Detecting parallel-moving objects in the monocular case employing CNN depth maps

Nolang Fanani[1], Matthias Ochs[1], and Rudolf Mester[1,2]

[1] Visual Sensorics & Information Processing Lab, Goethe University Frankfurt, Germany

[2] Computer Vision Laboratory, ISY, Linköping University, Sweden

Abstract. This paper presents a method for detecting independently moving objects (IMOs) from a monocular camera mounted on a moving car. We use an existing state of the art monocular sparse visual odometry/SLAM framework, and specifically attack the notorious problem of identifying those IMOs which move parallel to the ego-car motion, that is, in an 'epipolar-conformant' way. IMO candidate patches are obtained from an existing CNN-based car instance detector. While crossing IMOs can be identified as such by epipolar consistency checks, IMOs that move parallel to the camera motion are much harder to detect as their epipolar conformity allows to misinterpret them as static objects in a wrong distance. We employ a CNN to provide an appearance-based depth estimate, and the ambiguity problem can be solved through depth verification. The obtained motion labels (IMO/static) are then propagated over time using the combination of motion cues and appearance-based information of the IMO candidate patches. We evaluate the performance of our method on the KITTI dataset.

1 Introduction

Identifying moving objects is one of the main challenges in the context of autonomous driving. While the advancement of deep learning has shown convincing results to generate semantic segmentation of objects associated with moving objects (e.g. cars, bicycles, pedestrians, etc.), it is still a challenging task to verify whether such object is independently moving or in a static mode. We summarize such moving objects under the term *independently moving objects (IMOs)*.

We propose to combine the deep learning method and the classical geometry approach to identfy IMOs using monocular camera. It is well known that the frame-to-frame egomotion induces the *epipolar constraint* which all corresponding points in two images have to obey to. Points or areas which do not move conformant to the epipolar geometry are obviously candidates for belonging to independently moving objects. However, IMOs can also be *epipolar-conformant*, when they move parallel to the camera motion (for an illustration and a formal definition, see figure 2 and section 4.1).

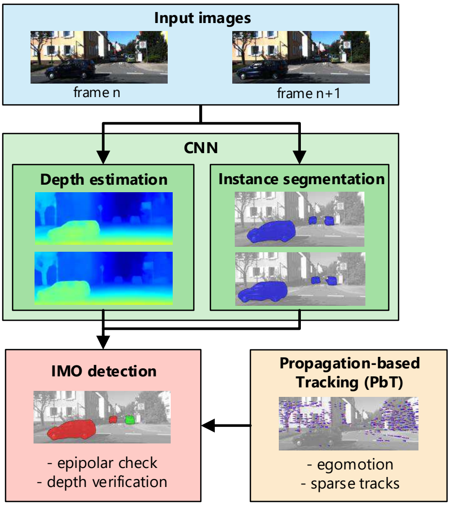

Fig. 1. The scheme of the our novel proposed method to identify IMOs. The *IMO detection* and *depth estimation* blocks are the new contributions.

How to detect IMOs which do not move parallel to the camera motion using monocular camera has been discussed in many papers ([1],[2],[3]). The present paper focuses on the more challenging problem of detecting epipolar-conformant IMOs. The proposed method is built on top of the *propagation-based tracking (PbT)* framework [4], a recently proposed sparse monocular odometry scheme, made available to us by its authors. PbT is one of the leading published monocular visual odometry methods in the KITTI visual odometry benchmark.

The main contribution of our approach is to solve the inherently hard problem for the sparse monocular visual odometry: detecting moving objects which move parallel (or anti-parallel) to the camera motion, such as cars in the same or adjacent lanes, including oncoming traffic. As illustrated in see figure 2, due to its epipolar consistency, a parallel-moving point visually appears exactly as an static point, but in a different (pseudo)distance.

The approach presented here is to employ two CNNs: one that provides a car instance segmentation [5], also used in [4], and a new one designed for and described in this paper that provides depth estimates for single monocular images. In the decoder part of this residual encoder-decoder network, we introduce the new upsampling block. The depth map from the CNN allows us to compare distances obtained from geometric triangulation with such obtained from appearance, and thus supports the detection of epipolar-conformant IMOs also for the monocular case.

We emphasise that the system component presented here, a module that discriminates real moving objects from objects that could be moving ones but are actually standing still currently, is built upon an existing and properly working visual odometry system (PbT/PMO, [4]). This visual odometry (VO) system could be replaced by any other one that works properly and which is (like PMO/PbT) not disturbed by moving objects. In other words, the component we focus on in this paper is independent of the choice of the visual odometry platform it is attached to, as long as this platform fullfils certain functional requirements.

2 Related work

As we focus on the detection of moving cars from a moving ego-vehicle, the scenario is very different to others such as handheld cameras or general robot vision [1, 2], because the motion is strongly constrained by the car dynamics. In the area of advanced driver assistance systems (ADAS), many approaches work with additional information such as using a stereo system [6–9] to identify IMOs. In contrast to these approaches, we want to show that it is possible to reliably detect IMOs from a monocular camera only.

Previously published monocular algorithms on moving vehicle detection can be differentiated into two categories: appearance-based approaches (e.g. [1],[10]) and motion-based approaches (e.g. [11],[12]). We aim at providing an approach that combines both approaches, in a way similar to [13], using the following cues to determine the presence of an independently moving object and to track it: a) the appearance of a car (in terms of a CNN-based car instance detector) as well as b) motion cues from sparse optical flow, considering the epipolar geometry. Our approach shares some similarities with [14] who use two separate CNNs to determine visual odometry and object localization and fuse their results to obtain object localizations. Our method is also related to [15] where CNNs are used to obtain a rigidity score for each object and this is combined with motion cues from optical flow. Bai et al. [16] estimate the dense optical flow fields from each IMO candidate using an approach similar to ours, by employing a CNN to provide the car region candidates. However, they focus only on obtaining the optical flow and do not identify whether the car patches are moving in 3D or not.

Crossing IMOs can be identified because the crossing motions induce inconsistency w.r.t. the epipolar geometry, as discussed in [9]. However, parallel-moving IMOs are epipolar conformant. Klappstein et al. [17] proposed a positive height and depth constraint, but IMOs moving in opposite direction to the ego-car were only detected using a heuristic approach. Wong et al. [18] utilized the size and contours of cars to detect parallel-moving IMOs.

Appearance based dense depth estimate from a single monocular image is one of the key components of our proposed monocular framework, similar to the work by Ranftl et al [19]. We use an encoder-decoder architecture for our CNN. The encoder of our network consists of the ResNet-50 architecture, which was proposed by He et. al [20]. To retrieve the origin size of the input image from last layer of the encoder, we use a decoder, which follows the ideas of the fully convolutional networks [21]. A quite similar encoder-decoder architecture for dense depth map estimation has recently been proposed by Laina et al. [22], but they do not add long skip connections to refine the output.

Those networks can be trained in an unsupervised or supervised way [23] or by a combination of both. The drawback of supervised learning is always the lack of much good labeled training data. To avoid the downside of supervised learning, the authors of [24–26] introduced an (semi-)unsupervised approach for estimating monocular depth maps, where they use stereo images during training to learn the disparity between both images.

Our depth estimation approach belongs to the category of supervised learning. During the training phase (only), we use LIDAR measurements and fuse them to with depth maps, which are computed by SGM [27]. Combining this training idea and our new decoder architecture, we are capable to generate state-of-the-art appearance based depth maps from a single image, which we need to identify for IMO candidates to fully solve the task of detection of parallel moving objects detection through depth verification.

3 Framework overview

The proposed IMO detection scheme builds on a monocular visual odometry framework, the *propagation based tracking (PbT)* scheme [4], which was made available to us by its authors. An important principle of PbT is that each new relative camera pose for a new frame $n + 1$ is *predicted* using the car ego-dynamics. This prediction is used for a soft epipolar tracking (excluding gross deviations from the epipolar structure). Subsequently, a *refined* relative pose is computed only on the basis of keypoints that have been tracked at least twice, this means: keypoints which already passed a stringent test of belonging to the epipolar-conformant environment. All keypoints, including the new ones generated in sparsely covered areas of a new frame, are tracked in an epipolar-guided manner as discussed in more detailed way in section 3.1. All IMO candidate patches in image n are to be classified in one of the three states: `static`, `IMO`, or `undetermined`.

We tackle the problem of IMO detection by classifying the IMOs into two categories: epipolar-conformant IMOs and non-epipolar-conformant IMOs. The keypoints on non-epipolar-conformant IMOs cannot be tracked by the PbT framework, because PbT restricts the matches to be along the epipolar lines. Not finding a photometric consistent match on or close to the epipolar line is thus the basis of labeling keypoints as *'cannot belong to static background'*. This fact serves as the basis of our strategy to detect non-epipolar-conformant IMOs. Failure to track a majority or even all keypoints on an IMO candidate indicates that the IMO candidate is highly likely an IMO.

Detecting epipolar-conformant IMOs, i.e. parallel-moving IMOs, is much more challenging. Monocular camera has an inherent limitation to identify objects moving parallel to the camera. Both static keypoints and parallel-moving keypoints can be tracked using epipolar-style PbT and they look exactly the same by the monocular camera as illustrated in figure 2. This means, a keypoint

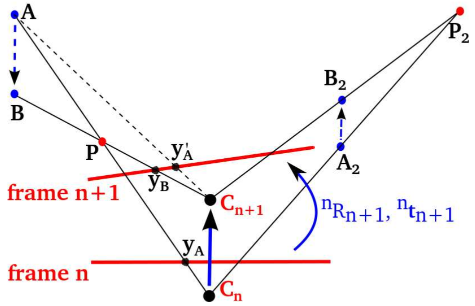

Fig. 2. Bird-eye view. A camera moves from C_n to C_{n+1}. A keypoint which moves parallel to the camera, from A to B, is visually identical to a static point P for the camera. This parallel-moving keypoint is epipolar-conformant. See text for details.

correspondence from a parallel-moving IMO could lead also to an ambiguous static point.

We employ a CNN to provide depth map estimates. With the depth map in hand, we can now detect epipolar-conformant IMOs using a depth verification scheme, consisting of the following two steps:

- Comparison of the depth information between triangulated depth by PbT and CNN depth map on the tracked keypoints observed on IMO candidates.
- Comparison of the relative depth difference extracted from two time-consecutive CNN depth maps of IMO candidates and the egomotion estimates from PbT.

3.1 PbT framework

The principle of keypoint tracking from PbT is used also during IMO detection, thus we give some details in the following. The egomotion of the ego-car is estimated using keypoints which have been confirmed to be static (= belonging to the static environment). These keypoints are the union of keypoints which are not in a CNN-detected car patch, and keypoints from car patches that have been classified as static. In addition, PbT with its epipolar constraint is able to propagate the static label of a car patch on subsequent frames as long as the keypoints inside that car patch are successfully tracked.

As the matching and tracking processes used in the present paper are guided by the epipolar geometry, patches which have a local structure with only one dominant orientation (e.g. lines and straight edges) can be matched as long as the dominant orientation is sufficiently well inclined relative to the epipolar line under consideration. In order to track the keypoint on subsequent frames, we employ an iterative differential matching which minimizes the photometric error between the patch correspondences. A keypoint is finally accepted and used for pose estimation when it has been tracked on at least three consecutive frames which reflects its 3D consistency.

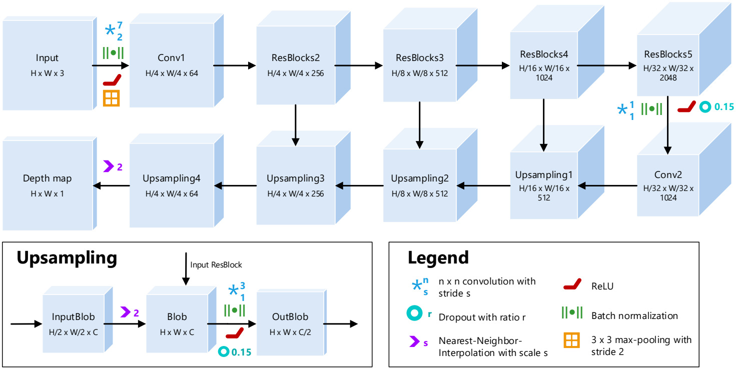

Fig. 3. The proposed network architecture for the depth map estimation. The decoder part is built upon a ResNet-50. We have replaced the fully connected and the global average pooling layer from the original ResNet-50 with our new upsampling block, which takes long skip connection into account. The output of our network has the same size as the input.

3.2 CNN-based IMO candidate patches

IMO candidate patches are obtained by a instance-level segmentation CNN which detects vehicles. We employ the *deep contours* approach, proposed in [5]. The CNN has been trained to label individual vehicles using the Cityscapes dataset [28]. The output of this CNN are 5 channels: one for the semantic label of vehicles and four channels representing the left, right, top, and bottom contours of each vehicle. Based on these information the instances of the vehicles are separated as independent patches, which we use as IMO candidate patches in our proposed framework.

3.3 Propagation of label information

We track the IMO labels over frames by using a dynamic motion model and simple image patch descriptors. Each IMO patch is represented by a feature vector consisting of its center of mass, its size (pixel count), its mean gray value and the gray value standard deviation. We predict the position of the car patch at the next frame using the information of the three last positions based on the assumption of constant 2D acceleration.

The association between the 'old' patches in the image n and a new patch in the image $n+1$ is performed in a looping greedy manner (forward and backward), whenever car patches are observed in both image n and image $n+1$, subjecting each potential association between a patch in image n and a patch in image $n+1$. An association match between two car patches is accepted only when the pair of car patches reciprocally chooses each other as the best match.

3.4 CNN-based depth image

For generating the appearance based dense depth maps, a second CNN takes as input an RGB image and estimates the inverse depth $\rho(\boldsymbol{x}) = d(\boldsymbol{x})^{-1} \in [0,1]$ for each pixel location $\boldsymbol{x} \in \Omega$. Thus, the output size of the encoder-decoder network must be the same as for the input. The architecture of the depth estimation network is depicted in figure 3.

The structure of the encoder part from our network is adopted from the ResNet-50, which was proposed by He et al.[20]. The original ResNets were designed to classify images into different object categories. Hence, their last layers consist of a global average pooling and a fully connected layer to predict the class labels. We replace these layers with our novel upsampling blocks, which act as the decoder. Following the remarks of Odena et al. [29], we do not use unpooling or strided convolution operations to increase the size of the feature maps. Instead, we use nearest-neighbor interpolation to magnify the feature maps. If we increased only the size of the feature maps in this way, the predicted depth maps could not resolve fine structured elements of the image. To solve this problem, we add the output of some residual blocks of the encoder network via long skip connections to the interpolated feature maps. In this way, we allow the network to estimate the depth for fine details in the image. Afterwards, we apply a convolution layer with a kernel size of 3×3 and striding of 1, followed by batch normalization, ReLU and a dropout layer with a dropout ratio of 0.15. The structure of such a upsampling block is shown in the bottom of figure 3.

Our supervised loss function $\mathcal{L}$ is only based on the absolute difference of the estimated inverse depth $\rho_{CNN}(\boldsymbol{x})$ and a measured one $\rho_{GT}(\boldsymbol{x})$, which acts as ground truth. We evaluate the loss only on pixel positions $\boldsymbol{x} \in \Omega_{GT}$, where $\rho_{GT}(\boldsymbol{x})$ is available by a valid measurement. The number of valid measurements is denoted as N.

$$\mathcal{L} = \frac{1}{N} \sum_{\boldsymbol{x} \in \Omega_{GT}} |\rho_{CNN}(\boldsymbol{x}) - \rho_{GT}(\boldsymbol{x})| \tag{1}$$

As training data, we use the raw KITTI data from [30]. We take the KITTI split of [25], which separates the data into a training, validation and testing set. The training set consists of 29000 images. For the ground truth inverse depth map, we fuse for each image the sparse LIDAR measurements from KITTI with the inverse depth map, which is computed with corresponding stereo image by SGM [27]. This allows us to evaluate the loss on many more positions than by only using the sparse LIDAR data. Furthermore, SGM builds a coherency between the image and the depth map, which is crucial for training a CNN. The LIDAR data do not cover this issue, because they are not necessarily synchronized with the camera and the center of the sensors do not coincide, which can lead to unwanted ambiguities.

The encoder part of the network is initialized with pretrained weights from ImageNet. The weights of the decoder part are randomly initialized with the proposed method of [20]. To avoid overfitting, we include dropout layers into the decoder network. We trained our network with a mini-batch size of 4 and use the ADAM optimizer. The network converged after 90 epochs.

4 Detection of epipolar-conformant IMOs

In the monocular case, assuming rigidity for the complete set of points, epipolar-conformant point sets on moving objects will be assigned wrong distance values. Therefore, if we have some information about the depth of a candidate point set, we can design a test on conformity to the static background. The depth map which is needed as side information for this purpose is provided by the described CNN. We detect an epipolar-conformant IMO by showing that the depth of the IMO candidate car, as provided by the depth map, would not fit to the predicted depth calculated with the assumption that the car is static.

The triangulated depth of the target car, with the assumption that the car is static, can be provided by the PbT framework as long as there are some keypoint correspondences on the target car patch. However on some occasions, such as in a fast highway scene where long displacement occurs, no matched keypoint is available on the target car patch. Without tracked keypoints, no triangulation can be done, hence there is no depth prediction.

In order to handle the case when there are no tracked keypoints on a car patch, the predicted depth of the car is obtained from the depth information on the previous frame, and then shifted by the estimated egomotion of the ego-car. We name the above two approaches as keypoint-based and keypoint-free depth verifications.

In this section, first we will prove that parallel-moving objects are consistent to the epipolar constraint. Second, we show that the speed ratio of a parallel-moving point w.r.t. the ego-car speed directly determines the depth of the triangulated ambiguous static point. Then we explain the keypoint-based and keypoint-free depth verifications to detect parallel-moving IMOs.

We assume that the egomotion estimates, more precisely: the relative pose between frames n and $n+1$, are already provided by PbT. We denote $\mathbf{R}$ and $\boldsymbol{t}$ as the relative rotation and relative translation to transform a fixed point $\boldsymbol{z}$ in the world of the camera coordinate system at frame n (CCS_n) to frame $n+1$ (CCS_{n+1}),

$$\boldsymbol{z}_{n+1} = \begin{pmatrix} {}^{n}\mathbf{R}_{n+1} & {}^{n}\boldsymbol{t}_{n+1} \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{z}_n \\ 1 \end{pmatrix} = {}^{n}\mathbf{R}_{n+1} \cdot \boldsymbol{z}_n + {}^{n}\boldsymbol{t}_{n+1} \tag{2}$$

4.1 Proof that a parallel-moving keypoint is epipolar-conformant

We refer to figure 2. Let $\boldsymbol{z}_{A(n)}$ be the 3D coordinate of a position A in CCS_n. The corresponding 3D coordinate in CCS_{n+1} is denoted by $\boldsymbol{z}_{A(n+1)}$ and is given by

$$\boldsymbol{z}_{A(n+1)} = {}^{n}\mathbf{R}_{n+1} \cdot \boldsymbol{z}_{A(n)} + {}^{n}\boldsymbol{t}_{n+1}. \tag{3}$$

Let $\boldsymbol{y}_A$ and $\boldsymbol{y}'_A$ be respectively the normalized image coordinate of $\boldsymbol{z}_{A(n)}$ and $\boldsymbol{z}_{A(n+1)}$ such that,

$$\boldsymbol{z}_{A(n)} = d_{A(n)} \cdot \boldsymbol{y}_A \tag{4}$$

$$\boldsymbol{z}_{A(n+1)} = d_{A(n+1)} \cdot \boldsymbol{y}'_A \tag{5}$$

where d_A is the depth of position A from the camera center. If $\mathbf{E}$ is the essential matrix between the two frames, the epipolar relation can be written as,

$$\boldsymbol{y}_A'^T \cdot \mathbf{E} \cdot \boldsymbol{y}_A = 0 \tag{6}$$

where the essential matrix is given by

$$\mathbf{E} = [\ {}^{n}\boldsymbol{t}_{n+1}]_\times \cdot\ {}^{n}\mathbf{R}_{n+1}. \tag{7}$$

In general, equation (6) applies to every static point. In other words, every static point is epipolar-conformant.

Now, let us consider a moving point which starts at position A at frame n to position B at frame $n+1$. It is important to note that the movement is parallel to the camera motion from frame n to frame $n+1$, as shown in figure 2. The new position at position B after the parallel motion, denoted as $\boldsymbol{z}_{B(n+1)}$, can be expressed as the old position at A plus a shift along the translation direction

$$\boldsymbol{z}_{B(n+1)} = \boldsymbol{z}_{A(n+1)} - v \cdot\ {}^{n}\boldsymbol{t}_{n+1}, \tag{8}$$

where v is a scale parameter describing the speed ratio of the point w.r.t. the ego-car speed. As the relative translation ${}^{n}\boldsymbol{t}_{n+1}$ defined in equation (2) actually describes how the world relatively moves w.r.t. the camera, we need the minus sign in front of v.

Let $\boldsymbol{y}_B$ be the normalized image coordinate of $\boldsymbol{z}_{B(n+1)}$ and d_B is the depth of position B such that the following applies

$$\boldsymbol{z}_{B(n+1)} = d_B \cdot \boldsymbol{y}_B. \tag{9}$$

Now, we can check the epipolar conformity of the moving point

$$\begin{aligned} &\boldsymbol{y}_B^T \cdot \mathbf{E} \cdot \boldsymbol{y}_A \\ &= \frac{(d_{A(n+1)} \cdot \boldsymbol{y}_A' - v \cdot\ {}^{n}\boldsymbol{t}_{n+1})^T}{d_B} \cdot \mathbf{E} \cdot \boldsymbol{y}_A \\ &= \frac{d_{A(n+1)}}{d_B} \underbrace{\cdot \boldsymbol{y}_A'^T \cdot \mathbf{E} \cdot \boldsymbol{y}_A}_{0} - \frac{v}{d_B} \cdot \underbrace{{}^{n}\boldsymbol{t}_{n+1}^T \cdot \mathbf{E}}_{\mathbf{0}} \cdot \boldsymbol{y}_A = 0. \end{aligned} \tag{10}$$

We show in equation (10) that a parallel-moving keypoint also satisfies the epipolar constraint from frame n to frame $n+1$. That means, we have shown that a point moving parallel to the camera motion is epipolar-conformant.

4.2 Depth relation between parallel-moving points and ambiguous static points

As illustrated by figure 2, a keypoint correspondence $\boldsymbol{y}_A$ in frame n and $\boldsymbol{y}_B$ in frame $n+1$ can represent both a moving point from A to B, and an ambiguous triangulated static point P. Let $\boldsymbol{z}_P$ be the 3D coordinate at position P. We investigate the relation between the motion of the parallel-moving point (see equation (8)) and the position of the ambiguous triangulated static point $\boldsymbol{z}_P$.

We compute the intersection of two rays, one from the camera center at CCS_n crossing $\boldsymbol{z}_A$ and another one from the camera center at CCS_{n+1} crossing $\boldsymbol{z}_B$. We transform all coordinates into CCS_{n+1}, thus having the following two equations representing the rays:

$$\boldsymbol{z}_P^{(1)} = \mathbf{0} + \alpha \cdot (\boldsymbol{z}_{B(n+1)} - \mathbf{0}) = \alpha \cdot \boldsymbol{z}_{B(n+1)} = \alpha(1-v) \cdot {}^{n}\boldsymbol{t}_{n+1} + \alpha \cdot ({}^{n}\mathbf{R}_{n+1} \cdot \boldsymbol{z}_{A(n)}) \tag{11}$$

$$\boldsymbol{z}_P^{(2)} = {}^{n}\boldsymbol{t}_{n+1} + \beta \cdot (\boldsymbol{z}_{A(n+1)} - {}^{n}\boldsymbol{t}_{n+1}) = {}^{n}\boldsymbol{t}_{n+1} + \beta \cdot ({}^{n}\mathbf{R}_{n+1} \cdot \boldsymbol{z}_{A(n)}). \tag{12}$$

By comparing equation (11) and (12), as long as ${}^{n}\mathbf{R}_{n+1} \cdot \boldsymbol{z}_{A(n)}$ is not a multiple of ${}^{n}\boldsymbol{t}_{n+1}$, we come to the conclusion that

$$\alpha(1-v) = 1 \quad \rightarrow \quad \alpha = \frac{1}{1-v} \tag{13}$$

applies. It is important to note that α is also the depth ratio between positions B and P (see equation (11)), denoted as d_B and d_P.

$$\frac{d_P}{d_B} = \frac{1}{1-v} \quad \rightarrow \quad (1-v)d_P = d_B \tag{14}$$

Hence, we can identify several cases of parallel-moving points based on the analysis of v:

- If the point moves on the opposite direction w.r.t. camera motion ($v < 0$), then the ambiguous static point is nearer than the moving point ($d_P < d_B$).
- If the point moves at the same direction w.r.t. camera motion with lower speed ($0 < v < 1$), then the ambiguous static point is farther than the moving point ($d_P > d_B$).
- If the point moves at the same direction w.r.t. camera motion with the same speed ($v = 1$), then the ambiguous static point is at infinity ($d_P \rightarrow \infty$).
- If the point moves at the same direction w.r.t. camera motion with higher speed ($v > 1$), then the ambiguous static point $\boldsymbol{z}_P$ is found behind the camera.

4.3 Keypoint-based depth verification

Let $Q(n)$ and $Q(n+1)$ be two associated car patches corresponding to the same car from two consecutive frames n and $n+1$. We employ epipolar matching within $Q(n)$ and $Q(n+1)$ to obtain keypoint correspondences $\boldsymbol{x}_i(n)$ and $\boldsymbol{x}_i(n+1)$, for $i = 1, 2, .., m$. This approach is considered only when the number of correspondences is at least τ_{mc}. Then, we triangulate the correspondences to obtain the 3D coordinates $\boldsymbol{z}_{Pi}$.

We compute the relative difference $\varDelta d_i$ between the triangulated depth d_{Pi} and the depth information from the CNN depth map d_{Bi}:

$$\varDelta d_i = \frac{|d_{Bi} - d_{Pi}|}{d_{Pi}} = \frac{|(1-v)d_{Pi} - d_{Pi}|}{d_{Pi}} = |v|. \tag{15}$$

The keypoint $\boldsymbol{x}_i$ on the car patch Q is recognized as a moving point, if the relative depth difference exceeds τ_v. Hence, τ_v also describes the maximum speed ratio w.r.t. the ego-car speed that can be detected as a moving point.

$$\varDelta d_i > \tau_v \rightarrow \text{moving point} \tag{16}$$

Let m_i be the number of moving points found in patch Q. The car patch Q is identified as an IMO, if the ratio of moving keypoints exceeds τ_{rm}:

$$\frac{m_i}{m} > \tau_{rm} \rightarrow \texttt{IMO}. \tag{17}$$

4.4 Keypoint-free depth verification

For keypoint-free depth comparison, we look into the car patches $Q(n)$ and $Q(n+1)$. Combining the 2D pixel position of the patches and the depth information from the CNN, each car patch can be represented by a single 3D point derived from the 2D center of mass of the patch and the median of the depth values.

The 2D center of masses of the patches $Q(n)$ and $Q(n+1)$ are given by $\boldsymbol{c}(n)$ and $\boldsymbol{c}(n+1)$, respectively. The median depth of patches $Q(n)$ and $Q(n+1)$ are denoted as $d(n)$ and $d(n+1)$. Hence, each patch can be represented by a 3D point $\boldsymbol{z}$ whose x and y positions are defined by the center of mass $\boldsymbol{c}$ and the z position is given by the median depth d.

$$\boldsymbol{z} = d \cdot \mathbf{K}^{-1} \cdot \begin{pmatrix} \boldsymbol{c} \\ 1 \end{pmatrix}, \tag{18}$$

where $\mathbf{K}$ is the intrinsic camera matrix.

Now, the patch $Q(n)$ and $Q(n+1)$ are represented by the 3D points $\boldsymbol{z}(n)$ and $\boldsymbol{z}(n+1)$. However, both 3D points are measured based on their respective camera coordinate systems (CCS). In order to compare them, we transform $\boldsymbol{z}(n)$ into CCS_{n+1},

$$\boldsymbol{z}(n \rightarrow n+1) = {}^{n}\mathbf{R}_{n+1} \cdot \boldsymbol{z}(n) + {}^{n}\boldsymbol{t}_{n+1}. \tag{19}$$

Now, we can calculate the absolute distance between the 3D points representing patches $Q(n)$ and $Q(n+1)$:

$$\varDelta z = |\boldsymbol{z}(n+1) - \boldsymbol{z}(n \rightarrow n+1)|. \tag{20}$$

If both 3D points $\boldsymbol{z}(n \rightarrow n+1)$ and $\boldsymbol{z}(n+1)$ are similar, it indicates that the patch Q corresponds to a static car. However, if they significantly differ, we identify the car as an IMO.

As we deal with parallel-moving cars, the relative position of these cars change only in one axis corresponding to the depth value (z-axis in our setup), hence the depth consistency is the focus to analyze. The x and y components of $\boldsymbol{z}(n \to n+1)$ and $\boldsymbol{z}(n+1)$ are almost always the same. We set τ_{xy} as the maximum value for both x and y components of Δz to be classified as a static car.

Let $d_m(n)$ and $d_m(n+1)$ be the depth (z) components of $\boldsymbol{z}(n \to n+1)$ and $\boldsymbol{z}(n+1)$. We compute the relative depth difference r_{dm} by

$$r_{dm} = \frac{|d_m(n) - d_m(n+1)|}{\min\left(d_m(n), d_m(n+1)\right)}. \tag{21}$$

The car patch is categorized as an IMO, if the relative depth difference is more than $\tau_{dm,IMO}$ and as a static car, if it is less than $\tau_{dm,static}$.

5 Experiments

We tested our method on the KITTI dataset [30]. Since KITTI does not provide IMO labels for the KITTI odometry dataset, we have created our own dataset to evaluate our approach. We also used KITTI MoSeg dataset [31] to compare our results with competing method.

5.1 IMO candidates dataset

For our new dataset, we used the 11 training sequences from the KITTI visual odometry dataset, which consists of 23201 images. The proposed CNN from van den Brand et al. [5] was utilized to generate candidate labels for the vehicle instances. In the current state of the dataset, we have limited the detected objects to vehicles only. This can be further extended to other objects, like pedestrians or bicycle in future work.

Given these segmented candidate labels, we manually assign to each candidate patch in all images one of the following class labels: 0 - background (non-vehicles), 1 - independently moving vehicle, 2 - static (non-moving) vehicle, 3 - far away vehicles (median distance greater than 50m) and 4 - undetermined. We labeled a candidate as undetermined, if the patch does not show a vehicle or if the patch is stretched over more than one vehicles, which do not fall into same category, like static or IMO. Some examples of this dataset are shown in fig 4.

5.2 Evaluation of IMO detection

In our experiments, we used the following values: $\tau_{mc} = 3$, $\tau_v = 0.3$, $\tau_{rm} = 0.4$, $\tau_{xy} = 0.1$, $\tau_{dm,IMO} = 0.05$, $\tau_{dm,static} = 0.01$. As the CNN-based IMO candidate patches can reliably detect IMOs up to a distance of 50 meters, the proposed IMO detection is also evaluated for the same maximum distance. We combine our method with a method from [3] which handles detection for non-parallel-moving objects.

The performance of the IMO classification is expressed by recall R, specificity S, and accuracy A. We also measure the decisiveness of the proposed method

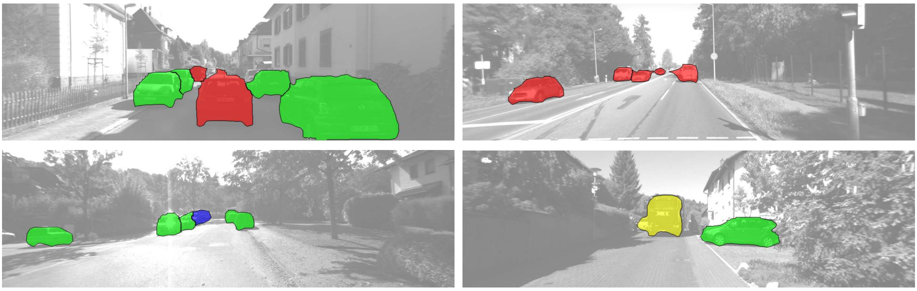

Fig. 4. Examples from our new IMO candidates dataset. The colored overlay encoding is as follows: red ↔ IMO (class 0), green ↔ static (class 1), blue ↔ too far away (class 2) and yellow ↔ undetermined (class 3).

to give definite output (`IMO`/`static`) as compared to `undetermined`. We define the decisiveness level D as

$$D = \frac{n_{IMO} + n_{static}}{n_{IMO} + n_{static} + n_{undetermined}} \tag{22}$$

where n_{IMO}, n_{static}, and $n_{undetermined}$ are respectively the number of outputs as `IMO`, `static`, and `undetermined`.

Accuracy on the KITTI MoSeg dataset Table 1 presents the precision of the IMO detection using our method and using MODNet [31]. The precision of our method is better on both identifying static cars and moving cars. The average precision of our method is 0.79 as compared to 0.66 of MODNet. Figure 5 shows the exemplary results of the IMO detection using our method and using MODNet.

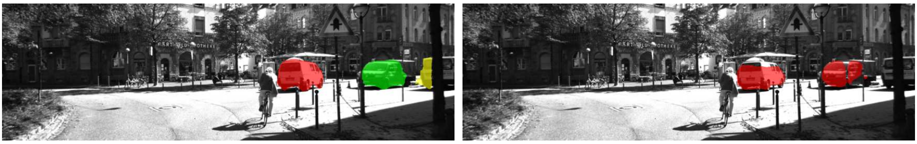

Fig. 5. Exemplary results of the car classification into static and IMO labels on KITTI MoSeg dataset: using our method (**left**) and using MODNet (**right**). Red color represents IMO, green color represents static car, and yellow color represents undetermined. The comparison shows that our method correctly identifies a static parked car while MODNet wrongly classifies it as an IMO.

Accuracy on the KITTI odometry dataset The results of the proposed IMO detection on the KITTI odometry dataset are presented in table 2. The

Table 1. Accuracy of IMO detection on the KITTI MoSeg dataset.

Method	P **static**	P **moving**	P **average**
MODNet [31]	0.65	0.67	0.66
Ours	**0.74**	**0.84**	**0.79**

overall decisiveness level is 91%. That means, the undetermined outputs only happen in about 9% of the total car appearances and they mostly occur when the cars are first time observed in the scene. The recall rate, or the true positive rate, has an overall value of 87% which reflects the high accuracy of the IMO detection. The overall specificity rate, or the true negative rate, is 83%, while the overall accuracy is 84%.

Sequence 01 and sequence 04 are notably full of epipolar-conformant IMOs, both parallel and anti-parallel cases. The results in table 2 for both sequences indicate that the proposed IMO detection is able to identify almost all IMOs. Figure 6 (left image) shows the IMO detection for KITTI sequence 09. The parallel-moving cars are correctly detected and marked with red colors. The static cars are also correctly identified in green colors.

The accuracy level is directly influenced by the user-defined threshold τ_v (see equation (16)) that describes the maximum detectable speed ratio of the moving car w.r.t. the ego-car speed. The threshold τ_v should be low enough in order to be able to detect even slow moving objects, while at the same it cannot be too low to anticipate measurement errors. If an IMO moves very slowly below τ_v, the proposed framework cannot identify it as a moving object, as happened in KITTI sequence 10, when a truck moves backward slowly (see the right image of figure 6). Similarly, if the error in determining triangulated 3D position is too high (e.g. from matching error or egomotion error), it could lead to false positive or false negative classifications.

Table 2. Accuracy of IMO detection on KITTI dataset.

Sequence	D	R	S	A
0	0.90	0.41	0.81	0.81
1	0.84	0.97	n.a.	0.97
2	0.90	0.71	0.82	0.82
3	0.89	1.00	0.86	0.91
4	0.96	1.00	1.00	1.00
5	0.90	0.95	0.86	0.86
6	0.86	1.00	0.86	0.86
7	0.93	0.87	0.89	0.89
8	0.93	0.61	0.81	0.81
9	0.92	0.76	0.90	0.90
10	0.95	0.68	0.97	0.92
Overall	0.91	0.87	0.83	0.84

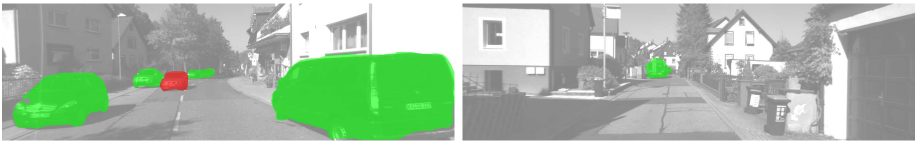

Fig. 6. Exemplary results of the car classification into static and IMO labels on the KITTI odometry dataset sequence 09 (left) and sequence 10 (right). Red color represents IMO while green color represents static car.

6 Conclusion

This paper presents an IMO detection method for the case of a moving monocular camera. The proposed method employs a CNN to provide IMO candidates, and a novel CNN that estimates depth maps from single images. While crossing IMOs can be detected by an epipolar consistency check, we focussed here on the parallel-moving IMOs which are identified through the proposed depth verification scheme. The motion labels (IMO/static) are propagated over time by establishing patch label association between two consecutive frames based on the cue combination of motion and appearance. Experiments on the new KITTI IMO label dataset we created show encouraging performance of the proposed method.

References

1. Jung, B., Sukhatme, G.S.: Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In: International Conference on Intelligent Autonomous Systems. (2004) 980–987
2. Kundu, A., Jawahar, C.V., Krishna, K.M.: Realtime moving object detection from a freely moving monocular camera. In: 2010 IEEE International Conference on Robotics and Biomimetics. (2010) 1635–1640
3. Fanani, N., Ochs, M., Stürck, A., Mester, R.: CNN-based multi-frame IMO detection from a monocular camera. In: Intelligent Vehicles Symposium (IV), IEEE (2017)
4. Fanani, N., Stürck, A., Ochs, M., Bradler, H., Mester, R.: Predictive monocular odometry (PMO): What is possible without RANSAC and multiframe bundle adjustment? Image and Vision Computing (2017)
5. van den Brand, J., Ochs, M., Mester, R.: Instance-level segmentation of vehicles by deep contours. In: Asian Conference on Computer Vision 2016 – Workshops, Springer (2016) 477–492
6. Wedel, A., Meißner, A., Rabe, C., Franke, U., Cremers, D.: Detection and segmentation of independently moving objects from dense scene flow. In: Energy minimization methods in computer vision and pattern recognition, Springer (2009) 14–27
7. Lenz, P., Ziegler, J., Geiger, A., Roser, M.: Sparse scene flow segmentation for moving object detection in urban environments. In: 2011 IEEE Intelligent Vehicles Symposium (IV). (2011) 926–932

8. Ošep, A., Mehner, W., Mathias, M., Leibe, B.: Combined image- and world-space tracking in traffic scenes. In: ICRA. (2017)
9. Zhou, D., Frémont, V., Quost, B., Dai, Y., Li, H.: Moving object detection and segmentation in urban environments from a moving platform. Image and Vision Computing (2017)
10. López-Rubio, F.J., López-Rubio, E.: Foreground detection for moving cameras with stochastic approximation. Pattern Recognition Letters **68** (2015) 161 – 168
11. Yamaguchi, K., Kato, T., Ninomiya, Y.: Vehicle ego-motion estimation and moving object detection using a monocular camera. In: 18th International Conference on Pattern Recognition (ICPR'06). Volume 4. (2006) 610–613
12. Jazayeri, A., Cai, H., Zheng, J.Y., Tuceryan, M.: Vehicle detection and tracking in car video based on motion model. IEEE Transactions on Intelligent Transportation Systems **12**(2) (2011) 583–595
13. Ramirez, A., Ohn-Bar, E., Trivedi, M.M.: Go with the flow: Improving multi-view vehicle detection with motion cues. In: 2014 22nd International Conference on Pattern Recognition. (2014) 4140–4145
14. Oliveira, G.L., Radwan, N., Burgard, W., Brox, T.: Topometric localization with deep learning. ArXiv preprint (2017)
15. Wulff, J., Sevilla-Lara, L., Black, M.J.: Optical flow in mostly rigid scenes. arXiv preprint arXiv:1705.01352 (2017)
16. Bai, M., Luo, W., Kundu, K., Urtasun, R.: Exploiting semantic information and deep matching for optical flow. In: European Conference on Computer Vision, Springer (2016) 154–170
17. Klappstein, J., Stein, F., Franke, U.: Monocular motion detection using spatial constraints in a unified manner. In: Intelligent Vehicles Symposium, 2006 IEEE, IEEE (2006) 261–267
18. Wong, C.C., Siu, W.C., Jennings, P., Barnes, S., Fong, B.: A smart moving vehicle detection system using motion vectors and generic line features. IEEE Transactions on Consumer Electronics **61**(3) (2015) 384–392
19. Ranftl, R., Vineet, V., Chen, Q., Koltun, V.: Dense monocular depth estimation in complex dynamic scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4058–4066
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 770 – 778
21. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 3431–3440
22. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper Depth Prediction with Fully Convolutional Residual Networks. In: International Conference on 3D Vision (3DV). (2016) 239–248
23. Eigen, D., Puhrsch, C., Fergus, R.: Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In: Conference on Neural Information Processing Systems (NIPS). (2014) 2366–2374
24. Garg, R., BG, V.K., Carneiro, G., Reid, I.: Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. In: European Conference on Computer Vision (ECCV). (2016) 740–756
25. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised Monocular Depth Estimation with Left-Right Consistency. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2017)

26. Kuznietsov, Y., Stückler, J., Leibe, B.: Semi-Supervised Deep Learning for Monocular Depth Map Prediction. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
27. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. Transactions on Pattern Analysis and Machine Intelligence (PAMI) **30**(2) (2008) 328–341
28. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes Dataset for Semantic Urban Scene Understanding. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 3213–3223
29. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill (2016)
30. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. The International Journal of Robotics Research **32**(11) (2013) 1231–1237
31. Siam, M., Mahgoub, H., Zahran, M., Yogamani, S., Jagersand, M., El-Sallab, A.: Modnet: Moving object detection network with motion and appearance for autonomous driving. arXiv preprint arXiv:1709.04821 (2017)