Scene Understanding From a Moving Camera for Object Detection and Free Space Estimation

Vladimir Haltakov, Heidrun Belzner and Slobodan Ilic

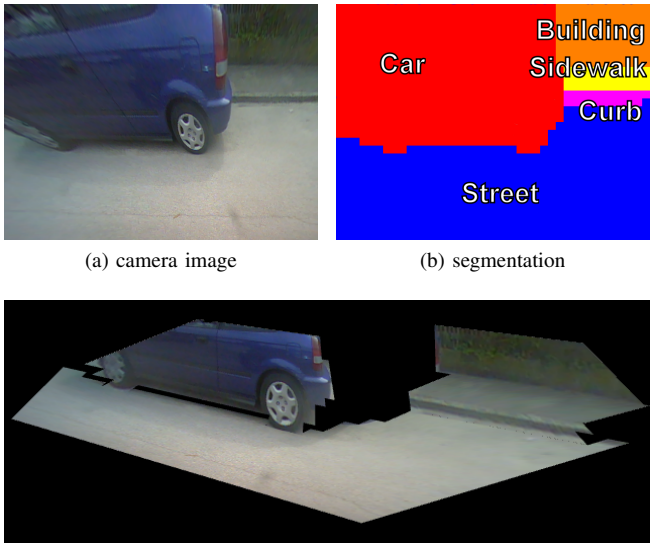
***Abstract*— Modern vehicles are equipped with multiple cameras which are already used in various practical applications. Advanced driver assistance systems (ADAS) are of particular interest because of the safety and comfort features they offer to the driver. Camera based scene understanding is an important scientific problem that has to be addressed in order to provide the information needed for camera based driver assistance systems. While frontal cameras are widely used, there are applications where cameras observing lateral space can deliver better results. Fish eye cameras mounted in the side mirrors are particularly interesting, because they can observe a big area on the side of the vehicle and can be used for several applications for which the traditional front facing cameras are not suitable.**

We present a general method for scene understanding using 3D reconstruction of the environment around the vehicle. It is based on pixel-wise image labeling using a conditional random field (CRF). Our method is able to create a simple 3D model of the scene and also to provide semantic labels of the different objects and areas in the image, like for example cars, sidewalks, and buildings.

We demonstrate how our method can be used for two applications that are of high importance for various driver assistance systems - car detection and free space estimation. We show that our system is able to perform in real time for speeds of up to 63 km/h.

I. INTRODUCTION

Intelligent camera systems are becoming increasingly important for a number of advanced driver assistance systems (ADAS) in modern vehicles. Today almost all cars on the market can be equipped with lane departure warning assistants, collision detection assistants and other safety or comfort systems that rely on video input from multiple cameras. Most of those systems are based on a front facing camera, mounted behind the windshield. Such cameras have a good overview over the area in front of the car and they are also usable at night because of the active lighting provided by the head lights. Furthermore, the package allows for mounting two cameras creating a stereo pair, which further improves the quality of the systems. Therefore, front facing cameras are the main point of research related to driver assistance systems. However, this type of cameras does also have several significant disadvantages. They cannot observe the environment around the vehicle, which is important for some systems like parking assistants for example. While human drivers are always able to turn their head while driving in order to examine the lateral space, this cannot be achieved with a fixed front facing camera and for those cases sideways mounted cameras are needed. There are also cases when lane detection that is based only on the front camera may fail, because the road markings are hidden by another vehicle driving closely in front (as often happens in traffic jams). Therefore, cameras looking on the side of the car are becoming increasingly important as a source of information for various driver assistance systems.



(a) camera image (b) segmentation

(c) textured 3d model

Fig. 1: Example result of our reconstruction algorithm. (a) the calibrated input camera image, (b) the segmentation of the image, (c) the reconstructed 3D model with textures

This work presents a general approach of scene understanding based on a fish eye camera mounted on the side of a vehicle. The method is based on segmenting the image in regions with different semantic meaning, like for example the street, cars and buildings. Those regions are then used to reconstruct a simple 3D pop-up model of the environment. The reconstruction can be performed only from the segmentation of a single image, in contrast to motion stereo approaches, where at least two overlapping images are needed. The first part of our work is mostly similar to [1], which shows state-of-the-art results in image segmentation and is demonstrated using a front facing camera on a moving vehicle. However, because of the differences in the viewpoints of the front facing camera and the sideways facing fish eye camera, we show that several optimizations can be performed in order to significantly speed up the segmentation. The second part

This work is supported by the BMW Group.

V. Haltakov and H. Belzner are with BMW Group, Munich, Germany Firstname.Lastname@bmw.de

S. Ilic is with the Technische Universität München, Garching b. München, Germany Slobodan.Ilic@in.tum.de

of our method is concerned with the construction of a 3D model of the scene by resolving the projective ambiguity in the camera image, using the segmented image and several geometric assumptions that we can derive from our knowledge about the environment. An example is shown on Fig. 1.

We evaluate our image segmentation algorithm against a dataset of manually labeled images. We also show how our method can be used for car detection and free space estimation on the side of the vehicle. We evaluate our results by dividing the space on the right side of the car in intervals of a fixed width and we classify them as a free part of the street or as occupied by a car.

II. RELATED WORK

There are several different approaches for scene understanding with a monocular camera mounted on a vehicle. A lot of methods are focused on creating a 3D map of the world with motion stereo approaches [**?**], [**?**], [2]. Recent works show good quality of the reconstruction and real-time performance [**?**], [**?**], but most of the systems do not give information about the types of the objects in the scene.

Some more application specific works focus on explicit reasoning about the street layout of the scene and the detection of certain object classes in it [**?**], [**?**].

One way to acquire more semantic information about the scene is to use an object detector, which is able to find object of a certain category in the image, like for example cars or people [4], [**?**], [**?**]. Most of these methods return a bounding box around the detected object, while some are also able to estimate its 3D position and orientation [6], [7], [10]. Unfortunately, they cannot be used for amorphous objects that do not have a well defined shape such as streets and buildings and therefore cannot provide a complete description of the whole image. Such detectors are already built-in in most driver assistance systems available on the market.

Another way for acquiring information for all parts of the image is to use segmentation methods that divide the image into regions belonging to different semantic classes. Especially interesting approach is the semantic pixel-wise labeling, which classifies each pixel of the image as belonging to a certain class, like for example to a car, street or a pedestrian. Due to recent advances in the fields of discrete optimization and probabilistic graphical models, conditional random fields (CRFs) have become the standard way for solving such segmentation problems [1], [8], [9], [**?**].

Especially interesting are works that perform labeling and segmentation on images from realistic outdoor scenarios. In [1] the authors combine a static labeling approach with a shape based car detector and a dynamic model, which improves the recognition of vehicles. However, the presented approach requires the extraction of image features at multiple scales, while in our proposed method only one scale is required even though similar features are used. The TextonBoost approach proposed in [8] uses significantly more complex texton features, but shows results similar to [1]. The work presented in [11] uses a pixel-wise segmentation as a basis for the scene interpretation, but they do not perform explicit higher level reasoning. Instead, additional classifiers are trained in order to detect only the presence, but not the exact location of certain objects in the scene.

Notable are also methods for creating rough 3D models from a single frame. These methods try to infer the general layout of the scene in order to resolve the projective ambiguity in the image [**?**], [**?**], [**?**]. Those methods, however, rely mostly on unsupervised segmentation algorithms, which are not very robust.

III. METHOD

The method that we propose consists of two separate steps. The first one is to segment the image into regions corresponding to different semantic classes and the second step considers the reconstruction of a 3D model of the environment from multiple frames, given the segmented images and the odometry data from the car.

A. Image segmentation

In order to segment the camera image into regions, belonging to different objects and areas, we employ an image labeling method. The image is divided into small cells according to a regular grid. Each image is of size 640 x 480 pixels and we choose the size of the grid to be 40 x 30 in order to get cells of size 16 x 16 pixels. Each cell is then assigned a class depending on its appearance and the classes of the surrounding cells, like for example car, street, building, sidewalk, etc. In this way we get an implicit segmentation of the image - grouping neighboring cells of the same class gives us one distinct segment.

1) The model: We introduce one random variable y_i for each cell in the grid which encodes the class assigned to it. The values of the pixels of each cell are indicated by another variable x_i and the set of possible classes is denoted by L. We model the probability of a certain assignment of all cells $\mathbf{y}$, given the whole image $\mathbf{x}$ as a pairwise conditional random field over the regular grid. According to the Hammersley-Clifford theorem the posterior conditional distribution can be written as:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_i \psi_u(y_i, x_i, \lambda) \prod_{(i,j)} \psi_p(y_i, y_j, \mathbf{x}, \theta) \quad (1)$$

The first product in the model equation iterates over all cells i and the second product iterates over all pairwise neighboring cells (i, j) in a 4-connected neighborhood. The unary and pairwise potential functions are denoted with ψ_u and ψ_p respectively and λ and θ are the model parameter sets. Because the potential functions are not required to output values between 0 and 1, a separate normalization factor is needed in order to formulate the model as probability distribution. This normalization factor is represented by the partition function $Z(\mathbf{x})$:

$$Z(\mathbf{x}) = \sum_{\mathbf{y}'} \prod_{i} \psi_u(y'_i, x_i, \lambda) \prod_{(i,j)} \psi_p(y'_i, y'_j, \mathbf{x}, \theta) \quad (2)$$

where $\mathbf{y}'$ iterates over all possible assignments of the classes to the cells. The optimal label assignment of the camera image can then be retrieved by choosing the labeling $\mathbf{y}^*$ with the highest probability.

2) Features: The pixel values of the cell are not directly used by the classifier, but a 50 element feature vector $f(x_i)$ is extracted from them. For the features we use the 2D Walsh-Hadamard transform, which is a discrete approximation of the cosine transform, which can be computed very efficiently. We compute the coefficients of the transform in the dyadic order as described in [12]. The first 16 coefficients are extracted from each of the tree channels of the camera image in the *Lab* color space and stacked together in order to build the feature vector. Additionally the grid coordinates of the cell are appended to the vector in order to include location context. In the case of a front facing camera as in [1], one has to perform the Walsh-Hadamard transform at several different scales in order to account for the big variations in the appearance of the objects. However, in our case, because of the viewpoint of the side camera, such big differences are not present and we can save a significant amount of time by using only one scale.

3) Potential Functions: The potential functions indicate how likely it is that a given cell has a certain class. Since the potential functions have to be non-negative it is very convenient to represent them in a negative exponential form:

$$\psi_u(y_i, x_i, \lambda) = e^{-\phi_u(y_i, x_i, \lambda)} \quad (3)$$

$$\psi_p(y_i, y_j, \mathbf{x}, \theta) = e^{-\phi_p(y_i, y_j, \mathbf{x}, \theta)} \quad (4)$$

For the unary potential functions one AdaBoost binary classifier is trained for each class and we denote the confidence of the classifier that the cell x_i has the class c as $B_c(x_i)$. A linear combination of the confidences of all classifiers is used in order to determine the value of the unary potential functions:

$$\phi_u(y_i, x_i, \lambda) = \lambda_0^{y_i} + \sum_{c \in L} \lambda_c^{y_i} B_c(x_i) \quad (5)$$

where $\lambda_c^{y_i}$ are the parameters of the linear combination for each class. The classifiers are trained and the parameters $\lambda_c^{y_i}$ are estimated in an offline learning phase.

The pairwise potentials encourage neighboring cells to have the same label, unless there is a big difference in the appearance of the two cells:

$$\phi_p(y_i, y_j, \mathbf{x}, \theta) = \theta_{y_i, y_j} |f(x_i) - f(x_j)| \quad (6)$$

Here again the parameters θ_{y_i, y_j} are estimated from training data. For the difference in the appearance of the cells we use the property of the Walsh-Hadamard transform that similar images have small difference in the coefficients and therefore we can take the length of the difference vector as an indication of how different the appearance of the cells is.

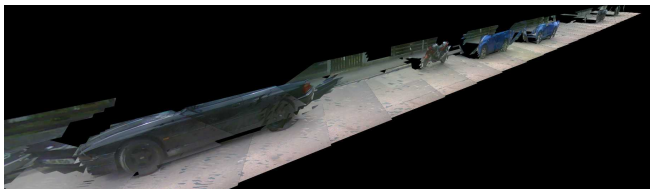


Fig. 2: 3D reconstruction of the street consisting of multiple frames.

4) Parameter estimation: Before the model can be used to label new images, the parameter sets λ and θ should be estimated. We prepared a set of 360 manually labeled images, of which 80% were used as a training dataset in order to estimate the parameters and 20% for validation. An exact maximum likelihood estimation requires computation of the normalization function $Z(\mathbf{x})$ for each image, which should iterate over 7^{1200} different assignments for a 40 x 30 grid and 7 classes. Since such computation is clearly intractable, we adopted a piecewise learning approach as described in [13] in order to approximate $Z(\mathbf{x})$ locally.

5) Inference: The problem of finding the optimal labels for a new camera image does not require the computation of the normalization function, but is highly non-linear. Taking the negative logarithm of the main model equation, allows us to reformulate the inference problem as a free energy minimization problem:

$$E(\mathbf{y}) = \sum_{i} \phi_u(y_i, x_i, \lambda) + \sum_{(i,j)} \phi_p(y_i, y_j, \mathbf{x}, \theta) \quad (7)$$

In this case a big variety of discrete optimization methods can be used in order to find an approximate solution. We used the efficient α-expansion algorithm [14], which is specifically designed for minimizing energy functions from pairwise graphical models with multiple classes.

B. 3D environment reconstruction

In order to be able to reason about distances in the real world, we must make the transition from the 2D image into the 3D world. Generally such transition from a single image is impossible, because of the ambiguity in the perspective projection.

We propose a method, which can reconstruct a relatively simple model only from a single image, given its segmentation in semantic regions. In order to resolve the projective ambiguity, we make several assumptions about the layout of the scene which represent certain geometric constraints like for example that cars should be standing on the street and not floating in the air. We also assume that regions of a certain class always lie in the same plane and that this plane is either parallel to the ground plane or perpendicular to the ground plane and parallel to the direction of travel. For example we assume that cars are always perpendicular and the sidewalk is always parallel to the street. This assumption

TABLE I: The geometric assumptions about the environment incorporated into the model

Class	Plane orientation	Possible underlying classes
STREET	horizontal	-
CAR	vertical	STREET
CURB	vertical	STREET
SIDEWALK	horizontal	STREET, CURB, GRASS
BUILDING	vertical	STREET, CURB, SIDEWALK, GRASS
GRASS	horizontal	STREET, CURB, SIDEWALK
OBJECT	vertical	STREET, CURB, SIDEWALK, GRASS

may seem very restrictive, because not all parts of the side of the car lie in the same plane, but it is relatively well suited for measuring sizes of the objects and the distances between them. Table I gives a complete description of the used geometric assumptions.

In order to reconstruct a 3D model of the scene we choose the origin of the world coordinate system to be on the ground plane exactly under the camera with the z axis pointing upwards, the x axis pointing in the direction of travel of the vehicle and the y axis perpendicular of the traveling direction pointing away from the car. We calibrated the extrinsic parameters of the camera with a pattern on the ground in order to obtain the position of the camera with respect to the street. In this way we can derive the equation of the ground plane. From the camera calibration data we can compute the camera projection matrix $\mathbf{P}$ and then for each 2D image point $\mathbf{x}$ we can compute the ray on which the real 3D world point $\mathbf{X}$ should lie in order to be projected on $\mathbf{x}$. From the image segmentation we know which points belong to the ground and from the intersection of the ground plane and the projection ray we can compute the exact location of the 3D point $\mathbf{X}$:

$$\mathbf{X} = \mathbf{P}^{\dagger}\mathbf{x} - \frac{\pi^{\mathbf{T}}\mathbf{P}^{\dagger}\mathbf{x}}{\pi^{\mathbf{T}}\mathbf{C}}\mathbf{C} \tag{8}$$

where $\mathbf{C}$ is the camera center, π is the ground plane vector in homogeneous coordinates and $\mathbf{P}^{\dagger}$ is the Moore-Penrose pseudoinverse, which has the property $\mathbf{P}^{\dagger}\mathbf{P} = \mathbf{I}$.

After all points on the ground plane have been reconstructed, we proceed with the objects standing on the ground. For example if we have a car region above the street region, we assume that the car touches the street in the lowest left image point belonging to the car region. This point belongs to both the car and the street and therefore we know its exact 3D position. Together with the assumption that cars lie in a vertical plane parallel to the x axis, we can compute the equation of this plane. After we have the plane equation, we can use the same method as for the ground in order to find the 3D positions of all points belonging to the car. Regions of other classes are processed analogously, where regions inconsistent with our geometric assumptions are removed.

In order to acquire a 3D model of a whole sequence of images, we take an image every 2 meters and we register the images to one another using the odometry data from the car, assuming a motion only along the x axis (see Fig. 2). We should note that the straight motion assumption is not a principle limitation of our approach, but only requires more complex vehicle motion model.

IV. RESULTS

We demonstrate and evaluate our method in two separate steps - image segmentation and 3D reconstruction. The quality of the segmentation influences the quality of the free space estimation and car detection in most cases, but the relationship is not always directly visible as we will show.

A. Image segmentation

In order to evaluate the performance of the image segmentation algorithm we prepared a dataset of 360 calibrated camera images of resolution of 640 x 480 pixels, taken with a moving car at speeds of up to 60 km/h (see Fig. 3 for examples). The dataset was split into a training dataset of 288 images and validation dataset of 72 images. All of them were manually labeled in order to provide ground truth data. We used 7 classes in the experiments: STREET, CAR, CURB, SIDEWALK, BUILDING, GRASS and OBJECT. The image was divided in a regular grid 40 x 30 cells corresponding to a cell of size 16 x 16 pixels.

After the offline training of the model parameters with the training dataset, we evaluate the number of correctly classified cells over all images in the validation dataset, achieving a precision of 85.5%. The resulting confusion matrix can be seen in Table II - here the rows indicate the true class of the cell and the columns show the class predicted by our model. While the classes STREET, CAR, BUILDING and GRASS have relatively good detection rates, the CURB, SIDEWALK and OBJECT classes do not perform so well because of several reasons. The errors for the CURB class are mostly due to the relatively small size of the curb in comparison to the cell size of 16 pixels. Because of this almost all cells containing a piece of the curb do also contain a piece of the street or the sidewalk. The confusion matrix shows that indeed in most of the cases the curb is mistaken either for street or for sidewalk. The misclassified instances of the SIDEWALK class are mainly due to the very similar appearance of the street and sidewalk surfaces, which can also be seen in the confusion matrix. The class OBJECT consists of all objects that cannot be assigned to any other class and in general has very few instances in the dataset.

The segmentation of one image takes 80 ms on average on an Intel Core 2 Duo, 2.4 GHz, 4 GB RAM machine without using SSE instructions optimization.

B. Car detection and free space estimation

In order to demonstrate how our method can be used in real life scenarios, we developed an example application for car detection and free space estimation on the side of the vehicle. In order to evaluate our method, we discretize the area on the right side of the vehicle in intervals of 0.5 m along the x axis. In this way we can formulate the task as a classification problem, where each interval has to be assigned one of the 3 possible labels: CAR, FREE SPACE or OCCUPIED SPACE. A CAR is an interval, where more than

TABLE II: Confusion matrix of the image segmentation algorithm

	STREET	CAR	CURB	SIDEWALK	BUILDING	GRASS	OBJECT
STREET	**94.58**%	2.10%	0.12%	3.07%	0.13%	0.00%	0.00%
CAR	1.26%	**91.21**%	0.32%	1.79%	5.31%	0.01%	0.10%
CURB	35.80%	11.26%	**34.83**%	12.56%	4.75%	0.71%	0.08%
SIDEWALK	25.60%	8.58%	0.29%	**59.88**%	5.49%	0.15%	0.00%
BUILDING	0.44%	10.07%	0.26%	7.62%	**81.10**%	0.09%	0.42%
GRASS	0.09%	0.09%	0.09%	0.47%	14.19%	**85.06**%	0.00%
OBJECT	0.88%	3.27%	0.76%	7.30%	35.52%	0.00%	**52.27**%

50% are occupied by a vehicle, a FREE SPACE is a part of the street where cars are free to drive and OCCUPIED SPACE is a part of the infrastructure around the street, where driving is not allowed as for example the sidewalk, grass areas or buildings. This problem representation gives us a good way to evaluate the method's performance and to analyze the relationship between the errors in the image segmentation algorithm and the street layout classification algorithm.

We conducted tests by driving a car around the city streets for about 1 km at speed of up to 60 km/h. The streets for the test were of several different layout types: small one way streets with cars parked on the side, streets with no cars parked on the side, multiple lane streets while driving in the right most lane or the middle lane. Because the reconstruction model assumes motion only along the x axis (but it does not assume a constant speed) we evaluated only the results were the car is driving straight with a turning angle of no more than 3 degrees. The classified intervals were then manually checked and the correct and wrong classifications were counted. The overall achieved detection rate is 87.3% and the corresponding confusion matrix is shown in Table III. As we can see from the confusion matrix, cars were recognized correctly in almost all cases (99.59%) with false positives almost only at intervals, where the car starts or ends. These errors are mainly due to the geometric assumptions that we make. From the viewpoint of the sideways mounted camera, almost only the side of the car is observed, which is lying basically in the same plane. The assumption is violated only at the corners of the car, where small parts of the front or the backside are also visible, resulting in a reconstruction that is slightly wider than the actual object. An example of the street layout classification algorithm can be seen on Fig. 4.

In this application most of the errors in the image segmentation do not have a direct influence of the layout classification - for example if a GRASS area is mistaken for a BUILDING, it is still considered as an occupied area. The error that influences the application the most is when a part of the SIDEWALK is wrongly classified as STREET (25.6% in the image segmentation). One can see the influence of this error also in the confusion matrix of the layout classification - 21.23% of the OCCUPIED intervals are recognized as FREE.

The time needed for the creation of the 3D model and the classification of the intervals is on average 35 ms per frame. Including the time for the image segmentation the method requires 115 ms per frame. We should also note, that the proposed application requires only one image on every 2 meters to be processed, which means that the method is able to process all images in real time for driving speeds of up to 63 km/h which is enough in most cases in city scenarios.

TABLE III: Confusion matrix of the street layout classification algorithm

	CAR	FREE	OCCUPIED
CAR	**99.59**%	0.00%	0.41%
FREE SPACE	7.05%	**87.67**%	5.28%
OCCUPIED SPACE	8.94%	21.23%	**69.83**%

V. CONCLUSION

We presented a two stage method for creating a simple 3D model of the area on the side of a moving vehicle with a sideways mounted fish eye camera. The model also contains semantic information about the type of the objects in the scene like cars, street, buildings and others. We showed how this generic method can be used for car detection and free space estimation. The method is able to perform in real time while driving with a speed of up to 63 km/h and to detect cars and free spaces with a detection rate of 87.3%.

ACKNOWLEDGMENTS

We would like to thank the BMW Group for supporting this work.

REFERENCES

[1] C. Wojek and B. Schiele, "A dynamic conditional random field model for joint labeling of object and scene classes," in *ECCV*, 2008, pp. 733–747.

[2] W. van der Mark and D. M. Gavrila, "Real-time dense stereo for intelligent vehicles," *IEEE Trans. on Intell. Transport. Syst.*, vol. 7, no. 1, pp. 38–50, 2006.

[3] G. Zhang, J. Jia, T.-T. Wong, H. Bao, and H. Bao, "Consistent depth maps recovery from a video sequence." in *PAMI*, vol. 31, no. 6, 2009, pp. 974–988.

[4] J. K. Suhr, H. G. Jung, K. Bae, and J. Kim, "Automatic free parking space detection by using motion stereo-based 3d reconstruction," *Machine Vision and Applications*, vol. 21, 2010.

[5] C. Unger, E. Wahl, and S. Ilic, "Parking assistance using dense motion-stereo," *Machine Vision and Applications*, pp. 1–21, 2011, 10.1007/s00138-011-0385-1.

[6] K.-T. Song and H.-Y. Chen, "Lateral driving assistance using optical flow and scene analysis," in *IEEE Intelligent Vehicles Symposium*, 2007, pp. 624–629.

[7] A. Geiger, C. Wojek, and R. Urtasun, "Joint 3d estimation of objects and scene layout," in *NIPS*, Granada, Spain, December 2011.

[8] A. Geiger, M. Lauer, and R. Urtasun, "A generative model for 3d urban scene understanding from movable platforms," in *CVPR*, Colorado Springs, USA, June 2011.

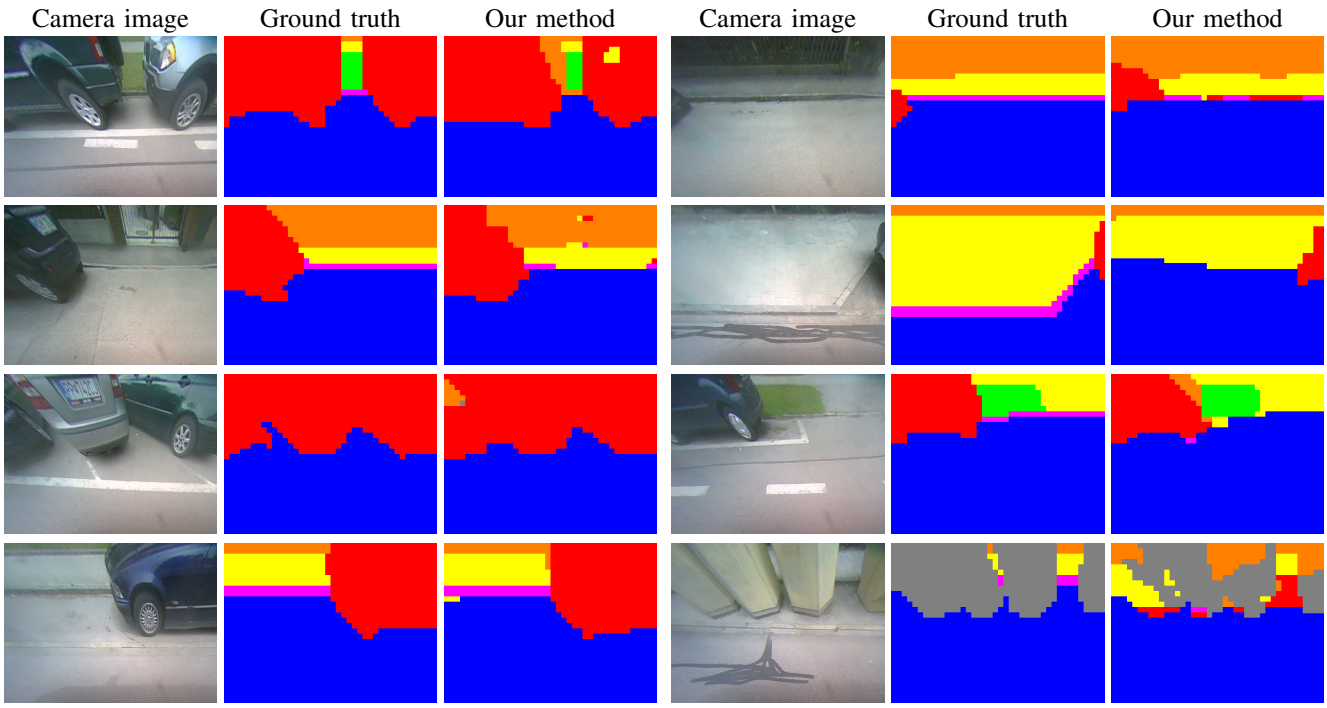
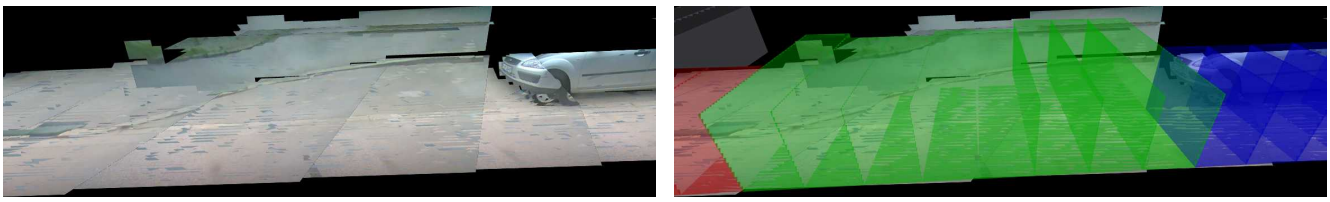


Fig. 3: Example results of the image segmentation algorithm. The assignment of the classes is coded with colors: STREET - blue, CAR - red, CURB - purple, SIDEWALK - yellow, BUILDING - orange, GRASS - green, OBJECT - gray. The image is best viewed in color.



(a) camera image (b) segmentation

Fig. 4: Example of the street layout classification algorithm. The left image shows the 3D model obtained by our reconstruction method and the right image shows the classification of the intervals, where the colors red, green and blue indicate the classes CAR, FREE SPACE and OCCUPIED SPACE respectively. The image is best viewed in color.

[9] P. Viola and M. Jones, "Robust real-time object detection," in *Second International Workshop On Statistical And Computational Theories Of Vision - Modeling, Learning, Computing, And Sampling*, 2001.

[10] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "Moving obstacle detection in highly dynamic scenes," in *ICRA*, ser. ICRA'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 4451–4458.

[11] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *PAMI*, vol. 32, 2010.

[12] M. Stark, M. Goesele, and B. Schiele, "Back to the future: Learning shape models from 3D CAD data," in *BMVC*, Aberystwyth, Wales, 2010.

[13] J. Liebelt and C. Schmid, "Multi-view object class detection with a 3d geometric model," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2010, pp. 1688–1695.

[14] C. Wojek, S. Roth, K. Schindler, and B. Schiele, "Monocular 3d scene modeling and inference: Understanding multi-object traffic scenes," in *ECCV*, 2010.

[15] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," IJCV, 2007.

[16] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *CVPR*, 2008.

[17] X. He, R. Zemel, and M. Carreira-Perpin, "Multiscale conditional random fields for image labeling," in *CVPR*, 2004.

[18] A. Ess, T. Mueller, H. Grabner, and L. J. V. Gool, "Segmentation-based urban traffic scene understanding," in *BMVC*, 2009.

[19] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3-D Scene Structure from a Single Still Image," *PAMI*, 2008.

[20] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering surface layout from an image," *IJCV*, vol. 75, pp. 151–172, October 2007.

[21] A. Gupta, A. A. Efros, and M. Hebert, "Blocks world revisited: Image understanding using qualitative geometry and mechanics," in *ECCV*, 2010.

[22] Y. Hel-Or and H. Hel-Or, "Real time pattern matching using projection kernels," *ICCV*, vol. 2, p. 1486, 2003.

[23] C. Sutton and A. McCallum, "Piecewise training for undirected models," in *Proceedings of the Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*. Arlington, Virginia: AUAI Press, 2005, pp. 568–575.

[24] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *PAMI*, vol. 23, pp. 1222–1239, November 2001.