Rendered Benchmark Data Set for Evaluation of Occlusion-Handling Strategies of a Parts-Based Car Detector

Marvin Struwe[1](✉), Stephan Hasler[2], and Ute Bauer-Wersing[1]

[1] Frankfurt University of Applied Sciences, Frankfurt, Germany
{mstruwe,ubauer}@fb2.fra-uas.de

[2] Honda Research Institute Europe GmbH, Offenbach, Germany
stephan.hasler@honda-ri.de

Abstract. Despite extensive efforts, state-of-the-art detection approaches show a strong degradation of performance with increasing level of occlusion. A fundamental problem for the development and analysis of occlusion-handling strategies is that occlusion information can not be labeled accurately enough in real world video streams. In this paper we present a rendered car detection benchmark with controlled levels of occlusion and use it to extensively evaluate a visibility-based existing occlusion-handling strategy for a parts-based detection approach. Thereby we determine the limitations and the optimal parameter settings of this framework. Based on these findings we later propose an improved strategy which is especially helpful for strongly occluded views.

Keywords: Object detection · Benchmark data set · Occlusion-handling

1 Introduction

Perception of traffic participants is a fundamental component in driver assistant systems. Despite extensive research visual detection of objects in natural scenes is still not robustly solved. The reason for this is the large appearance variation in which objects or classes occur. A very challenging variation is occlusion which is caused by the constellation of objects in a scene. Occlusion reduces the number of visible features of an object but also causes accidental features. Current object representations show acceptable results during a low to medium level of occlusion but fail for stronger occlusions. Methods like [1,11] train a holistic object template in a discriminative manner and focus resources on differences between classes. This strong specialization on the training problem results in a stronger decrease of performance for occluded objects when trained on unoccluded views. In contrast to this parts-based methods like [7,8] accumulate local features in a voting manner. Also when trained with unoccluded views, these methods can handle arbitrary occlusion patterns, but require that sufficiently many features can still be detected. However, in general the voting methods

T. Bräunl et al. (Eds.): PSIVT 2015, LNCS 9431, pp. 99–110, 2016.
DOI: 10.1007/978-3-319-29451-3_9

perform worse than the discriminative ones, whenever test and training set do not show such systematic differences, as discussed in [15] and confirmed by the detection results in [2]. In this paper we want to extensively evaluate visibility-based occlusion-handling strategies for a parts-based detection approach using a rendered benchmark data set.

Some methods make use of context to explicitly deal with occlusion information, i.e. to exploit knowledge about the possible constellation of objects. The two approaches in [4,14] use Markov-Random-Fields to infer if neighboring features are consistent with a single detected instance of an object or have to be assigned to different ones. This allows both approaches to reason about relative depth of objects and to produce a coarse segmentation. However, the process over the whole input image leads to a time consuming iteration. Besides instance-instance relations, also knowledge about general occlusion patterns can be used. In [13] the authors handle vertical occlusion generated by the image border, which means that they have some knowledge about the occlusion constellation. This idea can be extended to the whole image if information about the occluding object is provided.

Occlusion is related to the 3D relation of objects. A general cue of 3D information is depth which can be used to check the physical plausibility of an object's position and size [6] or to segment and put attention to individual scene elements [12]. In [12] temporal differences between RGB-D(epth) views are used to discover movable parts for action representation.

Other strategies make use of 3D annotated data of car views. A common strategy for occlusion-handling is the use of the deformable part model (DPM) [3]. In [10] the 3D annotated data of the KITTI data set [5] is used to generate bounding boxes of the occluder, the occluding object, and their union and for each of the three types a separate DPM is trained. In [16] the authors used hand-annotated 3D CAD models and generated part models additionally to the full car view. A single component DPM detector is trained for each part configuration. To handle occlusion 288 occluder masks are generated for the training data. The approach works not in real time and can handle only occlusion cases which match somehow with the generated occlusion masks.

A parts-based detection approach with explicit occlusion-handling is shown in [9]. For the occlusion-handling depth information is used to determine the visibility of a car hypothesis. This information is used for a re-weighting of the score.

In general most approaches with explicit occlusion-handling make use of information of the occluding object delivered from different methods or sensors. We also want to integrate mask information of the occluding object to reason about visibility of features and to re-weight the activation score of a possible car hypothesis. The occluding mask can be provided in a real system by 3D or depth information. We want to show the limitations and the optimal parameter setting for parts-based detection approaches which provides a mask of the occluding object.

In Sect. 2 we describe how we generated a rendered benchmark data set. Section 3 outlines the used parts-based detection framework. Finally, in Sect. 4

we extensively evaluate a visibility-based occlusion-handling strategy for a parts-based detection approach, before we propose an improved strategy. Finally we are drawing the conclusions and further work in Sect. 5.

2 Rendered Benchmark Data Set

Benchmark data sets like KITTI show precise bounding boxes for objects but do not provide pixel-level information of the constellation of the occlusion, which is quite important to evaluate different occlusion strategies. To get more accurate labeling and a better control of the scene conditions we used a render framework to generate a data set. With the render framework we can define the position of the car, the position of the light source, the intensity of the light, and the angle of rotation of the car. Additionally we can estimate which pixels of the rendered image belongs to which object and can store this information in a mask which is more precise than a bounding box. After rendering we pasted the car models in the center of a car free street scene to get realistic clutter in the background. For our data set we defined the ranges of variation for the parameters for the scene conditions. At the rendering randomly values for the different parameters are used inside the defined ranges. An overview of the different car models with the changing scene conditions can be seen in Fig. 1.

The size of each rendered segment is set to 175 × 70 which correlates to the optimal size for our car models plus a border. We normalized the size of each car model in a way that the side view covers a given width of the segment. In general parts-based detection approaches are trained to a limited view of rotation of an object. For full rotation several detectors have to be trained. Since we want to concentrate on the evaluation of occlusion-handling strategies we only used side views of cars in a range of 30 degrees of variation at a total side view and omit the rotation handling. We randomly shifted the center position of the car for a better generalization after training. Because we want to evaluate occlusion strategies we use a fixed size for each car. In the upper line of Fig. 2 some segments with cluttered background for the training can be seen. In the bottom line the corresponding masks are shown.

For our data set we used for each car model 400 views. Segments of 44 car models each with 400 views are used for the training set while the other 44 car models each with 400 views are used for the test set. The training set includes non-occluded car views with a segment size of 175 × 70 pixels.

We generated different test sets with different rates of occlusion. The set with 0 percent of occlusion only shows a car object in the center of a car free street scene. These images were then used with an occluding object to generate test sets with 20, 40, 60, and 80 percent of occlusion. To get a car like shape for the occluding object we used an ellipse shaped patch. Instead a black colored occluding object which is unnatural since it includes no features we cropped patches out of car free street scenes. Figure 3 shows an example for a generated test image and the corresponding masks. For a consistence evaluation of the investigated occlusion-handling approach we also generated occluding constellations without any car. For the training set we generated in total 17600 segments

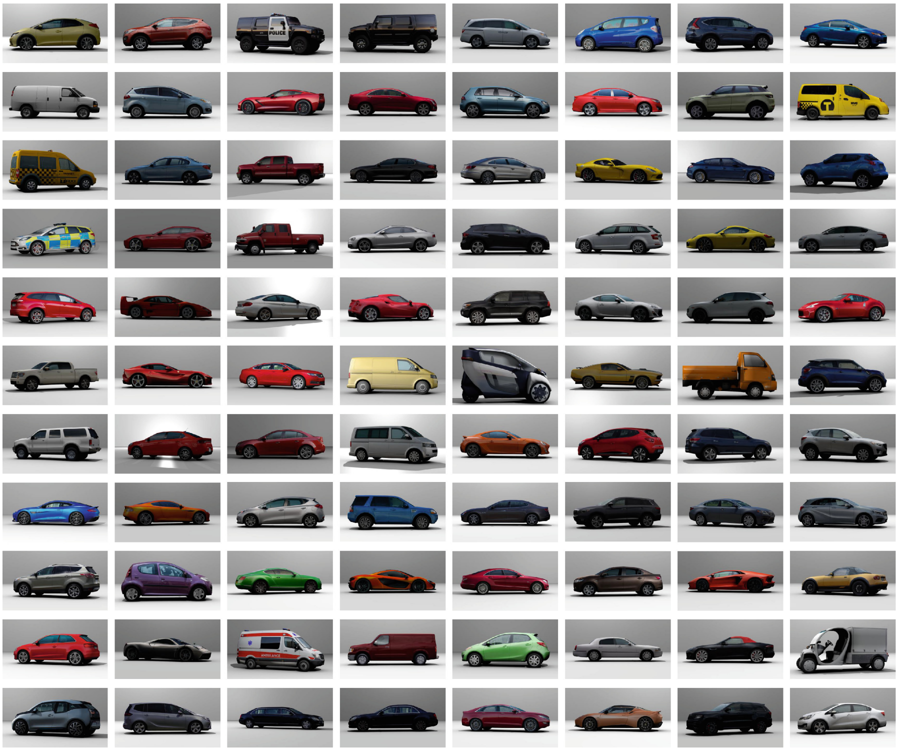

Fig. 1. Data set with different scene conditions: A data set with randomly chosen conditions for light position, light intensity, and angle of rotation is generated.

of un-occluded car views and for the test set 105600 images with different rates of occlusion.

In the next section we will use this data set to train and test our parts-based car detector.

3 Parts-Based Car Detector

Parts-based approaches detect the occurrence of part features over the image where each feature can vote casts for the object center. At the end a confidence map determines plausible detections. In this section we describe which descriptors are used for the features. We also show how the visual feature codebook is build and later used for detection.

3.1 Extraction of Texture Descriptors

Voting methods like [7] use a key point detector to localize interest points in the input image. At these points some texture descriptors are used, e.g. SIFT [8].

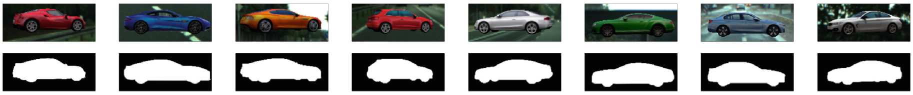

Fig. 2. Segments used for training: The upper line shows car segments with a cluttered background while the bottom line shows the corresponding masks.

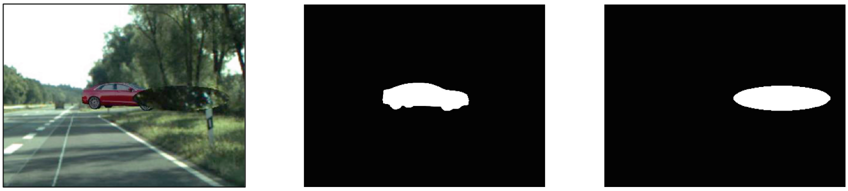

Fig. 3. Example of the test set: At the left a test image with an occluding ellipse is shown. The two other images show the mask for the background object without any occlusion and the mask of the occluding object, respectively.

We noticed that these points often result only at high textured areas. To handle this drawback we decided to use a dense grid over the image for the extraction of texture descriptors. We tested several distances for the gap between grid points. A gap of 5 pixels shows the best compromise between the number of keypoints and the computing time. After some evaluation we decided to use SIFT descriptor without scaling what means that the receptive field size of 16 pixels correlates with the size of the patches used in the segment. To avoid the extraction of untextured descriptors we use an edge detector with the same size of the receptive field as the descriptor on each grid point. Only patches with corners or edges provide enough texture information to get specific features. We simply check if the resulting patch of the edge detector shows a minimum percentage of marked pixels. Patches that do not achieve this criterion are unselected. With those we can see a threefold reduction of the numbers of features (Fig. 4).

3.2 Learning of Parts-Based Object Representation

The object representation of a parts-based detector is stored in a visual codebook. The codebook includes the features and the relative position to the object's center. One method to build the codebook is to use a clustering method. The resulting clusters are the features of the codebook. Approaches that are trained on natural data use the whole training segment which leads to extraction of descriptors also at the background. These accidental non-car features have to be filtered out which is a challenging task by itself. By using our rendered data we can use the mask information of the object to limit the extraction on the object that should be learned. For training we used the training segments without occlusion. To build the codebook a MiniBatchKMeans clustering is performed.

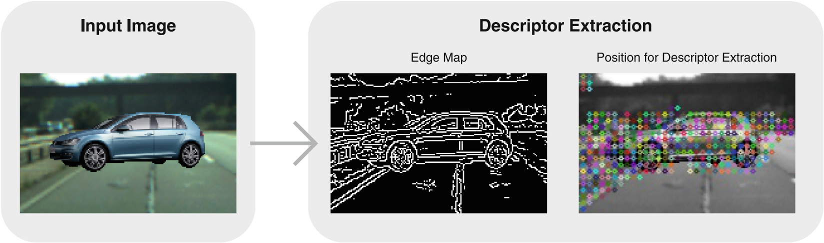

Fig. 4. Extraction of Texture Descriptors: The extraction layer shows the edge map and the resulting position for the descriptor extraction.

In contrast to the normal KMeans this clustering method splits the data into chunks to save memory and computing time.

Figure 5 shows two features of the visual codebook. On the left the so called feature maps are shown which describe the distribution of the occurrence of each stimuli of the codebook cluster. The sum of each feature map is normalized to one. On the right some corresponding stimuli of the descriptor clusters are shown. A well-balanced amount of clusters have to be used to get feature maps that are not too specific but also not too general. Too specific maps show only activations at small compact areas and represent non-generalized training results. Too general maps show a broad distribution of the stimuli and a non-specific training result. In our case 220 clusters provided a good compromise between both criteria. In the next section we show how the codebook is used for detection.

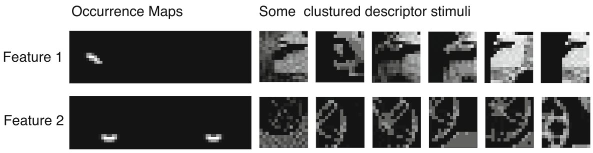

Fig. 5. Example of codebook features: On the left the so called occurrence maps are shown which describe the distribution of the occurrence of each stimuli of the codebook cluster. On the right some corresponding stimuli of the descriptor cluster are shown.

3.3 Parts-Based Detection Framework

In this section we will show a parts-based framework for car detection. First we extract descriptors for the full test image like described in Sect. 3.1. For each descriptor we determine the best matching feature. This feature votes with its occurrence map for the objects' center. An accumulation of all votes of all features at the test image is used to build the activation map. We refer to the accumulated score at each single position with γ (Fig. 6).

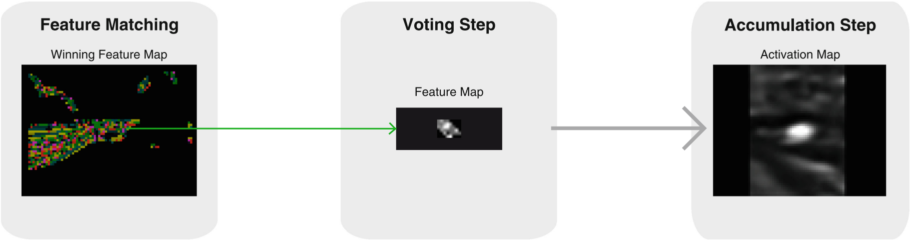

Fig. 6. Detection framework: The winning feature map shows the best matching feature for each pixel. Each feature votes with its occurrence map for the objects' center. An accumulation of all features is used to build the activation map.

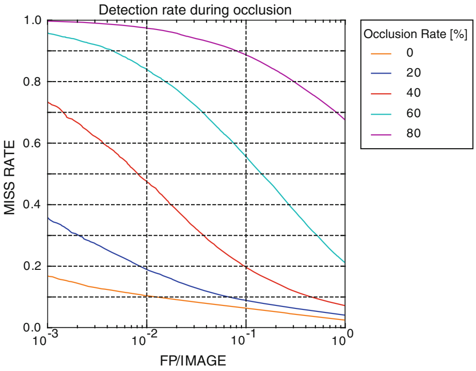

Fig. 7. Detection Performance: The ROC plot shows the detection result of our visibility-based parts-based car detector for five rates of occlusion.

Figure 7 shows the detection performance of this parts-based car detector for 0, 20, 40, 60, and 80 percent of occlusion. A loss in detection performance with an increase of occlusion rate can be seen. Car hypothesis that shows a minimal overlap of 80 percent in the height and 60 percent in the width of the ground truth are marked as detected car.

4 Occlusion-Handling of the Parts-Based Car Detector

As described in Sect. 1 parts-based methods like [9] use an explicit occlusion-handling to re-weight the accumulated score by making use of the predicted visibility of the car hypothesis. Motivated by this we will show a visibility-based occlusion-handling strategy for our parts-based detection framework. Like described before the parts-based approaches use an accumulative step to calculate the score for the car hypothesis. Occlusion will reduce this score. A visibility-based occlusion-handling strategy is to predict the occlusion of an object by

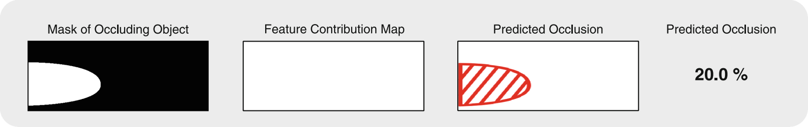

Fig. 8. Uniform contribution for β calculation: On the left the mask of the occluding object for a support window can be seen. The feature contribution map shows a uniform contribution which results in a predicted occlusion rate of 20 percent.

using the mask of an occluding object and to re-weight the score γ by taking the predicted occlusion β into account. For this we use a so called support window. The support window covers the field of the features that potentially contribute to the hypothesis. The support window has the same size like our feature maps. To calculate β a uniform distribution of each pixel of the support window is supposed (Fig. 8).

The predicted occlusion of the support window is used to reject detected features inside the occluding area before accumulating the score. We refer to the score without occlusion-handling with γ'. The final score γ'' will be calculated with the predicted rate of occlusion β for the supporting window (Fig. 9 with (A) for the re-weighting), i.e.

$$\gamma'' = \gamma'/(1 - \beta) \tag{1}$$

If β is 0 than the re-weighting has no effect. But by increasing β at the evaluation the re-weighting will increase the score while the support of γ' is getting lower. This can generate false positives at high values of β. Our goal is to find the maximum value for β that improves the detection performance. So we need a limitation of β up to a defined maximum occlusion rate for the use of the occlusion-handling. To find the optimal value for this limit β_{max} we evaluate the detection performance of our car detector by using values from 0.1 to 0.9 in steps of 0.1 for β_{max}. To see also effect of β_{max} at different rates of occlusion we use occlusion rates of 20, 40, 60, and 80 percent at the evaluation. The detection results can be seen in Fig. 10. For 20 percent occlusion a β_{max} of 0.2 shows the best results while for 80 percent occlusion a β_{max} of 0.5 yields the best results. In general the best detection performance can be seen if β_{max} is equal or in most cases lower than the predicted occlusion rate.

However, the results show that for each occlusion rate another β_{max} has to be used. For a detection system only a fixed β_{max} can be used. So we need an optimal β_{max} for all occlusion rates. To find such a value we plotted the result for the full data set including all occlusion rates for each β_{max} of an occlusion rate. Figure 11(left) shows the best improvements at all occlusion rates by using a β_{max} of 0.2. Figure 11(right) shows an improved detection performance for all occlusion rates by using the determined optimal value of 0.2 for β_{max}. The most gain can be seen at 40 and 60 percent of occlusion.

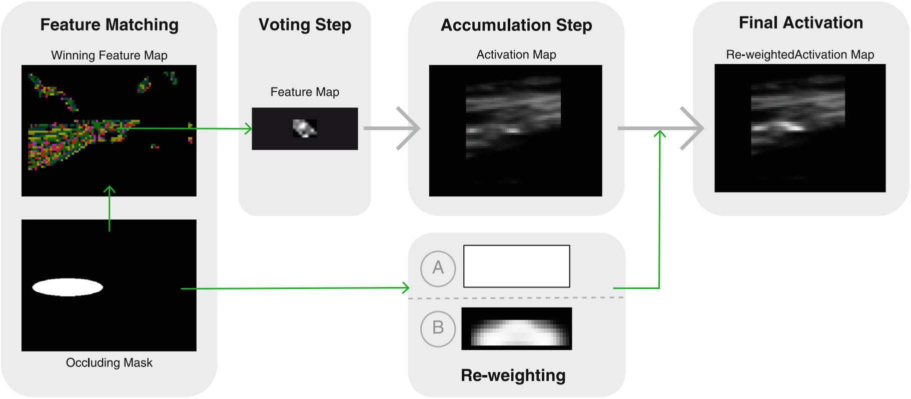

Fig. 9. Detection framework with occlusion-handling strategy: The mask of the occluding object is used to calculate the activation score by deselect the winning features at the occluding area. For the uniform occlusion-handling the re-weighting (A) is used to generate the final score. For the contribution-aware occlusion-handling the re-weighting (B) is used and will be explained in Sect. 4.1.

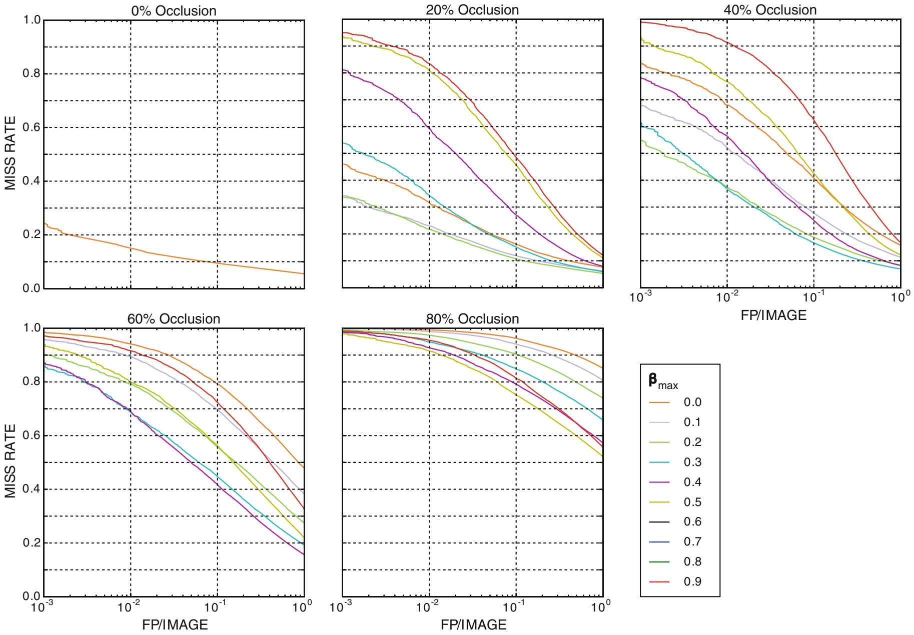

Fig. 10. Evaluation of the Detection Performance: The plots show the detection result for 0, 20, 40, 60, and 80 percent of occlusion separately.

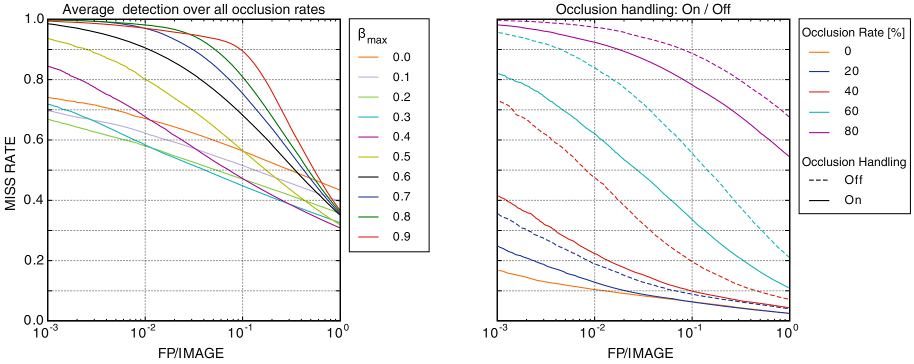

Fig. 11. Optimal parameter setting for occlusion-handling: (Left) Detection result for the full data set including all occlusion rates for each β_{max}. (Right) shows the detection performance for different occlusion rates by using the determined optimal value of 0.2 for β_{max}. The dotted lines show the detection result by using the parts-based car detector without occlusion-handling while the solid lines belong to the detector with uniform occlusion-handling.

4.1 Contribution-Aware Strategy for Occlusion-Handling of a Parts-Based Car Detector

A uniform distribution of the parts of a car can not be expected by using a parts-based detection approach. Therefore we developed a way to account for the contribution to the accumulative score of the features. To get a more realistic estimation of the missing score we want to use a so called feature contribution map. For this we used the activation maps of each training segment and add all maps together in the feature contribution map. This map is used for calculating γ' and β. Now γ' shows a more realistic score after the rejection of the feature of the occluding area (Fig. 9 with (B) for the re-weighting). Also β presents now a more realistic occlusion rate of the car hypothesis (Fig. 12).

Like in Sect. 4 we plotted the detection result for the full data set including all occlusion rates for each β_{max} of an occlusion rate. In Fig. 13(left) the use of 0.4

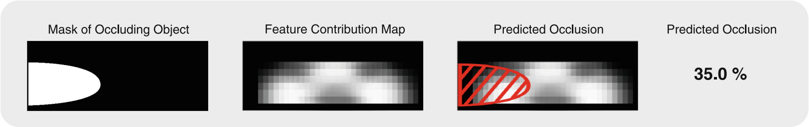

Fig. 12. Contribution-aware calculation for β calculation: On the left the mask of the occluding object for a support window can be seen. The feature contribution map shows a realistic contribution what result in a predicted occlusion rate of 35 percent by covering 20 percent of the area of the supporting window.

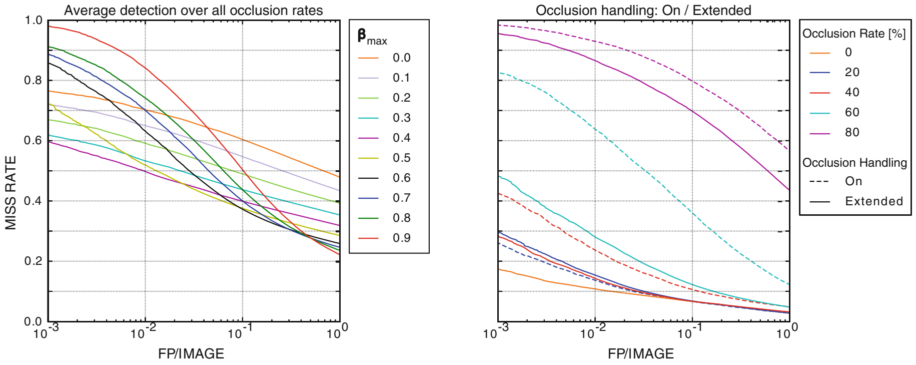

Fig. 13. Optimal parameter setting for contribution-aware occlusion-handling: (Left) Detection result for the full data set including all occlusion rates for each β_{max}. (Right) shows the detection results by using the determined optimal value of 0.4 for β_{max}. The dotted lines show the detection performance using the uniform occlusion-handling while the solid lines results from the contribution-aware occlusion-handling.

for β_{max} shows the best result. We used this β_{max} of 0.4 for the car detector with the contribution-aware occlusion-handling and show the detection performance in Fig. 13(right). The plot shows an improved detection performance at occlusion rates of 40, 60, and 80 percent while a small lost can be seen for 20 percent of occlusion.

5 Conclusion and Further Work

In this paper we introduced a rendered benchmark data set with controlled levels of occlusion and scene conditions. We determined the limitations and the optimal parameter settings of a parts-based car detector by extensively evaluating a visibility-based occlusion-handling strategy. The result of this evaluation is used to configure the car detector, which shows an improved detection performance at all occlusion rates. We also showed an improved strategy for occlusion-handling which boosts the detection performance specially for higher occlusion rates. The proposed occlusion-handling strategies are applicable to other detection approaches that include an accumulation step, like e.g. [3,10]. The results in Fig. 13 show that the detection performance can not be improved for 80 percent of occlusion as much as for 40 and 60 rates of occlusion. A very challenging effect is the number of false positives that are generated at very high levels of occlusion if too strong re-weighting is applied. This limitation of the system can not be solved by the used input information. More scene understanding and information is necessary to have a kind of possibility check to reduce the number of said false positives and to boost the detection performance.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the CVPR, pp. 886–893 (2005)
2. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: a benchmark. In: Proceedings of the CVPR, pp. 304–311 (2009)
3. Felzenszwalb, P.F., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. In: Proceedings of the PAMI, pp. 1627–1645 (2010)
4. Gao, T., Packer, B., Koller, D.: A segmentation-aware object detection model with occlusion handling. In: Proceedings of the CVPR, pp. 1361–1368 (2011)
5. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: Proceedings of the CVPR, pp. 3354–3361 (2012)
6. Gould, S., Baumstarck, P., Quigley, M., Ng, A.Y., Koller, D.: Integrating visual and range data for robotic object detection. In: ECCV Workshop M2SFA2 (2008)
7. Leibe, B., Schiele, B.: Interleaved object categorization and segmentation. In: Proceedings of the BMVC, pp. 759–768 (2003)
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2), 91–110 (2004)
9. Makris, A., Perrollaz, M., Laugier, C.: Probabilistic integration of intensity and depth information for part-based vehicle detection. IEEE Trans. Intell. Transp. Syst. **14**, 1896–1906 (2013)
10. Pepik, B., Stark, M., Gehler, P., Schiele, B.: Occlusion patterns for object class detection. In: Proceedings of the CVPR, pp. 3286–3293 (2013)
11. Struwe, M., Hasler, S., Bauer-Wersing, U.: Using the analytic feature framework for the detection of occluded objects. In: Mladenov, V., Koprinkova-Hristova, P., Palm, G., Villa, A.E.P., Appollini, B., Kasabov, N. (eds.) ICANN 2013. LNCS, vol. 8131, pp. 603–610. Springer, Heidelberg (2013)
12. Stückler, J., Behnke, S.: Hierarchical object discovery and dense modelling from motion cues in RGB-D video. In: Proceedings of the IJCAI, pp. 2502–2509 (2013)
13. Vedaldi, A., Zisserman, A.: Structured output regression for detection with partial truncation. In: Proceedings of the NIPS, pp. 1928–1936 (2009)
14. Winn, J., Shotton, J.D.J.: The layout consistent random field for recognizing and segmenting partially occluded objects. In: Proceedings of the CVPR, pp. 37–44 (2006)
15. Yi-Hsin, L., Tz-Huan, H., Tsai, A., Wen-Kai, L., Jui-Yang, T., Yung-Yu, C.: Pedestrian detection in images by integrating heterogeneous detectors. In: IEEE Computer Symposium (ICS), pp. 252–257 (2010)
16. Zia, M.Z., Stark, M., Schindler, K.: Explicit occlusion modeling for 3D object class representations. In: Proceedings of the CVPR, pp. 3326–3333 (2013)