work that is directly related to the currently proposed system.

A closely related method is the Semantic Stixel World [97]. In that work, semantic information from a neural network is strongly fused within the cost-optimization function of the disparity Stixel World algorithm, so that the geometry and the semantics are inferred jointly. The authors show that this leads to both an improved geometric modeling and to an improved semantic segmentation. However, in this algorithm, the scene is still over-segmented, since the stixels are not clustered into objects.

Additionally, several preliminary studies related to stixel clustering and semantic classification have been conducted in VI-DAS-related project studies. Research on performing geometric stixel clustering using graph cuts, alone or combined with an Expectation-Maximization stage, showed that the geometrical features were insufficient to provide a stable, reliable object-segmentation result with these algorithms [145]. Follow-up research performed a lightweight geometrical clustering using DBSCAN [146], followed by assigning class labels using a probability distribution over the cluster dimension and its location in the scene with respect to the ego-vehicle. This distribution was a-priori modeled by analyzing the semantic labels and disparity data that is included in the CityScapes training dataset [147]. The evaluation procedure on the Cityscapes validation data showed that the performance of the DBSCAN clustering was reasonable, while the features used to assign the class probabilities were not sufficiently discriminative for reliable semantic classification of the clusters. In other words, the approach provides unreliable semantic labels, but decent cluster shapes [148].

In other recent work, conducted in parallel with ours, the semantic Stixel World optimization function is extended further with a cost penalty on object-instance segmentation. As a result, stixels are assigned a semantic label and are coherently clustered into objects at the same time [99]. This could be an interesting approach for future integration work or a comparison study, but this exploration is beyond the scope of this current research. Instead, the adopted strategy here is a combination of semantic stixels and customized DBSCAN clustering, which is presented in the next subsection.

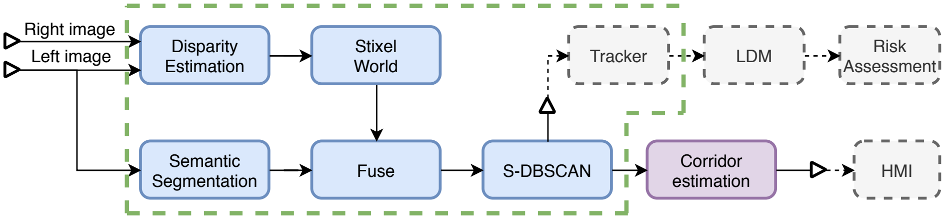


Figure 6.11 — *System diagram of the proposed SSCOD module (the blue blocks), showing its connection to the risk-assessment path and, for the demonstration, its connection towards the Human-Machine Interface (HMI) via the corridor-estimation block (purple). The green dotted line delineates the 'Outside Sensing' module from Figure 6.10.*

6.3.4 Semantic Stixel Clustering for Object Detection (SSCOD)

This section briefly presents our approach to 3D object detection and classification within the VI-DAS project. The method combines the efficient geometric model as generated by the disparity Stixel World algorithm with the output of a neural network that provides pixel-level semantic labeling. Our system pipeline is depicted in Figure 6.11, which is explained below.

Using a stereo camera as input, the first stage estimates disparity, for which we rely on the SGBM algorithm [56]. The disparity is then analyzed with the Stixel World algorithm, resulting in the standard obstacle-versus-ground stixel representation. In parallel, a semantic pixel-level segmentation is performed by executing a deep neural network. More specifically, we deploy an early version of the work presented in [149], which is a deep fully convolutional network (FCN) that is trained on the Cityscapes [147] and Vistas [150] datasets. These two streams of information are combined in a small fusion stage that assigns a class to each stixel by calculating the most common label in the stixel's image rectangle, resulting in semantic stixels. This is analogous to the *disparity first* baseline of [97].

The semantic stixels are then clustered. This processing stage first projects the center point of each semantic stixel to its 3D world coordinate. This metric point cloud is clustered with S-DBSCAN, a newly proposed clustering step. This method is based on the original DBSCAN algorithm [146] and has a customized distance function.

DBSCAN is a lightweight clustering algorithm with the additional benefits that it intrinsically estimates the number of clusters (in contrast to clustering algorithms like k-means) and it relies upon only two parameters. These parameters are (1) the maximum distance between points within the same cluster and (2) the minimum amount of points that should belong to a cluster. As a result, points in a sparsely populated region will be likely not assigned to a cluster, so that DBSCAN generally performs well in identifying dense blobs in noisy data [146].

The original DBSCAN algorithm relies on the Euclidean distance between points. To incorporate the semantic labels, we additionally enforce that all points in a cluster should have the same semantic label, as defined by

$$\text{Distance}_{\text{S-DBSCAN}}(\boldsymbol{s}_1, \boldsymbol{s}_2) \quad = \begin{cases} ||\boldsymbol{x}_1 - \boldsymbol{x}_2||_2, & \text{if } l_1 == l_2; \\ \infty, & \text{otherwise.} \end{cases} \tag{6.1}$$

In the above equation, $\boldsymbol{s}_i$ is a stixel with a center point $\boldsymbol{x}_i$ (expressed in either metric 3D coordinates - meters, or a (u, v, d) triplet - image pixels) and a class label l_i, for $i \in \{1, 2\}$.

With this distance function, the S-DBSCAN algorithm generates clusters of stixels with a class label. The bounding box of the combined stixels in each cluster provides an estimation of the object dimensions and location. Subsequently, these results are passed on to two different modules. Firstly, as discussed in Section 6.3.2, objects of interest (dynamic traffic participants such as cars and pedestrians) are tracked and registered to the local dynamic map (LDM) for risk assessment. Secondly, we propose a rudimentary forward collision warning (FCW) system which utilizes corridor estimation, depicted in purple in Figure 6.11. The FCW estimates the length of the free corridor directly in front of the ego-vehicle. To this end, it calculates a moving average of the closest detected distance of any object that is present in the volume area of 2 meters high and wide, and 50 meters long, in front of the vehicle. This corridor estimation stage provides a tangible result of the SSCOD system that can be communicated to a user interface (Human-Machine Interface, HMI) and facilitates standalone experimental validation via the integrated FCW functionality.

6.3.5 Evaluation strategy

This section presents how and in which traffic situations we have performed the qualitative evaluation of the proposed system.

A. Test scenario

Within the VI-DAS project, natural driving data and real-world accident reports have been analyzed. Based on that, several scenarios have been defined in which the technology for handover/takeover-assist is most relevant. For our forward-looking obstacle-detection system, the primary use case involves forward collision warning in automated driving mode. The scenario that is selected for demonstration is defined as follows. The driver switches on L3 automated driving. The ego-vehicle receives information about a road blockage roughly several hundred meters ahead via V2I communication. The vehicle warns the driver to take back control to handle this special situation. If the driver is paying attention and takes back control, all is considered well. However, if the driver is distracted, which will be detected by the Driver Monitoring System (DMS, included in the inside sensing module), the system will provide a more urgent warning. As the ego-vehicle is getting closer to the obstacle, it becomes critical to have a more accurate measurement of the obstruction location than the one communicated by the cloud service. Therefore, the camera-based system will detect the obstacle and will initiate an emergency breaking procedure, provided that the driver still remains unresponsive.

B. System setup

To integrate the different system modules, the existing RTMaps (*Real-time Multisensor applications*) software package is used[2]. It facilitates synchronized communication of different modular building blocks, which can be developed in C++ or Python. The diagram of our SSCOD system with its internal stages is shown in Figure 6.12. Throughout the presented experiments, we configure S-DBSCAN with the radius parameter $epsilon$ as 30 pixels when relying on (u, v, d) triplets, and an $epsilon$ of 3 m is recommended when employing metric coordinates. In both cases, we set the minimum amount of points of a cluster to 3. The detected obstacle bounding boxes are provided to the corridor-estimation stage. Additionally, our diagram contains modules from different project partners, namely the object tracker developed by Dublin City University, Ireland (for integration with the LDM module, not included) and the driver-monitoring component of the inside sensing module, developed by a collaborative industrial partner (Intel, Germany). Additionally, we provide a component that monitors the state of the vehicle via specific relevant CAN messages (such as speed, acceleration and the state of the autonomous driving mode). All these systems communicate via a socket component to an HMI module of another participating company (Akiani, France). The HMI module resides in a different RTMaps diagram and decides which warnings should be generated (depending on vehicle state, corridor analysis and driver state), for which it can utilize several display devices.

C. Demonstration data and experiments

We have performed a feasibility study by executing several test drives with the SSCOD system active in the car. The drives were performed both in live traffic and on a closed road under controlled traffic conditions at a maximum speed of 30 km/h. Note that in both cases, the SSCOD system was not linked directly to the automated braking system, due to regulatory constraints. We have tested the system both during daytime and nighttime conditions. On top of that, two live demonstrations of our system were given at the ITS European Congress in Helmond and Eindhoven, the Netherlands, June 2019. A visual example of this demonstration event and venue is portrayed by Figure 6.13.

We execute our ASTEROID algorithm as presented in Chapter 5 on the same data to compare the two different approaches as an additional experiment. The evaluation in this chapter is of a qualitative nature, providing a proof-of-concept demonstration at best, since no true positions of obstacles were annotated.

[2] The software webpage can be found at `https://intempora.com/products/rtmaps.html`

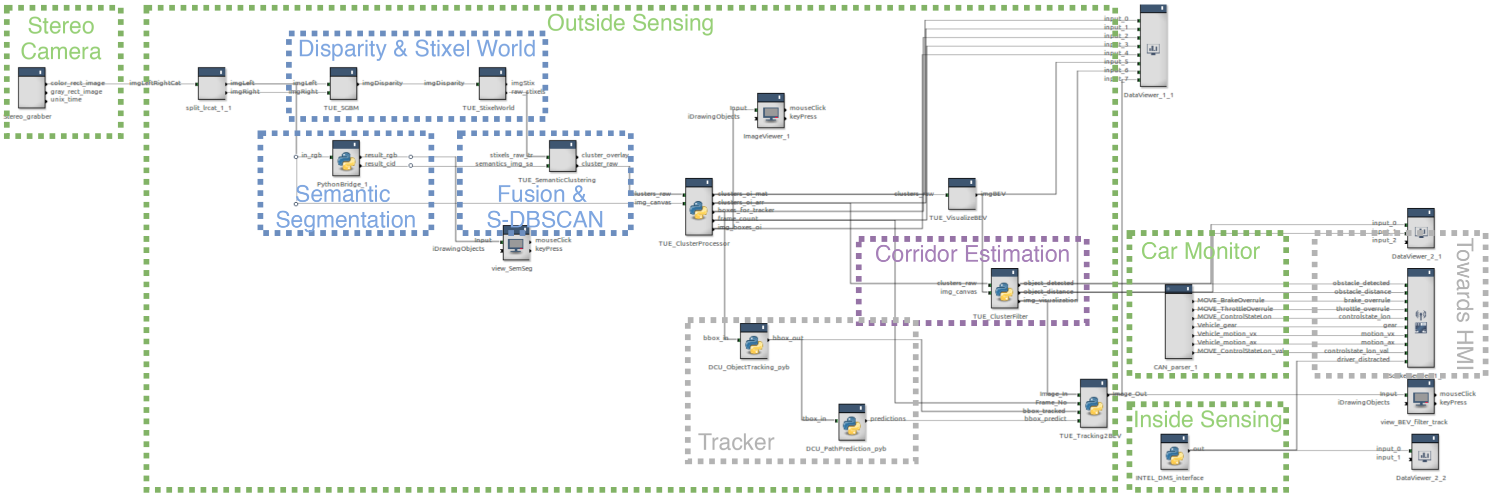


Figure 6.12 — *RTMaps diagram depicting the integrated processing blocks for the VI-DAS demonstrator. The dotted rectangles indicate how the RTMaps components correspond to the modules in Figure 6.10 (in green) and the enclosed stages in Figure 6.11 (in blue, purple and gray). The remaining RTMaps components contain data interfaces to support the integration and visualizations for validation. The diagram contains modules from different project partners, as described in the related text in Subsection 6.3.5 B.*

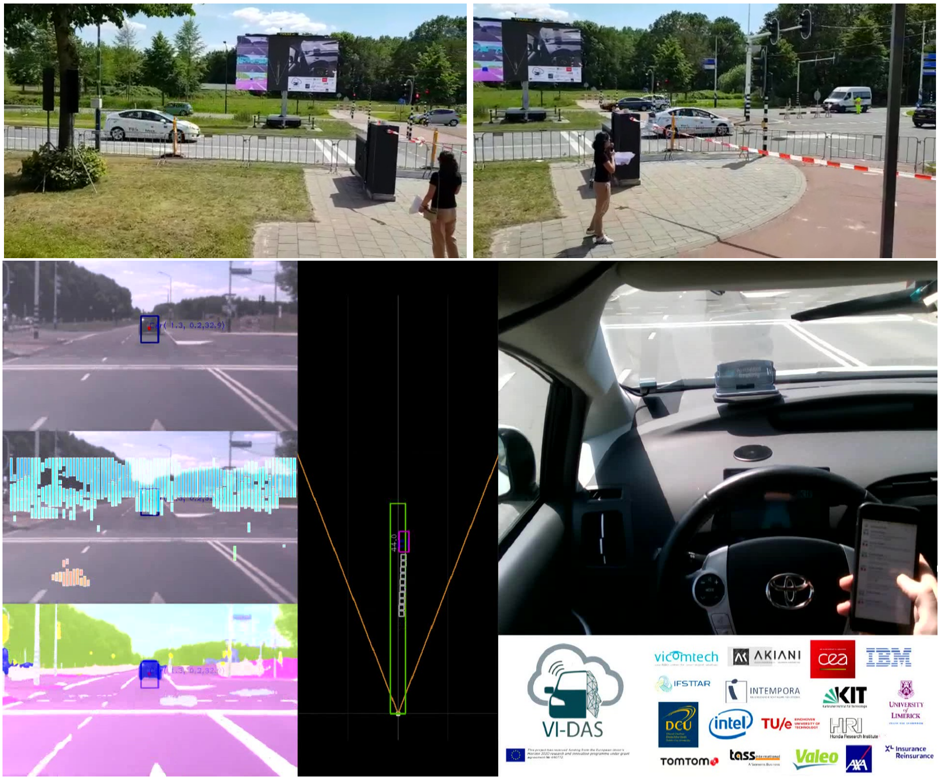


Figure 6.13 — *Live demonstration of the VI-DAS project at the ITS European Congress 2019. Top: two snapshots from the viewpoint of the audience, showing our demonstration vehicle in white, a stopped black vehicle that blocks the road and the large screen for the audience with a live stream of the processing in the car. Bottom: an enlarged example of that processing. The demo narrator is project leader Oihana Otaegui of Vicomtech.*

6.3.6 Qualitative results

Typical results of the experiments in daylight are visualized in Figure 6.14. The SSCOD algorithm detects the obstacle for the first time at a distance of 40-50 meters (as estimated by the system itself). The length of the estimated free corridor is typically overestimated by the corridor-estimation stage, due to the delay introduced by its internal sliding window buffer.

The ASTERIODS approach consistently generates a collision warning for the first time with a ttc between 1.6 and 2.2 seconds, in correspondence with the results presented in Section 5.6. Note that the estimated ttc increases in the snapshots, since the ego-vehicle is slowing down while it is approaching the obstacle, to avoid an actual real-world collision.

The results on nighttime data are illustrated in Figures 6.15 and 6.16. The results of SSCOD during nighttime are less consistent than during the day when inspecting the frame-by-frame detections. However, this is handled correctly by the freespace calculation in the corridor-estimation stage that considers multiple frames. This effect is clearly visible in the example in Figure 6.16. The first two frames displayed in the figure do not visually show a detected obstacle in the current frame, but the system has already detected the car in previous and intermediate frames, so that the free corridor is already decreased from 50 m to 40.9 m and 24.1 m, respectively. As a result, the system displays a warning in red text in the top-view visualization of the scene.

The first distance at which obstacles are detected at nighttime is less consistent when compared to drives during daylight. At night, first detections occur typically at a distance of 30-50 meters. Similarly, the ASTEROID collision warnings are generated closer in time to the potential collision event at night than in daytime. Typically, the first warning is provided at a *ttc* of around 1.0-1.5 seconds. The next section discusses the strengths and pitfalls of both systems.

Discussion

Both the disparity estimation and the semantic scene parsing suffer from the nighttime conditions. For instance, the leftmost scene in Figure 6.17 is full of false stixels due to poor performance of the SGBM algorithm, which has not been tuned for the considered night conditions. However, the ASTEROID system performs well in this situation, which is explained by the tracking functionality and the probabilistic strategy. More specifically, consistent stixels that also correspond to true obstacles, can be tracked and are more stable in the measurements, so that they stand out in the Bayesian filter, whereas the impact of false stixels due to noisy data is mitigated by that same process. In contrast, the SSCOD system does not have this benefit. The FCN suffers from the unseen night conditions, leading to dangerous miss-classifications (like the pedestrian marked as a tree in the third scene of Figure 6.17). It can also lead to false positives in the demo scenario, as visible e.g. in the rightmost scene in Figure 6.17. The network seems to have a bias towards the vehicle class, also under daytime conditions. As a result, small clusters of stixels on roadside obstacles are easily falsely marked as a vehicle, potentially blocking the free corridor and leading to a false warning. This problem may be reduced by either retraining the network, or by employing the strongly fused version of the Semantic Stixel World, which jointly infers a semantic scene geometry from pixel-wise disparity and per-pixel semantic labels [97]. Since ASTEROIDS are class-agnostic, they do not suffer from miss-classification issues, thereby inherently making that approach more robust against new conditions.

6.3.7 Conclusion

We have presented a 3D semantic object detection system supplying its detections to a risk-assessment module, which in turn uses an HD map to do trajectory prediction and planning within the VI-DAS project framework. Since this requires

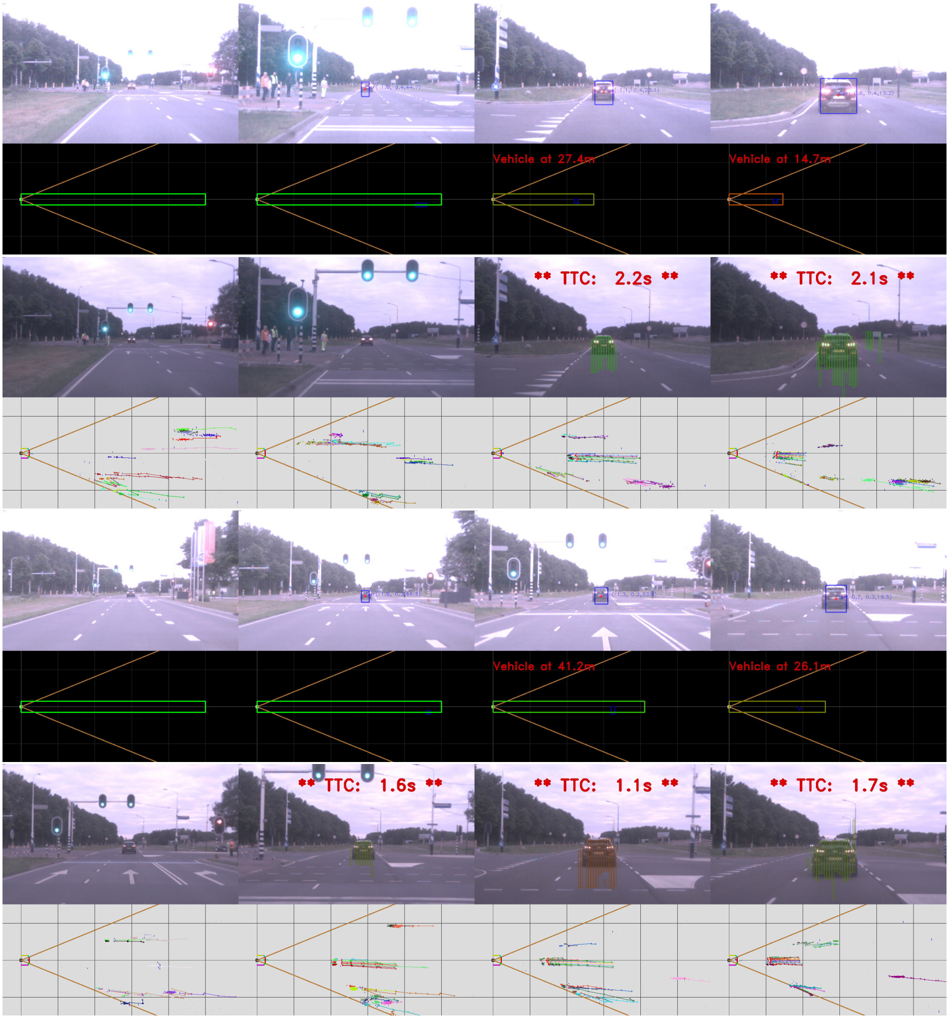


Figure 6.14 — *Comparing SSCOD (Rows 1 and 3) and ASTEROIDS (Rows 2 and 4) in daytime with two demonstration drives at a closed road. The two top rows present snapshots that were captured two seconds apart, the two bottom rows show snapshots of one second apart.*

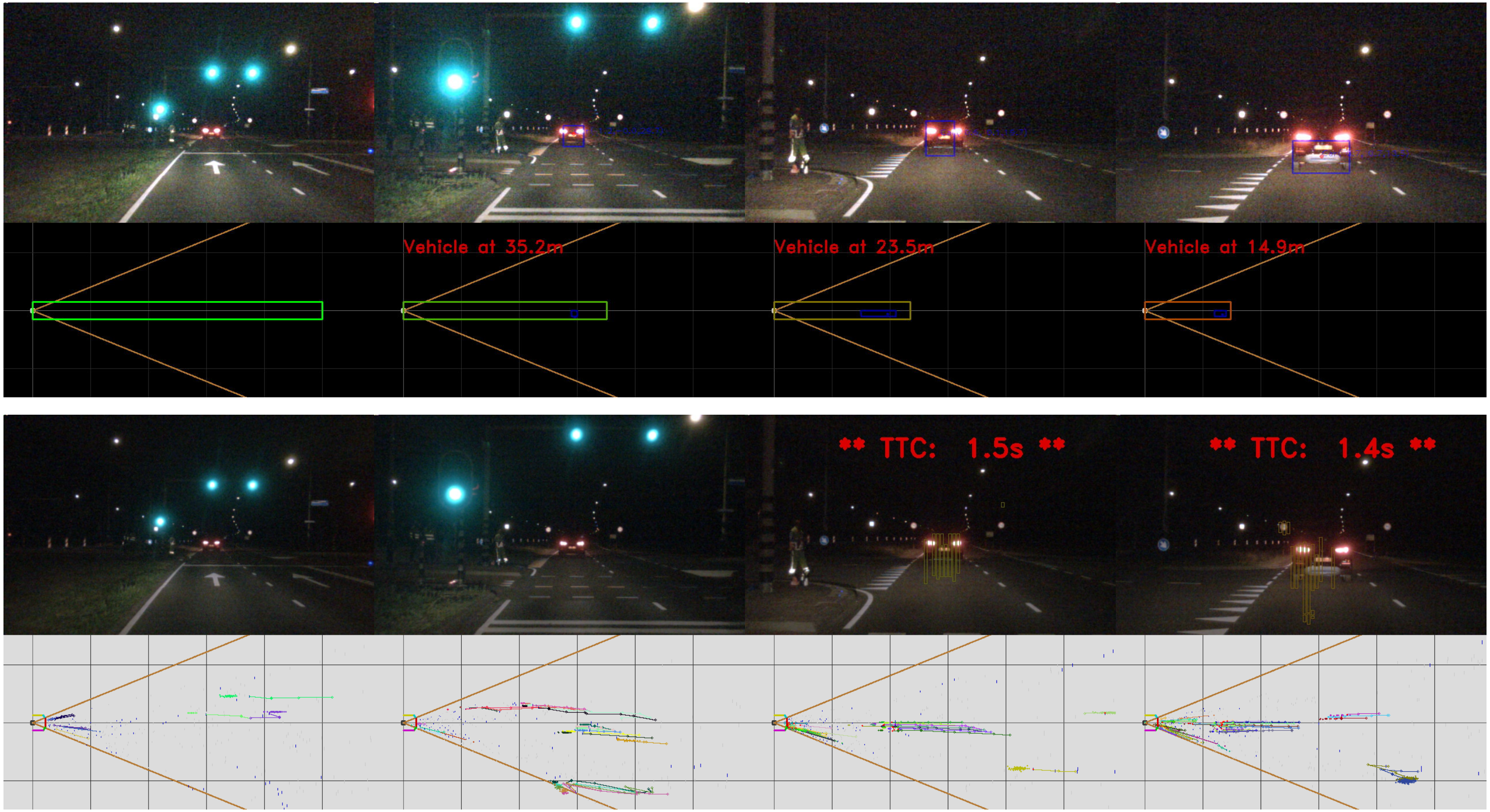


Figure 6.15 — *Comparing SSCOD (top row) and ASTEROIDS (bottom row) at night, with a demonstration drive at a closed road, all these snapshots were captured one second apart.*

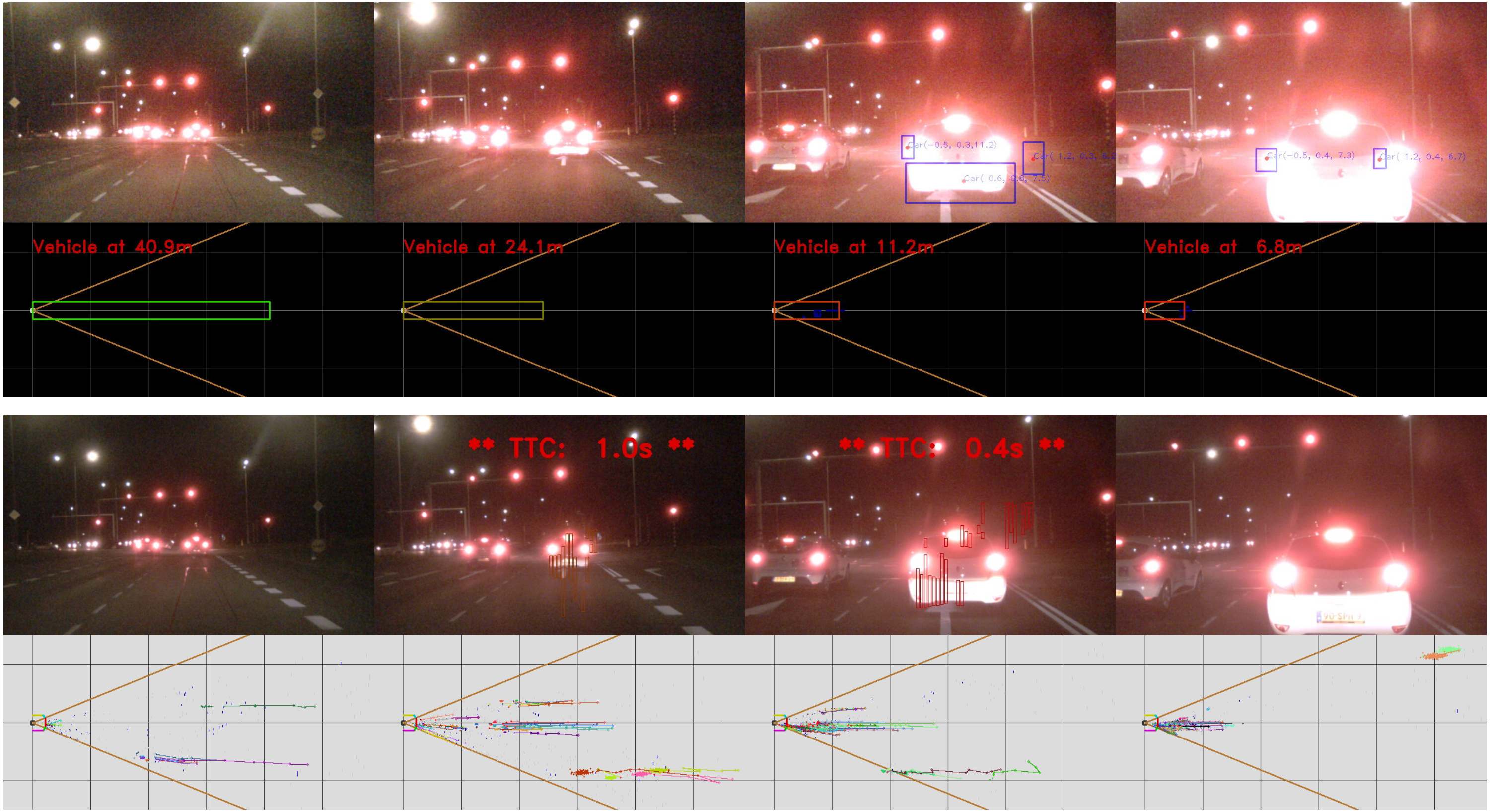


Figure 6.16 — *Comparing SSCOD (top row) and ASTEROIDS (bottom row) at night, with a demonstration drive a in live traffic, all these snapshots were captured one second apart.*

both semantic labels such as pedestrian or vehicle, and a geometric representation, we have fused the outcome of a deep neural network for semantic scene parsing with the Stixel World model, and processed that with a lightweight, customized semantic DBSCAN clustering step, to combine stixels into labeled objects.

All modules are implemented and integrated to facilitate real-time operation within a prototype vehicle. To demonstrate the feasibility of this algorithm, we have used it in combination with a corridor-estimation stage as a Forward Collision Warning (FCW) system, to perform several experiments. These were carried out under nighttime and daytime conditions, and both in live traffic and under controlled conditions on closed roads. The demonstrations indicate a good performance for the test scenario defined in the VI-DAS project, which addresses scene monitoring for handover and takeover scenarios. The FCW system should detect vehicles on the road ahead and warn the driver or initiate an emergency braking maneuver. Even though the evaluation is of a qualitative nature and by no means an automotive-grade validation, the system seems to be able to handle the defined scenarios well under daytime conditions. Even without an a-priori requirement on nighttime conditions, the SSCOD system still shows promising results within the scope of the selected, limited demonstration scenario.

Moreover, we have exploited the demonstration data to perform an additional evaluation of the ASTEROID system of Chapter 5, just for the sake of further comparison. ASTEROIDS seem to perform robustly on the assessed data and are even able to handle the noisy nighttime data without requiring any fine-tuning of the parameters.

The SSCOD and ASTEROIDS algorithms have a fundamentally different approach. Namely, SSCOD uses semantics and measures distance, whereas ASTEROIDS is class-agnostic and measures time. For an FCW system, the ASTEROID system currently outperforms the SSCOD approach. However, the SSCOD system is intended to be a submodule within the VI-DAS project framework, so that a direct comparison is not entirely fair. Unfortunately, the complete VI-DAS risk-assessment pipeline could not be integrated into our experimental vehicle in time to perform a fully fair and more complete comparative study.

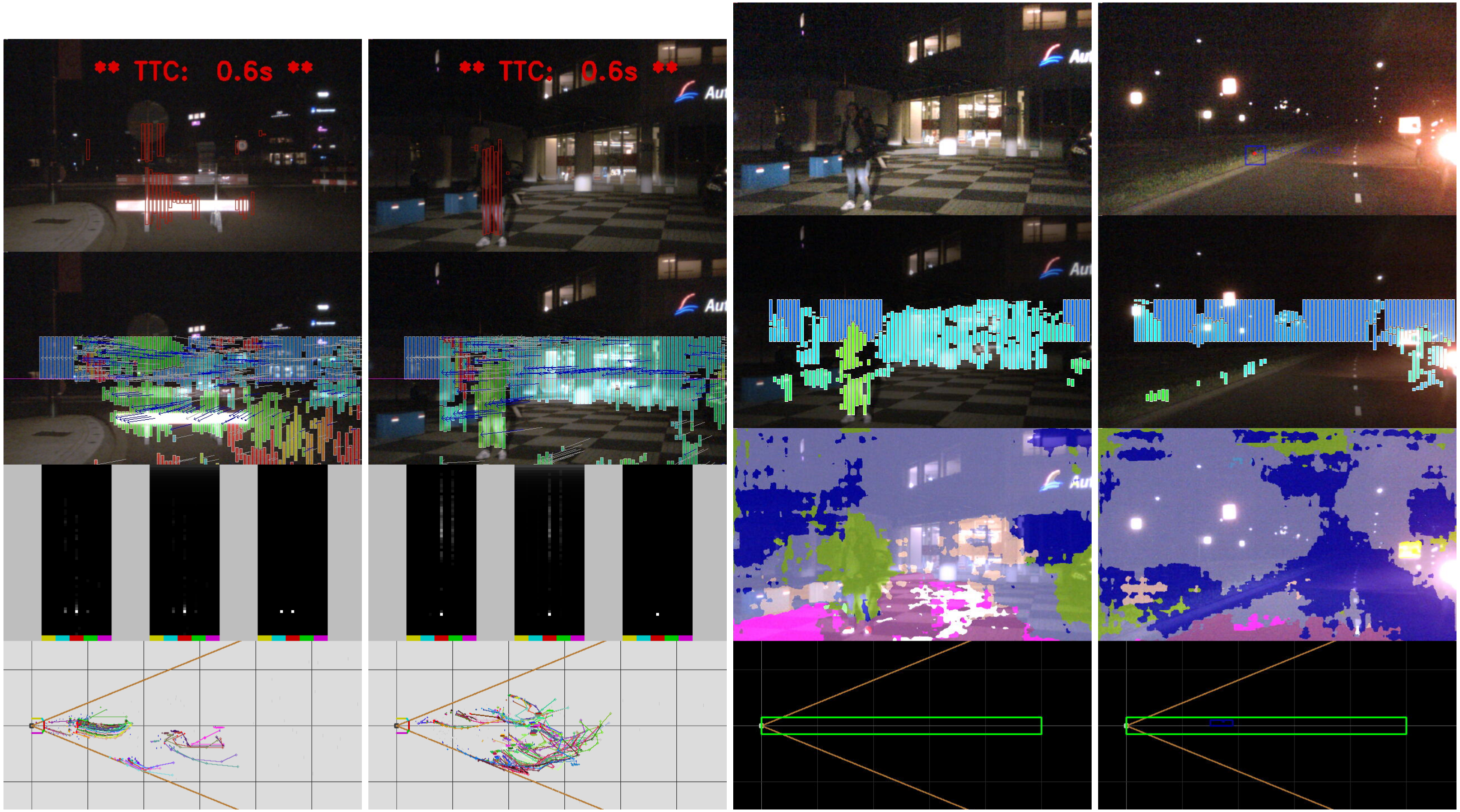


Figure 6.17 — *Examples with intermediate processing steps. The two left columns visualize the ASTEROIDS process, with from top to bottom: (a) images with warning overlay; (b) stixels with flow; (c) three state-space visualizations for every side-of-impact and angle-of-impact (measurement data histogram, collision belief, and CFAR detections); (d) top-view of the scene with tracked asteroids. The two right columns show results of the SSCOD steps, including some issues. Going from top to bottom: (a) SSCOD with no vehicle detection (correct), (b) stixel representation result, (c) semantic scene segmentation with the pedestrians classified as tree and cars on building regions (incorrect), and (d) the top-down view with the estimated free corridor (correct). In the rightmost column, SSCOD falsely detects a vehicle on the road curb, visible from the small blue box in the top image and in the free-corridor visualization at the bottom (as a result, the free corridor is falsely reduced in the subsequent frame).*

Chapter 7 Conclusions

This thesis addresses camera-based freespace segmentation, obstacle modeling and collision warning systems. This final chapter summarizes the conclusions of the individual chapters, discusses the findings regarding the research questions as defined at the start of the thesis and presents an outlook on future developments.

7.1 Conclusions of individual chapters

This section provides an overview of the main conclusions of each chapter.

Chapter 2 has presented a color extension to the disparity-based Stixel World algorithm, to more robustly segment freespace versus obstacles in traffic scenes, by the online learning of color models in a self-supervised fashion. This extension particularly improves the robustness of the segmentation against erroneous disparity estimates, which inevitably occur during challenging low-texture imaging situations, regardless of the quality of the applied stereo camera. The chapter presents two main contributions. First, the Stixel World optimization criterion is extended with a color-based cost term and its related color feature representation. Second, we present a self-supervised online training stage for using a simple color model that is kept representative during operational driving through different scenes. As a key result, in detecting drivable distance (the novel application-inspired metric), the proposed method increases the F_1 score from 0.86 to 0.97. This result clearly indicates that the Color-extended Stixel World method, based on strong fusion of disparity and color modalities, is an accurate and robust method for road versus obstacle segmentation.

Chapter 3 has introduced a stixel-based probabilistic framework for color-based freespace versus obstacle segmentation. Similar to the previous chapter, this research relies on self-supervised online color modeling via disparity analysis. The new contribution in this chapter is the reduced dependency on actual disparity measurements, facilitating a latency reduction of the analysis. To this end, color processing is adopted with an informative color-pair representation, using the first and second mode of an online-adapted indexed color space. This is further enhanced by distance-aware color-histogram processing based on real-world metric pixel surfaces, to address perspective camera distortion. The experiments show that the proposed system improves the quality of the freespace analysis, while simultaneously approximately halving the latency (compared to both disparity-based methods) and the computational load of the freespace segmentation algorithm (compared to the *strong*-fusion approach). The quality improvement is measured by a 4% lower overestimation of freespace compared to the disparity-based baseline. As a result, the proposed system offers a reduced response time when measuring from data acquisition input to data analysis output, combined with an increased accuracy.

Chapter 4 leverages the strength of convolutional neural networks for freespace segmentation. To preserve a low complexity as earlier, again a self-supervised online training scheme is adopted, allowing a network with a small memory footprint and fast execution. This is implemented with a Fully Convolutional Patch Network (FCPN), which is trained in a self-supervised fashion. The experiments show that the proposed algorithms with online training ($F_{\text{max}} \simeq 0.92$ and $AP \simeq 0.98$) outperform the offline reference methods with 5%, both for F_{max} and AP. More importantly, without online training, the FCPN performs worse than the baseline ($F_{\text{max}} = 0.90$) on the new data. This indicates that the online training strategy is a good and efficient proposal to enable the use of a small neural network. The added value of the online framework is most pronounced in the rainy-images subset of the data, where it outperforms the baseline with 4.2%. The FCPN has a low memory footprint and fast inference time, while it is able to handle a wide variety of scenes, requiring neither manually labeled training data, nor disparity estimation in its critical segmentation path.

Chapter 5 presents a vision-based collision warning system for ADAS in intelligent vehicles. The approach is class-agnostic: it detects general obstacles that lay on a collision trajectory with the ego-vehicle without relying on semantic information. The approach in this chapter has three main contributions. First, the proposed algorithm is a probabilistic method with a newly introduced particle sampling (asteroids) method. These asteroids are applied to leverage the efficient and well-known disparity stixels in a generic collision warning system. Second, the fully probabilistic and specialized asteroid approach with error propagation is robust against noisy input data, which considerably reduces the computation of the dense optical flow. Third, the asteroid system employs a state space that is newly designed and specific for collision warning, based on axes over impact time and angle. These two physical quantities directly offer insight in the relevant collision dynamics of the surrounding objects in the scene, in contrast to commonly used static occupancy grids. The evaluations on three different datasets show that the system (a) does not generate any false warnings on the real-world KITTI dataset, (b) detects all collisions except one in a newly simulated dataset, (c) provides error-free performance on a new qualitative real-world dataset with near-collisions during both day- and nighttime.

Chapter 6 describes the integration of specialized and extended versions of the described work into two real-world prototypes.

First, a system is presented that builds a 3D textured model of scenes, captured with a high-resolution stereo camera from a vehicle. This model is highly customized towards the specific goal of facilitating pixel-level change detection between live and historic images. The context of the complete system is in military surveillance, where patrol vehicles repeatedly drive over similar trajectories. Image comparisons at the same location present large viewpoint differences, which severely complicate change detection. The proposed solution is able to model the scenes in 3D and generates synthetic images with the historic data from the viewpoint of the live camera, all in real-time operation. The proposed solution exploits a customized stixel representation that both contains slanted surfaces, interpolated stixels and masked texture regions. The combination of these additions maintain the efficiency of the original Stixel World model, while simultaneously improving the modeling accuracy for varying conditions. The proposed additions together increase the pixel-level registration accuracy with 6 % on new, manually annotated data. With low lateral viewpoint offset between live and historic data, the enhanced system scores $Acc\,(\delta p \leq 5) = 97\,\%$ and even $Acc\,(\delta p \leq 1) = 79\,\%$, where δp represents the allowed pixel margin. When the live recording is from an offset exceeding a regular lane width, the system still achieves reasonable accuracy of about 70 %.

Second, a 3D semantic object-detection system is presented that provides its detections to a risk-assessment module, which uses an HD map for trajectory prediction and planning within the VI-DAS (an EU H2020 project) framework. Since this requires both semantic labels such as car and vehicle and a geometric representation, the proposed algorithm fuses a deep neural network for semantic scene parsing with the Stixel World model, and processes that with a lightweight, customized semantic DBSCAN clustering step. The first contribution in this project is the fusion of the semantic information from the neural network with the geometry of the stixels. Second, a new semantic cost function for the point clustering step is proposed, which together with the semantic stixels of the first contribution, results in 3D semantic object detection. All modules have been implemented and integrated to fully operate in live modus in a car. For demonstration, a corridor-estimation module as a Forward Collision Warning system is designed and added to show suitable performance. The demonstrations during both day- and night-time indicate a good performance for the test scenario defined in the VI-DAS project, which addresses scene monitoring for handover and takeover scenarios in partially automated driving.

7.2 Discussion of the findings on the research questions

As stated in Section 1.3, the objective of this thesis is to improve computer vision-based scene modeling in three different but related fields: freespace segmentation, static obstacle modeling and dynamic obstacle analysis for collision warning. These research directions are clearly visible in the contributions presented in this thesis: Chapters 2, 3 and 4 concern improved freespace segmentation; Chapters 2 and 6A address improved modeling of static-obstacle regions of the scene; Chapters 5 and 6B present collision warning systems. This section answers the research questions as presented in Section 1.4.5 using the performed work in this thesis.

RQ1 Improvement of the performance of freespace segmentation systems under adverse imaging conditions and their robustness towards changing conditions and environments.

- RQ1a: *What are the common artifacts in current freespace algorithms and what is their root cause?*

  The research in the first three chapters of this thesis discusses stereo camera-based approaches, which is a commonly used sensing modality for computer vision-based ADAS. The stereo data are exploited for geometry analysis via disparity estimation. For this disparity signal, a popular state-of-the-art algorithm to delineate ground versus obstacles in traffic scenes is the disparity Stixel World algorithm. A pitfall of this algorithm is that it is prone to producing false stixels on disparity artifacts, reducing the detected freespace. These disparity artifacts originate from adverse imaging conditions such as sharp shadows, light reflections, rain droplets or image regions that contain either no texture or strongly repetitive patterns, which are all realistic issues in everyday scenes. Additionally, algorithms that rely on machine learning, for instance via pretrained color models for freespace, typically fail when encountering situations that are either unseen (when the algorithm does not have enough capacity to generalize), or uncommon situations (when the algorithm does not have enough capacity to keep all relevant information at hand).

- RQ1b: *In what way can different data modalities from a stereo camera be jointly leveraged in this context?*

  Typically, the artifacts described above are not directly clearly discernible from the color images themselves, or at least show less indication of the presence of a potential obstacle at the artifact location. Hence, the strategy in our freespace segmentation work is to combine disparity and color analysis, for which three different ways are presented. Chapter 2 presents a system, employing *strong* fusion of color and disparity to avoid false stixel detections, thereby improving the F_1 score of drivable distance detection from 0.86 to 0.97. The key of this improvement is a more accurate freespace segmentation due to the additional color analysis of the surrounding scene. Additionally, Chapters 3 and 4 present specialized methods that rely on *weak* fusion of color and disparity data. The richer color analyses in these chapters provide a more accurate scene model, contributing to the freespace segmentation. The experiments in those chapters show that this provides up to 4-5% improvement in several quality-oriented metrics. On top of this, these systems facilitate a latency gain, since they do not require actual disparity estimation to analyze the current frame. As a result, the response time from video input to obtaining the freespace output can be halved without reducing the quality of the results.

Chapter 7

- RQ1c: *How can color models be extended for freespace segmentation, while retaining a low complexity?*

  Both the color-extended (Ch. 2) and the color-based (Ch. 3) Stixel World algorithms rely on histograms to model the two classes of interest (ground and obstacles). Our research has further addressed the optimization of three aspects, namely (1) the selection of preprocessing such as histogram equalization, (2) adopting an appropriate color space (such as RGB, HS(I) or customized) and potential quantization, and (3) choosing the most suitable representation values to store in the histograms (such as color modes or gradient strengths). Additionally, we have proposed to create the histograms in a distance-aware fashion, so that colors nearby the ego-vehicle are properly balanced with those from image regions representing areas far away. These extensions effectively reduce the overestimation of freespace with 4% (*i.e.*, it resolves otherwise missed obstacles), compared to disparity baselines. Lastly, we have exploited a small convolutional neural network as a better color-encoding strategy to more accurately extract freespace and obstacle appearance information (Ch. 4). The objective of this strategy is to increase the encoding capacity and flexibility of the color modeling, since some of the experiments of preceding chapters have illustrated potential gains in this direction. As a result, the overall quality performance of the freespace segmentation is increased with about 4%, specifically on dark, rainy frames, when compared to the baseline system.

- RQ1d: *What is the added value of self-supervised online learning for increasing robustness?*

  When a freespace segmentation system is implemented with low-complexity color modeling, its capacity is typically insufficient for addressing large variations in scenes, which lowers the system robustness. The proposed freespace segmentation systems presented in Chapters 2, 3 and 4 rely on a self-supervised online learning strategy that updates the information in the color modeling during operational driving, so that the limited capacity of the model is primarily focused on the actual scene appearance. Several of our experiments show that the complete freespace segmentation framework can operate robustly in new situations, even when the modeling capacity is limited. This difference is most pronounced when using the small neural network that was successfully employed on the relatively homogeneous KITTI-road dataset, but failed on our newly recorded dataset with more variation. However, when enhanced with our online training strategy, it then actually outperforms the other baselines on that dataset with 4-5 %.

  The online strategy requires a supervision strategy that can also be executed during operational driving. To this end, we rely on the disparity Stixel World algorithm and gather the potentially partially erroneous labels over several frames to create a small dataset. The experiments show that our modeling can generalize from these samples to improve the analysis of the new frame, at least when the errors are within reasonable boundaries (*i.e.*, not dominant in pixel count and/or not persistent over all frames in the small online training data).

For this and the above questions on robustness of freespace segmentation, we have released the new EHV-road datasets publicly in three batches (2014, 2015, 2017). It is a relatively small dataset (around 500 annotated frames in total), but it is focused on relevant corner cases with difficult imaging conditions or structures, making it relevant to test specific pitfalls. By using this dataset, we have been able to show an improved robustness for most of these cases.

RQ2 Leverage of computer vision to improve dynamic collision-warning functionality.

- RQ2a: *How can stereo disparity imaging be exploited for collision warnings?*

  Two of the contributions in this thesis provide information for collision warning: SSCOD (Chapter 6B) and ASTEROIDS (Chapter 5 and 6B).

  Both algorithms have a fundamentally different approach: SSCOD uses semantics and measures distance, while ASTEROIDS is class-agnostic and measures time. SSCOD has been developed as a submodule within the VI-DAS system, and only its FCW capabilities are qualitatively validated in this thesis, showing promising results in a live prototype vehicle for the project scenario in the context of partially automated driving. The ASTEROID system is evaluated on simulated data to incorporate collisions, real-world data without collisions (the KITTI tracking dataset) and newly recorded data with many near-collisions. All experiments confirm the feasibility of the proposed approach in timely detecting collisions without generating false warnings.

  The key of using disparity analysis for both systems is to leverage a disparity Stixel World algorithm that provides a model of the scene geometry in an efficient way. In this regards, the additional, particularly attractive property of the Stixel World algorithm is that it provides a generic analysis of any scene, without specific pretrained knowledge about classes or scenarios (e.g. for ASTEROIDS). Furthermore, it can also be extended to incorporate semantic knowledge, if desirable and available (e.g. for SSCOD in VI-DAS).

  To leverage the disparity stixels for generic collision warning in ASTEROIDS, several extensions are required, such as (1) assigning the static stixels with dynamic flow information and then tracking them over multiple frames, and (2) generating a full error propagation to translate the discrete and quantized stixels into smooth probabilistic processing. Specifically, the successful results on the nighttime data illustrate the robustness of our method. Even though the per-frame stixel segmentation consists mostly of erroneous segmentations, the ASTEROID system reliably generates solely correct collision warnings.

- RQ2b: *Is it possible to prepare an ADAS module such that future sensor fusion can be exploited beneficially?*

  All of our work has been performed using a generic stereo camera and does not rely on costly hardware such as RTK-GPS, Lidar, or concepts that require external infrastructure and large-scale societal adoption like V2V communication or HD maps. It should be noted that in both of the experimental systems presented in Chapter 6, other modules do rely on such sensors for positioning or extra information from external communication, but this extra information is not used for our algorithms. This adheres both to the objective of relying on affordable hardware and to the objective of presenting cost-efficient systems that can provide relevant information in a standalone setup. Nonetheless, several of our algorithms have been designed to facilitate fusion with other modules, if they would be available and could offer relevant data. For instance, the FCPN-based freespace segmentation algorithm of Chapter 4 does not merely generate binary masks, but instead provides pixel-level confidences that are suitable for probabilistic fusion. Likewise, the ASTEROID collision warning system of Chapter 5 is fully probabilistic, allowing it to either incorporate information of other modules, or feed subsequent modules with its results including a measure of confidence. Therefore, although this was not experimentally verified, the author has the opinion that the presented systems offer generic ways of integration into larger systems, which then can leverage complementary sensor modalities such as Radar.

- RQ2c: *How should dynamic measurement data be represented efficiently for direct support to collision warning?*

  To generate collision warnings that are actually valuable in mitigating the collision, the originating direction of the danger is crucial information, together with its time of impact. To readily assess these dynamics, the proposed ASTEROID system exploits a newly designed state space with two dimensions, namely angle-of-impact and time-to-collision. As a result, the stereo camera measurements are directly translated to the relevant physical values for collision warning from any direction. The evaluation setup is not extensive enough for a full quantitative evaluation of the multi-directional approach (due to the use of a single forward-looking stereo camera), but qualitative experiments have shown the expected and promising behavior for multi-directional collision warning. The other presented collision warning system has addressed forward collision warning, using semantic stixel clustering for object detection (SSCOD). For the addressed use case of the project VI-DAS, the only relevant measure is the distance to any blocking vehicle in the path of the ego-vehicle, without the need of a speed or time measurement. Therefore, this system employs a basic and efficient strategy of filtering consecutive distance measurements towards tracked vehicles in front of the ego-vehicle.

RQ3 Exploration of real-time applications of AI and 3D geometry in traffic-scene and road-scene analysis.

- RQ3a: *What methods can reduce the computational requirements of neural networks to facilitate deployment in real-world systems?*

  Typically, most state-of-the-art systems employing neural networks rely on architectures that are of increasing size and complexity, thereby hampering real-world embedded deployment. In contrast, the work in this thesis contains several optimized employment aspects, mainly to allow for networks that are several orders of magnitude smaller than the commonly used alternatives. The key strategy of the proposed algorithms is leveraging strengths of conventional, non-AI algorithms to support the analysis (a so-called *hybrid* approach), instead of developing fully end-to-end models where all intelligence is included in the neural network, which is surely considerably more complex.

  First, for the freespace segmentation in Chapter 4, we adopt a small neural network and exploit overtraining of the network, to adapt it to changing environmental conditions during operational driving. The training labels for this purpose are being generated by a conventional algorithm, resulting in a hybrid system that combines the flexibility of the neural network, the strength of the training method and the core reliability of the conventional method. A point of interest for future research for a system that relies on online training is the required amount of computational resources for the training process. At present, this is typically not optimized in embedded platforms. A further future complicating point is the assessment of the safety certification of a system that can update itself after deployment.

  Second, the ASTEROID system of Chapter 5 consists of conventional processing steps such as the well-known Bayesian histogram filter and a CFAR detector, while the optical flow is generated with a state-of-the-art neural network. We have found that the experiments indicate that the smallest version of FlowNet2 is sufficient for ASTEROIDS, which executes 17 times faster than the full version of that network. This performance gain is at the expense of a reduced pixel-level accuracy that is halved compared to the largest version. Fortunately and by intentional design, the drop in this specific performance metric is clearly addressed and circumvented in ASTEROIDS. Most importantly, the results of the ASTEROIDS experiments illustrate that conventional methods can provide reliable results, even with inaccurate data from small efficient neural networks.

- RQ3b: *How can 3D scene geometry be modeled efficiently and accurately, to make it suitable for real-time synthetic view rendering?*

  Two of our algorithms concern modeling the obstacle region of traffic scenes, treating everything as static information. The integrated work in the Change Detection 2.0 project concerns improving the representation power and accuracy of obstacles in the scene (Chapter 6 A). To this end, the stixel models are extended by (1) incorporating a slanting angle to better align fronto-parallel stixels with slanted surfaces, (2) interpolate between adjacent stixels and (3) mask away background content within stixel rectangles. This customized stixel modeling facilitates rendering textured views from the scene, as if taken from a different viewpoint, which can then be used for change detection analysis. Additionally, the proposed work on the color-extended Stixel World algorithm provides better obstacle modeling, even though it is mainly evaluated for improving freespace segmentation, while it is not exploited for live-view rendering (Chapter 2).

7.3 Discussion and outlook

The main contributions to ADAS in this thesis are relying on a combination of computer vision and AI techniques, while bearing efficiency in mind for realizing real-world applicable systems. Since the rise of complex neural networks in the field of artificial intelligence for computer vision analysis, they are hampered in their employability onto embedded platforms, since the resource requirements have typically risen as fast as, or even faster than the increase in performance. To mitigate that issue, our strategy is to rely on hybrid systems that leverage both traditional computation and the methods from artificial intelligence, so that the complexity of the AI building blocks can be reduced by relying on the modeling that the traditional methods provide. This trend of *designing for resource-constrained platforms* is expected to grow in the coming years. This is already visible in recent contributions that provide small and efficient neural networks with competitive task performance, such as the SqueezeNet [151] and EfficientNet [152]. One particular interesting upcoming field is that of neural architecture search, and its subfield hardware-aware neural architecture search (HA-NAS). HA-NAS designs neural networks automatically case-by-case, by jointly optimizing task performance (such as freespace segmentation) and resource efficiency (such as latency on a target embedded device), using increasingly efficient search and optimization techniques. By further developing such strategies that provide efficiency-by-design, more elaborate functionality and further performance gains can be expected, bringing applicable AI closer to reality.

As a result of the above, more ADAS are expected to become available at a large scale, as they are and will be integrated in newer car models in different price ranges. Observing current trends and prospectives on the mobility market, our preferred design philosophy is on pursuing *small incremental steps of automation* by extending functionality and the operational domain in alternating fashion, moving gradually up the SAE levels of automation, in contrast to pursuing an L5 vehicle straight away. More specifically, the safety and accessibility of mobility can also be improved with the intermediate partial automation via ADAS functionality to which this thesis is related. Extrapolating from commonly accepted systems such as anti-lock breaking system (ABS) and cruise control, via increasingly popular blind-spot warning systems, automated parking, and lane-keep assist, this increasingly automated support of drivers seems a natural extension for various reasons. First, it offers the potential of maintaining a successful car-making business with attractive price-performance trade-offs. Secondly, it facilitates a smooth path for societal deployment and acceptance. Thirdly, it offers a reasonable time path for developing legislation and car-insurance policies.

Besides this, in general an important factor of the successful large-scale deployment of ADAS is the *governmental support* of these advances in automated driving, in subsidies, in law making and in standardization or international regulations as well. For cross-border system applicability, it is required that either ADAS can operate standalone completely (as is the approach in most of the work in this thesis), or that communication protocols for collaborative approaches are standardized globally. Effort towards the latter are undertaken via for instance European subsidy projects, but global scalability of V2V communication or HD maps availability remains a large unknown factor in the developments.

Regarding standalone ADAS and specifically our strategy of making them self-adapting is the problem of assessing *reliability of online tuning for ADAS* for real-world applications. Any ADAS would have to undergo a series of tests by a governmental institute prior to commercial deployment. A rightful question then is how to assess a system that can change itself after releasing it on the roads. This involves proving a limited range of adaptability, and severe safety guarantees of the underlying control mechanisms. An interesting line of research would be to look into self-assessment, potentially via the upcoming field of *AI for causal reasoning*, where the system should reset to a known safe state when it detects that its modeling drifts far from trusted reference points. Perhaps an interesting compromise is a *centralized and crowd-sourced updating* scheme, so that updated models can be verified on offline data by the central unit prior to deployment in the car. This would generate a delay in the adaptivity of the modeling, but provides control over systems that are road-operational at a large scale.

A related discussion revolves around *ethical responsibilities and insurance*, to define which party (driver, owner, manufacturer) is responsible in eventual accidents. These are complicated cases for which both legislation, insurance and various consumer groups need to be consulted, from which a feasible introduction strategy should be derived and implemented. In these development trajectories, it has to be considered that ADAS, especially in early-stage mixed traffic, can and will still cause accidents. In that period, the focus should be on managing the trade-off of implementation benefits and safety risks.

In that light, the recent book by Verkade and te Brömmelstroet discusses the several difficulties of designing the public space that has to be shared between a wide variety of users, while it is currently dominated by the default view in favoring the car everywhere, every time [153]. Although ADAS are not the final and only solution to the discussions raised in that work, we foresee a positively contributing role of increasing the situational awareness of vehicles to enhance sharing public space more evenly between relevant parties. Ultimately, the increase of ADAS performance and functionality contributes on numerous aspects of mobility, changing the experience of an individual user as well as impacting its perspective within society as a whole.

Bibliography

[1] *A Field Guide to the Future of Mobility*. white paper at World Economic Forum (WEF) 291215. the Global Agenda Council on the Future of Automotive & Personal Transport, Jan. 2016.

[2] *Transport and Mobility*. Den Haag: Centraal Bureau voor de Statistiek, 2016. ISBN: 978-90-357-2056-5.

[3] M. Cloïn, A. van den Broek, R. van den Dool, de Haan Jos, J. de Hart, P. van Houwelingen, A. Tiessen-Raaphorst, N. Sonck, and J. Spit. *Met het oog op de tijd. Een blik op de tijdsbesteding van Nederlanders*. Den Haag: Sociaal en Cultureel Planbureau, 2013. ISBN: 978 90 377 0670 3.

[4] T. A. Dingus, F. Guo, S. Lee, J. F. Antin, M. Perez, M. Buchanan-King, and J. Hankey. "Driver crash risk factors and prevalence evaluation using naturalistic driving data". In: *Proceedings of the National Academy of Sciences* 113.10 (2016), pp. 2636–2641.

[5] S. Singh. *Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey. (Traffic Safety Facts Crash Stats)*. Tech. rep. DOT HS 812 506. National Highway Traffic Safety Administration (NHTSA), National Center for Statistics and Analysis, Mar. 2018.

[6] S. Shaheen, H. Totte, and A. Stocker. *Future of Mobility*. white paper for California Transportation plan 2050 (CalTrans). Jan. 2018.

[7] *Surface Vehicle Recommended Practice*. Tech. rep. J3016. [Online; accessed June 2019]. SAE International, June 2018.

[8] *Visual chart for the levels of driving automation-standard*. `https://www.sae.org/news/press-room/2018/12/sae-international-releases-updated-visual-chart-for-its-%E2%80%9Clevels-of-driving-automation%E2%80%9D-standard-for-self-driving-vehicles`. news article. [Online; accessed June 2019]. SAE International, Dec. 11, 2018.

[9] E. Frazzoli. *Perception for Action: on NuTonomy's Vision for Autonomous Driving*. invited lecture. Sicily: International Computer Vision Summer School (ICVSS), July 10, 2017.

[10] S. M. Casner, E. L. Hutchins, and D. Norman. "The challenges of partially automated driving". In: *Communications of the ACM* (May 2016).

[11] J. C. De Winter, R. Happee, M. H. Martens, and N. A. Stanton. "Effects of adaptive cruise control and highly automated driving on workload and situation awareness: A review of the empirical evidence". In: *Transportation research part F: traffic psychology and behaviour* 27 (2014), pp. 196–217.

[12] R. E. Llaneras, J. Salinger, and C. A. Green. "Human factors issues associated with limited ability autonomous driving systems: Drivers' allocation of visual attention to the forward roadway". In: *7th International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design*. 2013, pp. 92–98.

[13] R. Urtasun. *Towards Affordable Self Driving Cars*. invited lecture. Sicily: International Computer Vision Summer School (ICVSS), July 14, 2017.

[14] F. Iandola. *Perception systems for autonomous vehicles using energy-efficient deep neural networks*. keynote talk. Burlingame, USA: IS&T Electronic Imaging - Autonomous Vehicles and Machines, Jan. 16, 2019.

[15] K. Spieser, K. Treleaven, R. Zhang, E. Frazzoli, D. Morton, and M. Pavone. "Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in Singapore". In: *Road vehicle automation*. Springer, 2014, pp. 229–245.

[16] V. Distler, C. Lallemand, and T. Bellet. "Acceptability and acceptance of autonomous mobility on demand: The impact of an immersive experience". In: *Proceedings of the 2018 CHI conference on human factors in computing systems*. 2018, pp. 1–10.

[17] A. J. Niranjan and G. de Haan. "Public Opinion About Self-Driving Vehicles in the Netherlands". In: *Proceedings of the 36th European Conference on Cognitive Ergonomics*. 2018, pp. 1–4.

[18] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, et al. "Making Bertha drive - An autonomous journey on a historic route". In: *IEEE Intelligent Transportation Systems Magazine (ITSM)* 6.2 (2014), pp. 8–20.

[19] P. Bannon, A. Karpathy, S. Bowers, and E. Musk. *Tesla Autonomy Day*. `https://www.youtube.com/watch?v=Ucp0TTmvqOE`. Apr. 22, 2019.

[20] B. Fowler, J. Pei, C. Schroeder, and A. Shashua. *Sensing and Perceiving for Autonomous Driving*. panel discussion. Burlingame, USA: IS&T Electronic Imaging - Autonomous Vehicles and Machines, Jan. 14, 2019.

[21] M. Aeberhard, S. Rauch, M. Bahram, G. Tanzmeister, J. Thomas, Y. Pilat, F. Homm, W. Huber, and N. Kaempchen. "Experience, results and lessons learned from automated driving on Germany's highways". In: *IEEE Intelligent transportation systems magazine* 7.1 (2015), pp. 42–57.

[22] J. Elfring, R. Appeldoorn, S. Van den Dries, and M. Kwakkernaat. "Effective world modeling: Multisensor data fusion methodology for automated driving". In: *Sensors* 16.10 (2016), p. 1668.

[23] L. Eldada. *Solid-state LiDAR sensors: The future of autonomous vehicles*. keynote talk. Burlingame, USA: IS&T Electronic Imaging - Autonomous Vehicles and Machines, Jan. 16, 2019.

[24] J. L. Lee. "Intel's Mobileye demos autonomous car equipped only with cameras, no other sensors". In: (2020). [online; accessed June 2020].

[25] M. Nieto, O. Otaegui, G. Vélez, J. D. Ortega, and A. Cortés. "On creating vision-based advanced driver assistance systems". In: *IET intelligent transport systems* 9.1 (2014), pp. 59–66.

[26] M. Nieto, G. Vélez, O. Otaegui, S. Gaines, and G. Van Cutsem. "Optimising computer vision based ADAS: vehicle detection case study". In: *IET Intelligent Transport Systems* 10.3 (2016), pp. 157–164.

[27] Y. Zhang, Z. Qiu, J. Liu, T. Yao, D. Liu, and T. Mei. "Customizable architecture search for semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11641–11650.

[28] A. Shaw, D. Hunter, F. Iandola, and S. Sidhu. "Squeezenas: Fast neural architecture search for faster semantic segmentation". In: *Proceedings of the IEEE international conference on computer vision workshops*. 2019.

[29] H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han. "Once for All: Train One Network and Specialize it for Efficient Deployment". In: *International Conference on Learning Representations (ICLR)*. 2020.

[30] J. Xu, A. G. Schwing, and R. Urtasun. "Learning to segment under various forms of weak supervision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3781–3790.

[31] V. Koltun. *Learning to Act with Natural Supervision*. invited lecture. Sicily: International Computer Vision Summer School (ICVSS), July 10, 2017.

[32] M. Waibel, M. Beetz, J. Civera, R. d'Andrea, J. Elfring, D. Galvez-Lopez, K. Häussermann, R. Janssen, J. Montiel, A. Perzylo, et al. "Roboearth". In: *IEEE Robotics & Automation Magazine* 18.2 (2011), pp. 69–82.

[33] W. P. Sanberg, G. Dubbelman, and P. H. N. de With. "Extending the stixel world with online self-supervised color modeling for road-versus-obstacle segmentation". In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2014, pp. 1400–1407.

[34] W. P. Sanberg, G. Dubbelman, and P. H. N. de With. "Color-based free-space segmentation using online disparity-supervised learning". In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE. 2015, pp. 906–912.

[35] W. P. Sanberg, G. Dubbelman, and P. H. N. de With. "Free-Space Detection using Online Disparity-supervised Color Modeling". In: *7th IROS Workshop on Planning, Perception and Navigation for Intelligent Vehicles (IROS-PPNIV)*. Sept. 2015, pp. 105–110.

[36] W. P. Sanberg, G. Dubbelman, and P. H. N. de With. " Free-Space Segmentation based on Online Disparity-supervised Color Modeling". In: *the Netherlands Conference on Computer Vision (NCCV)*. Sept. 2015, (abstract).

[37] W. P. Sanberg, G. Dubbelman, and P. H. N. de With. "FCNs for Free-Space Detection with Self-Supervised Online Training". In: *IEEE IV - DeepDriving: Learning Representations for Intelligent Vehicles Workshop (IV-DD)*. June 2016, (extended abstract).

[38] W. P. Sanberg, G. Dubbelman, and P. H. N. de With. "Self-Supervised Online Training of FCNs for Free-Space Detection". In: *the Netherlands Conference on Computer Vision (NCCV)*. Dec. 2016, (abstract).

[39] W. P. Sanberg, G. Dubbelman, and P. H. N. de With. "Free-Space Detection with Self-Supervised and Online Trained Fully Convolutional Networks". In: *IS&T Electronic Imaging - Autonomous Vehicles and Machines (EI-AVM)*. Jan. 2017, pp. 54–61.

[40] W. P. Sanberg, G. Dubbelman, and P. H. N. de With. "From Stixels to Asteroids: Towards a Collision Warning System using Stereo Vision". In: *IS&T Electronic Imaging - Autonomous Vehicles and Machines (EI-AVM)*. Jan. 2019.

[41] W. P. Sanberg, G. Dubbelman, and P. H. N. de With. "ASTEROIDS: A Stixel Tracking Extrapolation-based Relevant Obstacle Impact Detection System". In: *IEEE Trans. on Intelligent Vehicles (T-IV)* (2020), (IN PRESS).

[42] D. W. J. M. van de Wouw, W. P. Sanberg, G. Dubbelman, and P. H. N. de With. "Fast 3D Scene Alignment with Stereo Images using a Stixel-based 3D Model". In: *Int. Conf. on Computer Vision Theory and Applications (VISAPP)*. Jan. 2018, pp. 250–259. ISBN: 978-989-758-290-5.

[43] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez. "Road scene segmentation from a single image". In: *European Conference on Computer Vision*. Springer. 2012, pp. 376–389.

[44] J. M. Alvarez, M. Salzmann, and N. Barnes. "Learning appearance models for road detection". In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2013, pp. 423–429.

[45] A. M. Neto, A. C. Victorino, I. Fantoni, and J. V. Ferreira. "Real-time estimation of drivable image area based on monocular vision". In: *2013 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. IEEE. 2013, pp. 63–68.

[46] G. Dubbelman, W. van der Mark, J. C. van den Heuvel, and F. C. Groen. "Obstacle detection during day and night conditions using stereo vision". In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2007, pp. 109–116.

[47] R. Labayrade, D. Aubert, and J.-P. Tarel. "Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation". In: *Intelligent Vehicle Symposium, 2002. IEEE*. Vol. 2. IEEE. 2002, pp. 646–651.

[48] C. G. Keller, M. Enzweiler, M. Rohrbach, D. F. Llorca, C. Schnorr, and D. M. Gavrila. "The benefits of dense stereo for pedestrian detection". In: *IEEE transactions on intelligent transportation systems* 12.4 (2011), pp. 1096–1106.

[49] T. Scharwächter, M. Enzweiler, U. Franke, and S. Roth. "Efficient multi-cue scene segmentation". In: *German Conference on Pattern Recognition*. Springer. 2013, pp. 435–445.

[50] M. L. L. Rompen, W. P. Sanberg, G. Dubbelman, and P. H. N. de With. "Online self-supervised learning for road detection". In: *WIC/IEEE Symp. on Information Theory and Signal Processing in the Benelux (SITB)*. 2014, pp. 148–155.

[51] D. Pfeiffer. *The stixel world*. Humboldt-Universität zu Berlin, Mathematisch - Naturwissenschaftliche Fakultät II, 2012.

[52] M. Bajracharya, A. Howard, L. H. Matthies, B. Tang, and M. Turmon. "Autonomous off-road navigation with end-to-end learning for the LAGR program". In: *Journal of Field Robotics* 26.1 (2009), pp. 3–25.

[53] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al. "Stanley: The robot that won the DARPA Grand Challenge". In: *Journal of field Robotics* 23.9 (2006), pp. 661–692.

[54] P. Heckbert. *Color image quantization for frame buffer display*. Vol. 16. 3. ACM, 1982.

[55] J. Fritsch, T. Kuehnl, and A. Geiger. "A new performance measure and evaluation benchmark for road detection algorithms". In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE. 2013, pp. 1693–1700.

[56] H. Hirschmuller. "Stereo processing by semiglobal matching and mutual information". In: *IEEE Transactions on pattern analysis and machine intelligence* 30.2 (2008), pp. 328–341.

[57] D. Pfeiffer, S. Gehrig, and N. Schneider. "Exploiting the power of stereo confidences". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 297–304.

[58] A. Geiger, P. Lenz, and R. Urtasun. "Are we ready for autonomous driving? The KITTI vision benchmark suite". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 3354–3361.

[59] W. Van Der Mark and D. M. Gavrila. "Real-time dense stereo for intelligent vehicles". In: *IEEE Transactions on intelligent transportation systems* 7.1 (2006), pp. 38–50.

[60] S. K. Gehrig, F. Eberli, and T. Meyer. "A real-time low-power stereo vision engine using semi-global matching". In: *International Conference on Computer Vision Systems*. Springer. 2009, pp. 134–143.

[61] R. C. Gonzales and R. E. Woods. *Digital Image Processing* (third edition). Pearson Prentice Hall, 2008. ISBN: 0-13-505267-X.

[62] J. M. Alvarez and A. M. Lopez. "Road detection based on illuminant invariance". In: *IEEE Transactions on Intelligent Transportation Systems* 12.1 (2011), pp. 184–193.

[63] H. G. J. Groot. *Computational Performance Study of Color-based Free-Space Segmentation*. Tech. rep. 1265. Eindhoven University of Technology, dep. of Electrical Engineering, SPS-VCA group, 2015.

[64] J. Zbontar and Y. LeCun. "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches." In: *Journal of Machine Learning Research* 17.1-32 (2016), p. 2.

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

[66] R. Mohan. "Deep deconvolutional networks for scene parsing". In: *arXiv:1411.4101* (2014).

[67] C.-A. Brust, S. Sickert, M. Simon, E. Rodner, and J. Denzler. "Convolutional patch networks with spatial prior for road detection and urban scene understanding". In: *Int. Conf. on Computer Vision Theory and Applications (VISAPP)*. 2015.

[68] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems (NIPS)*. 2012, pp. 1097–1105.

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

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

[71] J. Long, E. Shelhamer, and T. Darrell. "Fully convolutional networks for semantic segmentation". In: *IEEE conf. on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3431–3440.

[72] P. Voigtlaender and B. Leibe. "Online adaptation of convolutional neural networks for video object segmentation". In: *arXiv preprint arXiv:1706.09364* (2017).

[73] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. "Discriminative unsupervised feature learning with convolutional neural networks". In: *Advances in neural information processing systems*. 2014, pp. 766–774.

[74] P. O. Pinheiro and R. Collobert. "From image-level to pixel-level labeling with convolutional networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1713–1721.

[75] G. E. Hinton, S. Osindero, and Y.-W. Teh. "A fast learning algorithm for deep belief nets". In: *Neural computation* 18.7 (2006), pp. 1527–1554.

[76] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. "Why does unsupervised pre-training help deep learning?" In: *Journal of Machine Learning Research* 11.Feb (2010), pp. 625–660.

[77] M. D. Zeiler, G. W. Taylor, R. Fergus, et al. "Adaptive deconvolutional networks for mid and high level feature learning." In: *ICCV*. Vol. 1. 2. 2011, p. 6.

[78] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. "Decaf: A deep convolutional activation feature for generic visual recognition". In: *International conference on machine learning*. 2014, pp. 647–655.

[79] M. D. Zeiler and R. Fergus. "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. Springer. 2014, pp. 818–833.

[80] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. "Return of the devil in the details: Delving deep into convolutional nets". In: *arXiv preprint arXiv:1405.3531* (2014).

[81] S. Branson, G. Van Horn, S. Belongie, and P. Perona. "Bird species categorization using pose normalized deep convolutional nets". In: *arXiv preprint arXiv:1406.2952* (2014).

[82] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller, and Y. LeCun. "Learning long-range vision for autonomous off-road driving". In: *Journal of Field Robotics* 26.2 (2009), pp. 120–144.

[83] D. Hoiem, A. A. Efros, and M. Hebert. "Recovering surface layout from an image". In: *International Journal of Computer Vision* 75.1 (2007), pp. 151–172.

[84] C.-A. Brust, S. Sickert, M. Simon, E. Rodner, and J. Denzler. "Efficient convolutional patch networks for scene understanding". In: *CVPR Scene Understanding Workshop*. 2015.

[85] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[86] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

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

[88] L.-C. Liu, C.-Y. Fang, and S.-W. Chen. "A Novel distance estimation method leading a forward collision avoidance assist system for vehicles on highways". In: *IEEE Transactions on Intelligent Transportation Systems* 18.4 (2017), pp. 937–949.

[89] S. Cherng, C.-Y. Fang, C.-P. Chen, and S.-W. Chen. "Critical motion detection of nearby moving vehicles in a vision-based driver-assistance system". In: *IEEE Transactions on Intelligent Transportation Systems* 10.1 (2009), pp. 70–82.

[90] A. Ess, K. Schindler, B. Leibe, and L. Van Gool. "Object detection and tracking for autonomous navigation in dynamic environments". In: *The International Journal of Robotics Research* 29.14 (2010), pp. 1707–1725.

[91] MobilEye. *User Manual - Series 6 (DOC000600 REV A02 - ENG)*. [Online; accessed October 2018].

[92] D. Lee and H. Yeo. "Real-time rear-end collision-warning system using a multilayer perceptron neural network". In: *IEEE Trans. on Intelligent Transportation Systems* 17-11 (2016), pp. 3087–3097.

[93] X. Xiong, L. Chen, and J. Liang. "A new framework of vehicle collision prediction by combining SVM and HMM". In: *IEEE Trans. on Intelligent Transportation Systems* 19-3 (2018), pp. 699–710.

[94] C. Olariu, H. Assem, J. D. Ortega, and M. Nieto. "A Cloud-Based AI Framework for Machine Learning Orchestration: A "Driving or Not-Driving" Case-Study for Self-Driving Cars". In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2019, pp. 1715–1722.

[95] H. Badino, U. Franke, and D. Pfeiffer. "The stixel world-a compact medium level representation of the 3d-world". In: *Joint Pattern Recognition Symposium*. Springer. 2009, pp. 51–60.

[96] D. Pfeiffer and U. Franke. "Towards a Global Optimal Multi-Layer Stixel Representation of Dense 3D Data." In: *BMVC*. 2011, pp. 1–12.

[97] L. Schneider, M. Cordts, T. Rehfeld, D. Pfeiffer, M. Enzweiler, U. Franke, M. Pollefeys, and S. Roth. "Semantic stixels: Depth is not enough". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2016, pp. 110–117.

[98] M. Enzweiler, M. Hummel, D. Pfeiffer, and U. Franke. "Efficient stixel-based object recognition". In: *2012 IEEE Intelligent Vehicles Symposium*. IEEE. 2012, pp. 1066–1071.

[99] J. F. P. K. T. Hehn and D. M. Gavrila. "Instance Stixels: Segmenting and Grouping Stixels into Objects". In: *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*. June 2019.

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

[101] A. Ošep, W. Mehner, P. Voigtlaender, and B. Leibe. “Track, then decide: Category-agnostic vision-based multi-object tracking”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1–8.

[102] R. Schubert, E. Richter, and G. Wanielik. “Comparison and evaluation of advanced motion models for vehicle tracking”. In: *11th Int. Conf. on Information Fusion*. IEEE. 2008.

[103] M. P. Muresan and S. Nedevschi. “Multimodal sparse LIDAR object tracking in clutter”. In: *IEEE 14th Int. Conf. on Intelligent Computer Communication and Processing (ICCP)*. Sept. 2018, pp. 215–221.

[104] C. G. Keller and D. M. Gavrila. “Will the pedestrian cross? A study on pedestrian path prediction”. In: *IEEE Transactions on Intelligent Transportation Systems* 15.2 (2014), pp. 494–506.

[105] J. Elfring, R. Van De Molengraft, and M. Steinbuch. “Learning intentions for improved human motion prediction”. In: *Robotics and Autonomous Systems* 62.4 (2014), pp. 591–602.

[106] N. Wojke, A. Bewley, and D. Paulus. “Simple online and realtime tracking with a deep association metric”. In: *IEEE Int. Conf. on Image Processing (ICIP)*. IEEE. 2017, pp. 3645–3649.

[107] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. “Dynamic 3d scene analysis from a moving vehicle”. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2007, pp. 1–8.

[108] R. Danescu, F. Oniga, and S. Nedevschi. “Modeling and tracking the driving environment with a particle-based occupancy grid”. In: *IEEE Trans. on Intelligent Transportation Systems* 12-4 (2011), pp. 1331–1342.

[109] C. Laugier, I. E. Paromtchik, M. Perrollaz, M. Yong, J.-D. Yoder, C. Tay, K. Mekhnacha, and A. Nègre. “Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety”. In: *IEEE Intelligent Transportation Systems Mag.* 3-4 (2011), pp. 4–19.

[110] D. Kochanov, A. Ošep, J. Stückler, and B. Leibe. “Scene flow propagation for semantic mapping and object discovery in dynamic street scenes”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2016, pp. 1785–1792.

[111] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press Cambridge, 2005.

[112] R. D. Yates and D. J. Goodman. *Probability and stochastic processes: a friendly introduction for electrical and computer engineers (2nd edition)*. John Wiley & Sons, 2005.

[113] H. H. Ku et al. “Notes on the use of propagation of error formulas”. In: *Journal of Research of the National Bureau of Standards* 70.4 (1966).

[114] S. Ramos, S. Gehrig, P. Pinggera, U. Franke, and C. Rother. “Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling”. In: *IEEE Intelligent Vehicles Symp. (IV)*. IEEE. 2017, pp. 1025–1032.

[115] B. Groenen. *Stixel motion prediction for collision warning*. Tech. rep. 1321. Eindhoven University of Technology, dep. of Electrical Engineering, SPS-VCA group, 2018.

[116] H. Rohling. “Radar CFAR thresholding in clutter and multiple target situations”. In: *IEEE Trans. on Aerospace and Electronic Systems* 4 (1983), pp. 608–621.

[117] B. R. Mahafza. “Radar signal analysis and processing using MATLAB”. In: CRC Press/Taylor & Francis Group, 2010. Chap. 7.10, pp. 293–295. ISBN: 978-1-4200-6644-9.

[118] C. G. Keller, T. Dang, H. Fritz, A. Joos, C. Rabe, and D. M. Gavrila. “Active pedestrian safety by automatic braking and evasive steering”. In: *IEEE Transactions on Intelligent Transportation Systems* 12.4 (2011), pp. 1292–1304.

[119] M. Tideman. “Scenario-Based Simulation Environment for Assistance Systems”. In: *ATZautotechnology* 10-1 (Jan. 2010), pp. 28–32. ISSN: 2192-886X.

[120] *Vehicle Stopping Distance and Time*. `https://nacto.org/docs/usdg/vehicle_stopping_distance_and_time_upenn.pdf`. [Online; accessed May 2019].

[121] *Safety1nNum3ers: Speeding*. `https://one.nhtsa.gov/nhtsa/Safety1nNum3ers/august2015/S1N_Speeding-August2015_812008.pdf`. [Online; accessed May 2019]. Aug. 2015.

[122] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. “A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4040–4048.

[123] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. "Flownet 2.0: Evolution of optical flow estimation with deep networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2462–2470.

[124] D. W. J. M. van de Wouw, F. B. ter Haar, G. Dubbelman, and P. H. N. de With. "Development and analysis of a real-time system for automated detection of improvised explosive device indicators from ground vehicles". In: *Journal of Electronic Imaging* 28.4 (2019), pp. 1–16.

[125] G. Zhang, Y. He, W. Chen, J. Jia, and H. Bao. "Multi-viewpoint panorama construction with wide-baseline images". In: *IEEE Transactions on Image Processing* 25.7 (2016), pp. 3099–3111.

[126] H.-R. Su and S.-H. Lai. "Non-rigid registration of images with geometric and photometric deformation by using local affine Fourier-moment matching". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 2874–2882.

[127] Z. Lou and T. Gevers. "Image alignment by piecewise planar region matching". In: *IEEE Transactions on Multimedia* 16.7 (2014), pp. 2052–2061.

[128] D. W. J. M. van de Wouw, G. Dubbelman, and P. H. N. de With. "Hierarchical 2.5-D Scene Alignment for Change Detection With Large Viewpoint Differences". In: *IEEE Robotics and Automation Letters* (2016), pp. 361–368.

[129] L. Vosters, C. Varekamp, and G. de Haan. "Overview of efficient high-quality state-of-the-art depth enhancement methods by thorough design space exploration". In: *Journal of Real-Time Image Processing* 16.2 (2019), pp. 355–375.

[130] A.-L. Chauve, P. Labatut, and J.-P. Pons. "Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pp. 1261–1268.

[131] A. Maiti and D. Chakravarty. "Performance analysis of different surface reconstruction algorithms for 3D reconstruction of outdoor objects from their digital images". In: *SpringerPlus* 5.1 (2016), p. 932.

[132] D. Stutz, A. Hermans, and B. Leibe. "Superpixels: An evaluation of the state-of-the-art". In: *Computer Vision and Image Understanding* 166 (2018), pp. 1–27.

[133] P. F. Felzenszwalb and D. P. Huttenlocher. "Efficient graph-based image segmentation". In: *International journal of computer vision* 59.2 (2004), pp. 167–181.

[134] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. "SLIC superpixels compared to state-of-the-art superpixel methods". In: *IEEE transactions on pattern analysis and machine intelligence* 34.11 (2012), pp. 2274–2282.

[135] O. Veksler, Y. Boykov, and P. Mehrani. "Superpixels and supervoxels in an energy optimization framework". In: *European conference on Computer vision*. Springer. 2010, pp. 211–224.

[136] J. Strom, A. Richardson, and E. Olson. "Graph-based segmentation for colored 3D laser point clouds". In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2010, pp. 2131–2136.

[137] W. P. Sanberg, L. Do, et al. "Flexible multi-modal graph-based segmentation". In: *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer. 2013, pp. 492–503.

[138] M. Cordts, T. Rehfeld, M. Enzweiler, U. Franke, and S. Roth. "Tree-structured models for efficient multi-cue scene labeling". In: *IEEE transactions on pattern analysis and machine intelligence* 39.7 (2016), pp. 1444–1454.

[139] K. Yamaguchi, D. McAllester, and R. Urtasun. "Efficient joint segmentation, occlusion labeling, stereo and flow estimation". In: *European Conference on Computer Vision*. Springer. 2014, pp. 756–771.

[140] A. Broggi, S. Cattani, M. Patander, M. Sabbatelli, and P. Zani. "A full-3D voxel-based dynamic obstacle detection for urban scenario using stereo vision". In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE. 2013, pp. 71–76.

[141] D. W. J. M. A. van de Wouw, M. A. R. Pieck, G. Dubbelman, and P. H. N. de With. "Real-time estimation of the 3D transformation between images with large viewpoint differences in cluttered environments". In: *Electronic Imaging* 2017.13 (2017), pp. 109–116.

[142] T. Bellet and G. Pelzer. *VI-DAS: Use cases definition using Human-Centred Design methodology*. public deliverable D1.3. Version third version. VI-DAS consortium, Sept. 2018.

[143] J. Eggert, D. Salazar, T. Puphal, and B. Flade. “Driving situation analysis with relational local dynamic maps (R-LDM)”. In: *Proc. Symp. Future Active Safety Technology*. 2017.

[144] G. Pelzer. *VIDAS: Requirements, specifications and reference architecture*. public deliverable D1.6. VI-DAS consortium, Sept. 2018.

[145] R. Imbriaco. *From Stixel-Based Superpixels to 3D Scene-Modeling*. Tech. rep. 1275. internship report. Eindhoven University of Technology, dep. of Electrical Engineering, SPS-VCA group, 2016.

[146] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. “A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD’96. Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[147] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. “The cityscapes dataset for semantic urban scene understanding”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3213–3223.

[148] J. Sloot. *Semantic Traffic Scene SegmentationUsing Prior Class Information*. Tech. rep. BSc. graduation thesis. Eindhoven University of Technology, dep. of Electrical Engineering, SPS-VCA group, 2017.

[149] P. Meletis and G. Dubbelman. “Training of Convolutional Networks on Multiple Heterogeneous Datasets for Street Scene Semantic Segmentation”. In: *IEEE Intelligent Vehicles Symposium (IV)*. IEEE. June 2018.

[150] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder. “The mapillary vistas dataset for semantic understanding of street scenes”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4990–4999.

[151] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size”. In: *arXiv preprint arXiv:1602.07360* (2016).

[152] M. Tan and Q. V. Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *arXiv preprint arXiv:1905.11946* (2019).

[153] T. Verkade and M. te Brömmelstroet. *Het recht van de snelste - Hoe ons verkeer steeds associaler werd*. De Correspondent Uitgevers, 2020. ISBN: 9789083000718.

Publication List

The following conference and journal papers have been (co-)published by the author of this thesis.

Journal articles

[J1] **W. P. Sanberg**, G. Dubbelman, and P. H. N. de With. “ASTEROIDS: A Stixel Tracking Extrapolation-based Relevant Object Impact Detection System”. In: *IEEE Trans. on Intelligent Vehicles* (2020), [IN PRESS].

[J2] N. Kemsaram, **W. P. Sanberg**, A. Das, G. Dubbelman, and P. H. N. de With. “Design and Development of a Stixel-based Stereo Vision System for Cooperative Automated Vehicles”. In: *Journal of Automotive Software Engineering* (2020), [IN PREPARATION].

International conference contributions

[IC9] **W. P. Sanberg**, G. Dubbelman, and P. H. N. de With. “From Stixels to Asteroids: Towards a Collision Warning System using Stereo Vision”. In: *IS&T Electronic Imaging - Autonomous Vehicles and Machines (EI-AVM)*. Jan. 2019, ***Best Paper Award***.

[IC8] D. W. J. M. van de Wouw, **W. P. Sanberg**, G. Dubbelman, and P. H. N. de With. “Fast 3D Scene Alignment with Stereo Images using a Stixel-based 3D Model”. In: *Int. Conf. on Computer Vision Theory and Applications (VISAPP)*. Jan. 2018, pp. 250–259. ISBN: 978-989-758-290-5.

[IC7] **W. P. Sanberg**, G. Dubbelman, and P. H. N. de With. “Free-Space Detection with Self-Supervised and Online Trained Fully Convolutional Networks”. In: *IS&T Electronic Imaging - Autonomous Vehicles and Machines (EI-AVM)*. Jan. 2017, pp. 54–61.

[IC6] **W. P. Sanberg**, G. Dubbelman, and P. H. N. de With. “FCNs for Free-Space Detection with Self-Supervised Online Training”. In: *IEEE IV - DeepDriving: Learning Representations for Intelligent Vehicles Workshop (IV-DD)*. June 2016, (extended abstract).

[IC5] **W. P. Sanberg**, G. Dubbelman, and P. H. N. de With. “Free-Space Detection using Online Disparity-supervised Color Modeling”. In: *7th IROS Workshop on Planning, Perception and Navigation for Intelligent Vehicles (IROS-PPNIV)*. Sept. 2015, pp. 105–110.

[IC4] **W. P. Sanberg**, G. Dubbelman, and P. H. N. de With. “Color-based Free-Space Segmentation using Online Disparity-supervised Learning”. In: *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*. Sept. 2015, pp. 906–912.

[IC3] **W. P. Sanberg**, G. Dubbelman, and P. H. N. de With. “Extending the Stixel World with Online Self-Supervised Color Modeling for Road-Versus-Obstacle Segmentation”. In: *IEEE Conference on Intelligent Transportation Systems (ITSC)*. Oct. 2014, pp. 1400–1407.

[IC2] M. L. Rompen, **W. P. Sanberg**, G. Dubbelman, and P. H. N. de With. “Online Self-supervised Learning for Road Detection”. In: *WIC/IEEE SP Symposium on Information Theory and Signal Processing in the Benelux*. May 2014, pp. 148–155.

[IC1] **W. P. Sanberg**, L. Q. Do, and P. H. N. de With. “Flexible multi-modal graph-based segmentation”. In: *Advanced Concepts for Intelligent Vision Systems (ACIVS) Vol. 8192. Lecture Notes in Computer Science*. Oct. 2013, pp. 492–503.

Regional conference contributions

[RC4] **W. P. Sanberg**, G. Dubbelman, and P. H. N. de With. “Self-Supervised Online Training of FCNs for Free-Space Detection”. In: *the Netherlands Conference on Computer Vision (NCCV)*. Dec. 2016, (abstract).

[RC3] **W. P. Sanberg**, G. Dubbelman, and P. H. N. de With. “Online Self-Supervised End-to-End Learning of Fully Convolutional Networks for Free-Space Detection”. In: *IEEE sb-E Western European Student and Young Professional congress (WESYP)*. June 2016, (abstract).

[RC2] **W. P. Sanberg**, G. Dubbelman, and P. H. N. de With. “ Free-Space Segmentation based on Online Disparity-supervised Color Modeling”. In: *the Netherlands Conference on Computer Vision (NCCV)*. Sept. 2015, (abstract).

[RC1] **W. P. Sanberg**, G. Dubbelman, and P. H. N. de With. “Color-based Free-Space Segmentation using Online Disparity-supervised Learning”. In: *IEEE sb-E Western European Student and Young Professional congress (WESYP)*. May 2015, (abstract).

Acronyms

ADAS Advanced Driver Assistance Systems

AI Artificial Intelligence

aoi angle of impact

ASTEROIDS A Stixel Tracking Extrapolation-based Relevant Obstacle Impact Detection System

BEV Birds Eye View

(C)ACC (Cooperative) Adaptive Cruise Control

CAN Controller Area Network

CFAR Constant False-Alarm Rate

CNN Convolutional Neural Network

CPU Central Processing Unit

FCW Forward-Collision Warning

DCM Driver Monitoring System

FPGA Field-Programmable Gate Array

FPR False Positive Rate

FCPN Fully Convolutional Patch Network

fps frames per second

GPS Global Positioning System

GPU Graphics Processing Unit

GUI Graphical User Interface

HD-map High-Definition map

HEQ Histogram Equalization

HMI Human-Machine Interface

HS(I) Hue, Saturation(, Intensity)

IED Improvised Explosive Device

IMU Inertial Measurement Unit

LDM Local Dynamic Map

LDW Lane-Departure Warning

Lidar Light detection and ranging

LKA Lane-Keep Assist

LW Learning Window

MaaS Mobility as a Service

MAP Maximum A-Posteriori

(HA-)NAS (Hardware-Aware) Neural Architecture Search

NN Neural Network

Radar Radio detection and ranging

RGB Red, Green, Blue

ROC Receiver Operator Characteristic

ROI Region Of Interest

RTK-GPS Real-Time Kinematic Global Positioning System

SAE Society of Automotive Engineers

SGBM Semi-Global Block Matching

soi side of impact

SSCOD Semantic Stixel Clustering for Object Detection

SVD Singular Value Decomposition

TM Training Mask

TSR Traffic-Sign Recognition

ttc time to collision

UHD Ultra-High Definition

V2I Vehicle-to-Infrastructure (-communication)

V2V Vehicle-to-Vehicle (-communication)

V2X Vehicle-to-Something (-communication)

VI-DAS Vision-Inspired Driver Assistance Systems

VRU Vulnerable Road User

Acknowledgements

Pursuing a PhD degree has been a key storyline in my life for several years and, honestly, it feels slightly unsettling now that I am actually finalizing it. My PhD was surely no walk in the park, but building personal experiences is as important as the work itself and I was positively amazed by the broad and invigorating possibilities of doing research at university. It is a complex but brilliant job, and if it somehow matches your personality - go for it.

Naturally, much of the experience originates from the environment in which you operate: the events, the opportunities, and most importantly, the people around you, both in- and outside work.

Peter, the energy, effort and time you make available for your PhD candidates is amazing. You are always busy, while you are also always available if a problem or question arises. I highly value your keen eye for both details and high-level context. You consider the complete story, whether it is a technical discussion, a strategic vision or in conversations larger than work. I happily took one of your early lessons a bit far ("*Schedules are a means, not an end.*"), and I am grateful that you showed me to not let potential weaknesses of a project steal all its thunder and always highlight the positive sides as well.

Gijs! I worked towards my PhD under your daily supervision, which was very easy, for several reasons. Firstly, you occupied the desk right in front of me. Secondly, we right away had many interesting, open and enthusiastic discussions on research strategies and opportunities. You successfully grew your research group and activities, so that I had to share your attention and help out with secondary tasks - but in return, my work became broader and more dynamic, which I highly value. As an ultimate example of this, you strategically introduced me to Gerardo at NXP, where I now have an interesting job that is strongly aimed at bringing TU/e and NXP closer together on your research topics - smart move, and much appreciated!

Finalizing this process officially is made possible by my committee members, and I am honored with the involvement and feedback of prof. Leibe, prof. Gavrila and dr. Elfring as experts from both academia and the industry. Prof. de Haan, your critical questions stood out on both my internship and my MSc. graduation project, so I am excited that you are present here at my PhD defense as well. Dr. Nieto, Marcos, I highly enjoyed our collaboration for VI-DAS and the meetings in San Sebastian, I hope we can somehow toast on this in person soon.

The VI-DAS project is an easy bridge to close colleague Jos. You were hired after I could/should have been finished, but I am glad that I stayed around. You did not speed up my personal process, but mostly with interestingly distracting stuff. And some beers. And late-night discussions. And a fire alarm. Thanks.

The other main project in which I was active is Change Detection 2.0, for the Dutch Minitry of Defence. I closely worked with Dennis, who had the enjoyable task of integrating my non-ViNotion-style code into the prototype framework. It was an interesting project, thank you Dennis for your support and inspirational collaboration! This also holds for Egbert, the director of ViNotion, who gave me a role in the project and facilitated my contributions by letting Eric help me with my CUDA work, thank you both. Additionally, I've had the honor of supervising some students along the way: Marco, Herman, Rafaelle, Joep, Ralph and Bas. Some of you became colleagues later, and several of your ideas contributed to the work in this thesis, thank you very much for your efforts.

At VCA, many friendly colleagues contributed to the good atmosphere each in their own way throughout the years, via all kinds of talks, karting events, QuizNightXLs, joint conference visits, VCAThinkTanks, lunch and coffee breaks and much more. Thanks for instance to Debby, Egor, Hani, Ivo, Joost, Kostas, Lykele, Matthijs, Merijn, Onno, Patrick, Ronald and Sveta. Of course there also was secretary Anja with her empathic ear and willingness to help out, which I appreciate very much. Similarly, the MPS cluster grew rapidly, with enthusiastic new researchers Panos, Arash, Chenyang, Ariyan, Fabrizio, and Liang. Moreover, I still collaborate with MPS via Anweshan, Narsimlu and Daan, and it is nice to have that connection. And of course I won't easily forget Floris, my much-appreciated neighbor the last year at university (Hekkie? Bucketing. When in doubt - just debayer again). I'm glad that we still meet at Monk now and then. It's good to have been part of these groups and I wish you all the best for your own journeys.

And then of course there is another wonderful guy there - dear Fons. We made it through our studies together and then both happily found our places in the VCA group. Even while operating in different domains, we were never short of discussion material on research, university politics, life, the universe, and everything. I am proud to have witnessed what you are achieving and am honored to have you with me as friend and paranymph. My other paranymph is Robert, meneer de president. From doing numerous committees together through just being friends, you've been with me in all kinds of ups and downs - and not only while riding a bicycle. Thanks for that, now and in times to come.

Unfortunately, there were only two formal spots to fill in this ceremonial event, but know that there are several others that I would have gladly positioned there. Jos, Mattie, Heepie: I wrote you guys all ridderstukjes, so I will keep it short here. You've been there good times and bad, and I know you'll have my back when I marry. Or if I need a drink (whatever comes first).

Similarly with so many other Drago people... the red thread has been having a beer, which more than often led to interesting conversation. Or skiing trips. Or Oktoberfests. Or bachelor parties without the bachelor. Or... well, you can read that elsewhere. The group is too big to name you all, but let me take a risk and highlight Blowie, BoB, Daniel, Eddy, Eric, Gerard, Henry, Marc, Michael, PP, van K, Yves, and of course brave new PhDers Fab, Joost and Yorick (I'm glad my stories did not stop you). I'm looking forward to the upcoming trips and events, small or big, scheduled or not, near-future or later, to catch up on the sense and nonsense of life.

Because life is a wonderful and intricate endeavor. Dear Sanne, we shared a significant portion of it together, and that contained it all: brilliant adventures, tragic loss, happy trips, emotional moments and discussions on science with quarrels about the concept of infinity. I am proud to have been with you through it all. Dear Sanderijn, knowing you, I doubt you expected to be here, but our time together changed my view on life in more ways than you are probably aware of. Thank you for being so bright, sweet, energetic and open. Dear Carine, I still cherish the story of how we met, which was way more unrealistic than many romcom movie scripts combined. Just my luck to bump into that one French girl who is not into food and wine, and instead a driven and dedicated professor (flying hot air balloons as a bonus). Thank you for the shared adventure and the unfiltered non-Dutch view on the world that came with it.

On a less sentimental note, a big thanks to my other friends that were not yet in the above, but who complete my wine crew, my group of fellow ex board members, the people that gave me the excuse to finally dive into sailing, my quarantine bubble - you know to which group(s) you belong: Rick, Midas, Vogel, Doreth, Jesse, Ingmar, Roel, Marcel. Thanks for luring me into other things than work, I often forget how important that is. Oh, and my almost co-enterpreneur René - glad we considered it, glad we didn't go for it. Paulus, dude, studying, URE, random drinks and dinners with deep discussions: wouldn't want to miss it. Marijke, the shared dancing lessons and reflective conversations meant a lot to me. Rick and Renske, I speak to you probably less than yearly, but it has been very valuable to me nonetheless. Thank you for the warm welcomes and enjoyable times. Jaap, Jan and Tim - knowing you close to 30 years now! Oh my. Thanks for still popping up now and then, a process that is simplified by our parents being close friends and facilitated via your creativity in art exhibitions, and projects such as quarantine radio and remodeling a former summer house. I suddenly might have time to help out very soon.

There are very few people that are in my life longer. My family is small but important with uncles and aunts Koen, Marjan, Wim and Irene, Annelies, Ad en Anneke and Fieke and Jaap. Let's not forget to have family events now and then, including the new generations. And of course there is my sister Ruth, whom I cherish for her sisterly support and wisdom, and therefore I am very happy with the warm family she is building together with Lancelot (our non-Brabander-but-still-brilliant and relaxed guy that brings forth interesting new perspectives into our family discussions) and my sweet nieces Salomé and Rozemarijn. It's heartwarming to see how you're doing and to be a part of it in one way or the other. Dearest parents, lieve papa en mama, I'm simply so happy to be your son. Thank you for your unshakable confidence and pride, thanks for being there, no matter what is on my mind.

Curriculum vitae

Willem Sanberg was born on 15 June 1986, in Tilburg, the Netherlands. He obtained both his BSc. degree in Electrical Engineering and a certificate Technical Management from Eindhoven University of Technology, the Netherlands, in 2011. After completing an internship at Philips Research on facial hair detection in textureless 3D face scans and a university graduation project on graph-based RGB+D segmentation at the Video Coding & Architectures Research group of the TU/e, Willem obtained his MSc. degree in Electrical Engineering in 2013. He continued at the VCA group as a Ph.D. researcher within the Mobile Perception Systems subgroup. His research addresses computer vision algorithms for Advanced Driver Assistance Systems, using a color stereo camera as the main sensor. Within this field, he worked successfully first on freespace segmentation with online self-supervised models, second on efficient and accurate 3D scene modeling in a surveillance context, and third, on a generic, class-agnostic collision-warning system. His work has been published at several IEEE conferences and a journal and was rewarded a best-paper award on IS&T 's Autonomous Vehicles and Machine conference in 2019. His algorithms are designed to be efficient and yet robust against adverse imaging conditions, and several have been demonstrated in live prototype vehicles.

As a Ph.D. researcher, Willem has been a short-term visiting research engineer at ViNotion B.V. in 2016, participated in Connekt's ITS Leadership Exchange Program 2015-2016 and in the International Computer Vision Summer School in 2017. He has contributed to several European and national projects with various international partners such as the Dutch Ministry of Defence, TNO, ViNotion, ViComTech, Dublin City University and the Honda Research Institute in Germany. As a student, Willem has been an active member of student associations E.S.V. Demos and University Racing Eindhoven, developing managerial and multi-disciplinary engineering skills. In his spare time, you can find him riding a road bicycle, reading a book, cooking dinner, playing a piano or sailing a boat.

Willem joined NXP Semiconductors as an AI System Engineer at the CTO Automotive System Innovations Department in December 2019, engineering and steering AI research towards realistic real-world solutions.