Beyond Thinking in Common Categories: Predicting Obstacle Vulnerability using Large Random Codebooks

Johannes Rühle, Erik Rodner, Joachim Denzler
Computer Vision Group, Friedrich Schiller University Jena
{johannes.ruehle, erik.rodner, joachim.denzler}@uni-jena.de
http://www.inf-cv.uni-jena.de

Abstract

Obstacle detection for advanced driver assistance systems has focused on building detectors for only a few number of object categories so far, such as pedestrians and cars. However, vulnerable obstacles of other categories are often dismissed, such as wheel-chairs and baby strollers. In our work, we try to tackle this limitation by presenting an approach which is able to predict the vulnerability of an arbitrary obstacle independently from its category. This allows for using models not specifically tuned for category recognition. To classify the vulnerability, we apply a generic category-free approach based on large random bag-of-visual-words representations (BoW), where we make use of both the intensity image as well as a given disparity map. In experimental results, we achieve a classification accuracy of over 80% for predicting one of four vulnerability levels for each of the 10000 obstacle hypotheses detected in a challenging dataset of real urban street scenes. Vulnerability prediction in general and our working algorithm in particular, pave the way to more advanced reasoning in autonomous driving, emergency route planning, as well as reducing the false-positive rate of obstacle warning systems.

1 Introduction

Vehicle vision systems are a key component of today's advanced driver assistance systems (ADAS) and especially automatic obstacle detection systems increase road safety and driver awareness.

In this paper, we contribute towards increased road safety by putting the objects in front of the vehicle into focus. We propose an approach that allows for a precise vulnerability classification of arbitrary obstacles that goes beyond using detectors built for a few specific categorical objects, like pedestrians and cars.

Our work builds upon the definition of vulnerability levels for arbitrary obstacles suggested in [13]. These classes predict the damage severity from the *obstacle's* perspective when assuming a collision with the driver's vehicle. Four classes express small, medium, heavy, and fatal collision consequences. The vulnerability levels form supersets of object categories and provide a more general vulnerability distinction of obstacles of the scenery ahead the vehicle, which is important to rely on in situations of accident prevention or mitigation. To further illustrate the significance and complexity of the problem, Fig. 1 shows such an emergency situation where the driver is required to take actions to prevent crashing with an obstacle ahead, *i.e.*, the person in the wheel chair. Knowing about the vulnerabilities leads to new evasion route planing with calculated risks for the obstacles and increased safety for very vulnerable ones. In case of Fig. 1, evading into the least

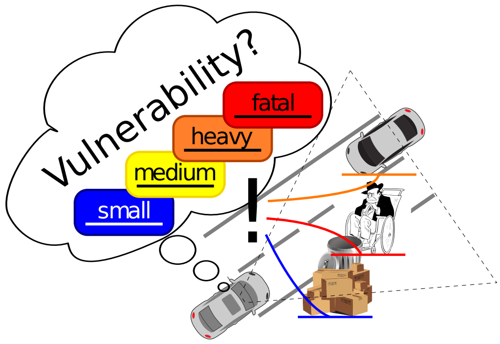

Figure 1: Evading in emergency situations in which breaking is too late: Knowing the vulnerability of obstacles ahead is important for the driver's evasion strategy to save the person in the wheel chair.

vulnerable boxes would save the person in the wheel chair without crashing into the oncoming traffic.

For vulnerability classification, we use large random codebooks of local descriptors built in a completely unsupervised manner from given gray scale images and disparity maps. We present an in-depth analysis of our approach in all its aspects and design choices. Comparing with the baseline classification of [13], we show that our enhanced approach leads to an improved vulnerability classification benefiting from using multi-scale features extracted from multi-cue data of gray scale images and disparity maps. The approach we present here can be applied to the results of any obstacle detection algorithm. We evaluate our algorithm on a very challenging real-world street scene dataset, which provides human-annotated vulnerability labels and includes scenarios where reasoning beyond the object-specific category spectrum is necessary. In a human experiment, we show that our problem of evaluating limited visual information to infer vulnerability is challenging even for human experts.

The paper is organized as follows: First, we review related work in Sect. 2 and explain the vulnerability classes in Sect. 3. Our vulnerability classification approach of obstacle hypotheses is described in Sect. 4. In Sect. 5, we state the experimental setup and discuss the obtained evaluation results. We conclude the paper in Sect. 6 with a summary.

2 Related Work

Our method is based on techniques used in image categorization, such as local features [2] and bag-of-visual-word models [4, 9]. In automotive applications, these methods have been applied, *e.g.*, for understanding street scenes by semantic pixelwise labeling [14]. Our application scenario is also related to object and obstacle detection. Vulnerable road users are commonly recognized as individual objects by specialized category-specific detectors for a limited number of ob-

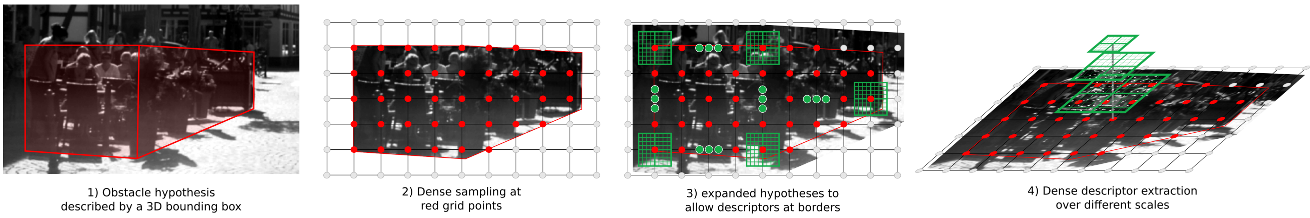

Figure 2: Visualization of an example obstacle hypothesis and the steps for densely extracting multi-scale descriptors within the hypothesis' expanded boundaries using the underlying image data.

jects, like *e.g.*, pedestrians [5], bicyclists [7], or vehicles [3]. The KITTI benchmark [8] provides an overview of current state-of-the-art object-specific detection algorithms. Unfortunately, we cannot use the extensive KITTI stereo vision dataset, since it only focuses on a few common object categories, provides no vulnerability label information, and completely neglects uncommon categories as well as obstacles of *small* vulnerability. In general, pre-defining important categories in automotive environments is likely to be limiting and unable to reflect the huge variety of obstacles that can be observed. It is simply impossible to build enough specific classifiers for each object that might appear in front of a vehicle. In that context, the vulnerability levels, first presented by us in [13], provide a more general measure, not bound to specific objects but visual patterns.

More generally, obstacle detection provides object-independent hypotheses of obstacles ahead a vehicle. We only point to a few related works in that area, since we do not perform obstacle detection, but apply our methods on hypotheses provided by the dataset we use. For example, [15] fuses 3D Lidar data and single camera images to generate obstacle hypotheses. In the related area of generic object detection [1, 16] predict bounding boxes of arbitrary objects in images independently of their object category. These methods can also be extended to videos and temporal consistent hypotheses [11] and used to generate input hypotheses for our algorithm, which then predicts the vulnerability level. Therefore, this line of research is complementary to ours.

3 Levels Of Vulnerability

Following the suggested vulnerability classes in [13], we distinguish between four discrete levels of vulnerability shown in Fig. 1 with their color coding. They express the expected severity of damage from the *object's* perspective when assuming a collision with the driver's vehicle. The highest, *fatal* vulnerability is assigned to vulnerable road users and indicates all human-related objects like, *e.g.*, pedestrians, a baby stroller, a bicycle with driver, or a wheel chair with a person. Furthermore, the vulnerability classes *heavy* (oncoming traffic) and *medium* (parked or ahead driving vehicles) distinguish vehicle obstacles evaluating how protected a driver is by its car and how severe accident consequences would be. All background obstacles like, *e.g.*, walls, trees, or poles, are considered as *small* vulnerability since they are not related to humans. In case of different vulnerable obstacles in a hypothesis, we prioritize the most vulnerable obstacle to set the vulnerability class for the entire hypothesis.

4 Vulnerability Prediction Of Obstacles

Our approach consists of two steps: obtaining obstacle hypotheses and predicting their vulnerability level. We briefly describe the step of obstacle hypotheses generation, but explain the vulnerability prediction in detail.

4.1 Obtaining obstacle hypotheses

Our vulnerability prediction method works on given bounding boxes as obstacle hypotheses for a single image. In general, any obstacle detection method could be used for this step (Sect. 2). In our case, we use hypotheses generated in real-time by a proprietary vehicle stereo camera [6] with build-in obstacle detection. Some hypotheses have the challenging properties that (1) obstacles located close together can result in a single large object hypothesis, and (2) a distinct real-world object might only be partially covered by a hypothesis.

The hypotheses we obtain are given as a 3D bounding box or a 3D plane segment (Fig. 2.1). After applying the stereo geometry and projecting the 3D hypotheses into the image coordinate system, we use their polygonal 2D shape to obtain the obstacles image areas for subsequent feature extraction steps (Fig. 2.2).

4.2 Vulnerability prediction with local statistics

Our approach focuses on building a classification pipeline without object category-specific models and, in contrast, tries to predict vulnerability levels directly. The image features for our application need to generalize visual properties for the training examples, so that also new obstacles can be described and assigned to one of the vulnerability classes.

The challenges described in Sect. 4.1 rule out features which assume a rigid constellation of the objects, such as HOG templates. The large variety of the appearances in each of the vulnerability levels demands for a fairly complex feature representation. To tackle both, we use a large bag-of-visual words representation [4] extracted in a completely unsupervised manner. The histograms created by the BoW pipeline reflect distributions of visual features inside a given obstacle hypothesis without assuming a rigid constellation and describe similarities to the training obstacles. Without spatial pooling the histograms can handle partial observations as well as important substructures in large hypotheses independently of their position in the hypothesis. After L_1 normalization, a BoW histogram is also invariant to the number of extracted descriptors in the hypothesis, which is important especially in our case with obstacles of varying distance to the car.

Dense local features We follow the recommendation of [9] and apply a dense grid feature extraction only at grid points inside the bounds of the obstacle hypothesis (Fig. 2.2). In particular, we use RootSIFT descriptors [2] and extract them from the gray scale image as well as the disparity map. The RootSIFT descriptor forms a local description of the underlying image patch by building histograms of image gradients. To describe obstacles independently from their distance to the camera and size in the image (scale

Table 1: Evaluation results for different BoW approaches on single- and multi-cue data. Also shown are the results for a random classification based on the dataset's class distribution and the results of a human expert classification.

Algorithm	**ACC**	**ARR**
Random Guessing	43.9	25.0
Human Expert	*91.7*	*83.6*
*Single cue: image **or** disparity*		
image cue (presented in [13])	79.4	57.9
disparity cue	79.5	65.6
*Multi cue: image **and** disparity*		
Fusion of SIFT descriptors	**83.6**	**69.1**
Fusion of BoW histograms	83.5	67.1

invariance), we extract the descriptors using multiple scales. Applying S different scales (*i.e.* cell sizes) s_i ($1 \leq i \leq S$) during dense descriptor extraction, we obtain S descriptors at each grid point (Fig. 2.4) Given the cell sizes s_i, we expand a given hypothesis by $2 \cdot \max_{1 \leq i \leq S}(s_i)$ pixels (Fig. 2.3). So, we obtain more descriptors close to the original border incorporating information about the obstacle's shape properties and background separation.

Large random codebooks All descriptors extracted from all training object hypotheses are used to build a codebook of N visual words describing reoccurring and striking visual features. We show that using large codebooks built by randomly selecting $N > 10000$ descriptors as cluster centers [4] is indeed beneficial for our application, due to the vast variety of the input data. In our experiments, we also compare this approach with the common k-Means clustering scheme. Histograms are computed for all obstacle hypotheses by finding the best matching codebook entries for each RootSIFT descriptor (hard quantization). During this histogram generation all spatial information inside a hypothesis is discarded on purpose as described above. We further transform the $L1$ normalized BoW histograms again by performing elementwise square-rooting, to benefit from the non-linearity of the Hellinger kernel [12] and being able to still use a linear SVM for classification.

5 Experimental Evaluation

5.1 Dataset

For the evaluation, we use the labeled dataset we presented in [13] which was recorded by a proprietary stereo camera mounted behind the front windshield. It consists of 17 authentic real-world street sequences with 2428 gray scale images and disparity maps (1024×460px), as well as over 9950 obstacle hypotheses obtained with the camera's built-in obstacle detection system. All hypotheses contain ground-truth vulnerability labels and show much more *small* vulnerable obstacles than *fatal* ones.

5.2 Evaluation results

We quantitatively evaluate our algorithms with the accuracy (ACC) and the average recognition rate (ARR) to account for an unbalanced class setup. We perform 17 leave-one-out evaluation splits using 16 sequences for training and the remaining one for testing. We run 5 trials per split to account for the random subset selection during codebook creation.

Human and random performance To understand the difficulty of the task we are confronted with,

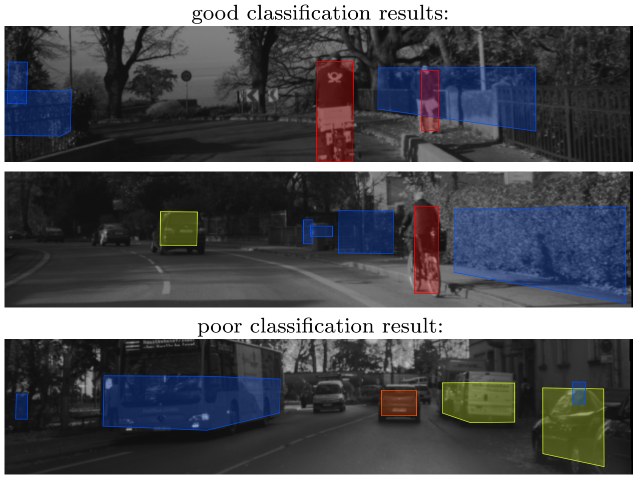

Figure 3: Qualitative overview of some good and poor results of classifying vulnerability classes using our proposed approach.[1] Best viewed in color. Color coding according to Figure 1.

we state the upper and lower bound of the classification performance in Tab. 1. Randomly assigning a vulnerability class based on the dataset's class distribution gives a lower bound of 43.9% ACC and 25.0% ARR (*Random Guessing*). For the upper bound, a *human expert* classified the vulnerability of a subset of the hypotheses by hand. Seeing only cropped image regions, the expert achieved only 91.7% ACC and 83.6% ARR.

Advantages of large random codebooks We choose codebook sizes N between 100 and 200000 entries computed from RootSIFT descriptors on the image and disparity map cues with a fixed number of 4 different scales. Empirically, we found that large random codebooks with $N^* = 25000$ codebook entries provide the best result of 83.0% ACC and 66.5% ARR. k-Means clustering leads to a comparable performance but requires a higher computational time unsuitable for large codebook sizes.

Evaluation of multiple scales In our experiments, we used sets of five to ten scales with increasing cell sizes ranging between 4 and 34 pixels. Using $N^* = 25000$ from the previous experiment, we obtained the best overall classification performance of 83.6% ACC and 69.1% ARR when using $S^* = 9$ different scales. Fig. 3 gives an impression of some results of our algorithm using this setup.[1] The ROC curves in Fig. 4 (right) indicate that especially classifying *fatal* vulnerable hypotheses profits from more scales.

Combination of intensity and disparity Furthermore, we evaluate the impact of using the combination of image and disparity data and compare two strategies for fusion of the multi-cue features:

1. Fusion of RootSIFT descriptors: Concatenating the descriptors extracted from image and disparity data at the same position and scale leads to a more complex local descriptor of dimension 256.
2. Fusion of BoW histograms: Two independent BoW histograms for each cue are concatenated.

Based on the previous experimental results, we use $N^* = 25000$ and $S^* = 9$. The results of the comparison are shown in Tab. 1. They show that the fusion at the descriptor level outperforms the second approach. Comparing to our single cue baseline of

[1] More evaluation results and images can be found on our website: http://www.cv-inf.uni-jena.de/vulnerability

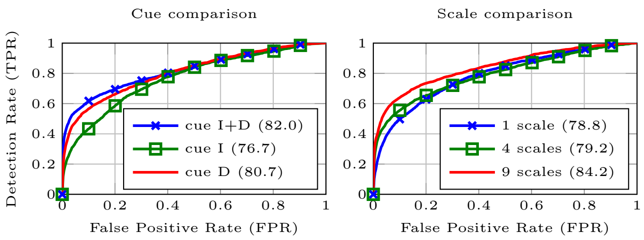

Figure 4: ROC curves for the classification performance of the BoW pipeline for the *fatal* class: (left) using RootSIFT features of different cues, gray scale images (cue I, [13]), disparity maps (cue D), and both gray scale image and disparity maps (cue I+D); (right) evaluation with different number of scales. The individual AUC in percent is shown in parentheses.

Table 2: Comparison of our algorithm with a proprietary pedestrian detector for the task of detecting vulnerable obstacles of the *fatal* class.

Algorithm	ACC	ARR
Our approach	**95.0**	**66.8**
Pedestrian detector	93.2	65.3

[13] that only uses gray scale images and small codebooks, our presented multi-cue and multi-scale approaches achieve a notable higher classification performance. Fig. 4 (left) shows the receiver operating characteristic curves (ROC curves) for the cue comparison for the *fatal* class. It can be seen that using both the image and disparity map cue lead to a gain of 6.9% in area under the curve (AUC).

In preliminary experiments, we also tried other local descriptors such as histograms of oriented gradients (HOG, [7]) and local binary patterns (LBP, [10]), but both resulted in a lower overall performance.

Predicting fatal vulnerability is more than pedestrian detection To show that vulnerability prediction goes beyond classical pedestrian detection, we also compared with a proprietary state-of-the-art pedestrian detector. Since this detector can only distinguish between *fatal* (pedestrian) and all other vulnerability levels (no pedestrian), we also reduced the outputs of our algorithm to this binary setting. The results presented in Tab. 2 show that our approach outperforms the pedestrian detector, demonstrating that a rigid detector alone is not enough for detecting vulnerable obstacles.

Computational Runtime The largest part of the computational runtime is covered by the multi-scale and multi-cue feature extraction (90%), highly depending on S. The histogram generation only takes about 7% and the classification 3% (linear SVM). The highest speed-up can be achieved by reducing the number of extracted scales S decreasing the feature extraction runtime. On an Intel i7-4770 with 3.4 GHz and implemented in MATLAB (single threaded) our multi-cue algorithm runs in between 76ms ($S = 1$, $N = 1000$) and 969ms ($S^* = 9$, $N^* = 25000$) per image.

6 Conclusion

In this paper, we present an approach to classify the vulnerability of arbitrary obstacle hypotheses independently of object category models, such as pedestrian or car detectors. The main goal is to directly predict the vulnerability of the obstacle from the appearance without an intermediate layer restricted to human categorization. In our approach, we use a bag-of-visual-words approach with a large random codebook and we evaluated our approach on several challenging street scenes with ground-truth vulnerability labels. We were able to show that this challenging problem is feasible with current state-of-the-art techniques and a combination of disparity and intensity information. Furthermore, we outperform a proprietary pedestrian detector by classifying vulnerable obstacles that do not naturally fall into one of the typical object categories. From an application point of view, we show that the functionality of an existing stereo vision camera that is already being deployed in real-world cars with a basic obstacle detection can easily be extended towards vulnerability prediction and advanced reasoning about the obstacle.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *PAMI*, 34(11):2189–2202, 2012.

[2] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, pages 2911–2918. IEEE, 2012.

[3] J. Arrospide, L. Salgado, and J. Marinas. Hog-like gradient-based descriptor for visual vehicle detection. In *IV*, pages 223 –228, 2012.

[4] A. Coates and A. Y. Ng. The importance of encoding versus training with sparse coding and vector quantization. In *ICML*, pages 921–928, 2011.

[5] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 34(99):743–761, 2012.

[6] A. Eckert, A. Hohm, and S. Lueke. Integrated ADAS Solution for Pedestrian Collision Avoidance. *Intl Conf on Enhanced Safety of Vehicles*, (13-0298-O), 2013.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010.

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *IJRR*, 32(11):1231–1237, 2013.

[9] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *ECCV*, pages 490–503. Springer, 2006.

[10] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24(7):971–987, 2002.

[11] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *ECCV*, pages 737–752. Springer, 2014.

[12] F. Perronnin, J. Sánchez, and Y. Liu. Large-scale image categorization with explicit data embedding. In *CVPR*, pages 2297–2304. IEEE, 2010.

[13] J. Rühle, M. Arbitmann, and J. Denzler. Vulnerability classification of generic object hypotheses using a visual words approach. In *Proc of the FISITA 2014 World Automotive Congress*, 2014. F2014-AST-045.

[14] T. Scharwächter, M. Enzweiler, U. Franke, and S. Roth. Efficient multi-cue scene segmentation. In *GCPR*, pages 435–445, 2013.

[15] P. Shinzato, D. Wolf, and C. Stiller. Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion. In *IV*, pages 687–692, 2014.

[16] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.