https://doi.org/10.2352/ISSN.2470-1173.2019.15.AVM-034


# From Stixels to Asteroids: Towards a Collision Warning System using Stereo Vision

***Willem P. Sanberg, Gijs Dubbelman and Peter H.N. de With***
***Department of Electrical Engineering, Eindhoven University of Technology, the Netherlands***

## Abstract

*This paper explores the use of stixels in a probabilistic stereo vision-based collision-warning system that can be part of an ADAS for intelligent vehicles. In most current systems, collision warnings are based on radar or on monocular vision using pattern recognition (and ultra-sound for park assist). Since detecting collisions is such a core functionality of intelligent vehicles, redundancy is key. Therefore, we explore the use of stereo vision for reliable collision prediction. Our algorithm consists of a Bayesian histogram filter that provides the probability of collision for multiple interception regions and angles towards the vehicle. This could additionally be fused with other sources of information in larger systems. Our algorithm builds upon the disparity Stixel World that has been developed for efficient automotive vision applications. Combined with image flow and uncertainty modeling, our system samples and propagates asteroids, which are dynamic particles that can be utilized for collision prediction. At best, our independent system detects all 31 simulated collisions (2 false warnings), while this setting generates 12 false warnings on the real-world data.*

## Introduction

This paper presents a stereo vision-based collision warning system for assisted or automated driving. For various years, obstacle detection and its counter problem, i.e. drivable or freespace detection, have been an active research direction in the field of intelligent vehicles [1, 2], together with early extensions on control [3, 4]. The objective of this research is to reduce traffic accidents, predominantly by avoiding or mitigating collisions. This requires detecting potential collisions accurately and timely, irrespective of whether the avoidance will be executed by a human driver or automatically by a follow-up system.

The 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 [5, 6]. 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 [6], for example, 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, independent, forward collision warning system using a stereo camera. Stereo cameras are increasingly employed in cars with ADAS, mainly for high-level semantic reasoning and scene geometry estimation. Therefore, our research explores how to exploit stereo vision further. During the past years, the so-called Stixel World algorithm has gained momentum for efficient automotive vision analysis.

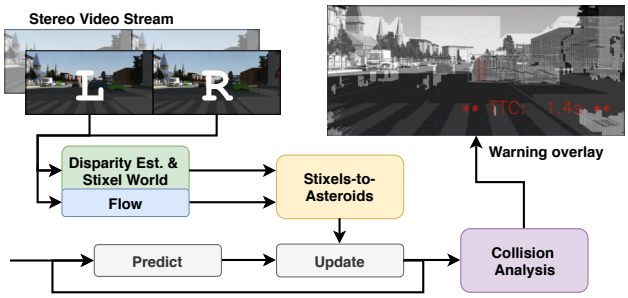



***Figure 1.*** *Schematic overview of our collision warning system. It extracts flow and disparity from stereo video and generates asteroids from stixels to analyse potential collisions.*

Originally, it addressed representing scene geometry efficiently from disparity data [7]. Taking a data view, the disparity analysis has been extended with color data [8] and semantic class probabilities [9]. In a functional view, it has been extended with e.g. dynamics [10] and object recognition [11]. In our case, we want to exploit stixels in a collision warning system. For this purpose, we provide a fully probabilistic method to facilitate fusing it into a larger system, for instance, to complement short-range radars that typically are employed for this task.

The remainder of this paper is organized as follows. Section '*Related Work*' provides a more specific overview of related work. Section '*Method*' presents our proposed method, followed by the description of our evaluation strategy and their results in Sections '*Validation*' and '*Results*'. Section '*Conclusion*' concludes the overall work.

## 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-equiped car).

Stixels are vertical superpixels with fixed pixel width, which are produced by analyzing disparity data with the Stixel World algorithm [12]. This algorithm processes the data in a column-based way and divides the scene into either ground or fronto-parallel, rectangular obstacle patches that are assigned a single disparity value. This forms an efficient representation of the scene geometry and has a proven value for different subdomains. The disaprity Stixel World has been fused with deep neural nets for semantic scene understanding [9], stixels have also been clustered to detect and recognize objects [11], and the Stixel World analy-

sis can provide a supervisory function in an online training setup for free-space segmentation [13, 14]. Given this broad promising range of applications, we want to extend it even further and explore how to extract relevant collision warning information, starting from the bare disparity stixels. We aim at designing a generic method, so that it can always benefit from the more advanced versions of the Stixel World proposals under development, e.g. that realize object clustering or find semantic labels.

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 [15]. 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 warnings 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 [16]. 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 [17]. 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 [18]. In contrast, our system can handle any tangible object, without knowing it's type. That makes the system more robust and widely applicable, since it is not limited to a 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 [6] or intention estimation [19] in itself. Additionally, we do not employ any advanced dynamic models and instead rely on simple constant-velocity kinematics. 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, note that 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, 8, 13, 14], 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.

## Method

The high-level structure of our system consists of a Bayesian histogram filter. It models a state space that contains the belief in 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 that are repeated at each time step. Additionally, we have a Collision Analysis module that interprets the state and generates warnings accordingly. Figure 1 portrays the whole system. The state space and the three processing blocks are described in the following subsections.

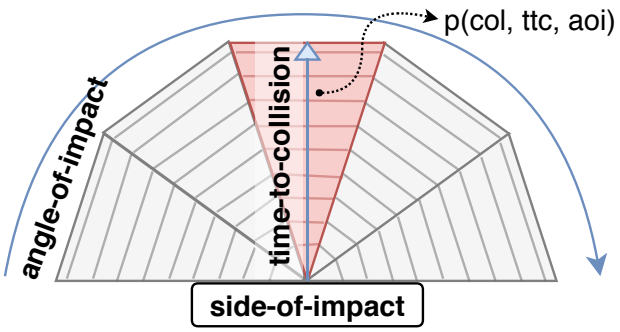



***Figure 2.*** *Schematic visualization of the state space with discretizations over angle-of-impact, time-to-collision and $p(col,ttc,aoi)$ versus $p(\neg col,ttc,aoi)$ for a single side of impact. We focus our evaluation on the area highlighted in red for the front side of the ego-vehicle.*

### State Space

We define a three-dimensional state space, the axis of which are *time-to-collision* ($ttc$), *angle-of-impact* ($aoi$) and *collision belief* ($col$). Figure 2 shows a schematic visualization. The system can monitor a state space like this for every desired side of the vehicle (as long as it is within the field of view of the sensor setup), although we focus on the frontal view in this work. We discretize the time axis with steps of half the sample time of the input data stream (this equals 0.05 seconds), and split the angle-of-impact uniformly in five non-overlapping ranges of 36° each. To have a complete joint probabilistic distribution, we calculate both the belief in *no collision* $p(\neg col,ttc,aoi)$ and the collision belief $p(col,ttc,aoi)$ for each angle and time pair.

### 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 the amount of bins over the time-to-collision axis that corresponds to the sampling rate of the camera. Additionally, we apply a box-averaging filter with the same size as the shift. This introduces a dispersion of the belief to reflect the uncertainty in the prediction step, i.e. the process noise.

### Measurement Update

The principal stage of our histogram filter is the measurement update and consists of several steps, depicted in Figure 3. The aim is to convert the stereo video data at the input via stixel and asteroid processing into a probability $p(measurement|col,aoi,ttc)$. 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 (which are colored yellow in the diagram).

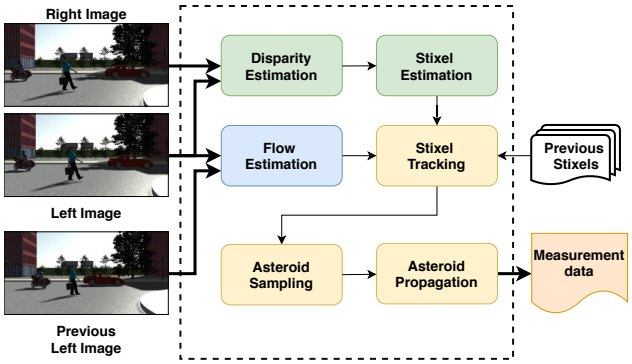



***Figure 3.*** *Schematic overview of the measurement update stage of our collision warning system.*

### *Stixel Tracking*

The *Stixel Tracking* block first extracts the 2D image 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. In this step we also remove both stixels that cannot be matched and clear outliers, to avoid cluttering the subsequent process. The tracked stixel should have a confidence probability 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. The colored tails of the stixels in Figure 4 illustrate the result of the tracking process.

### *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 three one-dimensional Gaussian distributions, one for each of the $x$-, $y$- and $z$-velocities. These distributions follow the standard uncertainty propagation in disparity estimation and a camera pinhole model. Additionally, we incorporate the stixel estimation and our matching process into the error propagation process. More specifically, a taller stixel has a more certain position estimate, whereas a low-confident stixel match will lead to a less certain previous stixel position. Similarly, a longer stixel track leads to a more certain asteroid velocity; and stixels with a low confidence or a high internal disparity variation will generate less asteroids. Moreover, the amount of asteroids that is sampled for a stixel is a function of the metric stixel surface and the asteroid density (a system parameter). The BEV in Figure 4 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, showing more uncertainty in those measurements.

### *Asteroid Propagation*

The third block, *Asteroid Propagation*, takes the cloud of asteroids, propagates them linearly along their generated trajectory (i.e., a constant velocity model) and monitors which ones are going to impact a safety bubble around the ego-vehicle and the time to impact. This can be solved efficiently as a standard geometric line-segment intersection problem. For the truck at the left of the scene in Figure 4, the asteroids are clearly projected in front of the object (marked in bright red) from analyzing the corresponding stixels tracks.

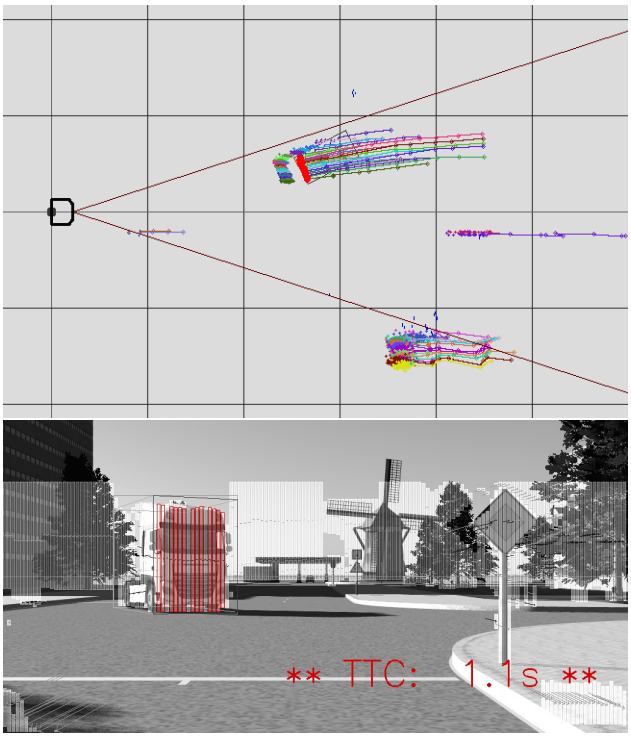



***Figure 4.*** *Top: our processing visualized in a Bird's Eye View (BEV) with the ego-vehicle at the center left (moving to the right), stixel tracks, sampled asteroid clouds and detected collisions (in bright red) and grid lines at every 10 m; the dark red lines represent the $aoi$ for the current warning, not the FOV. Bottom: corresponding camera image with the collision warning overlay.*

The results of asteroid propagation are represented in a 2D histogram, of which the bins match the configuration of the state space. This histogram is then translated into a joint probability distribution with a linear model, as illustrated in Figure 5. The number of asteroids at which the rule saturates is set equal to the the asteroid density. This means that a fully confident surface of 1 $m^2$ will generate enough asteroids to saturize a histogram bin, independent of the density parameter. After this translation, the joined probability distribution is fed into the Bayesian filter-update stage. Additionally, the collision probabilities are further processed in the *Collision Analysis* block, described below.

### *Collision Analysis*

This stage analyzes the state and generates warnings if necessary. It adds robustness by filtering at two levels: at the time-to-collision estimation and at the collisions-over-frame analysis.

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 $p(col|ttc,aoi)$.

Secondly, it smoothens $p_{col}$ over the time axis by convolving it with a Hanning window of length 11 in a single impact direction and then executes a peak detector to suppress false peaks[1]. This

[1]We rely on the peak detection method as presented in `http://nbviewer.ipython.org/github/demotu/BMC/blob/master/notebooks/DetectPeaks.ipynb`, using default parameters.

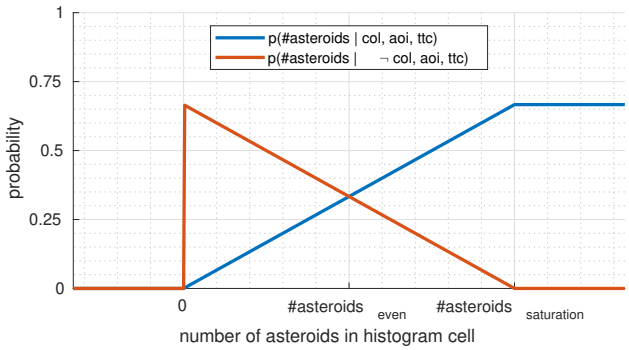



***Figure 5.*** *Illustration of translating the histogram data into the joint probability distribution.*

step addresses the fact that the asteroids in the histogram are sampled, and hence, they travel towards the car as a dispersed cloud. The peak detector tries to find the center of that cloud, which is assumed to correspond to the true time-to-collision.

This analysis results in a collision detection for each frame individually. In reality, collisions do not occur just in a single frame, they are events spanning multiple frames. Therefore, thirdly, we consider these per-frame detections as a noisy time series and filter those to generate the final warnings on potential collision events. We enforce the detections within the event to be consistent by thresholding the frame-to-frame differences in detected $ttc$, and then employ a simple moving average filter. Effectively, the system generates a warning if and only if there are 3 frames with a consistently detected collision within 6 consecutive frames. This suppresses spurious detections and simultaneously is cautious towards missed detections at the frame level.

Figure 4 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 bird's eye view and also in the overlay on the camera image.

## Validation

This section explains the validation of the proposed system by addressing the selection of data sets, performance metrics and the performed experiments.

### *Data*

We use two different data sets in our validation process: one 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 and exported in KITTI format for compatibility. We have included this simulated data into our evaluation, so that we can test actual collisions and easily evaluate different relevant scenarios. We have created 4 sequences in the PreScan environment: *Straight*, *Figure-8*, *Y-crossings* and *Mixed*. Sequence *Straight* is a large rectangular trajectory with head-on collisions with static objects of decreasing sizes (*e.g.* from truck to car to motorcycle down to regular pedestrian to kid). Sequence *Figure-8* contains similar obstacles but now on a curved road, so that the ego-vehicle is constantly changing its heading. The *Y-crossings* sequence contains a straight trajectory for the ego-vehicle, with different objects approaching on collision course from the right at consecutive $y$-crossings. Finally, the *Mixed* sequence is a busy, fully dressed city center. It contains all kinds of vehicles and pedestrians that are either on a safe or on a collision course, straight and from different angles. Figure 6 shows example frames of each subset. The simulated stereo camera has a baseline of 30 cm, a resolution of $1024 \times 512$ pixels and a field of view of $46.2 \times 24.1$ degrees.

Additionally, we evaluate our system on the KITTI-tracking dataset. This KITTI data has no collisions and only a handful of near collisions, but a crucial aspect is to quantify the false alarm rate on real-world data. We use the training subset, to exploit the annotated object bounding boxes and positions to generate ground-truth collision warnings.

We set the relevant $ttc$ range for our system to operate in as $ttc \in [0.3, 2.0]$ seconds. Some frames in the datasets have potential collisions outside of that range; those frames are ignored in the analysis. Table 1 depicts the properties of the employed data.

**Table 1. Dataset overview**

| Dataset | | #Pos | #Neg | #Ign |
|---|---|---|---|---|
| PSSC | Frames | 491 | 1,747 | 227 |
| ( 4.1 minutes) | Events | 31 | n.a. | n.a. |
| KITTI | Frames | 22 | 7,868 | 76 |
| (13.3 minutes) | Events | 2 | n.a. | n.a. |

### *Metrics*

We quantify the performance of our collision warning system at two levels: at the lower level of frames and at the higher level of collision events that span multiple frames. Ultimately, the goal is to design a system that handles the events properly. Consequently, it is acceptable that the system misses a collision in some frames if it still detects the corresponding event relying on other frames. We show the evaluation on a frame basis to give an idea on the intermediate, raw level of performance. The frame level contains more samples by definition, which increases the reliability of the analysis, while it also provides insights into strengths or weaknesses in the pipeline.

We measure True Positive Rate (TPR) and False Alarm Rate (FAR) on a per-frame basis. It is conceptually misleading to properly define a *negative event*. Therefore we do not measure the number of true negatives at the level of events. Instead, we calculate the harmonic mean of the TPR and Precision of collision events to provide a balanced view of the system performance.

### *Experiments*

The core of the validation goal 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 robustness of the system, we have evaluated the performance over different settings of the core system parameters, being the asteroid density, the maximum tracking length and the threshold on $p_{col}$ within the *Collision Analysis* block.

With these experiments, the focus of this validation is on the full system itself, and not on the algorithms that generate the disparity and flow input data. We have decided to keep those parameters constant for each data set. Specifically, we use SGBM [20] and FlowNet2 [22] for PreScan, and DispNet [21] and FlowNet2

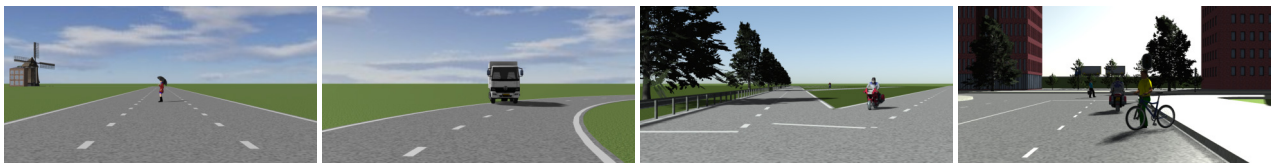


***Figure 6.*** *Examples of our PreScanStereoCollision data. From left to right: subsequences* Straight, Figure-8, Y-crossings *and* Mixed

(both tuned on KITTI) [22] for the KITTI data sets. Additionally, we limit the evaluation to the detection of collisions at the front of the vehicle, since the data has been recorded with a single, forward looking stereo camera.

## Results

The key quantitative results of the frame-based analysis on the simulated PSSC data set are shown in Figure 7. Subfigure (a) shows the results on all subsequences combined. The system performs best with an asteroid density of 75, a maximum stixel tracking length of 10 frames and a threshold on $p_{col}$ of 0.035, leading to an $F_1$ score of 0.79. Subfigures (b)-(e) present the results on the individual subsets. The sequence *Figure-8* is clearly the most difficult, indicating that the system under performs during non-straight ego motion. It is noteworthy that the tracking parameter has the most impact in that sequence, which may indicate the root of the problem in the current pipeline.

The corresponding results on the KITTI data set are shown in Figure 8. Due to the limited amount of positive samples, this graph just shows the false alarms as a function of the probability threshold. The system has a frame-based FAR of 0.45 on KITTI data at the optimal settings from the PSSC analysis.

The results of the event-based analysis are presented in Figure 9 and Table 2. The graphs in Figure 9 exhibit that the event-based analysis performs optimally for threshold on $p_{col}$ of around 0.05-0.15, which is higher than the optimal thresholds of the frame-based analysis of around 0.035. This indicates that the system can indeed cope with missing detections in some frames and still recognize the collision event as a whole.

The exact numbers of true, false and missed collision event detections on PSSC and KITTI data are shown in Table 2. First of all, the missed detections in PSSC data are all in the *Figure-8* sequence. The collision with a person laying down on the ground is almost always missed, indicating a low sensitivity to small obstacles. From Table 2, the optimal system settings are using an asteroid density of 10, tracking stixels for maximally 5 frames and thresholding with 0.095. We hypothesize that the tracking and prediction steps are currently not sufficiently robust to curved ego-vehicle motion. This could be resolved by employing a more advanced dynamic model for the ego-vehicle, which incorporates the steering angle.

Note that we have used PSSC data to find optimal settings and applied those to KITTI data. For completeness, we also analyzed the KITTI data itself. This analysis revealed that the optimal settings for KITTI and PSSC only differ in the value for the threshold. This setting is slightly higher for KITTI, which is appropriate due to the presence of more clutter in that data. However, both the optimal asteroid density and the optimal tracking length turned out to be identical, which illustrates that the system is robust to different environments with respect to those settings.

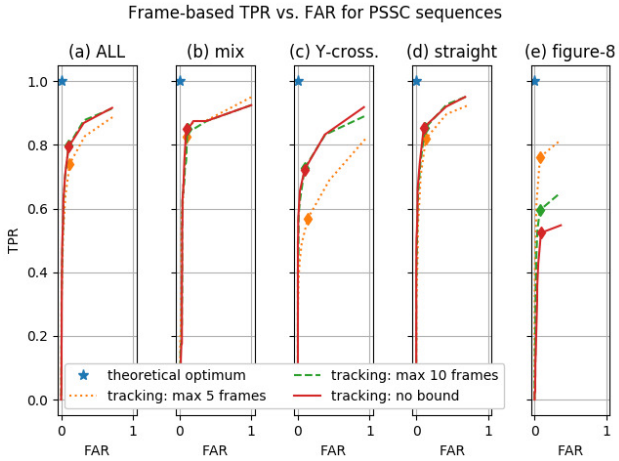



***Figure 7.*** *Frame-based quantitative results of complete set (a) and split over different PreScan sequences (b-e), showing that the system has trouble with non-straight ego-vehicle motion (e).*

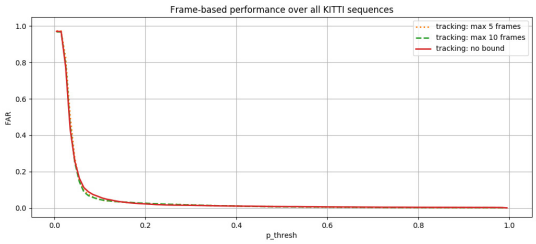



***Figure 8.*** *Frame-based quantitative results on all KITTI-tracking sequences combined, using the optimal settings from the PSSC analysis. Since KITTI has very few positive samples, only FAR is shown here.*

### *Timing*

Our three proposed processing blocks run on CPU and together take typically 10-20ms on a desktop PC (Intel Xeon CPU E5-1660 0 @3.30GHz12; 15.6GiB).

## Conclusion

This work has presented a stereo vision-based collision warning system that is suited for real-time execution in a car. The system is structured as a Bayesian histogram filter and provides fully probabilistic results, which can be supplied to other ADAS modules for data fusion or processed independently. We have also presented an example of the latter and evaluated it on both simulated and real-world data. At best, our independent system detects all 31 simulated collisions with 2 false warnings, while this setting

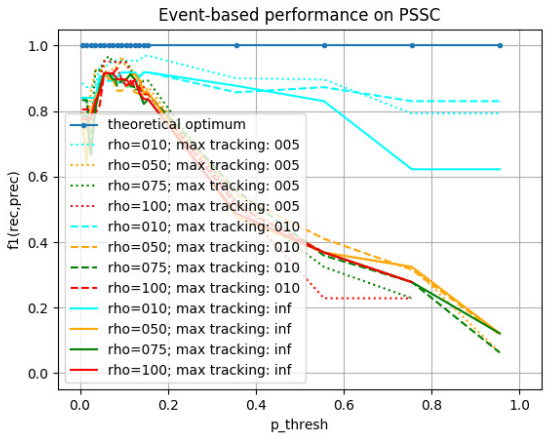



***Figure 9.*** *Event-based quantitative results on PSSC data.*

**Table 2. Event-based collision-warning results**

| Settings | | | PSSC data | | | KITTI data | | |
|---|---|---|---|---|---|---|---|---|
| Rho | Track | Thr | TP | FP | FN | TP | FP | FN |
| 10 | 5 | 0.095 | 31 | 2 | 0 | 1 | 12 | 1 |
| 50 | 5 | 0.055 | 29 | 0 | 2 | 2 | 20 | 0 |
| 75 | 5 | 0.045 | 30 | 1 | 1 | 2 | 35 | 0 |
| 100 | 5 | 0.055 | 30 | 2 | 1 | 2 | 93 | 0 |
| 10 | 10 | 0.085 | 28 | 2 | 3 | 1 | 8 | 1 |
| 50 | 10 | 0.055 | 27 | 1 | 4 | 2 | 50 | 0 |
| 75 | 10 | 0.055 | 27 | 1 | 4 | 2 | 56 | 0 |
| 100 | 10 | 0.055 | 27 | 0 | 4 | 2 | 36 | 0 |
| 10 | inf | 0.085 | 28 | 2 | 3 | 2 | 32 | 0 |
| 50 | inf | 0.055 | 27 | 1 | 4 | 2 | 56 | 0 |
| 75 | inf | 0.055 | 27 | 1 | 4 | 2 | 65 | 0 |
| 100 | inf | 0.055 | 27 | 1 | 4 | 2 | 72 | 0 |

generates 12 false warnings in the real-world data. This inspires future work to concentrate more on the performance of the stixel tracking and collision analysis modules, and provide an in-depth analysis on feasible object sizes and velocities.

## Acknowledgments

This work has been supported by the European Union Horizon 2020 project VI-DAS (grant agreement number 690772), by TASS International who facilitated using the PreScan software; and by B. Groenen, who explored the early stage of this project.

## References

[1] D. M. Gavrila, V. Philomin: Real-time object detection for "smart" vehicles, *IEEE ICCV*, 1999.

[2] R. Labayrade, D. Aubert, J.-P. Tarel: Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation, *IEEE IVS*, 2002.

[3] C. Thorpe, M. H. Hebert, T. Kanade, S. A. Shafer: Vision and navigation for the Carnegie-Mellon navlab, *IEEE T-PAMI*, 1988.

[4] E. D. Dickmanns, A. Zapp: A curvature-based scheme for improving road vehicle guidance by computer vision, *Mobile Robots I*, vol. 727, International Society for Optics and Photonics, 1987.

[5] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke *et al.* : Making Bertha drive an autonomous journey on a historic route, *IEEE ITS-M*, vol. 6, no. 2, 2014.

[6] M. Aeberhard, S. Rauch, M. Bahram, G. Tanzmeister, J. Thomas, Y. Pilat, F. Homm, W. Huber, N. Kaempchen: Experience, results and lessons learned from automated driving on Germany's highways, *IEEE ITS-M*, vol. 7, no. 1, pp. 42–57, 2015.

[7] H. Badino, U. Franke, D. Pfeiffer: The stixel world-a compact medium level representation of the 3d-world, Joint Pattern Recognition Symposium. Springer, Berlin, Heidelberg, 2009.

[8] W.P. Sanberg, G. Dubbelman, P.H.N. de With: Extending the stixel world with online self-supervised color modeling for road-versus-obstacle segmentation, *IEEE ITSC*, pp. 1400-1407, October 2014.

[9] L. Schneider, M. Cordts, T. Rehfeld, D. Pfeiffer, M. Enzweiler, U. Franke, M. Pollefeys, S. Roth: Semantic stixels: Depth is not enough, *IEEE IVS*, pp. 110-117, 2016.

[10] D. Pfeiffer, U. Franke: Efficient representation of traffic scenes by means of dynamic stixels, *IEEE IVS*, 2010.

[11] M. Enzweiler, M. Hummel, D. Pfeiffer, U. Franke: Efficient stixel-based object recognition. *IEEE IVS*, pp. 1066-1071, 2012.

[12] D. Pfeiffer, U. Franke: Towards a Global Optimal Multi-Layer Stixel Representation of Dense 3D Data *BMVC*, 2011.

[13] W.P. Sanberg, G. Dubbelman, P.H.N. de With: Color-based free-space segmentation using online disparity-supervised learning. *IEEE ITSC*, pp. 906–912, September 2015.

[14] W.P. Sanberg, G. Dubbelman, P.H.N. de With: Free-Space Detection with Self-Supervised and Online Trained Fully Convolutional Networks, *IS&T EI-AVM*, 2017.

[15] L.C. Liu, C.Y. Fang, S.W. Chen: A novel distance estimation method leading a forward collision avoidance assist system for vehicles on highways, *IEEE T-ITS*, vol. 18, no. 4, pp. 937–949, 2017.

[16] Mobileye: User Manual - Series 6 (DOC000600 REV A02 - ENG), available at `https://drive.google.com/file/d/0B99trsbO9s2eRlJiNjlwQ1hzUm8/view`, last visit: Oktober 2018.

[17] S. Cherng, C.Y. Fang, C.P. Chen, S.W. Chen: Critical motion detection of nearby moving vehicles in a vision-based driver-assistance system, *IEEE T-ITS*, vol. 10, no. 1, pp. 70–82, 2009.

[18] A. Ess, K. Schindler, B. Leibe and L. Van Gool: Object detection and tracking for autonomous navigation in dynamic environments, *IJRR*, vol. 29, no. 14, pp. 1707–1725, 2010.

[19] C.G. Keller, D.M. Gavrila: Will the Pedestrian Cross? A Study on Pedestrian Path Prediction, *IEEE T-ITS*, vol. 15, no. 2, April 2014

[20] H. Hirschmüller: Stereo processing by semiglobal matching and mutual information, *IEEE T-PAMI*, vol. 30, no. 2, pp. 328–41, 2008.

[21] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, T. Brox: A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation, *IEEE CVPR*, 2016.

[22] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, T. Brox: FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks, *IEEE CVPR*, 2017.

## Author Biography

*Willem Sanberg received both his BSc. and his MSc.-degree (2013) in Electrical Engineering and a certificate in Technology Management from Eindhoven Univ. of Technology in the Netherlands. 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 both governmental, industrial and research partners.*

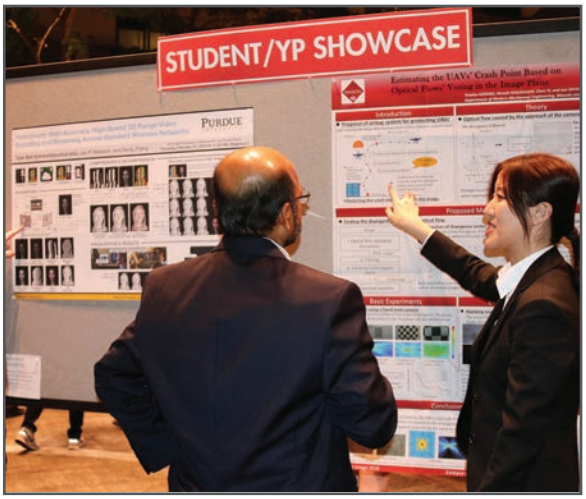
# JOIN US AT THE NEXT EI!

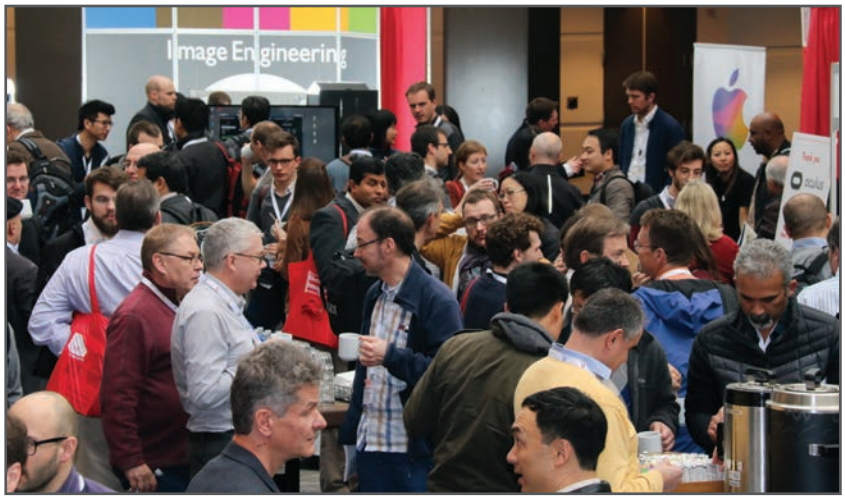
IS&T International Symposium on

# Electronic Imaging

SCIENCE AND TECHNOLOGY

*Imaging across applications . . . Where industry and academia meet!*





**• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •**
**• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •**

**www.electronicimaging.org**

IS&T
imaging.org