ASTEROIDS: A Stixel Tracking Extrapolation-based Relevant Obstacle Impact Detection System

ASTEROIDS: A Stixel Tracking Extrapolation-Based Relevant Obstacle Impact Detection System

Willem P. Sanberg, *Member, IEEE*, Gijs Dubbelman, *Member, IEEE*, and Peter H. N. de With, *Fellow, IEEE*

***Abstract*—This paper presents a vision-based collision-warning system for ADAS in intelligent vehicles, with a focus on urban scenarios. In most current systems, collision warnings are based on radar, or on monocular vision using pattern recognition. Since detecting collisions is a core functionality of intelligent vehicles, redundancy is essential, so that we explore the use of stereo vision. First, our approach is generic and class-agnostic, since it can detect general obstacles that are on a colliding path with the ego-vehicle without relying on semantic information. The framework estimates disparity and flow from a stereo video stream and calculates stixels. Then, the second contribution is the use of the new *asteroids* concept as a consecutive step. This step samples particles based on a probabilistic uncertainty analysis of the measurement process to model potential collisions. Third, this is all enclosed in a Bayesian histogram filter around a newly introduced time-to-collision versus angle-of-impact state space. The evaluation shows that the system correctly avoids any false warnings on the real-world KITTI dataset, detects all collisions in a newly simulated dataset when the obstacle is higher than 0.4 m, and performs excellent on our new qualitative real-world data with near-collisions, both in daytime and nighttime conditions.**

***Index Terms*—ADAS, collision warning, time to collision, stereo vision, bayesian histogram filter.**

I. Introduction

THIS paper presents a stereo vision-based collision-warning system for assisted or automated driving. Started in the past, and recently boosted by new technology, obstacle or drivable space detection have been an active research area for intelligent vehicles [1], [2], together with early extensions to control [3]. An objective of this research is to reduce traffic accidents, predominantly by avoiding collisions. This requires detecting potential collisions accurately and timely, irrespective of whether the avoidance will be executed by a human driver or an automated control system.

The most advanced vision-based collision avoidance systems currently presented in literature rely on a combination of sensor modalities, like LIDAR, V2I or V2V communication, Radar, GNSS+IMU, cameras and HD maps [4]–[7]. The benefit of such an approach is that it facilitates redundancy over modalities in the perception system of a car. This is an important vehicle safety aspect for real-world applicability [5], *e.g.* to reduce the effect of sensor malfunctioning or to remove blind spots in the perception of the surroundings.

To this end, we propose to develop a generic forward-collision warning system using a stereo camera. Stereo cameras are increasingly employed in cars with Advanced Driver Assist Systems (ADAS), mainly for high-level semantic reasoning and scene-geometry estimation. Therefore, our research looks into further exploiting stereo vision, and aims at identifying strong and weak points of a disparity-based approach. During the past years, the so-called Stixel World algorithm has gained momentum for efficient automotive vision analysis. Originally, it presented an efficient representation of scene geometry from disparity data [8]. This has been enhanced in two ways. Taking a data view, the disparity analysis is extended with color data [9] and probabilities of semantic classes [10]. In a functional view, it has been extended with *e.g.* dynamics [11] and object recognition [12]. In our case, we want to exploit stixels in a collision-warning system for urban scenarios, where different types of traffic participants can pass close by the ego-vehicle (the ADAS-equipped car), at maximum speeds of around 50 km/h. For this purpose, we provide an end-to-end probabilistic method.

The benefit of a probabilistic method is that it can handle noise from the disparity estimation process on difficult, low-texture regions, smooth out the spatial quantization arising from the Stixel World processing, and maintaining uncertain measurements in the system. In addition, it facilitates fusing information into a larger system, for instance, to complement short-range radars that typically are employed for this task, although this is beyond the scope of this work. Fig. 1 provides an example result of our system. Summarizing, this paper addresses collision warning by exploiting probabilistic modeling of uncertain disparity and flow measurements, where the representation facilitates fusion with other ADAS processes.

The key contributions of this paper can be summarized as follows. We introduce

- a generic collision-warning system that is not limited to predefined classes or scenarios;
- a particle-sampling strategy (asteroids) for probabilistic analysis of noisy, dynamic disparity stixels;

Manuscript received July 26, 2019; revised January 14, 2020; accepted April 17, 2020. Date of publication May 4, 2020; date of current version February 24, 2021. This work was supported in part by the European Union Horizon 2020 project VI-DAS under Grant 690772 and in part by TASS International who facilitated using the PreScan software. *(Corresponding author: Willem P. Sanberg.)*

The authors are with the Department of Electrical Engineering, Eindhoven University of Technology, 5612 AZ Eindhoven, the Netherlands (e-mail: wpsanberg@gmail.com; g.dubbelman@tue.nl; p.h.n.de.with@tue.nl).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIV.2020.2992086

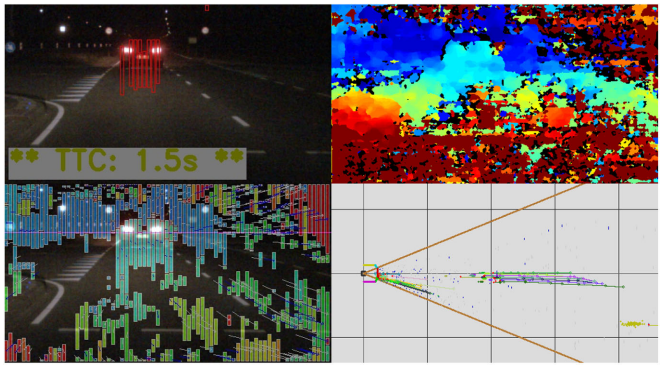

Fig. 1. Illustrating ASTEROIDS in action: generating a collision warning at night (top left) from noisy stixels (bottom left) caused by noisy disparity (top right); bottom right: top-down view of tracked potential obstacles.

- a state space that is designed specifically for collision warnings, based on axes over impact time and angle.

Our algorithm can be employed on affordable hardware and has no requirements on car connectivity or HD maps. Our validation utilizes both known and new public data, featuring both real-world and simulated data and includes scenarios at nighttime.

The remainder of this paper is organized as follows. Section II provides a more specific overview of related work. Section III presents the high-level structure of our proposed method, followed by a detailed description of the key algorithmic modules in Section III. Sections IV and V present our evaluation strategy and the corresponding results. Section VI concludes the overall work.

II. Related Work

Our research aims at exploiting stixels to generate reliable collision warnings in an urban setting, where many and multiple types of traffic participants (cars, pedestrians, cyclists, buses, etc.) can pass close by to the ego-vehicle (*e.g.* the ADAS-equipped car), at maximum speeds of around 50 km/h.

Stixels are vertical superpixels with fixed pixel width, which are produced by analyzing disparity data with the Stixel World algorithm [13]. This algorithm processes the data in a column-based manner and divides the scene into either ground or fronto-parallel, rectangular obstacle patches, which are assigned a single disparity value. This forms an efficient representation of the scene geometry and has a proven value for different subdomains. For instance, the disparity Stixel World has been fused with deep neural nets for both semantic scene segmentation [10] and instance segmentation [14], where stixels have been also clustered to detect and recognize objects [12]. Additionally, the Stixel World analysis can provide a supervisory function in an online training setup for free-space segmentation [15], [16]. Given this broad promising range of applications, we want to extend it even further and explore the strengths and weaknesses of a stixel-based approach to extract relevant collision-warning information.

We start from the bare disparity stixels, but aim at designing a generic method, so that it can always benefit from the more advanced versions of the Stixel World proposals under development, e.g. with object clustering or semantic labels.

In related work on collision-warning systems, we have observed several limitations that we mitigate or avoid altogether. First of all, most current systems are limited to highway scenarios [6], [7], [17]. Although those can operate at higher vehicle speeds, the systems will not be able to deal with street crossings, non-vehicle traffic or oncoming traffic, which is not a fundamental limitation in our method.

Second, most collision-warning systems rely on vision with trained pattern recognition. For instance, a MobilEye system will only recognize cars, trucks, motorcycles, cyclists and pedestrians, with the additional limitation to fully visible rear-ends for vehicle detection [18]. Similarly, the system of Cherng *et al.* classifies situations into five pre-defined dangerous motions that are limited to the ego-direction (such as cut-ins) and can handle only regularly-sized cars, just one of which may be in view in a scenario [19]. Both these approaches rule out handling crossing, oncoming, and passing traffic, in contrast to our algorithm.

The mono-camera based system of Ess *et al.* deploys several class-specific detectors, for instance for cars and pedestrians. Subsequently, they rely on class-specific motion models to predict object trajectories for enhanced accuracy [20]. In contrast, our system can handle any tangible object, without knowing its type. This aspect makes the system more robust and widely applicable, since it is not limited to the set of objects for which it was trained.

Moreover, we model objects in a very generic way and aim at a procedure that also does not rely on high-level knowledge such as infrastructure layout [5] or intention estimation [21]. In the same light, we do not rely on V2V or V2I-communication streams and/or centralized roadside compute [6], [7], but instead focus on a pure independent ego-car strategy.

Since our framework concerns tracking elements over time and predicting their future path, a motion model and a data-association strategy should be selected. Models for motion are available in different levels of complexity, varying by the incorporation of steering angles, yaw rate, acceleration and velocity [22], which can also be employed in parallel and fused afterwards to handle cluttered measurements in highly dynamic urban environments [23]. Since we aim at an class-agnostic analysis and execute on a medium-level stixel representation, and not on object level, we do not model model higher order dynamics. Instead, we use simple constant-velocity kinematics without any rotational component in this work and rely on the power over having multiple stixels per object and generating multiple asteroids in a probabilistic fashion to justify this simplification. Regarding the problem of data association for tracking, we propose a strategy similar to the extended SORT algorithm [24], which is a box-overlap analysis, enhanced with appearance modeling. In contrast to [24], we simplify the appearance encoding into a histogram, which does neither require training on class-specific examples nor has to execute a neural net during the association process.

These constraints will inherently limit the time horizon within which our predictions are reliable. Our goal is to explore these boundaries and identify the strengths and weaknesses of the

stixel-based approach, rather than providing a stand-alone all-encompassing collision-warning solution. However, our method is able to utilize additional information by design, if it would be available from other system modules.

Thirdly, other previous work addresses free-space detection (the area in front of the vehicle where it can drive) [2], [9], [15], [16], which is a related or even the dual problem of collision warning. With our proposed method, we explicitly add motion estimation, motion prediction and timing into the system and analyze the *obstacle* part of the scene instead of the *ground* part. This extends the analysis to dynamic data instead of using only static data.

Related to that, systems in literature typically use geometry-oriented state spaces, for instance by storing locations in occupancy grids, and then derive motion as a secondary signal [25], [26]. In contrast, our design of the state space directly stores the relevant information, namely time-to-collision and angle-of-impact. This is in line with our goal of designing a collision-warning system. This will be explained in more detail in the next section.

An earlier version of this system has been presented at a high level in [27]. This current work provides two main extensions. First, we present the full algorithmic details of all processing blocks, several of which have been updated and improved. Second, we provide an extended experimental analysis on more data, illustrating the practical applicability of the design. More specifically, the previous version generated, at best, minimally 12 false warnings on KITTI data, whereas the current system correctly generates no false warnings on the same set and performs reliably on newly recorded real-world data with true potential collisions. Even though this is not an automotive-grade industrial validation, it shows both theoretical and practical feasibility of the proposed system.

Summarizing, we focus our design on an urban setting with medium driving speeds, with nearby traffic and obstacles. Furthermore, we do not limit ourselves to specific classes of objects or types of scenarios and aim at generic collision cases and broad usage. To further generalize, we avoid relying on semantic information on traffic layout or participant intentions and restrict ourselves in this work to affordable sensor hardware without V2V or V2I communication infrastructure. Our state space is designed to directly model the quantity of interest, namely angle and time of impact.

III. Proposed Method

A. System Architecture

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

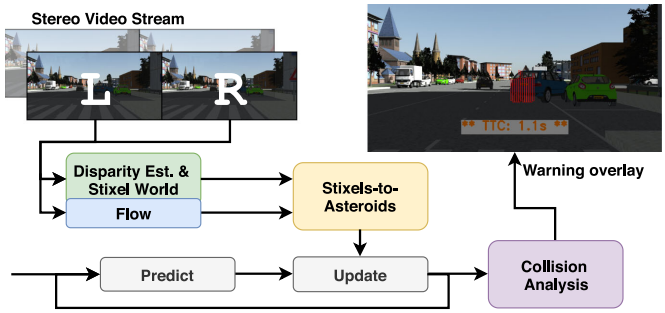

Fig. 2. High-level schematic overview of our collision warning system. It extracts flow and disparity from stereo video and generates asteroids from stixels to analyze potential collisions.

is to employ a probabilistic processing pipeline. This facilitates maintaining any uncertain information in the system for as long as possible.

Motion particle sampling in time and direction is the first probabilistic aspect. We propose a probabilistic method that is specifically designed to capture typical noise in our stixel-based approach. It revolves around particle sampling to model the probability density function of obstacle motion. In short, it consists of the following steps. First, in the process of estimating the velocity of a stixel, the corresponding uncertainty is modeled as well. Second, we sample moving particles, which we call *asteroids*, from that velocity model to propagate its uncertainty into the collision-warning process. The model captures uncertainty both in speed and in direction. If both are measured with a high confidence, this will generate a very dense ball of asteroids, traversing through space. However, a stixel with a confident direction but with an uncertain speed will generate a laser-beam like stripe of asteroids: they might hit the car all at the same point, but will arrive in a time interval. Alternatively, a stixel with a confident speed but with an uncertain direction will generate a set of asteroids in a wave-front, potentially hitting the car from different directions at similar times. When both direction and speed are uncertain, a dispersed cloud of asteroids can be expected. More details on this approach are provided in Section III-F. Concluding, this modeling fluently combines accurate and uncertain measurements of noisy, dynamic data, so that it can be analyzed further in the processing pipeline.

Representation of collision data is the second key design choice for the following stages. Since analyzing and predicting dynamic processes in general benefit from filtering over time, we propose to utilize a Bayesian histogram filter. A histogram offers an efficient yet flexible representation. It can represent multi-modal distributions directly without enforcing high-level assumptions on the modeled data, which suits our aim of providing class-agnostic collision warnings.

Our Bayesian histogram filter models a state space that contains the probability of a collision with the ego-vehicle from a certain angle at a certain time-to-collision. Naturally, the Bayesian filter encompasses a *prediction* and a *measurement-update* phase which are both repeated at each time step. Additionally, we have a Collision Analysis module that interprets the state and generates warnings accordingly. Fig. 2 portrays

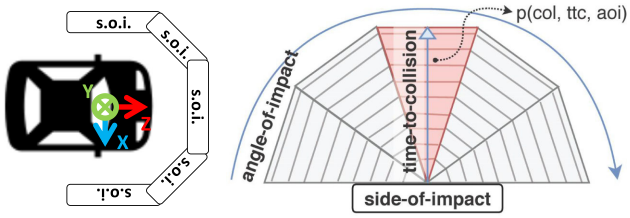

Fig. 3. Schematic visualization of the sides of impact (s.o.i.) around the ego-vehicle with the employed coordinate system (left) and the state space with discretized angle-of-impact, time-to-collision and $p(col, ttc, aoi)$ versus $p(\neg col, ttc, aoi)$ for a single side of impact (right). Our evaluation is focused on the area highlighted in red for the front side of the ego-vehicle.

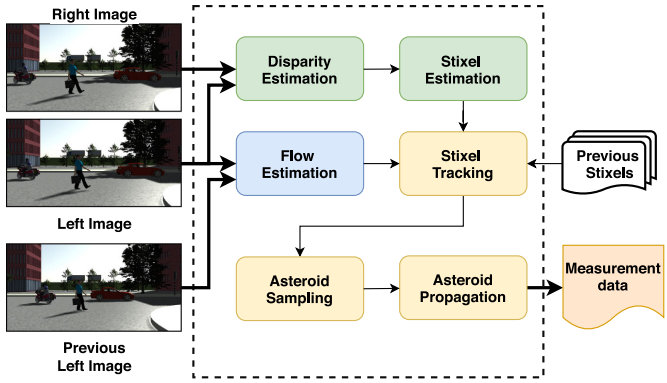

Fig. 4. Schematic overview of the measurement update stage, generating the data likelihood for the Bayesian filter of our collision-warning system. Note that this a more detailed view of the top-left block of Fig. 2.

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

B. State Space

Since the goal of our system is to provide collision warnings, we introduce a state-space design that is directly suited to address such warning information. To this end, we define a three-dimensional state space, the axes of which are *time-to-collision* (ttc), *angle-of-impact* (aoi) and *collision* (col). Fig. 3 shows a schematic visualization.

The system monitors such a state space for the five sides-of-impact (s.o.i.) of the vehicle, as shown in Fig. 3. We focus on the frontal view in this work, since that is within the field of view of the sensor setup. We discretize the time axis with steps of half the sample time of the input data stream (this equals 0.05 seconds) to a maximum of 5 seconds, and split the angle-of-impact uniformly in five non-overlapping ranges of 36^o each. To obtain a complete joint probability distribution, we calculate the belief in *no collision* $p(\neg col, ttc, aoi)$ and the *collision belief* $p(col, ttc, aoi)$ for each angle and time pair.

C. Bayesian Filter: Prediction

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

D. Bayesian Filter: Measurement Update

The principal stage of our histogram filter is the measurement update and consists of several steps, depicted in Fig. 4. This figure presents a more detailed view of the top-left region of Fig. 2. The aim is to convert the stereo video data at the input via stixel and asteroid processing into a likelihood $p(measurement|col, aoi, ttc)$. First, the stereo image pair and the previous left camera image are used to estimate disparity and flow. The disparity is processed with the Stixel World algorithm to build fronto-parallel rectangular superpixels. Our main contribution is in the introduction of the following three processing blocks, depicted in yellow, which are discussed in the next subsections.

E. Measurement Update: Stixel Tracking

The *Stixel Tracking* block first extracts the median 2D optical flow for each stixel. Next, it translates this to 3D-world motion by trying to match each stixel to the corresponding stixel of the previous set. This matching process first moves a current stixel to its previous image position, using its optical flow as pixel translation. Then, it analyzes the overlapping stixels of the previous set at that location. The stixel is disregarded if less then 75% of its moved area falls within the image, or if less than 50% has overlap with previous obstacle stixels. The overlap analysis is conceptually illustrated in Fig. 6 and further explained below. If there is only one overlapping stixel, this is considered the match. If there are multiple overlapping stixels, these candidates are analyzed in a small selection process. First, candidates that have an overlap ratio of less then $1/(N_{oa}+1)$ are disregarded, where N_{oa} is the number of candidates. If this still leaves multiple candidates, the stixel is matched to the candidate with the highest Bhattacharyya coefficient, comparing the stixel texture-wise via a normalized one-dimensional color histograms (10 bins per channel). If no such candidate exists, no match is made. Each resulting match is assigned a corresponding confidence from this overlap analysis, c_{oa}, which is defined as follows:

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

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

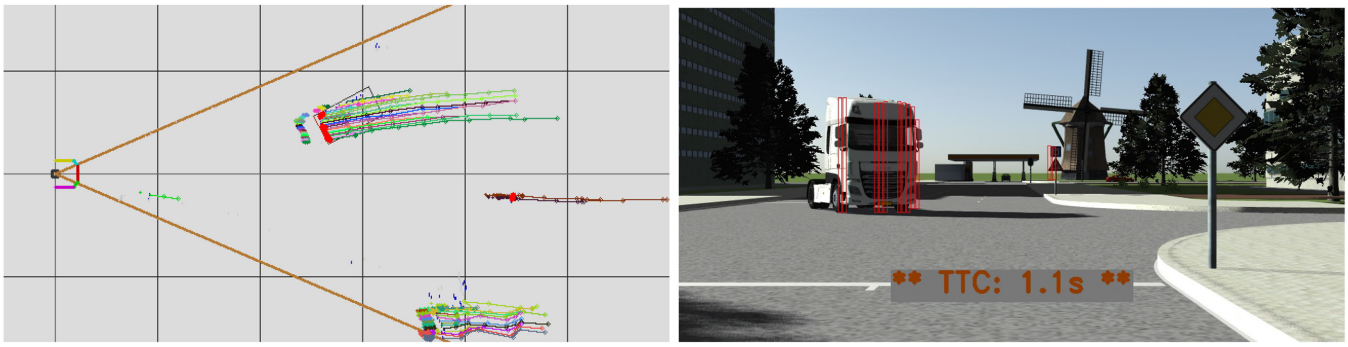

Fig. 5. Left: our processing visualized in a top-down view with the ego-vehicle at the center left (moving to the right), the camera field-of-view in dark orange; five colored sides-of-impact and grid lines at every 10 m. Furthermore, the figure shows stixel tracks, sampled asteroid clouds and detected colliding asteroids (in bright red). Right: corresponding camera image with the collision warning overlay.

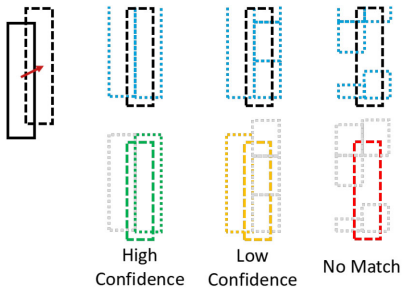

Fig. 6. Conceptual illustration of the stixel tracking process by overlap analysis: move a stixel according to its image flow (top, black striped) and analyze the overlapping stixels of the previous set (top, dotted light blue); this leads to a match with a certain confidence (bottom middle, green/yellow), or no match at all (red, right).

process, both stixels that could not be matched and stixels that are clear outliers are removed, to avoid cluttering the subsequent process, while still facilitating a high inclusion of measurements. The tracked stixel should have a confidence of more than 0.5; be within relevant range of the ego-vehicle (at most 30 m to the left or right, 2.5 m up or down; up to 60 m in front) and should have a relative speed below 150 km/h, considering that the maximum allowed absolute speed is 50 km/h. The colored tails of the stixels in Fig. 5 illustrate the result of the tracking process.

F. Measurement Update: Asteroid Sampling

The tracked stixels are supplied to the subsequent *Asteroid Sampling* block, which generates so-called asteroids for each stixel. An *asteroid* is a particle with a trajectory sampled from two one-dimensional Gaussian distributions, one for each of the x- and z-velocities, so that

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

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

The average velocity in each axis is calculated from the N_{trk} previous positions in the stixel track, hence (analogous for z):

$$\overline{v_x} = \frac{1}{T_{\text{sample}} \cdot N_{\text{trk}}} \sum_{k=0}^{N_{\text{trk}}-1} x_{t-k} - x_{t-k-1}, \tag{3}$$

The variances of distributions in Equation (2) are derived by extending the standard uncertainty propagation in disparity estimation using a camera pinhole model with the stixel estimation and our matching process. First of all, the error propagation for the velocity estimate, using standard probability theory calculation rules [28], results in

$$\sigma^2_{v_x}(\boldsymbol{s}) = \frac{\sigma^2_{x_t} + \sigma^2_{x_{t-1}}}{T^2_{\text{sample}}}, \tag{4}$$

and a similar propagation for z. Second, the stixel-position variances can be defined from applying two camera pinhole models. These cameras have a stereo camera baseline b, the u-coordinate of the left-camera's principal point u_0 and left camera's focal length f_u. The obtained disparity estimation process comes with uncertainty σ^2_{disp}, which is fixed at 0.5 pixels. This modeling leads to the following equations:

$$\sigma^2_{x_t}(\boldsymbol{s}) = \frac{\sigma^2_{disp}}{h} \cdot \left(\frac{b \cdot (u_{c,t} - u_0)}{d^2_t}\right)^2,$$

$$\sigma^2_{x_{t-1}}(\boldsymbol{s}) = \frac{\sigma^2_{disp}}{c_{oa} \cdot h} \cdot \left(\frac{b \cdot (u_{c,t-1} - u_0)}{d^2_{t-1}}\right)^2,$$

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

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

where the variables from the stixel under analysis h, u_c, d and c_{oa} represent height, central u-coordinate, disparity, and overlap-analysis confidence, respectively. Intuitively, stixels that have a larger height also have a more certain x- and z-position, since each row of the stixel can be considered an additional

measurement. Additionally, a sub-optimal overlap confidence ($c_{oa} < 1$) will increase the uncertainty in the estimate of the previous position.

Third, the amount of asteroids that should be generated for each stixel depends on several aspects, which are included in the following equation:

$$N_{\text{ast}}(\boldsymbol{s}) = \mathcal{A}(\boldsymbol{s}) \cdot \rho_{\text{ast}} \cdot c_{\text{fit}}(\boldsymbol{s}) \cdot c_{\sigma_d^2}(\boldsymbol{s}). \tag{5}$$

The core value in this equation is $\mathcal{A}(\boldsymbol{s})$, which is the stixel surface in m^2, calculated by translating all four stixel u, v, d corner-points to 3D world coordinates. This surface is multiplied with the asteroid density (ρ_{ast}), a system parameter, to come to an initial number of asteroids. However, the equation also incorporates two confidence values, that both can reduce the number of generated asteroids. The first one, $c_{\text{fit}}(\boldsymbol{s})$, is adapted from [29] and defined as follows:

$$c_{\text{fit}}(\boldsymbol{s}) = 1/(1 + \exp\left(e_{\text{obstacle}}(\boldsymbol{s}) - e_{\text{ground}}(\boldsymbol{s})\right)), \tag{6}$$

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

$$e_{\text{obstacle}}(\boldsymbol{s}) = \tfrac{1}{h} \sum_{v \in v_c \pm h/2} |d_v - d|, \tag{7}$$

$$e_{\text{ground}}(\boldsymbol{s}) = \tfrac{1}{h} \sum_{v \in v_c \pm h/2} |d_v - \Delta d_g \cdot (v_c - v) - d|, \tag{8}$$

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

The confidence $c_{\text{fit}}(\boldsymbol{s})$ expresses how well the stixel model fits the raw disparity it covers, knowing that the optimization process explores two options (ground or obstacle). It compares fitting either a fronto-parallel surface or a sloped surface to the condensed single disparity column in the stixel. The other confidence, $c_{\sigma_d^2}(\boldsymbol{s})$, is the normalized inverted variance of the disparity within the stixel region. This is more generic than the previous one, since it also considers the fact that a stixel spans multiple columns. Both these confidence values aim to decrease the chance of generating false asteroids from stixels in noisy disparity data, by reducing the generated amount.

The top-down view in Fig. 5 shows the sampled asteroid clouds as colored blobs at the end of stixel tracks. The asteroid clouds from the trees (on the right of the ego-vehicle) are larger, illustrating more uncertainty in those measurements.

G. Measurement Update: Asteroid Propagation

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

The propagation process relies on a constant velocity model, no advanced dynamic models are applied at the current stage. The constant velocity model is a reasonable choice given the goal of offering a generic, class-agnostic analysis. Naturally, this constraint will limit the time horizon where our predictions are reliable. The goal is to explore these boundaries and identify the strengths and weaknesses of the stixel-based approach, rather than providing a stand-alone all-encompassing collision-warning solution. However, note that our method is able to utilize additional information by design, if it would be available from other system modules.

Performing the collision assessment based on a linear trajectory extrapolation can be solved efficiently as a standard geometric line-segment intersection problem, as presented in [30]. We formulate both the trajectory of the asteroid and the side-of-impact lines with an origin point ($\underline{\alpha}_o$ and $\underline{\text{soi}}_o$) and a direction vector ($\underline{\alpha}_v$ and $\underline{\text{soi}}_v$), and find τ and ζ such that

$$\underline{\alpha}_o + \tau \cdot \underline{\alpha}_v = \underline{\text{soi}}_o + \zeta \cdot \underline{\text{soi}}_v. \tag{9}$$

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

$$(0 < \zeta < |\text{soi}|) \quad \wedge \quad (0 < \tau < \infty), \tag{10}$$

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

For the truck at the left of the scene in Fig. 5, the asteroids are clearly projected in front of the object (marked in bright red) from analyzing the corresponding stixels tracks.

H. Measurement Update: From Histogram to Distribution

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

$$p(m_{\text{ast}}|col, aoi, ttc) = 2/\rho_{\text{ast}} \cdot m_{\text{ast}}/\rho_{\text{ast}}, \tag{11}$$

$$p(m_{\text{ast}}|\neg col, aoi, ttc) = 2/\rho_{\text{ast}} \cdot (1 - m_{\text{ast}}/\rho_{\text{ast}}). \tag{12}$$

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

I. Collision Analysis on the State Space

The collision analysis block (see Fig. 2) processes the state and generates warnings if necessary. This module completes the system pipeline and facilitates assessing the reliability of the analysis in a tangible way.

First of all, this block extracts a collision probability for each state cell from the joint probability, by marginalizing over the

collision axis, so it calculates $p(col|ttc, aoi)$:

$$p(col|ttc, aoi) = \frac{p(col, ttc, aoi)}{p(col, ttc, aoi) + p(\neg col, ttc, aoi)}. \quad (13)$$

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

1) CFAR: Peak Detection: The next step is to identify peaks in the probability distribution that correspond to potential collisions. This process addresses the fact that the asteroids in the histogram are sampled from the noisy tracked stixel data, and hence, they travel towards the car as a dispersed cloud. Since this shows similarities to detecting objects in noisy RADAR data, we propose to employ a well-established method from that field and use a Constant False-Alarm-Rate (CFAR) detection scheme [31], [32]. CFAR is an adaptive thresholding technique to find relevant peaks against noisy background clutter. In theory, it provides the desired detections at the cost of a pre-defined false-alarm rate, hence the name. This is based on assumptions on the distribution of background clutter. We now briefly describe this, adapted to our context.

A CFAR detector checks if the probability in a cell is a local maximum and higher than a certain threshold. This threshold is derived from the neighboring cells to adapt it to the local noise caused by outlying measurements. We treat each *angle-of-impact* (aoi) as an independent sequence of measurements, which means that our CFAR neighborhood is one dimensional, along the *time-to-collision*-axis (ttc) only. Therefore, we leave out the aoi index for brevity in the following equations. Formally, our CFAR collision-peak detection $C_{\text{cfar}}(ttc)$ is defined with the following set of equations:

$$C_{\text{cfar}}(ttc) = \ldots$$

$$(p(col|ttc) > T_{\text{cfar}}(ttc)) \ \wedge \ \left(ttc \equiv \underset{\tau \in \Theta_{\text{cfar}}}{\arg\max}\, p(col|\tau)\right) \quad (14)$$

$$T_{\text{cfar}}(ttc) = \frac{\alpha_{\text{cfar}}}{N_{\text{cfar}}} \sum_{\tau \in \Theta_{\text{cfar}}} p(col|\tau), \quad (15)$$

$$\alpha_{\text{cfar}} = N_{\text{cfar}}(p_{fa}^{-1/N_{\text{cfar}}} - 1), \quad (16)$$

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

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

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

Fig. 5 presents an example result where the bright red stixels on the front of the truck are stixels that cause the generated collision warning, visible in the top-down view and also in the overlay on the camera image.

IV. Evaluation Approach

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

A. Data

Our validation process is performed on three different data sets: two with real-world data and one with simulated data.

The simulated PreScanStereoCollision data (PSSC) is newly made for this research with the PreScan software package [33] and exported in KITTI format for compatibility. This simulated data is included in our evaluation to test actual collisions and easily evaluate different relevant scenarios. We have created 5 sequences in the PreScan environment: *Straight*, *Figure-8*, *Y-crossings-Fast*, *Y-crossings-Slow* and *Mixed*. Fig. 7 shows example frames of each sequence. Sequence *Straight* is a large rectangular trajectory with head-on collisions with static objects of decreasing sizes (*e.g.* from truck to car, down to kids) and an empty road with common side objects. Sequence *Figure-8* is similar, but now on a curved road, so that the heading changes constantly. The *Y-crossings* sequences contain a straight trajectory with different objects approaching on collision course from the right at consecutive y-crossings. In the *Fast* version, each participant moves at its own nominal speed, while in the *Slow* one, speeds are decreased so to match the maximum relative speed of the *Straight* sequence. Finally, the *Mixed* sequence is a busy, fully dressed city center with multiple traffic participant approaching from various directions (either safe or on collision). The simulated stereo camera has a baseline of 30 cm, a resolution of 1024×512 pixels and a field of view of 46.2×24.1 degrees.

Additionally, we evaluate our system on the KITTI-tracking dataset. This KITTI data has no collisions and only a handful of near-collisions, but a crucial aspect is to quantify any false alarms

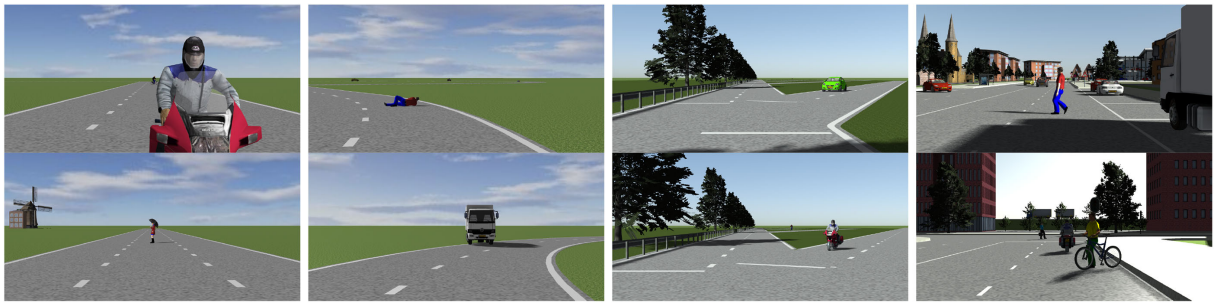

Fig. 7. Examples of our PreScanStereoCollision (PSSC) sequences; from left to right: *Straight*, *Figure-8*, *Y-crossings* and *Mixed*

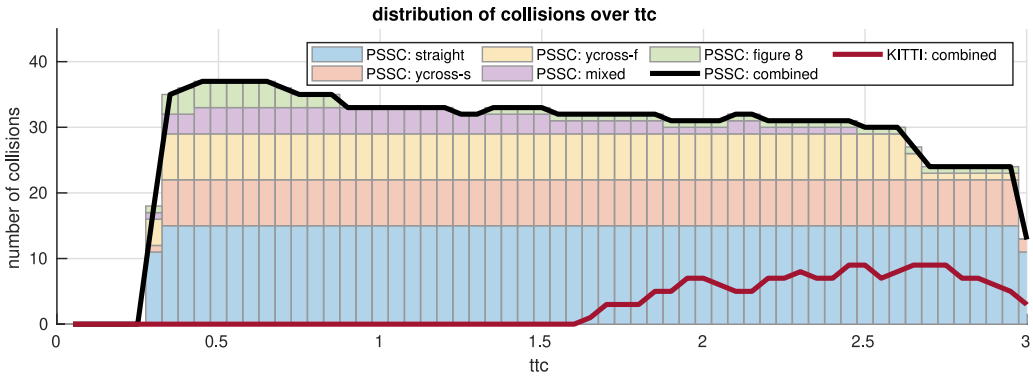

Fig. 8. Distribution of collisions over ttcs for the PSSC and KITTI data sets; the stacked histogram shows the different PSSC sub sequences. It shows that the KITTI recordings contains very few potential collisions, explaining the need for the complementary simulated PSSC data.

on real-world data. The evaluation requires ego motion as well as true object positions. Hence, the evaluation is limited to the training set of KITTI-tracking. This is the only part of the data set for which the annotated object bounding boxes and positions are available, which can be exploited to generate ground-truth collision warnings. We generate stixels on the surface of object bounding boxes, and set their motion according to the annotated motion of the object. These stixels are then used to generate a single asteroid each, with fixed motion that is extrapolated to produce the reference ttc labels.

On top of that, we have recorded a real-world dataset, *TUE&ACNL*, at the Eindhoven University of Technology campus (TUE), the Automotive Campus Netherlands (ACNL) at Helmond and the roads in between. During these recordings, the car is driven around the TUE campus in normal traffic for 30 minutes and is also steered towards near-collision with other traffic or static obstacles. The recordings at and towards ACNL are partially in normal traffic and partially on temporarily closed roads. This data has been captured under both overcast and bright sunny conditions, including several cases of strong back light. Notably, several sequences were recorded during nighttime. This dataset has no annotations of true obstacle positions and motion for a full quantitative evaluation. However, it offers a valuable qualitative insight into the collision warnings that the system generates in real-world conditions, since we have used a regular automotive-grade stereo camera. Table I summarizes the properties of the employed annotated data, regarding the duration and the number of frames and collision events.

TABLE I
EVALUATION DATA OVERVIEW

Dataset	camera[1]		#Pos	#Neg
PSSC	30 cm; 1024 px;	Frames	880	1,914
(4.7 minutes)	46^o; 10 Hz	Events	40	n.a.
KITTI	54 cm; 1242 px;	Frames	155	7,811
(13.3 minutes)	80^o; 10 Hz	Events	40	n.a.
TUE+ACNL[2]	30 cm; 480 px;	Frames	23,000[3]	
(63 minutes)	44^o; 6 Hz	Events	$\pm$100	n.a.

[1] Provided are: baseline, image width, horizontal field-of-view and frame rate.
[2] TUE&ACNL has no ground-truth annotations.
[3] Roughly 5,000 of these frames were recorded at night.

B. Metrics

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

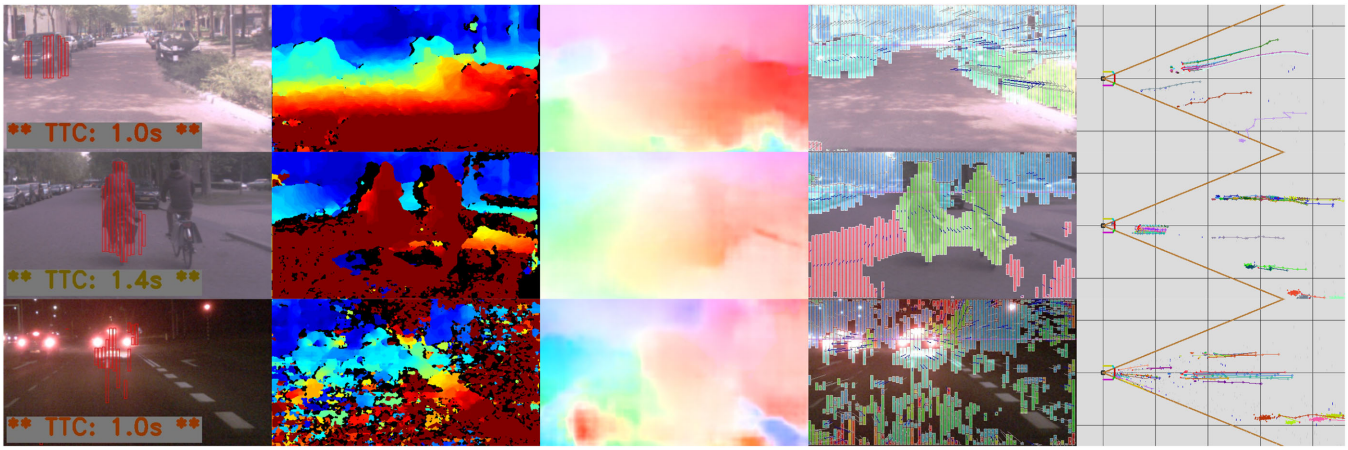

Fig. 9. Three typical collision warnings illustrating clean results from good (top), medium (middle) or noisy (bottom) input; from left to right: left camera image with warning overlay, disparity data, flow data, stixels with flow vector, top-down view. Note that the ego-vehicle is cornering in the first example, so the warning looks false but is correct.

quantified by calculating the Recall, the Precision and their harmonic mean (F_1 score), both before and after peak tracking.

C. Time Range

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

To link our evaluation to real-world conditions, we rely on stopping-distance guidelines, as used by the NACTO [34] and NHTSA [35]. They provide ballpark figures for feasible deacceleration, that are said to range from 6 m/s^2 for a reasonably skilled driver to 9.8 m/s^2 for a professional driver under good conditions. Since our system is designed for urban scenery, the ego-vehicle speed is around 50 km/h. In that case, it would require somewhere between 1.4–2.3 s to fully stop the ego-vehicle, depending on driver skills and conditions. Therefore, if the collision-detection module is integrated tightly into the car control system (*e.g.* with automated emergency braking), it should operate reliably at least up to 1.4 s. However, if the module purely generates warnings to assist a human driver, it should operate reliably to up to 2.3 s.

Fig. 8 shows the distribution of collision events in our data over time-to-collision. The events in our simulated data are distributed rather homogeneously between 0.3 and 3.0 s. The graph of the KITTI-event distribution, however, confirms that this data was recorded during a clean drive. Namely, the handful of short, potential collision events are never closer than 1.5 s. Therefore, this part of the experimental validation is focused on avoiding false warnings on the KITTI data within the above-mentioned time ranges, while obtaining a high F_1 score on the PSSC in that same time range.

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

D. Experiments

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

Additionally, we will test the influence of the quality of the input data by selecting different algorithms or settings to generate the disparity and flow data. For disparity, we compare using the traditional, widely adopted Semi-Global Block Matching (SGBM) algorithm [36] to using a newer, state-of-the-art deep learning-based method, namely DispNet [37]. For flow estimation, we rely on FlowNet2 [38], also based on deep learning. The authors of FlowNet2 have presented several architectures and made them publicly available. The different available versions vary highly in inference speed, with a trade-off against pixel-level performance quality. This can be exploited to experimentally quantify the robustness of our data processing against degraded input flow data. In turn, this will offer relevant insights on the trade-off between system latency and performance quality. Since our our system is stixel-based rather than pixel-based, we aim at being robust to these lower-quality, yet faster versions of flow estimation.

In the design of our state space and by the structure of the ego-vehicle's impact bubble, the system is able to handle collisions from all directions and at different sides of impact. However, the evaluation will be limited to the detection of collisions at the front of the vehicle. The cause of this constraint is that the annotated real-world data has been recorded with a single,

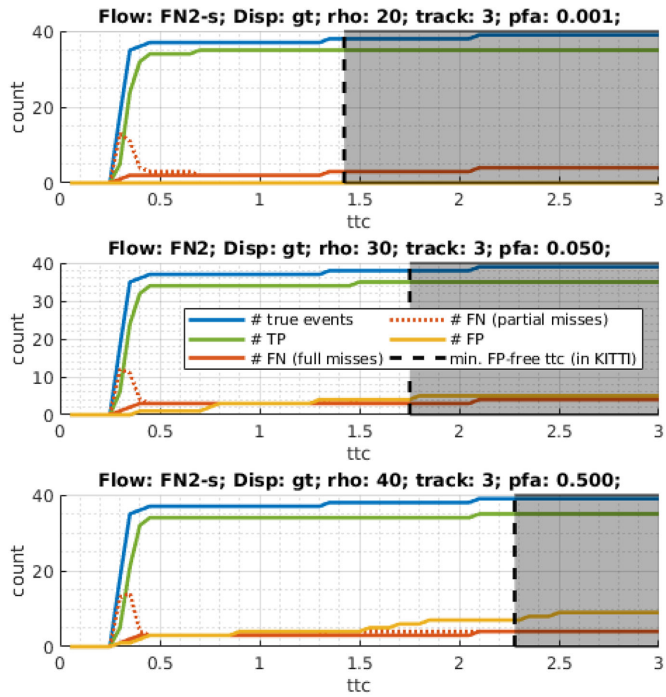

Fig. 10. Best performance on PSSC using settings that produce no false positives on the KITTI data set with ttc <1.4 s (top), ttc <1.75 s (middle) and ttc <2.3 s (bottom).

forward looking stereo camera, so that it is currently not feasible to validate this functionality.

V. Results

A first indication of the results is provided in Fig. 9. It illustrates typical performance on low-quality flow and noisy disparity, whereas our probabilistic approach is still able to extract relevant information.

A. Quantitative Evaluation on KITTI and PSSC

As a first quantitative evaluation, we present the performance with respect to the ego-vehicle breaking times, as discussed in Section IV-C. Fig. 10 portrays the performance of three system configurations that do not detect any false events on the KITTI dataset for the use case of an integrated system (no false positives with ttc <1.4 s, top graph), the use case of a human-in-the-loop (no false positives with ttc <2.3 s, bottom graph) and an intermediate case (no false positives with ttc <1.8 s, middle graph). Within the subset of configurations that comply with that constraint, we present the one with the highest F_1 on the simulated PSSC data. On PSSC, these settings suffer from up to three false negatives, all in the Fig. 8 sequence. The main cause of these misses is the curved ego-motion, which (1) makes the potential collisions short, barely being the minimum required for the event detection module, and (2) does not fit the prediction step, only considering straight motion. On top of that, one collision is with a man lying down on the road (second picture in Fig. 7), which is so low positioned that it is barely represented with stixels. CFAR detects a peak at ttc =0.6 s, which is too late for the peak tracking to activate. This shows that our current system is vulnerable concerning objects lower than around 0.4 m, and could be better adapted to curved ego-motion.

Secondly, Fig. 11 provides an analysis on how different system parameters influence the detection performance, using the PSSC data. Each row shows a different parameter: the method of disparity estimation, the method of flow estimation, the maximal length of stixel tracks, the asteroid density and the CFAR parameter p_{fa}. We accumulated the results of all parameter combinations, and generated the surfaces by averaging all sub-experiments with a specific value of the parameter under test. The left set of graphs show the peak-detection results, the right set those after peak tracking. Both present the Recall, Precision and F_1 scores.

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

A noteworthy observation is how little the system performance is impacted by the choice of the flow method. Using the smallest FlowNet2 version (*FN2-s*) yields practically identical performance to using the full version, although the former one can be executed roughly 17 times faster than the latter (7 ms on an GTX-1080 GPU), at the cost of a drop in pixel-accuracy of the optical-flow result of up to a factor of 2 [38]. This shows the potential in robustness of combining a superpixel strategy with probabilistic sampling and filtering (Section III-A).

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

A similar analysis for the impact of system configuration on the results on KITTI data is provided in Fig. 12. Since there are so few actual potential collisions, we only focus on the number of false positives here, which should be preferably low. The surface plots show that most false CFAR peaks (before tracking) occur at a large ttc. However, no false *events* occur at high ttc. This can be explained by the fact that the measurements at large ttc are more uncertain and tracking is not yet able to support the estimation, which leads to inconsistent peak detections within the CFAR module. Subsequently, the peak-tracking step filters these out, thereby improving the system robustness.

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

B. Timing

The algorithmic contributions are implemented in C++ and tested on a desktop PC (Xeon E5-1660 0 CPU @3.30 GHz × 12; 15.6 GB). On the KITTI data, the Stixel World algorithm takes roughly 20 ms. The bottleneck within our proposed blocks is

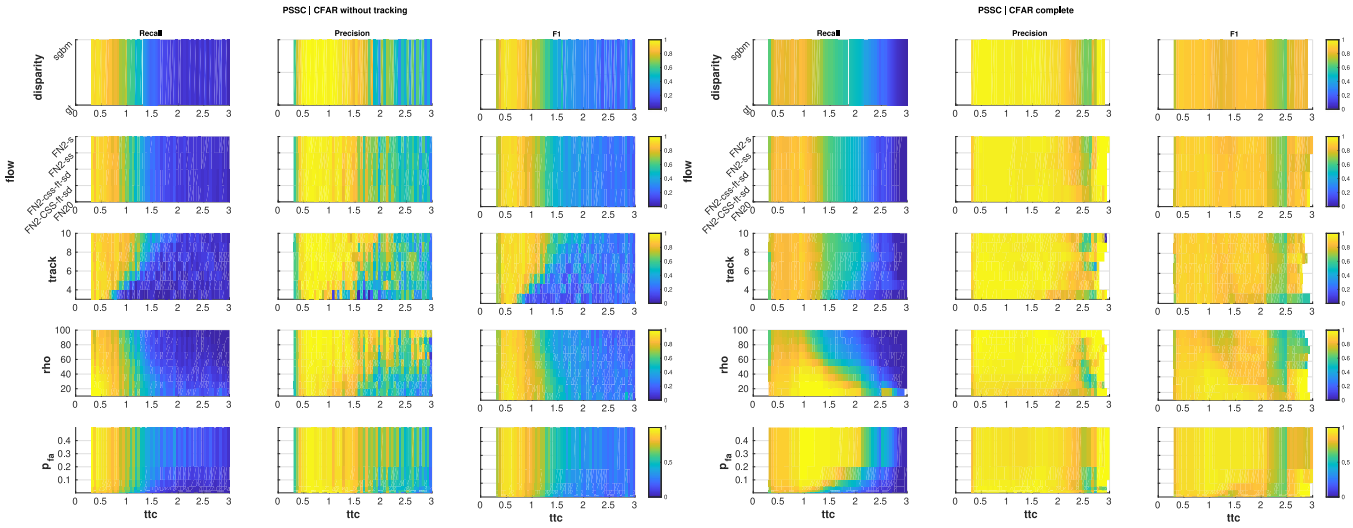

Fig. 11. Impact analysis of system configuration on the system performance using PSSC data, split over different parameters and plotted over time-to-collision. The color yellow represents a desired high score. Note that reducing the quality of the flow and/or disparity has little impact on the performance (top 2 rows).

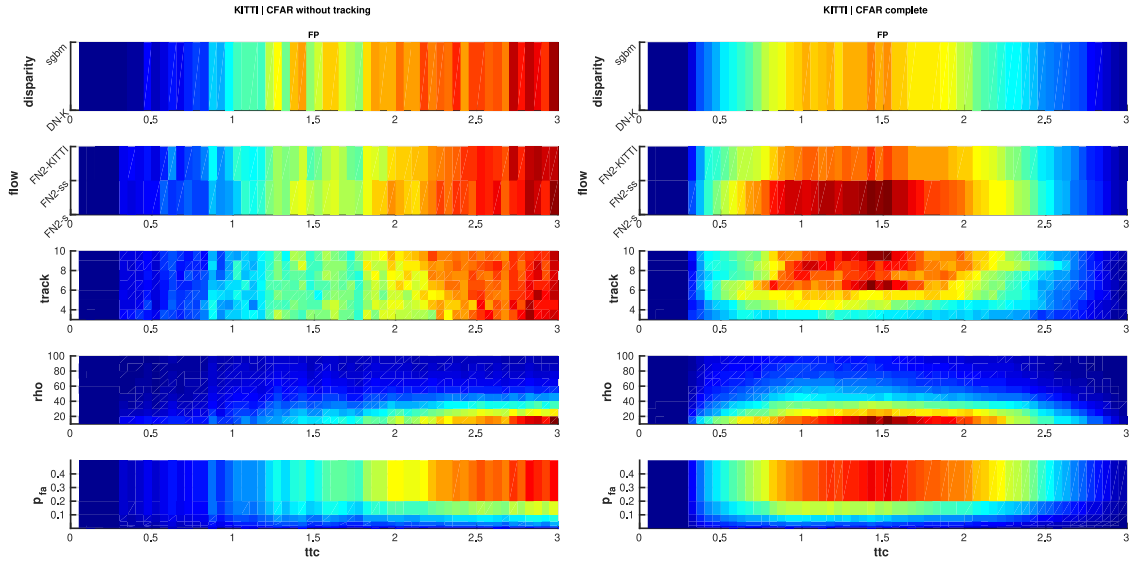

Fig. 12. Analysis of the *relative* impact of system configuration on performance with KITTI data, split over different parameters and plotted over time-to-collision. Because of the low actual potential collisions, we show only false positives. Color towards red represents an undesired (relatively) high FP count. The color ranges are stretched individually to emphasize relative performance within each parameter; comparing results between parameters or with and without peak tracking in an absolute sense is not the goal here.

the Stixel Tracking module, requiring 35-45 ms at present. It is faster on the PSSC data (15–20 ms), since the stixel segmentation is much cleaner, which indicates that removing clutter stixels prior to the matching process can speed up processing. Asteroid Sampling, their propagation including the collision check, and the histogram filter need 1–4 ms, the CFAR detection 0-1 ms. Together, this results in a processing speed of 15-17 fps, which is sufficiently fast for real-time operation on the 10-fps datasets.

C. Qualitative Evaluation on TUE&ACNL

This section presents the qualitative results on the TUE&ACNL data: real-world recordings of 63 minutes in normal traffic and on closed roads, partially during the night, with several intentional near-collisions. Fig. 13 presents four examples of typical ASTEROIDS performance on near-collisions. The first three examples show frames that were 0.5 s apart, the snapshots of the rightmost example (with the small pole) are 1/6 of a second apart, since it was only shortly on a collision trajectory. Even though we cannot quantify the estimated ttcs, the warnings generated by the system seem natural and plausible to the driver. Moreover, there was not a single false warning during the whole recording. Three-quarters of the data was recorded in bright sunny weather, causing sharp shadows, high contrast, temporal flicker, direct sunlight and reflections, which our setup could all handle. The rest of the data was captured

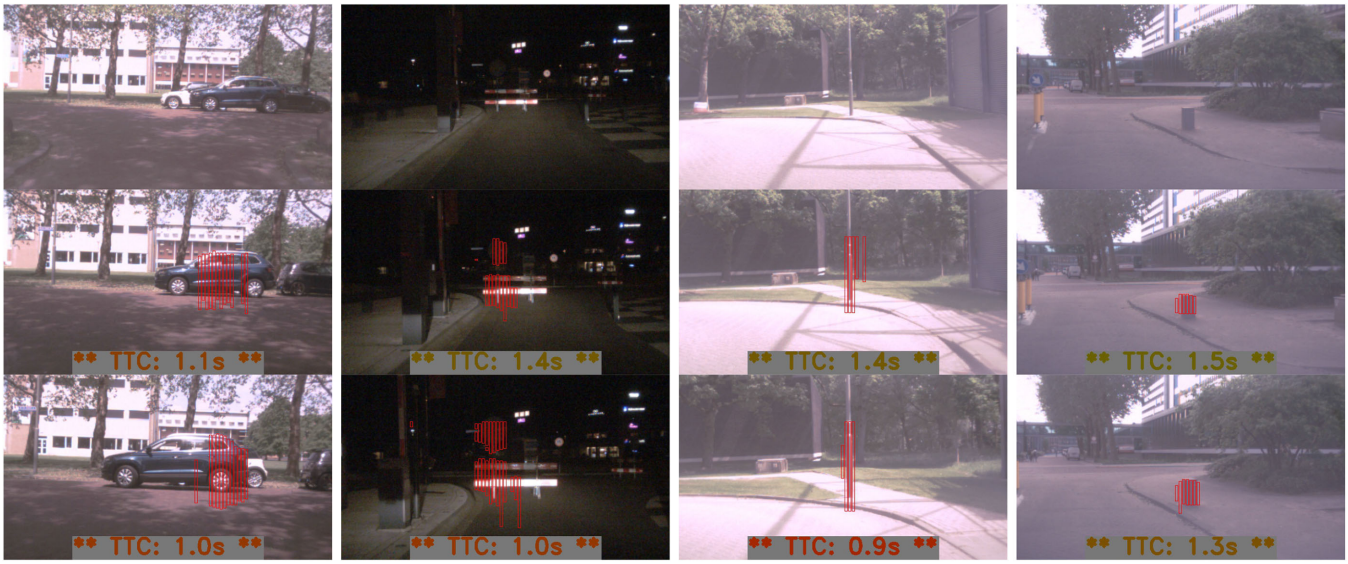

Fig. 13. Examples of our system in action on TUE&ACNL data, in each column three snapshots per collision event; from left to right: a car crossing and slowing down; a road works fence at night; a slim lamppost and a low pole at the side of the road.

during nighttime. The system was still able to generate warnings for near-collisions in the dark, although they typically occurred later (earliest at $ttc \approx 1.5$ s). In conclusion, this experiment supports our proof-of-concept evaluation with the findings on the KITTI and PSSC datasets and shows promising real-world applicability.

VI. Conclusion

This paper has presented a vision-based collision-warning system for ADAS in intelligent vehicles. The approach is class-agnostic as it detects general obstacles that lay on a collision trajectory with the ego-vehicle without relying on semantic information. This is in contrast with most current systems, that rely on pre-trained pattern recognition and are limited to predefined object classes or situations. Our framework estimates disparity and flow from a stereo video stream, extracts stixels, and samples so-called asteroids based on an uncertainty analysis of the measurement process to model potential collisions. This is all enclosed in a Bayesian histogram filter around a time-to-collision versus angle-of-impact state space. The end-to-end probabilistic approach is specifically designed to handle the noisy disparity and flow data, so that the system does not require an accurate and computationally expensive estimation of those signals. The evaluation shows that the system correctly avoids false warnings on the real-world KITTI dataset, detects all but one collisions in a newly simulated dataset, and performs excellent on our new qualitative real-world data with near-collisions, both during daytime and nighttime conditions.

References

[1] D. M. Gavrila and V. Philomin, "Real-time object detection for 'smart' vehicles," in *Proc. IEEE Int. Conf. Comput. Vision*, 1999, vol. 1, pp. 87–93.

[2] R. Labayrade, D. Aubert, and J.-P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through 'v-disparity' representation," in *Proc. IEEE Intell. Veh. Symp.*, 2002, vol. 2, pp. 646–651.

[3] C. Thorpe, M. H. Hebert, T. Kanade, and S. A. Shafer, "Vision and navigation for the carnegie-mellon navlab," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 3, pp. 362–373, May 1988.

[4] J. Ziegler *et al.*, "Making Bertha drive-an autonomous journey on a historic route," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, Apr. 2014.

[5] M. Aeberhard *et al.*, "Experience, results and lessons learned from automated driving on Germany's highways," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 1, pp. 42–57, Jan. 2015.

[6] D. Lee and H. Yeo, "Real-time rear-end collision-warning system using a multilayer perceptron neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3087–3097, Nov. 2016.

[7] X. Xiong, L. Chen, and J. Liang, "A new framework of vehicle collision prediction by combining SVM and HMM," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 699–710, Mar. 2018.

[8] H. Badino, U. Franke, and D. Pfeiffer, "The stixel world-A compact medium level representation of the 3d-world," in *Proc. Joint Pattern Recognit. Symp.*, 2009, pp. 51–60.

[9] W. P. Sanberg, G. Dubbelman, and P. H. N. de With, "Extending the stixel world with online self-supervised color modeling for road-versus-obstacle segmentation," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2014, pp. 1400–1407.

[10] L. Schneider *et al.*, "Semantic stixels: Depth is not enough," in *Proc. IEEE Intell. Vehicles Symp.*, 2016, pp. 110–117.

[11] D. Pfeiffer and U. Franke, "Efficient representation of traffic scenes by means of dynamic stixels," in *Proc. IEEE Intell. Veh. Symp.*, 2010, pp. 217–224.

[12] M. Enzweiler, M. Hummel, D. Pfeiffer, and U. Franke, "Efficient stixel-based object recognition," in *Proc. IEEE Intell. Veh. Symp.*, 2012, pp. 1066–1071.

[13] D. Pfeiffer and U. Franke, "Towards a global optimal multi-layer stixel representation of dense 3d data," in *Proc. Brit. Mach. Vision Conf.*, 2011, pp. 1–12.

[14] T. Hehn, J. F. P. Kooij, and D. M. Gavrila, "Instance stixels: Segmenting and grouping stixels into objects," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2019, pp. 2542–2549.

[15] W. P. Sanberg, G. Dubbelman, and P. H. N. de With, "Color-based free-space segmentation using online disparity-supervised learning," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2015, pp. 906–912.

[16] W. P. Sanberg, G. Dubbelman, and P. H. N. de With, "Free-space detection with self-supervised and online trained fully convolutional networks," in *Proc. EI - Auton. Veh. Mach.*, 2017, vol. 2017, pp. 54–61.

[17] L.-C. Liu, C.-Y. Fang, and S.-W. Chen, "A novel distance estimation method leading a forward collision avoidance assist system for vehicles on highways," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 4, pp. 937–949, Apr. 2017.
[18] MobilEye, *User Manual - Series 6 (DOC000600 REV A02 - ENG)*, Accessed: Oct. 18, 2015. [Online]. Available: https://drive.google.com/file/d/0B99trsbO9s2eRlJiNjlwQ1hzUm8/view
[19] S. Cherng, C.-Y. Fang, C.-P. Chen, and S.-W. Chen, "Critical motion detection of nearby moving vehicles in a vision-based driver-assistance system," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 70–82, Mar. 2009.
[20] A. Ess, K. Schindler, B. Leibe, and L. V. Gool, "Object detection and tracking for autonomous navigation in dynamic environments," *J. Robot. Res.*, vol. 29-14, pp. 1707–1725, 2010.
[21] C. G. Keller and D. M. Gavrila, "Will the pedestrian cross? a study on pedestrian path prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 494–506, Apr. 2014.
[22] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *Proc. IEEE 11th Int. Conf. Inf. Fusion*, 2008, pp. 1–6.
[23] M. P. Muresan and S. Nedevschi, "Multimodal sparse lidar object tracking in clutter," in *Proc. IEEE 14th Int. Conf. Intell. Comput. Commun. Process.*, Sep. 2018, pp. 215–221.
[24] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 3645–3649.
[25] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and tracking the driving environment with a particle-based occupancy grid," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1331–1342, Dec. 2011.
[26] C. Laugier *et al.*, "Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety," *IEEE Intell. Transp. Syst. Mag.*, vol. 3, no. 4, pp. 4–19, Oct. 2011.
[27] W. P. Sanberg, G. Dubbelman, and P. H. N. de With, "From stixels to asteroids: Towards a collision warning system using stereo vision," in *Proc. EI - Auton. Veh. Mach*, Jan. 2019, pp. 215–221.
[28] R. D. Yates and D. J. Goodman, *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers*. 2nd Ed. Hoboken, NJ, USA: Wiley, 2005.
[29] S. Ramos, S. Gehrig, P. Pinggera, U. Franke, and C. Rother, "Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 1025–1032.
[30] B. Groenen, "Stixel motion prediction for collision warning," Dept. Elect. Eng., Eindhoven Univ. Technol., SPS-VCA group, Eindhoven, Netherlands, Tech. Rep. 1321, 2018.
[31] H. Rohling, "Radar CFAR thresholding in clutter and multiple target situations," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-19, no. 4, pp. 608–621, Jul. 1983.
[32] B. R. Mahafza, *Radar Signal Analysis and Processing Using MATLAB*. Boca Raton, FL, USA: CRC Press, 2010, ch. 7.10, pp. 293–295.
[33] M. Tideman, "Scenario-based simulation environment for assistance systems," *ATZautotechnology*, vol. 10-1, pp. 28–32, Jan. 2010. [Online]. Available: https://doi.org/10.1007/BF03247153
[34] "Vehicle stopping distance and time," Accessed: Mar. 28, 2006. [Online]. Available: https://nacto.org/docs/usdg/vehicle_stopping_distance_and_time_upenn.pdf
[35] "Safety1nnum3ers: Speeding," Accessed: Aug. 2015. [Online]. Available: https://one.nhtsa.gov/nhtsa/Safety1nNum3ers/august2015/S1N_Speeding-Aug%ust2015_812008.pdf
[36] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.
[37] N. Mayer *et al.*, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 4040–4048.
[38] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 2462–2470.

Willem P. Sanberg (Member, IEEE) received the BSc. and MSc. degrees in electrical engineering and a certificate in Technology Management from the Eindhoven Univ. of Technology, The Netherlands in 2013. His PhD research aims at improving the understanding of 3D modeled environments for intelligent vehicles by developing efficient yet robust methods that handle the dynamic environment of everyday traffic. He works in international projects with governmental, industrial, and research partners.

Gijs Dubbelman (Member, IEEE) is an Assistant Professor with the Eindhoven University of Technology and focuses on signal processing technologies that allow mobile sensor platforms to perceive the world around them. He obtained his PhD. research in 2011 on the topic of Visual SLAM. In 2011 and 2012 he was a member of the Field Robotics Center of Carnegie Mellon's Robotics Institute, where he performed research on 3-D computer vision systems for autonomous robots and vehicles.

Peter H. N. de With (Fellow, IEEE) received his Ph.D. degree from the University of Technology Delft, The Netherlands. After positions at Philips Research, University Mannheim, LogicaCMG and CycloMedia, he became Full Professor at the Eindhoven University of Technology. He is an International Expert in surveillance for safety/security and was involved in multiple EU projects on video surveillance analysis with the Harbor of Rotterdam, Dutch Defense, Bosch Security, TKH-Security, ViNotion, etc. He is Board Member of DITSS and R&D advisor to multiple companies. He is IEEE Fellow, has co-authored over 300 papers on video analysis, systems and architectures, with multiple awards of the IEEE, VCIP, and EURASIP.