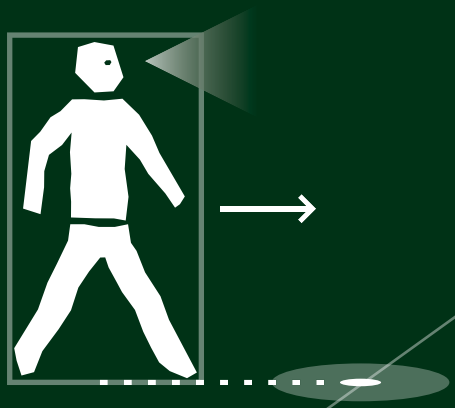
Vulnerable road user detection and orientation estimation for context-aware automated driving

Flohr, F.B.

Publication date
2018

Document Version
Final published version

License
Other



Vulnerable Road User Detection and Orientation Estimation for Context-Aware Automated Driving

Fabian B. Flohr

Vulnerable Road User Detection and Orientation Estimation for Context-Aware Automated Driving

Fabian Berthold Flohr

Cover design: Lukas Flohr
Printing: Ridderprint BV
ISBN: 978-94-6375-100-1

Vulnerable Road User Detection and Orientation Estimation for Context-Aware Automated Driving

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. K. I. J. Maex
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op vrijdag 26 oktober 2018, te 14:00 uur

door

Fabian Berthold Flohr

geboren te Biberach an der Riß, Duitsland

Promotiecommissie

Promotor:	Prof. dr. D. M. Gavrila	Universiteit van Amsterdam
Copromoter:	Dr. J. F. P. Kooij	Technische Universiteit Delft
Overige leden:	Prof. dr. ir. B. J. A. Kröse	Universiteit van Amsterdam
	Prof. dr. C. G. M. Snoek	Universiteit van Amsterdam
	Prof. dr. C. Stiller	Karlsruher Institut für Technologie
	Prof. dr. M. Welling	Universiteit van Amsterdam
	Prof. dr. C. Wöhler	Technische Universität Dortmund

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

This research has received funding from the German Federal Ministry for Economic Affairs and Energy under grant agreement N^{o} 19S12008A, the UR:BAN project, and from the European Community's Eighth Framework Program (Horizon2020) under grant agreement N^{o} 634149, the PROSPECT project. The research also received funding from the Daimler AG.

CONTENTS

SUMMARY

Vulnerable Road User Detection and Orientation Estimation for Context-Aware Automated Driving

Fabian Berthold Flohr

This thesis addresses the detection, segmentation and orientation estimation of persons in visual data. While the possible application domains of the proposed methods are manifold, ranging from image editing over surveillance to robotics and the intelligent vehicle domain, the latter is in focus of this work. In particular, the work focuses on the role of the Vulnerable Road Users (VRU, e.g. pedestrians and cyclists), being among the most critical objects for the realization of self-driving vehicles. A human driver is able to efficiently detect and localize a VRU on the street. Furthermore, a human driver recognizes important context information of the VRU (e.g. awareness, intention) and the environment (e.g. infrastructure elements), helping him to anticipate the VRU behavior. From an automation perspective, it is desirable to imitate or even outperform the skills of a human driver with a machine system. Motivated by this, the aim of this work is to establish an accurate machine representation of the VRU by using image-based cues to support context-aware automated driving.

The first addressed problem is a reliable ***detection of the VRU***, being a crucial preliminary step for all subsequent modules. The detection of VRUs is especially challenging due to their wide variation in appearance, arising from articulated pose, clothing, background and visibility conditions (time of day,

weather). To cope with these challenges, a stereo-based superpixel representation (i.e. stixels) is applied for efficient proposal generation. The resulting proposals are used within a Deep Convolutional Neural Network architecture to gain a robust object detection. Results are discussed on a newly introduced dataset, being the first dataset of this size, focusing on the challenging detection of cyclists in urban areas. Even with a significantly reduced proposal count compared to commonly used 2D proposal methods, competitive detection results are gained.

Based on the robust detection, ***pixel-wise VRU segmentation*** is considered to facilitate higher-level, semantic scene analysis (e.g. body part labeling, pose estimation, activity analysis). Furthermore a pixel-wise segmentation has the potential to enhance the detection and localization performance in itself. The large variety of VRU appearances make the problem again challenging. On the other hand, focusing on a single object class makes it possible to introduce a fair amount of prior knowledge on how pedestrians appear in images. The proposed method combines generative shape models and multiple data cues within an iterative framework. In each iteration, shape and data cues are refined leading to an accurate segmentation after only a few iterations. Experiments on a public segmentation dataset suggest that the proposed method outperforms state-of-the-art. To analyse the benefit of using additional disparity cues for segmentation, a new pedestrian segmentation dataset has been introduced.

Looking at ***head and body orientation of a VRU*** supports a human driver to estimate the motion state and attention of a VRU within a fraction of a second. Therefore, a new method for joint probabilistic head and body orientation estimation has been created that handles faulty part detections, continuous orientation estimation, coupling of the body and head localization and orientation, and temporal integration. For both head and body parts, responses of a set of orientation-specific detectors are converted into a (continuous) probability density function. Head and body parts are estimated jointly to account for anatomical constraints. The joint single-frame orientation estimates are integrated over time by particle filtering. The experiments involve data from a vehicle-mounted stereo vision camera in a realistic traffic setting. It is shown that the proposed method reduces the mean absolute head and body orientation error significantly compared to simpler methods, resulting in stable orientation estimates which remain fairly constant up to a distance of $25\,\mathrm{m}$.

Methods have been applied in ***realtime system integrations*** on-board of experimental vehicles and tested in complex, real-world traffic scenarios. Visual

context cues are deployed to gain an improved VRU path prediction. In particular, head and body orientation estimates are used to anticipate the behavior of a pedestrian by modeling situational awareness within a context-based Switching Linear Dynamic System. System components are described and influences on the vehicle intervention strategy are pointed out for the pedestrian case. Based on real world test sequences it can be confirmed that the prediction horizon can be increased up to $1\,\mathrm{s}$ without increasing the false alarm rate. A demonstration design has been worked out to present the system in an understandable way.

Data annotation and management are indispensable components for the development of machine learning applications. Accurate and correct data annotation has a direct influence on the quality of machine learning results. Furthermore, a data management and provisioning process is needed for handling the large amount of data needed to train complex models. Two software tools are presented to gain an efficient data annotation and management process. The tools have been used for all data annotation tasks in this thesis and have been shared with partners of public research projects.

The thesis completes with a conclusion of the individual chapters and overall insights. Various findings are discussed in relation to each other, and directions for future work are put forward.

SAMENVATTING

Vulnerable Road User Detection and Orientation Estimation for Context-Aware Automated Driving

Fabian Berthold Flohr

Dit proefschrift behandelt verschillende problemen uit de computervisie, zoals objectdetectie, segmentatie en oriëntatieschatting. Hoewel de mogelijke toepassingsdomeinen van de voorgestelde methoden veelvuldig zijn, variërend van applicaties in beveiliging tot intelligente voertuigen, focust dit werk vooral op dit laatste applicatie domein. In het bijzonder benadrukt het de rol van de kwetsbare weggebruikers, "Vulnerable Road Users" (VRU) in het Engels genaamd. Dit zijn bijvoorbeeld voetgangers en fietsers, die tot de meest kritische objecten behoren waarmee volledig geautomatiseerde voertuigen dienen om te gaan. Menselijke bestuurders zijn in staat om VRUs efficiënt te detecteren en lokaliseren. Verder herkent een menselijke bestuurder belangrijke contextinformatie van de VRU (bijvoorbeeld aandacht, intentie) en van de omgeving (bijvoorbeeld infrastructuurelementen), waardoor deze kan anticiperen op het gedrag van de weggebruiker. Vanuit een automatiseringsperspectief is het wenselijk om deze vaardigheden van een menselijke bestuurder te imiteren met een computer, of zelfs te overtreffen. Dit doel motiveert het gepresenteerde werk, namelijk om een nauwkeurige geautomatiseerde representatie van de VRU tot stand te brengen op basis van beeldinformatie, en met deze context het geautomatiseerd rijden te ondersteunen.

Het eerste probleem dat wordt aangepakt is een betrouwbare ***detectie van de VRU***, wat een cruciale eerste stap is voor alle volgende modules. De detectie van zulke weggebruikers is zeer uitdagend vanwege het de grote variatie in uiterlijk, lichaamshouding, kleding, achtergrond- en zichtomstandigheden (tijdstip van de dag, weer). Een efficiënte methode wordt gepresenteerd om potentiele regio's te vinden die mogelijk voetganger bevatten, gebruikmakend van een superpixel representatie gebaseerd op stereo beelden (zogenaamde "stixels"). De resulterende voorstellen worden in verschillende Deep Convolutional Neurale Netwerk-architecturen gebruikt om een robuuste objectdetectie te behalen. Resultaten worden gepresenteerd op een nieuwe dataset, de eerste van deze omvang, die de nadruk legt op de uitdagende detectie van fietsers in stedelijke gebieden. Zelfs met een aanzienlijk gereduceerd aantal voorgestelde detectie kandidaten wordt een competitief resultaat behaald, in vergelijking met detectie kandidaten uit gebruikelijke 2D methoden.

Na de robuuste detectie, wordt ***per-pixel segmentatie*** van de VRU bestudeerd, wat de semantische analyseren van de omgeving op een hoger niveau faciliteert (bijvoorbeeld het detecteren van lichaamsdelen, afschatten van de lichaamshouding, activiteitsanalyse). Bovendien heeft een per-pixel segmentatie de potentie om de detectie- en lokalisatieprestaties an sich te verbeteren. De grote verscheidenheid aan VRU voorkomen, veroorzaakt door gezichtspunt, houding, kleding en belichting, maakt ook dit probleem bijzonder uitdagend. Aan de andere kant ligt de focus ditmaal op een enkele type object, wat het mogelijk maakt om een behoorlijke hoeveelheid voorkennis te benutten over hoe voetgangers eruit zien in camerabeelden. De voorgestelde methode combineert generatieve vorm modellen met verscheidene gegevens in een iteratief raamwerk. In elke iteratie worden vorm en geïnterpreteerde gegevens verfijnd, wat leidt tot een nauwkeurige segmentatie in slechts enkele iteraties. Experimenten op een openbare segmentatie dataset suggereren dat de voorgestelde methode beter presteert dan het best beschikbare. Om het nut van beeld dispariteit aan te tonen voor segmentatie is een nieuwe voetganger segmentatie dataset geïntroduceerd.

Het observeren van de ***hoofd- en lichaamshouding*** van een VRU ondersteunt menselijke bestuurders om de beweging en aandacht van een VRU binnen een fractie van een seconde in te schatten. Om die reden is ook een nieuwe methode ontworpen die tegelijkertijd zowel hoofd- als lichaamsoriëntatie probabilistisch inschat, en die om kan gaan met incorrect gedetecteerde onderdelen, continue oriëntatie-schatting, de relaties tussen lichaam- en hoofdlokalisatie- en

-oriëntatie, met temporele integratie. Voor zowel hoofd- als lichaamsdelen worden de responsen van een reeks oriëntatiespecifieke detectoren omgezet in een (continue) kansdichtheidsfunctie. Hoofd- en lichaamsdelen worden gezamenlijk geschat om rekening te houden met anatomische beperkingen. De gezamenlijke per-beeld oriëntatie schattingen worden geïntegreerd over tijd door een deeltjesfilter. De experimenten worden uitgevoerd op data verzameld met een stereo-camera gemonteerd op een voertuig in realistische verkeersomgevingen. Aangetoond wordt dat de voorgestelde methode de gemiddelde absolute schattingsfout van de hoofd- en lichaamsorientatie significant vermindert in vergelijking met eenvoudigere methoden, wat resulteert in een stabiele schattingen van de oriëntatie die redelijk constant blijft tot een afstand van 25m.

De methoden zijn toegepast in ***realtime geïntegreerde systemen*** aan boord van experimentele voertuigen, die zijn getest in complexe verkeersscenarios in de echte wereld. Visuele contextaanwijzingen worden geïmplementeerd om een verbeterde voorspelling van het VRU pad te verkrijgen. In het bijzonder worden schattingen van hoofd- en lichaamsoriëntatie gebruikt om het gedrag van een voetganger te anticiperen, door de aandacht van de voetganger voor diens omgeving te modelleren in een context-gebaseerd Switching Linear Dynamic System. De benodigde systeemcomponenten worden beschreven, en de effecten op de voertuiginterventiestrategie worden beschreven voor dit voetganger scenario. Op basis van meerdere tests in de echte wereld kan worden bevestigd dat de voorspellingshorizon kan worden verhoogd tot 1s zonder het percentage valse alarmen te verhogen. Ter verduidelijking van het demonstratiesysteem wordt diens ontwerp uitgewerkt en gepresenteerd.

Data annotatie en beheer zijn uitdagende en onmisbare componenten voor de ontwikkeling van machine learning-applicaties. Nauwkeurige en correcte annotatie van gegevens heeft een directe invloed op de kwaliteit van de resultaten. Verder is een gegevensbeheer- en bevoorradingsproces nodig voor het verwerken van de grote hoeveelheid gegevens die dienen om complexe modellen te trainen. Twee software gereedschappen worden gepresenteerd om een efficiënt gegevens annotatie- en beheerproces te verkrijgen. De hulpmiddelen zijn gebruikt voor alle data annotatietaken in dit proefschrift, en zijn gedeeld met partners van publieke onderzoeksprojecten.

Het proefschrift sluit af met het trekken van conclusies uit de voorgaande hoofdstukken. Verschillende bevindingen worden in relatie tot elkaar besproken, en mogelijke richtingen voor toekomstig onderzoek worden voorgesteld.

1 INTRODUCTION

Our senses help us to capture and interpret the environment around us. Having powerful visual sensor capabilities available, the human uses implicit and learned knowledge on the structure of objects to interpret this sensor information. Mainly by vision, we are able to detect and classify different objects in various poses and locations. We can distinguish them from complex background and with our stereoscopic perception we can even perform an accurate localization of these objects. Additional extracted context information help us to understand objects, predict their behavior and finally to interact with our environment and other persons. Humans are using these capabilities for various tasks in daily life. For some of these tasks it is desirable to support or even replace the human by modern computer systems. Applications for such systems are manifold and are ranging from image and video editing over surveillance to the intelligent vehicles domain.

In a self-driving vehicle [82] for example, the driver gets completely replaced by a computer system. The system needs to perform the driver tasks with a similar or even with a better performance than the human, especially in critical situations. But what information does the human use to interpret traffic scenes? And what should a computer system use?

Humans make strong use of context information extracted from other ob-

jects to assess a situation. Based on a common scene in urban areas, Figure 1.1 provides some examples of such object context cues a human might use in an upcoming critical situation involving multiple pedestrians. Beside a *robust detection*, *classification* and *localization* of body parts, a human driver is able to instantly infer the *motion state* (e.g. by using the *body pose*) and the *intention* or *awareness* (e.g. by using the *head orientation*) of a pedestrian. While temporal information would provide an even more complex and richer input, a human gains already a lot of information from a static assessment as shown in Figure 1.1. The human links the extracted object context to information extracted

Figure 1.1: A typical scene in an urban area involving multiple pedestrians. The human driver makes implicitly use of additional object context information which helps him to interpret the scene. Intuitively he is able to detect and classify each object on the street. Even without temporal information a human driver can also judge on the motion state, intention and awareness of pedestrians and use these cues to better predict the upcoming situation.

from the environment, e.g. the location of street infrastructure elements or the position and speed of other objects. With the combination of environment and object context information a human improves his knowledge about the future behavior of other traffic participants and is able to perform a detailed assessment of a situation. Can a computer system extract the same detailed information?

Looking at other application domains, similar disciplines can be identified. Similar to the above example, *object detection* and *interpretation* plays an important role in the surveillance domain [113, 128]. Automatic detection and interpretation of objects greatly reduce the reliance and workload on the human surveillance operator, making him focus on more complex tasks. *Object segmentation* in a video sequence is a tedious but common task often used in the movie industry [225], e.g. to remove certain objects or persons in a video sequence. It is also often used in surveillance applications to receive better object localizations in the image [132]. *Object orientation* plays also a strong role in social robotics, where the aim is that a robot socially interacts with persons in its environment [72]. The robot can greatly benefit from knowing *head* and *body orientation* of the conversational partners to improve the social interaction.

Different challenges in the application domains steer the final algorithm design. While most applications in surveillance can work with a static background model, objects in intelligent vehicle or social robot applications are observed against an ever-changing complex dynamic backdrop and are subject to stronger realtime constraints. In contrast to that, image or video editing applications are often used in offline-processing mode.

This thesis focuses on intelligent vehicles as application domain. An accurate image-based machine representation of the Vulnerable Road User (VRU, e.g. pedestrians and cyclists) is established to support context-aware automated driving. Guided and motivated by Figure 1.1, the thesis touches various topics from *computer vision*. It deals with *object detection*, including an accurate *segmentation* of the object, and *object orientation estimation* to extract valuable image-based object context cues to build up the VRU representation. To solve these tasks object-specific knowledge is combined with rich bottom-up feature information, similar to the way human perception works. Looking again at the situation in Figure 1.1, it can be also asked: What will happen next? Which pedestrian will finally cross the street in front of the approaching vehicle and is likely to run in a critical situation? This question touches the area of *object tracking and prediction* and a better answer might be given by using the extracted VRU details to improve the VRU *path prediction* within a concrete *vehicle integration*. Furthermore we discuss the *data labeling* and *management* processes as an important preliminary step to develop and optimize those intelligent algorithms. The proposed methods can be applied (after appropriate adjustment of assumptions and parameters) also to other object classes and / or other application domains (e.g. automated surveillance, entertainment, image

and video editing and robotics). Since our object of interest is the VRU, the next sections are devoted to its special role in the intelligent vehicle domain. At the end of this chapter the thesis content is summarized and contributions of the performed work are pointed out.

1.1 The Vulnerable Road User

Definition 1 *A Vulnerable Road User (VRU) is a road user with an increased risk of being injured or killed in traffic because he is not surrounded by a protective cover which would significantly reduce the severity of an accident.*

With Definition 1 all kinds of pedestrians, cyclists and motorcyclists, persons with disabilities or reduced mobility (e.g. wheelchair users or children in a stroller) are covered. On the other hand, this definition excludes e.g. driver of cars and passengers of public transport, since they are protected by a protective cover.

In 2013, 1.25 million people have died on the roads worldwide. Unfortunately, road traffic injuries are still the number one cause of death among people aged between 15-29 years [232]. As Figure 1.2 points out, roughly half of those killed (i.e. about $612{,}500$ persons) belong to the group of VRUs. The *Cyclist* class contains all non-motorized two- or three-wheelers. The *Other* class covers road users types, which were not specified in the accident data used for the report. There is a high probability that this class contains even more VRU fatalities. While most of the reported VRU fatalities occur in Asia-Pacific and African countries, on average 70 VRUs are killed in every hour worldwide [232]. This depressive statistics show that a better protection of the VRU is required, leading to an improved public health and road safety.

1.1.1 Improving Road Safety for the VRU

Luckily, the number of road traffic deaths has plateaued since 2007 [232] in relation to the global increase of population. That means that interventions taken during the last years have improved global road safety significantly. A lot of cities have reduced the amount of motorized vehicles by establishing a better infrastructure for public transport and car sharing offers, and have better separated motorized traffic from unmotorized traffic [232]. The implementation of low speed areas, areas with limited traffic, roundabouts, traffic lights and traffic

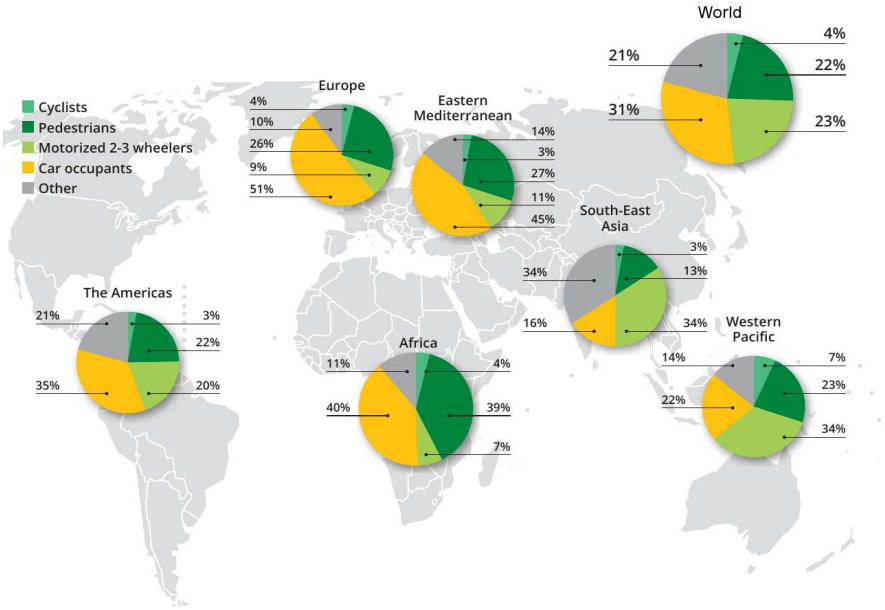


Figure 1.2: Road traffic deaths by type of road users [232].

signs had also a positive effect on the road safety in inner cities. The same can be found on highways where better road layout and infrastructure (introduction of variable speed limit signs and automatic traffic jam and accident recognition) lead to safer roads on high-speed routes [232].

Another major direction involves improving visibility and communication between road participants. Often, accidents occur because the VRU was just not seen (e.g. because of occlusion) or low perceptibility (e.g. in bad light conditions). Thus, reducing bad road layouts can also lead to a major improvement [232]. Also new rules have been introduced, like for example the obligation to wear helmets for cyclists and drivers of motorized two-wheelers. But what if the infrastructure is perfectly established and maintained. Could we then already reach the *zero vision* [230] in inner cities - having zero VRU fatalities?

Even if road infrastructure is well established, the VRU or the driver of a vehicle can make errors. Careless entry of the roadway, e.g. due to increased

distractions by smart phones and tablets or other attention spots are examples where the VRU is not attentive enough. But also a driver might be distracted, e.g. by the on-board entertainment system or by other actions around the vehicle. In these cases the driver might lose attention and is likely to oversee critical situations involving VRUs. But even an attentive driver is not always able to interpret the VRU's behavior correctly and accidents might occur because of a misinterpretation of the VRU behavior. Same applies also if the VRU misinterprets the intention of the vehicle driver.

During the last decade, artificial intelligence found its way into nowadays production vehicles. Emerging technologies in intelligent vehicle systems that automatically recognize and react to critical situations might help to solve such remaining conflicts. Those systems might improve road safety for VRUs (and also for the driver). It is too early to entirely prove this statement, since only a minor percentage of vehicles are equipped with these intelligent systems protecting the VRU. Nevertheless, already available studies and tests [228, 235] suggest that the impact is significant.

1.1.2 Vehicle Automation and the VRU

Various Advanced Driver Assistance Systems (ADAS) [93, 179] are already available to support the driver either in common driving maneuvers or in near accident cases. Most of these available systems are working either with camera, radar, ultrasonic or a combination of these sensor technologies. Upcoming systems designed for higher automation level will even use lidar technologies [186]. Although still expensive, lidar sensors can measure accurate distances and are therefore very promising for the development of ADAS or self-driving vehicles.

The Society of Automotive Engineers (SAE) identified six levels of driving automation [201] from *no automation* to *full automation*. With a higher automation level there is also a strong need for "*seeing*" systems, which are able to detect and classify the environment in front of the vehicle based on appearance. Color and illumination are important cues for classification and therefore indispensable for understanding objects and the environment. In contrast to a camera image, radar and lidar sensor measurements are usually not dense and fail to measure color and illumination accurately. Therefore mono- or stereo-based vision sensor systems are often used for these tasks, since they can tackle various classification challenges and have low production costs. Another advantage of camera-based systems is that the development process of vision-based

machine learning applications can often be easier derived from human perception [136].

With increasing automation level, also the role of VRUs and how to handle them changes. In case of current available ADAS technologies (level 1-2 [201]), the driver is always in the loop and still responsible in all situations. This has a direct consequence for the practical implementation of a VRU protection system. Since the driver is always in charge, and should in theory be able to handle all critical situations, it is not urgently necessary to detect all critical situations automatically. On the other side, a wrong and unnecessary intervention of the system, e.g. an emergency brake or evasive maneuver, is hardly acceptable. Wrong interventions or warnings would annoy the driver or even put him or other road users in danger. When developing automatic recognition systems there is usually a trade-off between the False Positive Rate (FPR), being the proportion of wrong recognitions, and the True Positive Rate (TPR), being the proportion of correct recognitions. While a high TPR and a low FPR should be preferred for VRU protection systems, increasing the TPR also increases the FPR in most cases. Practically that means that ADAS (level 1-2) might choose a rather low FPR for VRU protection systems, avoiding dangerous wrong interventions but accepting the risk to oversee something.

The New Car Assessment Program (NCAP) [228] is testing the safety for the driver and other road users in various disciplines. Beside passive pedestrian safety tests, tests on pedestrian protection systems (i.e. Automatic Emergency Brake for pedestrians [44]) have been introduced in 2016 [169], with test speeds up to 60 km/h. Vehicles passing these tests gain additional safety points in the overall rating. Even in the relatively simple use cases, in most cases a collision can only be avoided completely up to 30-40 km/h by the tested systems [169]. In the latest NCAP safety ratings only a few systems manage to avoid a collision with the full tested speeds up to 60 km/h (e.g. Volvo V90, Subaru XV, Mercedes-Benz X-Class). Until now, the pedestrian was in focus of these tests. In 2018, cyclist use cases will be added to the NCAP tests.

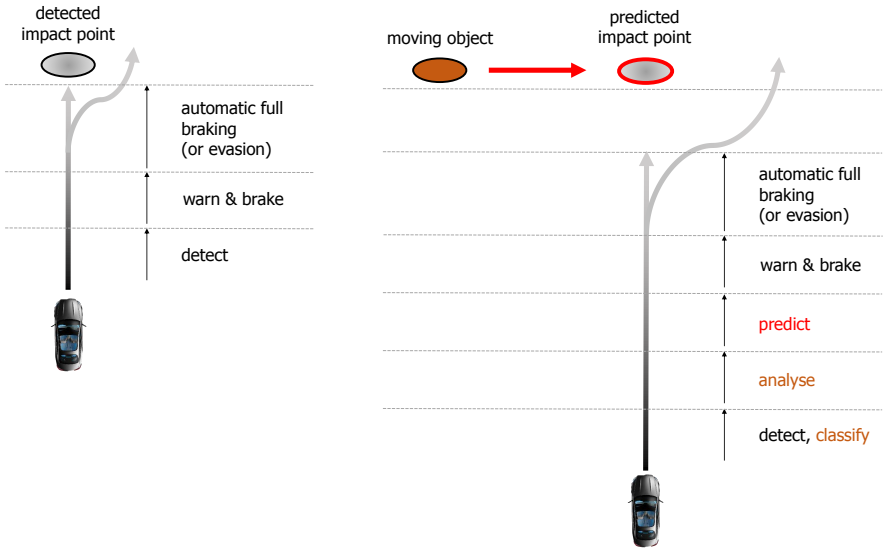
For an increased automation level (level 3-5 [201]) the problem gets even tougher. While a very low FPR is still desired to avoid wrong interventions, it is not possible for a self-driving vehicle to oversee a critical situation, since the driver might not be in charge anymore or even not existent. This means that already for conditional automation (level 3), where VRUs might be involved, “all” critical situations need to be tackled by the system. That poses a major challenge to companies taking the race in autonomous driving [148].

1.2 Towards a Better Understanding of the VRU

With the introduction of VRU protection system [124], classification has been added in the intelligent vehicle to explicitly differ a vehicle intervention on VRUs to other objects. Limited computing capabilities and the lack of appropriate methods prevented a detailed analysis or prediction of the object for these systems. The earlier a VRU can be detected and the better his motion can be predicted, the better an ADAS or self-driving vehicle can react on this VRU.

Unfortunately, just increasing the prediction horizon also increases the uncertainty and therefore this might also increase the FPR, which is not desired. With the entry of a bigger bandwidth and modern high speed CPUs / GPUs into the intelligent vehicles domain, a closer look on the scene becomes possible. With the right methods at hand, the VRU can be detected earlier. Furthermore, the extraction of valuable context information allows a better analysis and finally a more accurate prediction of the VRU's motion. Figure 1.3b shows the main phases of this extended system. Note that in comparison to Figure 1.3a, an earlier and stable prediction of the VRU motion leads to a safer and smoother intervention, in the optimal case without increasing the FPR. While the possible motion of VRUs is inherently different, e.g. a pedestrian's motion pattern is completely different to that of a cyclist, it is also important to distinguish between different VRU types before an accurate path prediction for each potential critical VRU can be performed. In particular for VRUs, path prediction is challenging due to their high maneuverability - e.g. pedestrians can change their walking direction or accelerate / decelerate at a whim. Any auxiliary information that can help to reduce this uncertainty in path prediction is welcome. The first obvious thing is to improve localization accuracy of the VRU. Camera-based sensors deliver dense measurements, and a stereo-camera system can even deliver accurate 3D positions for each pixel. By using the disparity information, 3D localization can be greatly improved compared to a simple flat world assumption commonly used with mono-camera systems [92]. A pixel-wise segmentation of the pedestrian can further improve the localization accuracy [233] and serves as an enabler for extracting more specific cues like head and body part locations and an orientation of those parts. There has been a lot of fundamental work in detection [52, 66, 75, 180] and segmentation [87, 198, 217] for various object types. Recently, it was shown that using *Deep Learning* methods [39, 97, 233] can greatly improve the results in these two disciplines.

As pointed out, the human driver can immediately judge on the attentive-



(a) Classical object detection

(b) Advanced object detection

Figure 1.3: Different phases for (a) classical object and (b) advanced object detection systems.

ness and motion state of VRUs. Empirical evidence suggests that the pedestrian body and head orientation are indeed good indicators a human is also using. They can provide important information what the pedestrian will do next. For example, a human factors study by Schmidt and Färber [204] had several test participants watch videos of pedestrians walking towards the curbside and decide whether the pedestrians would stop or cross, at various time instants. The study varied the amount of visual information provided to the test participants and examined its effect on their classification performance. The study shows that head motion was among the most important indicators for future pedestrian action. Furthermore, Hamaoka *et al* [104] show that the head turning frequency increases towards the entry of a crosswalk and a conflict point [104].

Let us conclude: Having an earlier detection of the VRU available, allows us to extract behavior relevant details (e.g. the head or body orientation as VRU context cue). This more detailed analysis makes it possible to apply better motion models for the VRU and as a result it allows us to increase the prediction

horizon. Finally, a self-driving vehicle (or ADAS) can react earlier on a critical situation with a VRU.

1.2.1 Objectives of the Thesis

Based on the drawn conclusion from the previous section we can define the following research objectives of this thesis:

1. **Detect and Classify** Provide a robust and efficient *detection* and *classification* for VRUs which might produce a critical situation in the near future. Discussed in Chapter 3.

2. **Analyze (extract VRU shape)** Infer a better localization of detected VRUs (and their body parts), e.g. by means of a *pixel-wise segmentation*. This accurate segmentation can serve as an enabler for extracting further important VRU context cues. Discussed in Chapter 4.

3. **Analyze (extract VRU context cues)** Extract VRU context cues that give an indication about the future VRU behavior and intention, e.g. *head* and *body orientation* of the VRUs. Discussed in Chapter 5.

4. **Predict** Use extracted VRU context cues to improve VRU *path prediction* within an *integrated system*. Discussed in Chapter 6.

5. **Data Handling** Perform continuous data *collection* and *labeling* to develop and optimize the methods. Recorded sequence data and labeling information must be *managed* efficiently and accessible. Discussed in Chapter 7.

Having these objectives in mind, concrete implementation of these objectives face various challenges, which are summarized in the next section.

1.2.2 Challenges

In the following we list the most prominent challenges for the intelligent vehicles domain, and in particular we focus on those having an influence on a camera-based detection and analysis of VRUs. These different factors are greatly influencing the appearance of a VRU to the observer, making detection, classification, analysis and prediction of the VRU especially challenging.

- **Low resolution** Supposing we have to perform a braking maneuver on a VRU, the minimum distance in which a VRU needs to be detected depends on the speed of the ego vehicle and the braking distance. With a common city speed of 50 km/h, the minimum braking distance on a dry / wet street (with a friction coefficient of $\mu = 0.7$ / $\mu = 0.4$) is 14 / 25 m. Adding a system latency of 400 ms, a VRU needs to be detected in 19.6 m / 30.6 m ahead to react in time. Supposing a typical available automotive image sensor (1176 px $\times$ 640px), a VRU with a height of 1.80 m has an image height of only 70 px in a distance of 30 m. Figure 1.4a shows a pedestrian in this distance. It is clear that already the detection becomes challenging here.

- **Moving vehicle** Because the ego vehicle is moving, background is dynamic and ever-changing. This affects the separation of the target object from the background. Compared to the surveillance domain, where background is often static and background subtraction can be applied, special attention needs to be paid here. Often object / background separation is not trivial, e.g. due to low contrast regions or similar background patterns (Figure 1.4b).

  Additionally target objects need to be tracked over time to get additional robustness and for path prediction. Therefore the ego motion needs to be compensated. Ego motion can be estimated either from inertial sensors or image [11].

- **Complex object motion** VRUs can exhibit complex and highly nonlinear object motion. A pedestrian can change his walking direction in a whim. Accurate path prediction is one of the main challenges for ADAS or self-driving vehicle systems. Uncertainty of the future VRU behavior is ubiquitous and dependent on different environment factors including other traffic participants. The pedestrians in Figure 1.4c are *aware* of the approaching vehicle. Will they stop, or not?

- **Nonrigid or articulated nature of VRUs** Compared to vehicles and most other street objects, VRUs, and in particular pedestrians, are strongly nonrigid objects. Depending on their motion state, they have a different shape (e.g. gait cycle). See Figure 1.4d for an uncommon pose of a human, leaving his vehicle.

- **Blur and other noise** To still gain a bright image, camera exposure time

needs to be higher at lower light conditions. With increasing exposure time, motion blur can occur (Figure 1.4e), in particular when objects or ego vehicle are moving fast. Other noise can occur due to weather conditions (e.g. by snow, rain, dust and fog).

- **Object occlusions and truncation** Especially in street scenes, different occlusion types and occlusion levels can occur. VRUs can be occluded by vehicles and other street objects. In the optimal case, a VRU needs to be detected before he steps out behind a parking vehicle. VRUs can also be occluded by other VRUs (Figure 1.4f) and can occlude themselves (self-occlusion), e.g. by carried bags, clothing. Additionally, object truncation occurs due to the limited sensor field of view at image borders.

- **Illumination changes** Illumination is strongly dependent on weather conditions; light source position and shadow strongly influence the appearance of an object (Figure 1.4g).

- **Individual appearance and shape** Target objects have a great variance in appearance itself. VRUs appear in different sizes and various clothing styles, colors and textures (Figure 1.4h). Some appearance patterns might even have a special meaning (e.g. police, road worker), which need to be recognized. This complements the final appearance to the observer which is also influenced by aforementioned factors (e.g. articulation, occlusion, illumination, noise and object motion).

- **Realtime processing requirements** ADAS and self-driving vehicles require low latency and are bound to hard limits regarding computation time and resources. Suddenly occurring situations (Figure 1.4i) need to be detected fast to react in time.

Figure 1.4: Image samples taken from the Tsinghua-Daimler Cyclist Benchmark Dataset [144] showing challenges for detection and segmentation of VRUs: (a) low resolution, (b) difficult background, (c) complex object motion, (d) object pose, (e) blur and noise, (f) occlusion, (g) illumination changes, (h) individual appearance and (i) a time-critical situation.

1.3 Contributions

Regarding the thesis objectives listed in the previous section and the challenges associated with them, the main contributions of the thesis are:

1. A new architecture for efficient object detection is introduced, combining Fast-RCNN [95] with a stixel-based proposal generation [68]. The stixel-based proposal generation allows an easy adaptation to different runtime conditions. Other stereo proposal methods [40, 101, 146] show runtimes greater than 1 s and are therefore not appropriate for a realtime application in the vehicle. Compared to [68], stixel-based proposals are created in multiple aspect ratios to capture the various viewpoints of different VRU types. Competitive detection performance is shown on the Tsinghua-Daimler Cyclist Benchmark Dataset, which was introduced as part of this work.

2. An iterative framework for accurate pedestrian segmentation, combining generative shape models and multiple data cues is presented. By combining top-down shape knowledge with bottom-up cues the method is able to cope with the large variation of pedestrian appearances, across cluttered backgrounds. It is in spirit most related to that of Kokkinos and Maragos [130], although Expectation- and Maximization-steps are defined differently to [130]. The objects of interest, pedestrians, feature a larger appearance variation than the frontal faces and sideway cars of [130]. To cope with the stronger shape aspect variations, Multiple Statistical Shape Models (MSSMs) [94] are applied. Beside experiments on a newly introduced dataset, results show that the proposed method outperforms state-of-the-art.

3. A new method for joint probabilistic head and body orientation estimation is presented that handles faulty part detections and delivers robust continuous orientation estimates. Head and body localization and orientation are jointly estimated using a constrained temporal integration within a Dynamic Bayesian Network. This is in contrast to [14, 207], where no coupling between head and body was used at all. Furthermore in contrast to [14, 37, 38, 195] the work differentiates in several ways. The intelligent vehicles domain is considered, introducing additional challenges like complex and ever-changing background and lighting conditions. The

joint part localization and orientation estimation also accounts for the possibility of occluded body parts or false positives. Evaluations are performed on data from a vehicle-mounted stereo vision camera in a realistic traffic setting. The method has been successfully used for improving path prediction [133].

4. Visual context cues extracted from the VRU are used within the work of Kooij *et al* [133] to enhance path prediction. A realtime system integration of this work is presented, modeling the scenario of the crossing pedestrian who might stop or continue walking at the curb. Extensions for the cyclist case are also discussed. It is demonstrated how the extracted visual context cues (e.g. pedestrian head orientation) are used to model a more accurate path prediction. Influences on the vehicle intervention strategy are pointed out for the pedestrian case. Based on a real world test set it can be confirmed that the prediction horizon can be increased up to $1\,\mathrm{s}$ without increasing the false alarm rate. A demonstration design has been worked out to present the system in an understandable way.

5. A complete label data workflow has been specified, and then implemented together with an external company. The developed software, supporting multiple label mechanisms, has been used for all data annotation tasks in this thesis and was also made available to partners in the public project UR:BAN [33]. Compared to other label tool chains [43, 158, 183, 200] the proposed tools combine all major image labeling concepts needed for segmentation, classification and object detection tasks within an integrated workflow for the development of machine learning applications. It is connected to a database application, managing the sensor and label data. Data queries, label task creation and task controlling facilitate an easy and large-scale distribution of labeling work. More than one million labels have already been labeled with these tools.

6. Complex vehicle setups have been build up for the recording of multiple datasets. The two published datasets (for detection and segmentation tasks) make it easy for the research community to develop and benchmark new methods on real world automotive data. The richly annotated Tsinghua-Daimler Cyclist Benchmark Dataset (TDC) is introduced to benchmark cyclist detection. The dataset provides, among other objects, a total number of $22{,}161$ cyclist objects, nearly an order of magnitude more

than other state-of-the-art datasets [61, 90, 240]. Due to the lack of comparable segmentation datasets for pedestrian segmentation, the Daimler Pedestrian Segmentation Dataset (DPSD) for pixelwise pedestrian segmentation is introduced. The dataset contains 785 pixel-wise annotated pedestrian image cut outs including disparity information.

1.4 Outline of the Thesis

This Section provides a short overview over the different thesis chapters.

Previous Work Chapter 2 gives an overview of related works with the focus on feature extraction for an accurate object detection and an analysis by segmentation and orientation estimation, helping to better model the Vulnerable Road User. Also related works regarding the proposed system integration, data management and datasets are discussed. The focus lies on the intelligent vehicles domain, although ideas from other domains (e.g. automated surveillance, entertainment) are used and transferred.

Vulnerable Road User Detection A reliable detection of the VRU is the first step in understanding the VRU. In Chapter 3, an efficient method for the detection of VRUs is presented, utilizing the power of stixel-based proposals combined with a deep neural network architecture. Further, the chapter looks on details of the newly introduced Tsinghua-Daimler Cyclist Benchmark Dataset, being the first data set with this size, focusing on the challenging detection of cyclists in urban areas. Recordings have performed in Beijing - the city of bikes. In the following described experiments, the method is carefully compared with architectures using different proposal methods and to other state-of-the-art methods without a deep feature representation. The chapter ends with a critical discussion on weaknesses of the dataset and the introduced method and proposes further improvements. This chapter is based on the work published in [144].

Pedestrian Segmentation using Shape and Data Cues As object detection, also image segmentation is a key computer vision problem for a number of application domains, such as image editing [225], surveillance [132] and intelligent vehicles [203]. It facilitates higher-level, semantic scene analysis (e.g.

body part labeling, pose estimation, activity analysis) and can enhance the detection and the localization performance of the object or object parts in itself. Therefore, Chapter 4 focuses on a pixel-wise segmentation of a person which is observed against a complex and possibly dynamic backdrop, as it is the case on-board a moving vehicle. The large variety of pedestrian appearances, induced by viewpoint, pose, clothing and lighting, makes the problem especially challenging. On the other hand, by focusing on a single object class, makes it possible to introduce a fair amount of prior knowledge on how pedestrians appear in images. Given the intelligent vehicle context, we are also interested in the optional use of disparity data obtained from stereo vision. We shall look at experiment details where two different datasets are used to evaluate the new method. Possible extensions of the method are discussed. This chapter is based on the work published in [81].

Head and Body Orientation Estimation Knowing the object orientation facilitates a better model of the future object movement and allows also relevant applications in mobile robotics, where the aim is that a robot interacts socially with persons in its environment [72]. Also a human driver uses the head orientation of the VRU to better predict the future VRU behavior [204]. Thus, head and body orientations are important cues to enrich our model of the VRU. Based on an available bounding box detection and optional segmented body parts, a probabilistic framework for the joint estimation of pedestrian head and body orientation is presented. Head and body orientations are estimated jointly to account for anatomical constraints and integrated over time to gain further robustness of the estimate. After a detailed derivation of the mathematical formulation of the joint continuous orientation estimation, detailed experiments are presented, making use of a real world stereo-vision dataset recorded on-board a moving vehicle. The chapter concludes with a discussion on how the proposed method can be improved. Chapter 5 is based on the work published in [79, 80].

System Integration Concrete system integrations and demonstration can provide valuable feedback, confirming that a valid research path has been followed and that the developed methods are also applicable in a realistic environment. Therefore, Chapter 6 presents a concrete implementation and integration of the context-based path prediction (developed in [133]), making use of extracted visual VRU context features developed in this work. In particular, pedestrians head orientation is used to model the awareness of the pedestrian within the

path prediction formulation. System components are described, the influence on the intervention strategy is evaluated and possible adaptions are discussed. The system was shown during multiple public events. The used demonstration design is presented and possible extensions to the system are discussed. Chapter 6 is based on the work described in [79, 131, 133].

Labeling and Data Management In research and pre-development, tools need to be custom-tailored and fast adapted. Correct and efficient data labeling and management are among the most challenging parts here. The quality of data and data labels have a direct influence on the quality of machine learning results. Furthermore, a well-developed data management and provisioning process is needed for handling a large amount of data from different sensors. Data has become a major competitive advantage and since data acquisition and annotation is not unified among competitors, it is difficult to share tools here. Chapter 7 proposes a complete label data workflow including a label tool suite, which has been specified, and then developed together with an external company to gain an efficient data labeling and management process. The toolchain serves as a template for a data center facility, managing a fleet of self-driving vehicles. The discussion section finally provides an outlook and next steps are discussed. Chapter 7 is based on the work described in [81, 79, 133, 144], where the discussed labeling mechanisms have been applied and improved.

Conclusion and Future Work Chapter 8 summarizes the conclusions from previous chapters, and identifies relations between the encountered problems and findings. Open issues and directions for future work are discussed.

2 PREVIOUS WORK

This chapter presents an overview of related works in the main topics discussed in this thesis, i.e. detection, segmentation and orientation estimation. Related works regarding the presented vehicle integration and data management and labeling tools are also discussed.

2.1 Detection

The detection of objects in images is a key task in a number of important application domains, such as intelligent vehicles, surveillance, and robotics. A reliable detection and classification of the object of interest is a crucial preliminary step for a more detailed object analysis.

Classical object detection usually applies some kind of proposal generation to generate Region of Interests (RoI), which are likely to contain an object of interest, followed by a classification step. The simplest approach to gain these proposals is the sliding window approach, as used in [177, 223]. Proposals are generated based on heuristic knowledge about object size and location in the image, e.g. by using a planar world assumption [92]. Common sliding window approaches produce a vast number of RoIs, which makes it hard to meet realtime requirements even with efficient classifier approaches. Instead of us-

ing heuristic knowledge only, proposals can be generated or pre-selected based on the sensor data itself. Such data-driven proposal generators considerably reduce the computation time in the classification step and thus, enable the use of more complex classifiers consuming more computational power. According to Hosang *et al* [112], data-driven proposal methods can roughly be classified into grouping methods and window scoring methods. Grouping proposal methods [7, 35, 221] try to merge multiple (possibly overlapping) segments together so that it is likely they contain an object. Selective Search (SS) [221], for example, greedily merges superpixels to generate proposals and was used successfully with R-CNN [97]. Although grouping methods can provide proposal segments, often only the enclosing bounding box is used in the classification step. Window scoring methods [3, 41, 60, 243] directly generate bounding box proposals by scoring initial boxes according to how likely it is they contain an object. EdgeBox [243], for example, is using Structured Decision Forests [60] to score the object boundary from initial sliding window boxes. While window scoring methods are usually faster than grouping proposal methods, returned bounding boxes have lower location accuracy [112].

Recent works [40, 68, 101, 146] also show performance improvements by taking advantage of 3D sensors, including RGB-D and stereo cameras. Lin *et al* [146] extend the Constrained Parametric Min-Cut (CPMC) [35] framework in order to segment and understand indoor scenes based on RGB-D data. Gupta *et al* [101] propose an integrated system for scene understanding from RGB-D images by augmenting the Multiscale Combinatorial Grouping (MCG) [7] framework with depth information. Chen *et al* [40] solve the 3D object proposals generation within an energy formulation encoding several object priors and depth features. Enzweiler *et al* [68] generate grouping proposals for vehicle detection based on the stixel world [182]. It is shown, that compared to a depth filtering of sliding window proposals the proposed method allows an order of magnitude less generated proposals.

Object classification approaches can be roughly divided into methods using hand-crafted feature and those using learned feature representations. Hand-crafted features, like the Histogram of Oriented Gaussians (HOG) [52], are often used in combination with support vector machines [135] for different vision tasks. The use of boosting methods makes it possible to work with simpler feature representations like the Integral [59] and Aggregated Channel Features (ACF) [58, 236], reaching top scores in object detection benchmarks. The Deformable Part Model (DPM) [75] and its extensions [180] make already clear

that classification can be greatly improved by not restricting all object parts to one specific feature representation. Instead multiple feature representations are learned for different positions and scales. Although designing hand-crafted features before-hand makes it easier to understand the classification process, it restricts the learning process to the given feature representation.

The domain of object classification and detection is currently dominated by Deep Learning approaches [40, 50, 70, 97, 96, 149, 191, 194, 233]. Compared to hand-crafted features, a deep neural network jointly learns the feature representation during the training process. Girshick *et al* [97] combine region proposals generated by Selective Search [221] with deep Convolutional Neural Network architectures. The region CNNs (R-CNN) achieved a break-through in detection performance on the Pascal VOC 2012 challenge [71, 70]. Since then, a number of improvements have been made attempting to make R-CNN realtime capable. Fast R-CNN [96] introduces RoI feature pooling to avoid redundant feature calculation for every RoI. Recently, Ren *et al* [194] proposed a joint alternating training using a Region Proposal Network (RPN) based on predefined anchor boxes to optimize proposal generation and classification. Faster R-CNN [194] predicts offsets and confidences based on the introduced anchor boxes. If objects exhibit significant scale and viewpoint variations, occlusions or truncations, the number of region proposals or anchor boxes needed can become a bottleneck in these architectures.

In the context of autonomous driving, the current top algorithms in object detection on the KITTI vision benchmark [91], include Subcategory-aware CNNs (SubCNN) [233] and 3D Object Proposals (3DOP) [40], using efficient proposal generation combined with CNN methods. While [233] combines a subcategory-aware RPN with a detection network for joint detection and subcategory classification, [40] achieves an improvement in 3D object detection by exploiting stereo vision to generate proposals directly in 3D space. Although more efficient proposal methods are used, computation times greater than 1 s reported in [40, 101, 146] make a realtime application in an intelligent vehicle domain without major optimization not possible.

Recently, Single Shot Detection [149] and YOLOv2 [191] showed realtime capabilities and superior results using a set of optimized anchor boxes similar to Faster-RCNN. A fully convolutional approach is used in [50] resulting in position-sensitive score maps avoiding per-region subnetworks.

In the intelligent vehicles domain, multiple sensor modalities are usually applied. Beside mono- and stereo-cameras [17, 109], radar [4, 57, 85] or li-

dar [171, 186] sensors can be used to verify or fuse vision-based detections. Thus, vehicle intervention (braking, steering) does not solely rely on camera-based object detection. Single-frame detections are commonly also integrated over time (tracking) [20, 212] to further improve the detection stability.

Common metrics to measure the detection performance are Reciever Operating Charateristics (ROC, i.e. TPR, FPR) [66], recall (i.e. TPR) and precision [53, 71], and the (log) miss rate over the false positives per image. To express results using one single value, the area under the ROC curve, the mean Average Precision (mAP) [71], or the log-log average miss rate [240] can be used.

2.2 Segmentation

Separating object information from clutter is broadly used in image and video editing. For surveillance and intelligent vehicle applications it sets the basis for a more detailed analysis on the object. Localization accuracy of the whole object or object parts can be greatly improved by an accurate segmentation.

Image segmentation approaches can be roughly divided into data-driven approaches [22, 26, 105, 106, 139, 140, 150, 198, 217, 239], top-down approaches [5, 47, 46, 78, 87, 94], and methods that combine top-down knowledge with data-driven cues [21, 29, 69, 130, 137, 190]. Data driven approaches [22, 26, 139, 140, 198, 217, 239] based on Conditional Random Field (CRF) formulations [26, 139] show promising segmentation results. For example, the GrabCut framework [198] involves segmentation with a minimum of user assistance based on an iterative, pixel-wise Gaussian Mixture Models (GMMs) fitting. In order to generate more discriminative and robust features, [161, 140, 217, 239] aggregate features over oversegmented regions. Superpixel-based features based on dense SIFT (dSIFT), introduced by Bosch *et al* [22], show powerful results on a local level [140, 217]. Boosted Decision Trees (BDT) are used in [217] to classify these features. A preliminary and necessary step for these methods is to cluster the SIFT features to obtain a visual code book representation. This is often done with k-means [217]. As shown in [162], generating visual code books using Decision Trees is more discriminative as using k-means. In [239], superpixel-based features (e.g. surface normals, planarities, distance to camera path) are constructed from a dense depth map and classified using Decision Trees. Recently, deep fully convolutional neural networks (FCN) [150] showed superior performance for scene segmentation. Furthermore, works de-

scribed in [105, 106] show how to combine FCN-like architectures with Faster-RCNN to gain additional instance-based segmentations for each extracted RoI.

The previous discussed methods are ignoring shape information about the object. Taking a segmented pedestrian as an example, with the data-driven methods it is not possible (without further assessment) to directly infer concrete body part locations. Explicit shape knowledge becomes necessary here. Top-down approaches making use of explicit shape knowledge [5, 47, 46, 78, 87, 94] can be distinguished by whether they use global or part-based shape representations. Global shape representations can be discrete, in terms of a Set of Exemplars (i.e. shape templates from a training set). Hierarchical representations can be derived on top of these for an increased matching efficiency [87]. Active Shape Models (ASMs) [47] combine Statistical Shape Models (SSMs) (compact, linear-subspace probabilistic representations) with means to match these to images. Active Appearance Models (AAMs) [46] extend ASMs by capturing shape and texture information jointly. Unlike exemplar-based representations, ASMs and AAMs require feature correspondence. Fitting them to an image can result in suboptimal solutions because of local maxima. Giebel and Gavrila [94] represent shapes by multiple SSMs (MSSMs) to account for different shape aspects (pedestrian feet apart vs. feet closed) and orientations. Part-based representations, like pictorial structures [5, 78] offer a modular representation; tree-structured graphical models can be used for modeling the dependency between parts [5].

In terms of combining object models with data-driven cues [21, 29, 69, 130, 137, 190], a number of approaches [29, 77, 137] are based on GrabCut. Kumar *et al* [137] introduce the Object Category Specific Contrast Dependent Random Field and shows that multiple shape potentials can be used by formulating a linear weighted sum of energies, which makes solution by efficient methods (e.g. Graph Cut) still possible. Non-articulated objects are represented by a set of exemplars, which capture shape and appearance. For articulated objects, a PS model is used. Interaction between parts is additionally modeled by a Markov Random Field (MRF). A layer indicator for each part is added to handle occlusion. Based on multiple views, Bray *et al* [29] describe an iterative pose estimation approach using a stick model joined with data cues in a CRF formulation (similar to [137]). Kokkinos and Maragos [130] describe an EM segmentation framework, in which the segmentation result from the E-step is used to estimate parameters of an AAM in the M-step. Eslami and Williams [69] describe a generative framework based on a Shape Boltzmann Machine combined with

appearance cues. In [190], a Metropolis-Hastings sampler takes the results of an edge-based part detector as starting point to generate proposals of each body part. Learned appearance histograms for all parts are used for segmentation. Bo and Fowlkes [21] use hierarchical decomposition of parts and compute scores of matching part-based mean templates with additional color cues. Latest works explore methods to combine shape priors within a deep neural network architecture [105, 174].

2.3 Orientation Estimation

Object orientation can serve as an important cue to better model or infer the behavior of an object. An extensive amount of research meanwhile exists on articulated pose estimation. In this section, the focus is set on head and body orientation estimation.

Approaches in body part orientation estimation are largely application dependent (see surveys [166] and [110] for head orientation estimation). Applications in Human-Machine Interaction (HMI) [8, 73] or entertainment [154] typically consider high resolution images and cooperative subjects under controlled backgrounds to estimate the body part orientation. Applications in surveillance [16, 38, 145, 176, 195, 213] and in intelligent vehicle [67, 193, 207] domains need to cope with low resolution images, with complex and dynamic backgrounds, and changing lighting conditions.

To cope with these challenges, approaches often use robust lower-level image features like SIFT / HOG [14, 16, 38, 67, 84], Haar [207, 210], Local Receptive Fields (LRFs) [67], Local Binary Patterns (LBP) [193] and distance metrics [14, 73, 176] in combination with different classification schemes (e.g. Support Vector Machines (SVMs) [38, 67, 84, 176, 210], Neural Networks (NNs) [67], Random Regression / Decision Trees or Ferns [14, 16, 73] or Boosting Cascades [207]) to perform orientation estimation.

Such data-driven approaches can be used for both, head and body orientation estimation. Schulz *et al* [207] train a boosting cascade of Haar-like features for eight head orientation classes in an one-versus-all manner. The maximum classifier response over all possible hypotheses (different scales and locations) and the eight orientation classes is selected as the final estimation result for head position and head orientation. Enzweiler and Gavrila [67] use four orientation specific classifiers jointly to give the final pedestrian detection output. The same classifiers are reused to infer a continuous orientation estimate for

the detected pedestrian. Benfold and Reid [14] use a random fern architecture with a combination of HOG and color based features to infer head orientation, after the head is found by a HOG-based head detector. Training is done with eight orientation classes. While the majority of above mentioned methods use manually annotated training data, Benfold and Reid [16] learn head orientations unsupervised by using the output of a tracking system [15], supposing that head orientation is dependent on walking direction. The walking direction can also be used as a proxy for body orientation (e.g. [195]), thus assuming that people only move forward. Recently, deep neural networks have also been used for estimating orientations of common objects in traffic scenarios [28, 40, 214, 219]. Beyer *et al* [18] show a continuous regression using a Biternion model in the loss function, regressing fine-grained orientation angles for head pose estimation in the surveillance domain. The Biternion formulation was recently also used by Braun *et al* [28] to infer detection and orientation jointly within a deep neural network.

Model-driven strategies make use of explicit shape knowledge and represent a possible extension to pure data-driven approaches. Orientation information of body parts can often directly be inferred from the applied shape model. In particular, ASMs and AAMs introduced by Cootes [46, 47] are often used for inferring head or body orientation estimation. The idea of multiple linear subspaces (i.e. MSSMs [94]) is also adopted by Lee and Kriegman [141]. They use the method of Hall *et al* [102, 103] and apply an incremental online update of multiple linear-subspace models, each representing a face orientation. Zhu [242] uses a mixture of trees to infer face detection and orientation estimation. The trees share a pool of facial landmarks and use global mixtures (similar to AAM) to capture topological changes due to viewpoint. While there is a limited ability of using accurate shape information for the head with decreasing resolution, body orientation estimation can exploit, even in lower resolution images, prior knowledge about the body shape by matching shape models (e.g. [88]).

To improve orientation estimations, [16, 37, 38, 193, 195, 213, 241] introduce constraints which set head and body orientation into relation of each other. Chen and Odobez [38] use such constraints directly during classifier training, while Zhao *et al* [241] use body orientation information to differentiate online between opposite head directions. Smith *et al* [213] constrain the head location with respect to the body location to obtain physically possible configuration. Benfold and Reid [16] apply a Conditional Random Field (CRF) for modeling the interaction between the head orientation, walking direction and appearance

to recover gaze direction. While Robertson and Reid [195] constrain the head orientation on the velocity direction, Chen *et al* [37] introduce a coupling between head and body orientation and between body orientation and velocity direction. The constraints are modeled by Von Mises distributions. Rehder *et al* [193] jointly estimate head, body and motion direction with a logistic regression model and marginalizes over possible configurations to gain a coupled head orientation estimate. Recently, Cao *et al* [34] achieved accurate pose estimations using deep neural networks. Body part orientations and anatomical constraints are implicitly encoded.

By tracking, single frame orientation estimations can be smoothed and results can be further improved [8, 9, 37, 49, 84, 151, 195, 210, 213, 237]. A simple approach is to choose the most frequent direction over a fixed number of frames [210]. More sophisticated models use for example Hidden Markov Model (HMM) [84, 193] or particle filter frameworks [8, 9, 37, 195, 213] to keep track of a body part orientation distribution over time. Constraints between body parts can then be modeled efficiently within the used dynamic model as in [37, 195]. Smith [213] uses a Reversible-Jump Markov Chain Monte Carlo (RJMCMC) sampling scheme for particle filtering to handle a large state space consisting of inter-person (multi person tracking) and intra-person (localization between head and body) interactions.

Finally, there has been extensive work done on articulated 3D body pose recovery, e.g. see surveys [86, 160]. These typically require multiple cameras and / or are computationally intensive and still have issues with robustness.

2.4 System Integration

Predicting the motion of objects is a key challenge for various intelligent systems applied in surveillance and mobile robotics. Intelligent vehicles need to perform a reliable prediction of all moving objects on the street to react and drive safely.

Unfortunately, VRUs can change their motion in a whim and therefore the movement can be highly non-linear. Greater changes in motion types can hardly be handled by a simple single motion model (e.g. constant velocity or constant position models [20]) as commonly used with Kalman Filters (KF) [20] or Extended or Unscented KFs [157]. For multiple motion types, a Switching Linear Dynamic System (SLDS) [20, 196] applies multiple LDS in parallel and a Markov chain is used to switch between the various dynamics of each LDS.

This technique has been already applied for path prediction in the intelligent vehicle domain [123, 205]. Exact solution is intractable and needs to be approximated e.g. by sampling with Markov Chain Monte Carlo method [172, 196] or applying Assumed Density Filtering (ADF) [19, 159]. In an SLDS, a dynamic switch is only happening if there is enough evidence by the positional observations, which contradict the current dynamic model. Instead of waiting until the dynamic of an object is actually changing, one can also try to anticipate the dynamic switch. To anticipates the pedestrian behavior, agent models have been successfully applied [12, 125].

With the help of computer vision, various additional context cues can be extracted to support the behavior estimation [31, 133, 192]. Kooij *et al* [133] divide context cues into static environment, dynamic environment and object context cues. Static environment context cues can be used to model the influence of infrastructure elements on the VRU behavior. For instance, different motion dynamics can be observed at specific locations in common traffic environments [134]. The work presented in [125, 192] implicitly learns the influence of detected semantic regions on the object behavior.

Dynamic environment context cues model the influence of other moving objects in the nearby environment. The Time-To-Collision (TTC) [124], i.e. the time to a potential collision between two objects keeping their course and speed, is a simple example for such a cue. More complex social force models [6, 108] can be used to model the interactions between multiple VRUs.

Object context cues provide insights on VRU intention or VRU awareness. For instance, [104, 204] show the importance of head motion as an important indicator for the future pedestrian behavior and the potential for an improved intention prediction [127]. Kooij *et al* [133] suggest to model the awareness of a pedestrian (i.e. object context cue) over the head orientation. Brouwer *et al* [31] model the probability for a certain moving direction dependent on the time the pedestrian is looking in this direction. Recently, deep CNNs have been used to infer detailed pose and skeleton models [34], which can be used to model further object context cues.

Using the extracted features, context cue uncertainty can be directly incorporated into a probabilistic model [10, 123, 133, 208]. Ba and Odobez [10] combine context cues in a Dynamic Bayesian Network (DBN) [164] to model influence of group interaction on the visual focus of attention. Kooij *et al* [133] combine the object context (i.e. the awareness of the pedestrian) with the environment context cues within a DBN formulation, leading to an improved path

prediction in the relevant short term horizon up to 1 s. Brouwer *et al* [31] fuse different motion models (incorporating object and environment context) within a probability grid to improve the collision probability estimation. Also non-parametric models have been studied to model if a crossing pedestrian stops at the sidewalk or not [123].

In contrast to these methods, behavior prediction can be also treated as classification problem. The work presented in [129, 189] considers action classification for determining if a standing pedestrian starts to walk or not. Fang *et al* [74] use skeleton estimates [34] to estimate pedestrian's motion state. Poppe [185] provides an extensive survey over human action classification. Recently, R-CNN architectures have been also applied to jointly detect and classify human actions [99, 98] and Lo Presti and La Cascia [187] have shown how 3D skeletons can be used to robustly estimate human actions over time.

Vehicle pilot systems with conditional automation (level 3 [201]) are nowadays already available for the end-user. Tesla was one of the first vehicle manufacturers claiming to have capabilities for a self-driving vehicle available [173]. Not all functions here are already implemented and a few accidents [175, 202] have proven that there is still some way to go. Especially when VRUs are involved, methods seem to be still not mature enough to handle all critical situations occurring in inner cities [143]. Currently there can be only a rough guess on how VRU object motion is modeled and predicted within the self-driving vehicles of e.g. Tesla, Uber and Waymo [148]. No concrete implementation details have been presented to the public yet.

2.5 Label and Data Management

Building a large labeled dataset with a diverse set of objects is costly and challenging. Problems here occur in handling large amounts of data, specifying and keeping the labeling quality high and the labeling process efficient. In the following the focus is set again on camera sensors and therefore on image-based label data. Although a few tools support also the labeling of other sensor modalities (e.g. radar, lidar) [43, 183] this is out of scope of this work.

Multiple freely available label tools are focusing on tag-based bounding box labeling [158, 200, 218, 220]. An object is annotated by drawing an enclosing rectangle and tags (class or attribute) can be assigned to the object. Each bounding box gets an unique instance id. Furthermore, there exist also tools supporting semantic image segmentation [48, 114, 126, 200]. The LabelMe annotation

tool [200] is a browser-based annotation tool focused on polygon-based segmentation labeling. Once the polygon is closed, a user can enter an object class for the object. Similar, the Cityscapes label tool [48] represents a simple and efficient python-based segmentation labeling tool. The tool was used for creating the Cityscapes dataset [48].

Proprietary annotation tools [2, 43, 183] provide a richer set of ADAS-specific annotation facilities. The Philosys label editor [183] supports the ADTF file format [224]. The frontend provides multiple views on the data, separated by a video view for fast video playback, a sequence-based view for object annotation over time and an object view for a detailed modification of the geometrical object labels. A few tools [43, 158, 183] provide also semi-automated methods to support the labeling process or to perform a pre-labeling by automatic detection and tracking using state-of-the-art machine learning algorithms. C.Label [43] and LabelBox [138] provide ideas of how to assess Key Performance Indicators (KPIs) for a supervision of the performed label process and the label quality.

Beside the labeling process itself, it is important to store the annotations in an efficient way and making them, along a link to the raw sensor data, available for other applications (e.g. machine learning applications). A database management of label data [138, 200] is advantageous over simple file-based label formats [2, 43, 48, 114, 126, 183]. Version control mechanisms become easier possible and data integrity can be better preserved. By a database system, a scalable and better collaboration of the labeling process becomes possible, e.g. by integrating a label task management system [138].

With the increased success of deep learning methods there is a need for companies providing large-scale commercial and customized labeling services for getting a greater amount of labeled data in a reasonable time. Those companies either use private labeling farms, having many experienced labelers available on-premise, or they use additional crowd services. The idea of the latter is to distribute the label task to potential non-experts, making use of distributed computing and modern web frontends. One of the best known services used here is Amazon Mechanical Turk [32, 107]. Label tasks can be also integrated in other services (e.g. gaming services) [116]. The user of the service is often rewarded (e.g. granted access to a specific feature in a game or website) when fulfilling a label task. Making use of crowd services for label tasks requires a carefully designed label process which brakes down a complicated task in multiple easy steps so that a non-expert user can understand what to do. The quality of the

final label is strongly dependent on this process design and on the intuitive user interface. If implemented well, such concepts can provide a fast, scalable and cost-effective label process [120]. Unfortunately a crowd-based approach is difficult to apply for the development of ADAS systems because of data privacy and data protection issues.

2.6 Datasets

Challenging datasets have promoted technological progress in computer vision. They are especially useful for supervised learning of object categories, localization and segmentation. Major progress in the development of those methods has been reached by making use of publicly available, large-scale image datasets such as ImageNet [199], PASCAL VOC [71], PASCAL-Context [163] and Microsoft COCO [147] that allow machine learning methods to develop their full potential.

Also the research progress in the intelligent vehicle domain can be heavily linked to the existence of various datasets, in particular the KITTI Vision Benchmark Suite [90], CamVid [30], Leuven [142], Daimler Urban Segmentation [203] and the recently published Cityscapes [48] and Mapillary dataset [168]. Those datasets have been recorded on-board a moving vehicle. Although such domain tailored datasets are often smaller than datasets addressing more general settings, they are capturing multiple important aspects for the development of autonomous vehicles. While KITTI and Cityscapes datasets provide additional object instance information (i.e. which pixel or bounding box belongs to which object), there are even more specific public datasets promoting the method development with the focus on VRU instance detection, such as the INRIA [52], Caltech [61] or the Daimler [66, 122] pedestrian detection datasets. The Penn-Fudan dataset [226] was created for instance segmentation of pedestrians. The recently published CityPersons dataset [240] was published in addition to the Cityscapes dataset [48], providing a rich and diverse set of bounding box annotated VRUs throughout Germany.

Although cyclists are often encountered in traffic accidents, especially also in developing countries, there was no challenging cyclist dataset publicly available. In all the above mentioned datasets cyclist instances are excluded, ignored or underrepresented - i.e. 0 cyclists in Caltech [61], only around 3300 cyclists in KITTI [90] (estimated number based on published label information) and 3454 cyclists in the CityPersons [240] dataset.

3

VULNERABLE ROAD USER DETECTION

The detection of objects in images is a key task in a number of important application domains, such as intelligent vehicles, surveillance, and robotics, and is a crucial preliminary step for a more detailed object analysis. Only by knowing if there is an object of interest (detection), we can pay further attention to this object and finally react on it. And, only by knowing what kind of object it is (classification), we can deploy class-specific prior knowledge, which can contribute to reach a better object understanding. Due to its relevance and challenges, person detection often features as canonical task to assess the performance of generic object detectors.

3.1 Introduction

An efficient bounding box based representation is often applied to detect and classify VRUs in the image. Figure 3.1 visualizes the available information after a successful detection process on a common street scene. VRUs are due to their wide variation in appearance, arising from articulated pose, clothing, background and visibility conditions (time of day, weather), especially challenging

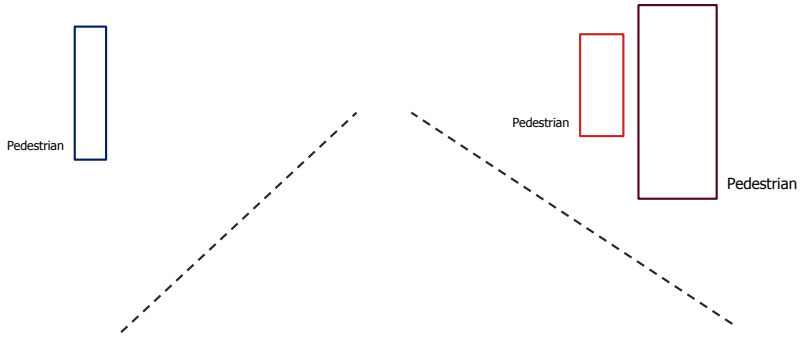


Figure 3.1: Bounding box based detection (and classification). The simple box representation will be extended by subsequent modules (e.g. by object segmentation, orientation estimation) to allow a better VRU analysis.

for object detection. Therefore, sophisticated feature representations are necessary to perform a robust VRU detection and classification.

Handcrafted feature representation like the Aggregated Channel Features (ACF) [58, 236] or the Locally Decorrelated Channel Features (LDCF) [167] showed good performance for various object detection tasks. Features are aggregated from gradient magnitudes, gradient histograms and color channels. Compared to ACF, LDCF additionally removes correlations in local neighborhoods of feature channels [167]. Boosting [83] is used to learn decision trees over these features to distinguish object from background.

Recently, region-based CNNs (R-CNNs) [95, 97] showed superior results for object detection, being able to capture complex context information by using a powerful multi-layer feature representation. For R-CNNs the detection performance depends strongly on the used proposal method providing Region of Interest (RoI) candidates for classification. The recall [53] (i.e. the True Positive Rate) of the proposal method specifies an upper bound for the overall detection performance. The problem even gets tougher if object classes appear in various viewpoints and RoI proposals need to cover this variations. Selective Search (SS) [221] and Edge Box (EB) [243] are commonly used 2D data-driven proposal methods. The grouping method Selective Search greed-

ily merges superpixels to generate RoI proposals and was already used successfully within the Fast R-CNN (FRCN) architecture [95]. The faster window scoring method EdgeBox uses Structured Decision Forests [60] to score the object boundary from initial sliding window boxes and outperforms [112] other proposal methods on PASCAL VOC [71] dataset when applied in FRCN [112]. Recent works [40, 68, 101, 146] show that the detection performance can be improved by making use of 3D RoI proposals from lidar or stereo data to better estimate object extensions within multiple viewpoints.

This chapter focuses on the challenging task of cyclist detection. Nevertheless, the discussed methods can be also applied to other or multiple classes with minor parameter adaptions. Cyclists appear with a large viewpoint variation, i.e. a cyclist from the back looks completely different to a cyclist from the side. To better distinguish cyclists to e.g. pedestrians or other riders, the cyclist object should contain the bike. By this, the variation in viewpoint even increases (i.e. different object aspect ratios for different viewpoints) and there is a strong need to handle the different viewpoints in the detection process.

Inspired by the good results achieved in [68] the stixel world [182] is used to produce a set of RoI proposals from each stixel for a robust VRU detection covering large viewpoint variations. The RoI proposals are applied within the FRCN architecture to gain competitive detection results. Additionally, the FRCN configurations are compared to ACF- and LDCF-based detector ensembles [144].

A fundamental prerequisite for developing a robust detection and classification algorithm is data. Unfortunately in all available datasets cyclist instances are excluded, ignored or underrepresented - i.e. only around 3300 cyclists in KITTI [90] and around 3454 cyclists in the recently appeared CityPersons [240] dataset. Although cyclists are often encountered in traffic accidents, especially in developing countries, there has been no challenging cyclist datasets publicly available yet. Therefore, the TDC dataset is introduced, being the first intelligent vehicle dataset focusing on vision-based cyclist detection. Furthermore, it is also the first intelligent vehicle dataset covering realistic and challenging VRU scenarios recorded in urban districts in China. A detailed overview of this dataset is given in this chapter. With $22{,}161$ manually labeled cyclist instances, the introduced dataset contains nearly an order of magnitude more cyclists than available datasets.

The main contributions of this chapter can be summarized as:

- A new detection architecture (called SP-FRCN) is introduced making

use of efficient stixel-based RoI proposals [68] to capture various object scales and different viewpoints. The RoI proposals are used within the FRCN architecture [95].

- Overall competitive detection results are shown on a challenging and newly introduced dataset for cyclist detection benchmarking. Compared to other state-of-the-art automotive datasets [90, 240] the published dataset contains nearly an order of magnitude more labeled cyclists objects. It is also the first Chinese automotive dataset recorded in an urban area with a reasonable size.

- Detection performance of SP-FRCN is further compared to FRCN architectures applying well-known 2D proposal methods [221, 243], and to detector ensembles (as developed in [144]) applying handcrafted feature representations [167, 236]. Results on the new dataset show that SP-FRCN is outperforming the other methods for cyclist detection.

This chapter is based on the work presented in [144].

3.2 Stixel-based Cyclist Detection

To achieve a robust cyclist detection the Fast R-CNN (FRCN) [95] architecture is used. But instead of using Selective Search [221] as proposed in the original paper [95], stixel-based RoI proposals are maped to the computed feature maps and RoI pooling is applied on these feature maps. Figure 3.2 shows the final system architecture based on FRCN. In the following this new architecture is called SP-FRCN. For feature extraction, different base networks (e.g. ZF [238], VGG [211], GoogleNet [216]) can be applied.

The stixel world described in [181, 182] is a disparity-based superpixel segmentation, where each superpixel is a thin vertical segment (stick) with an associated class label (i.e. ground or object) and a 3D planar depth model (i.e. distance and height). The width of a stixel is fixed to a certain size (e.g. five pixels). The stixel generation is based on the assumption that both, ground and object class are planar surfaces. While object segments are modeled with one disparity value, the disparity of a ground segment is modeled with a simple linear function. The segmentation problem can be solved efficiently by dynamic programming. Hence stixels are an efficient and sparse representation of objects

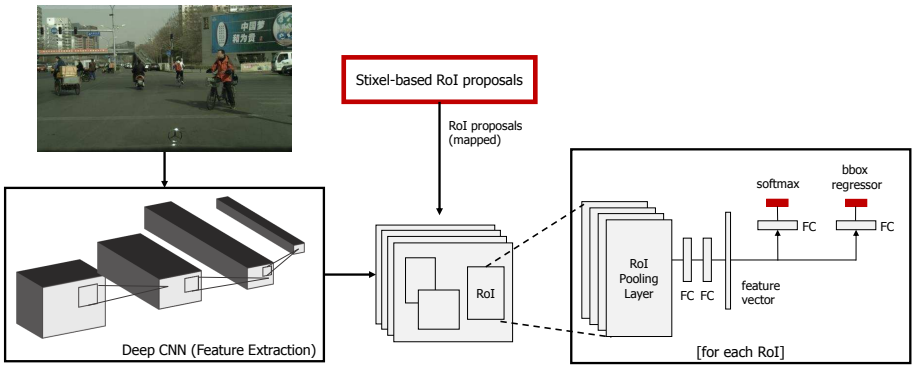


Figure 3.2: Fast R-CNN version [95] with stixel-based RoI proposals. RoI feature extraction is done via the RoI pooling layer on the feature maps. Compared to the original R-CNN version [97], where RoI feature extraction is done on the input image, this saves time because expensive scaling and feature computation is only done once.

having approximately vertical surfaces like VRUs and vehicles. For the following experiments the width of the stixels is fixed to five pixels for the energy minimization.

A priori knowledge about the size of the objects of interest (i.e. the cyclist) is used to filter stixel candidates for proposal generation. Let a stixel be described in the vehicle coordinate system with its lateral, longitudinal position and height $[x_c, z_c, h_c]$. For each stixel in the range $z_c \in [4\,\mathrm{m}, 100\,\mathrm{m}]$ and $h_c \in [1.2\,\mathrm{m}, 2.4\,\mathrm{m}]$ different proposal aspect ratios (width / height) are sampled with a proposal height of $h_p = h_c$. For stixels with $h_c > 2.4\,\mathrm{m}$, the proposal height is varied from $h_p = 1.2\,\mathrm{m}$ to $h_p = 2.4\,\mathrm{m}$ with a step size of $0.3\,\mathrm{m}$. The latter is a workaround to handle oversegmentation errors of the stixel world. For VRUs this can happen when a VRU is walking close to a wall and a segment cut between the wall and the VRU is not established in the optimization. The proposal set is afterwards augmented by an additional jittering. For each stixel-based proposal four additional proposals are sampled in the surrounding. Therefore, the position of the original proposal is adapted by $10\,\%$ of the height to the left, right, top and bottom. Figure 3.3 shows the different steps resulting in a robust proposal extraction based on stixels. Figure 3.4 visualizes the proposal generation based on a single filtered stixel.

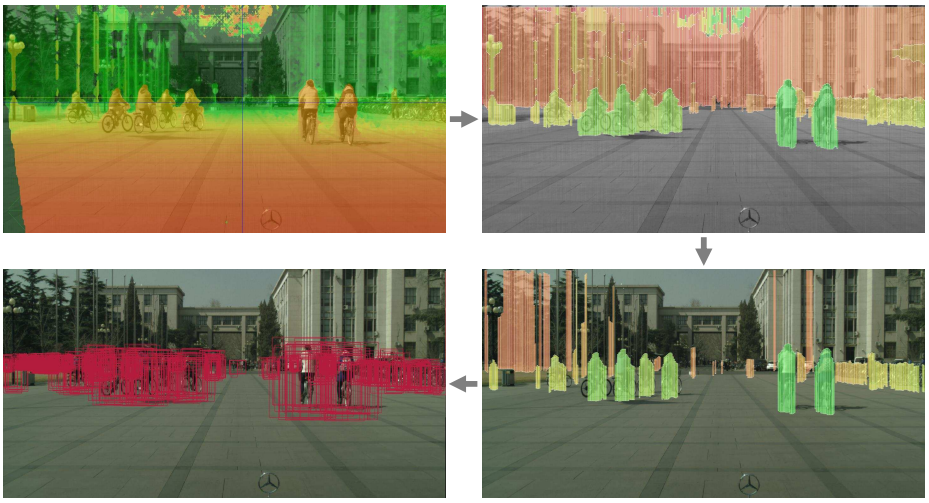


Figure 3.3: Stixel-based proposal generation. Example on an image from the TDC dataset showing disparity map (top left), stixel world solution (top right), filtered stixels based on a priori object knowledge (bottom right) and extracted proposals based on the filtered stixels (bottom left).

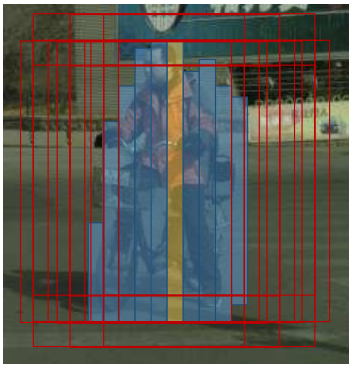


Figure 3.4: Stixel-based bounding box proposals generated for different aspect ratios based on one filtered stixel. Proposals are additionally jittered to compensate stixel and road estimation errors.

3.3 Tsinghua-Daimler Cyclist Benchmark Dataset

About 21,200 cyclists die in China every year [232]. Due to the lack of information in the available accident statistics, the true fatality rate is expected to be even significantly higher [232]. In Beijing alone there are around seven million registered cyclists making this city a suitable candidate for a more detailed assessment. Despite the fact that cyclists are often encountered in traffic accidents especially in developing countries, there existed no challenging cyclist dataset publicly available yet. Therefore, a richly annotated cyclist dataset was created for training and evaluating cyclist detection and classification methods.

Because of the time consuming and costly labeling process, the dataset publication was splitted into two dataset releases, TDC-v1 [144] and TDC-v2 (planned for Q2/2019). The aim of TDC-v1 is to provide a common point of reference for cyclist detection evaluation. In total, the already published dataset contains annotated bounding boxes of 22,161 cyclists and 10,200 other VRU objects (i.e. pedestrians and other riders like tri-cyclists, motor-cyclists, moped-riders etc.) from 14,674 labeled images, varying widely in appearance, pose, scale, occlusion and viewpoint. The dataset includes left color images, vehicle state, camera information and corresponding disparity images, which were recorded from a vehicle-mounted stereo vision camera. TDC-v2 is an extension of TDC-v1 and will handle the VRU in general. It will also focus on other riders and pedestrians. More images, labels and further object attributes (e.g. body part labelings and orientation) will be added.

3.3.1 Data Collection

Approximately six recording hours were collected from a vehicle-mounted stereo vision camera (image resolution of 2048×1024 pixel, baseline of 20 cm) at 20 fps driving through regular urban Chinese traffic during 5 different days. The recordings were performed in the northern city parts of Beijing, which were chosen for their relatively high concentration of cyclists and pedestrians. In particular, recordings took place in Haidian and Chaoyang District. Besides the images and the disparity maps, vehicle information including velocity, longitudinal acceleration, yaw rate, was captured concurrently to offer useful vehicle information for different research tasks.

3.3.2 Labeling Details

For TDC-v1 / TDC-v2, 14,674 / 19,230 frames from more than five million recorded images have been manually labeled, containing 32,361 / 150,000[1] labeled VRU objects, including cyclists (22,161 / 38,200[1]), pedestrians (8919 / 101,200[1]) and other riders (1281 / 10,400[1]). TDC-v1 contains already an order of magnitude more cyclists than other available computer vision datasets from the intelligent vehicles domain. See Table 3.1.

	Caltech [61]	KITTI [90]	CityPersons [240]	TDC-v1 [144]	TDC-v2 (planned)
# images	249,000	14,999	5000	14,674	19,230
# cyclists	0	3300[1]	3454[2]	22,161	38,200[3]
resolution	640×480	1240×376	2048×1024	2048×1024	2048×1024

[1] Total estimate based on published label data.
[2] Rider class. No separate cyclist class information available.
[3] Estimated based on the current labeling process.

Table 3.1: Number of labeled cyclist objects for different available datasets.

Table 3.2 summarizes the labeling statistics. In the fully labeled (f.l.) dataset parts (i.e. Set-2, Set-4) all VRUs higher than 20 pixels, not occluded or truncated by more than 80% were annotated. A bounding box indicating the extent of the pedestrian or the rider including the vehicle is annotated. The bounding box does include worn clothes of the VRU, but excludes bags and other accessories.

To gain more cyclist samples with the available resources, Set-1a was annotated only partly (p.l.). In this dataset part, only "ideal" cyclists (including the bike) were annotated with a tight bounding box. "Non-ideal" cyclists, which are lower than 60 pixels, occluded or truncated by more than $10\,\%$ were not labeled, and neither were pedestrians or other riders. TDC-v2 will extend Set-1a to the fully labeled version of Set-1a. Furthermore, a non-public test set (only images are given out) will be added in TDC-v2 to facilitate a fair benchmarking procedure.

From the recorded sequences, every 10th frame was labeled for Set-1a and every 5th frame was labeled for Set-2 and Set-4. A higher sampling rate is chosen here to allow also the evaluation of tracking algorithms in future work. Additionally, 1000 images were extracted without any VRUs to supplement the

[1] Estimated based on the current labeling process.

	Training set			Test set	Total
	Set-1a (p.l.)	Set-2 (f.l.)	Set-3 (non-VRU)	Set-4 (f.l.)	
# Images (labeled)	9741	1019	1000	2914	14,674
# VRU Objects	16,202	3016	0	13,143	32,361
# Cyclists	16,202	1301	0	4658	22,161
# Pedestrians	0	1539	0	7380	8919
# Other Riders	0	176	0	1105	1281

Table 3.2: Overview of the different dataset parts of the Tsinghua-Daimler Cyclist Benchmark Dataset version 1 (TDC-v1) [144] showing the number of objects for available classes. Test and training parts are uncorrelated. Set-1a is partly labeled (p.l.) in TDC-v1 and will be extended to a fully labeled version in TDC-v2. Furthermore, a non-public test set will be added in TDC-v2.

training dataset, termed Set-3 (non-VRU). Test and train splits are not correlated and have been recorded at different locations. In addition to the class of each bounding box, different attributes were added to objects and images. Occlusion / Truncation levels as well as image tags are already available in TDC-v1 which will be extended by head / body / vehicle bounding boxes, orientations, track IDs and explicit ignore and group regions in TDC-v2. Table 3.3 shows an overview of the additionally labeled attributes. Figure 3.5 shows some labeled image examples from the published TDC-v1 dataset. Figure 3.6 shows the distribution of bounding box heights and aspect ratios on the fully labeled subsets (Set-2 and Set-4). Compared to the aspect ratio of pedestrians (cf. [61]), the aspect ratios of cyclists (in this case the cyclist contains a bike and a rider) have a larger variation. This needs to be taken into account for proposal generation. Furthermore, around 86.6% of the cyclists are not occluded / truncated, 9.9% are partially and 3.5% of the cyclists are heavily occluded / truncated. Details on the labeling process and mechanisms used to create this dataset are further provided in Chapter 7.

Labeled Attribute	Description	Dataset version
Occlusion	Bounding box occlusion is labeled into three levels. *No Occl.*: BB-occlusion less than 10%; *Partial Occl.*: BB-occlusion between 10-40%; *Heavy Occl.*: BB-occlusion between 41-80%;	v1 [144]
Truncation	Bounding box is truncated. Three levels (same as for occlusion) are used: *No / Partial / Heavy Truncation*.	v1 [144]
Head / Vehicle Bounding Box	Bounding box covering the head of the VRU including hair, excluding clothes; Separate bounding box covering only the vehicle (e.g. bike, motorcycle, moped, tricycle) of the rider.	v2
Orientation	Continuous head orientation for rider and pedestrian; continuous body orientation for pedestrian; continuous vehicle orientation for the vehicle of a rider.	v2
Track ID	Unique ID per person instance over time.	v2
Image tags	Image tags (lens flare, foggy, rain, motion blur) available for each image if applicable.	v1 [144]
Ignore regions	Explicit ignore regions defining areas where the labeler cannot tell if an object of interest is present or absent. Treating such regions as background has a negative effect on training [240].	v2
Groups	Class-specific object groups are used for image regions with small but still identifiable objects.	v2

Table 3.3: Additionally labeled attributes for TDC-v1 [144] and TDC-v2.

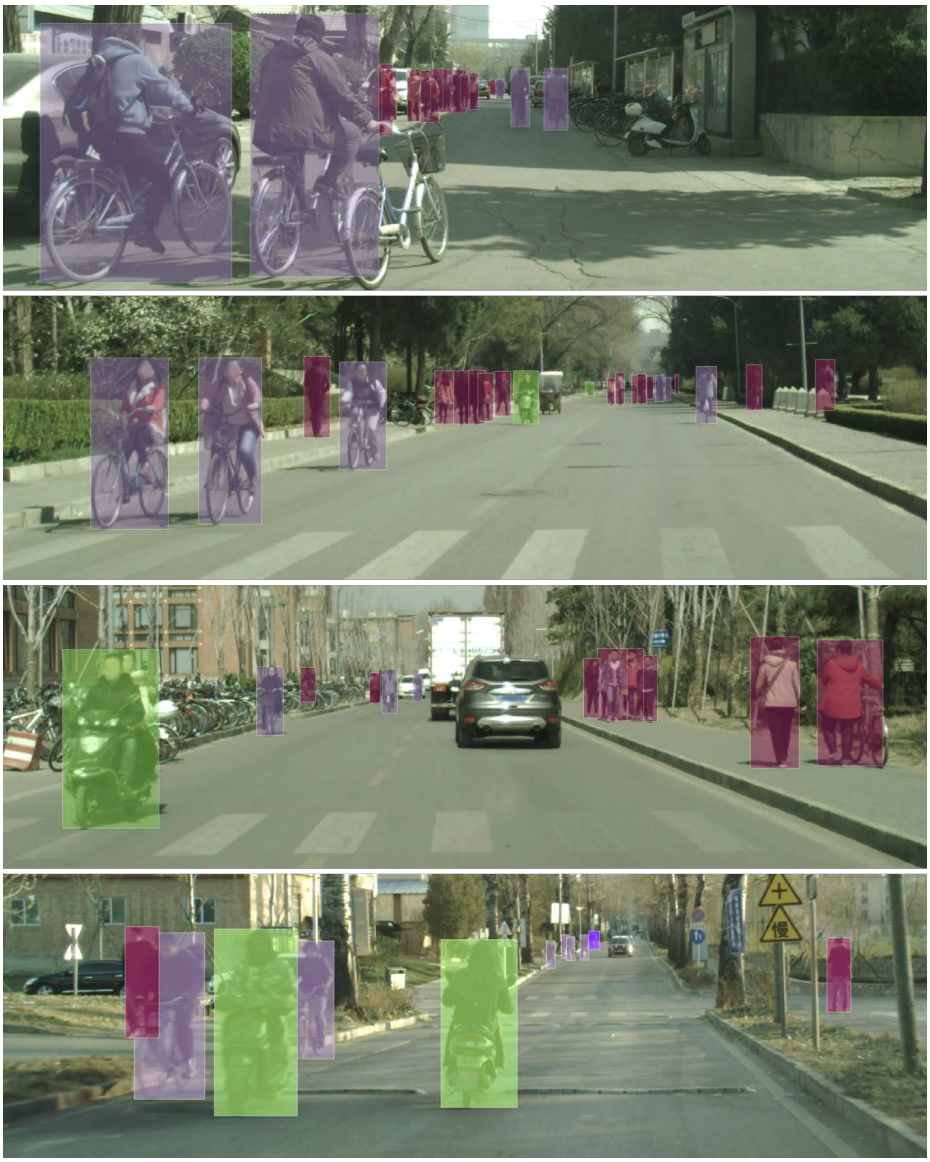
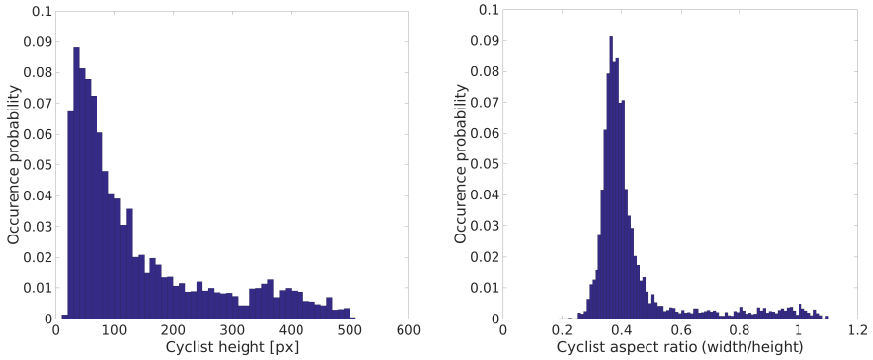


Figure 3.5: Image examples from the published TDC-v1 dataset (Set-4) with labeled bounding boxes for cyclists (purple), pedestrians (red) and other riders (green).



(a) Bounding box heights (b) Bounding box aspect ratios

Figure 3.6: Label statistics over the fully labeled dataset (i.e. Set-2, Set-4) showing (a) bounding box heights and (b) bounding box aspect ratios (width / height).

3.4 Experiments

All experiments described in this section are performed on the already introduced dataset parts of TDC-v1. Set-1a, Set-2 and Set-3 are used for training and Set-4 for testing. For evaluation, metrics proposed in the PASCAL object detection challenges [71] are used, showing the relationship between precision and recall rate. To summarize the performance of precision / recall curves, the Average Precision (AP) is applied as described in [71]. A bounding box detection is assigned to a ground truth bounding box object if the Intersection over Union (IoU) ratio exceeds a threshold of 0.5.

For a more detailed evaluation, three test subsets (using Set-4) are specified with different difficulty levels:

- Easy (2656 cyclists): cyclists with bounding boxes higher than 60 pixels with no occlusion and truncation.
- Moderate (3527 cyclists): cyclists with bounding boxes higher than 45 pixels with no / partial occlusion and truncation.
- Hard (4338 cyclists): cyclists with bounding boxes higher than 30 pixels with no / partial / heavy occlusion and truncation.

During evaluation, objects not included in the used test subset are ignored. Beside that, the ability of the detectors to distinguish cyclists from semantically similar neighboring classes (i.e. pedestrians or other riders) is evaluated. Some applications might require their precise distinction (*enforce*) whereas others might not (*ignore*). In the latter case, during matching correct / false detections are not credited / penalized. Neighboring classes are treated as background in training, i.e. no multi-class training was applied.

In Section 3.4.1, different proposal configurations are identified and evaluated in isolation. In Section 3.4.2 selected proposal configurations are applied in the FRCN architecture and the overall detection performance is evaluated and compared to other methods.

3.4.1 Evaluation of Stixel-based Proposals

As described in Section 3.2, multiple proposals with different proposal aspect ratios (width / height) are sampled for each stixel based on the stixel height. To evaluate appropriate proposal aspect ratio configurations, aspect ratios are generated over a range of $[0.1, 1.1]$ using a step size of 0.05. Different configurations are evaluated on the fully labeled training subset (Set-2). In order to compare the configurations quantitatively, the proposal recall is computed for different IoU ratios (0.4 to 1.0, with step size 0.025) on the ground truth bounding boxes. The Average Recall (AR) [112] is computed within this range to gain a single performance value. Starting with the largest possible set of aspect ratios (i.e. 20 distinct aspect ratios), the aspect ratio with the lowest effect on the AR is removed from the set in each iteration. By repeating this for each reduced aspect ratio set, the top 1-20 aspect ratios are identified. The top one (SP-1), three (SP-3), five (SP-5) and seven (SP-7) aspect ratios are then selected for further evaluation. Figure 3.7 shows the number of images that resulted in particular proposal counts for these four different configurations.

The four stixel-based proposal configurations (SP-1, SP-3, SP-5, SP-7) are also compared to Edge Box (EB) and Selective Search (SS) proposals, which are utilized with default parameters. A fixed proposal count of 2000 and 4000 is set for the EB configurations, i.e. EB-2k and EB-4k. Selective Search proposals are generated (using fast mode) on downscaled images with an image width of 1024. This results in a min / max / average proposal count of 1631 / 7098 / 3863 on the test set (Set-4) for the first configuration (SS-4k). For the second Selective Search configuration (SS-2k), the proposal count is reduced to an average number of 2000 proposals based on the proposal confidence [221].

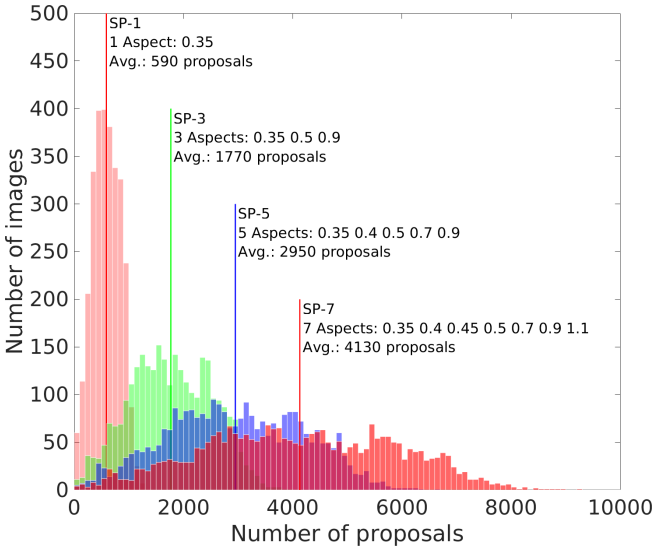
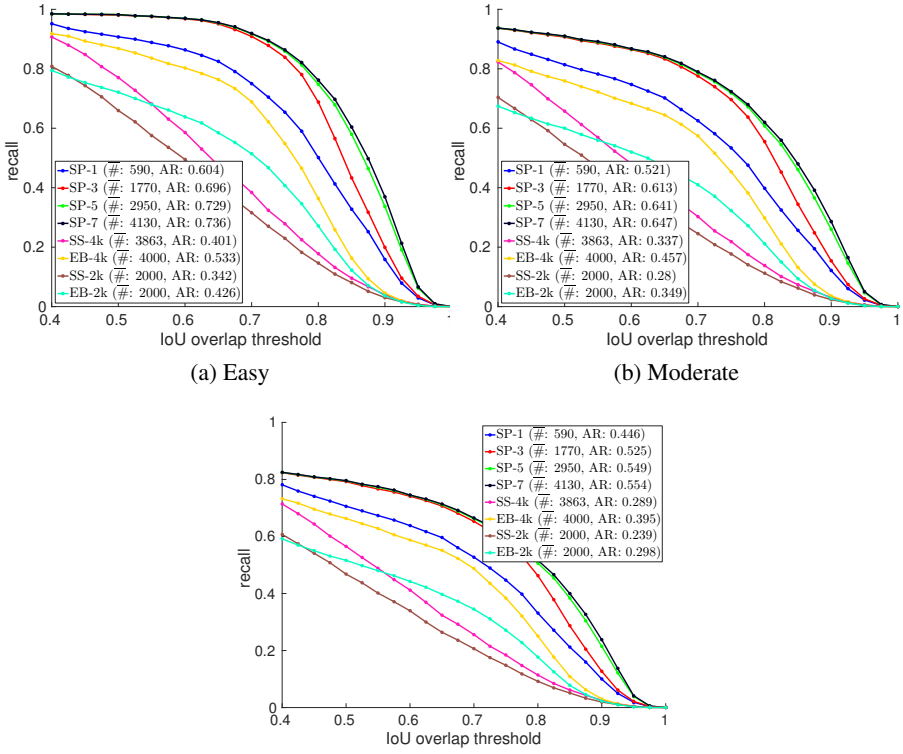


Figure 3.7: Distribution of the proposal count for different configurations. Four different stixel-based proposal configurations with the top 1 (SP-1), 3 (SP-3), 5 (SP-5) and 7 (SP-7) aspect ratios are shown. The number of stixel-based proposals dynamically adapts to the complexity of the image scene.

Figure 3.8 shows that stixel-based proposals provide high recalls for all evaluated IoU ratios on the test set (Set-4). With only 1770 proposals on average (SP-3), a recall of 98.5%, 93.7% and 82.4% is achieved for the easy, moderate and hard test subset respectively with an IoU ratio of 0.5. The stixel-based proposal method outperforms EB and SS proposals, suggesting that the stixel-based proposal generation is also an ideal candidate for region-based neural networks like FRCN. For SP-5 and SP-7, a better recall can be observed only for IoU ratios greater than 0.65, while the recall between 0.40 and 0.65 does not change significantly anymore compared to SP-3.

The computation of the stixel-based proposals including an optimized stixel world estimation [181] runs with $0.03\,\mathrm{s}$ per image on average. In comparison to EB proposals ($0.4\,\mathrm{s}$) and SS ($4.6\,\mathrm{s}$), this confirms the efficiency of the stixel-based proposal method.



(a) Easy

(b) Moderate

(c) Hard

Figure 3.8: Proposal recall for differents IoU ratios, shown for (a) easy, (b) moderate and (c) hard subset of Set-4. SP, EB and SS indicate stixel-based proposals, Edge Boxes and Selective Search proposals respectively. The average number of proposals ($\overline{\#}$) and the Average Recall (AR) is also listed following each methods name.

3.4.2 Cyclist Detection Performance

Results are presented on Fast R-CNN (FRCN) architectures using the evaluated proposal methods. Stixel-based proposal configurations (SP-X) are applied with VGG-16 [211] (SP-X-FRCN-VGG) and ZF [238] networks (SP-X-FRCN-ZF). EB and SS propsals are applied with VGG-16 using a maximum of 4000 proposals on average (i.e. EB-4k-FRCN-VGG and SS-4k-FRCN-VGG). Additionally a comparison to ACF / LDCF-based detector ensembles, as proposed in [144], is done.

Training For training the FRCN methods, the open source Matlab version of FRCN [95] is used. As in the original version described in [95], initial network models (VGG-16 and ZF) pretrained on the ImageNet dataset [199], are applied. For finetuning the networks, final sibling layers are adapted to model the two class problem (i.e. cyclist versus background). Each mini-batch is constructed from 2 images. Images are scaled down to 1300×650 due to limited graphics memory. The first image is chosen randomly from Set-1a, the second image is chosen randomly from Set-2 or Set-3. Mini-batches of size 32 are used, sampling positive proposal samples (max. $40\,\%$ of mini-batch size) from the images of Set-1a or Set-2 with an IoU ratio greater than 0.5 to a ground-truth bounding box. Negative proposal samples were sampled from all the images of the training dataset with a maximum IoU ratio of 0.5. Negative samples from the neighborhood of ground truth bounding boxes are extracted randomly from Set-1a and Set-2 with a minimum IoU ratio of 0.1. Additionally, bootstrapping is applied every 10k iterations to mine additional hard negative samples from Set-2 and Set-3 using the current model. In case the negative proposal count is not sufficient to fill up the mini-batch, additional normal negatives (i.e. not touching a cyclist ground truth) are extracted randomly from the images of Set-2 and Set-3. All FRCN models are trained for 50k iterations and a learning rate of 0.001 is used for the first 25k iterations and is reduced to 0.0001 for the final iterations. Separate training and testing is done for each proposal method.

Results are compared to ACF- and LDCF-based detector ensembles. To cover the different viewpoints of the cyclist, ACF and LDCF detectors are trained separately for different cyclist viewpoints (wide, intermediate, narrow) as described in [144]. Thus, for ACF and LDCF three detectors are trained and applied in parallel as a detector ensemble. To allow a fair comparison, downscaled images (i.e. 1300×650 px) are also used for the ACF- / LDCF-based detector ensembles.

Figure 3.9: Qualitative results for the cyclist detection using SP-3-FRCN-VGG (precision of 0.9). Column one and two showing good detection results. Column three and four showing various error sources (confusion to other VRU classes, multi-detections and false negatives).

All detectors deploy a greedy non-maximum suppression to suppress bounding boxes with lower scores, similar to the method in [236]. For ACF and LDCF detector ensembles the non-maximum suppression is performed over all results from the detectors in the ensemble.

Evaluation Qualitative results and common error sources are shown in Figure 3.9 for the top performing SP-FRCN configuration.

Figure 3.10, 3.11 and 3.12 illustrates the overall detection performance of the evaluated detectors for the easy, moderate and hard test subset (Set-4). In the easy subset, all the solution families (FRCN, ACF, LDCF) achieve a high competitive Average Precision (AP) when ignoring neighboring classes. With the subset becoming harder, the performance of all the detectors decrease gradually, where cyclists are at lower resolution and under partial / heavy occlusion. All SP-FRCN configurations outperform SS-FRCN and EB-FRCN, which illustrates the good performance of stixel-based proposal method for cyclist detection. Furthermore, using three aspect ratios (SP-3) in combination with a VGG network also outperforms ACF / LDCF-based detector ensembles, although the ACF- and LDCF-based detector ensembles reach a higher precision for lower recalls. Interestingly, it can be observed that the detection performance slightly degrades with more than three stixel-based proposal aspect ratios (SP-5, SP-7).

A higher Average Recall due to more proposals does not automatically result in an increased AP. This is also observed in [97]. With a higher number of redundant proposals for one object, also the probability of multi-detections increases. If multi-detections are allowed in evaluation, the achieved AP for SP-5 and SP-7 is almost the same as for the SP-3 configuration.

If the detection of pedestrians and other riders is enforced (Figure 3.10b, 3.11b, 3.12b), the performance of all detectors drop significantly, which illustrates that all detectors have problems distinguishing cyclists from pedestrians and other riders. It should be noted that multi-class training was not applied here, i.e. all classes other than cyclist are combined in the background class. A better separation can be expected when the classes are explicitly trained against each other.

The average CNN inference time per image increases linearly with the proposal count. Figure 3.13 shows this for the applied VGG-16 and ZF network. With SP-3-FRCN-ZF, near-realtime capabilities ($0.22\,\mathrm{s}$ on average including stixel-based proposal generation and stixel world estimation) are reached when deploying a ZF network architecture for feature extraction. Due to the smaller base network, a slight performance drop is visible for SP-3-FRCN-ZF on the easy / moderate / hard test subset, i.e. a difference of -4.1 / -4.8 / -4.3 (for the ignore case) and -4.7 / -4.6 / -4.1 (for the enforce case) on AP compared to the SP-3-FRCN-VGG configuration.

3.5 Discussion

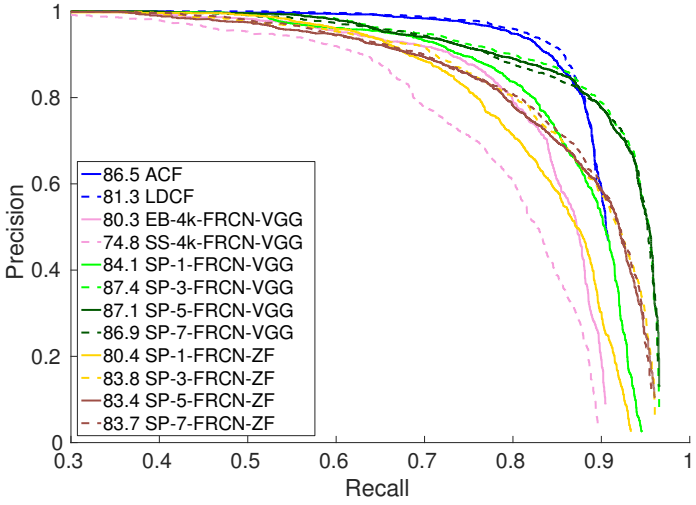
A large and richly annotated automotive stereo vision dataset for training and evaluating VRU detection was introduced. The first published part of the dataset, TDC-v1, focuses on cyclist detection evaluation. State-of-the-art object detection methods were carefully benchmarked on the task of cyclist detection and compared to the newly introduced architecture called SP-FRCN. SP-FRCN utilizes the power of FRCN combined with efficient stixel-based proposals extracted from the stixel world. The application of stixel-based proposals showed an improved cyclist detection performance compared to state-of-the-art mono-based proposal methods. SP-FRCN also slightly outperformed traditional methods (i.e. ACF- and LDCF-based ensemble detectors), suggesting that the deep feature representations better cover complex object variations (e.g. different sizes, viewpoints and occlusions).

Nevertheless, the results in Section 3.4.2 show that the performance of all

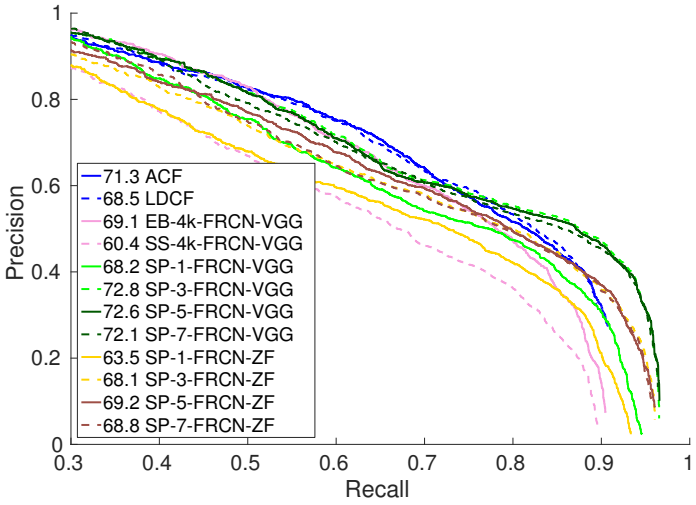
detection methods drops significantly when evaluation is done on the moderate and hard test set. During the experiments it became clear that one reason for that is a dataset limitation, since training set Set-1a only contains labeled training data of not occluded cyclists with a height greater than 60 pixels. Only a relatively small number of moderate and hard positive samples could be extracted based on Set-2. The results suggest that this is not enough to learn a robust occlusion model and representation of small cyclists. To cope with this limitations, TDC-v2 will provide significantly more data and extend Set-1a to a fully labeled version (Set-1b) including all difficulty levels and all relevant classes (i.e. pedestrians and other rider objects including their vehicles). Furthermore, groups and ignore regions will be added as suggested in [240]. The fully labeled train set also enables multi-class training. A better separation can then be expected between cyclists and pedestrians or other riders. For a fair benchmarking procedure an additional non-public test set (only images are given out) will be added in TDC-v2.

Parameters for the stixel world are chosen to be applicable in a realtime application and are not optimal for a detection evaluation. Especially for very small or highly occluded objects the proposal generation will fail because of missing support from ground-based stixels. Future work will therefore include a careful tuning of stixel and proposal parameters (e.g. stixel width and segmentation costs) to improve detection results. Furthermore, stixel clustering can be applied to reduce the proposal count and to better select appropriate proposal aspect ratios.

Recently, Ren *et al* [194] proposed the method Faster-RCNN, a FRCN variant with an integrated region proposal generation network. Unfortunately the original version underperformed for the detection on small persons [240] as well. Zhang *et al* [240] showed a few steps how Faster-RCNN performance can be tuned. Among these steps are object-specific anchor box adaptions, input image upscaling, smaller stride sizes, solver otpimization and already mentioned explicit ignore region / group handling, showing superior performance for person detection. Most of these improvements can also be transferred to the proposed SP-FRCN architecture, with the potential to further improve detection results.

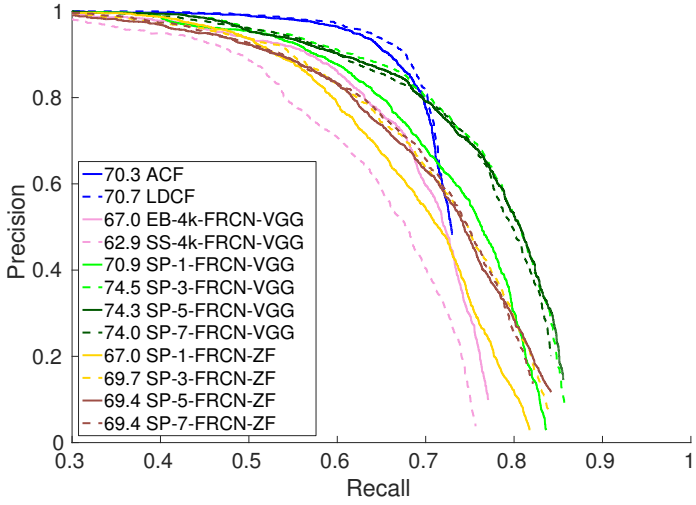


(a) Easy (ignore)

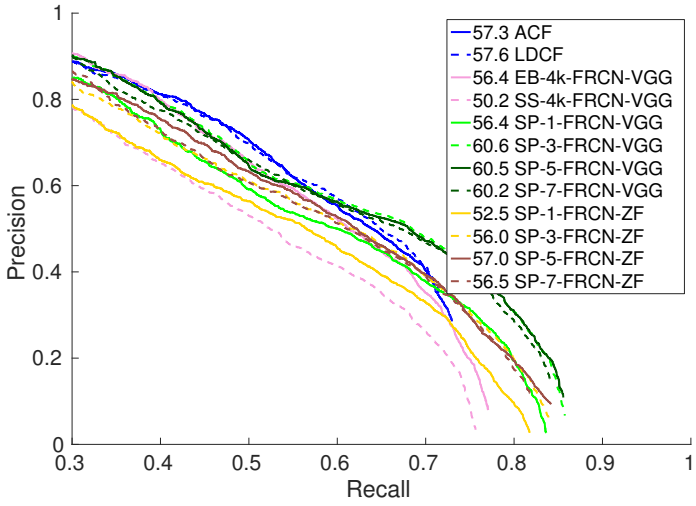


(b) Easy (enforce)

Figure 3.10: Precision-Recall curves of various detector configurations shown on the **easy** subset of the TDC test set (Set-4). When evaluating cyclist detectors, a differentiation is done between (a) ignoring and (b) enforcing the detection of neighbor classes (i.e. pedestrians and other riders). The Average Precision (AP) for each method is shown in the legend.

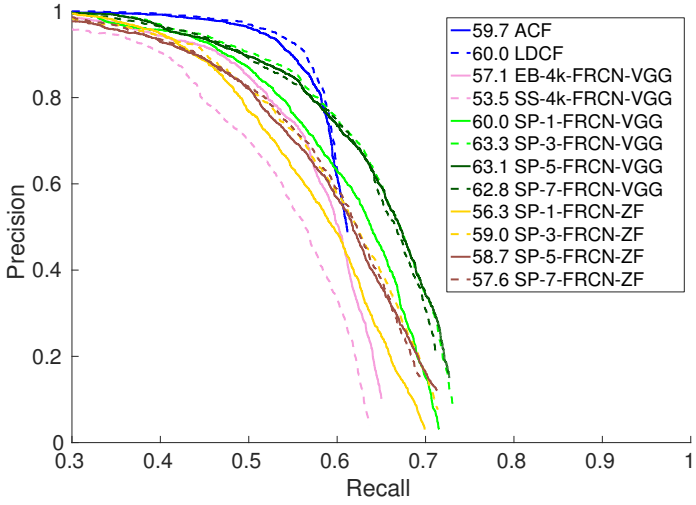


(a) Moderate (ignore)

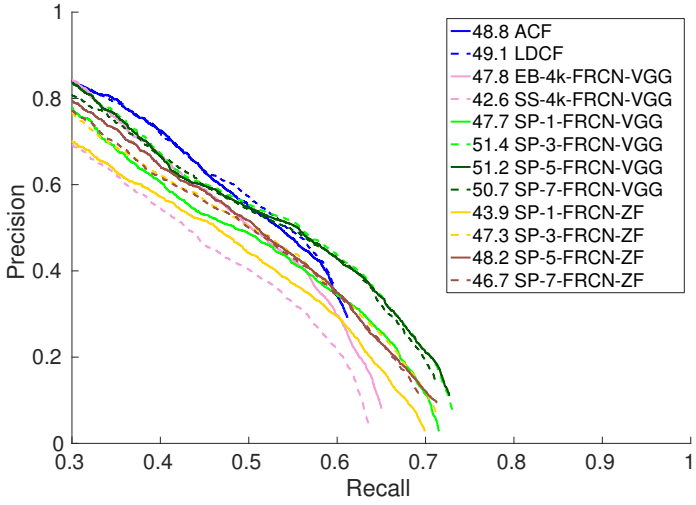


(b) Moderate (enforce)

Figure 3.11: Precision-Recall curves of various detector configurations shown on the **moderate** subset of the TDC test set (Set-4). When evaluating cyclist detectors, a differentiation is done between (a) ignoring and (b) enforcing the detection of neighbor classes (i.e. pedestrians and other riders). The Average Precision (AP) for each method is shown in the legend.



(a) Hard (ignore)



(b) Hard (enforce)

Figure 3.12: Precision-Recall curves of various detector configurations shown on the **hard** subset of the TDC test set (Set-4). When evaluating cyclist detectors, a differentiation is done between (a) ignoring and (b) enforcing the detection of neighbor classes (i.e. pedestrians and other riders). The Average Precision (AP) for each method is shown in the legend.

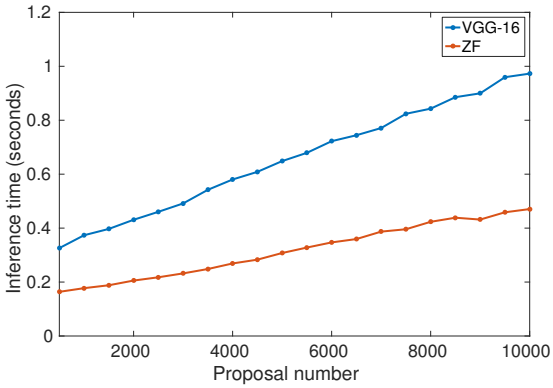


Figure 3.13: CNN inference time per image (in seconds) depending on proposal count, for VGG16 (blue) and ZF network (red). Inference times are averaged over 200 images of same proposal count.

4

PEDESTRIAN SEGMENTATION USING SHAPE AND DATA CUES

Separating object information from clutter is broadly used in image and video editing. For surveillance and intelligent vehicle applications it sets the basis for a more detailed analysis on the object. Localization accuracy of the whole object or object parts can be greatly improved by an accurate segmentation. Thus, a segmentation of the VRU can serve as a basis for extracting further VRU details, such as part-based orientations or posture details like e.g. the gait cycle of a pedestrian. Given the intelligent vehicle domain, disparity information obtained from a stereo camera might also contribute to an accurate pixel-wise segmentation.

4.1 Introduction

An accurate pixel-wise pedestrian segmentation is aided by utilizing already available bounding box detections, e.g. provided by the stixel-based detector

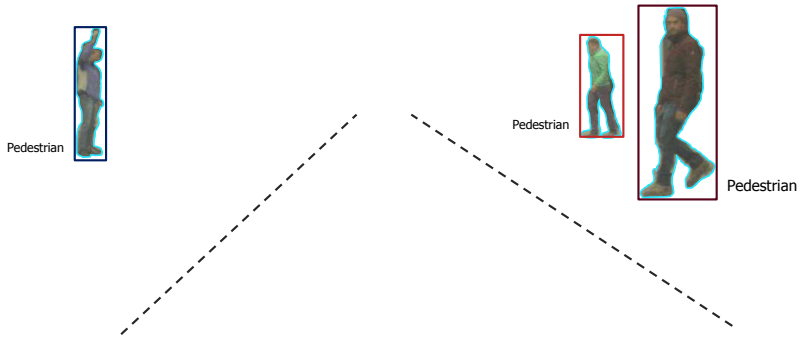


Figure 4.1: Accurate pixel-wise pedestrian segmentation based on initial bounding box detections (cf. Figure 3.1) can help to extract further VRU details, such as part-based orientations or posture details like the gait cycle.

presented in Chapter 3. This is visualized in Figure 4.1. Based on the detected VRU, this chapter introduces *PedCut*, a novel method for pedestrian segmentation in the intelligent vehicles domain. The name highlights the idea to cut out a pedestrian from the image, i.e. find the optimal closed contour to separate the pedestrian from background. The framework combines generative shape models and multiple data cues. For online inference, the iterative framework uses an approach inspired by the Expectation-Maximization algorithm [19] to optimize both, the image segmentation and the underlying shape model. Framework initialization is done from a learned Multi Statistical Shape Model (MSSM) [94] applying Chamfer matching [87] for initialization and shape refinement. In the Expectation step (E-step), the shape information is introduced in the terms of a Conditional Random Field (CRF) formulation, joining other data terms derived from color, texture and disparity cues. In the Maximization step (M-step), the resulting segmentation is used to adapt the Statistical Shape Model (SSM), after which the process alternates. Figure 4.2 shows an overview of the main components of the proposed approach. Experiments on the public Penn-Fudan pedestrian dataset show that the proposed method outperforms the state-of-the-art. Results are further provided on a newly introduced dataset, captured from on-board a vehicle, which includes disparity data. This dataset is made public

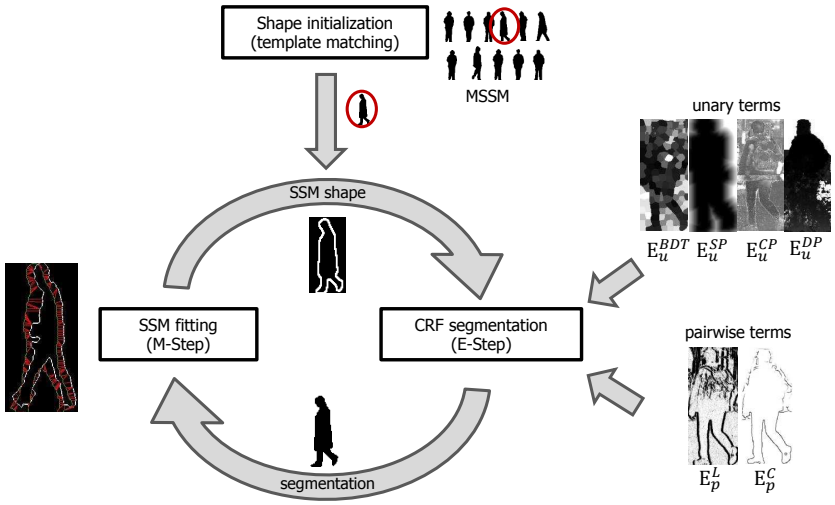


Figure 4.2: Overview of the iterative EM-like segmentation framework, alternating CRF-based segmentation (E-step) and SSM fitting (M-step), given a shape initialization from a MSSM model.

to facilitate benchmarking.

The main contributions of this chapter can be summarized as:

- A proposed pixel-wise segmentation approach that combines shape information with data cues to cope with the large variation of pedestrian appearances across cluttered backgrounds. In contrast to [130] multiple data cues (color, disparity and a discriminative superpixel classifier) are combined within a Conditional Random Field (CRF) formulation and Multiple Statistical Shape Models (MSSMs) [94] are used to cope with stronger shape aspect variations. In an EM-like manner shape and data cues are refined over multiple iterations improving the final binary segmentation.

- Results are presented on the public Penn-Fudan pedestrian dataset, showing that the proposed method outperforms the state-of-the-art. To show the benefit of using additional disparity cues, a new pedestrian segmentation dataset has been introduced and was made public to facilitate further research.

This chapter is based on the work published in [81].

4.2 CRF Segmentation (Expectation Step)

The method uses a CRF energy formulation [165] of the form

$$E(\boldsymbol{y}|\boldsymbol{X}) = \sum_{i \in \mathcal{V}} E_u(y_i|\boldsymbol{X}) + \sum_{i,j \in \mathcal{E}} E_p(y_i, y_j|\boldsymbol{X}) \tag{4.1}$$

defined on the index set $\mathcal{V}$ with an eight-connected edge neighborhood $\mathcal{E}$ combining unary terms (E_u) and pairwise terms (E_p). The posterior of this energy formulation can be defined with $p(\boldsymbol{y}|\boldsymbol{X}) = \frac{1}{Z} exp(-E(\boldsymbol{y}|\boldsymbol{X}))$, where $\boldsymbol{X}$ represents the data input and Z is the partition function. It follows, that the most probable labeling $\boldsymbol{y}^* \in \boldsymbol{\mathcal{L}}$ (MAP estimate) can be found by minimizing the energy $E(\boldsymbol{y}|\boldsymbol{X})$:

$$\boldsymbol{y}^* = \underset{\boldsymbol{y} \in \boldsymbol{\mathcal{L}}}{\operatorname{argmin}} E(\boldsymbol{y}|\boldsymbol{X}) \tag{4.2}$$

As the pairwise term stays submodular, efficient inference using Graph Cut [25] can be performed.

Four unary (Section 4.2.1) and two pairwise potentials (Section 4.2.2) are combined in the CRF formulation. In the following, let us define $\boldsymbol{I}_i$ as the value in Lab color space [184] and $\boldsymbol{D}_i$ as the corresponding disparity value at pixel i. Semi Global Matching (SGM) [109] is used for disparity computation. Let $\boldsymbol{S}_i$ be the feature vector of the superpixel containing pixel i. Furthermore, let $\boldsymbol{\Omega}$ be the current shape template extracted from shape initialization (Section 4.3.1) as used in the first iteration, or after the SSM fitting step (Section 4.3.2) as used after the first iteration.

4.2.1 Unary Terms for CRF Segmentation

The four unary potentials are based on superpixel classification (BDT), shape contour (SP), color (CP) and disparity (DP). The combined unary term E_u becomes:

$$\begin{aligned} E_u(y_i|\boldsymbol{S}, \boldsymbol{\Omega}, \boldsymbol{I}, \boldsymbol{D}) = \alpha_u^{BDT} E_u^{BDT}(y_i|\boldsymbol{S}) + \alpha_u^{SP} E_u^{SP}(y_i|\boldsymbol{\Omega}) \\ + \alpha_u^{CP} E_u^{CP}(y_i|\boldsymbol{I}) + \alpha_u^{DP} E_u^{DP}(y_i|\boldsymbol{D}) \end{aligned} \tag{4.3}$$

Weights α_u^{BDT}, α_u^{SP}, α_u^{CP} and α_u^{DP} control the influence for the specified unary potentials. The BDT potential remains constant over the framework iterations and is only computed once. The other potentials are being refined in each iteration. Figure 4.3 shows a visualization of the computed unary potentials with

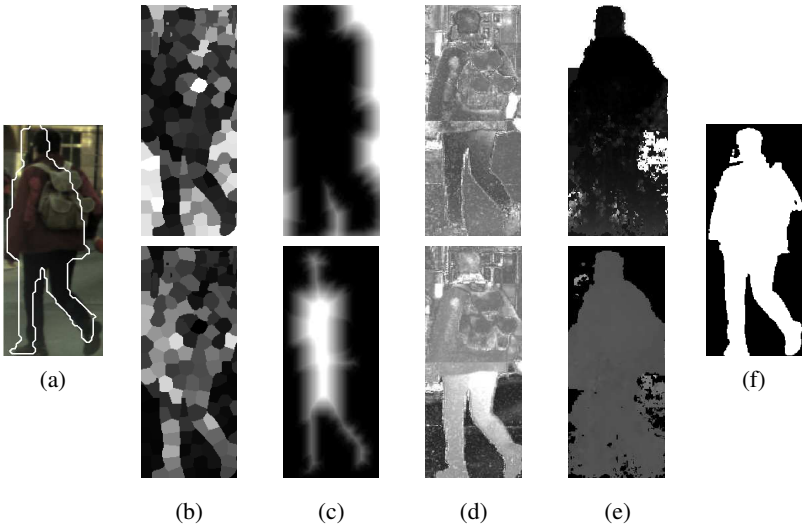


Figure 4.3: (a) Image with initial Chamfer match; (b)-(e) visualization of foreground (first row) and background potentials (second row) after the first iteration: (b) BDT potential, (c) shape potential, (d) color potential and (e) disparity potential; (f) shows the final segmentation.

initial matched shape template and resulting segmentation. In the following, the unary terms are described in detail.

4.2.1.1 Superpixel Classification with Boosted Decision Trees

Bag of visual words are extracted from oversegmented regions produced by SLIC superpixels [1] to train a Boosted Decision Trees (BDT) classifier with N_C trees as in [111], using the logistic regression version of Adaboost [45, 83]. For *visual codebook* generation, a set of local image descriptors are clustered to generate representative *visual codewords*. It has been shown that using Decision Tree Ensembles as encoder is superior to the original proposed k-means clustering [162, 203].

As local image descriptors dense SIFT (dSIFT) [23] features are extracted over the given image with a step width of four pixel. An ensemble of N_V BDTs is used as encoder. Each tree is pruned to get about 100 leaf nodes per tree.

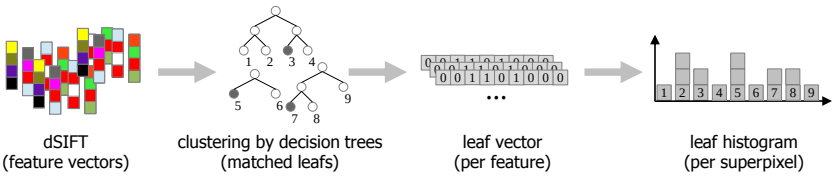


Figure 4.4: Generation of superpixel-based visual word histograms using Decision Trees. Clustering is done using a trained Decision Tree ensemble. For each tree, a given feature will end up in one of the leafs. The visual words are summed up over the area of a superpixel to gain a discriminative representation of a superpixel.

The size of the resulting visual word vector is equal to the number of all leafs in the BDT ensemble. Each encoded visual word vector then contains N_V one entries. Visual word vectors are summed up over the area of a superpixel. By this, a histogram is gained which captures a discriminative representation of a superpixel. Figure 4.4 outlines the generation of the visual word histogram for a superpixel.

Equal to the dSIFT codebook, an additional codebook is created based on extracted Maximum Response (MR8) [222] filter responses. The resulting MR8 superpixel histogram is appended to the dSIFT superpixel histogram. This concatenated histogram vector is then used as a superpixel classification feature $\boldsymbol{S}_i$.

The function $f_{\mathcal{B}}(\boldsymbol{S}_i)$ defines the BDT superpixel classifier output for the superpixel feature vector $\boldsymbol{S}_i$ of the superpixel which contains pixel i. It is a log-likelihood ratio score [111] which is used as potential in the CRF after sigmoid conversion:

$$E_u^{BDT}(y_i|\boldsymbol{S}) = -\log p(y_i|\boldsymbol{S}_i) \tag{4.4}$$

where

$$p(y_i = 1|\boldsymbol{S}_i) = \frac{1}{1 + exp(-f_{\mathcal{B}}(\boldsymbol{S}_i))} \tag{4.5}$$

and $p(y_i = 0|\boldsymbol{S}_i) = 1 - p(y_i = 1|\boldsymbol{S}_i)$.

4.2.1.2 Shape Potential (SP)

The shape template found in the shape initialization (Section 4.3.1) is used in the first iteration. From the second iteration on, the refined SSM (Section 4.3.2) is used to compute the shape potential.

A distance transformation from the current shape representation is calculated - denoting with $dist(loc_i, \mathbf{\Omega})$ the distance of the pixel location loc_i on the grid, to the nearest contour point on $\mathbf{\Omega}$. If pixel i lies inside the shape contour, $dist(loc_i, \mathbf{\Omega})$ is negative, otherwise $dist(loc_i, \mathbf{\Omega})$ is positive (see also [137]).

The resulting shape potential is defined with

$$E_u^{SP}(y_i|\mathbf{\Omega}) = -\log p(y_i|\mathbf{\Omega}) \tag{4.6}$$

where

$$p(y_i = 1|\mathbf{\Omega}) = \frac{1}{1 + exp(\mu_s \cdot dist(loc_i, \mathbf{\Omega}))} \tag{4.7}$$

and $p(y_i = 0|\mathbf{\Omega}) = 1 - p(y_i = 1|\mathbf{\Omega})$. The parameter μ_s determines the penalization of points outside, compared to points inside the shape.

4.2.1.3 Color Potential (CP)

Based on the segmentation at the previous iteration (or the initial shape at the first iteration), two Gaussian Mixture Models (GMMs) [19] are fitted to foreground (fg) and background (bg) in Lab color space, with K_C^{fg} and K_C^{bg} mixture components. With the additional vector $\boldsymbol{k} = \{k_1, ..k_i, ..k_N\}$ each pixel is assigned a unique component with $k_i \in \{1, .., K_C\}$ for foreground ($y_i = 1$) or background ($y_i = 0$). This hard assignment was also successfully used in [198]. Thus, the color potential is defined as:

$$E_u^{CP}(y_i|\boldsymbol{I}) = -\log p\left(\boldsymbol{I}_i|y_i, k_i, \theta_{k_i}\right) - \log \pi(y_i, k_i), \text{ with } y_i = \{0, 1\} \tag{4.8}$$

Above, k_i is the best component of the GMM chosen for pixel i with learned Gaussian parameters θ_{k_i} and component weight $\pi(y_i, k_i)$.

4.2.1.4 Disparity Potential (DP)

The disparity potential for the foreground is defined with one Gaussian distribution

$$E_u^{DP}(y_i = 1|\boldsymbol{D}) = -\log p(\boldsymbol{D}_i|y_i = 1, \theta_d) \tag{4.9}$$

with parameters $\theta_d = \{\tilde{d}, \sigma_d\}$. Here $\tilde{d}$ denotes the median value over all disparity values labeled as pedestrian in the current segmentation ($y_i = 1$). The disparity variance σ_d^2 can be learned over all disparity values and their neighborhoods within true pedestrian areas (using the ground truth segmentation).

The background potential $E_u^{DP}(y_i = 0|\boldsymbol{D})$ is modeled based on all disparity values $\boldsymbol{D}_i$ with values in the range $\boldsymbol{D}_i < \tilde{d} - 3\sigma_d$ and $\boldsymbol{D}_i > \tilde{d} + 3\sigma_d$ using a GMM as in the color potential. Like in the color potential only the best out of K_D^{bg} GMM components is selected for each pixel.

4.2.2 Pairwise Terms for CRF Segmentation

The pairwise term takes the form of a generalized Potts model [24] and is defined with

$$E_p(y_i, y_j|\boldsymbol{I}, \boldsymbol{D}) = \\ \left(\alpha_p^L E_p^L\left(y_i, y_j|\boldsymbol{I}\right) + \alpha_p^C E_p^C\left(y_i, y_j|\boldsymbol{I}, \boldsymbol{D}\right)\right) \times \delta(y_i \neq y_j), \tag{4.10}$$

with

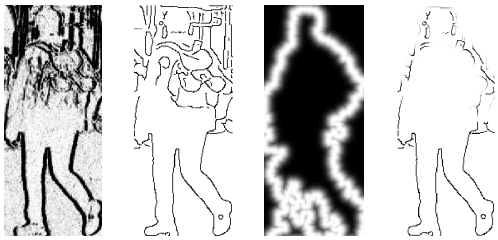
$$\delta(y_i \neq y_j) = \begin{cases} 1, & y_i \neq y_j \\ 0, & \text{otherwise.} \end{cases} \tag{4.11}$$

The combination of the color-sensitive (E_p^L, Figure 4.5a) and a contour-sensitive (E_p^C, Figure 4.5b) term forces consistent regions and assigns a lower cost to edges that lie on true contours. In the contour-sensitive term an optional disparity-based weighting (Figure 4.5c) can be added. Figure 4.5d shows that disparity information can improve the contour-sensitive pairwise term and masks out misleading edge contours. Weights α_p^L and α_p^C control the influence for the specified pairwise potentials.

4.2.2.1 Color-sensitive Potential

The color-sensitive term is specified such that it increases the cost of a graph edge inversely proportional to the color difference in Lab color space of two neighbored pixels i and j. The resulting term has the form

$$E_p^L(y_i, y_j|\boldsymbol{I}) = \\ \exp\left(\frac{-||\boldsymbol{I}_i - \boldsymbol{I}_j||}{2\sigma_l^2}\right) \frac{1}{dist(loc_i, loc_j)}, \tag{4.12}$$



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

Figure 4.5: Visualization of pairwise terms: (a) color-sensitive; (b) contour-sensitive; (c) computed disparity-based weighting based on the contour boundary pixels enclosing the disparity-based segmentation; (d) resulting disparity-weighted contour potential.

where loc_i denoting again the location of a given pixel i on the grid. The variance σ_l^2 can be set according to the camera noise [26].

4.2.2.2 Contour-sensitive Potential

The contour-sensitive term increases the cost of a graph edge inversely proportional to the edge magnitude between pixels i and j.

The optional disparity weighting term is used in the contour sensitive term if disparity information is available. Let $f_U(\boldsymbol{D}) \mapsto \boldsymbol{U}_D$ define a mapping from disparity values $\boldsymbol{D}$ to the set of contour boundary pixels $\boldsymbol{U}_D = \{u_1, u_2, .., u_n\}$ enclosing the disparity-based foreground segmentation (cf. Section 4.2.1.4). The disparity weighting $w_D(m, \boldsymbol{D})$ finally adds a higher importance to image edges which are lying near disparity-based boundary pixels $\boldsymbol{U}_D$. The weighting term is modeled with a Gaussian kernel:

$$w_D(m, \boldsymbol{D}) = \exp\left(\frac{-\left(\min_{u \in f(\boldsymbol{D})}\left[dist\left(loc_u, loc_m\right)\right] - \mu_{dp}\right)^2}{2\sigma_{dp}^2}\right) \tag{4.13}$$

Mean distance μ_{dp} and variance σ_{dp}^2 can be learned using the minimum distance of disparity-based segmentation contours to ground truth pedestrian contours.

The final contour-sensitive term becomes:

$$E_p^C(y_i, y_j|\boldsymbol{I}, \boldsymbol{D}) = \exp\left(\frac{-\max_{m\in\overline{ij}}\left[|\nabla \boldsymbol{I}_m| \cdot w_D\left(m, \boldsymbol{D}\right)\right]}{2\sigma_c^2}\right)\frac{1}{dist(loc_i, loc_j)} \tag{4.14}$$

The notation $\overline{ij}$ denotes the line containing all pixels between pixel i and j, while $|\nabla \boldsymbol{I}_m|$ denotes the edge magnitude at pixel m.

4.3 Shape Representation and Alignment

Statistical Shape Models (SSM) [47], also known as Point Distribution Models (PDM), are used to represent the shape information within the CRF formulation (Section 4.2.1.2). A SSM describes a linear subspace model based on a set of aligned and registrated landmark points on a n-dimensional shape. Each of the n landmark points is representing the same anatomical location on a sample, e.g. the landmark (x_5, y_5) is always representing the left toe of a pedestrian. Assuming the scattering across the samples is Gaussian, a compact representation using Principal Component Analysis (PCA) [65] can be found for the shape model.

For highly articulated objects, like pedestrians, a single linear subspace model cannot capture all the complex deformations of such an object. One straightforward solution is to divide the registrated and aligned samples into N_C possible object deformation cluster and use one SSM per cluster to capture object deformations within this cluster. This results in a Multi Statistical Shape Model (MSSM) [94].

The shape training for PedCut consists of a set of $N_T = 10946$ pedestrian shape exemplars, obtained by manual labeling. A MSSM is derived from this training set, based on shape registration and clustering, as described in [94]. A total of $N_C = 12$ clusters are obtained; each involve a SSM for a particular shape aspect (frontal pose feet closed, rightwards feet open, etc.). The dimensionality of the linear subspace (i.e. the number of eigenvectors) were chosen dynamically to cover 95% of total variance.

The iterative process starts with a MSSM instantiation supplied by a shape initialization module (Section 4.3.1). This MSSM instantiation is used as shape potential, joining the data-driven cues in the CRF-based segmentation (Section 4.2), i.e. the E-step. The resulting binary segmentation (from the E-step)

allows to update the parameters of the MSSM instantiation (Section 4.3.2), i.e. the M-step. The process alternates until the CRF-based segmentation does not change appreciable any more or a maximum of N_{it} iterations is reached.

4.3.1 Shape Initialization

Input to shape initialization is an image region of interest (i.e. a bounding box) provided by a pedestrian detector front-end (optionally, this includes the associated disparity values). As Active Shape Models (ASMs) [47] defined on SSMs can get stuck in local minima, template matching in the region of interest is performed using the individual pedestrian shape exemplars. Chamfer matching, differentiated by gradient direction (in this case: four discretization intervals, not encoding the gradient sign) is applied, as described in [87]. The best matching shape exemplar is converted to its MSSM representation (i.e. SSM representation of respective cluster); it acts as a shape prior in the following CRF segmentation step.

4.3.2 Fitting the SSM (Maximization Step)

An ASM approach [47] is used to fit the SSM model to the obtained CRF segmentation after each iteration. Point correspondences between SSM and image are given by Chamfer matching [87]. As in shape initialization (Section 4.3.1), Chamfer matching can be differentiated by gradient direction. Since matching is performed against a binary image (i.e. it is known what pixels lie inside the segmentation), additional information about the gradient sign can be applied here to improve the matching process (i.e. eight instead of four edge discretization intervals for gradient direction). Figure 4.6 shows an example where ambiguities can be resolved by taking the gradient sign into account.

4.4 Daimler Pedestrian Segmentation Dataset

Due to the scarcity of public available pedestrian segmentation datasets containing stereo and color information, a further new dataset was created. The dataset, called Daimler Pedestrian Segmentation Dataset (DPSD), contains a collection of 785 pedestrian cutouts from 228 images captured with a stereo camera from on-board a moving vehicle (image resolution 1176×640 pixel, baseline 22 cm, frame rate 16.7 Hz). The cutouts are taken from color images and are available

Figure 4.6: ASM using Chamfer matching to obtain point correspondences (red) between current segmentation (white) and current SSM model (green). Note that ambiguities, e.g. on the right leg, can be resolved by making use of eight gradient orientations (i.e. using the gradient sign).

with mapped disparity values, accurate ground truth annotations and camera calibration parameters. To avoid negative border effects the cutouts are extended by 10% of width (of the labeled bounding box) to the left and right, and 10% of height (of the labeled bounding box) to the top and bottom. Figure 4.7 shows some examples of the introduced dataset. For details about the labeling process the reader is referred to Chapter 7.

4.5 Experiments

Various experiments have been performed using different datasets to validate the PedCut method. The influence of the different data and shape cues have been investigated and evaluations show the benefit of combining multiple cues within the CRF framework.

4.5.1 Used Datasets

Four different datasets are used for the following experiments. The Daimler Urban Segmentation Dataset (DUSD) [203] train set contains 521 segmented pedestrian regions and is used to train the superpixel classifier. For segmentation evaluation and parameter validation, the introduced Daimler Pedestrian Segmentation Dataset and the public available Penn-Fudan dataset [226] are

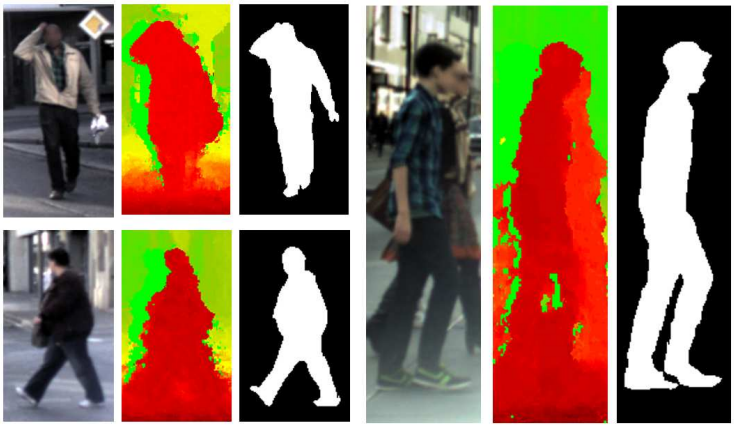


Figure 4.7: Samples from the Daimler Pedestrian Segmentation dataset (DPSD) showing color cutouts, corresponding disparity maps and ground truth segmentations.

used. The Penn-Fudan Dataset contains 170 color images with 345 box/shape-labeled pedestrians from which 169 labels are used in [21, 69]. The same data subset is used for an easier comparison to these methods. From the DPSD, only samples with a bounding box greater than 120 px are used. Table 4.1 gives an overview of the used segmentation datasets. The MSSM model with $N_C = 12$ linear subspace models is learned from the fourth dataset based on $N_T = 10946$ manually labeled pedestrian contour templates.

4.5.2 Parameter Setting

To build the visual codebooks for superpixel classification (Section 4.2.1.1), dSIFT and MR8 descriptors are extracted over the enclosing rectangular areas of the ground truth segmentations in the training set, containing background and pedestrian areas. Codebooks for the two descriptors are generated with $N_V = 30$ trees. The BDT superpixel classifier is finally trained with $N_C = 100$ trees, pruned to a maximum depth of 10, on the same data. The ground truth label of a superpixel is set to the class which covers most of the area of the superpixel. $K_C^{fg} = K_C^{bg} = 5$ GMM components are used to model the color unary potential (Section 4.2.1.3) and $K_D^{bg} = 3$ components are used for the

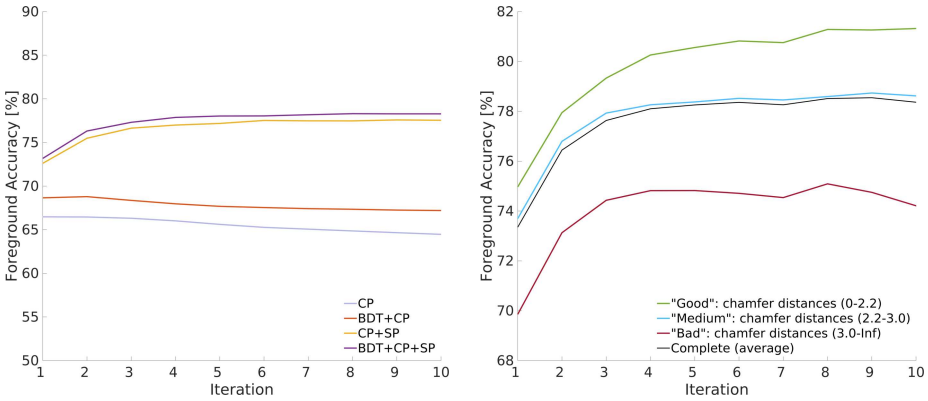
	DUSD [203] (train BDT)	DPSD [this paper] (validation / test)	Penn-Fudan [21, 226] (test)
#images	300	228	169
#pedestrians	521 (sel. 521)	785 (sel. 30 / 300)	169 (sel. 169)
min BB [h,w]	no BB	[121,34] pixel	[186,63] pixel
max BB [h,w]	no BB	[468,267] pixel	[373,207] pixel
color	no	yes	yes
disparity	yes	yes	no

Table 4.1: Used datasets and their characteristics.

disparity background unary potential (Section 4.2.1.4). Gaussian parameters modeling the disparity weighting (μ_{dp}, σ_{dp}) are learned on the validation set (cf. Section 4.2.2.2). Weight values $\boldsymbol{\alpha}$ for unary and pairwise terms and shape parameter μ_s have been tuned on the validation set. Parameters for the contrast-sensitive Potts models are set to $\sigma_l = 5$ and $\sigma_c = 1$.

4.5.3 Results on the Penn-Fudan Dataset

Segmentation accuracy is measured by the Intersection over Union (IoU) ratio as described in the PASCAL VOC challenge [71]. Figure 4.8a shows the FG segmentation accuracies using various cue combinations over the EM iterations (a similar plot applies for the BG). The incorporation of SP can be seen to have a major beneficial effect providing also a good initialization for the iterative process. The benefit of adding BDT is only substantial when SP is not available. Since the BDT potential is weak and not refined during the iterations, it does not help to restrict areas for the color potential well enough. Therefore the segmentation does not improve after repeated iterations. Figure 4.8b shows the dependence of the segmentation accuracy on shape initialization (i.e. availability of good contrast object contours). A better Chamfer match with a lower Chamfer distance at initialization improves the average segmentation accuracy. Table 4.2 shows the results for different cue combinations. Each additional cue improves the segmentation accuracy.



(a) Foreground accuracy

(b) Dependence on shape initialization

Figure 4.8: (a) FG segmentation accuracy over EM iterations, using various cue combinations. (b) Dependence on shape initialization (BDT+CP+SP)

	Foreground	Background	Average
BDT	42.8%	54.4%	48.6%
CP	64.3%	66.7%	65.5%
BDT+CP	67.1%	70.6%	68.9%
SP	67.2%	71.6%	69.4%
SP+CP	77.5%	80.3%	78.9%
BDT+SP+CP	**78.5%**	**81.5%**	**80.0%**

Table 4.2: Segmentation accuracy for various cue combinations on the Penn-Fudan dataset.

A comparison with the state-of-the-art [21, 69] is given in Table 4.3. The proposed method outperforms the methods presented in [21, 69] in foreground and background accuracy. Figure 4.9 shows representative segmentation results on the Penn-Fudan dataset with the best performing cues BDT+CP+SP.

	Foreground	Background	Average
Bo & Fowlkes [21]	73.3%	81.1%	77.2%
Eslami & Williams [69]	71.6%	73.8%	72.7%
This work (BDT+SP+CP)	**78.5%**	**81.5%**	**80.0%**

Table 4.3: Comparison with the state-of-the-art on the Penn-Fudan dataset

The available body-part label information associated with SSM points can be used to establish a basic component-based segmentation into head, upper and lower body. Note that the results cannot be compared to [21] and [69] for the upper and lower body, since they segment and label body components based on clothing instead of using true body proportions. For head segmentation, a comparison with the available ground truth is possible: a foreground accuracy of 57.1% compared to 51.8% [21] and 54.1% [69] is obtained for the head.

4.5.4 Results on the Daimler Pedestrian Segmentation Dataset

Table 4.4 shows results using different cue combinations on the DPSD dataset. When not having disparity data, the combination BDT+CP+SP performs best, as in the Penn-Fudan case. Adding disparity cues (DP) improves average segmentation accuracy slightly. Figure 4.10 shows representative segmentation results on the DPSD dataset with the best performing BDT+CP+SP+DP cues.

The unoptimized Matlab implementation requires about 2 s for segmenting a pedestrian in four EM iterations, running on a 3.33 GHz i7-CPU processor. Main bottleneck is the matching of templates during shape initialization (Section 4.3.1). In future work, the applied bruteforce matching can be replaced by the more efficient hierarchical approach presented in [87].

(a) (b) (c) (d) (e) (f) (g)

Figure 4.9: Results on the Penn-Fudan dataset after four EM iterations (BDT+SP+CP). First row: input images with initial / final SSM fit (red / white). Second row: correct / missing / excessive segmentation (white / red / cyan). Columns: (a)-(d) decent segmentations from decent shape initializations, (e)-(f) decent segmentations from poor shape initializations, and (g) poor segmentation from poor shape initialization.

4.6 Discussion

Supposing the pedestrian detection step is solved reasonable well, the proposed approach results in an accurate pedestrian segmentation by combining top-down shape information with multiple data-driven cues. On the public Penn-Fudan dataset, the proposed method outperformed state-of-the-art by more than five points on foreground accuracy while remaining ahead on background accuracy. Results were also shown on the newly introduced pedestrian segmentation dataset, containing a more challenging set of images recorded by an automotive camera. Since image resolution and color rendering will be improved with future automotive camera hardware, one can expect improvement in performance of the presented segmentation method as well.

The selection of a wrong MSSM cluster at shape initialization can lead to suboptimal segmentation results (e.g. Figure 4.10g). Currently, the framework cannot correct a wrong chosen cluster after the first iteration. One solution

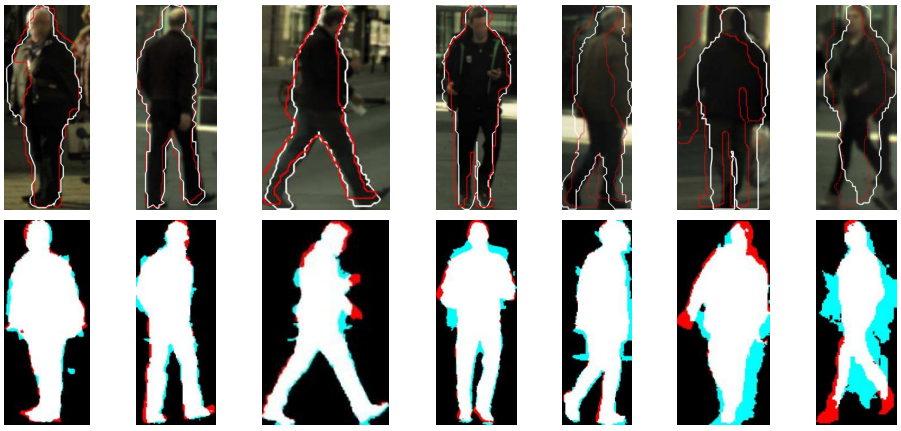
	Foreground	Background	Average
SP	68.9%	72.6%	70.7%
CP	60.2%	62.9%	61.6%
DP	63.8%	56.3%	60.1%
CP+DP	70.2%	70.9%	70.6%
SP+CP	73.5%	77.4%	75.5%
SP+DP	67.8%	69.0%	68.4%
SP+CP+DP	**76.4%**	**78.6%**	**77.5%**
BDT	56.6%	59.7%	58.1%
BDT+CP	65.4%	68.1%	66.7%
BDT+CP+DP	74.1%	75.2%	74.6%
BDT+SP+CP	74.9%	78.2%	76.5%
BDT+SP+CP+DP	**77.4%**	**79.6%**	**78.5%**

Table 4.4: Segmentation accuracy for different cue combinations on the newly introduced segmentation dataset (DPSD) with and without BDT potential.

would be to validate the cluster after the first iteration by e.g. the computed Chamfer distance. If the Chamfer distance to the current segmentation is not in a reasonable learned range, template matching should be repeated on the current segmentation to select a more appropriate MSSM cluster. On the other hand, Table 4.4 shows that by using disparity, a stable initialization can be reached also without using shape information. To increase the probability choosing the right MSSM cluster, template matching can be also applied after a disparity based initialization on the resulting segmentation. While these are ad-hoc solutions for the single-frame case, the introduction of a spatio-temporal MSSM model as presented in [94] provides a more generic solution. Hidden Markov Models (HMM) can be applied to model transition probabilities between the clusters and resolve ambiguities, providing a more stable MSSM representation over time.

While the application in this work was restricted to the pedestrian case, the method can be also applied to other object classes, after an appropriate adaption of SSM and superpixel classifier models.

A complete segmentation of the pedestrian was in focus of this work. Nevertheless, part-based segmentation are possible based on available point correspondences given by the SSMs (as shown for the head part). Thus, the method



(a) (b) (c) (d) (e) (f) (g)

Figure 4.10: Results on the DPSD dataset after four EM iterations (BDT+SP+CP+DP). First row: input images with initial / final SSM fit (red / white). Second row: correct / missing / excessive segmentation (white / red / cyan). Columns: (a)-(d) decent segmentations from decent shape initializations, (e)-(f) decent segmentations from poor shape initializations, and (g) poor segmentation from poor shape initialization.

can help to extract object details, like head and arm position of a cyclist or the gate cycle of a pedestrian. A better part-based localization can then also simplify the extraction of other VRU details (e.g. head or body orientation).

5

HEAD AND BODY ORIENTATION ESTIMATION

Object part orientations can serve as an important cue to better model or infer the behavior of an object. Empirical evidence suggests that the head and body orientation of a pedestrian are good indicators to a human what the pedestrian will do next [104, 204]. Therefore, this Chapter addresses the problem of estimating pedestrian head and body orientation over time from a mobile stereo vision platform. While the focus here is on an application for active pedestrian safety in the intelligent vehicle domain, the developed techniques could also be applied to mobile robotic applications, where the aim is that a robot interacts socially with persons in its environment [72].

5.1 Introduction

A principled probabilistic approach is presented for dealing with faulty part detections, continuous orientation estimation, coupling of the body and head localization and orientation, and temporal integration. By estimating parameters of the spatial body part configuration from real data, anatomical characteristics are modeled.

Figure 5.1 shows the desired part-based orientations on top of available detections (Chapter 3), optionally supported by a pixel-wise pedestrian segmentation (Chapter 4).

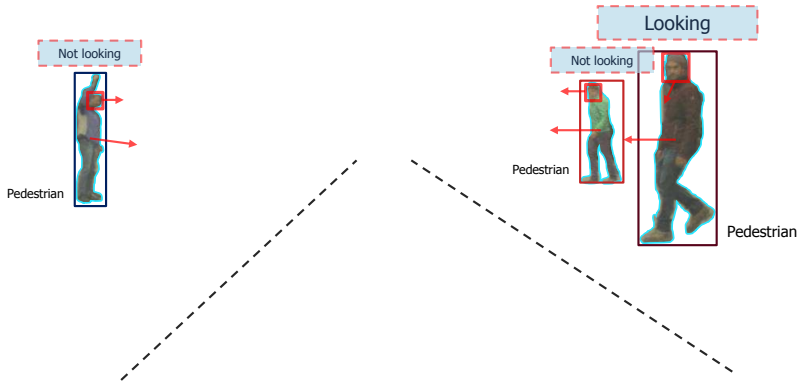


Figure 5.1: A human driver uses gaze direction and body language to make assumptions about VRU's intention and attentiveness. Extracted head and body orientation enable seeing machines to use similar information for a more sophisticated situation assessment.

Inputs to the proposed approach are bounding boxes provided by a state-of-the-art pedestrian tracker developed by the author (a bounding box pedestrian detector as described in Chapter 3 combined with a Kalman Filter [20], but that particular choice is not material for the contributions).

For both head and body parts, responses of a set of orientation-specific detectors are converted into a (continuous) probability density function. The parts are localized by means of a Pictorial Structure [76] approach, which balances part-based detector responses with spatial constraints. By estimating parameters of the spatial body part configuration from real data, anatomical characteristics are accounted for. The joint single-frame orientation estimates are integrated over time by particle filtering. Test and validation datasets were recorded using a stereo sensor setup (image resolution 1176×640 pixel, baseline 22 cm, frame rate 16.7 Hz), similar to the hardware setup nowadays already available in production vehicles. The recorded and labeled sequences contain multiple tracked pedestrians against various traffic backdrops with mostly benevolent il-

lumination conditions. Pedestrians cross or walk longitudinally with respect to the vehicle.

The contributions of this chapter can be summarized as:

- The Chapter presents a principled joint probabilistic head and body orientation estimation framework that handles faulty part detections, continuous orientation estimation, coupling of the body and head localization and orientation, and temporal integration.

- The proposed method differs to [14, 207], where no coupling between head and body was used at all, and differentiates to [14, 38, 37, 193, 195] in several ways. An intelligent vehicle context is considered, introducing additional challenges like complex and ever-changing background and lighting conditions. While a small set of detectors is used for canonical body part orientations, the joint observation model for head and body deals with continuous angles. It is also used to jointly localize the head and body (additionally exploiting disparity information from stereo vision and knowledge of body configuration) and still accounts for the possibility of occluded body parts or false positives. Furthermore, the temporal model couples the joint orientation dynamics and enforces temporal consistency.

- Experiments involve data from a vehicle-mounted stereo vision camera in a realistic traffic setting. The joint head an body orientation estimation reduces the mean absolute head and body orientation error up to 15° compared to simpler methods. The system runs in near-realtime ($8-9$ fps) and has been successfully used for improving path prediction [133].

5.2 Modeling Head and Body Parts with Orientation

Motivated by efficiency and the existence of previous modules, a decoupled pedestrian tracker is used that estimates for each time step t the pedestrian's position $\boldsymbol{x}_t = [x_t, y_t]$ and pedestrian's height h_t, defined in world coordinates on the ground plane, and velocity $\dot{\boldsymbol{x}}_t = [\dot{x}_t, \dot{y}_t]$. The pedestrian tracks are provided as input to the orientation tracker, which in turn tracks the head ω_t^H and body orientation ω_t^B jointly as $\boldsymbol{\omega}_t = [\omega_t^H, \omega_t^B]$. It is therefore assumed that all $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$ and h are known up to time t, and the focus shall be on the estimation of $\boldsymbol{\omega}_t$ only,

which is referred to as the state space. Additional constraints between head and body regarding the orientation are applied within the dynamic model.

Let $\boldsymbol{z}_t = [\boldsymbol{z}_t^H, \boldsymbol{z}_t^B]$ be the observed image data at time t, which can be decomposed into head observations $\boldsymbol{z}_t^H$ and body observations $\boldsymbol{z}_t^B$. Since the only input to the system is an estimate of the pedestrian's full bounding box, exact location of the body parts (head or full body) are not given. Therefore, multiple possible image regions need to be taken into account for both parts. For example, when there are N candidate regions for the head at time t, the corresponding observation can be written as $\boldsymbol{z}_t^H = [z_t^{H(1)}, z_t^{H(2)}, \cdots, z_t^{H(N)}]$.

Multiple detectors are used to evaluate how well an image region corresponds to a specific body part in a certain orientation. The angular domain of $[0°, 360°)$ is discretized into a fixed set of eight orientation classes, centered around $0°, 45°, \cdots 315°$ ($0°$ and $90°$ are associated with a frontal and left-facing pose, respectively, when viewed from the camera). Each class then has a detector, e.g. $f_0, f_{45}, \cdots, f_{315}$, for both head and body, such that the detector response $f_o(z)$ is strength for the evidence that image region z contains the body part in orientation class o. Note that this gives a tradeoff, as having more classes and detectors requires more training data and computational effort, but also yields more precise evidence of the true angle (up to some point). An additional non-target or background detector $f_-(z)$ assigns a likelihood to the case that z does not contain the body part. The output of all detectors $f_o(z)$ and $f_-(z)$ are then used to determine if and where a body part is present in the image region z, relying on disparity based image segmentation and a *Pictorial Structure* (PS) [76] on the head and body configuration as a spatial prior. Figure 5.2 shows an overview of the approach.

5.3 Joint Orientation Tracking

Let $\boldsymbol{z}_{1:t}$ denote all observations up to and including time t, and $\dot{\boldsymbol{x}}_{1:t}$ the corresponding pedestrian velocities provided by the position tracker. A Bayes filter is used to obtain the posterior, $p(\boldsymbol{\omega}_t|\boldsymbol{z}_{1:t}, \dot{\boldsymbol{x}}_{1:t})$, which represents the belief of the state at time t after observing $\boldsymbol{z}_{1:t}$. For each time instance the filter performs the following two steps:

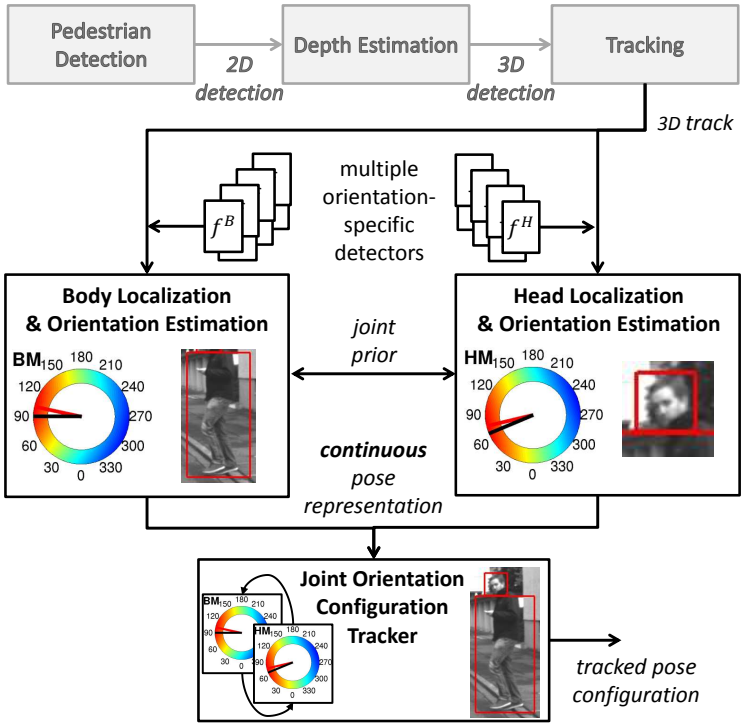


Figure 5.2: Proposed joint probabilistic orientation estimation approach (shaded modules are outside the scope of this chapter).

First, a prediction is made given all earlier observations,

$$p(\boldsymbol{\omega}_t|\boldsymbol{z}_{1:t-1}, \dot{\boldsymbol{x}}_{1:t}) = \int p(\boldsymbol{\omega}_t|\boldsymbol{\omega}_{t-1}, \dot{\boldsymbol{x}}_t) p(\boldsymbol{\omega}_{t-1}|\boldsymbol{z}_{1:t-1}, \dot{\boldsymbol{x}}_{1:t-1}) d\boldsymbol{\omega}_{t-1} \tag{5.1}$$

where $p(\boldsymbol{\omega}_{t-1}|\boldsymbol{z}_{1:t-1}, \dot{\boldsymbol{x}}_{1:t-1})$ is the posterior for the previous time step and $p(\boldsymbol{\omega}_t|\boldsymbol{\omega}_{t-1}, \dot{\boldsymbol{x}}_t)$ is the dynamic model. The applied dynamic model for the head and body orientations is

$$p(\boldsymbol{\omega}_t|\boldsymbol{\omega}_{t-1}, \dot{\boldsymbol{x}}_t) = p(\omega_t^H|\omega_{t-1}^H, \omega_t^B) p(\omega_t^B|\omega_{t-1}^B, \omega_{t-1}^H, \dot{\boldsymbol{x}}_t), \tag{5.2}$$

Section 5.3.1 and 5.3.2 will describe the modeled head and body dynamics in detail.

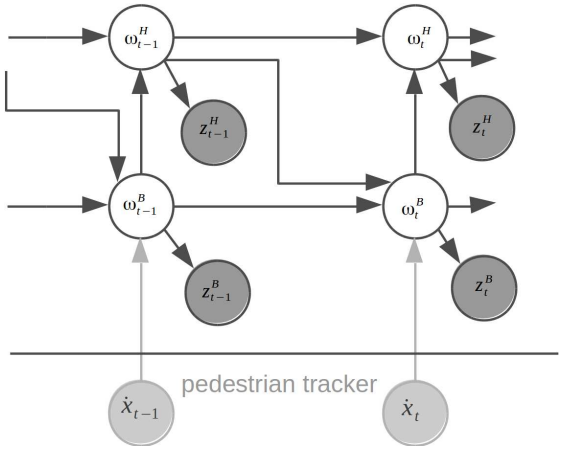


Figure 5.3: DBN showing the dependencies between latent variables (head and body orientation, ω^H and ω^B) and observed variables (head and body measurements, $\boldsymbol{z}^H$ and $\boldsymbol{z}^B$). Latent variables are unshaded, observed variables are shaded. Pedestrian velocity ($\dot{\boldsymbol{x}}$) is estimated by an external tracker.

Second, an update is made to incorporate new evidence $\boldsymbol{z}_t$ in the prediction, i.e.,

$$p(\boldsymbol{\omega}_t|\boldsymbol{z}_{1:t}, \dot{\boldsymbol{x}}_{1:t}) \propto p(\boldsymbol{z}_t|\boldsymbol{\omega}_t)p(\boldsymbol{\omega}_t|\boldsymbol{z}_{1:t-1}, \dot{\boldsymbol{x}}_{1:t}) \tag{5.3}$$

where $p(\boldsymbol{z}_t|\boldsymbol{\omega}_t)$ is the observation model, which will be discussed in Section 5.4.

Figure 5.3 shows the corresponding Dynamic Bayesian Network (DBN). Since exact inference is intractable, a Particle Filter (PF) [62] is used for an approximate inference. The PF represents the posterior distribution by a set of particles in the state space, which facilitates using a nonlinear and multimodal dynamic model.

5.3.1 Head Dynamics

Similar to [195], the head orientation at the current time step is constrained on the head orientation of the previous time step and on the current body orientation

with

$$p(\omega_t^H|\omega_{t-1}^H,\omega_t^B) = \alpha_{hh}\mathcal{V}(\omega_t^H;\omega_{t-1}^H,\kappa_{hh}) + (1-\alpha_{hh})\mathcal{V}(\omega_t^H;\omega_t^B,\kappa_{hb}), \tag{5.4}$$

where κ_{hh} and κ_{hb} are concentration parameters for the Von Mises distribution. The Von Mises $\mathcal{V}(\cdot;c,\kappa)$ is an analogue of the normal distribution for the circular domain, with mean angle c and concentration κ. It reduces to a circular uniform distribution when $\kappa = 0$. The first term in Equation (5.4) models the case that the current head orientation is distributed around the previous head orientation. The second term covers the (possibly alternative) case where the head has moved to a similar orientation as the body. The balance between temporal consistency and the assumption that the head orientation is around the body orientation is given by the weight α_{hh}.

5.3.2 Body Dynamics

The body orientation is constrained by conditioning it on the body and head orientation of the previous time step and on the current pedestrian velocity, i.e.,

$$\begin{aligned} p(\omega_t^B|\omega_{t-1}^B,\omega_{t-1}^H,\dot{\boldsymbol{x}}_t) = \\ \alpha_{bb}\mathcal{V}(\omega_t^B;\omega_{t-1}^B,\kappa_{bb}) + \alpha_{bh}\mathcal{V}(\omega_t^B;\omega_{t-1}^H,\kappa_{bh}) \\ + (1-\alpha_{bb}-\alpha_{bh})\mathcal{V}(\omega_t^B;ang(\dot{\boldsymbol{x}}_t),\kappa_{bv}). \end{aligned} \tag{5.5}$$

Let $ang()$ denote the angle of the velocity vector. Furthermore, $\alpha_{bb,bh} \in [0,1]$ (with $\alpha_{bb}+\alpha_{bh} \leq 1$) denote the weighting factors for the terms. The first term in Equation (5.5) expresses that the body orientation is typically centered around its previous orientation. There are cases when the body orientation changes to where the pedestrian is looking, which are captured by the second term. The last term expresses that the body orientation is typically aligned with the direction of motion. κ_{bb}, κ_{bh} and κ_{bv} denote concentration parameters. Concentration κ_{bv} depends on the velocity magnitude $||\dot{\boldsymbol{x}}_t||$ and on the pedestrian tracker confidence ξ_t (between 0 and 1). It was found that this dependency can be well represented by a logistic growth model [209] given parameters $\boldsymbol{\theta} = [\theta_1,\theta_2,\theta_3]$:

$$\kappa_{bv}(\dot{\boldsymbol{x}}_t,\xi_t) = \frac{\theta_1\xi_t}{1+\exp\left(-\theta_2(||\dot{\boldsymbol{x}}_t|| - \theta_3)\right)} \tag{5.6}$$

For a new confirmed pedestrian track, the orientation tracker is initialized by sampling the orientation ω_1^B and ω_1^H based on the estimated walking direction (i.e. velocity direction) of the pedestrian given concentrations κ_{bv} and κ_{hb}. Then, new evidence is incorporated by the given meassurement $\boldsymbol{z}_1 = [\boldsymbol{z}_1^H, \boldsymbol{z}_1^B]$ at this timestep.

5.4 Body Part Localization and Orientation Estimation

After having described head and body dynamics, this section continues with the observation model $p(\boldsymbol{z}_t|\boldsymbol{\omega}_t)$ used in Equation (5.3). The mapping between continuous and discrete orientation models is explained. Furthermore, auxiliary variables are used to cope with multiple potential head and body part regions. Finally, important priors are derived, which support the region selection process.

The proposed method assumes that the head and body observations are conditionally independent given the actual head and body state, i.e.

$$p(\boldsymbol{z}_t|\boldsymbol{\omega}_t) = p(\boldsymbol{z}_t^H|\omega_t^H)p(\boldsymbol{z}_t^B|\omega_t^B). \tag{5.7}$$

The superscripts refer to H for head, and B for body. Since both terms are computed in the same way, we can drop this superscript when referring to either term and, likewise, drop the time index t for simplicity, e.g. we write $p(\boldsymbol{z}|\omega)$ when referring to both $p(\boldsymbol{z}_t^H|\omega_t^H)$ and $p(\boldsymbol{z}_t^B|\omega_t^B)$.

5.4.1 From Continuous to Discrete Orientations

The orientation $\omega \in \mathbb{R}$ is a continuous value in the domain $[0°, 360°)$, but for the observation likelihoods the detectors for the discretized orientation classes are used. Therefore, the likelihood is defined in terms of the class Ω of the current $\boldsymbol{z}$, i.e.,

$$p(\boldsymbol{z}|\omega) = \sum_{\Omega} p(\boldsymbol{z}|\Omega)p(\Omega|\omega). \tag{5.8}$$

The probability $p(\Omega|\omega)$ expresses the probabilistic relationship between the continuous orientation angle ω and discrete class Ω, which is found by Bayes' rule, i.e.,

$$p(\Omega = o|\omega) = \frac{p(\omega|\Omega = o)p(\Omega = o)}{\sum_{k\in\Omega} p(\omega|\Omega = k)p(\Omega = k)}. \tag{5.9}$$

Here $p(\Omega)$ is a prior on the discrete class, and for each class o, $p(\omega|\Omega = o)$ is a Von Mises distribution, i.e.,

$$p(\omega|\Omega = o) = \mathcal{V}(\omega; c_o, \kappa_o) \tag{5.10}$$

where c_o and κ_o are the mean and concentration of the distribution for orientation class o, respectively. We now need to define the term $p(\boldsymbol{z}|\Omega)$, which is the observation likelihood given an orientation class instead of a continuous angle.

5.4.2 Handling Body Part Localization Uncertainty

An accurate body part localization is needed to estimate body part orientations correctly. Unfortunately we have only a pedestrian bounding box available, provided by the pedestrian tracker. Thus, we need to account for this location uncertainty for orientation estimation. To model the location uncertainty, two auxiliary variables, R and V, are introduced to express the discrete observation likelihood $p(\boldsymbol{z}|\Omega, R, V)$ in Section 5.4.2.1. In Section 5.4.2.2 we will then reduce the formulation again to $p(\boldsymbol{z}|\Omega)$ in terms of this extended likelihood.

5.4.2.1 Likelihood with Auxiliary Variables

The indicator variable $R = r, r \in \{1...N\}$ specifies which region $\boldsymbol{z}^{(r)}$ of the possible regions in $\boldsymbol{z}$ fits the sought head / body (and as a consequence, also specifies that all other regions do not fit the head / body). Additionally, the Boolean variable $V = v$ with $v \in \{0, 1\}$ indicates whether there exists a head / body in any of the N regions at all ($V = 1$), or whether none of the regions contain it ($V = 0$).

We can now express the region likelihood, given the auxiliary variables, in term of the detector responses as

$$p(z^{(s)}|\Omega = o, R = r, V) = \begin{cases} f_o(z^{(s)}) & \text{if } s = r \wedge V = 1, \\ f_-(z^{(s)}) & \text{otherwise.} \end{cases} \tag{5.11}$$

Since we assume that the all candidate regions are conditionally independent, the complete data likelihood is just

$$p(\boldsymbol{z}|\Omega, R, V) = \prod_{z^{(s)} \in \boldsymbol{z}} p(z^{(s)}|\Omega, R, V). \tag{5.12}$$

Intuitively, one would expect that all orientation classes are equally likely when the head / body is not contained in any region and therefore unobserved. This property indeed follows from Equation (5.11) and (5.12) since the observations are independent of the orientation Ω and selected region R when $V = 0$,

$$p(\boldsymbol{z}|\Omega, R, V = 0) = p(\boldsymbol{z}|V = 0) = \prod_{z^{(s)} \in \boldsymbol{z}} f_{-}(z^{(s)}). \tag{5.13}$$

5.4.2.2 Removing the Auxiliary Variables

First, the region likelihood is used to select an optimal value $\hat{r}$ for the region indicator R. Assuming that there is a head ($V^H = 1$) and a body ($V^B = 1$) in one of the head and body regions, the most probable head and body region configuration $\hat{r} = [\hat{r}^H, \hat{r}^B]$ is estimated by

$$\begin{aligned} \hat{r} = \underset{R^H, R^B}{\operatorname{argmax}} \Big[& \sum_{\Omega^H} p(\boldsymbol{z}^H|\Omega^H, V^H = 1, R^H) p(\Omega^H) \\ & \times \sum_{\Omega^B} p(\boldsymbol{z}^B|\Omega^B, V^B = 1, R^B) p(R^H, R^B|\Omega^B, \boldsymbol{D}) p(\Omega^B) \Big]. \end{aligned} \tag{5.14}$$

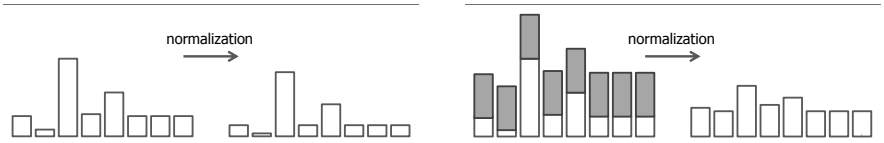
At this region selection step, several priors (i.e. prior to observing $\boldsymbol{z}$) are utilized. With $p(R^H, R^B|\Omega^B, \boldsymbol{D})$, which will be described in detail in Section 5.4.3, prior knowledge is introduced about the joint region configuration of head and body from of a PS model [76] dependent on the body orientation, and disparity data $\boldsymbol{D}$. The fixed Bernoulli distribution $p(V)$ and categorical distribution $p(\Omega)$ can be used to incorporate prior knowledge on the occurrences of false positives (e.g. set high probability for $p(V = 0)$ if false positives are common) and orientation classes, or just set to uniform to rely on the observation likelihoods only.

The term $p(\boldsymbol{z}|\Omega)$ without auxiliary variables is now be obtained by fixing R to $\hat{r}$ and integrating out the variable V, i.e.,

$$\begin{aligned} p(\boldsymbol{z}|\Omega) &= \sum_{v \in \{0,1\}} p(\boldsymbol{z}|\Omega, V = v, R = \hat{r}) p(V = v) \\ &= p(\boldsymbol{z}|\Omega, V = 1, \hat{r}) p(V = 1) + p(\boldsymbol{z}|V = 0) p(V = 0). \end{aligned} \tag{5.15}$$

Using Equation (5.11) and (5.12) to expand Equation (5.15) further, we see that the term can be efficiently evaluated up to a constant factor, i.e.,

$$p(\boldsymbol{z}|\Omega = o) \propto f_o(z^{(\hat{r})}) p(V = 1) + f_{-}(z^{(\hat{r})}) p(V = 0). \tag{5.16}$$



(a) Without f_- response (b) With f_- response

Figure 5.4: Visualization of the influence of a specific orientation response (a) without and (b) with adding f_- (shaded proportion) to the orientation detectors f_o (c.f. Equation (5.16)). If f_- is high, the influence of a specific detector response decreases after normalization. In the extreme case where only f_- responds, Equation (5.3) would reduce to the prior distribution.

It follows that the same is true for $p(\boldsymbol{z}|\omega)$ in Equation (5.8). The constant can be ignored, since it does not affect the posterior distribution of Equation (5.3) after normalization.

What can be also seen from Equation (5.16) is that a stronger background detector response f_- (relative to the orientation detectors f_o) leads to a higher weight of the second term, and therefore to a smaller relative differences between the likelihoods of the different orientation classes (after normalization). This means that in the extreme case where only f_- gives a strong response, the likelihood term is the same for all orientations. The posterior of Equation (5.3) would then reduce to just the prior distribution from the prediction step, i.e. no information on the true orientation was gained at this time step. This effect of f_- on the likelihood term is illustrated in Figure 5.4.

5.4.3 Spatial Prior over Body Part Regions

Let $\boldsymbol{h}_c(\boldsymbol{D})$ and $\boldsymbol{b}_c(\boldsymbol{D})$ be functions on the disparity $\boldsymbol{D}$ that give us an estimate of head and body position. We can factor the prior from Equation (5.14) into

$$p(R^H, R^B|\Omega^B, \boldsymbol{D}) \propto \\ p(\boldsymbol{h}_c(\boldsymbol{D})|R^H)p(\boldsymbol{b}_c(\boldsymbol{D})|R^B)p(R^H, R^B|\Omega^B). \tag{5.17}$$

The following two sections derive the terms used for this spatial prior.

5.4.3.1 Disparity-based Region Priors

Let $\boldsymbol{h}_c$ and $\boldsymbol{b}_c$ return the mean pixel location of head and body based on disparity values $\tilde{\boldsymbol{D}}$ in the range $\boldsymbol{D} < \tilde{d} - \epsilon$ and $\boldsymbol{D} > \tilde{d} + \epsilon$. Here, $\tilde{d}$ denotes the median value over all disparity values calculated over the given pedestrian track bounding box. The parameter $\epsilon = 1.5$ accounts for disparity estimation errors. Semiglobal Matching (SGM) [109] is used to calculate the disparity map $\boldsymbol{D}$. The likelihood of the head region is then modeled with

$$p(\boldsymbol{h}_c(\boldsymbol{D})|R^H = r^H) = \mathcal{N}(\boldsymbol{h}_c(\boldsymbol{D}); \boldsymbol{\mu}(r^H), \boldsymbol{C}^H), \tag{5.18}$$

where $\boldsymbol{\mu}(r^H)$ denotes the center (u- and v- coordinate) of a given head region r^H in image coordinates and $\boldsymbol{C}^H$ denotes the corresponding covariance. The likelihood $p(\boldsymbol{b}_c(\boldsymbol{D})|R^B = r^B)$ of the body region is modeled similarly.

5.4.3.2 Joint Region Prior

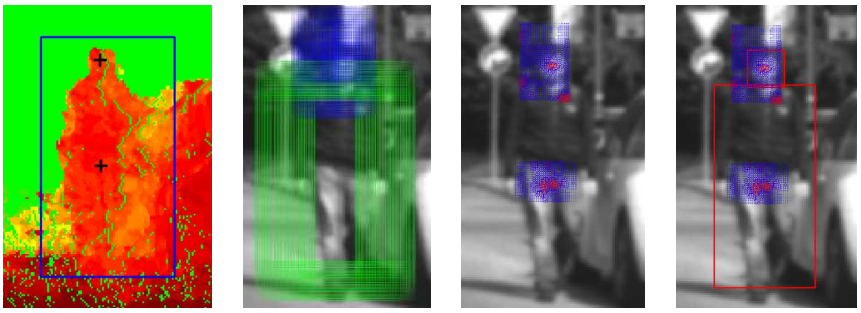
To joint spatial prior $p(R^H, R^B|\Omega^B)$ is based on a Pictoral Structure (PS) [76] model, which is dependent on the body orientation:

$$\begin{aligned} p(R^H = r^H, R^B = r^B|\Omega^B = o^B) = \\ \mathcal{N}(\boldsymbol{l}^D(r^H, r^B); \boldsymbol{\mu}^D_{o^B}, \boldsymbol{C}^D_{o^B}). \end{aligned} \tag{5.19}$$

$\boldsymbol{l}^D(r^H, r^B)$ denotes the distance between head and body region center divided by the width of the body region. $\boldsymbol{\mu}^D_{o^B}$ and $\boldsymbol{C}^D_{o^B}$ denote the mean and covariance for the model for the the body orientation class o^B.

5.4.3.3 Region Generation

Due to efficiency reasons, possible head and body regions are generated based on the tracked pedestrian height (h_t) and the estimated horizontal gravity line (g^x_c) inside the tracked bounding box. The gravity line g^x_c is estimated on the dominant peak in the histogram of the horizontally projected image locations of all disparity values $\tilde{\boldsymbol{D}}$. Since the tracked bounding box should already give an appropriate estimation of the body location, the number of used body hypotheses can be much less than it is for the head. The size of the generated head and body regions is set according to the estimated pedestrian height (h_t). The step size between regions is set dependent on the pedestrian distance. Figure 5.5 shows an example of the region generation process and the calculated region probabilities based on detector responses and the described prior information.



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

Figure 5.5: (a) The disparity map is used to estimate the centers (black crosses) of head and body ($\boldsymbol{h}_c$, $\boldsymbol{b}_c$), which serve in (b) as prior information over R^H (blue) and R^B (green) in addition to the detector responses. (c) Region probabilities after including the detector responses. Ambiguities here can be efficiently resolved by the combined head and body localization with the PS model. (d) Selected joint maximum.

5.5 Experiments

This section presents details on the used datasets, parameter settings and the performed experiments. Results show the benefit of a joint estimation of head and body orientation, applying anatomical constraints and the temporal integration within the proposed DBN formulation.

5.5.1 Datasets

For the training set head and body bounding boxes were extracted from 9300 manually labeled pedestrian samples from 6389 images. Pedestrian samples have a minimum / maximum / mean height of 69 / 344 / 122 pixels and are combined with background samples used for the training. Half of the background samples were sampled from false positive pedestrian detections in the area of the sought head / body. The other half was sampled around the head / body of a true positive pedestrian detection with a maximum overlap of 25% to the true head / body bounding box. No pedestrian occurring in the training set also occurs in the test set.

For the validation / test datasets, 32 / 60 image sequences were recorded on-board a moving vehicle using a stereo sensor setup (1176×640 pixel, baseline of 22 cm, frame rate of 17 Hz). They contain pedestrians against various traffic backdrops in mostly benevolent illumination conditions (i.e. no strong backlighting) and with limited occlusion (up to 20% of the pedestrian can be occluded). Pedestrians predominantly cross laterally or walk longitudinally with respect to the vehicle. The vehicle mostly drives straight (i.e. no vehicle turns at intersection) at speeds up to $10.3\,\mathrm{m/s}$. Ground Truth (GT) was obtained by manual labeling the body part orientations and the location of body parts (each by a bounding box). For details on the labeling process the reader is referred to Chapter 7.

Input to the framework are bounding boxes gained from pedestrian detector (similar to the one described in Chapter 3) tracked by a Kalman filter [20], implemented by the author (cf. the shaded modules of Figure 1). In each frame, a pedestrian location estimate is associated with a GT label when the distance between them is smaller than a threshold. This threshold is set according to a percentage of the Euclidean distance of the GT label to the camera. A different percentage of 8% and 12% was selected for lateral and longitudinal direction, since uncertainty in lateral direction is in general smaller.

For the evaluation of the orientation estimation performance, only estimated tracks in the test set are considered which follow more than 80% of their duration a particular GT track (several estimated tracks can correspond to a single GT track). All other estimated tracks are regarded as false positives and set aside. Furthermore, only track samples with a maximum lateral / longitudinal distance of 5 m / 35 m to the camera have been included. Doing so, 65 "valid" estimated tracks are obtained on the validation / test set (i.e. five GT tracks were split by the pedestrian tracker) with 3167 samples.

5.5.2 Detectors

Eight orientation-specific detectors $f_o(z)$ with class centers $o \in \{0°, 45°, \cdots, 315°\}$ are trained in a modified one versus all manner, including the background class. The originally labeled 16 orientation bins were reduced to 8 GT orientation bins, by merging three neighboring labeled orientation bins (i.e., labeled bins $337.5°$, $0°$ and $22.5°$ are merged to GT bin $0°$, labeled bins $22.5°$, $45°$ and $67.5°$ are merged to GT bin $45°$, etc.). By this merge, samples from a labeled bin are always represented in two resulting GT bins. Preliminary experiments [63] showed that this gives a better performance when combining the

detectors to estimate a continuous orientation. Figure 5.6 shows labeled head and body examples for the eight orientation classes.

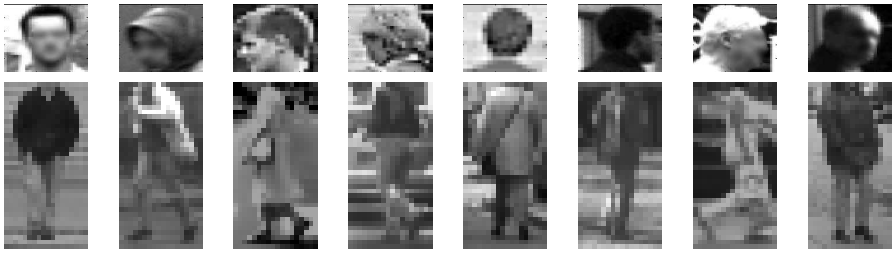


Figure 5.6: Examples of head and body training images in 8 aggregated orientation classes.

For the background detector $f_{-}(z)$, all background samples are trained against all orientation specific samples. For all detectors, multi-layer neural network architectures (NN/LRF) [231] are used with a 5×5 Local Receptive Field (LRF) size applying one hidden layer. The head is extracted by a fixed aspect ratio of 15% of the whole body centered below the highest labeled pedestrian contour point (contour labels were already available from an earlier pedestrian segmentation study). The body detectors use the lower 85% part of the pedestrian bounding box. All head samples were scaled to 16×16 pixels for training and testing, whereas the body samples were scaled to 18×36 pixels. A border of 2 / 4 pixels was added to head / body samples to avoid border effects. Additionally, eight samples are generated from each original sample by shifting the corresponding bounding box by 1-2 pixels.

While [36] showed that using responses from a discriminative classifier in the observation model is possible, they recommend to use Platt-Scaling [36] to calibrate the classifier responses beforehand to gain calibrated classifier predictions. The study by Niculescu-Mizil and Caruana [170] showed that NNs already yield well calibrated predictions and do not profit from post-calibration by e.g. Platt's method. Post-calibration can even hurt here, since the sigmoid at the output of neural nets already calibrates the predictions [170], and outputs can be regarded as scaled estimates of the posterior probabilities of the target class.

5.5.3 Parameter Setting

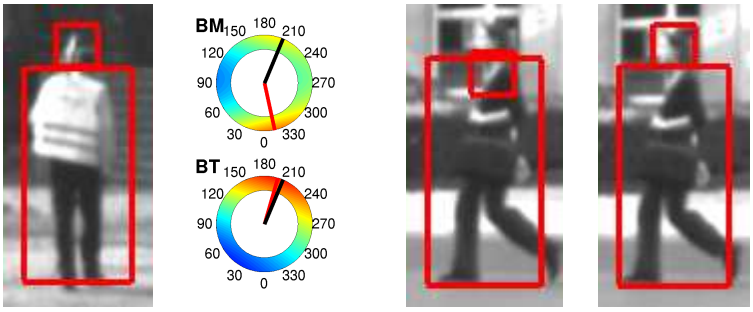
Since the proposed method use discrete GT orientation steps, time slots in the validation set were manually annotated if they contained a head, or a body movement. With these marked time slots, the temporal constraints have been estimated for head [κ_{hh}, Equation (5.4)] and body [κ_{bb}, Equation (5.5)] from differences of GT orientations between adjacent time steps. Anatomical constraints (κ_{hb}, κ_{bh}) are estimated on displacement cases (difference between head and body GT orientation greater than $45°$) on the complete validation set. For a set of pedestrian velocities $k \in \boldsymbol{V}$ (where $k = \{0, 0.1, \cdots 2\}\frac{m}{s}$) the concentration κ_{bv}^{k} is estimated on differences between inferred velocity direction and GT body direction on the complete validation set. Based on these concentration-velocity tuples, parameters of the logistic growth model [$\boldsymbol{\theta}$, Equation (5.6)] are estimated by keeping $\xi_t = 1$ constant. Mixture weights (α_{hh}, α_{bb}, α_{hb}) are tuned on the validation set. Concentration parameters [κ_{o^H}, κ_{o^B}, Equation (5.10)] are estimated on the training set using the 16 GT orientations. The interval between the canonical orientations of left and right neighbored detector is considered as the domain where these detector concentrations are estimated. More specifically, the distribution of the average belief of a detector is considered when presented with samples with GT orientation from this domain. Disparity based region prior parameters for head / body [$\boldsymbol{C}^H$, $\boldsymbol{C}^B$, Equation (5.18)] are estimated on image distances between estimated and GT head / body centers, normalized by GT pedestrian (bounding box) height. Joint region prior parameters [$\boldsymbol{\mu}^D$, $\boldsymbol{C}^D$, Equation (5.19)] are estimated on image distances between head and body GT bounding boxes, for each body orientation class o^B, normalized by GT pedestrian (bounding box) height.

Priors $p(\Omega)$ and $p(V)$ are modeled with uniform distributions for head and body orientation.

5.5.4 Results

To illustrate the method, Figure 5.7a shows a sample that gives a multimodal likelihood estimate, caused by confusing opposite directions. This ambiguity can be successful corrected by the proposed joint tracking approach. The effect of integrating the Pictoral Structure (PS) spatial constraint (Section 5.4.3.2) is shown in Figure 5.7b where localization of head and body is improved with the spatial PS constraint (right image).

Figure 5.8 shows an example of a tracked pedestrian with estimated head



(a) Multimodal likelihoods (b) PS constraint

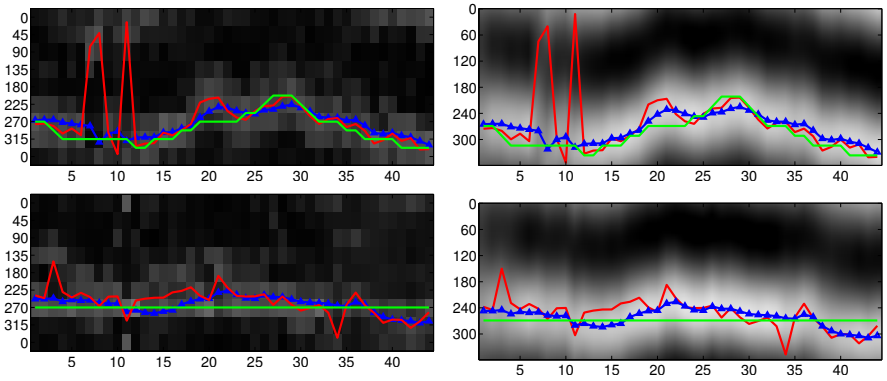
Figure 5.7: (a) Accurate localization of head and body (left) can still lead to multimodal likelihood estimations, here shown for the body (BM) (top right). In the tracked body posterior (BT) (bottom right) this ambiguity is resolved. Maximum likelihood / posterior estimate (red line) and GT orientation (black line) are shown. (b) Integrating the PS constraint (right) results in a better localization of head and body.

and body region displayed in image and disparity map. Figure 5.9 shows the corresponding detector outputs and filtered orientation distributions, over time, for the situation in Figure 5.8. The tracker is successfully smoothing over outliers and is able to react also to small changes in the orientation.

Figure 5.10 shows disparity and gray image of every sixth frame for four estimated tracks with continuous estimation results of the proposed approach. The joint tracking delivers good localization and a robust continuous orientation estimate for head and body. Even in cases with limited stereo support for the head (e.g. first row, fourth and fifth image), the head is localized correctly thanks to the detector outputs. The last track illustrates various problem cases of the proposed approach causing a wrong localization of head and body and therefore a bad orientation estimation. Reasons for that are stronger deviations of mean head position and rotation (second image), pedestrian groups (fifth image), or contrast and lighting issues.

(a) Left image (b) Disparity

Figure 5.8: Sample track showing: (a) Rectified gray value image and (b) disparity image. The red boxes show the estimates for the head and body region.



(a) Head and body detectors (b) Joint tracking of head and body

Figure 5.9: (a) Orientation specific output of head detectors (top) and body detectors (bottom) over the track time (frames) for the scenario shown in Figure 5.8. Brighter values indicate a higher detector confidences. (b) Continuous posterior distributions estimated by joint tracking of head (top) and body (bottom) over the track time (frames). In (a) and (b), labeled GT orientation (green line), single-frame estimation with PS (red line) and joint tracking result (blue line) are shown.

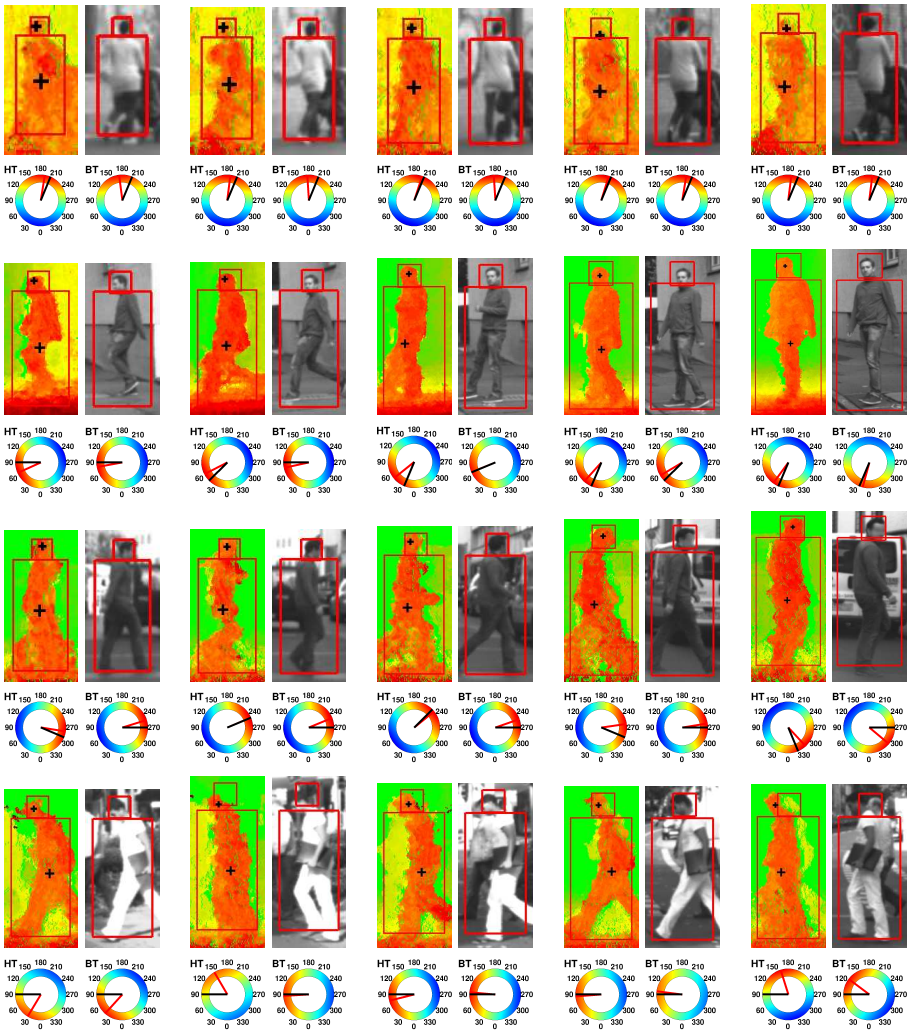


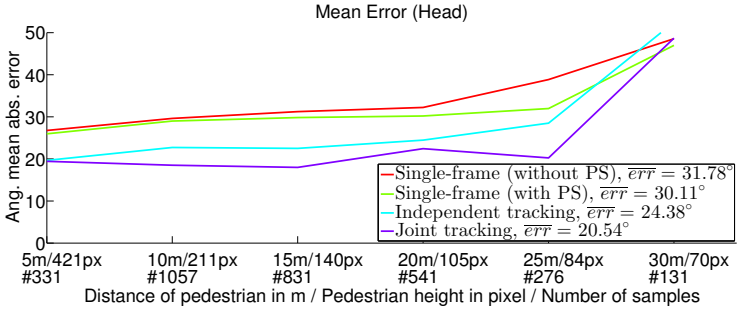
Figure 5.10: Disparity (left) and gray image (right) of every sixth frame of four estimated tracks. The red boxes show again the selected head and body region. Below the images the posterior distributions are visualized for the head (HT) and body orientation (BT) and maximum posterior estimate (red line) and GT orientation (black line) is shown. Black crosses in disparity images denote estimated head and body center. The last track (last row) shows bad estimation results due to inaccurate localization of head and body.

A quantitative evaluation is performed on the complete test set using all 65 valid, estimated tracks. In Figure 5.11, the angular mean absolute error for head and body orientation estimation is shown for an increasing distance of the pedestrian. The averaged error over all distances is shown in the legend. The proposed joint tracking approach is compared to the results of independent tracking and single-frame orientation estimation with and without PS [see Equation (5.19)]. Independent tracking refers to tracking of head and body without an orientation coupling as defined in Equation (5.2). For both independent and joint tracking, the spatial PS constraint is used. As we can see, the mean error can be significantly reduced by tracking. Joint tracking decreases the head / body orientation error over all samples by $11°$ / $15°$ compared to single-frame estimation without PS. This benefit is mainly caused by the removal of outliers compared to single-frame estimation (e.g. confusion between opposite body directions, which visually can look very similar). Furthermore in comparison to independent tracking, the error is decreased by $4°$ / $4°$ for head / body orientation. Anatomical and movement constraints within tracking as defined in Equation (5.2) help here to reject impossible configurations between head and body orientation.

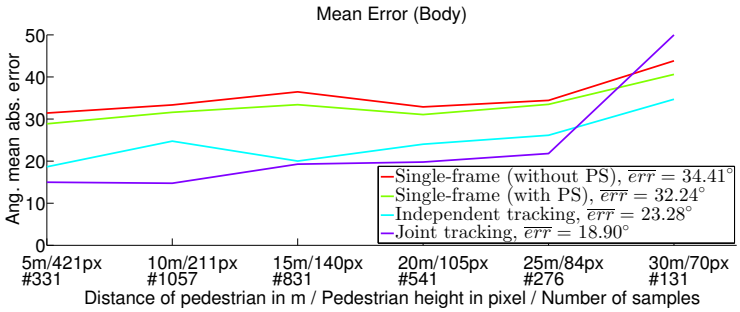
By evaluating only displacement cases (difference between head and body orientation greater than $45°$) the angular mean absolute error turns out to increase by $7°$ / $1°$ for head / body in the joint tracking case. Above 25 m the mean error and the outlier rate increases significantly due to weak detector responses at initial tracking states. Separate experiments (not included here) indicated that a further improvement of the pedestrian track quality (up to the use of actual GT tracks) did not lead to an appreciable improvement of orientation estimation. This can be seen as a strength of the stereo-based head and body localization procedure which can already handle the displaced pedestrian bounding boxes at this quality level.

To get a better impression of the orientation error distribution, Figure 5.12 shows an additional boxplot. It can be seen that adding more constraints reduces the uncertainty and outliers.

Figure 5.13 shows confusion matrices for the single-frame case without PS and for the joint tracking case, for head and body orientation, respectively. Adding the spatial and orientation constraints between the body parts and the temporal filtering results in a clear more diagonal structure of the confusion matrix (i.e. comparing Figure 5.13a with Figure 5.13c and Figure 5.13b with Figure 5.13d). Some hightened confusion can still be observed in Figure 5.13d for the body around $0°$ and $180°$.

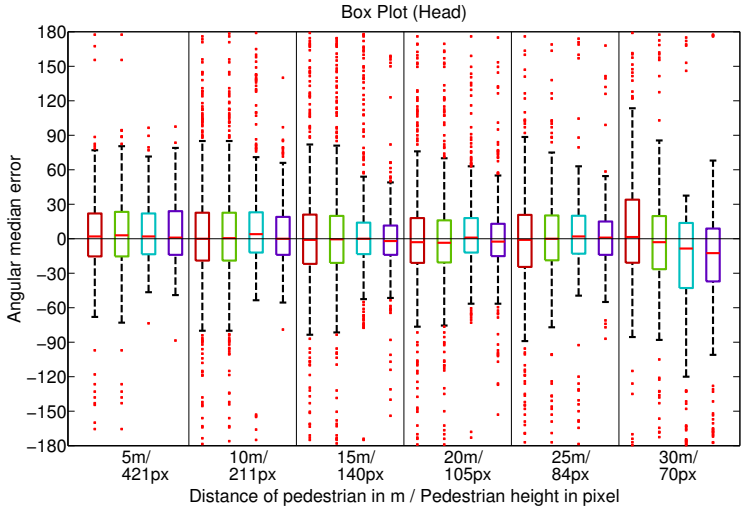


(a) Angular mean absolute error for head (in degrees)

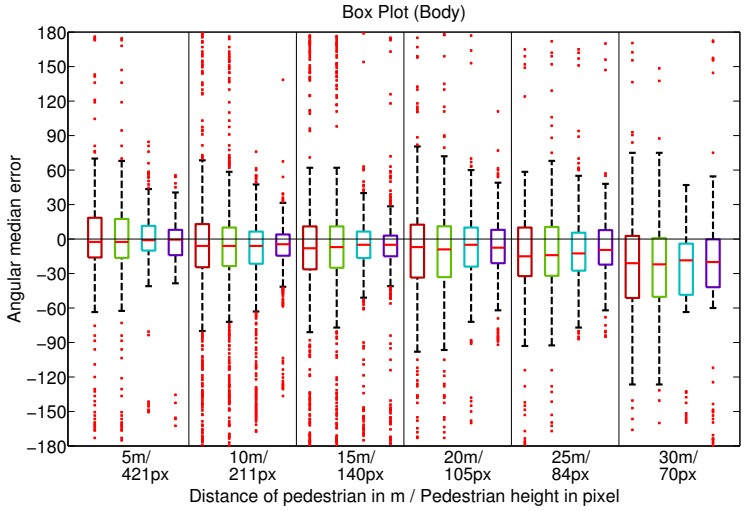


(b) Angular mean absolute error for body (in degrees)

Figure 5.11: Angular mean absolute error over increasing distance for (a) head and (b) body orientation estimation, using joint (purple), independent tracking (cyan) and single-frame orientation estimation with (green) and without (red) PS. Legend is also showing mean error over all samples ($\overline{err}$).

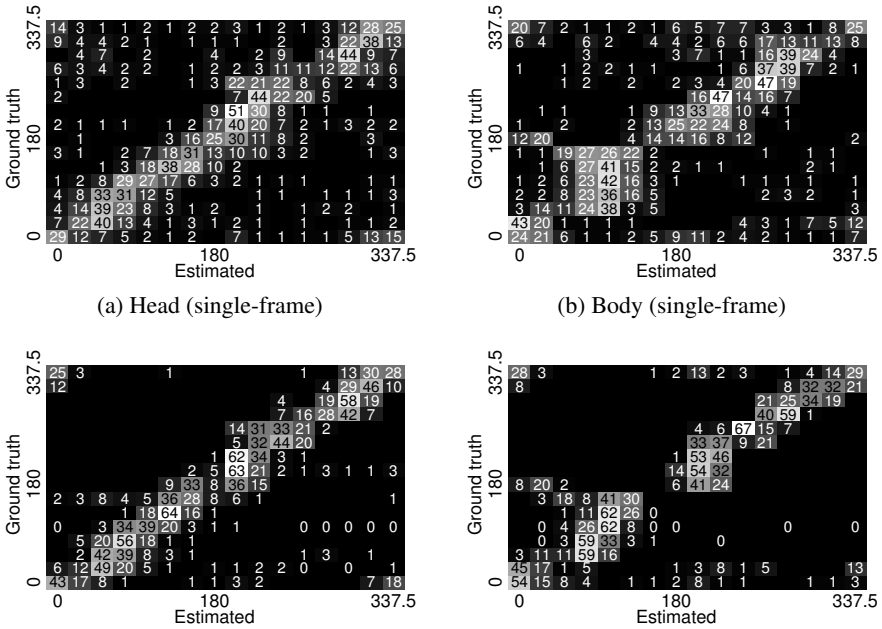


(a) Boxplot for head (median error in degrees)



(b) Boxplot for body (median error in degrees)

Figure 5.12: Boxplots showing median error (red line) and outliers (red crosses) for (a) head and (b) body orientation estimation in case of single-frame estimation without PS (red box), with PS (green box), independent tracking (cyan box) and joint tracking (purple box). Boxes contain 50% of samples. Used whiskers define 99.3% data coverage. By joint tracking a more robust estimation is achieved compared to single-frame estimations and independent tracking.



(a) Head (single-frame)

(b) Body (single-frame)

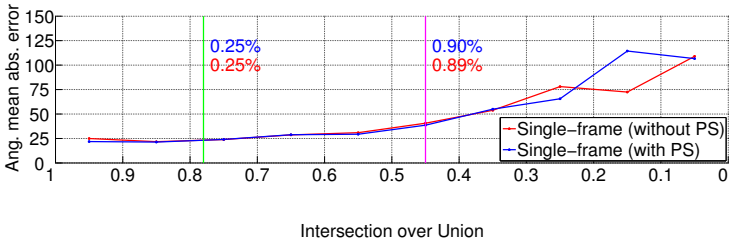
(c) Head (tracked)

(d) Body (tracked)

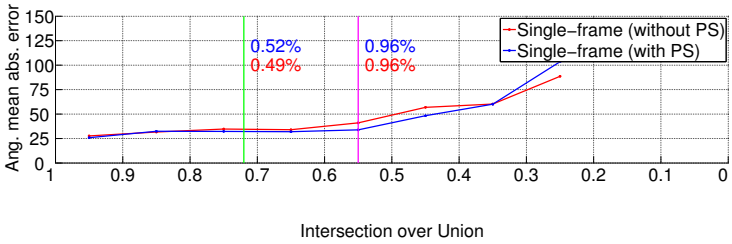
Figure 5.13: Confusion matrices for the single-frame case without PS for head (a) and body (b) orientation, respectively. Idem for joint probabilistic tracking, for head (c) and body (d) orientation, respectively. Occurrences with brighter color are more frequent. Numbers define the occurrences in percent.

Figure 5.14 shows how the angular mean absolute error is affected by image localization performance on the test set. The Intersection over Union (IoU) measure is computed between GT and estimated bounding box such that a value of 1 corresponds to a perfect overlap and a value of 0 corresponds to no overlap. Figure 5.14a shows that 90% (89% without PS) of the head samples have a localization performance better than 0.45 (magenta line), while still getting acceptable orientation estimates. The green line shows the computed localization performance threshold regarding a possible shift of 1 pixels for a 16×16 image in each direction, as done in the training to increase the amount of training samples. In practice,a lower localization performance (as showed by the magenta line) can be accepted. There are only a few samples with an IoU measure less than 0.2, resulting in the mean absolute angular error to be noisy. Figure 5.14b shows the same for the body. Relying only on single-frame measurements regarding head and body locations is suboptimal and it can be expected that integration over time introduces additional robustness here.

The implementation, running on a 3.33 GHz i7-CPU processor, needs on average less than 120 ms per image (this is down from 1 s per image in [80] thanks to multi-threading within each module). Figure 5.15 shows how this time is distributed among the different components used in the system. Starting with a pedestrian detector (Ped. Detection), a 3D world estimation (position and segmentation) is performed using stereo-vision (Ped. 3D Processing) afterwards. The pedestrian is then tracked by a standard Kalman filter (Ped. 3D Tracking). As described in this work, orientation configuration of head and body is estimated (Head Orientation / Body Orientation) using the PS Model and tracked by a particle filter (Orientation Tracker) using various constraints. Note that the head orientation estimation needs more time, since more region hypotheses are generated on average than for the body orientation estimation.



(a) Influence of head localization accuracy on head orientation error (degrees)



(b) Influence of body localization accuracy on body orientation error (degrees)

Figure 5.14: Angular mean absolute error (in degrees) over decreasing image localization accuracy (Intersection over Union) for (a) head and (b) body. The plot shows the theoretical maximum localization error (green line) compared to the still acceptable (observed) maximum localization error (magenta line). Percentages at the borders indicate the amount of test data having smaller localization error for single-frame measurements without (red) / with PS (blue).

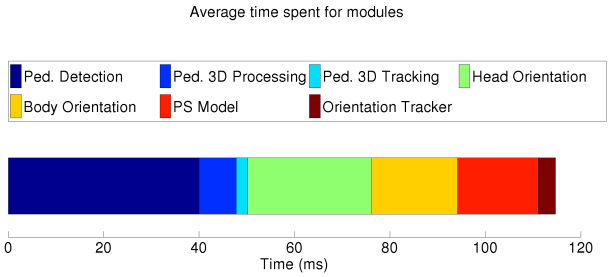


Figure 5.15: Different modules and their running time. Altogether the modules need on average approx. 120 ms per image.

5.6 Discussion

Given input pedestrian tracks of some decent quality (cf. Section 5.5.1), the proposed approach results in a mean absolute orientation error on pedestrian head and body orientation which is fairly constant up to a distance of 25 m, namely about 21° and 19°, respectively (cf. joint tracking in Figure 5.11). As this is one of the first works on pedestrian head and body orientation estimation from a mobile stereo vision platform, a direct comparison to other work is difficult due to the lack of appropriate datasets. It should be noted, that Rehder *et al* [193] report with joint tracking a mean absolute orientation error of 19° and Chen and Odobez list pedestrian head / body orientation errors of 36.0° / 35.6°, 30.0° / 29.4°, 23.6° / 23.6°, 18.4° / 17.4° in their four surveillance scenarios with static monocular camera (cf. [38], Table 1).

The experiments used a 0.75 megapixels camera which is already integrated into production vehicles on the market. In the foreseeable future, when the image resolution is doubled, the distance range for which the current performance is obtained can be extended from 25 m to 35 m, without any algorithm modification. This is a sensible range for a practical application, considering typical urban vehicle speeds and prediction horizons that are not likely to exceed 2 s, given the high pedestrian maneuverability.

Apart from using higher resolution images, performance benefits can be expected from a larger training datasets (e.g. using the upcoming TDC-v2 including head and body orientation labels for all VRU types) and from a more accurate head and body orientation localization method, i.e. by an accurate part-based pedestrian segmentation combining top-down knowledge (shape, texture) with data-driven cues (e.g. by the work presented in Chapter 4).

Recently, deep CNNs [95, 106, 178] showed great success and efficient feature sharing for a join estimation of multiple tasks. As suggested in [28], object orientation can be solved jointly with the detection. For incorporating the detection and orientation estimation of relatively small body parts (e.g. head), an adaption of the net architecture and the used feature map sizes becomes necessary.

The now following Chapter 6 presents a real world vehicle integration, demonstrating that we can expedite driver warning and vehicle braking significantly using head and body orientation cues within a context-based path prediction [133].

6

SYSTEM INTEGRATION

Predicting the motion of objects is a key challenge for various intelligent systems applied in surveillance and mobile robotics. Intelligent vehicles need to perform a reliable prediction of all moving objects on the street to react and drive safely. Unfortunately, the motion of VRUs is particularly difficult to model, since they can change their motion in a whim (e.g. stop / start walking suddenly or turn around). In the previous chapters various steps have been taken to improve VRU detection (Chapter 3) and segmentation (Chapter 4). Finally, it was shown how head and body orientation of the VRU can be extracted in a robust way (Chapter 5). This chapter now focuses on a joint and concrete application of these concepts for the automotive domain. In particular, extracted visual VRU context cues (e.g. by means of the head orientation) are used to improve VRU path prediction.

6.1 Introduction

Currently, there can only be a rough guess how VRU motions are modeled within self-driving vehicles from Tesla, Waymo or Uber [148]. To the author's knowledge there has been no demonstration yet clarifying if there are more sophisticated algorithms involved to predict the path of a VRU or if only standard

methods are used. In cases where the object motion is more or less linear, standard filters (e.g. a Kalman Filter) and simpler path prediction methods (e.g. supposing a constant velocity movement) can be applied. Unfortunately, VRUs can change their motion in a whim and therefore the movement can be highly non-linear. This is especially the case in heavily urbanized areas, when there are massive implicit and explicit interactions between traffic participants. And even if multiple motion models are used to model different motion types, e.g. by utilizing a Switching Linear Dynamic System (SLDS) [20, 196], a dynamic switch to other motion types is only estimated if there is enough evidence by positional observations which contradict the current dynamic model. Thus, standard methods will fail to predict a quick dynamic change (e.g. stopping, start walking) in time, leading to a higher false alarm rate when greater prediction horizons are used. Instead of waiting until the dynamic of an object is actually changing, one can try to anticipate the dynamic switch by using VRU context cues.

Making use of this idea, this Chapter presents a realtime system implementation of the context-based Switching Linear Dynamic System (C-SLDS), developed in [133], integrated within a complete stereo-vision system. The Chapter demonstrates that anticipating the VRU behavior by using extracted visual context cues, will indeed lead to an improved VRU path prediction. The system has been integrated in two vehicle demonstrators. Public online demonstrations of the system integrations provided valuable feedback and insights. The implemented system has been further used to model the interaction between the driver and the pedestrian [197] within an extended C-SLDS formulation.

Accident analysis shows that the scenario of a crossing pedestrian accounts for a majority of all pedestrian fatalities in traffic [155]. Therefore, the presented system implements the pedestrian scenario discussed in [133], where a crossing pedestrian might stop or continue walking into the driving corridor in front of the vehicle. In particular, pedestrian's situational awareness is assessed by the head orientation estimation as described in Chapter 5 to anticipate pedestrian's behavior. Based on the practical system integration, implications for a modified vehicle intervention strategy (i.e. earlier warning, braking or evasion) are pointed out based on the improved path prediction. Figure 6.1 visualizes the available extracted cues and the final inferred path prediction making use of these cues.

The main contributions of this Chapter can be summarized as:

- A realtime system integration is presented making use of the Context-based Switching Linear Dynamic System (C-SLDS) formulation described

by Kooij *et al* [133] to enhance path prediction. The scenario of a crossing pedestrian is chosen to demonstrate the benefit of VRU behavior cues integrated in the C-SLDS approach. In particular the head orientation from Chapter 5 is used to model pedestrian's awareness. The realtime system includes an intervention strategy based on the predicted collision probability.

- The system output is evaluated on different scenarios of crossing pedestrians. Experiments investigate how increasing the prediction horizon affects the false alarm rate. It is shown that the prediction horizon can be increased up to nearly $1\,\mathrm{s}$ without increasing the false alarm rate.

- For presenting the system, a demonstration design has been created, making a realistic experience of the system possible. Online, the system is compared to a simpler Linear Dynamic System (LDS) to show the benefit for an improved acoustical driver warning.

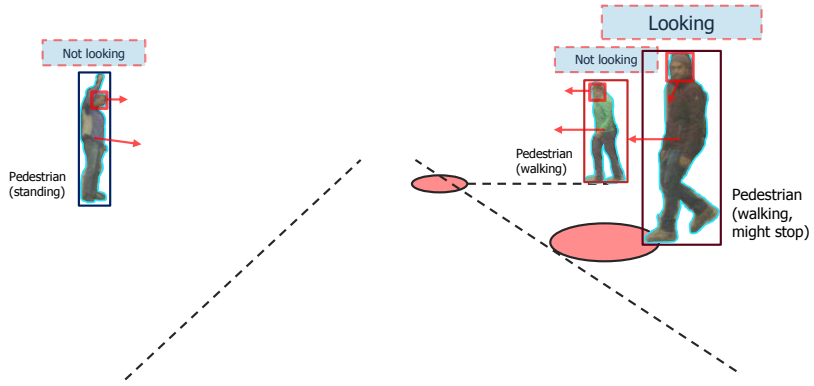


Figure 6.1: Improved path prediction using VRU context cues. In particular, based on a given bounding box detection and orientation estimates of head and body, pedestrian's awareness is estimated. Helped by additional environment context cues, VRU context cues allow a better modeling of the future motion path of a pedestrian, i.e. the prediction horizon can be increased without increasing the false alarm rate.

6.2 Approach

The scenario of a laterally crossing pedestrian is selected for the system integration. In this scenario it is argued that the pedestrian's decision to stop is largely influenced by the vehicle on collision course, the pedestrian's awareness thereof, and the position of the pedestrian with respect to the curbside. A joint application of those cues helps to better estimate the current motion state (walking or stopping) of the pedestrian. That is where a C-SLDS is applied in [133].

Section 6.2.1 will shortly recap the C-SLDS formulation developed in [133]. Section 6.2.2 will then provide a more detailed description of the used VRU context cues than [131, 133]. In addition to the described pedestrian context cues, a possible extension to the cyclist case [131] is also proposed. Section 6.2.3 then draws the complete picture of the implemented and integrated realtime system, making use of pedestrian context cues within the C-SLDS formulation to improve path prediction for the crossing pedestrian.

6.2.1 Context-based Switching Linear Dynamic System

Let X_t represent the latent dynamic state of the VRU at time t, consisting of the VRU's lateral and longitudinal position, and its velocity in both directions. Y_t is a vector containing the measured VRU lateral and longitudinal position at time t. Measurements are provided by a VRU detector, e.g. as described in Chapter 3. Localization in 3D space can be even improved by an accurate segmentation of the VRU, e.g. as described in Chapter 4.

Within a C-SLDS, context information provided by some latent variables Z_t can be used to actively influence the switching state M_t to choose the appropriate transformation matrix $A^{(M_t)}$ (i.e. the appropriate dynamic model) in a SLDS [20, 196] process:

$$X_t = A^{(M_t)} X_{t-1} + B\epsilon_t \qquad \epsilon_t \sim \mathcal{N}(0, Q) \tag{6.1}$$

$$Y_t = CX_t + \eta_t \qquad \eta_t \sim \mathcal{N}(0, R). \tag{6.2}$$

The current state X_t is gained by a linear transformation of the previous state X_{t-1} by $A^{(M_t)}$, with introduced process noise $\epsilon_t \sim \mathcal{N}(0, Q)$ added through linear transformation B. Observation Y_t results from a linear transformation C of the true state X_t with additional measurement noise $\eta_t \sim \mathcal{N}(0, R)$.

Kooij *et al* [133] suggest the use of four binary latent context cues within the C-SLDS formulation to anticipate the behavior of a VRU:

- Z_t^{STAT}: The static environment context models the location of the VRU with respect to the main infrastructure. Z_t^{STAT} is true, iff the VRU is in an area where dynamic changes typically occur.
- Z_t^{DYN}: The dynamic environment context models the presence of other traffic participants. Z_t^{DYN} refers, for example, to a possible collision course of the VRU with the ego-vehicle.
- Z_t^{ACT}: The VRU behavior context indicates if the VRU's current action provides insight in the VRU's intention or behavior.
- Z_t^{ACTED}: This last context serves as a memory for Z_t^{ACT} and is true iff the VRU performed behavior relevant actions in the past. It encodes simply a logical OR between the Boolean nodes Z_{t-1}^{ACTED} and Z_t^{ACT}.

Figure 6.2 shows all variables as nodes in a graphical representation of the DBN [133]. The arrows indicate that child nodes are conditionally dependent on their parents. The dashed arrows show conditional dependency on the nodes from the previous time step. Measurements $E_t = \{E_t^{STAT}, E_t^{DYN}, E_t^{ACT}\}$ provide evidence for the latent context variables Z_t through conditional probability $p(E_t|Z_t)$. The exact posterior of the DBN formulation includes $|M|^T$ Gaussian modes after T time steps. Therefore, exact inference of the DBN model is intractable. As described in [133], Assumed Density Filtering [19, 159] (ADF) can be used as an approximate inference technique.

6.2.2 Modeling the VRU Behavior Context

This section presents two possible models for the VRU context Z^{ACT}, one for the pedestrian and one for the cyclist. The cues are modeled using the measurements E^{ACT} as developed during this work. The two proposed VRU context cues have then been applied in [131, 133] in a C-SLDS formulation joining static and dynamic environment context cues.

6.2.2.1 Pedestrian's Situational Awareness

In the crossing pedestrian scenario [133], pedestrian's decision to continue walking or to stop (i.e. a switch between two motion models) is largely influenced

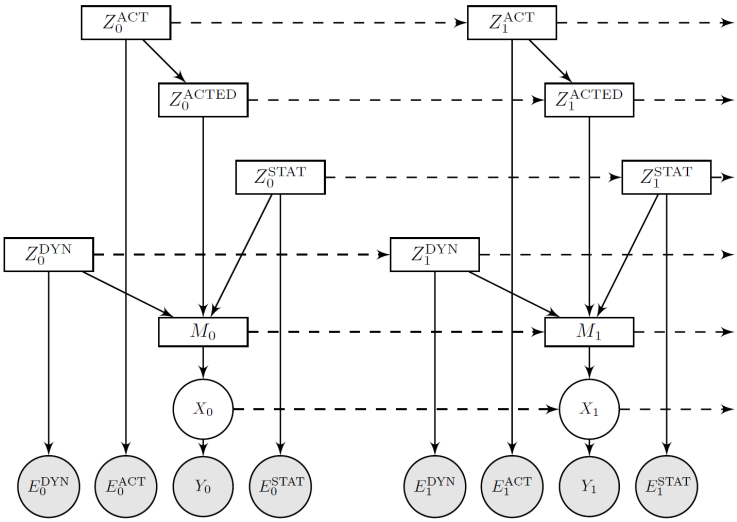


Figure 6.2: Context-based SLDS as directed graph, unrolled for two time slices. Discrete / continuous / observed nodes are rectangular / circular / shaded. Measurements E_t provide evidence for the latent context variables Z_t. The context cues influence the switching state M_t. Figure from [133].

by the position of the pedestrian with respect to the curbside (i.e. the static environment context Z^{STAT}), the existence of an approaching vehicle on collision course (i.e. the dynamic environment context Z^{DYN}) and the pedestrian's awareness thereof (i.e. the VRU context Z^{ACT}) [133].

Pedestrian's awareness can be approximated over his head orientation. Head orientation observables are extracted based on the method described in Chapter 5 with minor modifications. Since the C-SLDS formulation of [133] integrates the latent variables over time, single-frame head orientation estimates are used here as input. For the head localization, candidate regions are generated in the upper pedestrian detection bounding box from an available pedestrian segmentation. To improve the head localization, a pixel-wise segmentation using different cues (including shape priors) as presented in Chapter 4 could be used here. Focusing on a realtime system integration, a simple disparity-based segmentation (as in Section 5.4.3.1) is applied in this particular case. The most likely head image region $z^{\hat{r}}$ is selected from all candidate regions by applying joint region priors as described in Section 5.4.3. The likelihood $p(\boldsymbol{z}|\Omega = o)$

modeling the evidence that the observed image region $\boldsymbol{z}$ contains a head of the discrete orientation class $\Omega = o$, can be efficiently computed up to a constant factor. The final head observation vector $HO = [p(\boldsymbol{z}|\Omega = 0), \cdots, p(\boldsymbol{z}|\Omega = 315)]$ contains the confidences of the selected region, i.e. an (unnormalized) distribution over eight orientation classes. If the head is not clearly observed (e.g. it is too far, or in the shadow), specific orientation evidence is typically low and values are similar to each other. In this case the observed distribution provides less evidence for the true head orientation. This is also supported by the influence of the background detector response f_{-} in Equation (5.16). The computed body orientation is not used as an additional observable in this application, but it helps to select the most likely image region within the applied joint region prior (Section 5.4.3).

The VRU context Z^{ACT} captures whether the pedestrian is aware of the ego-vehicle or not. HO is modeled to be a sample from a Multinomial distribution conditioned on the latent context variable Z^{ACT} [133]

$$p(E^{ACT}|Z^{ACT} = sv) = \text{Mult}(HO|\boldsymbol{\theta}_{sv}), \tag{6.3}$$

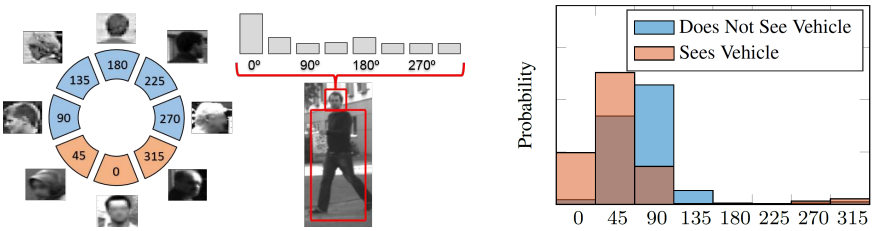
with parameter vector $\boldsymbol{\theta}_{sv}$, where $sv \in \{0, 1\}$ represents the two awareness states, *Does Not See Vehicle* and *Sees Vehicle*.

Figure 6.3 shows the head orientation binning into eight different classifiers responses resulting in the used observation vector HO (Figure 6.3a) and the final modeled Multinomial distributions (Figure 6.3b).

6.2.2.2 Cyclist's Intention

Very similar to the pedestrian case, VRU context cues can also help to model the intention of a cyclist approaching an intersection [131]. The cyclist may or may not turn left (i.e. switch between two motion models) at the intersection. The cyclist can express an explicit intent to do so by showing an arm signal (i.e. the VRU context cue Z^{ACT}). The existence of an approaching vehicle on collision course (i.e. the dynamic environment context Z^{DYN}) and the distance to the intersection (i.e. the static environment context Z^{STAT}) have additional influence on the cyclist's final motion. As for the pedestrian case the environment cues model where and when dynamic changes are most likely happening.

A simple method is presented to extract the cyclist's arm signal from an image, serving as a possible VRU context cue in a C-SLDS. To detect the arm of a cyclist, a multi-modal local representation is used based on shape $\boldsymbol{S}^r$, color $\boldsymbol{I}^r$ and disparity $\boldsymbol{D}^r$ information inside the region r of a tracked bounding box.



(a) Head observation model

(b) Pedestrian's awareness

Figure 6.3: Model for pedestrian awareness; (a) The head orientation is modeled with eight classifier responses resulting in the used observation vector HO; (b) Pedestrian's awareness, i.e. sees or not sees vehicle, is modeled with Multinomial distributions. Figures adapted from [131].

A binary segmentation of the cyclist is estimated based on disparity inside the region r as described in Section 5.4.3.1. With the image locations covered by the segmentation, the cyclist foreground is modeled with a Gaussian Mixture Model (GMM) [19] for the modalities $\boldsymbol{I}^r$ and $\boldsymbol{D}^r$ as described in Section 4.2.1.3 and 4.2.1.4. Three Gaussian components are used for each foreground model. GMMs with five Gaussian components are used to model the background based on all image locations in the region r not covered by the binary segmentation. The Expectation-Maximization [55] algorithm is used to learn the parameters for each GMM. Initialization is done with k-means [19].

Using the binary segmentation again, the rough image locations of the left and right shoulder are estimated by following simple anatomic conditions. Rectangular contour templates are sampled around a shoulder location with variations in length, width and angle. To account for the uncertainty in the location, additional templates are generated by positional jittering. For each contour template a matching score is calculated using Chamfer matching [87] differentiated by gradient direction (in this case: four discretization intervals, not encoding the gradient sign), as described in [94] and also used in Section 4.3.1. All templates $C = c_n$, $c_n \in 1...N$ having a matching score greater than τ_{ch} are used in the following. The final shape likelihood $p(\boldsymbol{S}^r|Arm, C)$ is modeled by a Gamma distribution similar to [87] for $Arm \in \{0, 1\}$. Figure 6.4 shows the different feature representations extracted from a tracked cyclist.

Two naive assumptions are made. First, it is assumed that the color and dis-

Figure 6.4: Shape-based arm detection (left) using simple rectangular templates, supported by learned color (top right) and disparity models (bottom right).

parity appearance of the raised arm is equal to the rest of the cyclist. Therefore the likelihoods $p(\boldsymbol{I}^r|Arm, C)$ and $p(\boldsymbol{D}^r|Arm, C)$ are modeled by the learned GMMs described above and are evaluated on pixel locations inside a contour template c_n. Second, it is assumed that all terms are independent. The likelihood of the model becomes

$$p(\boldsymbol{I}^r, \boldsymbol{D}^r, \boldsymbol{S}^r|Arm) = \sum_{c_n} p(\boldsymbol{I}^r|Arm, C) \times p(\boldsymbol{D}^r|Arm, C) \times p(\boldsymbol{S}^r|Arm, C) \times p(C). \tag{6.4}$$

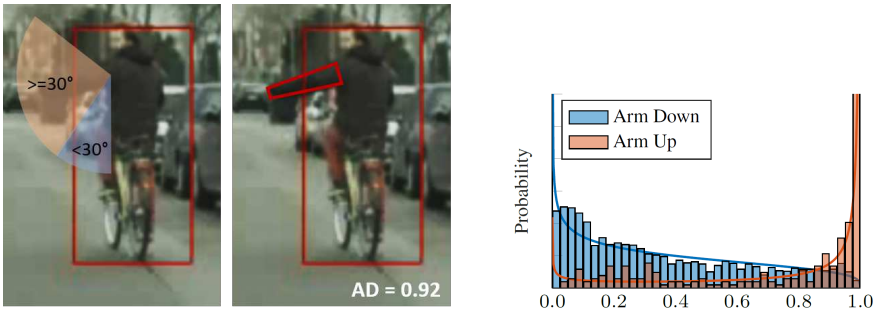
A uniform prior is set for $p(C)$. The arm detection score AD is defined with

$$AD = \frac{p(\boldsymbol{I}^r, \boldsymbol{D}^r, \boldsymbol{S}^r|Arm = 1)}{\sum_{Arm} p(\boldsymbol{I}^r, \boldsymbol{D}^r, \boldsymbol{S}^r|Arm)}. \tag{6.5}$$

The score in Equation 6.5 is not very well calibrated. One reason for that is the weak independence assumption in Equation 6.4, which does not hold in all cases. To gain a better likelihood, AD is modeled as a sample of a Beta distribution conditioned on the latent context variable Z^{ACT} [131] by

$$p(E^{ACT}|Z^{ACT} = ar) = \text{Beta}(AD|\boldsymbol{\theta}_{ar}), \tag{6.6}$$

with learned parameters $\boldsymbol{\theta}_{ar}$ and $ar \in \{0, 1\}$. Here, the VRU behavior context Z^{ACT} captures whether the cyclist raises an arm (ar = 1 for *Arm Up*, ar = 0 for



(a) Arm observation model

(b) Cyclist's intention

Figure 6.5: Model for cyclist intention; (a) Separation for $ar \in \{0, 1\}$ according to manually labeled GT arm angles (left). Final detection and arm detection score AD are used in (b) to model the cyclist intention by Beta distributions.

Arm Down) to indicate intent for a turn maneuver. Figure 6.5 shows an example of the arm detection and the final Beta distributions.

6.2.3 System Implementation

The pedestrian scenario from [133] has been implemented in a Mercedes Benz E-class (W212) test vehicle, using a stereo-vision camera setup (image resolution 1176×640 pixel, baseline 22 cm, frame rate 16.7 Hz), running in realtime (i.e. within one camera cycle). The cyclist case can be integrated in a similar way (after adaption of the latent context cues). Figure 6.6 shows an overview of the implemented components for the pedestrian scenario.

6.2.3.1 Disparity and Stixel Estimation

Standard *Image Preprocessing* steps are applied for left and right images, including a color correction, debayering and rectification of images. *Disparity Estimation* is performed on a FPGA [89] using Semi Global Matching [109]. The *Stixel World Estimation* [182] is computed based on the disparity map using all CPU cores in parallel.

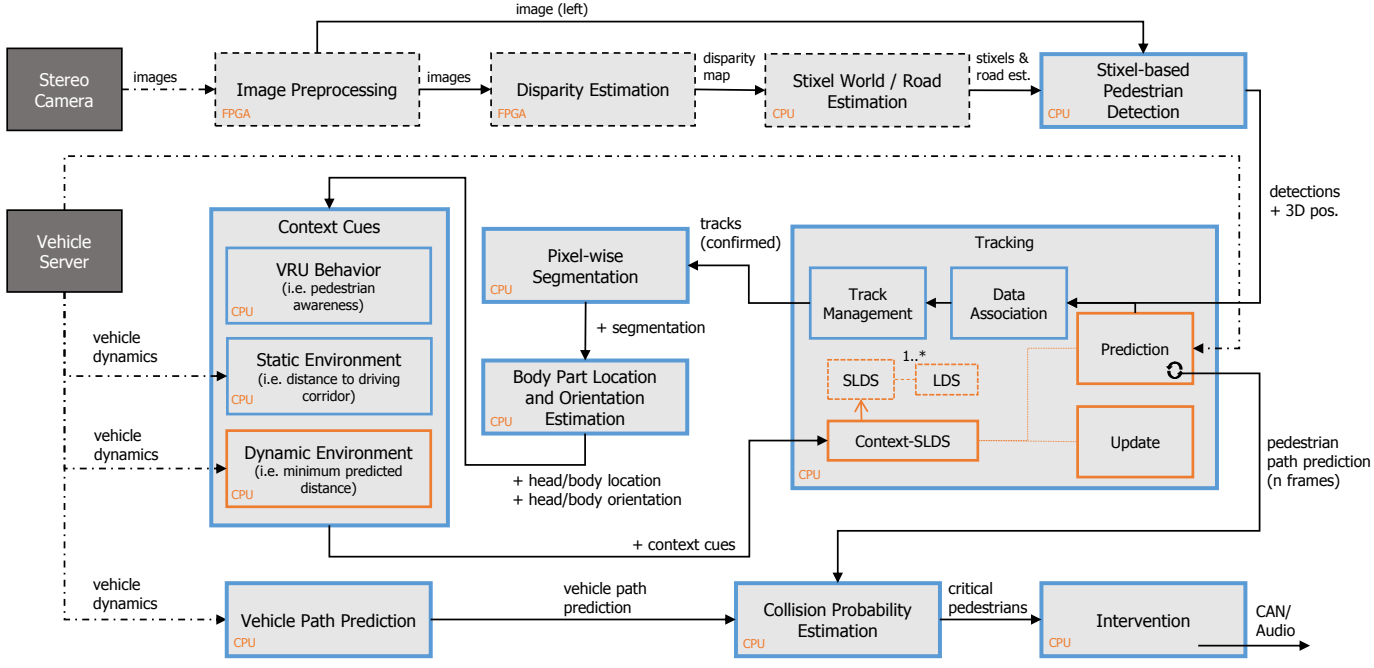


Figure 6.6: Realtime system architecture for the C-SLDS system implementing the crossing pedestrian scenario. Blue-rimmed algorithm modules have been implemented during this work. Black-rimmed modules existed in advance and have been kindly provided by the respective authors. These modules have been slightly modified / parameter-tuned. Orange-rimmed modules have been ported from the Matlab implementation provided by the authors of [133].

6.2.3.2 Detection

The resulting stixels are used with the left preprocessed image to detect pedestrians within the *Stixel-based Pedestrian Detection* module. Realtime capabilities for pedestrian detection are reached by using the efficient stixel-based proposal generation (described in Chapter 3). Furthermore, instead of using the slower FRCN architecture from Chapter 3, a shallow CNN architecture is applied using only one hidden layer (similar to the one used in [66, 231], Figure 6.7). Local

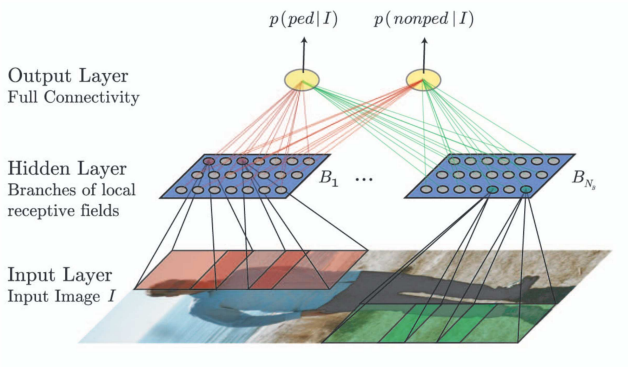


Figure 6.7: Shallow CNN architecture as used in [66, 231]. Image from [66].

receptive fields of a size 5×5 are shifted over the rescaled proposal image cut outs (36×18 px including a two pixel border) with a stride of two pixels. 48 feature branches are used.

For each detected pedestrian, the position (i.e. lateral and longitudinal position in meters) in the vehicle coordinate system (i.e. origin at center bottom of vehicle's rear axis) is estimated based on the disparity support for this detection, i.e. by using the median over all disparity values in the bounding box. Optionally, a more accurate 3D estimation can be gained by an accurate pixel-wise segmentation, as for example described in Chapter 4.

6.2.3.3 Tracking and Path Prediction

Tracking is performed in the vehicle coordinate system. The ego-motion of the vehicle needs to be taken into account when combining different measurements from different time steps. Therefore, at each time step, the predicted position

is transferred from the last vehicle coordinate system to the current vehicle coordinate system using the vehicle dynamics (i.e. velocity and yaw rate within an affine transformation). Based on the ego-motion compensated, predicted pedestrian position, *Data Association* is done on all (at this time step) available bounding box detections. For each existing track, a validation gate [13] is defined to preselect appropriate detections for this track. The size of the elliptical gating region is defined by a gating threshold γ which is set according to the inverse Chi-Square cumulative distribution at significance level ρ [13]. Thus, the validation gate defines a region of acceptance such that $100(1-\rho)$ of true detections are rejected. From the detections in the validation gate, only a single detection with the smallest Mahalanobis distance [152] to the predicted position of the track is associated [13]. For all tracks, measurements are associated in a greedy manner to exactly one track. If there is no measurement left in the validation gate, no measurement is associated to this track. A track can reach four states during its lifetime: *initialized*, *preliminary*, *confirmed* and *deleted*. The management of these states is the task of the *Track Management* module. At first, each not associated measurement is kept as a initialized track. It gets a preliminary track if the track score becomes greater than τ_{tpre}, and a confirmed track after this state has been hold for at least T_{tconf} time steps. It gets deleted if the track score sinks below a predefined threshold τ_{tdel}. A track score decreases when there are no measurements associated to a track. The track score ξ^t at time t is calculated based on the low pass filtered detection scores c^t_{det} of associated measurements, i.e. $\xi^t = \alpha \times c^t_{det} + (1-\alpha) \times \xi^{t-1}$. Two hysteresis thresholds (τ_{tha} and τ_{thb} with $\tau_{tha} > \tau_{thb}$) are used to better model the track score and different values for $\alpha \in \{\alpha_{up}, \alpha_{low}\}$ are used where $\alpha_{up} < \alpha_{low}$. The first time the track score reaches τ_{tha}, α_{up} is used instead of α_{low} in the low pass filter, until the track score sinks under τ_{thb} again.

For the pedestrian case, two motion models are used to cover the dynamics of the pedestrian. Walking is modeled with a constant velocity model [20], while a stopping pedestrian is covered by an additional constant position model [20]. The update of the track state is performed by incorporating the available information from all latent context cues (as described in [133]). Assumed Density Filtering is used as an approximate inference technique and after each update the track state is reduced to $|M|$ distributions again (collapse) [133].

6.2.3.4 Context Cues

Context cues are calculated for each new associated pedestrian detection. First, a *Pixel-wise Segmentation* is estimated (Section 5.4.3.1) to support the following *Body Part Localization and Orientation Estimation* as described in Section 6.2.2.1.

The dynamic environment context is based on the minimum distance D_{min} reached between the pedestrian and the vehicle if both would continue with a constant velocity. The distribution over D_{min} given Z^{DYN} is modeled as a Gamma distribution as described in [133].

The static environment context is modeled in [133] with the distance to the curb ridge. The curb ridge is estimated from the image. For the practical implementation in the test vehicle, the predicted vehicle path corridor serves as a replacement and adequate approximation for the curb ridge estimation. The *Distance-To-Curb* (DTC) is then approximated by the difference between the expected filtered position of the pedestrian and the nearest point on the driving corridor border (left or right side). The distribution over the approximated DTC given Z^{STAT} is modeled as a Normal distribution [133].

The VRU behavior context models pedestrian's awareness of the situation, which is approximated based on the head orientation (Section 6.2.2.1). If a pedestrian has seen the approaching vehicle in the past, he might be still aware of the vehicle, although the current evidence is not supporting that. This awareness memory is modeled by the latent context Z^{ACTED} [133].

6.2.3.5 Collision Probability and Intervention

The collision probability is calculated based on the intersection of the pedestrian's predicted probability density and the vehicle's driving corridor (i.e. the probability mass inside the driving corridor). The distribution over the future pedestrian position $\overline{P}(X_{t+t_p}|Y_{1:t})$ is estimated by repeating the inference steps t_p times (i.e. prediction horizon). However, for path prediction the update step only includes the static environment cues, since this cue can also be calculated based on the predicted pedestrian state. All other measurements are not existent and treated as missing. Incorporating the static environment context in the path prediction, makes it still possible to estimate when the change in dynamics will occur [131, 133].

Warnings get activated when the time to collision (TTC) becomes smaller than a defined threshold T_{ttc}. An acoustic warning of a module is finally sent

out when the collision probability exceeds the threshold of τ_{col}. In this implementation the collision probability is calculated based on the lateral pedestrian dynamic components only.

6.2.3.6 Timing

Image preprocessing and disparity estimation together need a full image cycle (i.e. about 60 ms) and are pre-calculated on subsequent images on an external device using an FPGA-based implementation, introducing a delay of one frame in the whole system. In parallel, computations are performed on the host machine on the previous image. Stixel world estimation runs in 30 ms on average, while the stixel-based pedestrian detection needs about 22 ms (computed on eight available CPU cores in parallel). Pedestrian segmentation, part localization and orientation estimation need together about 7 ms for each pedestrian. Each of the eight head orientation CNNs runs on a separate CPU core and thus, the eight orientation estimates can be computed in parallel for each head region. Computation times and resources needed to compute the latent context cues and to perform tracking and collision probability estimation are neglectable.

6.3 Evaluation

Using the implemented system described in Section 6.2.3 this section now shows the benefit of gaining greater prediction horizons by applying VRU behavior cues. Evaluations are performed on different scenarios of crossing pedestrians. Finally a live demonstration design is presented, which shows the benefit of the system also in comparison to a standard system not using context cues.

6.3.1 Parameter Setting

Head orientation estimation (Section 6.2.2.1) has been trained and evaluated separately as described in Section 5.5. As reported in Section 5.5, the mean absolute angular error based on a single-frame head orientation estimation (with applied Pictoral Structure) is around $30°$. Ground Truth (GT) for $Z^{ACT} = sv$ is taken from labeled GT head orientations. It is assumed that the pedestrian sees the vehicle (i.e. $sv = 1$) if the GT head orientations is in a range of $[-45°, 45°]$ (with $0°$ the head points towards the camera). For GT orientations not in the range it is assumed that the pedestrian doesn't see the vehicle (i.e. $sv = 0$). For

each ground truth label sv, class distributions are estimated by averaging the class weights in the corresponding head measurements [133].

Although not shown here, the proposed arm detector (Section 6.2.2.2) can be used in a DBN for cyclist path prediction [131], using a similar model as described for the pedestrian case. The Arm detection has been evaluated on 42 sequences recorded on-board a moving vehicle with a stereo camera setup (image resolution 2048×1024 pixel, baseline 21 cm, frame rate 20 Hz). Each sequence contains one out of 6 different riders. In 15 sequences the riders are performing a hand gesture. With the simple method described above, an average precision of 73.8 is reached for detecting a raised arm (single-frame) based on a given tracked bounding box of the cyclist. GT for $Z^{ACT} = ar$ is available by labeled GT arm angles. It is assumed that an arm is raised (i.e. $ar = 1$) if the GT arm angle is raised further than $30°$. Below $30°$ the arm is considered down (i.e. $ar = 0$) [133].

Other DBN parameters (e.g. environment context models, process noise of models) are estimated, as described in [133], by reusing the 48 "normal" sequences. "Normal" sequences refer to sequences where the pedestrian behaves as expected (i.e. stops in a critical situation when he is aware of the approaching vehicle and continues if he is not aware). The sequences are obtained from on-board a moving vehicle with a stereo sensor setup (image resolution 1176×640 pixel, baseline 22 cm, frame rate 16.7 Hz). Sequence labeling has been performed regarding different interaction levels (critical vs. non-critical), pedestrian behaviors (vehicle seen vs. vehicle not seen) and pedestrian motions (stopping at the curbside vs. crossing) [133]. Furthermore the sequences contain estimated 3D GT positions based on the disparity values covered by the labeled GT bounding boxes. For each sequence also the event tags for $TTE = 0$ (i.e. when the pedestrian reaches the curb) and GT head orientations have been labeled. *Leave-one-out* cross-validation is used to separate training and test sequences.

For pedestrian detection (Section 6.2.3.2), the CNN architecture has been trained as described in [66] using one bootstrapping iteration. The area for stixel-based proposal generation is limited to 35 meters in front of the vehicle and eight meters to each side from the mid of the vehicle. The stixel estimation parameters are tuned to gain around 200 proposals on average (aspect 2:1) on each image. Parameters for data association and track management have been tuned on a separate tracking validation dataset.

6.3.2 Effects on System Intervention

For the pedestrian case, implications for an improved intervention strategy (e.g. for possible warning, braking, evasive maneuvers) are shown.

Figure 6.8 and Figure 6.9 show estimated (averaged) collision probabilities using the 25 "normal" critical test sequences from [133] for an inattentive and attentive pedestrian respectively. The sequences are processed (with different parameter settings due to cross-validation) using the system as described in Section 6.2.3 to gain a collision probability for each track and for each TTE. The first row in Figure 6.8 and Figure 6.9 show collision probabilities for various collision horizons over the time to event (TTE). Yellow color corresponds to a high and blue to a low probability of collision. Row 2 / 3 / 4 in both figures show the average probability of collision for a prediction horizon of 1 / 8 / 16 time steps (frames) ahead. The left columns in Figure 6.8 and Figure 6.9 show predicted collision probabilities for a C-SLDS (including VRU behavior combined with static and dynamic environment context cues). The right columns show the same for standard LDS with a single constant velocity model. In the following, a collision is supposed to be detected when the calculated collision probability exceeds a collision threshold of $\tau_{col} = 0.8$ (red solid line in both figures).

In the first scenario (Figure 6.8), the pedestrian does not look at the vehicle and crosses the road in front of the vehicle. A collision is imminent. Since the LDS (Figure 6.8b) is modeled with a constant velocity model, a high process noise has been learned during training to compensate the various possible motion maneuvers of the pedestrian. Thus, the variance of the collision probability distribution increases fast for a greater prediction horizon, making it impossible to estimate a reliable collision probability. The collision cannot be predicted reliably at $TTE = 0$ (i.e. when the pedestrian reaches the driving corridor), but only 10 to 15 time steps (frames) later ($TTE > 10$). The C-SLDS method (Figure 6.8a) makes use of the context cues to recognize a critical situation with an inattentive pedestrians. The collision probability is already very high, even before the driving corridor is reached ($TTE < 0$). Therefore, a reliable warning based on the collision probability can be given much earlier in time at $TTE = -3$, i.e. 13 time steps earlier compared to the LDS, making a warning 780 ms earlier possible (supposing a camera cycle of 16.7 fps as used in the implementation).

In the second critical scenario (Figure 6.9), the pedestrian is aware of the vehicle and stops before entering the driving corridor. The C-SLDS (Figure 6.9a)

produces no warning since the collision probability density is again correctly estimated. Making use of the additional context cues, a model switch is correctly estimated and the stopping model (i.e. constant position model) gets more weight in this case. The variance of the prediction is much lower than for the LDS, since the switching dynamics of the pedestrian are shared between the used walking and stopping model. Thus, the prediction becomes more reliable. The LDS (Figure 6.9b) produces no warning here, since for greater prediction horizons the variance is already big enough that the calculated collision probability never reaches the threshold τ_{col} (i.e. not enough probability mass is contained in the driving corridor).

It can be summarized, that compared to a LDS, the C-SLDS enables a higher prediction horizon. On the same time the C-SLDS keeps the prediction reliable enough, avoiding an increase of false positive warnings in the presented crossing pedestrian scenario.

6.3.3 Online Demonstration

A demonstration design was created to make a realistic experience of the system possible. The demonstration layout can be found in Figure 6.10. In the live demonstration, a C-SLDS system, using a prediction horizon of 16 frames ahead, is compared with a common LDS system, as often found in nowadays production vehicles. The prediction for the LDS is set to 1 frame ahead, as otherwise the uncertainty would become too high.

Two scenarios were shown. In the first scenario the pedestrian is not aware of the vehicle and continues with the same velocity until he crosses the driving corridor in front of the vehicle (Figure 6.10a). The demo vehicle reaches a maximum speed of $9\,\mathrm{m/s}$ before a full braking maneuver with a deceleration of around $-7\,\mathrm{m/s^2}$ is performed by the driver in front of the crossing pedestrian. In the second scenario the pedestrian is aware of the vehicle and stops before entering the driving corridor. The demo vehicle again reaches a maximum speed of $9\,\mathrm{m/s}$ but this time the driver continues with the same velocity passing the pedestrian in a narrow distance (Figure 6.10b).

By tuning T_{ttc} for the specific demonstration localities, the staged scenarios provide still a realistic impression and at the same time safety guidelines can be taken into account and the actual warning can be sent out earlier for demonstration purposes.

System experience is created by two different acoustical warn tones of the two systems, if a collision is predicted. A high frequency warn tone is used

for the LDS and a low frequency warn tone is used for the C-SLDS collision warning.

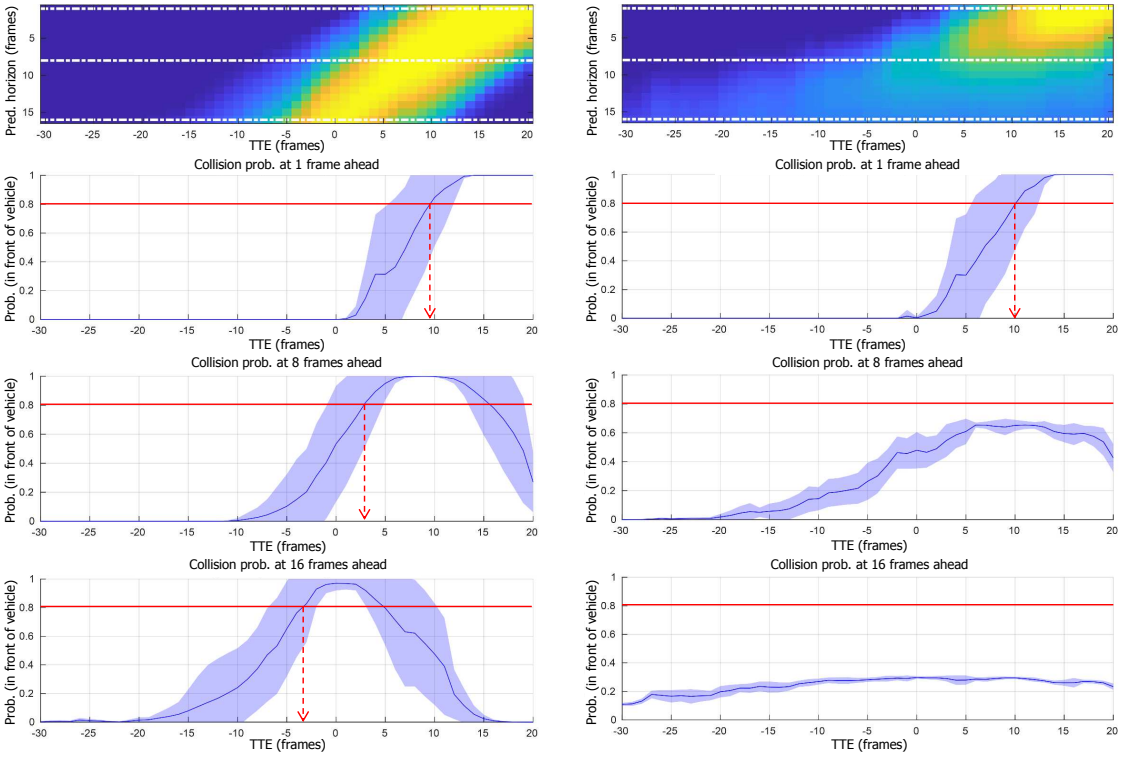
In nearly all performed crossing scenario demonstrations, the warning originated from the C-SLDS was sent out around 0.5 - 1.0 s earlier than the warning of the parallel running LDS system. When the pedestrian was aware and stopped before entering the driving corridor, close to zero false positive warnings occurred[1].

Figure 6.11 shows a visualization of the integrated system developed for demonstrating the pedestrian scenario. The system visualizes observables, the estimated latent context cues and the final prediction. Additionally, Figure 6.12a shows an important critical *anomalous* scenario, i.e. the pedestrian from Figure 6.11 continues crossing the street although he is aware of the vehicle. This pedestrian behavior is called an anomaly, as these behaviors were not used for DBN model parameter training. Because of the missing prediction in the LDS system, the warning of the C-SLDS system can still be sent out earlier than the one of the LDS system. The static environment cue provides strong support here. As soon as the pedestrian position passes the learned stopping area (from the static environment cue), the prediction is immediately corrected towards the walking model. Figure 6.12b plots the inferred latent variables to better visualize pedestrian dynamics and estimated context cues over the complete time of the pedestrian scenario showed in Figure 6.11a, 6.11b and 6.12b. The warning of the C-SLDS (blue line) can be still sent out about 5 time steps (i.e. 0.3 s) earlier than the LDS warning (yellow line).

Although future work will involve a better modeling for such anomalies (e.g. by adding other VRU behavior context cues), critical scenarios could be presented with the implemented system, showing a superior warning using the C-SLDS system compared to the LDS system.

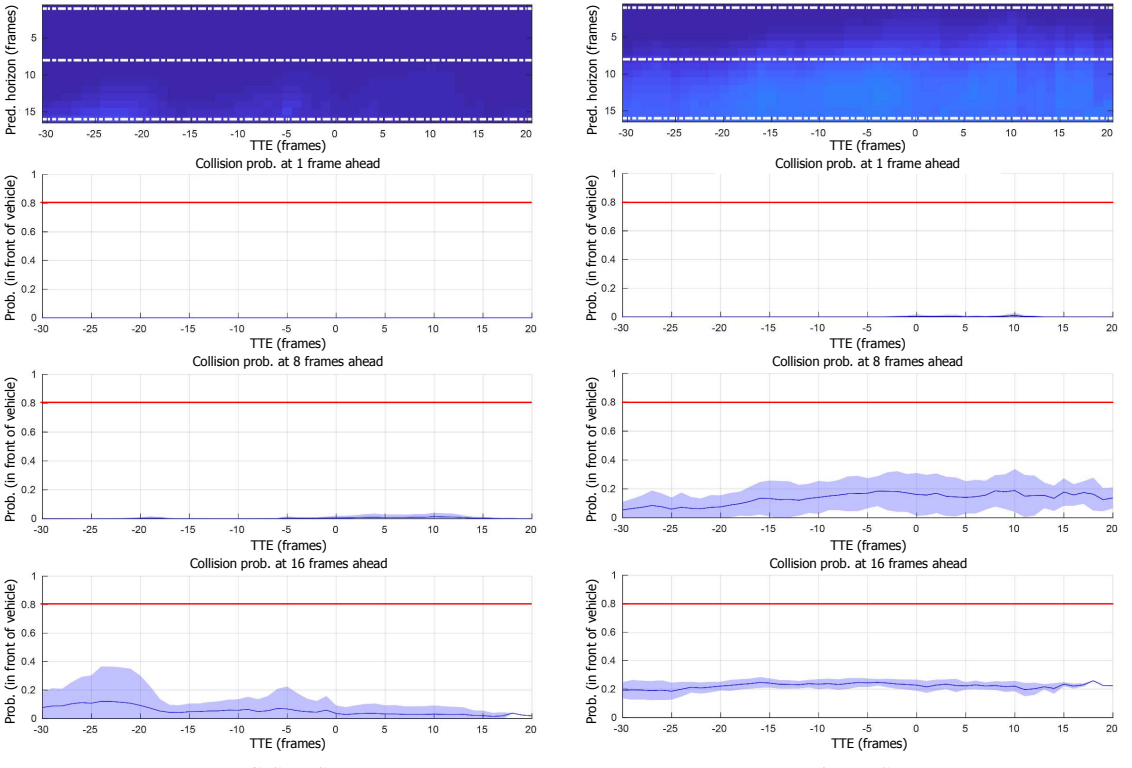
Figure 6.13 shows some impressions from the live demonstrations, performed during the final stage of the public founded project UR:BAN [33]. All demonstrations have been done with professional trained staff and accurate security guidelines. In a variation, the demonstration was also performed with a non-moving vehicle. In this case, a vehicle velocity of $9\,\mathrm{m/s}$ was simulated. By this, static and indoor exhibitions were also possible.

[1] False positive warnings occurred due to errors in the head orientation estimation because of bad lighting conditions or because the pedestrian has already entered the driving corridor.



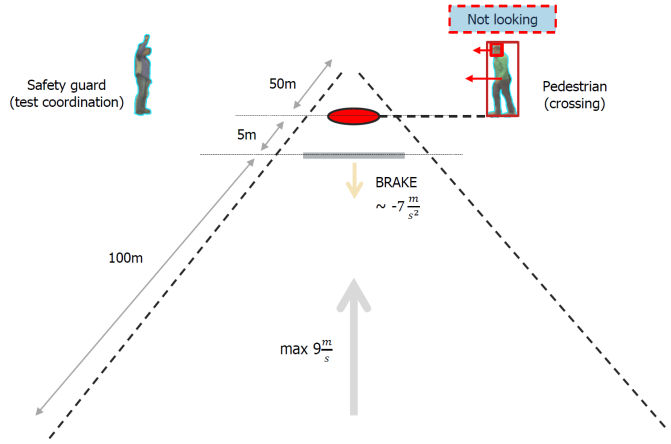
(a) C-SLDS (b) LDS

Figure 6.8: Critical situation introduced by an inattentive and crossing pedestrian. A collision is detected above 0.8 (red line). Estimated collision probability is shown for various prediction horizons (first row) and for predictions of 1 / 8 / 16 time steps (row 2 / 3 / 4) ahead. With the C-SLDS (a), the collision can be detected around 13 time steps earlier than with a LDS (b).

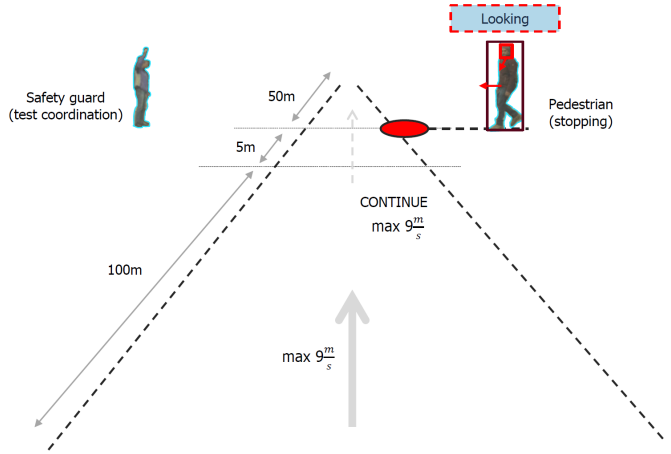


(a) C-SLDS (b) LDS

Figure 6.9: Critical situation with an attentive and stopping pedestrian, in contrast to the situation showed in Figure 6.8. The C-SLDS (a) makes use of context cues to better estimate the dynamic change (walking to stopping) of the pedestrian. The LDS (b) predicts "correct" because with the high uncertainty the collision threshold is never reached.

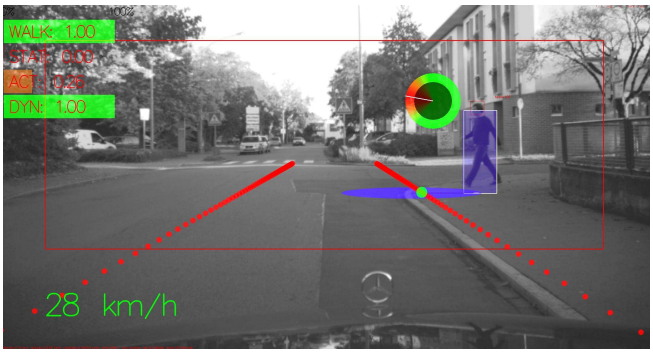


(a) Pedestrian not aware and crossing

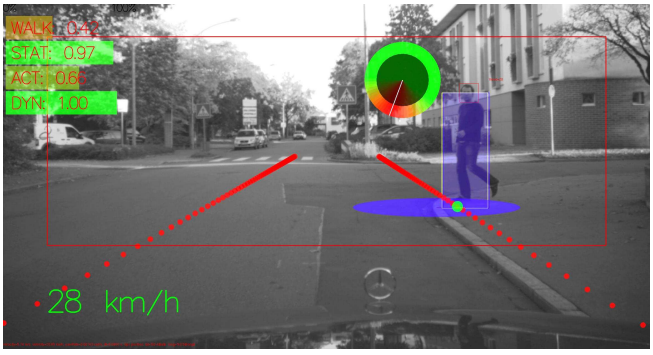


(b) Pedestrian aware and stopping

Figure 6.10: The two demonstration scenarios for explaining the context-based path prediction method from [133]. (a) Scenario where the pedestrian is not aware of the approaching vehicle and continues with the same velocity until he crosses the driving corridor in front of the vehicle. Maximum vehicle speed is $9\,\mathrm{m/s}$ and braking is performed manually with a deceleration around $-7\,\mathrm{m/s^2}$. (b) Scenario where the pedestrian is aware of the vehicle and stops before entering the driving corridor. The driver does not brake in this scenario.

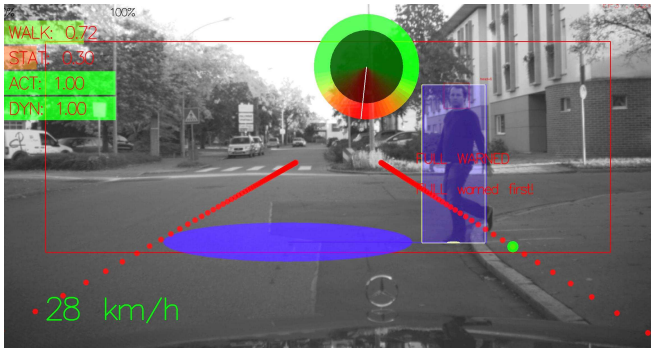


(a) Pedestrian is *not* attentive

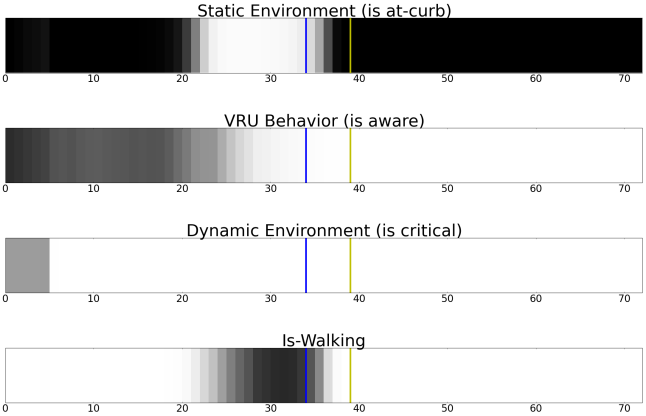


(b) Pedestrian becomes attentive

Figure 6.11: System visualization for the integrated context-based path prediction system, showing a situation where a pedestrian is walking towards the driving corridor. Images show estimated single-frame head orientation distributions (disc, red color means higher probability) and the collapsed predicted distribution of the C-SLDS model (blue ellipses show one standard deviation), which is used to compute the collision probability. Furthermore, bars (upper left) showing walking model probability (WALK) and probabilities of estimated context cues, i.e. static (STAT) and dynamic environment (DYN) and VRU context (ACTED). In (a) the pedestrian is *not* aware of the vehicle, in (b) the pedestrian becomes aware of the approaching vehicle. Using the context cues, the model predicts a higher chance that the pedestrian will stop before entering the vehicle corridor (see blue ellipse).

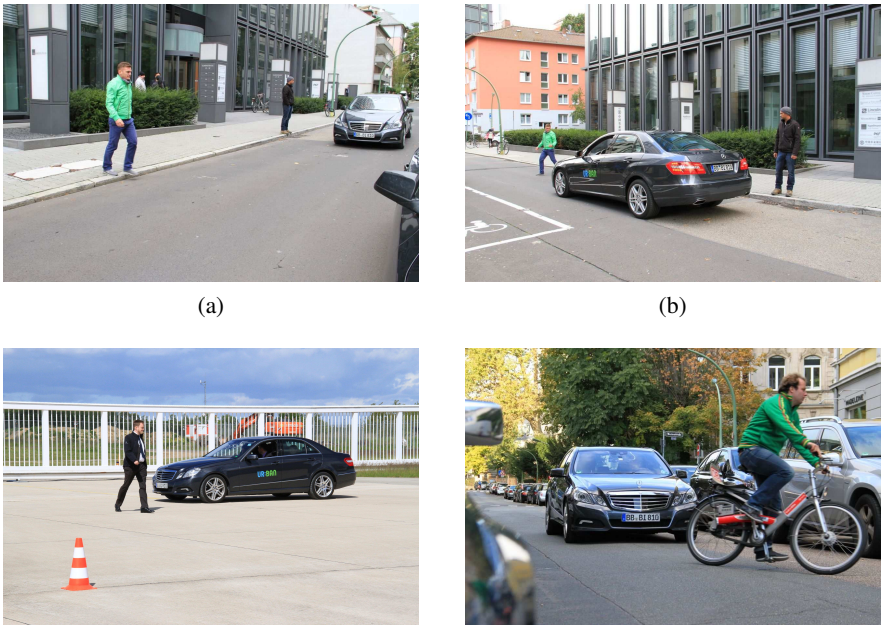


(a) Pedestrian continues



(b) Estimated latent variables plotted over time (frames)

Figure 6.12: (a) System behavior in case the pedestrian is attentive but continues crossing the street in front of the vehicle. The learned restrictions by the static environment context help here and support the estimation of the correct motion state. As soon as the pedestrian enters the driving corridor the prediction is corrected and the walking model becomes more probable again; (b) Latent variable plot to better visualize the pedestrian dynamics and estimated context cues over time (frames) for the tracked pedestrian (i.e. at frame 0 the track has been confirmed). When the pedestrian approaches the driving corridor, the walking model temporary gets a low probability (black). This can be corrected with support of the static environment cue although awareness of the pedestrian has a high probability (white). The warning of the C-SLDS (blue line) can be still sent out about 5 time steps (frames, i.e. 0.3 s) earlier than the LDS warning (yellow line).



(a) (b)

(c) (d)

Figure 6.13: Impressions from various demonstrations, presenting the context-based path prediction: (a)-(c) showing staged pedestrian scenarios in the inner city of Frankfurt. Trained staff has performed the critical pedestrian scenarios. (d) Staging for the future work on cyclist path prediction. Either the cyclist is bending in or moving straight in front of the vehicle.

6.4 Discussion

A realtime vehicle implementation of the method described in [133] making use of VRU and environment context cues to improve VRU path prediction was presented. The method has been integrated in multiple test vehicles, working within a stereo-vision system. The influence on the intervention concept has been studied. Through various demonstrations valuable feedback was gained and future work directions could be identified, which will be discussed now.

By focusing on the pedestrian crossing scenario, it is enough to model the lateral component of the pedestrian's position. For an application in an self-

driving vehicle, models need to be adapted to include the longitudinal dynamics as well. That poses some challenges on identifying, modeling and training the different scenarios that can occur here in common city areas (e.g. various straight scenarios, turn scenarios), also for different types of VRUs. The work described in [131] already shows an extension of the method to the cyclist case, where the dynamic models were modified to support bending-in dynamics of the cyclist.

The DBN formulation allows an integration of further context cues to improve path prediction. Of interest are VRU context cues which provide an indication about the current motion state (i.e. moving slow or fast, not moving) or cues providing a quick assessment on possible motion state changes (e.g. start walking, stopping, turning). A robust VRU skeleton estimation as shown in Figure 6.14 could provide a robust base representation, allowing an easy extraction of various VRU context observables e.g. by using simple discriminative classifiers to model further VRU behavior context cues. Orientation of body parts, for example, is already encoded and anatomical constraints between head and body are also covered by the skeleton estimate. The body angle can be used to extract immediate information about the motion state of a VRU. The faster a pedestrian moves, the stronger the body pitch is towards the moving direction. A cyclist performing a turn maneuver has a higher roll angle. A foot which is not standing on the ground provides a good indication that the VRU is in motion. This applies for both, pedestrians and cyclists. A standing pedestrian has both feet on the ground, while each foot is periodically lifted when the pedestrian is walking. A cyclist with one foot on the ground is most probably not in motion. Arm signals for the estimation of cyclist intentions can be also extracted. Recent deep CNN architectures [34] make a robust estimation of skeleton representations possible, as visualized in Figure 6.14. Preliminary experiments on such methods suggest that a transfer to the automotive domain is also possible and that the above mentioned context observables can be extracted.

Static environment context cues like the layout of signaled or unsignaled pedestrian crossings would allow a more accurate modeling of areas where VRU dynamics might change. A pedestrian approaching a green pedestrian traffic light will continue, although he might be aware of the approaching vehicles. A careful modeling of the influence on other traffic participants is needed for an accurate path prediction. For simplicity, only the ego-vehicle itself was taken into account in this work. Social force models [108] can help to model the complex interactions between VRUs and the influence on the VRU behavior.

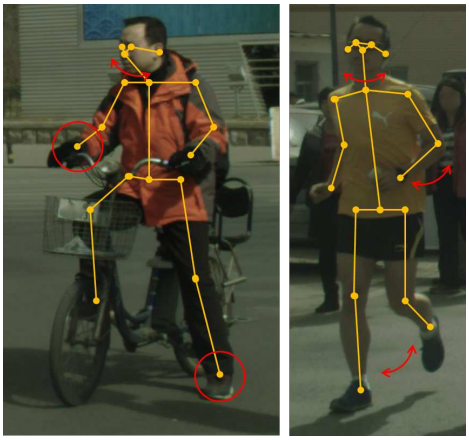


Figure 6.14: Skeleton representation as a robust representation for the extraction of VRU behavior cues. Based on the feet and leg positions, for example, it is instantly visible that the cyclist stopped, but the pedestrian is in motion. These cues can further support the path prediction in the short term horizon.

Also cultural and local differences need to be considered. In Asian countries for example, the interaction between traffic participants is often much more emphasized, compensating the lack of appropriate infrastructure.

Until now, the discussion has been focused on a stereo-vision system. In an intelligent vehicle, path prediction commonly takes place after sensor fusion. Thus, other sensor characteristics (e.g. from radar and lidar measurements) can be taken into account. Besides a more accurate detection and position accuracy by sensor fusion, radar micro-doppler effects [234] allow the instantaneous extraction of other useful VRU context cues, e.g. the leg movement of a pedestrian or the pedaling of a cyclist.

As was also found previously in last Chapters, gathering the correct data and specifying a correct data labeling process seem to be one of the major challenges here to further improve the discussed methods. Tools need to be adequately developed here to support the data management and labeling process. This will be discussed in the following Chapter 7.

7

LABELING AND DATA MANAGEMENT

Machine learning algorithms cannot operate without data. Therefore, this chapter sets the focus on the research topic of developing appropriate tools for data labeling and management. The proposed concepts serve as templates for a data center facilities managing the development, integration and improvement of machine learning algorithms as discussed in Chapter 3-6.

7.1 Introduction

As we have seen in the previous chapters, developing better models to improve the understanding of persons is essential for the success of increasing automation. Future systems will even be able to predict the behavior of objects based on various extracted cues. Sun *et al* [215] showed that the performance of deep learning methods increases logarithmically with the volume of data. First, this emphasizes the need for bigger datasets. Second, since the labeling complexity is steadily increasing with algorithms making use of additional and / or more detailed visual cues (e.g. scene segmentation [48, 150], orientation estimation [79] or VRU behavior estimation [131, 133, 197] algorithms), there is a strong need

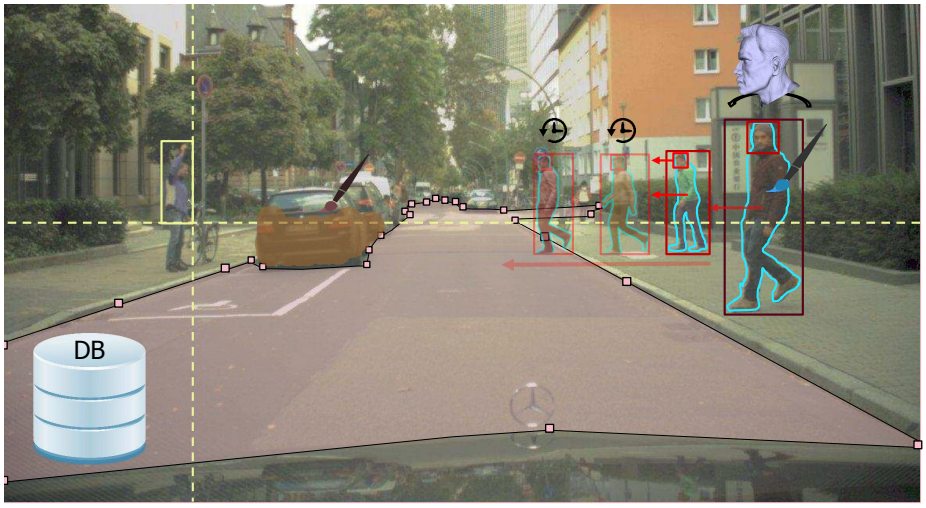


Figure 7.1: Efficient image labeling with various annotation types and levels of detail is necessary to provide enough and accurate data for self-driving vehicles. The labeling process, often done by non-technical staff, must be easy understandable and user interfaces must speak for themselves. It is only one part of a complex data workflow involving efficient data management and recording processes.

for an efficient labeling and data (sensor data, label data) organization process. Figure 7.1 shows that the complexity arises already within the image labeling when different labeling mechanisms need to be applied.

Motivated by the lack of an appropriate toolchain, especially tailored for the development and continuous improvement of machine learning algorithms, a complete data workflow including label and sensor data management is proposed. Concrete tool implementations cover all image-based label types applied in this work. Furthermore, label data is managed together with the sensor raw data in a developed data management system. The tools presented in this chapter were, and still are, continuously developed during the collection of new data and the creation of new datasets to meet new label and data requirements. In summary the contributions of this chapter can be summarized as:

- A complete label toolchain has been specified, and implemented together with an external company. Compared to other label tools [43, 158, 183,

200] the proposed toolchain combines all major image labeling concepts needed for segmentation, classification, detection and path prediction tasks within an integrated workflow for the development of machine learning applications.

- An integrated database backend manages the sensor and label data. Compared to other available label databases [138, 200], complex queries can be performed on sensor and label data jointly, allowing a powerful and unified data access. Furthermore, label task creation and task controlling facilitate an easy and large-scale distribution of labeling work.

- More than one million objects (including all datasets created during this work) have been already labeled with the proposed toolchain and feedback from labelers has been used to further improve the tools.

7.2 Label Data Workflow

With autonomous vehicles there comes a big amount of data produced by various different sensors. A two megapixel stereo camera running with 20 Hz produces already around 160 megabyte data per second. A fully equipped vehicle (with multiple different sensors) produces data amounts of 5 - 10 gigabyte data per second. It is clear that not all of this data can be stored continuously for all vehicles over the complete lifetime. The selection of the *right* data and data lifetime management are important research topics and require a careful data analysis process. Nevertheless, these topics are out of scope of this discussion. The reader is referred to [118, 119] for an overview on this challenging problem.

Figure 7.2 gives an overview of the proposed data workflow with a focus on the joint management of label and sensor data for the development of new machine learning algorithms. Having the *right* data at hand, the data is stored and gets imported in a database application where all sensor information is indexed with respect to the sensor stream (e.g. camera) and sensor timestamps. With this information available, further meta information (e.g. labels) can be linked to a specific recorded sensor sample (e.g. an image). The algorithm team uses the label client application to visualize sensor and label data, algorithm results and to validate new data concepts. The team plans new data recordings and specifies label tasks and datasets.

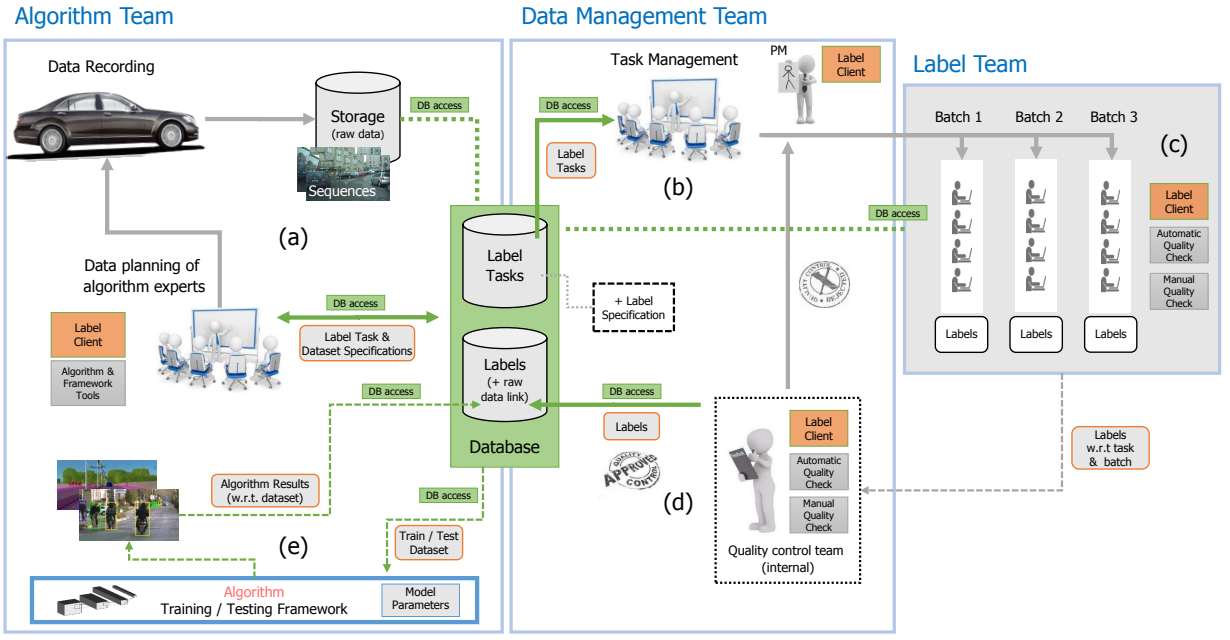


Figure 7.2: Proposed label data workflow for the intelligent vehicles domain. (a) New datasets are planned from the algorithm experts. Recorded sensor data for the datasets is stored and indexed in a database (green shaded) and is kept together with the label data for an efficient data access. (b) The data management team is responsible for coordination and quality assurance of the labeling process and via the Project Manager (PM) the (c) labeling team is supervised and taught how to use the label client (orange shaded) to fulfill the labeling task. (d) If the label data passes quality control, the label data is synced back to the database. (e) Based on the new available data, algorithm models are retrained and tested on the specified datasets. Algorithm results are pushed in the database. The algorithm team evaluates the results and plans the next steps (e.g. recording, label tasks).

A planned label task defines a set of samples from a recorded sensor stream along with a label specification describing what should be labeled and how it should be labeled. The label specification also contains quality guidelines (e.g. the maximum error in pixel for a labeled bounding box).

The data management team is responsible for coordination and quality assurance of the labeling process and via a Project Manager (PM) the labeling team is taught and supervised based on a given label task using the label specification. The label team needs to learn a proper handling of the label client application to fulfill the label tasks. Different label mechanisms (e.g. bounding box, polygon-based, orientation labeling) are needed here to complete different label tasks. To increase efficiency, assisted labeling with (semi-)automatic methods can be applied. An automatic quality control helps to reduce obvious errors of the labelers. Where larger data amounts are involved, the labeling team is often not on-premise. Nowadays, different companies offer *private label farms* which make it possible to outsource bigger label tasks. Nevertheless, it is desired to keep a direct access to a database application managing sensor and label data. This makes it possible to pull label tasks (i.e. data samples and label specification) and submit finished work. Thus, labeling is performed on the database instead of local copies of the data, increasing the data integrity.

The performed labeling quality gets checked within a quality control loop. If the label data does meet the label specification, the label data is approved in the database and made available to the algorithm team. Otherwise the PM is responsible to pass change requests to the labeling team.

Based on new recorded and labeled data, machine learning datasets are specified by the algorithm team using the database application. Algorithm models are retrained and tested on the specified datasets and the algorithm results are pushed back in the database. The algorithm team visualizes and checks the new results and next steps (new data recording, label tasks) can be derived based on the new insights. Additional to label tasks defined on new recorded data, a label task can also specify a review of algorithm results. By this, the label team can be instructed to identify errors within the provided and processed data samples. Only erroneous data samples can then be labeled according to the provided label specification. This step closes the data workflow and introduces an additional validation loop.

For the concrete development of a label data software suite implementing the proposed workflow, the following seven main aspects should be taken into account:

1. **Label mechanisms** Different label mechanisms are needed to support different label requirements (e.g. bounding box, segmentation, orientation or image / sequence labeling) to finally develop different machine learning applications. The required label quality can vary dependent on the requirements of the application.

2. **Assisted labeling** (Semi-) Automatic labeling mechanisms can assist the labeling process. For certain use cases, assisted labeling saves time for the labeler and can help to identify label errors.

3. **Data views** Beside the main label view, customizable data views for different user groups (e.g. algorithm engineer, labeler) and data levels are of advantage. All information needed for an efficient labeling and algorithm development process need to be visible for the users. Data views should cover different data aspects (e.g. object-based, image-based, sequence-based overviews).

4. **Data storage / integrity** Sensor and label data need to be stored and indexed in a unified and failsafe way. The data integrity (e.g. accuracy and consistency in data representation) should be kept among different sensor and label data representations. Additional version control mechanisms allow the management of multiple label and sensor data versions.

5. **Extensibility** Tools should be optimized among different data types and label mechanisms. From a software architecture perspective, common basic functionality and visualization should be shared across different features. This makes an extension of the tools easier.

6. **Label task management** The high amount of data that needs to be labeled requires a more sophisticated planning and management of label tasks to allow a scalable and organized distribution of tasks.

7. **Usability** The application of the tools should be easily understandable to the user. User interfaces (UIs) should reduce the amount of possible error sources and should allow a effective and efficient usage[115]. Optionally, gamification [56] can be applied in the labeling process to keep the labeler motivated and focused. An online estimation of Key Performance Indicators (KPIs) for the label process and the label quality would allow an even better assessment here.

Label tool	Label mechanisms	Assisted labeling	Additional data views	Data storage / sanity	Extensibility	Label task management	Usability
LABRADOR v1.0 [51]	Bounding box, track ID	Linear interpolation	Object details (text and visu), sequence view (text)	Image input, CSV-like label format (fixed structure)	View and logic not clearly separated, plugin mechanism	-	UI customizable, keyboard shortcuts
Cityscapes Label Tool [48]	Segmentation (polygon)	-	-	Image input, XML label format	View and logic not clearly separated, no common data type and visualization core available	-	Approved and optimized for segmentation labeling, UI not customizable, keyboard shortcuts
Philosys [183]	Bounding box, segmentation (polygon and brush), key point, track ID	Image stack for image pre-processing, flood fill, superpixel-based segmentation, interpolation and extrapolation (+vehicle odometry), detection and tracking	Object view (text), sequence view (visu)	ADTF [224] input, XML label format	N/A, diversity in label types and visualization, plugin API	-	ADTF-experience needed, UI customizable, keyboard shortcuts
LabelMe [200]	Bounding box, segmentation (polygon)	-	-	Image input, XML label format, database backend for labels	View and logic not clearly separated, no common data type and visualization core available	-	Browser-based (slow image loading), UI not customizable
C.Label [43]	Bounding box, segmentation (polygon), key point, track ID	Image stack for image pre-processing, detection and tracking	Object details (text), sequence view (visu)	Image and video input, XML label format, hierarchical label definitions	N/A, diversity in label types and visualization	KPI calculation for label process / quality	UI customizable, keyboard shortcuts
Labelbox [138]	Bounding box, segmentation (polygon), key point	-	-	Image input, various label formats, database backend for labels	Plugin interface, common visualization core, generic interfaces for label tasks	Task management and label quality control	Focused on task distribution, browser-based, access control
VoTT [158]	Bounding box, key point	Camshift tracking [27], loose coupling to FRCN	-	Video input, various label formats	View and logic not clearly separated, fixed GUI for configuration	-	Browser-based, UI not customizable
LABRADOR v3.1 (this work)	Bounding box, segmentation (polygon and brush), orientation, key point, skeleton, track ID	Image stack for image pre-processing, interpolation and extrapolation (+ vehicle odometry and stereo support), correlation tracker, snakes [121], GrabCut [198]	Object details (text and visu), sequence view (text), mouse-over, adaptive context menus	Image input, JSON label format, database connection for sensor and label data (LOSD), hierarchical label and sensor data definition	View and logic separated, plugin mechanism, shared visualization and data type concepts	Task management and tool configuration (LOSD)	Approved client-server application, access control, UI customizable, keyboard shortcuts

Table 7.1: Comparison of different available label tools with a focus on image-based labeling.

7.3 Existing Tools

Various labeling tools exist, providing a diverse set of labeling mechanisms and features. Some of them also provide customized functionality targeting the intelligent vehicles domain (e.g. making use of vehicle odometry for assisted labeling). Based on the previously inferred dimensions from Section 7.2, Table 7.1 compares a selection of already available tools. The focus is set on tools supporting an image-based labeling process.

Regarding functionality for the intelligent vehicles domain, C.Label [43] and Phylosis [183] show a great and diverse feature set, providing major label mechanisms (bounding box, segmentation and key points). Cityscapes [48] and LabelMe [200] provide efficient user interfaces focusing on segmentation labeling with polygons. C.Label and Phylosis provide an additional set of semi-automatic tools for assisted labeling including detection and tracking of objects. Also VoTT [158] present some first, not fully implemented ideas to couple machine learning algorithms (e.g. Fast-RCNN) tightly in to the labeling process. Furthermore, a CamShift [27] tracker is used to extrapolate labeled bounding boxes. Phylosis can apply additional vehicle odometry information to get more robustness in extrapolation / interpolation of bounding boxes.

C.Label, Phylosis and LABRADOR v1.0 [51] provide multiple textual and graphical label data views beside the main label window (where the actual image labeling takes place). With different data views it becomes easy to keep an overview of what was labeled in an image (e.g. how many objects, image tags) or for an object (e.g. body parts, orientation, object tags). Furthermore, a sequence-based overview summarizes the object information over multiple images, e.g. for each object instance using the track ID. Phylosis and C.Label provide additional visual views in form of time sequence bars, where tags and objects can be visualized.

While Phylosis is tightly coupled to the ADTF [224] framework, C.Label, LabelMe, VoTT [158], Cityscapes, LabelBox [138] can store labels in standard formats like XML or JSON. For an increased data amount, data integrity is difficult to guarantee when using single files, since formats, labels and / or sensor data storage location might change over time.

LabelMe and LabelBox show additional database backends for label data only. Labelbox presents a sophisticated label task management system, allowing a scalable collaboration in the labeling process. None of the existing tools integrates a database application for a joint management and a unified repre-

sentation of sensor and label data. Version control mechanisms are also not implemented.

Extensibility for closed-source tools (C.Label, Phylosis) is hard to estimate. It is supposed that tools showing various kind of mechanisms apply a common algorithm base and a shared visualization. Furthermore, C.Label and Phylosis provide plugin mechanisms to extend the functionality. Other tools, like Cityscapes, LabelMe, VoTT do not provide plugin functionality. Moreover, the software architecture of LabelMe, VoTT or Labrardor v1.0 do not show a clear separation of visualization and logic. The used interface elements (e.g. window layout, additional buttons and functions) can be customized (by configuration files) for Labrardor v1.0, C.Label and Phylosis. Other tools provide only a fixed view.

C.Label and LabelBox provide first ideas of how to assess KPIs for the labeling process, e.g. by measuring label speed and label accuracy over time. These KPIs could be used in the future to improve label mechanisms and allow also a possible gamification process to keep a labeler motivated and label accuracy high.

In summary, none of the presented tools provide all needed label mechanisms in combination with a well-integrated data workflow as proposed in Section 7.2. Therefore, during the work on the TDC dataset from Chapter 3, effort has been spent to improve tools for a better label and data workflow, allowing a continuous adaption to new label requirements and a steady improvement of label data and algorithms.

7.4 The LABRADOR Labeling Suite

New versions of the annotation tool LABRADOR were specified. The development of the tools was done together with an external company. LABRADOR v1.0 was originally developed by the Daimler AG and used internally for image-based labeling processes, mainly for bounding box tasks (cf. Table 7.1). In the remainder of the chapter the term LABRADOR refers to the new versions (up to v3.1) of the label tool developed during this work.

LABRADOR has been completely redesigned with respect to the seven aspects presented in Section 7.2. All label functionality and other features are added via a plugin-based mechanism allowing fast integration of new features. The tool provides a unified data format and currently supports detection and segmentation as well as orientation, skeleton and track-based label mechanisms

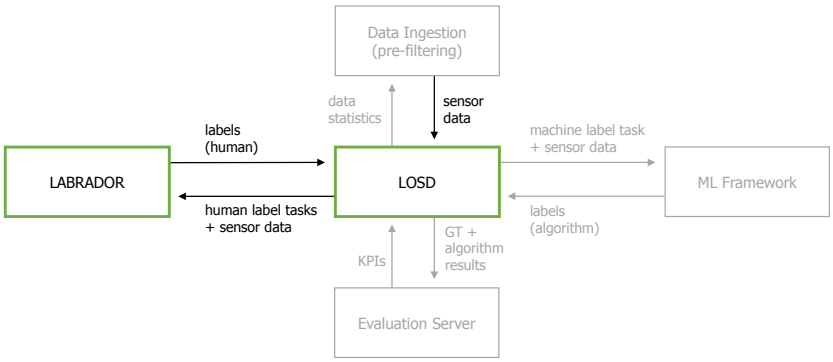


Figure 7.3: Developed software components and the interaction between them. Components not developed yet and left for future work are shown in grey.

for different sensor types (i.e. camera, stereo-camera, lidar and radar). LABRADOR was also provided to partners in the public project UR:BAN [33] to label different kinds of objects and features in camera images.

The LOSD (Label Object and Sequence Database) provides a unified data storage and access and therefore improves data integrity. Furthermore, a label task management system is integrated via LOSD. More details on these features are given in Section 7.4.3. Over an API, LOSD is tightly coupled to the LABRADOR client application. Figure 7.3 shows a more detailed view on the interaction between LABRADOR and LOSD. Over an *Ingestion Station* prefiltered sensor data (i.e. the *right* data) from the car fleet is added into the LOSD database. The LOSD database keeps indexes over different sensor streams and samples. It links the label data to this sensor data. With LABRADOR, the label team can pull new user-specific label tasks including sensor data and label specification. Labeling in LABRADOR can be performed using different label mechanisms. After completion of the label tasks, the label data is pushed back to the LOSD database. Using the *Machine Learning (ML) Framework*, algorithms are retrained on the new created datasets. Ground truth labels are compared to algorithm labels using an *Evaluation Server* providing the calculation of different KPIs (e.g. Precision-Recall, ROC curves for classification / detection problems) and their visualization. This whole process repeats until the KPIs are found satisfactory.

Figure 7.4 shows the LABRADOR UI with an example image from the

TDC dataset (Figure 7.4a). Different data views summarize the available labeled information of one or multiple images and allow an efficient labeling process (Figure 7.4f-7.4i). Object labels have a specified depth level, which allows to hide and show different objects (Figure 7.4c). In complex scenes this gives a better overview. If disparity information from a stereo-camera is available, the initial depth level of a labeled object is set automatically. Images are pre-processed according to the defined image stack (Figure 7.4d). Different plugins (e.g. debayering, contrast- or color-correction, smoothing, white-balancing) can be executed in a customizable order and allow an adaption to different image formats, recorded from different cameras.

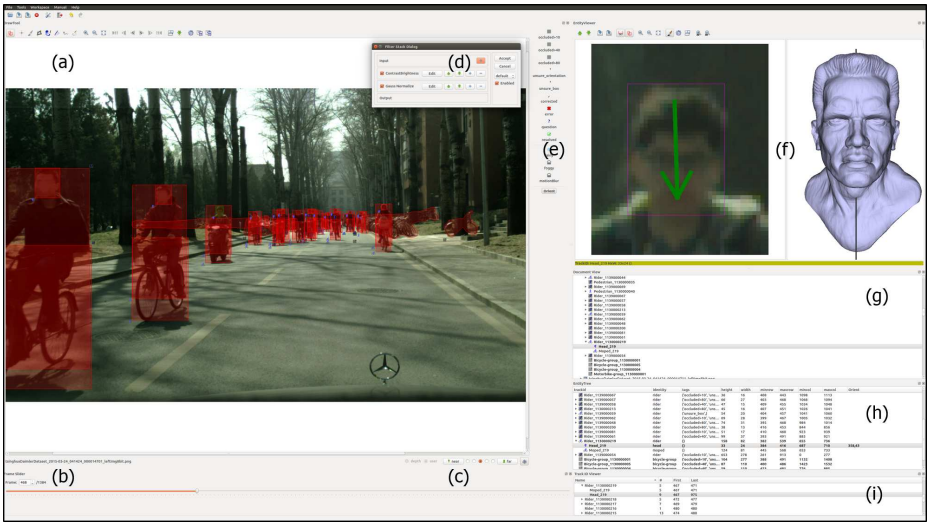


Figure 7.4: LABRADOR main window. Different data views allow an efficient labeling process. (a) Main label window. (b) Playback and Frame selection. (c) Hide / Show labels from different depth layers. (d) Image stack for pre-processing the image. (e) Quick-Tag labels for object-, object-part-, image- or sequence-based tagging. (f) *Entity Viewer* for a detailed object view. (g) *Document View* for an overview of data sample to label. (h) *Entity Tree* showing label details for each labeled entity in the selected image. (i) *Track ID Viewer* showing track details for all labeled object entities in the selected image.

All label types, classes, attributes and tags can be defined in a single JSON configuration file. Hierarchical relations between classes, attributes and tags are

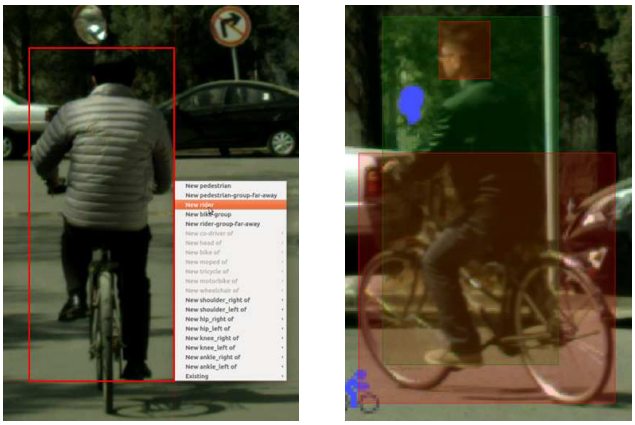
possible, e.g. *a bike belongs to a rider, a head has an orientation*. Attributes can be assigned to objects or object parts, e.g. a head can get an additional orientation attribute. Tags can be defined image- / object- or part-based, e.g. an object or a object part can be occluded. Most frequent used tags / attributes are placed next to the main labeling window (Figure 7.4e). For all important tool functionality (e.g. selection of label type, object class, tags) a keyboard short-cut can be defined. Complex scenes can be labeled without clicking a single button, making the labeling process more efficient. Table 7.1 (bottom) summarizes the LABRADOR v3.1 features in comparison to the other tools.

7.4.1 Image-based Labeling Mechanisms

This section presents the various image-based labeling mechanisms supported by LABRADOR. Furthermore, cross-references to the data which has been labeled during this work are drawn. LABRADOR v3.1 combines all label mechanisms in one tool. This combined implementation has the following advantages:

- **Extensibility** Different label mechanisms share the same core functionality. This makes an implementation of extensions easier and already existing functional components can be reused.
- **Data integrity** The same data format can be shared among different label types and a unified storage in a database (e.g. LOSD) becomes possible.
- **Usability** Base functionality of the tool stays the same and the user interface (e.g. point selection, object deletion) can be optimized among all label mechanisms. This reduces training time for labelers.
- **Combination** Often, labeling mechanisms are jointly used, e.g. a vehicle object is labeled with a bounding box, and a pedestrian is labeled using a pixel-wise segmentation.
- **Visualization** Visualization concepts can be unified.

The configuration file specifies not only class hierarchies, it also defines the user interface and the available context menus. Labeling starts with drawing a line, polygon, rectangle or area. From the context menu, the labeler can then select the root class type (e.g. rider in Figure 7.5a). Based on the root entity, new child entities (e.g. head and bike in Figure 7.5b) can be added based on optional



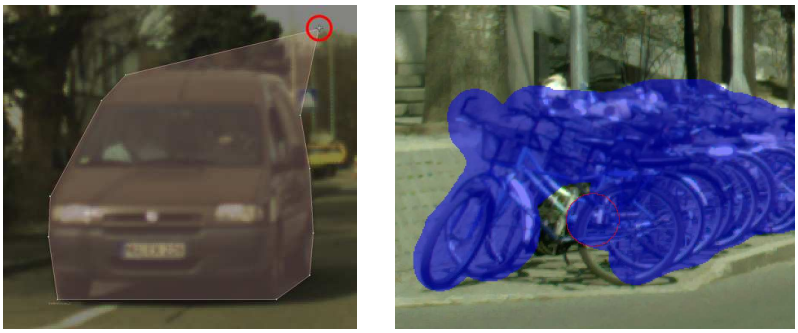
(a) Bounding box (b) Part-based representation

Figure 7.5: (a) Bounding box labeling. The class type is assigned afterwards via the customized context menu. (b) Hierarchical label definitions can be defined and parts (e.g. head, bike) are assigned to a root objects (e.g. rider).

restrictions (e.g. cardinality) defined in the configuration file (e.g. only one head per pedestrian).

Two basic types (Figure 7.6) exist for segmentation labeling: Polygon-based and brush-based labeling. For the first variant, points are sequentially added to the contour of an object. After the polygon is closed, the context menu opens to select the class type. Multiple points can be selected, translated or deleted. New points can be added by clicking on a connection between two points. Polygon-based labeling was used in [81] to segment the pedestrian and in [133] to label the street curb. Brush-based labeling allows a dense pixel-wise labeling with different brush sizes. The brush size can be varied quickly by keyboard shortcuts for an adaption to different image detail levels. The dense pixel-wise representation is converted to a polygon-based representation for data storage. For linear structures the point count is automatically reduced to gain a compressed and efficient representation of the polygon. Thus, both methods produce a variable amount of points depending on the object complexity and the used label precision. Brush-based labeling is currently applied in the TDC-v2 dataset (see Chapter 3) to label ignore and group regions. The Cityscapes dataset [48] has been labeled completely with polygon-based labeling using the Cityscapes la-

beling tool. Trained expert labelers reach a high label quality by using this approach. Especially straight structures (e.g. road, buildings) can be labeled more efficient than with a brush. Nevertheless, brush-based labeling shows good efficiency for strong non-linear structures e.g. round traffic signs or for traffic light bulbs. Where label precision is not the driving factor (e.g. ignore or group regions as showed in Figure 7.6b), brush-based labeling is usually also faster.

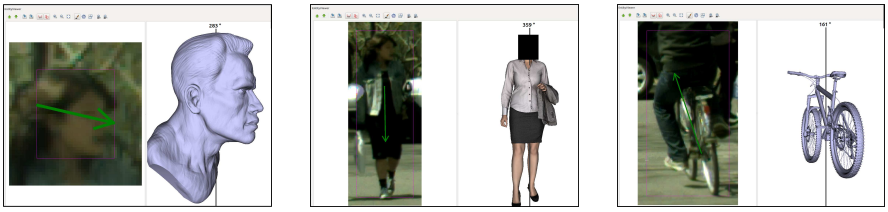


(a) Polygon-based (b) Brush-based

Figure 7.6: Polygon labeling. The polygon definition support holes and provides union and intersection operations. (a) Common polygon-based labeling. Points can be added, deleted and translated instantly. (b) Brush-based labeling. A (optimized) polygon is produced afterward.

The orientation of objects or object parts is labeled with 3D models (Figure 7.7). The use of 3D models makes an accurate labeling of continuous orientations possible. The 3D models can be rotated by mouse or by keyboard shortcuts. By comparing the image cut-out of the body part with the 3D model, a quick identification of the orientation becomes possible. Preliminary experiments during the work on [79] have shown that the labeler finds it easier to choose a head orientation by a 3D model than just using text or images showing a discrete set of orientation examples. Separate 3D models can be set in the configuration for each object or object part. Part-based bounding box labeling with orientations was applied for [28, 79, 80, 131, 133, 144, 197].

Skeleton labeling (i.e. the labeling of a set of specific anatomical limb points) was introduced to support future work on skeleton estimation. Furthermore, it has been applied the first time in [131] to label the cyclist arm pose.



(a) Head orientation (b) Body orientation (c) Bike orientation

Figure 7.7: Orientation labeling with 3D models. For each object or object part a separate model can be loaded, e.g. (a) head, (b) body and (c) bike 3D model.

A skeleton point configuration is defined in 3D model coordinates and is also visualized with the 3D model (Figure 7.8). Additionally, thumbnail images for each point are shown in the main label view for an easier identification. Skeleton points can be set manually or an initial configuration can be placed in the image, based on the preselected body orientation. In the latter case, skeleton points are projected on the correct location in the image based on the person's size and orientation. A skeleton point can be self-occluded (i.e. by other body parts) or occluded by other objects. A self-occlusion flag is set automatically when using an initial configuration by body orientation (i.e. the 3D skeleton specification defines which points are most probably occluded by other body parts).

The track ID is a unique identifier for a person over time. It is needed to evaluate tracking algorithms and to be able to predict the movement of an object into the future, based on previous observed information. The labeling of the track ID is supported with small thumbnails which are generated based on all objects appearing in the previously labeled image and are not assigned yet in the current image. Figure 7.9 shows the track ID labeling based on a cyclist instance. Track IDs have been used in the TDC dataset [144] as well as in the datasets for orientation estimation [28, 79, 80] and context-based path prediction [131, 133, 197].

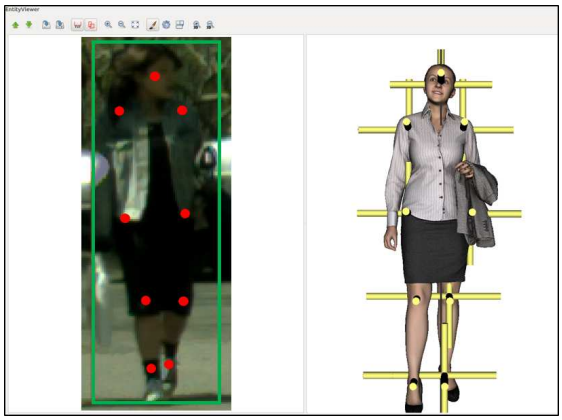


Figure 7.8: Skeleton labeling. Skeletons can be initially placed on the object dependent on the size and orientation of the person. Occlusion of points is estimated based on the 3D model.

Figure 7.9: Track ID labeling. Track IDs can be assigned by selecting thumbnails from previously labeled objects.

7.4.2 Assisted Labeling

Semi-automatic label methods have been introduced to accelerate the labeling process in certain cases (Figure 7.10). In addition to the manual polygon-based labeling, active contour models (snakes) [121] can be applied to perform a local optimization of a drawn contour (Figure 7.10a). Between a fixed point and the current mouse pointer the contour is fitted to the pixels with the overall strongest edge magnitude (within a pre-defined search area). Snakes can be used in between the manual polygon-based labeling. The fast switch (via keyboard shortcut) speeds up the labeling when there are decent contrast differences between the object and the background and the application of snakes becomes possible. Snakes has been used for segmentation-based labeling in [81] and the static environment cue (i.e. curb) labeling in [131, 133].

Based on a given bounding box, the GrabCut [198] algorithm can be applied. The algorithm solves a segmentation problem in the same spirit as the segmentation method described in Chapter 4 based on color features, but with user interaction. The labeler can specify fixed regions belonging to foreground (yellow) and regions belonging to background (red) to improve the initial contour estimate (Figure 7.10b). After a labeler interaction, the segmentation gets immediately updated based on the provided information. If the color appearance of background and foreground is reasonable different, this method works well. It accelerates the label process in such cases. As with brush-based labeling the resulting polygons are optimized afterwards.

For sequence labeling, simple linear prediction and interpolation procedures between two labels can be applied (Figure 7.11). With vehicle odometry and camera information available, vehicle egomotion compensation is applied to improve the label projection into the subsequent images. If additional disparity information is available, depth information can be utilized to get a more accurate positioning of the predicted / interpolated objects. Otherwise a simple planar world assumption [92] is used. For static objects or approximately linear moving objects, decent prediction / interpolation results can be gained by this method, speeding up the label process. In case of non-linear movements, imprecise odometry measurements or inaccurate depth estimations, it is necessary to check the automatic results to gain high quality label results. Interpolation has been used for the TDC dataset [144] to interpolate between not manually labeled frames.

(a) Snake-based polygon (b) GrabCut-based segmentation

Figure 7.10: Assisted polygon labeling. (a) Local contour optimization using snakes [121]. (b) Local optimization within a bounding box using GrabCut [198] showing initial estimated contour (left) and refined contour after six user interactions (right).

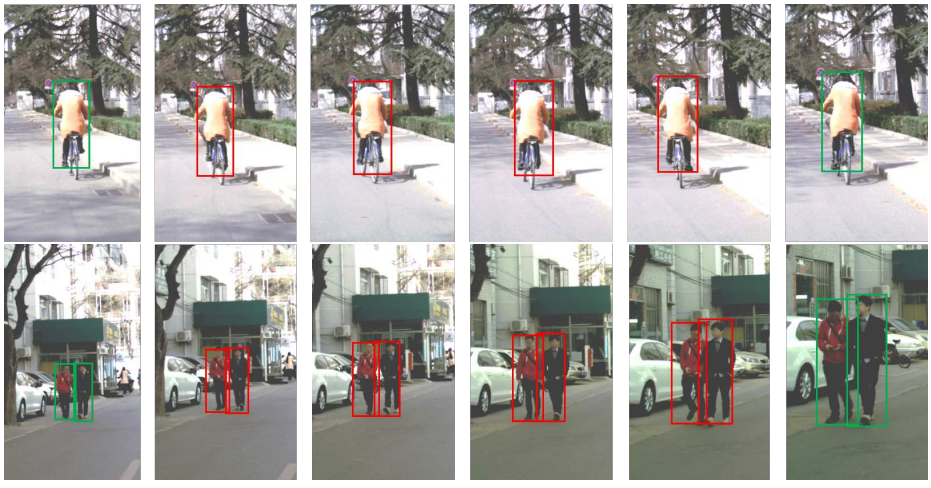


Figure 7.11: Interpolation with vehicle odometry support within 50 frames (i.e. 2.5 s with 20 Hz, showing every 10th frame). Ego vehicle and VRUs are moving. First and last frame have been labeled manually (green boxes).

7.4.3 Sensor and Label Data Management

The Label Object and Sequence Database (LOSD) manages sensor data from various sensor types (i.e. camera, lidar, radar). Metadata (e.g. labels as labeled with LABRADOR) can be added based on the indexed sensor data. Data formats from different robotic and automotive frameworks (e.g. ROS [188], ADTF [224]) are supported. Based on the indexed data, the user is able to query and select sensor data, e.g. for defining a labeling task. Completed annotations are then pushed back in the database. Data queries can be performed on sensor and label data jointly. Thanks to an extensive API, LOSD can be connected to the label software LABRADOR. In the future, the API enables also a direct connection to machine learning frameworks to train and test new methods using the direct data access. Figure 7.12 shows the LOSD dashboard providing an overview of sensor and label data indexed in the database. Data indexing of recorded sensor data can take a long time and is executed by the scheduler in the background. An overview of running, completed or failed scheduler jobs (e.g. sensor data uploading) is provided. Features of the current LOSD version (v2.1) are:

- Indexing of sensor data (e.g. camera, radar, lidar) from various framework formats (e.g. ROS, ADTF).
- Unified and unique access to all single sensor data samples (e.g. images) via a Uniform Resource Identifier (URI) address.
- Web-based frontend for label and sensor data management.
- C++ / Python API for connecting other applications to LOSD.
- User and access management to restrict data access. Access control allows role-based data views (i.e. administrator, label supervisor, labeler).
- Query mechanisms for sensor and label data.
- Export of sensor and label data.
- Management of label tasks.
- Definition of machine learning datasets (ML-datasets) based on sensor and label data.
- Grouping of data samples over multiple recorded sequences (MetaDataset).

- Version control of sensor and label data, MetaDataset, ML-datasets and label template configurations.
- Automatic data consistency and sanity checks by using hash codes [156] over data containers and labels.

To emphasize the role of LOSD for the labeling process, the next sections focus on the data query and task management aspect.

7.4.3.1 Data Queries and Meta-Datasets

Various kinds of sensor and label data are represented in a unified database format. This makes it possible to define a powerful query mechanism on the LOSD database. Each indexed sensor sample (e.g. image) is accessible by a unique URI. With this URI, label data can be linked to a sensor sample or stream.

The query wizard allows a combination of single simple queries (including regular expressions) to a more complex query with logical operations (Figure 7.13). Any available field from sensor and label data can be used. With this mechanism it is easily possible to define complex queries like: *"Sequence name contains 'tsinghua'* ***and*** *Image tags contain tag 'rainy'* ***and*** *Object class is 'Pedestrian'* ***and*** *(Pedestrian head orientation between '120'* ***and*** *'140 degree'* ***or*** *Pedestrian is 'heavy occluded')."*. The result can then be viewed on the different data levels (i.e. sequence, image, label object or object-part level). The query results can be extracted to a JSON file containing the label data information along with the URIs pointing to the location of the sensor data samples.

Based on data queries, Meta-Datasets can be defined. A Meta-Dataset contains different samples over one or multiple recorded sensor data sequences. This allows the specification of a data subset, optionally in combination with already labeled data. To allow multiple label data versions (e.g. through a later correction or augmentation) each label object has a version tag associated. When generating a Meta-Dataset different version tags can be taken into account and merged together.

Figure 7.12: LOSD dashboard showing an overview of available (a) indexed sensor data and (b) label data. (c) Running, completed or failed activities on the database are summarized.

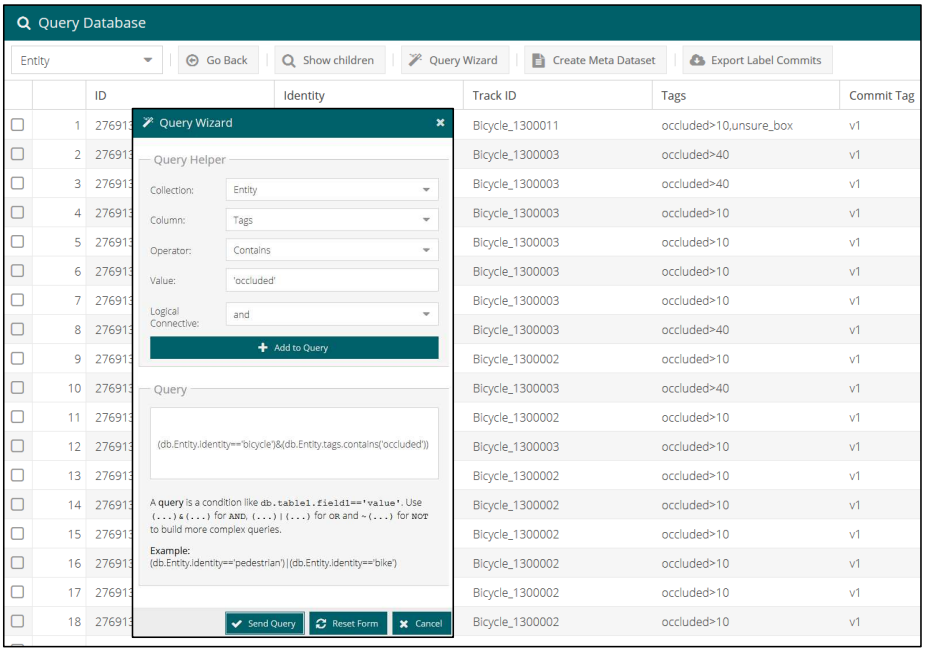


Figure 7.13: Query mechanism allowing the creation of complex data queries.

7.4.3.2 Label Specification and Label Tasks

The LOSD web application makes it possible to define labeling configurations for LABRADOR. The configuration includes:

- **Label hierarchy definition** Hierarchical object class definition including attributes, object and frame tags. Necessary resources (e.g. 3D model, skeleton definition, thumbnail images) are linked and also included.

- **Layout scheme** Layout specification defining the appearance of the LABRADOR user interface. It also defines which LABRADOR plugins are loaded and displayed.

- **Help and label specification documents** Additional resources explaining the usage of LABRADOR for a specific label task.

The label configuration editor (Figure 7.14) provides a graphical web user interface to compile this information. New class identities and hierarchical relationships between class identities, attributes, tags can be defined. For the visualization in LABRADOR, a label color (i.e. the color of the displayed rectangle or polygon of a specific class) and an icon can be added. Similar to sensor and label data, LABRADOR configurations are also under version control to allow a backtracking what configuration was used for a specific label task. When machine learning datasets are modified over time during the development process, this is an indispensable feature.

Label tasks (Figure 7.15) can be created on a previously defined Meta-Dataset. A label task is linked to a label configuration file and can be assigned to a responsible person (i.e. label supervisor or PM). Further task specifications (e.g. due date) can be added. The created label tasks can be splitted into smaller sub-tasks, which can be finally assigned to the executive labeler or a labeler group.

A logged in user can access the tasks assigned to him directly using a LABRADOR client application (Figure 7.16). Data samples, label configurations and existing labels are directly downloaded to the local running LABRADOR client. The label task overview shows the task status. Completed label results are saved directly to a temporary LOSD database location. If the supervisor accepts the performed labeling, results are pushed back into the main LOSD datasbase.

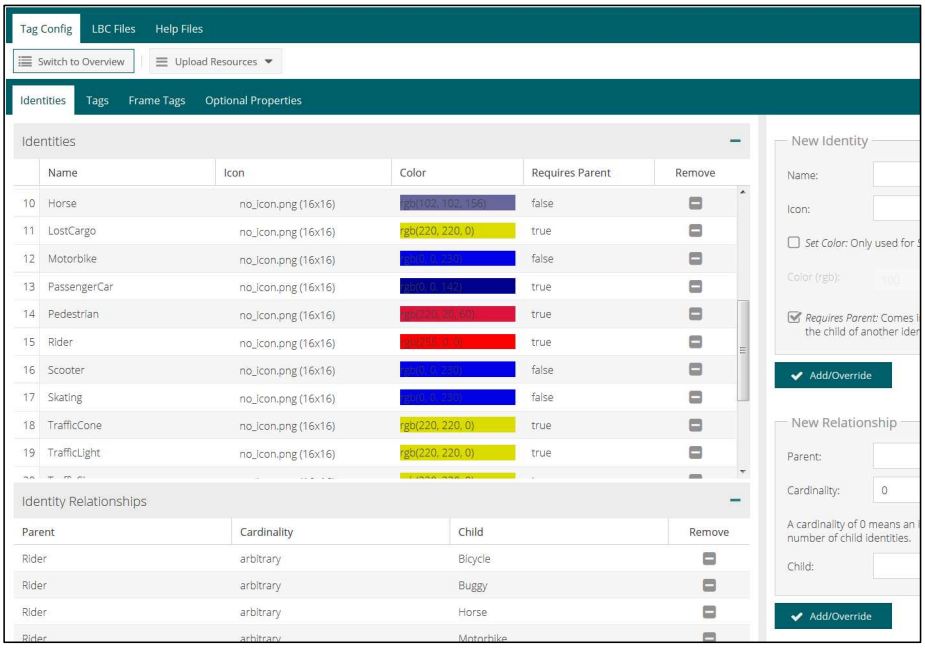


Figure 7.14: LOSD label configuration editor for the definition of label configurations. Label configurations specify the label hierarchy and define the user interface and help files for LABRADOR.

7.4.3.3 LOSD API

The LOSD functionality is accessible over a C++ and Python API. This API can be used to connect various other applications. Beside the query mechanisms and data exports, the API makes it possible to define version controlled ML-Datasets for training and testing machine learning algorithms. As for the label tasks, a ML-Dataset can be defined on a Meta-Dataset with a specified label version. Within the machine learning framework the sensor and label data specified via the ML-Dataset can then be downloaded and used to train and test new machine learning algorithms.

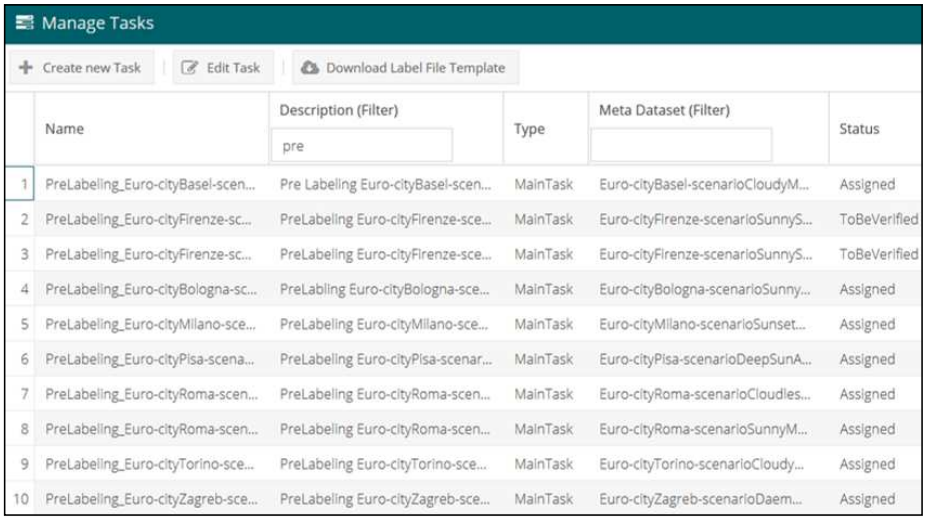


Figure 7.15: A label task overview is provided for different user roles. A label supervisor, for example, sees all tasks he assigned to the executive labelers.

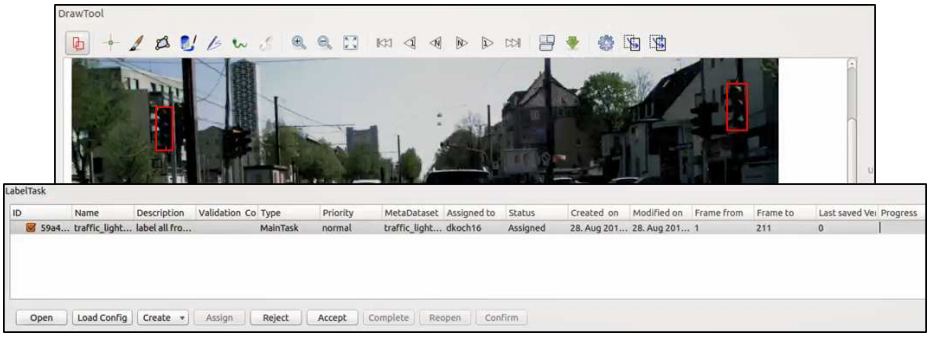


Figure 7.16: Assigned label tasks can be imported directly in LABRADOR. Associated sensor samples are downloaded and the label configuration defines the label hierarchy and the user interface.

7.5 Discussion

A complete label workflow has been proposed and an appropriate toolchain for data annotation and management has been developed. In particular, the annotation tool LABRADOR was presented, covering all important image-based annotation types used during the work on this thesis. Furthermore, LOSD was presented, a database application developed for the management of sensor and label data. Both tools can be used in a tightly coupled manner featuring an efficient labeling process by making use of unified and consistent data management and label task control mechanisms. This finally enables a faster development of data-hungry machine learning algorithms.

Qualitative evaluations have shown the benefit of combining different label mechanism (e.g. point-based polygons with automatic contour fitting using snakes [121]). Detailed studies on labeling time and label quality have not been performed yet. Future work should assess user acceptance of the developed tools with appropriate methods [206]. While the average labeling time for different label mechanisms can be easily measured, the label quality needs a deeper assessment. For some label types, e.g. object orientation, it is difficult to define a reference solution solely based on the image data. Additional reference sensors (e.g. lidar) can be used in this case. For other attributes, e.g. head localization and orientation, the camera sensor might be the only available sensor. In these cases, the label precision can be measured at least indirectly by a redundant execution of the label task to get a distribution over the label values.

The current LOSD implementation is based on a local database backend using MongoDB [42], a document-based database system. Using a single local database node does not scale with the amount of data and data access expected in a development process for self-driving vehicles. Nevertheless, the current developed concept can be transferred to a cloud-based distributed system, e.g. within a Hadoop [229] environment. This would allow a fast and efficient data access using concepts like mapr streams [64] and MapReduce [54].

Assisted labeling can be further extended by automatic detection (cf. Chapter 3), segmentation (cf. Chapter 4) and tracking (cf. Chapter 6). In the labeling process, quality is often more important than realtime processing. Therefore ensembles of different algorithm models [100] can be combined to reduce the bias introduced by a single algorithm. Operating points and parameters of the algorithms can be adapted to the specific use case. With a high recall, for example, the human labeler task could be reduced to accepting or rejecting propos-

als from an automatic detection algorithm. The PedCut segmentation method (Chapter 4) can be modified to allow the selection of the MSSM representation by the user. With this the user task reduces to provide an initial pose of the object and an automatic object segmentation can be performed. Further user interactions, similar to GrabCut, could be also applied to improve the used data cues.

The aforementioned example of using a high recall could also be used in combination with a crowdsourced labeling approach [117]. Here the labeling team (Figure 7.2) would be replaced by a labeler crowd, a large amount of anonymous, low-cost and (predominantly) untrained labelers (e.g. members of an online community). Therefore such crowd tasks need to be designed and described simple and easy understandable to get correct label solutions. For more complicated tasks (e.g. orientation, skeleton labeling) the same labeling task can be performed multiple times by the crowd (by different persons) to get a more accurate answer or an uncertainty over the label. The main challenge here is to find the best trade-off between reliability and redundancy [120]. Crowd-based labeling is a hot topic and bears a big potential for gaining efficient and cost-effective label results for training machine learning algorithms [153]. Amazon Mechanical Turk [32], for example, was successfully used for various tasks of the Microsoft COCO dataset [147]. Unfortunately, privacy and data protection pose some restrictions on the application for autonomous driving and VRUs.

8

CONCLUSIONS AND FUTURE WORK

This thesis addressed the detection, segmentation and orientation estimation of persons in visual data. The developed methods were used to realize components of an advanced Vulnerable Road User (VRU) detection system (Figure 1.3b). Starting with a robust VRU detection and classification, the VRU object was subjected to a deeper analysis by means of an accurate pixel-wise segmentation followed by an orientation estimation of body parts to gain a better representation of the VRU model. The extracted cues have been finally used to estimate the VRU behavior, allowing an improved path prediction. Beside a concrete system integration, the challenge of data management was emphasized as a constant companion for the development of machine learning systems. This chapter will now reflect on the thesis objectives outlined in Section 1.2.1, by drawing conclusions, and identifying directions of future work.

8.1 Conclusions

For the task of object detection and classification the method SP-FRCN was presented, utilizing the power of Fast-RCNN combined with efficient stixel-

based proposals at multiple scales and viewpoint aspects. Stixel-based proposals were evaluated for different configurations and it was shown that a significantly higher recall can be gained by using less proposals than commonly used 2D proposal method. Furthermore, SP-FRCN also outperformed ACF-based detector ensembles on the newly introduced Tsinghua Daimler Cyclist Benchmark dataset. The lack of appropriate datasets for cyclist detection benchmarking led to the creation of this richly annotated automotive stereo vision dataset, being the first published dataset of a reasonable size focusing on cyclist detection benchmarking.

Making use of the robust VRU detection, the object analysis started with a pixel-wise segmentation based on an iterative approach inspired by the EM algorithm, combining shape priors with bottom-up data cues within a Conditional Random Field formulation. The benefit of different cue combinations was shown on a newly introduced stereo-vision pedestrian segmentation dataset. Furthermore, the proposed method outperformed state-of-the-art methods on foreground accuracy. The inferred (part-) segmentation can be used to improve part localization performance, allowing a more detailed look on important object parts, like head and body of the VRU.

To infer details about object parts, a joint estimation of head and body orientation was performed within a probabilistic framework based on a Dynamic Bayesian Network model. The framework involved a principled way to deal with faulty part detections, continuous orientation estimation, coupling of the body- and head-localization and orientation, and temporal integration. Experiments showed that the proposed joint tracking of head and body orientations decreases the mean absolute head / body orientation error by $11°$ / $15°$compared to single frame estimation and further by $4°$ / $4°$compared to independent tracking. In absolute terms, this comes down to mean absolute head / body orientation error of about $21°$ / $19°$which remains fairly constant up to a distance of $25\,\mathrm{m}$.

The extracted VRU details were used to model context cues for a context-based SLDS formulation, which was integrated in a real-time vehicle system. In particular, situational awareness was assessed by means of the estimated head orientation distribution, leading to a more accurate pedestrian path prediction. A practical evaluation showed the ability of the system to improve driver warning and vehicle control strategies. A vehicle intervention on a potential critical situation with a crossing pedestrian can happen up to $1\,\mathrm{s}$ earlier without increasing the false alarm rate. The system benefit was presented during multiple demonstrations using an acoustical driver warning supported by a component-

wise visualization.

To cope with the steadily increasing amount of data and data complexity introduced, e.g. by methods described in this work, an efficient and integrated label data workflow becomes necessary. Various workflow requirements have been identified based on the experience of different system integrations. A detailed analysis showed that existing tools do not meet all requirements. Therefore, a complete label suite, including a label client (LABRADOR) and database application (LOSD), was proposed. The tools can be used in a tightly coupled manner to allow an efficient labeling process and a fast development of data-hungry machine learning algorithms. More than one million traffic objects have already been labeled with the tool and the feedback of executive label team has already led to an improvement of the different label mechanisms and their usability.

8.2 Future Work

While many important problems have been addressed in this thesis, multiple directions for improvements and future research were found.

The quality of the performed VRU analysis stays strongly dependent on the detection performance. Stixel-based proposals show a good efficiency but need to be improved for occlusion handling and tuned towards small objects. Future work will therefore include a careful tuning of stixel and proposal parameters (e.g. stixel width and segmentation costs). Furthermore, stixel clustering can be applied to reduce the proposal count and to better control the usage of specific aspect ratios. Also a combination with 2D proposal methods [112] for image regions with limited stereo support could be beneficial. Furthermore, some of the recently proposed improvements to RCNN architectures [240] can be also used within the proposed SP-FRCN method, bearing a big potential to further improve the detection results. The already planned extension of the TDC dataset (TDC-v2) will deliver a large amount additional labeled data. By making use of the additional labels and new label types (e.g. ignore and group regions), an improvement in detection performance for stronger occlusions and smaller object sizes is expected. Also multi-class training becomes possible to better differentiate cyclists from other VRU types.

Inferring pedestrian detection and body part orientation jointly with a shared feature representation increases efficiency. A first step is already taken in [28], where detection and orientation are jointly estimated within a multi-task FRCN

framework using stixel-based and lidar-based proposals. A joint extraction of the head orientation within the architecture presented in [28] stays challenging due to the limited support of CNN features on the small head sizes [227]. Here, a modification of the feature maps and the CNN architecture becomes necessary. Modeling orientation estimation as a regression problem, by e.g. using a Biternion model within a deep network [18, 28], seems also a more reasonable way to estimate continuous orientations than using a discrete set of detectors. Extracted environment cues can also serve as an additional prior to improve orientation estimation. Similar to [134], learned location-based head and body motion dynamics can be incorporated to guide and improve orientation estimates. Apart from that, better results can be expected by making use of a larger dataset, e.g. using the planned TDC-v2 including head and body orientation labels. This dataset allows also a transfer to other VRU types becomes possible, e.g. head and body orientation estimation of a cyclists. Furthermore, a common and fair benchmark for VRU orientation estimation will be established by TDC-v2.

Part-based segmentations can further improve the extraction of behavior related part locations, e.g. the head, arm, leg positions and movements. While a binary segmentation of the pedestrian was in focus of this work, part-based segmentation estimation is possible based on available point correspondences given by the SSMs. Probabilistic texture knowledge can provide additional support for the shape prior, e.g. by using Active Appearance Models (AAM) [46]. To better avoid local minima by a wrong initialization of the shape template, spatio-temporal MSSMs, as proposed in [94], can be used to model temporal transition between the SSMs, providing a more stable MSSM representation over time. Replacing the applied Decision Trees by a more powerful deep neural network segmentation [150] might also improve the instance segmentation results.

The proposed system integration focuses on specific cues designed and selected for a particular VRU scenario. This approach may not scale to scenarios with more varied and complex behavior. To further improve the context-based path prediction, a careful collection of a diverse set of traffic scenes, scene layouts, pedestrian interactions and appearances is necessary to study what other object context cues can help here. In the optimal case this leads also to a more generic object representation, allowing the extraction of various behavior observables, e.g. by means of deep skeleton models [34] for the VRU. To model complex object interactions and their influence on the final object behavior, social force models [108] could be applied. The incorporation of other static en-

vironment cues, e.g. the layout of a signaled or an unsignaled pedestrian crossings, would allow a more precise modeling of areas where object dynamics might change. For example, a pedestrian approaching a red traffic light signal will stop, although he is not aware of the approaching vehicle. Furthermore, other sensors and sensor combinations can be utilized, allowing an extraction of other useful object context cues. For instance, radar micro-doppler effects [234] allow an instantaneous extraction of pedestrian's leg movement or the turning wheels and pedaling of a cyclist.

For a better labeling process, detailed studies on user acceptance [206], labeling time and quality are needed to better validate the use of the proposed labeling workflow and the developed tools. The in-depth analysis can then lead to further improvements on usability and efficiency of the tools. Assisted labeling using powerful detector ensembles [100] can be used to give an initial and less biased pre-labeling. The main task of the labeler is then reduced to correct this automatic labelings. Of course, it is important that the correction itself introduces not more work than labeling without the assistance. Additional crowdsourcing loops [117] can help to scale up the labeling process. Main challenges here are the definition of the labeling task as simple as possible and finding the best trade-off between reliability and redundancy [120]. To support distributed access and greater data amounts, the current LOSD implementation can be ported to a cloud-based distributed system, e.g. within a Hadoop [229] environment. This would allow a fast and efficient data access using concepts like mapr streams [64] and MapReduce [54]. All these techniques will allow the development and optimization of upcoming, more complex machine learning algorithms also applied in self-driving vehicles.

At the moment there is a big competition about offering the first usable product or service with self-driving vehicles. First ones are already expected around 2020. In the year 2030 we will most probably find a lot of self-driving vehicles from different companies on our streets. A smooth interaction with traffic participants and especially with other humans will play a major role here. A self-driving vehicle which does not understand its environment will certainly not win the future competition. Much more complex VRU context cues, as already discussed in this thesis, need to be assessed for allowing a natural human like interaction of an autonomous vehicle with its environment, especially in urban areas. Also explicit gestures need to be assessed. A self-driving vehicle needs to fully understand certain situations, e.g. a police officer controlling the traffic with a defined gesture set. A first step for this would be to robustly

classify these special VRU types (e.g. police officers), but even this is challenging due to the different clothing styles in different regions and the very similar appearance to "normal" VRUs. Worldwide collection of diverse and representative data will therefore remain an important component and challenge for the development of self-driving vehicles.

This thesis has taken the whole pipeline of in-vehicle VRU analysis into consideration, ranging from data acquisition and labeling over detection, segmentation and orientation estimation to an integrated system improving VRU path prediction. This has led to advancements in the state-of-the-art in VRU analysis, and to an introduction of valuable novel benchmarks for this important research direction. New follow-up challenges and research directions have been identified and provide manifold ideas to further improve the proposed methods. The proposed methods not only find an application in the intelligent vehicle domain. Also in modern human machine interaction or mobile robotics it becomes more and more necessary to study the user in more detail to understand his interest and behavior. Finally, this makes an improved interaction with such systems possible. We can also observe an increased usage of action and behavior classification methods in the surveillance domain, with the aim to detect suspicious activities at public places or in buildings. Clearly, for all of these interesting tasks there are a lot of open issues in machine learning and computer vision to solve. Nevertheless, this thesis provides already some principled solutions to tackle some of these future problems.

9

BIBLIOGRAPHY

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, pp. 2274–2282, 2012.

[2] Aglaia GmbH, "ANNOStation labeling system," 2017. [Online]. Available: http://www.aglaia-gmbh.de/products/annostation-labeling-system (accessed: 2018-04-28).

[3] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 73–80.

[4] M. Andres, W. Menzel, H.-L. Bloecher, and J. Dickmann, "Detection of slow moving targets using automotive radar sensors," in *Proc. of the German Microwave Conference (GeMiC)*, 2012, pp. 1–4.

[5] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1014–1021.

[6] G. Antonini, S. V. Martinez, M. Bierlaire, and J. P. Thiran, "Behavioral priors for detection and tracking of pedestrians in video sequences," *International Journal of Computer Vision (IJCV)*, vol. 69, no. 2, pp. 159–180, 2006.

[7] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 328–335.

[8] S. O. Ba and J.-M. Odobez, "A Rao-Blackwellized mixed state particle filter for head pose tracking," in *Proc. of the ACM International Conf. on Multimodal Interface, Workshop on Multimodal Multiparty Meeting Processing*, 2005, pp. 9–16.

[9] S. O. Ba and J.-M. Odobez, "Probabilistic head pose tracking evaluation in single and multiple camera setups," in *Proc. of the Workshop on Classification of Events, Activities and Relationships (Multimodal Technologies for Perception of Humans)*, 2008, pp. 276–286.

[10] S. O. Ba and J.-M. Odobez, "Multiperson visual focus of attention from head pose and meeting contextual cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 33, no. 1, pp. 101–116, 2011.

[11] H. Badino, "A robust approach for ego-motion estimation using a mobile stereo platform," in *Complex Motion: First International Workshop (IWCM)*, 2007, pp. 198–208.

[12] T. Bandyopadhyay, K. Won, E. Frazzoli, D. Hsu, W. Lee, and D. Rus, "Intention-aware motion planning," in *Proc. of the Tenth Workshop on the Algorithmic Foundations of Robotics*, 2013, pp. 475–491.

[13] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Systems*, vol. 29, no. 6, pp. 82–100, 2009.

[14] B. Benfold and I. Reid, "Guiding visual surveillance by tracking human attention," in *Proc. of the British Machine Vision Conf. (BMVC)*, 2009, pp. 1–11.

[15] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3457–3464.

[16] B. Benfold and I. Reid, "Unsupervised learning of a scene-specific coarse gaze estimator," in *Proc. of the International Conf. on Computer Vision (ICCV)*, Nov. 2011, pp. 2344–2351.

[17] N. Bernini, M. Bertozzi, L. Castangia, M. Patander, and M. Sabbatelli, "Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey," in *Proc. of the IEEE Intelligent Transportation Systems Conf. (ITSC)*, 2014, pp. 873–878.

[18] L. Beyer, A. Hermans, and B. Leibe, "Biternion nets: Continuous head pose regression from discrete training labels," in *Proc. of the German Conference on Pattern Recognition (GCPR)*, 2015, pp. 157–168.

[19] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006, vol. 1.

[20] S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*. Artech House Norwood, 1999.

[21] Y. Bo and C. C. Fowlkes, "Shape-based pedestrian parsing," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 2265–2272.

[22] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via pLSA," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2006, pp. 517–530.

[23] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *Proc. of the International Conf. on Computer Vision (ICCV)*, 2007, pp. 1–8.

[24] Y. Boykov, O. Veksler, and R. Zabih, "Markov random fields with efficient approximations," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1998, pp. 648–655.

[25] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 23, no. 11, pp. 1222–1239, 2001.

[26] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *Proc. of the International Conf. on Computer Vision (ICCV)*, 2001, pp. 105–112.

[27] G. R. Bradski, "Real time face and object tracking as a component of a perceptual user interface," in *Proc. of the Winter Conf. on Applications of Computer Vision (WACV)*, 1998, pp. 214–219.

[28] M. Braun, Q. Rao, Y. Wang, and F. Flohr, "Pose-RCNN: Joint object detection and pose estimation using 3D object proposals," in *Proc. of the IEEE Intelligent Transportation Systems Conf. (ITSC)*, 2016, pp. 1546–1551.

[29] M. Bray, P. Kohli, and P. H. S. Torr, "Posecut: Simultaneous segmentation and 3D pose estimation of humans using dynamic graph-cuts," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2006, pp. 642–655.

[30] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[31] N. Brouwer, H. Kloeden, and C. Stiller, "Comparison and evaluation of pedestrian motion models for vehicle safety systems," in *Proc. of the IEEE Intelligent Transportation Systems Conf. (ITSC)*, 2016, pp. 2207–2212.

[32] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data?" *Perspectives on psychological science*, vol. 6, no. 1, pp. 3–5, 2011.

[33] Bundesministerium für Wirtschaft und Energie, "UR:BAN - Die Forschungsinitiative für den urbanen Verkehr. Ergebnisbuch." 2016.

[34] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, p. 7.

[35] J. Carreira, F. Li, and C. Sminchisescu, "Object recognition by sequential figure-ground ranking," *International Journal of Computer Vision (IJCV)*, vol. 98, no. 3, pp. 243–262, 2012.

[36] T. Chateau, V. Gay-Belille, F. Chausse, and J. T. Lapresté, "Real-time tracking with classifiers," in *Proc. of the International Conference on Dynamical Vision*, 2007, pp. 218–231.

[37] C. Chen, A. Heili, and J.-M. Odobez, "A joint estimation of head and body orientation cues in surveillance video," in *Proc. of the International Conf. on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 860–867.

[38] C. Chen and J. Odobez, "We are not contortionists: Coupled adaptive learning for head and body orientation estimation in surveillance video," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1544–1551.

[39] L.-C. Chen, G. Papandreou, I. Kokkinos, K. P. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proc. of the International Conf. on Learning Representations (ICLR)*, 2015.

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

[41] M. M. Cheng, Z. Zhang, W. Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3286–3293.

[42] K. Chodorow, *MongoDB: The definitive guide: Powerful and scalable data storage*. O'Reilly Media, Inc., 2013.

[43] CMORE Automotive GmbH, "C.Label - A smart and efficient way to label your data," 2017. [Online]. Available: http://www.cmore-automotive.com/produkte/software-tools/clabel/ (accessed: 2018-04-28).

[44] E. Coelingh, A. Eidehall, and M. Bengtsson, "Collision warning with full auto brake and pedestrian detection-a practical example of automatic emergency braking," in *Proc. of the IEEE Intelligent Transportation Systems Conf. (ITSC)*, 2010, pp. 155–160.

[45] M. Collins, R. E. Schapire, and Y. Singer, "Logistic regression, AdaBoost and Bregman distances," *Machine Learning*, vol. 48, no. 1, pp. 253–285, 2002.

[46] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 23, no. 6, pp. 681–685, 2001.

[47] T. F. Cootes, C. J. Taylor, D. H. Cooper, J. Graham, and Others, "Active shape models-their training and application," *Computer Vision and Image Understanding (CVIU)*, vol. 61, no. 1, pp. 38–59, 1995.

[48] 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 *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3213–3223.

[49] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani, "Probabilistic posture classification for human-behavior analysis," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 35, no. 1, pp. 42–54, 2005.

[50] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 379–387.

[51] Daimler AG, "LABRADOR v1.0 - label rapidly and orderly: Software documentation," 2013.

[52] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 886–893.

[53] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proc. of the International Conf. on Machine Learning (ICML)*, 2006, pp. 233–240.

[54] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[55] A. Dempster, N. M. Laird, and D. Rubin, "Maximum-Likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, pp. 1–38, 1977.

[56] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: defining gamification," in *Proc. of the International Academic MindTrek Conference. Envisioning future media environments*, 2011, pp. 9–15.

[57] J. Dickmann, N. Appenrodt, J. Klappstein, H.-L. Bloecher, M. Muntzinger, A. Sailer, M. Hahn, and C. Brenk, "Making bertha see even more: Radar contribution," *IEEE Access*, vol. 3, pp. 1233–1247, 2015.

[58] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast Feature Pyramids for Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 36, no. 8, pp. 1532–1545, Aug 2014.

[59] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *Proc. of the British Machine Vision Conf. (BMVC)*, 2009, pp. 91.1–91.11.

[60] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests. IEEE transactions on pattern analysis and machine intelligence," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 37, no. 8, pp. 1558–1570., 2015.

[61] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, no. 4, pp. 743–761, Apr. 2012.

[62] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Journal of Statisitcs and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[63] M. Dumitru-Guzu, "Joint probabilistic pedestrian head and body orientation estimation," Master Thesis, Delft University of Technology, 2014.

[64] T. Dunning and E. Friedman, *Streaming architecture: New designs using Apache Kafka and MapR streams*. O'Reilly Media, Inc., 2016.

[65] G. H. Dunteman, *Principal components analysis*. Sage Publications, 1989, vol. 69.

[66] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 31, no. 12, pp. 2179–2195, 2009.

[67] M. Enzweiler and D. M. Gavrila, "Integrated pedestrian classification and orientation estimation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 982–989.

[68] M. Enzweiler, M. Hummel, D. Pfeiffer, and U. Franke, "Efficient Stixel-based object recognition," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2012, pp. 1066–1071.

[69] S. M. A. Eslami and C. Williams, "A generative model for parts-based object segmentation," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 100–107.

[70] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *International Journal of Computer Vision (IJCV)*, vol. 111, no. 1, pp. 98–136, 2015.

[71] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *International Journal of Computer Vision (IJCV)*, vol. 88, no. 2, pp. 303–338, 2010.

[72] V. Evers, N. Menezes, L. Merino, D. M. Gavrila, F. Nabais, M. Pantic, P. Alvito, and D. Karreman, "The development and real-world deployment of FROG, the fun robotic outdoor guide," in *Proc. of the ACM/IEEE in Human-Robot Interaction*, 2014, p. 100.

[73] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 617–624.

[74] Z. Fang, D. Vázquez, and A. M. López, "On-board detection of pedestrian intentions," *Sensors (Basel, Switzerland)*, vol. 17, no. 10, p. 2193, 2017.

[75] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[76] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision (IJCV)*, vol. 61, no. 1, pp. 55–79, 2005.

[77] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[78] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on Computers*, vol. 100, no. 1, pp. 67–92, 1973.

[79] F. Flohr, M. Dumitru-Guzu, J. F. Kooij, and D. M. Gavrila, "A probabilistic framework for joint pedestrian head and body orientation estimation," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, vol. 16, no. 4, pp. 1872–1882, 2015.

[80] F. Flohr, M. Dumitru-Guzu, J. F. P. Kooij, and D. M. Gavrila, "Joint probabilistic pedestrian head and body orientation estimation," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2014, pp. 617–622.

[81] F. Flohr and D. M. Gavrila, "PedCut: An iterative framework for pedestrian segmentation combining shape models and multiple data cues," in *Proc. of the British Machine Vision Conf. (BMVC)*, 2013, pp. 66.1–66.11.

[82] U. Franke, D. Pfeiffer, C. Rabe, C. Knöppel, M. Enzweiler, F. Stein, and R. G. Herrtwich, "Making Bertha see," in *Proc. of the International Conf. on Computer Vision Workshops (ICCV Workshops)*, 2013, pp. 214–221.

[83] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *The annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[84] T. Gandhi and M. M. Trivedi, "Image based estimation of pedestrian orientation for improving path prediction," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2008, pp. 506–511.

[85] F. Garcia, P. Cerri, A. Broggi, J. M. Armingol, and A. De La Escalera, "Vehicle detection based on laser radar," in *Proc. of the International Conf. on Computer Aided Systems Theory*, 2009, pp. 391–397.

[86] D. M. Gavrila, "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding (CVIU)*, vol. 73, no. 1, pp. 82–98, 1999.

[87] D. M. Gavrila, "A Bayesian, exemplar-based approach to hierarchical shape matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29, no. 8, pp. 1408–1421, 2007.

[88] D. M. Gavrila and S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle," *International Journal of Computer Vision (IJCV)*, vol. 73, no. 1, pp. 41–59, 2007.

[89] S. K. Gehrig, R. Stalder, and N. Schneider, "A flexible high-resolution real-rime low-power stereo vision engine," in *Proc. of the International Conf. on Computer Vision System (ICVS)*, 2015, pp. 69–79.

[90] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[91] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI Vision Benchmark Suite," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2012, pp. 3354–3361.

[92] D. Gerónimo and A. M. López, *Vision-based pedestrian protection systems for intelligent vehicles*. Springer, 2014.

[93] D. Gerónimo, A. M. López, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 7, pp. 1239–1258, 2010.

[94] J. Giebel and D. M. Gavrila, "Multimodal shape tracking with point distribution models," in *Proc. of the DAGM Symposium on Pattern Recognition*, 2002, pp. 1–8.

[95] R. Girshick, "Fast R-CNN," in *Proc. of the International Conf. on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[96] R. Girshick, F. Iandola, T. Darrell, J. Malik, and U. C. Berkeley, "Deformable part models are convolutional neural networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 437–446.

[97] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[98] G. Gkioxari, R. Girshick, and J. Malik, "Contextual action recognition with R*CNN," in *Proc. of the International Conf. on Computer Vision (ICCV)*, 2015, pp. 1080–1088.

[99] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik, "R-CNNs for pose estimation and action detection," *arXiv preprint arXiv:1406.5212 [cs.CV]*, 2014.

[100] J. Guo and S. Gould, "Deep CNN ensemble with data augmentation for object detection," *arXiv preprint arXiv:1506.07224*, 2015.

[101] S. Gupta, R. B. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014, pp. 345–360.

[102] P. Hall, D. Marshall, and R. Martin, "Merging and splitting eigenspace models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 22, no. 9, pp. 1042–1049, 2000.

[103] P. M. Hall, A. D. Marshall, and R. R. Martin, "Incremental eigenanalysis for classification," in *Proc. of the British Machine Vision Conf. (BMVC)*, 1998, pp. 286–295.

[104] H. Hamaoka, T. Hagiwara, M. Tada, and K. Munehiro, "A study on the behavior of pedestrians when confirming approach of right/left-turning vehicle while crossing a crosswalk," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 106–110.

[105] Z. Hayder, X. He, and M. Salzmann, "Boundary-aware instance segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 587–595.

[106] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. of the International Conf. on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[107] J. Heer and M. Bostock, "Crowdsourcing graphical perception: using mechanical turk to assess visualization design," in *Proc. of the SIGCHI Conference on Human Factors in Computing Systems (ACM)*, 2010, pp. 203–212.

[108] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[109] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 30, no. 2, pp. 328–341, 2008.

[110] E. Hjelmås and B. K. Low, "Face detection: A survey," *Computer Vision and Image Understanding (CVIU)*, vol. 83, no. 3, pp. 236–274, 2001.

[111] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering surface layout from an image," *International Journal of Computer Vision (IJCV)*, vol. 75, no. 1, pp. 151–172, 2007.

[112] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?" *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 38, no. 4, pp. 814–830, 2016.

[113] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 3, pp. 334–352, 2004.

[114] D. Iakovidis, C. Smailis, T. Goudas, and I. Maglogiannis, "Ratsnake: A versatile image annotation tool with application to computer-aided diagnosis," *The Scientific World Journal*, vol. 2014, p. 12, 2014.

[115] International Organization for Standardization, "Ergonomics of human-system interaction – part 210: Human-centred design for interactive systems (iso 9241-210:2010)," *ISO/TC 159/SC 4 Ergonomics of human-system interaction*, vol. 1, pp. 1–32, 2010.

[116] N. Jafarinaimi, "Exploring the character of participation in social media: the case of google image labeler," in *Proc. of the iConference*, 2012, pp. 72–79.

[117] J. C. S. J. Junior, S. R. Musse, and C. R. Jung, "Crowd analysis using computer vision techniques," *IEEE Signal Processing Magazine*, vol. 27, no. 5, pp. 66–77, 2010.

[118] S. Kaisler, F. Armour, J. A. Espinosa, and W. Money, "Big data: Issues and challenges moving forward," in *Hawaii International Conference on System Sciences (HICSS)*, 2013, pp. 995–1004.

[119] K. Kambatla, G. Kollias, V. Kumar, and A. Grama, "Trends in big data analytics," *Journal of Parallel and Distributed Computing*, vol. 74, no. 7, pp. 2561–2573, 2014.

[120] D. R. Karger, S. Oh, and D. Shah, "Efficient crowdsourcing for multi-class labeling," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 1, pp. 81–92, 2013.

[121] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision (IJCV)*, vol. 1, no. 4, pp. 321–331, 1988.

[122] C. G. Keller, M. Enzweiler, and D. M. Gavrila, "A new benchmark for stereo-based pedestrian detection," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 691–696.

[123] C. G. Keller and D. M. Gavrila, "Will the pedestrian cross? A study on pedestrian path prediction," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, vol. 15, no. 2, pp. 494–506, Apr. 2014.

[124] C. G. Keller, T. Dang, H. Fritz, A. Joos, C. Rabe, and D. M. Gavrila, "Active pedestrian safety by automatic braking and evasive steering," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, vol. 12, no. 4, pp. 1292–1304, 2011.

[125] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert, "Activity forecasting," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2012, pp. 201–214.

[126] A. Kläser, "Image annotation tool with image masks," 2007. [Online]. Available: https://lear.inrialpes.fr/people/klaeser/software_image_annotation (accessed: 2018-04-28).

[127] H. Kloeden, N. Brouwer, S. Ries, and R. Rasshofer, "Potenzial der Kopfposenerkennung zur Absichtsvorhersage von Fußgängern im urbanen Verkehr," in *FAS Workshop Fahrerassistenzsysteme, Walting, Germany*, 2014.

[128] T. Ko, "A survey on behavior analysis in video surveillance for homeland security applications," in *Applied Imagery Pattern Recognition Workshop*, 2008, pp. 1–8.

[129] S. Köhler, B. Schreiner, S. Ronalter, K. Doll, U. Brunsmann, and K. Zindler, "Autonomous evasive maneuvers triggered by infrastructure-based detection of pedestrian intentions," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 519–526.

[130] I. Kokkinos and P. Maragos, "Synergy between object recognition and image segmentation using the expectation-maximization algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 31, no. 8, pp. 1486–1501, 2009.

[131] J. F. P. Kooij, F. Flohr, E. A. I. Pool, and D. M. Gavrila, "Context-based path prediction for targets with switching dynamics," *International Journal of Computer Vision (IJCV)*, 2018 [Accepted for publication, DOI: 10.1007/s11263-018-1104-4].

[132] J. F. P. Kooij, M. C. Liem, J. D. Krijnders, T. C. Andringa, and D. M. Gavrila, "Multi-modal human aggression detection," *Computer Vision and Image Understanding (CVIU)*, vol. 144, pp. 106–120, 2016.

[133] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila, "Context-based pedestrian path prediction," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014, pp. 618–633.

[134] J. F. P. Kooij, N. Schneider, and D. M. Gavrila, "Analysis of pedestrian dynamics from a vehicle perspective," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2014, pp. 1445–1450.

[135] U. H.-G. Kreßel, *Advances in Kernel Methods*. MIT Press, 1999, ch. Pairwise Classification and Support Vector Machines, pp. 255–268.

[136] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[137] M. P. Kumar, P. H. S. Torr, and A. Zisserman, "Objcut: Efficient segmentation using top-down and bottom-up cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 3, pp. 530–545, 2010.

[138] LabelBox, Inc, "LabelBox," 2018. [Online]. Available: https://github.com/Labelbox/Labelbox (accessed: 2018-04-28).

[139] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. of the International Conf. on Machine Learning (ICML)*, 2001, pp. 282–289.

[140] D. Larlus and F. Jurie, "Combining appearance models and markov random fields for category level object segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–7.

[141] K.-C. Lee and D. Kriegman, "Online learning of probabilistic appearance manifolds for video-based recognition and tracking," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 852–859.

[142] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool, "Dynamic 3D scene analysis from a moving vehicle," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.

[143] S. Levin and J. C. Wong, "Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian," The Guardian, 2018. [Online]. Available: https://www.theguardian.com (accessed: 2018-04-28).

[144] X. Li, F. Flohr, Y. Yang, H. Xiong, M. Braun, S. Pan, and D. M. Gavrila, "A new benchmark for vision-based cyclist detection," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 1028–1033.

[145] M. C. Liem and D. M. Gavrila, "Coupled person orientation estimation and appearance modeling using spherical harmonics," *Image and Vision Computing*, vol. 32, pp. 728–738, 2014.

[146] D. Lin, S. Fidler, and R. Urtasun, "Holistic scene understanding for 3D object detection with RGBD cameras," in *Proc. of the International Conf. on Computer Vision (ICCV)*, 2013, pp. 1417–1424.

[147] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014, pp. 740–755.

[148] T. Litman, *Autonomous vehicle implementation predictions*. Victoria Transport Policy Institute, 2017.

[149] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot Multibox Detector," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016, pp. 21–37.

[150] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[151] L. Lu, Z. Zhang, H. Shum, Z. Liu, and H. Chen, "Model and exemplar-based robust head pose tracking under occlusion and varying expression," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, 2001, pp. 1–8.

[152] P. C. Mahalanobis, "On the generalised distance in statistics," *Proc. of the National Institute of Science, India*, vol. 2, no. 1, p. 4955, 1936.

[153] L. Maier-Hein, S. Mersmann, D. Kondermann, S. Bodenstedt, A. Sanchez, C. Stock, H. G. Kenngott, M. Eisenmann, and S. Speidel, "Can masses of non-experts train highly accurate image classifiers?" in *Proc. of the International Conference on Medical Image Computing and Computer-assisted Intervention*, 2014, pp. 438–445.

[154] M. J. Marín-Jiménez, A. Zisserman, and V. Ferrari, ""Here's looking at you, kid". Detecting people looking at each other in videos," in *Proc. of the British Machine Vision Conf. (BMVC)*, 2011, pp. 1–12.

[155] M.-M. Meinecke, M. Obojski, D. M. Gavrila, E. Marc, R. Morris, M. Töns, and L. Lettelier, "Strategies in terms of vulnerable road user protection," in *EU Project SAVE-U, Deliverable D6*, 2003.

[156] A. J. Menezes, J. Katz, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 1996.

[157] M. Meuter, U. Iurgel, S.-B. Park, and A. Kummert, "Unscented Kalman filter for pedestrian tracking from a moving host," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2008, pp. 37–42.

[158] Microsoft Corporation, "VoTT: Visual Object Tagging Tool," 2017. [Online]. Available: https://github.com/Microsoft/VoTT/ (accessed: 2018-04-28).

[159] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *Proc. of Uncertainty of Artificial Intelligence (UAI)*, 2001, pp. 362–369.

[160] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding (CVIU)*, vol. 104, no. 2, pp. 90–126, 2006.

[161] A. Monroy and B. Ommer, "Beyond bounding-boxes: Learning object shape by model-driven grouping," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2012, pp. 580–593.

[162] F. Moosmann, B. Triggs, and F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests," *Advances in Neural Information Processing Systems (NIPS)*, vol. 19, pp. 985–992, 2007.

[163] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. L. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 891–898.

[164] K. P. Murphy, "Dynamic Bayesian Networks: Representation, inference and learning," Ph.D. dissertation, UC Berkeley, Computer Science Division, 2002.

[165] K. P. Murphy, *Machine learning: A probabilistic perspective*. MIT Press, 2012.

[166] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 31, no. 4, pp. 607–626, 2009.

[167] W. Nam, P. Dollár, and J. H. Han, "Local decorrelation for improved pedestrian detection," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 424–432.

[168] G. Neuhold, T. Ollmann, S. R. Bul, and P. Kontschieder, "The Mapillary Vistas dataset for semantic understanding of street scenes," in *Proc. of the International Conf. on Computer Vision (ICCV)*, 2017, pp. 5000–5009.

[169] New Car Assessment Program (NCAP), "AEB pedestrian," 2016. [Online]. Available: https://www.euroncap.com/ (accessed: 2018-04-28).

[170] A. Niculescu-Mizil and R. Caruana, "Obtaining calibrated probabilities from boosting," in *Proc. of Uncertainty of Artificial Intelligence (UAI)*, 2005, p. 413.

[171] T. Ogawa, H. Sakai, Y. Suzuki, K. Takagi, and K. Morikawa, "Pedestrian detection and tracking using in-vehicle lidar for automotive application," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 734–739.

[172] S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert, "Learning and inferring motion patterns using parametric segmental switching linear dynamic systems," *International Journal of Computer Vision (IJCV)*, vol. 77, no. 1-3, pp. 103–124, 2008.

[173] S. O'Kane, "How Tesla and Waymo are tackling a major problem for self-driving cars: Data," The Verge, 2018.

[174] O. Oktay, E. Ferrante, K. Kamnitsas, M. Heinrich, W. Bai, J. Caballero, S. A. Cook, A. de Marvao, T. Dawes, D. P. ORegan *et al.*, "Anatomically constrained neural networks (ACNNs): Application to cardiac image enhancement and segmentation," *IEEE Transactions on Medical Imaging*, vol. 37, no. 2, pp. 384–395, 2018.

[175] W. Oremus, "We finally know exactly what happened in last years fatal Tesla autopilot crash," Slate Magazin, 2017. [Online]. Available: http://www.slate.com (accessed: 2018-04-28).

[176] J. Orozco, S. Gong, and T. Xiang, "Head pose classification in crowded scenes," in *Proc. of the British Machine Vision Conf. (BMVC)*, 2009, pp. 120.1–120.11.

[177] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *International Journal of Computer Vision (IJCV)*, vol. 38, no. 1, pp. 15–33, 2000.

[178] D. Pathak, E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional multi-class multiple instance learning," *arXiv preprint arXiv:1412.7144*, 2015.

[179] A. Paul, R. Chauhan, R. Srivastava, and M. Baruah, "Advanced driver assistance systems," SAE Technical Paper, Tech. Rep., 2016.

[180] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Multi-view and 3D deformable part models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 37, no. 11, pp. 2232–2245, 2015.

[181] D. Pfeiffer, "The Stixel world: A compact medium-level representation for efficiently modeling dynamic three-dimensional environments," Ph.D. dissertation, Humboldt-Universität zu Berlin, Nov. 2011.

[182] D. Pfeiffer and U. Franke, "Towards a global optimal multi-layer stixel representation of dense 3D data," in *Proc. of the British Machine Vision Conf. (BMVC)*, 2011, pp. 51.1–51.12.

[183] Philosys, "Philosys Label Editor," 2017. [Online]. Available: https://www.philosys.de/en/products/label-editor (accessed: 2018-04-28).

[184] M. R. Pointer, "A comparison of the CIE 1976 colour spaces," *Color Research and Application*, vol. 6, no. 2, pp. 108–118, 1981.

[185] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.

[186] C. Premebida, G. Monteiro, U. Nunes, and P. Peixoto, "A lidar and vision-based approach for pedestrian and vehicle detection and tracking," in *Proc. of the IEEE Intelligent Transportation Systems Conf. (ITSC)*, 2007, pp. 1044–1049.

[187] L. L. Presti and M. La Cascia, "3D skeleton-based human action classification: A survey," *Pattern Recognition*, vol. 53, pp. 130–147, 2016.

[188] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source Robot Operating System," in *Proc. of the International Conf. on Robotics and Automation Workshops (ICRA Workshops)*, 2009, p. 5.

[189] R. Quintero, I. Parra, D. F. Llorca, and M. Sotelo, "Pedestrian path prediction based on body language and action classification," in *Proc. of the IEEE Intelligent Transportation Systems Conf. (ITSC)*, 2014, pp. 679–684.

[190] I. Rauschert and R. Collins, "A generative model for simultaneous estimation of human body shape and pixel-level segmentation," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2012, pp. 704–717.

[191] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525.

[192] E. Rehder and H. Kloeden, "Goal-directed pedestrian prediction," in *Proc. of the International Conf. on Computer Vision Workshops (ICCV Workshops)*, 2015, pp. 139–147.

[193] E. Rehder, H. Kloeden, and C. Stiller, "Head detection and orientation estimation for pedestrian safety," in *Proc. of the IEEE Intelligent Transportation Systems Conf. (ITSC)*, 2014, pp. 2292–2297.

[194] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 39, no. 6, pp. 1137–1149, 2017.

[195] N. Robertson and I. Reid, "Estimating gaze direction from low-resolution faces in video," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2006, pp. 402–415.

[196] A.-V. I. Rosti and M. J. F. Gales, "Rao-Blackwellised Gibbs sampling for switching linear dynamical systems," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2004, pp. 809–812.

[197] M. Roth, F. Flohr, and D. M. Gavrila, "Driver and pedestrian awareness-based collision risk analysis," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 454–459.

[198] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *Proc. of the ACM Transactions on Graphics (SIGGRAPH)*, vol. 23, no. 3, pp. 309–314, 2004.

[199] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, p. 211252, 2015.

[200] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A database and web-based tool for image annotation," *International Journal of Computer Vision (IJCV)*, vol. 77, no. 1-3, pp. 157–173, 2008.

[201] SAE International, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," SAE International, Tech. Rep., 2014.

[202] M. Sampathkumar, "Tesla crash: Car flips and injures five passengers," Independent, 2017. [Online]. Available: http://www.independent.co.uk (accessed: 2018-04-28).

[203] T. Scharwächter, M. Enzweiler, U. Franke, and S. Roth, "Efficient multi-cue scene segmentation," in *Proc. of the German Conference on Pattern Recognition (GCPR)*, 2013, pp. 435–445.

[204] S. Schmidt and B. Färber, "Pedestrians at the kerb - Recognising the action intentions of humans," *Transportation Research Part F*, vol. 12, no. 4, pp. 300–310, 2009.

[205] N. Schneider and D. M. Gavrila, "Pedestrian path prediction with recursive Bayesian filters: A comparative study," in *Proc. of the German Conference on Pattern Recognition (GCPR)*, 2013, pp. 174–183.

[206] M. Schrepp, A. Hinderks, and J. Thomaschewski, "User experience questionnaire," *Mensch und Computer 2017-Tagungsband: Spielend einfach interagieren*, vol. 17, p. 355, 2018.

[207] A. Schulz, N. Damer, M. Fischer, and R. Stiefelhagen, "Combined head localization and head pose estimation for video–based advanced driver assistance systems," in *Proc. of the DAGM Symposium on Pattern Recognition*, 2011, pp. 51–60.

[208] A. T. Schulz and R. Stiefelhagen, "A controlled Interactive Multiple Model Filter for combined pedestrian intention recognition and path prediction," in *Proc. of the IEEE Intelligent Transportation Systems Conf. (ITSC)*, 2015, pp. 173–178.

[209] G. A. F. Seber and C. J. Wild, *Nonlinear Regression*. John Wiley and Sons, 1989.

[210] H. Shimizu and T. Poggio, "Direction estimation of pedestrian from multiple still images," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2004, pp. 596–600.

[211] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint 1409.1556 [cs.CV]*, 2014.

[212] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 36, no. 7, pp. 1442–1468, 2014.

[213] K. Smith, S. O. Ba, J.-M. Odobez, and D. Gatica-Perez, "Tracking the visual focus of attention for a varying number of wandering people," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 30, no. 7, pp. 1212–1229, 2008.

[214] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views,"

in *Proc. of the International Conf. on Computer Vision (ICCV)*, 2015, pp. 2686–2694.

[215] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proc. of the International Conf. on Computer Vision (ICCV)*, 2017, pp. 843–852.

[216] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–12.

[217] J. Tighe and S. Lazebnik, "Superparsing," *International Journal of Computer Vision (IJCV)*, vol. 101, no. 2, pp. 329–349, 2013.

[218] F. Tost, "Images annotation programme," 2017. [Online]. Available: https://github.com/frederictost/images_annotation_programme (accessed: 2018-04-28).

[219] S. Tulsiani and J. Malik, "Viewpoints and keypoints," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1510–1519.

[220] Tzutalin, "LabelImg," 2017. [Online]. Available: https://github.com/tzutalin/labelImg (accessed: 2018-04-28).

[221] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision (IJCV)*, vol. 104, no. 2, pp. 154–171, 2013.

[222] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *International Journal of Computer Vision (IJCV)*, vol. 62, no. 1, pp. 61–81, 2005.

[223] P. Viola and M. J. Jones, "Robust real-time object detection," *International Journal of Computer Vision (IJCV)*, vol. 57, no. 2, pp. 137–154, 2004.

[224] P. Voigtländer, "ADTF: framework for driver assistance and safety systems," in *International Congress of Electronic Systems for Vehicles (VDI)*, 2007, pp. 701–710.

[225] J. Wang and M. Cohen, "Image and video matting: A survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 2, pp. 97–175, 2008.

[226] L. Wang, J. Shi, G. Song, and I.-F. Shen, "Object detection combining recognition and segmentation," in *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 2007, pp. 189–199.

[227] Y. Wang, "Pose-RCNN: Joint object detection and pose estimation," Master Thesis, University of Amsterdam, 2016.

[228] D. Ward, "Creating a global market for vehicle safety," New Car Assesment Programe, Tech. Rep., 2017. [Online]. Available: http://www.globalncap.org/wp-content/uploads/2017/06/Market-for-Vehicle-Safety.pdf (accessed: 2018-04-28).

[229] T. White, *Hadoop: The definitive guide*. O'Reilly Media, Inc., 2012.

[230] J. Whitelegg and G. Haq, "Vision Zero: Adopting a target of zero for road traffic fatalities and serious injuries," Stockholm Environment Institute, Tech. Rep., 2006.

[231] C. Wöhler and J. K. Anlauf, "A time delay neural network algorithm for estimating image-pattern shape and motion," *Image and Vision Computing*, vol. 17, no. 3, pp. 281–294, 1999.

[232] World Health Organization (WHO), "Global status report on road safety 2015," World Health Organization (WHO), Tech. Rep., 2015.

[233] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-aware convolutional neural networks for object proposals and detection," in *Proc. of the Winter Conf. on Applications of Computer Vision (WACV)*, 2017, pp. 924–933.

[234] H. Yan, W. Doerr, A. Ioffe, and H. Clasen, "Micro-doppler based classifying features for automotive radar vru target classification," in *Proc. of the International Tech. Conf. on the Enhanced Safety of Vehicles (ESV)*, 2017.

[235] M. Yanagisawa, E. Swanson, P. Azeredo, and W. Najm, "Estimation of potential safety benefits for pedestrian crash avoidance / mitigation systems," United States. National Highway Traffic Safety Administration, Tech. Rep., 2017.

[236] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Aggregate channel features for multi-view face detection," in *Proc. of the IEEE International Joint Conference on Biometrics*, 2014, pp. 1–8.

[237] R. Yang and Z. Zhang, "Model-based head pose tracking with stereovision," in *Proc. of the International Conf. on Automatic Face and Gesture Recognition (FG)*, 2002, pp. 255–260.

[238] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014, pp. 818–833.

[239] C. Zhang, L. Wang, and R. Yang, "Semantic segmentation of urban scenes using dense depth maps," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2010, pp. 708–721.

[240] S. Zhang, R. Benenson, and B. Schiele, "CityPersons: A diverse dataset for pedestrian detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3213–3221.

[241] G. Zhao, M. Takafumi, K. Shoji, and M. Kenji, "Video based estimation of pedestrian walking direction for pedestrian protection system," *Journal of Electronics (China)*, vol. 29, no. 1-2, pp. 72–81, 2012.

[242] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2879–2886.

[243] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014, pp. 391–405.

LIST OF PUBLICATIONS

- F. Flohr and D. M. Gavrila, "PedCut: an iterative framework for pedestrian segmentation combining shape models and multiple data cues," *in Proc. of the British Machine Vision Conf. (BMVC)*, 2013, pp. 66.1–66.11.

  Author contributions: F. Flohr created, implemented and evaluated the proposed method, D. M. Gavrila provided guidance and supervision.

- F. Flohr, M. Dumitru-Guzu, J. F. P. Kooij, and D. M. Gavrila, "Joint probabilistic pedestrian head and body orientation estimation," *in Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2014, pp. 617–622.

  Author contributions: F. Flohr created, implemented and evaluated proposed method, with the help of his master student M. Dumitru-Guzu who worked on temporal integration. J. F. P. Kooij helped with technical formulation of the model and mathematical details, D. M. Gavrila provided guidance and supervision.

- J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila, "Context-based pedestrian path prediction," *in Proc. of the European Conf. on Computer Vision (ECCV)*, 2014, pp. 618–633.

  Author contributions: J. F. P. Kooij created the model and experiments. N. Schneider and *F. Flohr* acquired the data, worked on visual feature extraction and on the real-time vehicle implementation. D. M. Gavrila provided guidance and supervision.

- F. Flohr, M. Dumitru-Guzu, J. F. P. Kooij, and D. M. Gavrila, "A probabilistic framework for joint pedestrian head and body orientation estimation," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 2015, vol. 16, no. 4, pp. 1872–1882.

  Author contributions: F. Flohr created, implemented and evaluated proposed method, with the help of his master student M. Dumitru-Guzu who worked on temporal integration. J. F. P. Kooij helped with technical formulation of the model and mathematical details. D. M. Gavrila provided guidance and supervision.

- X. Li, F. Flohr, Y. Yang, H. Xiong, M. Braun, S. Pan, K. Li, and D. M. Gavrila "A new benchmark for vision-based cyclist detection, " *in Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 1028–1033.

  Author contributions: X. Li implemented, modified and evaluated state-of-the-art methods, *F. Flohr* implemented and evaluated the method SP-FRCN and created the dataset together with Y. Yang. Further *F. Flohr* has build up the test vehicle for recording the dataset. H. Xiong and M. Braun helped with evaluation experiments, S. Pan, K. Li and D. M. Gavrila provided guidance and supervision

- M. Roth, F. Flohr, and D. M. Gavrila, "Driver and pedestrian awareness-based collision risk analysis," *in Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 454–459.

  Author contributions: M. Roth created, implemented and evaluated proposed method. *F. Flohr* worked on visual feature extraction of the pedestrian and helped with data aquisition and the technical and mathematical formulation of the model. D. M. Gavrila provided guidance and supervision.

- X. Li, L. Li, F. Flohr, J. Wang, H. Xiong, M. Bernhard, S. Pan, D. M. Gavrila, and K. Li, "A unified framework for concurrent pedestrian and cyclist detection," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 2017, vol. 18, no. 2, pp. 269–281.

  Author contributions: X. Li created, implemented and evaluated proposed method, with help of his student L. Li who supported the experimental part. *F. Flohr* provided supervision for dataset creation and the technical car setup. J. Wang, H. Xiong helped with the experimental parts. M.

Bernhard, S. Pan, D. M. Gavrila and K. Li. provided guidance and supervision.

- M. Braun, Q. Rao, Y. Wang, and F. Flohr "Pose-RCNN: Joint object detection and pose estimation using 3D object proposals," *in Proc. of the IEEE Intelligent Transportation Systems Conf. (ITSC)*, 2016, pp.1546–1551.

  Author contributions: M. Braun implemented and evaluated Fast-RCNN methods with stixel and lidar proposals. Q. Rao implemented Lidar proposal generation. Master student Y. Wang implemented the method Pose-RCNN together with M. Braun and *F. Flohr*. *F. Flohr* provided guidance and supervision.

- A. Aparicio, L. Sanz, G. Burnett, H. Stoll, M. Arbitmann, M. Kunert, *F. Flohr*, P. Seiniger, and D. Gavrila "Advancing active safety towards the protection of vulnerable road users: The Prospect project," *in Proc. of the International Technical Conf. on the Enhanced Safety of Vehicles (ESV)*, 2017.

  Author contributions: F. Flohr prepared the Daimler demonstrator car.

- J. F. P. Kooij, *F. Flohr*, E. A. I. Pool, and D. M. Gavrila, "Context-based path prediction for targets with switching dynamics," *International Journal of Computer Vision (IJCV)*, 2018. *[Accepted for publication, DOI: 10.1007/s11263-018-1104-4]*

  Author contributions: J. F. P. Kooij created the model and experiments. *F. Flohr* planned and performed data acquisition and annotation, and worked on visual VRU feature extraction. E. A. I. Pool developed environment context cues and helped with the experiments. D. M. Gavrila provided guidance and supervision.

- J. Bargalló, I. Cieslik, M. Kunert, J. Stoll, *F. Flohr*, M. Arbitmann, G. Burnett, D. M. Gavrila, and P. Seiniger "Next generation advanced driver assistance systems towards the protection of vulnerable road users - cyclists and pedestrians," *JSAE Annual Congress (Spring)*, 2018.

  Author contributions: F. Flohr prepared the Daimler demonstrator car.

- S. Krebs, B. Duraisamy, and *F. Flohr* "A survey on leveraging deep neural networks for object tracking," *in Proc. of the IEEE Intelligent Transportation Systems Conf. (ITSC)*, 2018, pp. 411–418.

Author contributions: S. Krebs collected relevant related works for the survey with support of B. Duraisamy. *F. Flohr* provided guidance and supervision.

- M. Braun, S. Krebs, *F. Flohr*, and D. M. Gavrila "The EuroCity Persons Dataset: A novel benchmark for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2018. *[In Review]*

  Author contributions: M. Braun, S. Krebs and *F. Flohr* integrated the relevant sensors in the vehicle and performed the data recordings. *F. Flohr* contributed with the data labeling tools. M. Braun performed the experiments, with some assistance from S. Krebs. D. M. Gavrila provided guidance and supervision.

ACKNOWLEDGEMENTS

During my scientific journey there have been a lot of people who inspired, supported and motivated me. At this point, I would like to mention some of these people in particular.

First, I would like to thank my promoter Dariu Gavrila. Thank you Dariu, for your patience, the exciting time and the many interesting topics that I could discover thanks to you. You have always asked the right critical questions and thus, you have greatly influenced the quality of my work. I am already looking forward to our further collaboration, with more classical lunch concerts and even more Kuhhandel evenings.

A big thank you goes also to Julian Kooij, a whiz in Bayesian Statistics, from whom I learned a lot here. Thank you Julian, for not only becoming a great friend who tries to teach me the dutch language, but also for accepting the co-promoter role and for supporting me till the very end in all concerns.

During my time at Daimler, I received great support from my team colleagues. First and foremost I would like to mention Ulrich Kreßel, who supported my ideas and projects all the time, Markus Enzweiler and Christoph Keller, who had always an open ear for my problems especially at the beginning of my PhD, Frank Lindner, with whom I build an autonomous city bus, Markus Braun, for his great Deep Learning expertise and his endurance and commitment in turbulent Daimler times, Markus Roth, for his extraordinary knowledge on scripting and data handling, Sebastian Krebs, for his invaluable support in PROSPECT, and Madalin Dumitru-Guzu and Yikang Wang who were fantastic

master students to work with. These amazing people became much more than just colleagues. It was a lot of fun and a honor for me to work with you all. I hope we will continue our joint work on exciting projects.

My dear friends that I know for many years have witnessed all my ups and downs. It is always a pleasure to meet you guys for a beer and a good conversation. In particular, I would like to thank Nicolas Schneider for sharing my fate as a longtime roommate at Daimler and my love for good coffee. The long working nights in the test vehicles would have been unbearable without you. A special thank you goes also to Michael Faber, for being my personal wingman in basically every situation. I am looking forward to the next co-working sessions with you and the many adventures we will experience now together with our great sons.

Thanks to my dear parents, Franziska and Berthold, I was able to pursue the academic path. Your tolerant education left me a lot of freedom and at the same time you were always there for me when things went wrong. Having such a safe shelter is one of the most valuable things one can have. It feels always good to return home. You are the best! Also my brothers, Lukas and Benjamin, showed great patience with me and it was always great to enjoy a down to earth drink with them. Thank you Lukas, for sharing your creativity and creating the beautiful book cover for this thesis.

And, of course, there is Franzi and Oskar, the brightest stars in my life. Thank you Franzi, for being my better side for already 13 years now. Without your continuous encouragement, I would have thrown in the towel many times. In March 2017 you made me the greatest gift ever, our son Oskar. Starting the morning with a smile from both of you has become the most wonderful thing I can imagine. Thank you, for accepting all my quirks and for showing me again and again what is really important in life. I love you both, to infinity and beyond.

This thesis addresses the detection, segmentation and orientation estimation of persons in visual data. In particular, the aim of this work is to establish an accurate machine representation of the Vulnerable Road Users (VRU, e.g. pedestrians, cyclists) by using image-based cues to support context-aware automated driving.

A robust detection of the VRU is achieved by applying efficient stereo-based proposals within region-based Convolutional Neural Networks. Various network and proposal configurations are compared on a newly introduced dataset focusing on the challenging detection of cyclists in urban areas.

A pixel-wise segmentation of the detected VRU facilitates higher-level, semantic scene analysis (e.g. pose estimation, activity analysis). Accurate object segmentations are gained by combining statistical shape models with multiple visual data cues within an iterative framework using a Conditional Random Field formulation.

Head and body part locations and orientations are jointly estimated from a set of orientation-specific detector responses. The applied Dynamic Bayesian Network model accounts for spatial and temporal anatomical constraints resulting in stable part localization and orientation estimates.

The inferred orientations are used to anticipate the behavior of the VRU by modeling situational awareness within a context-based Switching Linear Dynamic System. Experiments show that such context-aware models lead to a significant improvement in VRU path prediction.

Since data annotation and management are indispensable components for the development of complex machine learning applications, two software tools are proposed to support an efficient handling of sensor data and annotations.

ISBN 978-94-6375-100-1