Moving-Object Detection From Consecutive Stereo Pairs Using Slanted Plane Smoothing

Long Chen, *Member, IEEE*, Lei Fan, Guodong Xie, Kai Huang, and Andreas Nüchter, *Member, IEEE*

***Abstract*— Detecting moving objects is of great importance for autonomous unmanned vehicle systems, and a challenging task especially in complex dynamic environments. This paper proposes a novel approach for the detection of moving objects and the estimation of their motion states using consecutive stereo image pairs on mobile platforms. First, we use a variant of the semi-global matching algorithm to compute initial disparity maps. Second, assisted by the initial disparities, boundaries in the image segmentation produced by simple linear iterative clustering are classified into coplanar, hinge, and occlusion. Moving points are obtained during ego-motion estimation by a modified random sample consensus) algorithm without resorting to time-consuming dense optical flow. Finally, the moving objects are extracted by merging superpixels according to the boundary types and their movements. The proposed method is accelerated on the GPU at 20 frames per second. The data which we use for testing and benchmarking is released, thus completing similar data sets. It includes 812 image pairs and 924 moving objects with ground truth for better algorithms evaluation. Experimental results demonstrate that the proposed method achieves competitive results in terms of moving-object detection and their motion state estimation in challenging urban scenarios.**

***Index Terms*— Stereo vision, autonomous vehicles, simultaneous localization and mapping.**

I. Introduction

ENVIRONMENT perception is one of the core issues for advanced driver assistance systems (ADAS) and autonomous driving systems (ADS). It has been an active field of research in the intelligent transportation systems (ITS) area during the last decade. Detecting moving objects is a significant component of the perception system [1]–[4], as it is the basis for numerous applications, such as robot navigation [5], [6], simultaneous localization and mapping (SLAM) [6]–[8], traffic surveillance [9]–[12] and video segmentation [13], etc.

Moving-object detection in static environments has been studied extensively [14], and many effective solutions have been proposed, including background subtraction, frame difference, and optical flow. However, moving-object detection in dynamic environments is still a challenging task because it is not feasible to have a unique background model in a dynamic scene. Many kinds of sensors have been used to solve the moving-object detection problem, including monocular cameras, stereo cameras, LIDARs and radars [5], [15], [16]. Among these sensors, the monocular camera lacks scale reference and LIDAR is very expensive. Stereo cameras could provide both color and depth information of the environment at very low cost. This is the major driven-force that stereo-based perception is receiving more and more attention recently.

Most stereo-based methods adopt optical flow or scene flow to detect moving objects [3], [17]–[21]. These flow-based methods are very time-consuming, and moreover, the shadow of moving object would be easily misclassified by using dense optical flow calculation [17], [18]. Other kinds of moving-object detection methods [22]–[24] abandon the process of optical/scene flow calculation and use grid or voxel to establish specific 3D maps to extract moving targets. These methods could not provide pixel-level results which are critical for object segmentation and visual odometry. The above observations motivate us to design a real-time pixel-level moving-object detection method based on stereo cameras.

To solve the time-consuming problem of flow-based methods, in this paper, we employ the RANSAC process on feature points matched circularly between continuous stereo pairs, which requires much less time than optical flow calculation. To extract moving objects completely, we perform a segmentation of input image using superpixels. The relations between superpixels are then sorted into coplanar, hinge and occlusion by applying the slanted-plane method. For the shadow problem, we accurately segment the shadow projected by the moving objects to another plane, such as the ground, by taking the boundary categories into consideration. We estimate the motion of each superpixel according to formerly extracted feature points. Following the image segmentation, superpixels with great possibilities to form one single target and similar in the movement are merged into the final pixel-level detection results. Moreover, we design a stereo matching algorithm based on the GPU by parallelizing image segmentation and semi-dense disparity map computation, which produces a dense disparity map with a slanted plane model for moving-object detection in real-time.

To sum up, our moving-object detection method has four primary contributions, comparing with existing methods:

- To the best of our knowledge, the proposed method is the first one that leverages the type of relation between

Manuscript received July 9, 2016; revised January 16, 2017 and February 27, 2017; accepted March 7, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 41401525, in part by the Natural Science Foundation of Guangdong Province under Grant 2014A030313209, and in part by the CCFTencent Open Fund under Grant IAGR20150114. The Associate Editor for this paper was D. Fernandez-Llorca.

L. Chen, L. Fan, G. Xie, and K. Huang are with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510275, China (e-mail: chenl46@mail.sysu.edu.cn).

A. Nüchter is with the Informatics VII - Robotics and Telematics Group, University of Würzburg, 97070 Würzburg, Germany.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2017.2680538

RGBD and plane segmentation for moving-object detection, and it exhibits robustness and efficiency when dealing with multiple targets while removing the harmful influence of shadows.
- The proposed method is not object type specific and therefore is able to detect all moving objects. The detection result of our method also achieves pixel-level accuracy.
- We propose a GPU-based fast stereo matching algorithm by parallelizing independent processes, which delivers a dense disparity map and an image segmentation with boundary types. By using this matching algorithm, our moving-object detection method runs at 20 frames per second on the GTX 960 graphics card.
- Two datasets containing 812 frames and 924 moving objects selected from KITTI and our SYSU datasets with manually tagged ground truth are released for better algorithms evaluation, which has various objects including vehicles, pedestrians, and bicycles, etc. These datasets could be found at http://www.carlib.net/stereomovingobjects.html/.

The remainder of this paper is organized as follows: Section II briefly reviews the related literature. The strategy behind our method to detect moving objects using consecutive stereo pairs lies in four steps, which we will explain in detail in section III. In section IV, we present the experiments involving two datasets, including KITTI and SYSU. Section V concludes the paper.

II. Related Work

The detection of moving objects in dynamic fields have been studied using various sensors, such as monocular cameras [25], stereo cameras [26], [27] and laser scanners [2], [28], etc. Multiple sensor fusion methods [5], [15], [16] have been proposed recently to detect moving obstacles by combining input data streams. Chavez-Garcia and Aycard [16] considered objects of interest obtained at early stages from multiple sensors to reduce misdetections. Combining data from different sensors truly promotes the detection accuracy but is also challenging, namely in terms of multiple-sensor calibration, signal synchronization, and information association.

Stereo cameras provide image pairs with basic color information, and semi-dense or dense disparity maps could be produced applying stereo matching algorithms [29]–[32]. The method [30] proposed by Yamaguchi *et al.* achieved the state-of-art accuracy by plane-fitting to original disparity maps from semi-global matching method [29]. In our approach, the stereo matching step follows the previous work [30] while accelerating the computation time to 40 milliseconds per frame. Further, a better segmentation of the input image is embedded in our matching algorithm while still producing a boundary label map efficiently.

In [26], an effective approach for moving-object detection is proposed based on modeling the ego-motion uncertainty and then applying graph-cut based motion segmentation. The authors estimated relative camera poses through minimizing the sum of reprojection errors. By propagating the uncertainty of the ego-motion to the RIMF, a motion likelihood for each pixel is obtained. Pixels with a high motion likelihood and a similar depth are detected as a moving object. In contrast to detecting moving objects based on dense optical flow resulting in high computation time consumption, our method estimates the dynamic probability of superpixels [33] supervised by ego-motion knowledge and then merging similar segments to a complete target according to depth planes.

Grid-based data structures are good choices for integrating temporal data from the driving environment due to its high memory-efficiency. Using these data structures for moving-object detection usually starts by distinguishing grid cells of the dynamic environment as free or occupied, then segment and track these cells, to provide an object level representation of the scene. Nguyen *et al.* [23] used a stereo camera to build a two-dimensional (2D) occupancy grid map, and they applied a hierarchical segmentation method to cluster grid cells into object segments. In [24], particles were employed to estimate the occupancy and velocity of the cells in an occupancy grid map for modeling and tracking of a driving environment. Broggi *et al.* [22] presented a full 3D voxel-based dynamic obstacle detection for urban scenarios using stereo vision. Stereo-based point clouds were first sampled into a full voxel-based 3D map. Then, they segmented the voxels into a cluster structure using a flood-fill approach; finally, they labeled the clusters as stationary or moving obstacles based on the ego-motion information. Both grid-based or voxel-based detection methods require a previously established map in a specific format.

The deep-learning technique was introduced to stereo camera-based motion segmentation by Lin and Wang [34]. Instead of using point features that are based on 3D geometric constraints for moving-object detection, they employed high-level spatio-temporal features learned unsupervisedly from raw image data based on Reconstruction Independent Component Analysis (RICA). They proposed a framework that incorporates both the detected moving-point results and the learned spatio-temporal features as inputs to Recursive Neural Networks (RNN) which performs motion segmentation. Makris *et al.* [35] proposed an object detection and object class recognition method fusing intensity and depth information in a probabilistic framework. Further, a system that integrates fully automatic geometry estimation of the scene, 2D object detection, 3D localization, trajectory estimation, and tracking for dynamic scene interpretation from a moving vehicle has been proposed in [36]. The Structure-from-Motion (SfM) and the scene geometry is estimated in real time. In addition, they perform multi-view and multi-category object recognition to detect cars and pedestrians in both camera images in parallel. These methods need models of specific targets, and the final accuracy of moving-object detection is highly relative to the training process.

Moving-object detection datasets were released on the KITTI benchmark [21]. The datasets contain 400 dynamic scenes from raw data collection. Moving objects are selected out using detailed 3D CAD models. A 3D scene flow estimation model is also released in Menze and Geiger [21]. They segmented the scene and then gave each segment rigid motion parameters and corresponding object index, which improves

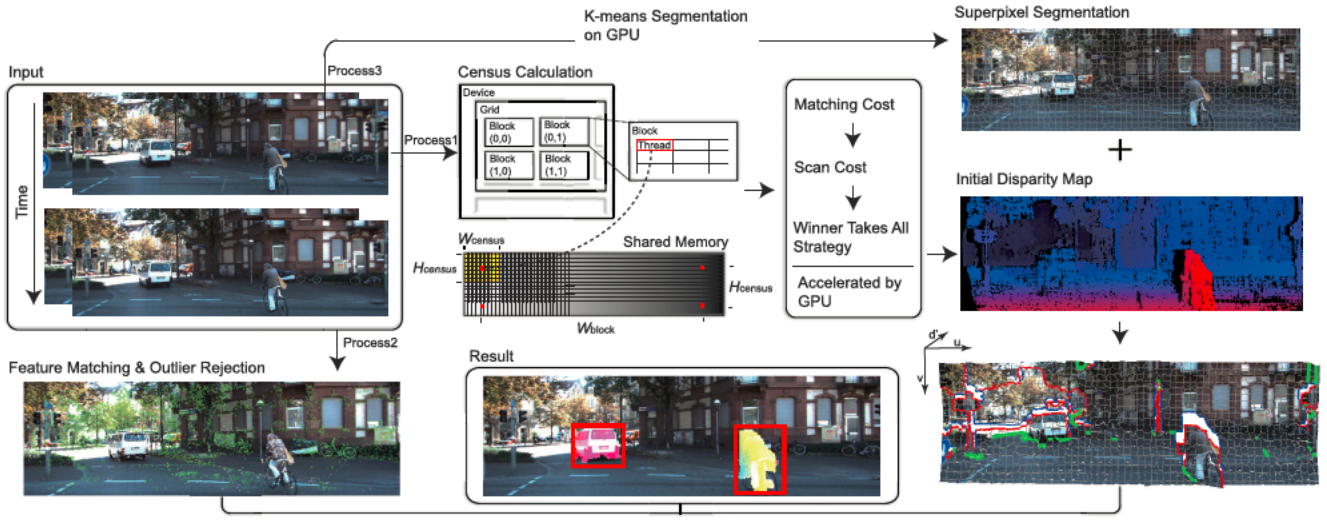


Fig. 1. Overview of the depth assisted moving-object detection method. The input is two consecutive stereo pairs, and the output is the detection result of moving objects covered by their depth. To accelerate, process 1, 2 and 3 run simultaneously before the final fusion.

its robustness. For the purpose of enhancing present available datasets, we release two datasets selected from KITTI raw data and our stereo sequences with manually tagged ground truth of moving objects (http://www.carlib.net/).

III. Slanted Plane Smoothing Assisted Moving-Object Detection

The overview of the proposed method is shown in Fig. 1. We use the fast slanted plane smoothing algorithm to segment the stereo pair and classify the boundary types between adjacent superpixels. We exploit the RANSAC outlier rejection method to calculate the pose change of the camera, to get the inliers (green) and outliers (white), respectively. Afterwards, we figure out the moving objects, which are colored by their depth value, based on the above results. The motion state of the moving objects, includes direction, velocity, and the trajectory are determined simultaneously. We will explain the key components of our methods in this section sequentially.

A. Fast Slanted Plane Stereo Matching

We propose a slanted plane model for stereo matching and boundary type classification based on the previous work [30]. The proposed method reduces computation time dramatically from $2s$ [30] to $0.04s$ with an acceptable loss in accuracy. Our fast stereo matching method consists of four parts including semi-global matching, superpixel segmentation, plane fitting, and energy definition and boundary classification.

1) Semi-Global Matching: The proposed approach first calculates the semi-dense depth map using a variant of the SGM method. This process computes cost between 9×7 patches by deriving Hamming distances on Census Transformation. The cost function is demonstrated as

$$C(p, d_p) = \sum_{q \in W(p)} \{H(T_{L,t}(q), T_{R,t}(q'_{st}(q, d_q))\} \quad (1)$$

where T denotes the Census Transformation. $H(.,.)$ is the Hamming distance between two binary descriptors, and $q'_{st}(q, d_q) = (q_x - d_q, q_y)$ denotes the corresponding pixel in the right image whose disparity is d_q. We utilize 16×16 blocks on the GPU with each thread assigned to a pixel to compute the Census Transformation, and then we employ 32×8 blocks to calculate the Hamming distance. For we omit the directional derivation in the cost function, the proposed method performs less good on the textureless area, such as the sky, but saves a lot of time. During CUDA programming, we apply a parallel reduction in shared memory and thread synchronization in each kernel function. 8 CUDA streams are designed to obtain the minimal scan cost during path aggregation while transferring data between the host (CPU) and the device (GPU).

2) Superpixel Segmentation: We implement our superpixel segmentation base on SLIC methods [33], [37]. Every pixel in the left input image is transformed from RGB color space to CIELab space on the GPU simultaneously. Instead of only distinguishing which segment the boundary pixel belongs to [30], we implement the k-means clustering algorithm by calculating Euclidean distances between every pixel and each center pixel of clusters in a given radius. The Euclidean distance is defined by

$$\Delta D(p, q) = D_{Lab} + D_{pos} \quad (2)$$

where p is the target pixel, and q is the center pixel of one segment.

After primary clustering, we then merge small superpixels by comparing their color with neighboring segments. Small superpixels will be absorbed into its adjacent segments once their color difference is smaller than

$$c_{min} = \alpha \min\{\|\bar{c_{s_i}}, \bar{c_{s_j}}\|_2, \{i, j\} \in \mathcal{N}_{seg}, s_i \cap s_j \neq 0\} \quad (3)$$

where s is a segmentation and c is the color data. Superpixels labeled as similar part will be merged into a single segment recursively. We apply this approach to avoid tiny superpixel noises. In our experiments, the constant α is set between 1.5 to 2.5.

3) Plane Fitting: We assume the 3D scene is piece-wise planar, which stands for the superpixel in our approaches.

After obtaining the initial disparity map, we estimate the plane function for each segment applying RANSAC strategy by extracting 3 pixels with already known disparities circularly. The RANSAC iteration time is dependent on the number of total pixels and the current inliers. The parameters of a plane function are then expressed as $\theta_i = (A_i, B_i, C_i)$. The disparities of unmatched pixels in this segment are estimated as

$$\hat{d}(p, \theta_i) = A_i u + B_i v + C_i \tag{4}$$

where $p = (u, v)$. Note that not all superpixels are given a plane function for there are not enough inliers after iterations.

4) Energy Definition and Boundary Classification: We minimize the energy function during the stereo matching process to obtain the results. We denote our overall energy as $E(\mathcal{I}, s, \theta, f, o, d)$ where $\mathcal{I}$ is the image in CIELab color space, s is a segmentation, θ is a plane expression to each superpixel, f is an outlier flag during plane fitting, o is the boundary classification between adjacent segments, and d is the semi-dense disparity map for smoothing. The energy function is defined as

$$\begin{aligned} E(\mathcal{I}, s, \theta, f, o, d) = & \underbrace{\sum_{p} E_{k-means}(p, c_{s_p}, \mu_{s_p})}_{k-means} \\ & + \underbrace{\lambda_{smo} \sum_{\{i,j\} \in \mathcal{N}_{seg}} E_{smo}(\theta_i, \theta_j, o_{i,j})}_{plane-smoothness} \\ & + \underbrace{\lambda_{com} \sum_{\{i,j\} \in \mathcal{N}_{seg}} E_{prior}(o_{i,j})}_{label-prior} \\ & + \underbrace{\lambda_{mer} \sum_{\{i,j\} \in \mathcal{N}_{seg}} E_{mer}(\bar{c_{s_i}}, \bar{c_{s_j}})}_{segment-merging} \end{aligned} \tag{5}$$

where c_{s_p} and μ_{s_p} denote the color data and location of the center pixel of a segment respectively. The energy function relies on $\mathcal{I}$ and d implicitly.

K-means segmentation: This term encourages pixels to become a superpixel if they are similar in both color and position

$$E_{k-means}(p, c_{s_p}, \mu_{s_p}) = \|p - \mu_{s_p}\|_2^2 + \|\mathcal{I}_{L,t}(p) - c_{s_p}\|_2^2 \tag{6}$$

Boundary classification: We define an energy term that encourages neighbor segments to be coplanar if they belong to the same object (i.e., the boundary between segments need to be classified).

$$E_{plane-smoothness}(\theta_i, \theta_j, o_{i,j}) = \begin{cases} \phi_{occ}(\theta_i, \theta_j) & \text{if } o_{i,j} = \text{fo} \\ \phi_{occ}(\theta_j, \theta_i) & \text{if } o_{i,j} = \text{bo} \\ \frac{1}{\mathcal{B}_{i,j}} \sum_{p \in \mathcal{B}_{i,j}} (\hat{d}(p, \theta_i) - \hat{d}(p, \theta_j))^2 & \text{if } o_{i,j} = \text{hi} \\ \frac{1}{\mathcal{S}_i \cup \mathcal{S}_j} \sum_{p \in \mathcal{S}_i \cup \mathcal{S}_j} (\hat{d}(p, \theta_i) - \hat{d}(p, \theta_j))^2 & \text{if } o_{i,j} = \text{co} \end{cases} \tag{7}$$

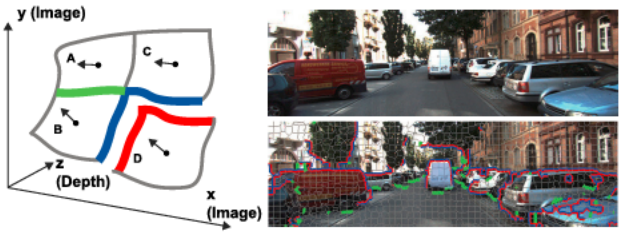


Fig. 2. The left part shows the relationships between four superpixels. The arrows denote the normal vector of each superpixel. The colors of the superpixels contain blue, green and red stands for coplanar, hinge and occlusion respectively. The right part shows the result of the superpixel segmentation.

where $\phi_{occ}(.,.)$ is defined as

$$\phi_{occ}(\theta_i, \theta_j) = \begin{cases} \lambda_{pen} & \text{if } \sum_{p \in \mathcal{B}_{i,j}} (\hat{d}(p, \theta_i) - \hat{d}(p, \theta_j)) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $\mathcal{B}_{i,j}$ are the pixels on the line between segment i and j, $\mathcal{S}_i$ is the set of pixels from segment i. Term fo, bo, hi and co stand for front occlusion, back occlusion, hinge and coplanar respectively. In other words, occlusion occurs between superpixels when the estimated disparity of boundary pixel p do not agree from two planes θ_i and θ_j. Hinge are superpixels agreeing on the boundary pixels instead of every pixel.

Complexity: Coplanarity has a higher prior compared to other two boundary types during classifying

$$E_{prior}(o_{i,j}) = \begin{cases} \lambda_{occ} & \text{if } o_{i,j} = \text{fo} \vee \text{if } o_{i,j} = \text{bo} \\ \lambda_{hinge} & \text{if } o_{i,j} = \text{hi} \\ 0 & \text{if } o_{i,j} = \text{co} \end{cases} \tag{9}$$

where constants are $\lambda_{occ} > \lambda_{hinge} > 0$

Segment merging: This is a term that encourages neighboring small superpixels to merge into a single segment

$$E_{mer}(\bar{c_{s_i}}, \bar{c_{s_j}}) = \begin{cases} 0 & \text{if } \|\bar{c_{s_i}}, \bar{c_{s_j}}\|_2 < c_{min} \\ \lambda_{sim} & \text{otherwise} \end{cases} \tag{10}$$

In the coordinate system (left image in Fig. 2), the boundary type between segment A and segment B is the hinge. The type of boundary between A and C is coplanar, and the type between C and D is occlusion. An example of segmentation is shown in the right part of Fig.2, where green boundaries denote the hinge, and the blue and red boundaries represent the left and right occlusions, respectively, and other boundaries are coplanar.

The use of the superpixel segmentation enables us to extract moving objects more accurately because superpixels more easily adhere to object boundaries. Another benefit is that it reduces the shadow effect. For moving objects in the sunshine, shadows are natural phenomena that move along with the objects. Using the optical flow method to detect moving objects leads to imprecise detections. In our method, by fusing the boundary types and other factors, we avoid this for the cases where the boundary between moving objects and shadows is a hinge or occlusion. We give a step-by-step result in Fig.3.

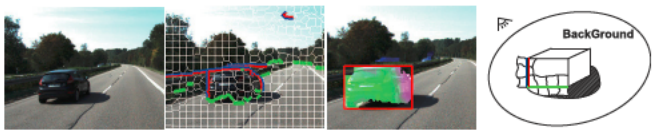


Fig. 3. A sample step-by-step result of the proposed method and the corresponding illustration. From left to right: original image, image segmentation, moving-object detection result (the colored area) with ground truth (the red box), and a sketch illustration.

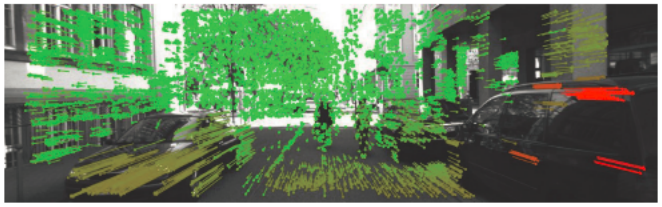


Fig. 4. The endpoint of each line denotes the coordinates of a feature point in continuous frames, and the color indicates the depth information. It is obvious that feature points near to us exhibit larger shifts than those far from us.

B. RANSAC Outlier Rejection

The pixels in the image are labeled as moving or static based on the RANSAC outlier rejection method. The labels extraction work is bound to the ego-motion estimation [38], [39], which obtains the rotation and translation $[R|t]$ of the camera between two consecutive frames. The mechanism expects the feature points to be matched between two frames of the left and right images.

The matching work starts with first finding the best match in the previous left image for the feature candidates in the current left image in an $M \times M$ search window. Continuing with the previous right image, the current right image, and finally matching the current left image again which forms a "circle" match work. Because of dynamic objects in the image, outliers mixed in the feature points are determined before calculating $[R|t]$. To obtain the correct ego-motion estimate, we assume that the static feature points always outnumber the points of the moving part, but there may be several moving objects in different motion states.

We utilize the RANSAC method to classify inliers and outliers among these feature points. Inliers are derived for ego-motion estimation while outliers are considered as labels on moving objects. Next, we select the inliers using a Euclidean reprojection error, and the feature points will be considered as inliers once the error is lower than a given threshold.

By exploiting multi-view geometry knowledge, we design a dynamic threshold when separating outliers based on the disparity image. The feature points located on the near scenes (large disparity value) are considered to have larger shifts between consecutive frames compared to those on the far scenes (small disparity value). The shifts of each feature point between previous and current frames are illustrated in Fig. 4.

Therefore, we assign a dynamic threshold $\mathcal{T}$ to pixel with coordinates (u, v) during the RANSAC process depending on its disparity, and the formula of threshold $\mathcal{T}$ is defined as

$$\mathcal{T}(u,v) = \lambda_{outlier} \frac{d(u,v)}{\bar{d}}$$
$$\bar{d} = \frac{\sum_{p \in I} d_p}{\mathcal{W}_I \times \mathcal{H}_I} \tag{11}$$

where $\lambda_{outlier}$ is a constant value, I stands for the input image, and $\mathcal{W}$, $\mathcal{H}$ denote the width and height. This method shows an obvious superiority by handling both the near objects and the far objects.

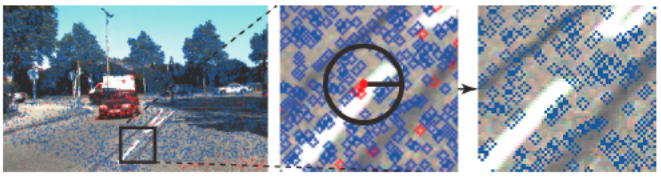


Fig. 5. The k-means like method is used to remove mismatch-caused outliers. We choose outliers as the circle center and calculate the ratio of outliers in a designed radius.

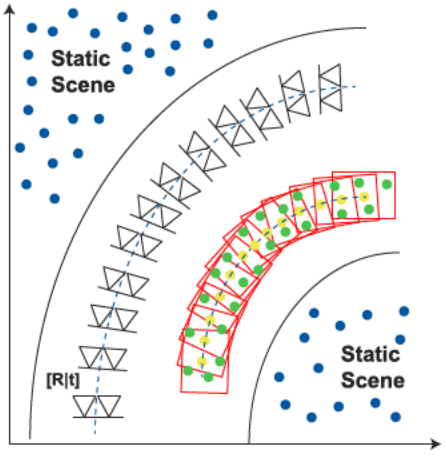


Fig. 6. The blue and green points represent the inliers and outliers produced during ego-motion estimation. The yellow points denote the positions of moving objects that belong to each frame.

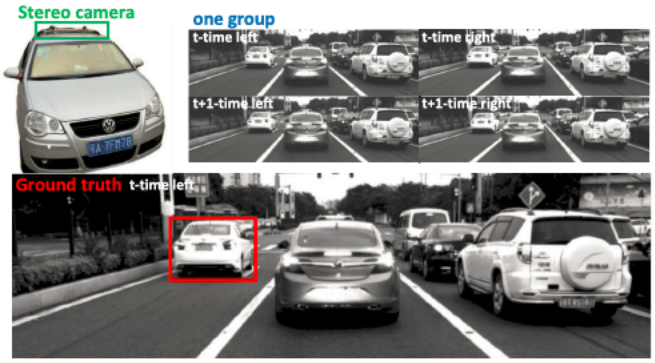


Fig. 7. Recording platform with stereo cameras (top-left), one group of the SYSU dataset including (t-time left image, t-time right image, $(t+1)$-time left image, $t+1$-time right image) and the ground truth.

C. Object Segmentation

Our proposed method utilizes superpixels as the smallest unit of detection as it is easy for them to adhere to the

Fig. 8. The results of KITTI datasets are displayed in column 1 and 2. The results on SYSU datasets are displayed in column 3. The bottom color bar stands for the disparity from large to small.

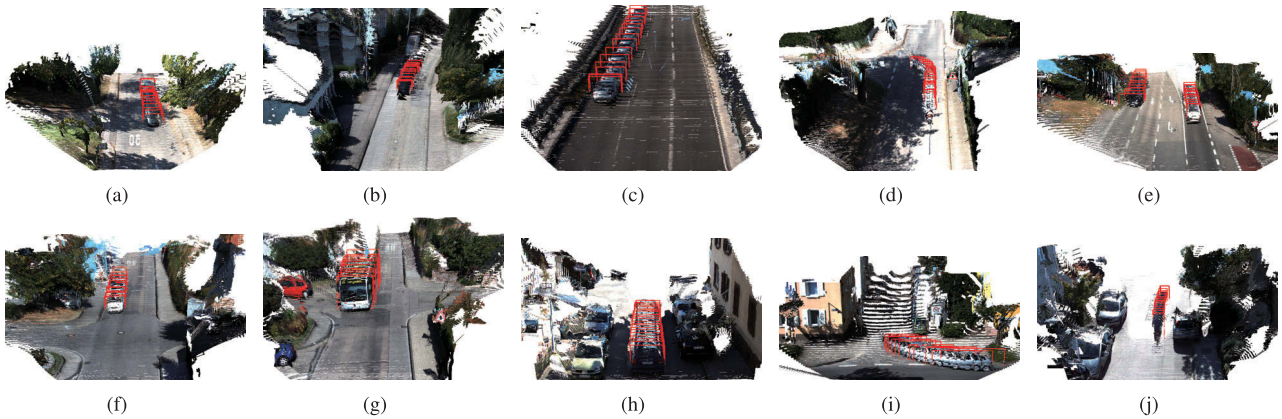


Fig. 9. The results of moving-objects' states estimation. The number of stereo pairs we use to reconstruct are: (a) 12, (b) 16, (c) 16, (d) 27, (e) 14, (f) 12, (g) 12, (h) 24, (i) 14 and (j) 15.

boundaries of objects. The difficulties involved in detecting moving objects and avoiding noises fall into three categories: (1) Sparse outliers are false located on static objects. (2) The number of feature points in a segment is not fixed, and (3) Outliers gather around the edge of the moving object.

Our algorithm starts by labeling each segment to obtain the initial detection results. It then uses the categories of boundaries and other measurements to optimize initial detection results. To filter out the noise of outliers and cause them to mark on target objects to the best, the quality of outliers has to be determined. We employ a k-means like method to measure the quality (Fig.5).

With the qualified outliers, noise may still be present in the result. Because we expect that all the superpixels on the moving objects are detected and no superpixels will be considered as moving in an incorrect way, the refinement should be derived based on the relationships between segments. The noise on the road is filtered out because of the relationships with adjacent segments, which are static and coplanar. The center of a moving object that is short of outliers is optimized by making comparisons with the neighbors. The refinement contains two parts: noises that are filtered and moving objects that are repaired. The critical factors of our algorithm are outlier and inlier, boundary type, segment relation, disparity image, and robotic motion.

D. Motion States Estimation

After obtaining the results of the moving-object detection, our algorithm estimates the states of moving objects by calculating the positions of moving objects in the global

TABLE I
THE TRAINING PART OF [21], OUR KITTI DATASET AND SYSU DATASET

Dataset	Frame	Moving Object	Car	Bicycle	Pedestrian	Van
training dataset[21]	400	472	461	2	0	9
KITTI	612	721	641	34	20	26
SYSU	200	200	172	27	0	1
ALL	1212	1393	1274	63	20	36

TABLE II
THE RESULTS OF OUR ALGORITHM ON KITTI SCENE FLOW TRAINING DATASET [21] AND THE RESULTS BY [22] AND [26]

KITTI	True Moving	False Moving	False Static	True Static	Overlapping (50.0%)	Car	Bicycle	Pedestrian	Van
Ours	442	29	18	n/a	93.64	431	2	0	9
Broggi's [22]	422	51	14	n/a	89.41	402	2	0	9
Zhou's [26]	409	50	27	n/a	86.65	398	2	0	9

TABLE III
THE RESULTS OF PROPOSED ALGORITHM ON OUR KITTI MOVING-OBJECT DATASET AND THE RESULTS BY [22] AND [26]

KITTI	True Moving	False Moving	False Static	True Static	Overlapping (50.0%)	Car	Bicycle	Pedestrian	Van
Ours	647	74	60	n/a	89.74	574	27	20	26
Broggi's [22]	624	130	19	n/a	86.55	551	28	19	26
Zhou's [26]	569	120	116	n/a	78.92	501	25	18	25

TABLE IV
THE RESULTS OF PROPOSED ALGORITHM ON OUR SYSU MOVING-OBJECT DATASET AND THE RESULTS BY [22] AND [26]

SYSU	True Moving	False Moving	False Static	True Static	Overlapping (50.0%)	Car	Bicycle	Pedestrian	Van
Ours	193	7	21	n/a	96.50	165	27	0	1
Broggi's [22]	183	36	7	n/a	91.50	157	25	0	1
Zhou's [26]	177	23	41	n/a	88.50	151	25	0	1

coordinate system. In our method, we use four measurements to describe the moving objects, i.e., the width, height, position, and velocity. Because we have obtained the $[R|t]$ between subsequent frames, we can calculate the position of moving objects using the following equation:

$$\begin{aligned}
\text{disp}_{\text{object}} &= \frac{\sum_{(u,v)\in \text{Area}_{\text{object}}} \text{d}(u,v)}{\text{Area}_{\text{object}}} \\
\text{Pos}_{\text{object}} &= [R|t] \cdot \begin{bmatrix} (u_{\text{object}} - cu) \times \text{baseline}/\text{disp}_{\text{object}} \\ (v_{\text{object}} - cv) \times \text{baseline}/\text{disp}_{\text{object}} \\ \text{focus} \times \text{baseline}/\text{disp}_{\text{object}} \end{bmatrix} \\
[R|t] &= [R|t]_{0,1} \times [R|t]_{1,2} \times [R|t]_{2,3} \times \cdots \\
&\quad \times\ [R|t]_{\text{cur}-1,\text{cur}} \qquad (12)
\end{aligned}$$

where u_{object} and v_{object} denote the average position of each moving object in the left image. It is accessible to determine the nearest or the farthest of the moving objects to the camera because all of the disparity values of the objects are already known.

Finally, we estimate their velocities based on the position of moving objects in the real world. We combine the timestamp of each frame to compute the velocity and the vector of the moving-object detection. An overview of the estimation work for the state of the moving objects is shown in Fig. 6.

IV. EXPERIMENTAL RESULTS

A. Datasets

The dataset [21] only contains pixel-level ground truth for moving vehicles instead of marking all moving obstacles, such as bicycles and pedestrians. In this work, we label moving objects in [21] manually. And to enhance available datasets for measuring the accuracy of moving objects, another two datasets are released with this paper from both KITTI [40] raw data and our stereo dataset. We 1) select several KITTI pairs and label the moving objects to form a KITTI moving objects dataset, which is called KITTI for short in this work, and 2) create a new stereo moving objects dataset by our moving platform, which is named SYSU (Sun Yat-sen University) dataset. KITTI contains 612 groups of images with the 1299×374 resolution, and each group consists of four frames including two successive stereo pairs, i.e., t-time left frame, t-time right frame, $(t+1)$-time left frame and $(t+1)$-time right frame. KITTI groups are selected from more than 5000 frames of images in KITTI benchmark. SYSU dataset containing 100 groups are collected by our stereo camera mounted on the roof of our moving platform with a resolution of 1299×374 under urban road environments conditions. Fig. 7 shows our moving platform and an example of SYSU dataset.

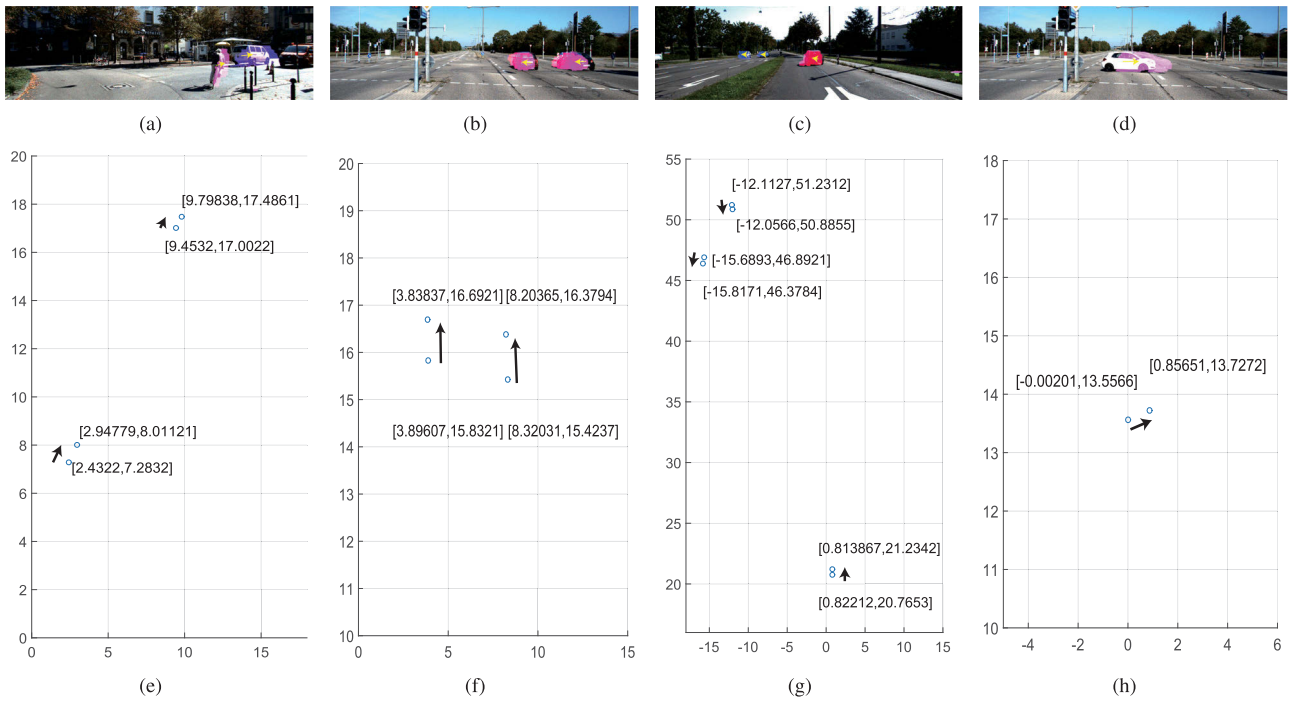


Fig. 10. The position of each moving objects in the next frame are demonstrated in (a), (b), (c) and (d) with their disparity, and the yellow arrow presents the motion direction. Detailed positions of moving objects are illustrated in (e), (f), (g) and (h) in airview.

Each frame of our DATMO datasets contains moving objects, and frames whose main part of the moving objects are out of the image are not selected into our datasets. There are 921 moving objects in all of our DATMO datasets and an average of 2.26 moving objects per frame. The ground truth of two datasets is generated by boxes marking the moving objects in the left current image. Objects sheltered of less than 50% and moving in the scene are also regarded as moving objects. We subdivide the moving objects into four categories including cars, vans, bicycles, and pedestrians. The details of these datasets are listed in Table I and they have already been published on http://www.carlib.net/.

B. Moving Object Detection Results

Detailed results containing consecutive frames of our method could be found at https://youtu.be/DUGcoNMu0S8. Fig. 8 shows twelve output examples from our moving-object detection method in both urban areas and highway situations. Different types of moving objects, such as vans, cars, and bicycles occur in these scenarios. In Fig. 8 (b), there are three moving vehicles in this image, and two of them move in the opposite direction of the other car. Noises might appear on the road for the highlight of the road surface easily lead to mismatching. These mismatches are reduced by judging its quality and utilizing the boundary types. The false negative appears in Fig. 8 (g) is primarily due to insufficient feature points marking on this moving vehicle. In Fig. 8, false positives occur in subgraph (c) and (f). The reason is the blur on the input images. The blur occurs when cameras move fast, and it leads to wrong matches between stereo images, which further makes the wrong classification of boundary types. Blur holds back the process of utilizing RANSAC to extract outliers as well during ego-motion estimation. Outliers located on static scenes are the primary cause of false positives in subfigure (a). In Fig. 8 (h), the man riding his bicycle is tagged out. Different from other pattern recognition based detection methods, our method is suitable for general moving objects with no need for specific types of objects. The proposed method shows more robustness for avoiding the situation that there is no corresponding pre-training model. In Fig. 8 (i), two vehicles overlapped each other do not affect the detection performance, which benefits from the superpixel segmentation to adhere to the boundaries of objects more accurately.

To quantitatively demonstrate the accuracy of our method, we compare our work with two related works in the literature [22], [26]. The results are summarized in Table II, Table III, and Table IV. Compared to [22], the overlapping area larger than 50% is 4.23%, 3.19% and 5.00% higher than [22] on three datasets respectively. From Table II, Table III and Table IV, we present the comparison with Zhou's [26]. The number of detection results with an overlapping area larger than 50% is 9.12% higher than [26] on three datasets totally. There are 73 less false moving while 127 more true moving detections with comparison to [26] among all 1393 moving objects ground truth. Table II, Table III and Table IV present the accuracy performances of three algorithms for detecting cars, bicycles, pedestrians, and vans. We present the results on Table II, III and IV when the recall value

$$\frac{TrueMoving}{TrueMoving + FalseMoving + FalseStatic} \tag{13}$$

reaches its least, which is influenced by the parameters of our algorithm. The shape of a moving object is usually

not a rectangle. Thus we figure out the moving objects by considering the overlapping regions larger than 50%.

C. Motion States Estimation Results

To show the path of moving objects, we tested our method on the KITTI odometry dataset. The results are shown in Fig. 9. For better demonstration, we reconstruct the 3D scene using consecutive frames, and it is obvious that moving objects appear in different positions (ghosting) in the same coordinate system. Therefore our moving-object detection method that combines the approaches of stereo matching and ego-motion estimation, 3D reconstruction benefits from this approach. The red box stands for the position of moving objects referring to each frame. The distance between the red boxes belonging to the same objects is utilized for the velocity estimation. The width and height of boxes denote the size of moving objects. In Fig. 9 (d), a man riding his motorcycle comes from the left road and then drives in the direction of our stereo cameras. The path of the motion is clearly visible.

With the time capturing each frame and the location, the velocity of moving objects is then calculated. The average speed of the vehicle in Fig. 9 (a) is $8.45m/s$, i.e., $30.42km/h$. Similarly, the car in Fig. 9 (b) moves at $18.49m/s$ or $66.56km/h$. The motion direction of these vehicles or bicycles between two frames is shown in Fig. 10. Our method computes the position of the moving objects in both current and previous images simultaneously, i.e., we draw the disparity of moving objects in current frame on the previous image which shows the shift apparently. The disparity of moving objects help us to estimate their location, and the direction is displayed with a yellow arrow in the 2D image. With this information, we are able to further predict the intention of moving vehicles based on former movements.

D. Operation Time

All experiments in this paper are completed on a desktop computer whose processor is Intel Core i7-4770 CPU 4 cores @ 3.4GHz with 8 GB RAM and GTX 960 graphics card. The stereo matching method, accelerated by CUDA, has reached $0.04s$ per frame. The average time of ego-motion estimation time is $0.02s$, and the time of moving objects capturing is $0.01s$. Compared to the average time of [26] with 10 frames per second, our method is faster while doing more work for other applications, such as 3D reconstruction.

V. Conclusion

In this paper, we have proposed a novel and efficient stereo moving-object detection method which obtains pixel-level results of moving targets at $20Hz$. The proposed method achieves accurate detection results in challenging scenarios and is independent of dense optical flow calculation. The states of moving objects, including location, direction, and velocity, are also obtained simultaneously as further byproducts. The proposed work is the first approach that introduces superpixel boundary classification into moving-object detection which ameliorates the shadow effect. Additionally, our method is not object type specific. Detection results in several videos also show the robustness of dealing with various kinds of targets. In future work, we will focus on the large-scale mapping coupling with moving-object detection for autonomous driving, especially by using low-cost cameras and embedded computing.

References

[1] C. Schlenoff, R. Madhavan, and T. Barbera, "A hierarchical, multi-resolutional moving object prediction approach for autonomous on-road driving," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2. Apr./May 2004, pp. 1956–1961.

[2] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, 2008.

[3] C.-W. Liang and C.-F. Juang, "Moving object classification using a combination of static appearance features and spatial and temporal entropy values of optical flows," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3453–3464, Dec. 2015.

[4] X. Du and K. K. Tan, "Comprehensive and practical vision system for self-driving vehicle lane-level localization," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2075–2088, May 2016.

[5] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *Int. J. Robot. Res.*, vol. 26, no. 9, pp. 889–916, 2007.

[6] J. Min, J. Kim, H. Kim, K. Kwak, and I. S. Kweon, "Hybrid vision-based SLAM coupled with moving object tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, May/Jun. 2014, pp. 867–874.

[7] S. Wangsiripitak and D. W. Murray, "Avoiding moving outliers in visual SLAM by tracking moving objects," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 375–380.

[8] J. Choi and M. Maurer, "Local volumetric hybrid-map-based simultaneous localization and mapping with moving object tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2440–2455, Sep. 2016.

[9] J. S. Kim, D. H. Yeom, and Y. H. Joo, "Fast and robust algorithm of tracking multiple moving objects for intelligent video surveillance systems," *IEEE Trans. Consum. Electron.*, vol. 57, no. 3, pp. 1165–1170, Aug. 2011.

[10] J. Hao, C. Li, Z. Kim, and Z. Xiong, "Spatio-temporal traffic scene modeling for object motion detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 295–302, Mar. 2013.

[11] J. H. Ko, B. A. Mudassar, and S. Mukhopadhyay, "An energy-efficient wireless video sensor node for moving object surveillance," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 1, no. 1, pp. 7–18, Jan./Mar. 2015.

[12] H. T. Nguyen, S.-W. Jung, and C. S. Won, "Order-preserving condensation of moving objects in surveillance videos," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2408–2418, Sep. 2016.

[13] S. Amri, W. Barhoumi, and E. Zagrouba, "A robust framework for joint background/foreground segmentation of complex video scenes filmed with freely moving camera," *Multimedia Tools Appl.*, vol. 46, nos. 2–3, pp. 175–205, 2010.

[14] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: A systematic survey," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 294–307, Mar. 2005.

[15] L. Montesano, J. Minguez, and L. Montano, "Modeling the static and the dynamic parts of the environment to improve sensor-based navigation," in *Proc. IEEE Int. Conf. Robot. Autom.* Apr. 2005, pp. 4556–4562.

[16] R. O. Chavez-Garcia and O. Aycard, "Multiple sensor fusion and classification for moving object detection and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 525–534, Feb. 2016.

[17] A. Talukder, S. Goldberg, L. Matthies, and A. Ansar, "Real-time detection of moving objects in a dynamic scene from moving robotic vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, vol. 2. Oct. 2003, pp. 1308–1313.

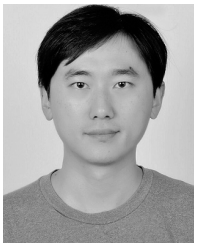
[18] X. Li and C. Xu, "Moving object detection in dynamic scenes based on optical flow and superpixels," in *Proc. IEEE Conf. Robot. Biomimetics*, Dec. 2015, pp. 84–89.

[19] S.-W. Yang and C.-C. Wang, "Multiple-model RANSAC for ego-motion estimation in highly dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3531–3538.

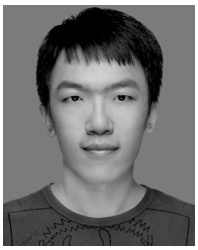
[20] M. Derome, A. Plyer, M. Sanfourche, and G. Le Besnerais, "Moving object detection in real-time using stereo from a mobile platform," *Unmanned Syst.*, vol. 3, no. 4, pp. 253–266, 2015.

[21] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3061–3070.

[22] A. Broggi, S. Cattani, M. Patander, M. Sabbatelli, and P. Zani, "A full-3D voxel-based dynamic obstacle detection for urban scenario using stereo vision," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Oct. 2013, pp. 71–76.
[23] T.-N. Nguyen, B. Michaelis, A. Al-Hamadi, M. Tornow, and M.-M. Meinecke, "Stereo-camera-based urban environment perception using occupancy grid and object tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 154–165, Mar. 2012.
[24] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and tracking the driving environment with a particle-based occupancy grid," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1331–1342, Dec. 2012.
[25] H. Lategahn and C. Stiller, "Vision-only localization," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, pp. 1246–1257, Jun. 2014.
[26] D. Zhou, V. Fremont, B. Quost, and B. Wang, "On modeling ego-motion uncertainty for moving object detection from a mobile platform," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 1332–1338.
[27] A. Ess, K. Schindler, B. Leibe, and L. Van Gool, "Object detection and tracking for autonomous navigation in dynamic environments," *Int. J. Robot. Res.*, vol. 29, no. 14, pp. 1707–1725, 2010.
[28] M. Montemerlo *et al.*, "Junior: The Stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, 2008.
[29] H. Hirschmüller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2005, pp. 807–814.
[30] K. Yamaguchi, D. McAllester, and R. Urtasun, "Efficient joint segmentation, occlusion labeling, stereo and flow estimation," in *Proc. IEEE Eur. Conf. Comput. Vis. Pattern Recognit.* Sep. 2014, pp. 756–771.
[31] A. Seki and M. Pollefeys, "Patch based confidence prediction for dense disparity map," in *Proc. Brit. Mach. Vis. Conf.*, vol. 10. 2016.
[32] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5695–5703.
[33] R. Birkus, "Accelerated gSLIC for superpixel generation used in object segmentation," in *Proc. CESCG*, vol. 15. 2015, pp. 9–16.
[34] T.-H. Lin and C.-C. Wang, "Deep learning of spatio-temporal features with geometric-based moving point detection for motion segmentation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May/Jun. 2014, pp. 3058–3065.
[35] A. Makris, M. Perrollaz, and C. Laugier, "Probabilistic integration of intensity and depth information for part-based vehicle detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1896–1906, Dec. 2013.
[36] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool, "Dynamic 3D scene analysis from a moving vehicle," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
[37] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
[38] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2010, pp. 486–492.
[39] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3D reconstruction in real-time," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2011, pp. 963–968.
[40] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.



Long Chen (M'12) received the B.Sc. degree in communication engineering and the Ph.D. degree in signal and information processing from Wuhan University, Wuhan, China, in 2007 and 2013, respectively. From 2008 to 2013, he was in charge of environmental perception system for autonomous vehicle Smart V-II with the Intelligent Vehicle Group, Wuhan University. From 2010 to 2012, he co-trained Ph.D. Student with the National University of Singapore. He is currently an Associate Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. His areas of interest include the perception system of intelligent vehicle.



Lei Fan is currently working toward the degree in computer science with Sun Yat-Sen University, China, under the supervision of L. Chen. His research interests include 3D reconstruction and stereo matching algorithms. He is focused on the detection and tracking of moving objects and 3D maps for robotics navigation.

Guodong Xie is working toward pursuing the master's degree in computer science with Sun Yat-Sen University, China. He is currently an Instructor. His research interests include stereo matching algorithm. His current research interests include real-time stereo match algorithm.

Kai Huang received the Ph.D. degree from ETH Zürich, Zürich, Switzerland, in 2010. He was a Senior Researcher with the Computer Science Department, Technical University of Munich, Germany, from 2012 to 2015, and a Research Group Leader with fortiss GmbH, Munich, Germany, in 2011. He joined Sun Yat-Sen University, as a Professor, in 2015. His research interests include techniques for the analysis, design, and optimization of embedded systems, particularly in the automotive domain. He was a recipient of Best Paper Awards ESTIMedia 2013, SAMOS 2009, and the General Chairs Recognition Award For Interactive Papers in CDC 2009. He is a Regional Editor of *Elsevier Journal of Circuits, Systems, and Computers*, and has been served as a member of the technical committee on Cybernetics for Cyber-Physical Systems of the IEEE SMC Society, since 2015.

Andreas Nüchter (M'00) received the Diploma degree in computer science and the Ph.D. degree (Dr. rer. nat) from University of Bonn. In summer 2013, he headed as an Assistant Professor with the Automation Group, Jacobs University, Bremen. He was with University of Osnabrück, Fraunhofer Institute for Autonomous Intelligent Systems (AIS), Sankt Augustin, University of Bonn, and Washington State University. He is currently a Professor of Computer Science (telematics) with the University of Würzburg, Germany. He was involved in robotics and automation, telematics, cognitive systems, and artificial intelligence. His main research interests include reliable robot control, 3D environment mapping, 3D vision, and laser scanning technologies. He is a member of the GI. His Ph.D. thesis was shortlisted for the EURON Ph.D. Award.