IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 10, 2017, accepted October 16, 2017, date of publication October 20, 2017,
date of current version November 14, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2764546

Dense Scene Flow Based Coarse-to-Fine Rigid Moving Object Detection for Autonomous Vehicle

ZHIPENG XIAO[1], BIN DAI[1], TAO WU[1], LIANG XIAO[1], AND TONGTONG CHEN [2]
[1]College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha 410073, China
[2]Beijing Special Engineering Design Institute, Beijing 100122, China

Corresponding authors: Zhipeng Xiao (xiaozhipeng.cs@hotmail.com) and Bin Dai (daibin.cs@hotmail.com)

This work was supported in part by the National Nature Science Foundation of China under Grant 61375050 and Grant 91220301.

ABSTRACT Many classical visual odometry and simultaneous localization and mapping methods are able to achieve excellent performance, but mainly are restricted on the static scenes and suffer degeneration when there are many dynamic objects. In this paper, an efficient coarse-to-fine algorithm is proposed for moving object detection in dynamic scenes for autonomous driving. A motion-based conditional random field for this task is modeled. Particularly, for initial dynamic–static segmentation, a superpixel-based binary segmentation is processed, and further for refinement, a pixel-level object segmentation in local region is performed. Additionally, to reduce the projection noise caused by disparity estimation, an approximate Mahalanobis normalization is provided. Finally, in order to evaluate the proposed method, two relative methods are compared as baseline on the public KITTI data set for visual odometry and moving object detection separately. The experiments show the effectiveness and improvement on odometry when the dynamic region is removed and also on moving objects detection.

INDEX TERMS Moving object detection, visual odometry, dynamic-static segmentation, conditional random field, approximate Mahalanobis normalization.

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is a fundamental module to autonomous vehicle and helps to fuse spacial and temporal information from multiple frames, improving the ability of perception for autonomous driving. With decades' efforts, the techniques of SLAM have been improved greatly [1]. However, typical SLAM algorithms tackle mainly static scenes and seldom dynamic cases [2]–[5], where the dynamic parts are removed by simple robust weighting strategy [5] or basic RANSAC scheme [6]. These often work when there is only a small part of dynamic objects. However, when the dynamic parts increase to a significant amount in a whole scene, the system suffers the degeneration. The features used for robust relative pose estimation probably come from the dynamic parts which contaminate the performance.

Therefore, many works begin to focus on motion segmentation or rigidly moving object detection in dynamic scenes.

Unlike some methods optimized only on super-pixel level or direct on pixel level [8], [9], our proposed algorithm combines both to introduce a dense optical flow based two-stage coarse-to-fine algorithm using conditional random field model. Figure 1 illustrates a typical result of our algorithm. Our goal is not only to segment the dynamic and static regions in a scene but also to estimate the 6-DOF motion of each object including the background, namely the ego-motion of the vehicle.

The contributions in this paper are summarized as follows:

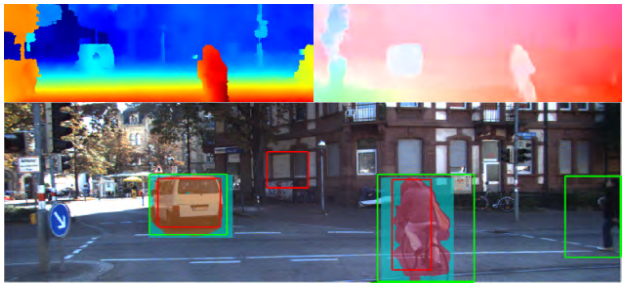
1) a pipeline for two-stage coarse-to-fine moving object detection is presented;
2) a dense scene flow based conditional random field model is proposed;
3) an approximate Mahalanobis normalization is modeled in order to reduce the projection noise caused by disparity estimation;
4) experiments are made to evaluate the effectiveness and improvement on visual odometry by removal of moving regions and also on moving object detection.

II. RELATED WORKS

For motion estimation and rigidly moving object detection in dynamic scenes, many representative approaches have been studied on this task.

In computer vision field for motion estimation, the related works of Vogel *et al.* [10] and Menze and Geiger [11]

(a)



(b)

FIGURE 1. Illustration on this issue. (a) It shows the overlap by reprojecting the previous image into the current image through ego-motion. The misalignments occur around the moving objects like cyclist and vehicle due to their motions. (b) The top row shows the disparity image and dense optical flow both coded in color respectively. The bottom row shows the proposed final moving objects detection in light cyan and segmentation in dark color compared to the baseline method [7] with the detection in red bounding boxes. Note that the green bounding boxes comes from the projection of ground truth 3D tracklets.

assume the dynamic scenes to be decomposed into many pieces of 3D planes under rigid motion, and propose delicated CRF model for pixel-wise highly accurate scene flow field estimation. Similarly, Yang and Li [12] model the scene to be multiple piecewise parametric models where the regions are determined adaptively. Different from other binary labeling problem, it has to infer both continuous and discrete variables at the same time. It exploits high order label cost constrains as in [13] and also multi-model fitting strategy as [14] to let regions with similar motions merge automatically, leading to efficient representation and robust flow estimation. More complicated, Sevilla-Lara *et al.* [15] extend the work of Sun *et al.* [16] and not only apply the robust local fully-connected layered model to segment the foreground and background, but also leverage the ability of deep networks to propose a semantic segmentation and estimate motions on different objects separately. However, their core idea is to find a best representation on the flow field, but all of them do not care much on integral objects, therefore it often appears unreasonably multiple motions on a rigid object.

Other similar works aim to dynamic object detection like [17]–[19] which regard different motions as subregions in a union space leading to a linear subspace clustering problem where it addresses to find the minimal linear subspaces that best represent the observed motion trajectories. This method has its advantage that it naturally combines the motion estimation and clustering automatically in a well defined mathmematical formula. However, it often has to tackle the situations where noise or outliers occur, because this kind of formulation is relatively ideal to the real word cases.

Instead of the well mathematically formulated method, a representative work for dynamic object detection from Lenz *et al.* [7] exploits the principle that moving objects have points with different motion compared to neighbors. Therefore, it builds a graph-like Delauny triangulation net to model the spatial relationship between each feature point and then applies Mahalanobis distance to normalize the velocity to eliminate the uncertainty caused by depth estimation. Finally use threshold strategy to merge and divide the points into groups by their motion discrepancies. Other more direct methods like [20] and [21] both provide dynamic object segmentation systems based on motion information in which they just intuitively apply RANSAC based methods to cope with dynamic scenes. Although they illustrate the effectiveness, yet it does not show significant improvement on performance.

Some other related works focus more on general object detection not specifically moving objects. Fulkerson *et al.* [8] propose a superpixel based graph model to identify and localize object classes in images. It firstly aggregates feature histograms in the neighborhood of each superpixel and then exploits CRF to refine the result. Wojek and Schiele [9] propose a dynamic CRF for scene labeling. It relies on classifiers including object detectors to infer the pixel's label. Further, in order to gain the moving object information, it exploits Kalman filter to predict the motion and sets it into a two layered CRF model. However, this filter-like processing gives a restrict motion model.

III. SYSTEM OVERVIEW

The proposed system assumes the scenes to comply with rigidity and also assumes that the objects of a scene are modeled as dynamic and static categories, where the stationary objects and the background are regarded as static parts while the moving objects are referred to dynamic parts. Note that, the output of the system is not only the detection of moving objects in a scene but also the motion estimation of each object including the ego-motion of the vehicle itself.

Figure 2 illustrates the pipeline of the proposed system and the procedures are detailed in the following sections. Section IV briefly describes the initialization module. Section V introduces the proposed conditional random field model as a fundamental formula for two-stage processes and then followed by section V-C1 and section V-C2 for explanation. The rest sections are complements to the whole procedures.

IV. INITIALIZATION

For initialization, some classic methods are utilized. In order to avoid missing some parts of the scene, a dense optical flow field is required for point-wise correspondences.

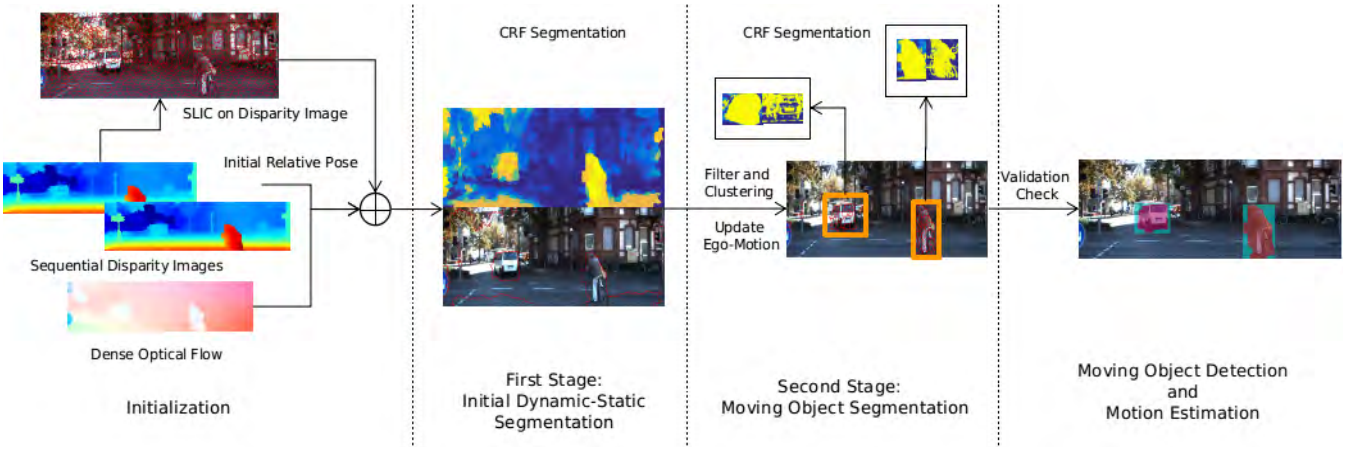


FIGURE 2. System overview. It consists of four main procedures. In the initialization stage, disparity map and dense optical flow field are computed. Then the superpixels are obtained from the disparity map. All these outputs are sent together to the CRF model for the next two procedures including the first stage of initial dynamic-static segmentation and the second stage of moving object segmentation. Finally, some post processing are taken for final outputs.

As many works on dense optical flow can provide outstanding performance [22]–[24], the GPU based FlowNet [24] is utilized for its computation efficiency. As for the excellent performance, the popular slanted-plane smoothing stereo matching (SPSStereo) method in [25] is applied. When these are acquired, the superpixels are computed by simple linear iterative clustering (SLIC) [26] algorithm on the disparity map. The initial and updated ego motion are calculated by method in [6].

V. CRF MODEL FOR COARSE-TO-FINE MOVING OBJECT DETECTION

In our task, as typical assumption for outdoor scenario, it is modeled as a set of rigid 3D structures where can be approximately represented as many pieceswise planar superpixels. It is also assumed that the objects in the scenario only move rigidly and the total number of them is not limited. As so, given two consecutive stereo images, our goal is to estimate both the static and dynamic region, and also their motions accordingly.

More formally, let $\mathbf{L} = \{0, 1\}$ and $\mathbf{O} = \{\mathbf{o}_k | k = 0, \cdots, K\}$, denote the sets of scene labeling and objects respectively, which K represents the number of moving objects but with no upper bound. For each pixel i, its attribute accompanies with label and object information $\{l_i, \mathbf{o}_k\}$, where $l_i \in \mathbf{L}$ and $\mathbf{o}_k$ means that it belongs to object class k with the mapping function $k = C(i)$. The scene is classified as background ($l = 0$) and foreground ($l = 1$). The background is of the static regions including the stationary objects while the foreground is of the moving objects like cars, pedestrians or something unclassified but moves. Each object $\mathbf{o}_k \subset \Omega^2$ represents subregions in a image domain Ω^2. Note that the background motion can be comprehensively regarded as $\mathbf{o}_0$. Given the consecutive stereo image pairs $\{\mathbf{I}_l^{t-1}, \mathbf{I}_r^{t-1}\}$ and $\{\mathbf{I}_l^t, \mathbf{I}_r^t\}$ at time $t-1$ and t with the subscripts denoting the left and right images, the goal is to infer $\mathbf{L}$ and $\mathbf{O}$ for all pixels in the first left reference image. In our method, the system generally includes two stages for coarse-to-fine processing while it can be concisely represented as a unified conditional random field model as Eq. 1:

$$\mathbf{E}(\mathbf{L}, \mathbf{O}) = E_D(\mathbf{L}, \mathbf{O}) + \lambda_C E_C(\mathbf{L}, \mathbf{O}). \quad (1)$$

Note that, if optical flow and depth variables are also included in the CRF model, the final optimization returns a more accurate estimation. However this consumes much more time. Actually, the optical flow and depth estimation algorithms in this work can provide a good result enough for inference at most cases, so that for efficiency trade-off, it is not considered in this paper.

A. DATA TERM

The data term mainly takes two assumptions into consideration for motion segmentation. One is that the end point in the current image from the optical flow should meet the reprojection point from the same previous image point; another is that the corresponding points should be similar in appearance. Therefore, the data term is modeled as *Motion Cue* plus *Appearance Cue*:

$$E_D = \sum_i (((1-\lambda_\alpha) \cdot \psi_i^{flow} + \lambda_\alpha \cdot \psi_i^{app}) \delta(l_i = 0) + \lambda_p \delta(l_i = 1)), \quad (2)$$

where it consists of motion and appearance constrains. The coefficients λ_α is a trade-off parameter to make the data term to be unit one in terms of smoothness term, l_i is the label of element i, $\delta(\cdot)$ is an indicator function and λ_p is a constant for penalty. Note that the basic element i has different meanings in coarse-to-fine procedures where it represents superpixel in the fisrt stage while pixel in the second stage.

1) MOTION CUE

The motion constrain can be simplely defined based on Norm-2 distance between two offsets:

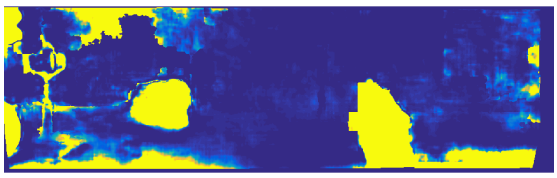
$$\psi_i^{flow} = \rho(\|f_i^{flow} - f_i^{proj}\|_2^2, \tau_{mo}), \tag{3}$$

where f_i^{flow} and f_i^{proj} represent the optical flow and the offset caused by reprojection respectively, and τ_{mo} is used as a parameter of robust trancated funtion $\rho(\cdot,\cdot)$ for normalization and robust estimation. The function is defined as Eq. 4:

$$\rho(x,\tau) = [\delta(\frac{x}{\tau} > 1) + (1 - \delta(\frac{x}{\tau} > 1)) \cdot (\frac{x}{\tau})]^p, \tag{4}$$

and the power $p = 3$ is experimentally chosen for nonlinear mapping to make the discrepancies more distinguished.

However, considering the absolute difference from both cues f_i^{flow} and f_i^{proj}, it does not refect the true relationship. Practically, this representation is often error-prone especially in the middel domain of image due to the noise of the perspective projection that makes the flow difference close to observer more accurate than that in the distance. The affect caused by this measurement noise is shown in Fig. 3. This clearly shows that with normalization it improves the performance significantly.



(a)

(b)

FIGURE 3. Comparison on affect with and without normalization. It is obviously that with normalization the motion discrepancies on vehicle can be identified significantly while there seems to be no responses when without normalization. (a) With Mahalanobis normalization. (b) Without normalization.

Therefore, in order to normalize the affect in different regions caused by the perspective projection. An approximate Mahalanobis distance is expoited and the flow difference is modified as Eq. 5:

$$\|\Delta_{f_i}\|_2^2 \Rightarrow (\Delta_{f_i})^T \Sigma^{-1} (\Delta_{f_i}), \tag{5}$$

where $\Delta_{f_i} = f_i^{flow} - f_i^{proj}$ and Σ is the covariance which can be represented as the error propagation by Eq. 6:

$$\Sigma = \mathbf{J}\mathbf{S}\mathbf{J}^T, \tag{6}$$

where $\mathbf{S}$ is the diagonal measurement noise matrix which assumes to be with Gaussian noise of 0.5 pixel. The derivation is detailed in **Appendix**.

Consequently, the modified data term for motion cue can be rewritten as Eq. 7:

$$\psi_i^{flow} = \rho((\Delta_{f_i})^T \Sigma^{-1} (\Delta_{f_i}), \tau_{mo}). \tag{7}$$

2) APPEARANCE CUE

Similarly, the appearance constrain is defined as Eq. 8:

$$\psi_i^{app} = \rho(C_i, \tau_{app}), \tag{8}$$

where C_i represents the appearance matching cost under the usual assumption that the corresponding points in consecutive images should have similar appearance. However, only assume the color cocnsistency can not provide a robust measurement due to ambiguity in many cases. Thus, some robust features are exploited such as census descriptors. In this work, in order to normalize the cost for convenience, the formulation used in [12] is introduced for robust comparison which also assumes that gradients of the rigid scene should keep consistent. Therefore, the cost function takes the form as Eq. 9:

$$C_i = (1-\alpha) \cdot \|I_i^{t-1} - I_i^t\|_2^2 + \alpha \cdot \|G_i^{t-1} - G_i^t\|_2^2, \tag{9}$$

where α is a weght factor, the matching cost is computed by warping each pixel according to the reprojection operation and I_i^{t-1} and I_i^t are the color values in consecutive images from time $t-1$ and t respectively, likewise the G_i^{t-1} and G_i^t are the corresponding gradients.

B. SMOOTHNESS TERM

The second order smoothness term forces the coherence of adjacent elements in terms of multiple cues including normals and depths as well as motions. It models the local energy of element i with its neighbors as in Eq. 10:

$$E_C = \sum_{i,j} (\gamma_n \psi_{ij}^{normal} + \gamma_d \psi_{ij}^{depth} + \gamma_m \psi_{ij}^{motion}) + \lambda_{potts} \delta(l_i \neq l_j), \tag{10}$$

Here, the three constrains show in similar way of exp nonlinear mapping for robust measurement, but they are modeled in different aspects. The last term is for potts model that controls inner consistency.

The normal constrain is detailed as:

$$\psi_{ij}^{normal} = \exp\ (-(\frac{\mathbf{deg}(i,j)}{\delta_n})^2), \tag{11}$$

where δ_n is the standard variance and the degree between neighbor normals as $\mathbf{deg}(i,j) = \arccos\ (|\mathbf{n}_i^T \mathbf{n}_j| / (\|\mathbf{n}_i\| \|\mathbf{n}_j\|))$, $\mathbf{n}$ for normal vector.

The depth constrain is shown as:

$$\psi_{ij}^{depth} = \exp(-\frac{(d_i - d_j)^2}{2}), \tag{12}$$

where d represents for depth values.

The motion constrain is modeled as:

$$\psi_{ij}^{motion} = \exp(-\frac{(\Delta_{f_{ij}})^T \Delta_{f_{ij}}}{2\delta_f^2}), \tag{13}$$

where δ_f is the standard variance, and $\Delta_{f_{ij}} = f_i^{flow} - f_j^{flow}$ is the difference of optical flow between neighbors.

C. OPTIMIZATION FOR THE PROBLEM

The optimization is a direct forward approach which minimizes the energy Eq. 1 sequentially for the label and then the object parameter sets assuming that when one is optimizing, the other remains fixed. The following discusses the two-stage procedures.

1) SUPERPIXEL LEVEL DYNAMIC-STATIC SEGMENTATION

For a typical scene, at the beginning the number of the foreground objects and their locations are unkown, which is the inevitable issue for initial dynamic and static segmentation. From optimization view, this is conducted as initial estimates and influences the ability to find the correct basin. Actually, this can be regarded as a two-class segmentation problem even though there are multiple dynamic regions with independed motions. Note that our goal is to find the final dynamic objects seperately, there is no need to consume a lot of time to get pixel-wise segmentation at this inital stage especially when there is large background region.

Therefore, the superpixel based dynamic-static segmentation is processed for efficiency, which can be called as the first stage for coarse-to-fine detection. In this work, a simple linear iterative clustering (SLIC) algorithm [26] is applied on the disparity image instead of color image to obtain the superpixels. This is because disparity image holds the depth information which can be helpful for object segmentation while the color image often leads to false segmentation because the edges of different color regions often are not identical to object boundaries.

Let $\mathbf{Sp}$ denotes the set of superpixels computed from the disparity image. Each superpixl sp_i contains N pixels, the corresponding depths and the computed plane normal. The energy function focuses on the variable $\mathbf{L}$ for two class segmentation and fixes the variable $\mathbf{O}$. Thus the data term in Eq. 7 and Eq. 8 are modified to compute the average values for superpixel i shown as:

$$\psi_i^{flow} = \frac{1}{N}\sum_{j\in sp_i}^{N}(\rho((\Delta_{f_j})^T\Sigma^{-1}(\Delta_{f_j}), \tau_{mo})), \tag{14}$$

$$\psi_i^{app} = \frac{1}{N}\sum_{j\in sp_i}^{N}\rho(C_j, \tau_{app}). \tag{15}$$

Similarly, the smoothness constrains have to be modified to calculate the energy on superpixels as well. Particularly, the normal constrain compares the normals of pairwise planar superpixels, while median depth and average optical flow are applied to compute the depth constrain and motion constrain respectively.

For inference, this is a binary classification problem which can be efficiently solved by classic graph-cut method as in [27].

2) PIXEL LEVEL MOVING OBJECT SEGMENTATION

When the first stage is done, the dynamic regions are detected and the candidate moving objects are included. However, this initial segmentation still remains some false positives due to the assumption that the scene is divided into multiple superpixels, making the boundaries not accurate and not suitable for further motion estimation. Additionally, the inaccurate estimation for boundaries also incluences the 3D reconstruction for dynamic scenes, making some ghost points in 3D mapping. Consequently, the second stage for pixel level segmentation is applied.

For preparation, a simple algorithm that merges the superpixels by their spacial distances is exploited to cluster all the superpixels segmented as dynamic into multiple subregions. Although some other region growing methods can provide better results, this proposed method can return a good proprosal. Therefore, the superpixels that belong to candidate moving objects are grouping into multiple regions and for each region, it is labeled as $\mathbf{o}_k$. Then a bounding box is calculated to warp this region for local pixel segmentation. Obviously, in this region, it not only includes the foreground of moving object but also the background of static region. Therefore, the goal in the second stage for the energy function in Eq. 1 is to find the binary segmentation for moving object detection. Contrary to the first stage, here, it focuses on the variable $\mathbf{O}$ for each object segmentation and fixes the variable $\mathbf{L}$.

In addition, to keep some helpful prior information from first stage, the data term is modified as Eq. 16:

$$E_D = \sum_{i\in R}([\beta_1((1-\lambda_\alpha)\cdot\psi_i^{flow} + \lambda_\alpha\cdot\psi_i^{app}) + \beta_2\delta(l_i^1 = 1)]\delta(l_i^2 = 0) + \lambda_p\delta(l_i^2 = 1)), \tag{16}$$

where R represents the region of the warped bounding box, and β_1, β_2 are coefficients to balance the weights between current observation and previous estimate from the first stage. Note that the superscript of variable l_i represents the stage number which 1 for first stage and 2 for the current second stage. Hence, the goal for inference in the object bounding box is to determine the label l_i^2 of each pixel $i \in R$. If it belongs to foreground, add it into the object set $\mathbf{o}_{C(i)}$ where $C(\cdot)$ is the aforementioned label assignment function. So as this discussion, it also can be regarded as a binary classification problem and solved by the same graph-cut method.

VI. COMPLEMENTARY PROCESSING

A. EGO-MOTION UPDATE

When the first stage finishes, the initial static region is segmented. Thus, the ego-motion should be updated because at the beginning, the RANSAC based ego-motion is estimated probably including dynamic regions. Therefore, it should be esitmated again on only the static regions, leading to a further distinguishable segmentation. Figure 4 illustrates this effect by showing the key points for RANSAC based method. Without the outliers, the rest points all belong to

FIGURE 4. Illustration on dynamic and static key points for RANSAC based odometry method. The red circles represent the inliers which are classified as static points while the cyan stars represent the dynamic points which are removed to update the ego-motion.

static regions which makes the visual odometry achieve a better performance.

B. POST-PROCESSING FOR OBJECT MOTION ESTIMATION

After the second stage segmentation, the moving objects are detected and a refined object region is proposed. However, some regions are outliers and then removed by some simiple geometry principles such as object size and its normal direction to the ground.

Since the purpose of optical flow is the pixel-wise motion, it often appears multiple motions on an object. While for a rigid object, it should have only one single motion. For each object, 3D rigid motion is estimated through RANSAC ICP. In addition, the background motion, also unkown as the relative pose of the camera, is estimated after removing all the dynamic regions.

VII. EXPERIMENTS AND DISCUSSIONS

A. SETTINGS

Basically, the experiments are tested on a laptop with 16GB RAM and single *Intel Core i7-7700HQ* with 2.8 GHz, and the GPU of *GTX 1050* with 4G GPU memory is used for optical flow computation. The algorithm is implemented in Matlab under Ubuntu 16.04. Through all the experiments, the expected total number of superpixels is $N = 1000$. The parameters for the algorithm are a little different in two stages where in the first stage. $\{\lambda_\alpha, \alpha, \tau_{mo}, \tau_{app}, \lambda_p, \gamma_n, \gamma_d, \gamma_m, \lambda_C, \lambda_{potts}, \delta_n, \delta_f\} = \{0.2, 0.9, 20, 1/255, 0.4, 0.2, 0.8, 0, 1000, 0.5, 10, 3\}$, while in the second stage, some of them are tuned and some are added: $\{\lambda_p, \gamma_n, \gamma_d, \gamma_m, \lambda_C, \lambda_{potts}, \beta_1, \beta_2\} = \{0.5, 0.3, 0.4, 0.3, 100, 0.05, 0.5, 0.5\}$.

B. DATASET AND BASELINE

In order to evaluate the algorithm, the famous public KITTI dataset [28] is used for two experiments, one for odometry evaluation which aims to validate the improvement on relative pose estimation by removing dynamic regions, and another with some modifications on ground truth for moving object detection which aims to evaluate the effectiveness and also the improvement compared to the baseline.

The visual odometry algorithm in [6] is utilized as the baseline denoted as **libviso2** for odometry comparison due to its efficiency and core idea typically based on RANSAC scheme. Meanwhile, the comparison for moving object detection is learnt from [7] because it is one of a few typical methods that mainly study the problem of moving object detection. Although its core algorithm also includes tracking part to improve the performance of entire system, detection is whatever fundamental to others. Thus, leaving tracking aside, its moving object detection algorithm is used as baseline denoted as **triTrack** for this task.

TABLE 1. Total average error for visual odometry on training dataset.

Method	$\mathbf{T}(\%)$	$\mathbf{R}(deg/m)$
libviso2	4.488168	0.000093
Ours	4.452304	0.000096

C. ODOMETRY COMPARISON

Table 1 shows the total average error for visual odometry on training dataset which consists of 11 sequences and about 23200 frames in total. Since **libviso2** [6] is a well studied method for visual odometry, it still remains issues on drift so that the performance cannot be improved dramatically even if the dynamic regions are removed. Additionally, the reason why they do not differ significantly is probably that the static scenes account for a large proportion of the dataset so that the influence only affects on some subsets of the sequences. Although the total average accuracy is similar to each other, the improvement can be found in multiple subsets of the long sequences.

Figure 5 show the cross comparisons on the curves of the average rotation and translation errors with respect to path length and speed. In terms of rotation errors, the discrepancies can be ignored since they are merely zeros, except that when the speed rises up, the rotation error of baseline method increases more significantly than that of our method in Fig 5(b). However, in terms of the translation, this can be illustrated in Fig. 5(c-d) that with dynamic region removal, the performance is often better than that of baseline. Particularly, it improves significantly at the spot with respect to $600m$ in Fig. 5(c) which is probably that there are more dynamic objects during that length scope so as to obtain a better result after removing the dynamic regions. When compared to the translation error with respect to speed, high speed probably brings incorrect motion estimation, making not only incorrect moving objects detection but also false responses due to motion blur. Nevertheless, the improvement in Fig. 5(d) shows very clearly that when the speed rises up, the translation error begins to blow up for both methods but relatively, our method reduces this affect dramatically.

D. MOVING OBJECT DETECTION

Different to objects detection task, this proposed method aims at moving objects while the stationary objects are regarded as negatives. Actually, for moving objects detection, there is no available dataset for this task specifically.

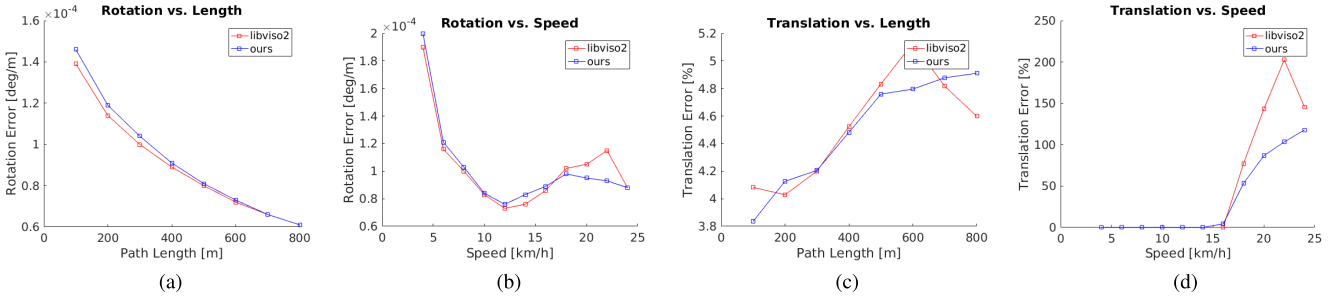


FIGURE 5. Comparisons on total average errors for visual odometry performance.

FIGURE 6. Typical qualitative results. The green bounding boxes are the 2D projections from ground truth 3D tracklets, the red ones are the detection results from baseline **triTrack**. Our method returns not only the bounding region in light cyan but also the object segmentation within it in relative darker colors. The first row of each subfigure shows the result of our method with Mahalanobis normalization while the second row without it.

Here, in this paper, a subset of $1-70$ frames of sequence ***2011_09_26_drive_0005_sync*** is selected for experiment because there are almost all objects are moving and the tracklets are available.

However, the original ground truth bounding boxes include both kinds of objects so that some modifications are made. The ones for stationay are ignored as well as the ones denoted as occlusive. Note that, from the qualitative results in Fig. 6, since the 2D bounding boxes of ground truth are from the projected corresponding 3D tracklets, the area is somehow larger than expected. All these bring some bias anyhow. Overall, for this settings, the moving objects detection is regarded as normal object detection task and the similar criteria is used that successful detection is made if the overlap of the detected bounding box with that of ground truth is over 50%. The criteria for average recall and precision take the forms

TABLE 2. Total average performance for moving object detection.

Method	Avg. Recall (%)	Avg. Precision (%)
triTrack	0.4030	0.2634
ours(-norm)	0.4776	0.6400
Ours(+norm)	0.6493	0.6214

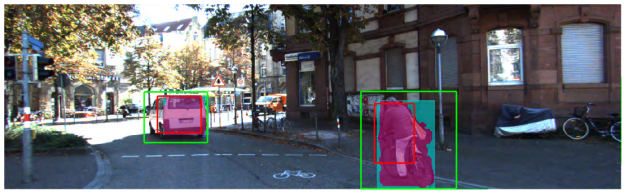
as Eq. 17:

$$\begin{cases} Recall = \dfrac{tp}{tp+fn} \\ Precision = \dfrac{tp}{tp+fp}, \end{cases} \tag{17}$$

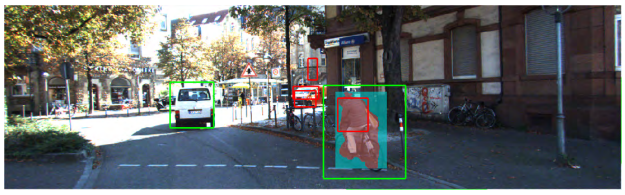
where tp is for true positives, fn is for false negatives and fp is for false positives.

Table 2 shows the total average performance compared among three methods where $+$**norm** and $-$**norm** mean that the method with and without Mahalanobis normalization respectively. From the recall, it shows that without Mahalanobis normalization, the false negatives are significantly more than that with normalization. Nevertheless, for our proposed method, it improves about 20% on recall and 35% on precision compared to the baseline.

Figure 6 shows some typical qualitative results. Among them, it shows the general performance on moving object detection and segmentation. In normal cases that no region is occlusive such as depicted in Fig. 6(a)(b), the detections are well achieved while the baseline **triTrack** method returns inaccurate bounding boxes. In hard cases like shown in Fig. 6(e)(f), the occlusion happens. Although the ground truth ignores the moving vehicle, our method still detects it and well segmented. Compared to our method without Mahalanobis normalization, the motion area is more discriminative, leading to a better detection for object region.



(a)



(b)

FIGURE 7. Failure cases. (a) Under segmentation on vehicle. (b) Missing target.

However, our method also suffers some restrictions, causing failure cases. In Fig. 7, it shows two typical situations where in Fig. 7(a), the final result only detects part of the moving vehicle. This missing part is caused probably by the different illumination to the backlight side so that it makes the motion similar to the background. In Fig. 7(b), the vehicle moves straightforward near the optical axis so that there is no motion discrepancies there as if the vehicle is stationary.

VIII. CONCLUSION

This paper proposes an efficient method for coarse-to-fine moving object detection in rigid scene for autonomous driving. It is right for dynamic scenes with rigidly moving objects and just complementary to a set of classic visual odometry methods which are designed originally for static scenes. The proposed method is basically divided into two stages. In the first stage, a superpixel based binary segmentation is applied for initial dynamic region detection. Further in the second stage, a pixel level segmentation is used for refinement. Additionally, in order to normalize the noise affected by depth estimation, an approximate Mahalanobis normalization is deduced. Although there is no specific dataset for this moving object detection task, the modifications are made and the experiments are taken to show the effectiveness compared to two baseline methods for odometry and moving object detection respectively. The results show that the proposed method improves the performance and has the ability to build a robust map in not only classic static scenes but also dynamic scenes.

In the future, with the ability of modeling for dynamic scenes, more advanced applications can be done such as multi-frame obstacle detection like [29] and 3D mapping with dynamic objects.

APPENDIX

To compute the discrepancy of optical flow and the reprojection offset is equal to compute the distance of the end points in the target image.

Note that $\Delta_f = f^{flow} - f^{proj}$ is modeled for motion consistency assumption. It can be decomposed as Eq. 18:

$$\begin{aligned} \Delta_f &= f^{flow} - f^{proj} \\ &= x_2^f - x_1^f - (x_2^t - x_1^t), \end{aligned} \tag{18}$$

where $[x_1^i, x_2^i]^T, i \in \{f, t\}$ represents the corresponding image positions in sequential times. The right superscripts represent the categories of flow f and reprojection offset t respectively. However, this two kinds of position differences share the same initial image position, i.e. $x_1^f = x_1^t$ and the formula is derived as Eq. 19:

$$\begin{aligned} \Delta_f &= x_2^f - x_2^t \\ &= [u^f, v^f, 1]^T - \mathbf{K} \cdot \mathbf{T} \cdot \mathbf{K}^{-1} \cdot [u^t, v^t, 1]^T, \end{aligned} \tag{19}$$

where $[u^i, v^i, 1]^T, i \in \{f, t\}$ represents for homogeneous image coordinates. $\mathbf{K}$ and $\mathbf{T}$ are camera intrisic matrix and transformation matrix respectively which are define as Eq. 20

and Eq. 21.

$$\mathbf{K} = \begin{bmatrix} f_o & 0 & u_0 \\ 0 & f_o & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{20}$$

where f_o here presents the focal length, $[u_0, v_0]$ is the principle point coordinate, and

$$\mathbf{T} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_1 \\ R_{21} & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \end{bmatrix}. \tag{21}$$

where all $R_{i,j}, t_i, \{i, j\} \in \{1, 2, 3\}$ are the elements of the rotation matrix and translation vector respectively.

Here, some auxilary equations are defined as Eq. 22:

$$\begin{cases} f_1(u, v) = f_o \cdot (R_{11} \cdot X + R_{12} \cdot Y + R_{13} \cdot Z + t_1) \\ f_2(u, v) = f_o \cdot (R_{21} \cdot X + R_{22} \cdot Y + R_{23} \cdot Z + t_2) \\ h(u, v) = R_{31} \cdot X + R_{32} \cdot Y + R_{33} \cdot Z + t_3, \end{cases} \tag{22}$$

where $[X, Y, Z]^T$ is the 3D space coordinate of corresponding image point $[u, v]^T$, which is shown as Eq. 23:

$$\begin{cases} X = Z \cdot \dfrac{u - u_0}{f_o} \\ Y = Z \cdot \dfrac{v - v_0}{f_o}. \end{cases} \tag{23}$$

Therefore, substituting these equations into Eq. 19, the motion discrepancy is finally derived as Eq. 24:

$$\Delta_f = [\Delta_f^1, \Delta_f^2]^T = \begin{bmatrix} u^f - \dfrac{f_o \cdot f_1(u^t, v^t)}{h(u^t, v^t)} - u_0 \\ v^f - \dfrac{f_o \cdot f_2(u^t, v^t)}{h(u^t, v^t)} - v_0 \end{bmatrix}. \tag{24}$$

In order to achieve covariance matrix for normalization, the Jacobian matrix of this Δ_f with respect to image coordinate is required. However, to simplify the derivation for optical flow with respect to image coordinate, it assumes to be weakly related between them because this is not the dominent influence compared to the disparity measurement. Therefore, assume u^f and v^f to be constant and remain u^t and v^t with the superscripts omitted for simplicity. Thus, the Jacobian matrix is approximated as Eq. 25:

$$J = \frac{\partial \Delta_f}{\partial uv} = \begin{bmatrix} \dfrac{\partial \Delta_f^1}{\partial u} & \dfrac{\partial \Delta_f^1}{\partial v} \\ \dfrac{\partial \Delta_f^2}{\partial u} & \dfrac{\partial \Delta_f^2}{\partial v} \end{bmatrix}, \tag{25}$$

where the derivatives are:

$$\begin{cases} \dfrac{\partial \Delta_f^1}{\partial u} = -\dfrac{[R_{11} \cdot Z \cdot h(u, v) - f_1(u, v) \cdot R_{31} \cdot Z]}{h^2(u, v)} \\ \dfrac{\partial \Delta_f^1}{\partial v} = -\dfrac{[R_{12} \cdot Z \cdot h(u, v) - f_1(u, v) \cdot R_{32} \cdot Z]}{h^2(u, v)} \\ \dfrac{\partial \Delta_f^2}{\partial u} = -\dfrac{[R_{21} \cdot Z \cdot h(u, v) - f_2(u, v) \cdot R_{31} \cdot Z]}{h^2(u, v)} \\ \dfrac{\partial \Delta_f^2}{\partial v} = -\dfrac{[R_{22} \cdot Z \cdot h(u, v) - f_2(u, v) \cdot R_{32} \cdot Z]}{h^2(u, v)}. \end{cases} \tag{26}$$

When the Jacobian matrix is obtained, the covariance is computed as:

$$\Sigma = \mathbf{JSJ}^T,$$

where $\mathbf{S}$ is the diagonal measurement noise matrix which assumes to be with Gaussian noise of 0.5 pixel as:

$$\mathbf{S} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}. \tag{27}$$

As discussed above, this simplified processing here is called as approximate Mahalanobis normalization.

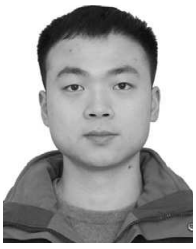
ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
[2] R. A. Newcombe *et al.*, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Real.*, Oct. 2011, pp. 127–136.
[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[4] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality (ISMAR)*, Nara, Japan, Nov. 2007, pp. 225–234.
[5] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sydney, NSW, Australia, Dec. 2013, pp. 1449–1456.
[6] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3D reconstruction in real-time," in *Proc. IEEE Intell. Veh. Symp.*, Baden-Baden, Germany, Jun. 2011, pp. 963–968.
[7] P. Lenz, J. Ziegler, A. Geiger, and M. Roser, "Sparse scene flow segmentation for moving object detection in urban environments," in *Proc. IEEE Intell. Veh. Symp.*, Baden-Baden, Germany, Jun. 2011, pp. 926–932.
[8] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *Proc. IEEE 12th Int. Conf. Comput. Vis. (ICCV)*, Kyoto, Japan, Sep. 2009, pp. 670–677.
[9] C. Wojek and B. Schiele, "A dynamic conditional random field model for joint labeling of object and scene classes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Marseille, France, Oct. 2008, pp. 733–747.
[10] C. Vogel, K. Schindler, and S. Roth, "3D scene flow estimation with a piecewise rigid scene model," *Int. J. Comput. Vis.*, vol. 115, no. 1, pp. 1–28, 2015.
[11] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 3061–3070.
[12] J. Yang and H. Li, "Dense, accurate optical flow estimation with piecewise parametric model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1019–1027.
[13] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, "Fast approximate energy minimization with label costs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Francisco, CA, USA, Jun. 2010, pp. 2173–2180.
[14] H. Li, "Two-view motion segmentation from linear programming relaxation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, Jun. 2007, pp. 1–8.
[15] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black, "Optical flow with semantic segmentation and localized layers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 3889–3898.
[16] D. Sun, J. Wulff, E. B. Sudderth, H. Pfister, and M. J. Black, "A fully-connected layered model of foreground and background flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 2451–2458.

[17] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1187–1200, Jun. 2014.
[18] S. R. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," in *Proc. 21st IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, Jun. 2008, pp. 1–8.
[19] C. Jiang, D. P. Paudel, Y. Fougerolle, D. Fofi, and C. Demonceaux, "Static-map and dynamic object reconstruction in outdoor scenes using 3-D motion segmentation," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 324–331, Jan. 2016.
[20] A. Dib and F. Charpillet, "Robust dense visual odometry for RGB-D cameras in a dynamic environment," in *Proc. Int. Conf. Adv. Robot. (ICAR)*, Istanbul, Turkey, Jul. 2015, pp. 1–7.
[21] P. F. Alcantarilla, J. J. Yebes, J. Almazn, and L. M. Bergasa, "On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 1290–1297.
[22] D. Sun, S. Roth, and M. J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *Int. J. Comput. Vis.*, vol. 106, no. 2, pp. 115–137, 2014.
[23] M. Menze, C. Heipke, and A. Geiger, "Discrete optimization for optical flow," in *Proc. German Conf. Pattern Recognit. (GCPR)*, 2015, pp. 16–28.
[24] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1–3. [Online]. Available: http://lmb.informatik.uni-freiburg.de//Publications/2017/IMKDB17
[25] K. Yamaguchi, D. Mcallester, and R. Urtasun, "Efficient joint segmentation, occlusion labeling, stereo and flow estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 756–771.
[26] 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.
[27] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
[28] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. Int. Conf. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[29] Z. Xiao *et al.*, "Gaussian process regression-based robust free space detection for autonomous vehicle by 3-D point cloud and 2-D appearance information fusion," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 4, p. 1729881417717058, 2017.



ZHIPENG XIAO received the B.S. degree in automation from Xi'an Jiaotong University, in 2011, and the M.S. degree in control science and engineering from the College of Mechatronic Engineering and Automation, National University of Defense Technology, in 2013, where he is currently pursuing the Ph.D. degree. He was a Visiting Ph.D. Student with The Australian National University for joint training. His research interests include computer vision, pattern recognition, and intelligent vehicles.

BIN DAI received the Ph.D. degree in control science and engineering from the National University of Defense Technology, Changsha, China, in 1998. He was a Visiting Scholar with the Intelligent Process Control and Robotics Laboratory, Karlsruhe Institute of Technology, in 2006. He is currently a Professor with the Institute of Unmanned Systems, National University of Defense Technology. His research interests include pattern recognition, computer vision, and intelligent vehicles.

TAO WU received the Ph.D. degree in control science and engineering from the National University of Defense Technology, Changsha, China, in 2004. He is currently an Associate Professor with the Institute of Unmanned Systems, National University of Defense Technology. His research interests include pattern recognition, computer vision, and intelligent vehicles.

LIANG XIAO received the B.S. degree in automation and the M.S. degree in control science and engineering from the College of Mechatronic Engineering and Automation, National University of Defense Technology, in 2010 and 2012, respectively, where he is currently pursuing the Ph.D. degree. His research interests include computer vision, pattern recognition, and intelligent vehicles.

TONGTONG CHEN received the Ph.D. degree in control science and engineering from the National University of Defense Technology, Changsha, China, in 2016. He is currently with the Beijing Special Engineering Design Institute, Beijing, China. His research interests include 3-D point cloud processing, 3-D object recognition, and laser-based vehicle detection and tracking.