Incomplete 3D Motion Trajectory Segmentation and 2D-to-3D Label Transfer for Dynamic Scene Analysis

Cansen Jiang, Danda Pani Paudel, Yohan Fougerolle, David Fofi, Cédric Demonceaux

Incomplete 3D Motion Trajectory Segmentation and 2D-to-3D Label Transfer for Dynamic Scene Analysis

Cansen Jiang[1], Danda Pani Paudel[2], Yohan Fougerolle[1], David Fofi[1] and Cédric Demonceaux[1]

***Abstract*— The knowledge of the static scene parts and the moving objects in a dynamic scene plays a vital role for scene modelling, understanding, and landmark-based robot navigation. The key information for these tasks lies on semantic labels of the scene parts and the motion trajectories of the dynamic objects. In this work, we propose a method that segments the 3D feature trajectories based on their motion behaviours, and assigns them semantic labels using 2D-to-3D label transfer. These feature trajectories are constructed by using the proposed trajectory recovery algorithm which takes the loss of feature tracking into account. We introduce a complete framework for static-map and dynamic objects' reconstruction, as well as semantic scene understanding for a calibrated and moving 2D-3D camera setup. Our motion segmentation approach is faster by two orders of magnitude, while performing better than the state-of-the-art 3D motion segmentation methods, and successfully handles the previously discarded incomplete trajectory scenarios.**

I. Introduction

The emergence of affordable 2D and 3D cameras allows us to capture both 2D image and 3D point cloud data for a wide range of computer vision and robotics applications, such as large-scale city modelling [1] and autonomous driving [2]. The reliability of these applications mainly depends on the accuracy of camera localization and scene understanding.

In literature, precise camera localization is achieved using 2D images [3] and 3D point clouds [4] serving for the 3D scene reconstructions. Such methods make the assumption of mostly static environments where only few moving objects exist. For dynamic scenes, these methods suffer from many difficulties in detecting and removing the dynamic objects, followed by the localization accuracy degradation [5]. Therefore, it is highly desirable to detect the moving objects prior to the camera motion estimation [6]. Besides, the detection of dynamic objects, along their trajectories, is a fundamental task for scene understanding and object behaviour analysis [8], [10]–[12].

Moving object detection, prior to camera localization, can be performed by segmenting the features' motion trajectories. However, most segmentation methods either assume the static camera [12] and/or work on 2D image space [8]. When moving 2D-3D camera setups are given, *e.g.* RGB camera and 3D laser scanner equipped robots, these methods turn out to be inadequate due to such simplistic assumptions. In our previous work [6], we proposed a method that performs motion segmentation directly on the 3D motion trajectories. Despite of the appealing framework for dynamic scene reconstruction, [6] suffers from three major limitations: (a) high computational time, (b) inability to handle incomplete feature trajectories, (c) lack of semantic labels for the scene understanding. In this paper, we address these three limitations with improved accuracy.

Although studies in Static Scene Modelling [1], [4], Dynamic Object Analysis [10]–[13] and Semantic Scene Understanding [14]–[16] have achieved adequate results independently, there exist very limited works in literature that systematically address these problems altogether. Therefore, this work offers a general scene understanding and modelling framework of dynamic environments using a moving 2D-3D camera platform. As illustrated in Fig. 1, the proposed system fulfils the functionalities of: moving object detection and segmentation, static-map and dynamic object reconstruction, and semantic understanding of scenes.

For a moving camera setup, the detection and the tracking of moving objects are very challenging due to the changes of scene background, the objects' independent motions, and the variation in the number of moving objects. Therefore, it is recommended to distinguish the dynamic and static scene parts based on their motion trajectories. In such cases, the objects that reciprocate the camera motion are considered to be static. Elhamifar and Vidal [8] demonstrate that distinctive motions can be identified by analysing their image feature trajectories as a motion subspaces segmentation problem. Considering each independent motion as a subspace, a feature trajectory can be approximated as a linear combination of other feature trajectories from the same subspace, so-called subspace self-expressive property. When 2D-3D camera setup is given, [6] argue that motion segmentation (MS) directly based on 3D space gives better performance.

Inspired by [6], [8], [9], we propose a more efficient and effective subspace clustering approach, called 3D-based SMooth Representation Clustering (3D-SMR). Different from [9], our method performs the MS in 3D space using raw motion trajectories, with spatial regularization constraints (linear and angular motion consistency energy) to improve the trajectory clustering performances. Alongside with the subspace self-expressive property, the 3D-SMR algorithm intends to separate the motion subspaces by enforcing the grouping effects as well as the motion consistency of their subspaces, and is detailed in Section III.

In many practical scenarios, many feature trajectories can be incomplete (broken) due to the loss of tracking. To overcome this issue, we present a novel feature trajectory construction approach that jointly benefits from feature track-

[1] Le2i, FRE CNRS 2005, Arts et Métiers, Université Bourgogne Franche-Comté, France. (e-mail: firstname.lastname@u-bourgogne.fr)

[2]Computer Vision Laboratory, ETH Zurich, Switzerland. (e-mail: paudel@vision.ee.ethz.ch)

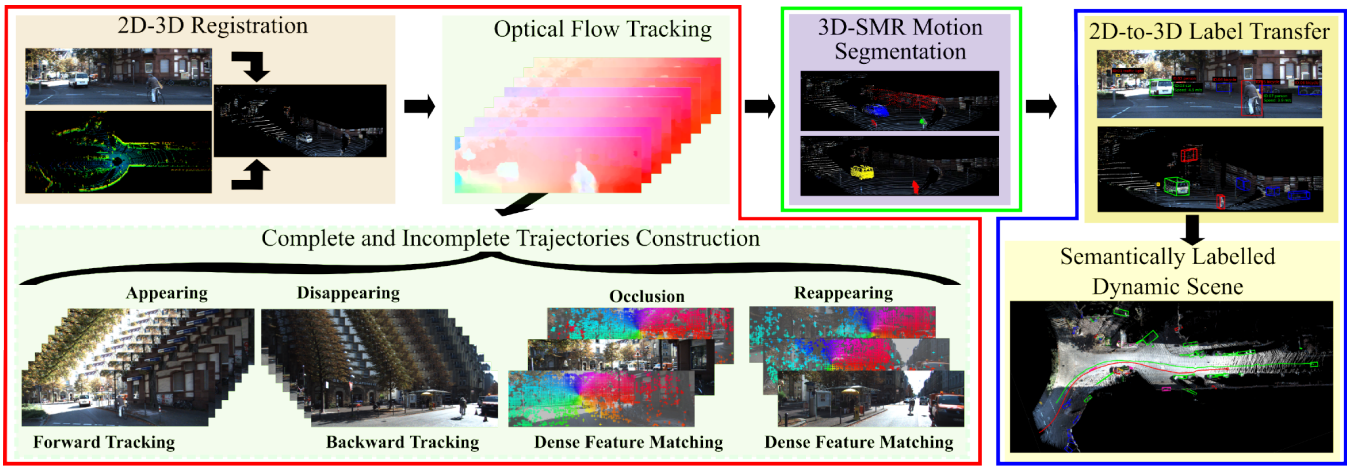


Fig. 1: Dynamic scene analysis pipeline. Red block shows the incomplete feature trajectory construction supported by forward and backward feature tracking and matching approach, and is detailed in Section IV. Green block depicts the moving object detection using motion segmentation on 3D feature trajectories, and is detailed in Section III. Blue block illustrates the 2D-to-3D label transfer for automatic semantic labelling of a dynamic scene, and is detailed in Section V and VI.

ing and matching techniques, and is detailed in Section IV.

To associate the semantic labels with the segmented motion trajectories, we transfer the object labels obtained from the corresponding images. The object labels on the images are obtained by using the deep learning-based Yolo [16] detector. Thanks to the recent advancement in deep learning techniques, it is now possible to obtain faithful semantic labels using image information. The transfer of these labels is carried out by the max-pooling over multiple detections. We argue that the semantic understanding of dynamic 3D scenes has obtained very little attention in literature. In this context, Geiger *et al.* [15] propose a 3D traffic scene understanding framework which predicts the motions of vehicle tracklets by fusing semantic (Sky, Road, and Traffic Lane) and 3D scene flow information. Different from [15], our method results in the motion trajectories of generic objects alongside with their labels (*e.g.* pedestrian, cyclist, car), also the labels of static parts (*e.g.* traffic lights).

To summarize, the major contributions of this work are:
– We propose an efficient algorithm for 3D motion segmentation that enforces the motion consistency constraint within the subspace. The proposed algorithm is faster by two orders of magnitude, which also outperforms the state-of-the-art 3D motion segmentation methods.
– We present a simple but effective method for incomplete trajectory construction, serving for motion segmentation, to handle the practical problem of lost tracking.
– We introduce a dynamic scene understanding framework for simultaneous dynamic object extraction, static-map reconstruction, and objects' semantic labels assignment for the data acquired by a 2D-3D camera platform moving in complex real-world environments.

II. Related Work

Among the numerous works on image-based motion segmentation (MS) for moving object detection, representative approaches such as Generalized Principal Component Analysis [17], RANSAC-based MS [18], and Agglomerative Subspace Clustering (ASC) [19] are intensively studied in [20]. Although these methods provide great insight on the MS problem, they are sensitive to noise/outliers or computationally expensive. ASC, a more robust method, combines the techniques of lossy compression, rank minimization, and sparse representation. A direct inspiration of ASC led Elhamifar and Vidal [8] to develop the Sparse Subspace Clustering (SSC) algorithm –a leading MS technique in present days– which relies on self-expressive sparse representation property of the data. Taking the advantage of self-expressive property, Hu *et al.* [9] propose a Smooth Representation Clustering model by enforcing the Grouping Effects of the subspaces, which achieves the best accuracy and efficiency in literatures. Apart from 2D-based MS, inspired by image feature based SSC, Jiang *et al.* [6] propose a 3D-based Sparse Subspace Clustering (3D-SSC) which achieves significantly better performances with a slightly higher computational complexity comparing to its 2D counterpart. Stuckler *et al.* [21] perform dense 3D MS using nearly static RGB-D cameras using Gaussian Mixture Models, which is not applicable for fast moving camera setups. Differently, Sofer *et al.* [22] address the 3D MS problem by using Active Machine Learning algorithm, while application specific training data are required.

In the context of localization and mapping, Wang *et al.* [23] propose simultaneous object tracking and map building method (SLAM-MOT), using either their map prior or the motion consistency assumption. However, for slow motions and temporally stationary objects, these assumptions are not valid. Pomerleau *et al.* [24] address the SLAM-MOT problem by using ray-tracing techniques, assuming that the dynamic parts have only a small scene coverage. Ambrus *et al.* [25] propose to identify the dynamic scene parts based on the difference between the observations and the reference

model. Yet, a clean reference model is required, which is impractical for unknown environments.

In 3D scene understanding, Geiger *et al.* [15] propose to predict vehicle motions using simple semantic information which is quite insufficient for human-interactive scene understanding. Kochanov *et al.* [26] build semantic maps by propagating the scene flow of the map occupancy and semantic belief. Menze *et al.* [27] detect the moving objects using object scene flow analysis. Such methods are bounded by the imprecise depth estimate and incapable to analyse the object motion behaviours.

III. 3D Motion Segmentation

For a set of feature trajectories of multiple moving objects, the motion segmentation (MS) aims to group the feature trajectories into their corresponding motions by comparing the similarities between the feature trajectories. Recent studies focus on the subspace clustering-based methods [6], [8], [9]. The principle of these methods is to construct the affinity matrix which encodes the similarity between the feature trajectories, followed by a spectral clustering algorithm to group the trajectories into their corresponding motion subspaces. The problem of MS can be framed as a minimization problem with different regularization terms. As a constrained optimization problem, the algorithm performances are affected by different regularization constraints. The following contents introduce the general MS model and discuss the proposed regularization constraints to improve the MS performances.

A. Background and Notations

Let $\mathsf{X}^n = [\mathsf{X}_1, \cdots, \mathsf{X}_F]^\mathsf{T}$ be a vectorized 3D feature trajectory of F frames, where $\mathsf{X}_i = [x, y, z] \in \mathbb{R}^3$ is a 3D feature at frame i. Let $\mathbf{X} = \{\mathsf{X}^n\}_{n=1,\cdots,P}$ be the assembly of P feature trajectories belonging to different motions. The general self-representation model of MS problem can be defined as:

$$\min_{\mathbf{Z}} \quad \|\mathbf{X} - D(\mathbf{X})\mathbf{Z}\|_l + \Omega(\mathbf{X}, \mathbf{Z}), \quad s.t. \quad \mathbf{Z} \in \mathcal{C}, \tag{1}$$

where $D(\mathbf{X})$ is the dictionary learned from $\mathbf{X}$, and $\|\cdot\|_l$ denotes the proper norm. $\Omega(\mathbf{X}, \mathbf{Z})$ is the regularization term and $\mathcal{C}$ is the constraint set on $\mathbf{Z}$. By solving Eq. (1), a desired self-representation matrix $\mathbf{Z}^*$ is obtained to construct the affinity matrix. For instance, in [8], $D(\mathbf{X}) = \mathbf{X}$, and $l_0 - norm$ is applied to constrain the sparsity of $\mathbf{Z}^*$. Here, constraint set $\mathcal{C} = \{\mathbf{Z} | \mathsf{Z}_{ii} = 0\}$ is necessary to avoid the trivial solution, so that X_i cannot be used to represent X_i itself. Regularizer $\Omega(\mathbf{X}, \mathbf{Z})$ is set to be zero. Once the sparse representation matrix $\mathbf{Z}^*$ is obtained, a symmetric affinity matrix $A = |\mathbf{Z}^*| + |\mathbf{Z}^*|^\mathsf{T}$ is used to perform spectral clustering to separate the subspaces.

B. 3D based Smooth Representation Model

Inspired by [9], a 3D-based MS algorithm using SMooth Representation clustering (3D-SMR) is proposed. Under the framework of self-expressive subspace clustering, the 3D-SMR enforces the Grouping Effect (GE) to facilitate the feature trajectory clustering problem. Unlike [9], our GE constraint describes the feature trajectories' closeness (distances) in the 3D Euclidean space. Doing so, 3D-SMR avoids the perspective effects that appear on the image measurements. The GE constraint is enforced as a regularization term:

$$\begin{aligned} \Omega(\mathbf{X}, \mathbf{Z}) &= \frac{1}{2} \sum_{i=1}^{P} \sum_{j=1}^{P} w_{ij} \|\mathsf{Z}_i - \mathsf{Z}_j\|_2^2 \\ &= \mathrm{tr}(\mathbf{Z}\boldsymbol{L}\mathbf{Z}^\mathsf{T}), \end{aligned} \tag{2}$$

where $\mathbf{Z} = [\mathsf{Z}_i, \cdots, \mathsf{Z}_P]$ is the $P \times P$ square-sized self-representation matrix. $W = (w_{ij})$ with $w_{ij} = \|\mathsf{X}_i - \mathsf{X}_j\|_2^2$ is the weight matrix defined by the spatial closeness of feature trajectories. $L = D - W$ is the Laplacian matrix, in which D is the diagonal matrix defined as $D_{ii} = \sum_{i=1}^{P} w_{ij}$. To construct the weight matrix W, a $0-1$ weighted k Nearest Neighbour (kNN) graph is used. Combining Eq. (1) and Eq. (2), the 3D-SMR model is obtained:

$$\min_{\mathbf{Z}} \quad \|\mathbf{X} - \mathbf{X}\mathbf{Z}\|_\mathsf{F}^2 + \mathrm{tr}(\mathbf{Z}\boldsymbol{L}\mathbf{Z}^\mathsf{T}), \tag{3}$$

where $\|\cdot\|_F^2$ denotes the square of Frobenius norm.

C. Motion Consistency Constraints

On top of the GE constraint on the spatial closeness of feature trajectories, we also exploit the motion consistency. We make the assumption that, for a short video sequence, the observed motion trajectories are smooth. In other words, the motion velocities and directions are locally consistent.

Let $\mathbf{V} = \{\mathsf{V}_i\}_{i=1,\cdots,P}$ and $\theta = \{\vartheta_i\}_{i=1,\cdots,P}$ be the motion velocities and directions of feature trajectories, respectively. To enforce the motion consistency constraint, we define a combined regularization term as:

$$\begin{aligned} \Omega(\mathbf{X}, \mathbf{V}, \theta, \mathbf{Z}) &= \frac{1}{2} \sum_{i=1}^{P} \sum_{j=1}^{P} \widetilde{w}_{ij} \|\mathsf{Z}_i - \mathsf{Z}_j\|_2^2 \\ &= \mathrm{tr}(\mathbf{Z}\widetilde{\boldsymbol{L}}\mathbf{Z}^\mathsf{T}), \end{aligned} \tag{4}$$

where $\widetilde{w}_{ij} = \mathcal{E}(\mathsf{X}_i, \mathsf{X}_j) + \varphi(\mathsf{V}_i, \mathsf{V}_j) + \psi(\vartheta_i, \vartheta_j)$, and $\widetilde{L} = D - \widetilde{W}$. Recall Eq. (2), the weight component $\mathcal{E}(\mathsf{X}_i, \mathsf{X}_j) = \|\mathsf{X}_i - \mathsf{X}_j\|_2^2$ describes the spatial closeness of the feature trajectories. $\varphi(\mathsf{V}_i, \mathsf{V}_j) = \alpha\|\bar{v_i} - \bar{v_j}\|_2^2$ measures the consistency of the motion velocity, where $\bar{v_i}$ and $\bar{v_j}$ are the median speeds of the feature trajectories V_i and V_j in 3D space. $\psi(\vartheta_i, \vartheta_j) = \beta\mathrm{atan2}(\vartheta_i \times \vartheta_j, \vartheta_i \cdot \vartheta_j)$ computes the directional difference between the feature trajectories, where $\mathrm{atan2}(\cdot)$ function calculates the angle between the motion vectors ϑ_i and ϑ_j with the appropriate quadrant. α and β are the constant values controlling the weights of the regularization terms. In our experiments, both α and β are set to be 1.5, which implies that the motion consistency regularization terms have higher weight than the spatial closeness term.

The Laplacian matrix in Eq. (4) can be written as: $\widetilde{L} = \widetilde{D} - \widetilde{W}$, where $\widetilde{D}_{ii} = \sum_{i=1}^{P} \widetilde{w}_{ij}$ and the weight function $\widetilde{W} = (\widetilde{w_{ij}})$. Replacing the regularization term of Eq. (3) with Eq. (4), a more practical 3D-SMR model is proposed as:

$$\min_{\mathbf{Z}} \quad \alpha\|\mathbf{X} - \mathbf{X}\mathbf{Z}\|_\mathsf{F}^2 + \mathrm{tr}(\mathbf{Z}\widetilde{\boldsymbol{L}}\mathbf{Z}^\mathsf{T}). \tag{5}$$

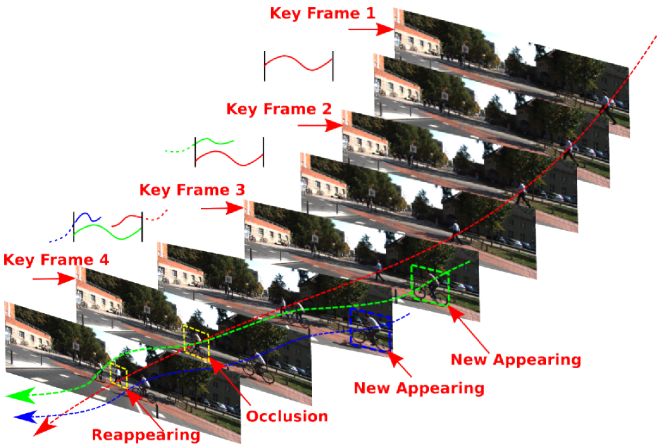


Fig. 2: Incomplete feature trajectories construction: the red, green and blue dashed lines represent the trajectories of the pedestrian and two cyclists, respectively. The green and blue rectangles highlight the appearing of the two cyclists, while the yellow rectangles highlight the pedestrian being occluded and reappearing. The solid lines (on top) represent the feature trajectories within two key frames, while the connected dashed lines are the forward or backward extended trajectories.

Since solving Eq. (5) is a smooth and convex problem, the desired optimal solution $\mathbf{Z}^*$ can be obtained by taking the first order derivative, such that:

$$\mathbf{X}^\mathsf{T}\mathbf{X}\mathbf{Z}^* + \mathbf{Z}^* L = \mathbf{X}^\mathsf{T}\mathbf{X}. \tag{6}$$

Equation (6) is a Sylvester equation having a unique solution which can be solved efficiently by the Bartels-Stewart algorithm [28] with computational complexity of $\mathcal{O}(n^3)$.

Following [8] and [9], we employ two different affinity matrices which are defined as $\mathcal{W}_1 = |\mathbf{Z}^*| + |\mathbf{Z}^*|^\mathsf{T}$ and $\mathcal{W}_2 = \left(\left|\frac{\mathsf{Z}_i^{*\mathsf{T}}\mathsf{Z}_j^*}{\|\mathsf{X}_i\|_2\|\mathsf{X}_j\|_2}\right|^\gamma\right)$, respectively. $\gamma > 0$ is a scale factor to control the affinity variances. Finally, a spectral clustering algorithm is applied to the affinity matrices $\mathcal{W}_1$ and $\mathcal{W}_2$ to segment the feature trajectories into their corresponding motions.

IV. Feature Trajectory Construction

Prior to motion segmentation, the feature trajectories are acquired by feature tracking across multiple consecutive frames. We use both 2D and 3D measurements to construct the feature trajectories in 3D space. For a calibrated 2D-3D camera setup, the 3D scene points are projected onto the 2D image to establish the 2D-to-3D correspondences, similar to [6]. These projections are considered as 2D feature points and tracked across the sequence using a dense optical flow method. To cover a wide range of speeds, a large displacement dense optical flow [29] tracking algorithm has been adopted. To reject the incorrectly tracked features, we utilize the forward and backward validation of optical flow tracking, similar to [30]. The 3D feature trajectories are then retrieved thanks to the 2D-to-3D correspondences.

In practice, the feature trajectories can be categorized into two sets: the complete and the incomplete trajectories. We define a trajectory to be complete if its feature is detected and tracked throughout the frames of interest (*i.e.* between two key frames), whereas, an incomplete trajectory is only partially detected and tracked between two key frames. The incomplete trajectories mainly come from the failure of feature tracking due to occlusions or object disappearances. As a remark, [6] and [9] simply discard such incomplete feature trajectories leaving some potential moving objects undiscovered. To address this problem, We propose the following simple but effective incomplete feature trajectory completion approach.

Recall that $\mathsf{X}^n = [\mathsf{X}_1, \cdots, \mathsf{X}_F]^\mathsf{T}$ with $\mathsf{X}_i = [x, y, z] \in \mathbb{R}^3$ is a complete 3D feature trajectory vector of F frames, and $\mathbf{X} = \{\mathsf{X}^n\}_{n=1,\cdots,P}$ is the combination of P complete feature trajectories. Denote $\hat{\mathsf{X}}^n = [\mathsf{X}_1, \cdots, \mathsf{X}_{\hat{F}}]^\mathsf{T}$ as an incomplete feature trajectory of $\hat{F}$ frames ($\hat{F} < F$), and $\hat{\mathbf{X}} = \{\hat{\mathsf{X}}^n\}_{n=1,\cdots,\hat{P}}$ as the collection of $\hat{P}$ incomplete feature trajectories. To perform motion segmentation on both complete $\mathbf{X}$ and incomplete $\hat{\mathbf{X}}$ trajectory sets simultaneously, the incomplete trajectories should be extended so that the length of $\hat{\mathsf{X}}^n$ is $3F$ (same as X^n), and the size of $\hat{\mathbf{X}}$ is $3F \times \hat{P}$. In other words, the row dimensions (trajectory length) of $\mathbf{X}$ and $\hat{\mathbf{X}}$ must be the same, while their column dimensions (feature numbers) are unconstrained.

We divide the incomplete feature trajectories into four different categories: new object appearance ($+$), tracked object disappearance ($-$), object going under occlusion (o), and previous object reappearance ($++$) as follows:

- **Newly appearing objects** are detected if new features are tracked through a minimum number of required frames for motion analysis.
- **Disappearing tracked objects** are detected using a feature tracking failure detection method [30].
- **Objects under occlusion** refer to a partial occlusion, where the object's features have both complete and incomplete trajectories.
- **Reappearing objects** are detected using the Deepmatching [31] between the features in key frames.

If a feature is not tracked throughout two key frames, a forward or backward tracking is activated to extend the feature trajectories, which yields to the extended incomplete trajectory having the same dimension as a complete trajectory, denoted as $\dim(\hat{\mathsf{X}}^\mathrm{n}) = \dim(\mathsf{X}^\mathrm{n})$. A forward feature tracking implies that the feature is tracked from frame t to frame $t+1$. On the contrary, the feature is tracked from frame t to frame $t-1$ is backward feature tracking. The forward/backward feature tracking is carried out until the extended incomplete feature trajectory has the same length as the complete trajectories.

Figure 2 illustrates the architecture of our algorithm in constructing the complete and incomplete trajectories. In this figure, there are only two moving objects (*i.e.* walking pedestrian and background) between key frames 1 and 2, and two new objects (cyclists) appear in between key frames 3

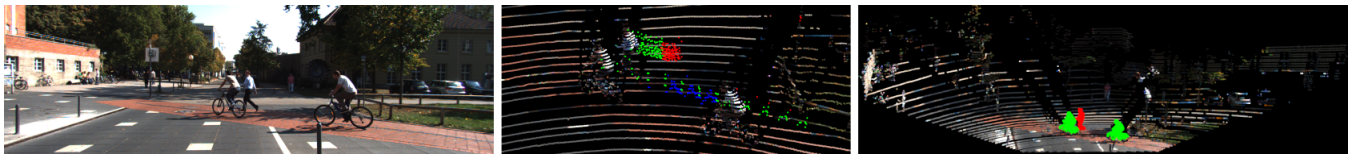


Fig. 3: Feature trajectories' completion for MS: left image shows the cyclist crossing the walking pedestrian. The green trajectories in the middle image are tracked features between two key frames, while the red and blue trajectories are acquired from backward and forward feature tracking, respectively. The right image shows the MS results.

and 4. Accordingly, the feature trajectories on the moving objects between key frames 1 and 2 are complete, while incomplete trajectories on moving objects occur due to the newly appearing or occlusion between key frame 3 and 4. Note that, because the feature tracking starts from the key frames, the objects not seen in the key frames (*e.g.* the newly appeared objects) are not tracked. This leads to some potential moving objects being omitted. Therefore, to overcome such issue, the incomplete trajectory construction is applied to re-tracked those neglected objects.

Figure 3 shows the constructed complete and incomplete trajectories with MS results. In this figure, the walking pedestrian was completely occluded by the passing cyclist, leading to incomplete trajectories of the pedestrian. Thus, the backward feature tracking is activated to extend the incomplete trajectories, see the red trajectories of the middle image. Besides, the newly appearing cyclist requires a forward feature tracking to extend the incomplete trajectories, see the blue trajectories of the middle image. Doing so, both of the complete and extended incomplete trajectory lengths have the same dimension, which allows the MS to overcome the loss of feature tracking. Note that the success of incomplete feature trajectory construction offers the following advantages: (a) The lost tracked objects are rediscovered and re-tracked. (b). The simultaneous motion segmentation on complete and incomplete feature trajectories now becomes possible.

V. Static-map and Dynamic Object Reconstruction

Once the sparse set of segmented feature trajectories is obtained from MS, we employ a 3D Region Growing technique [32] on the complete 3D scene points to obtain a dense segmentation of the point clouds. The multi-frame point clouds assigned to static scene parts are then registered together to incrementally build the static map, using minimal 3-point Random Sample Consensus (RANSAC) algorithm. The 3-point RANSAC algorithm uses Cayley representation of the rotation matrix, which allows to obtain rigid transformation between point clouds using linear solvers, similar to [6]. In addition, an Iterative Closest Point algorithm [7] is applied to refine the registration. Finally, the reconstruction of the moving objects is also obtained, in a very similar manner –by registering their observations from different view-ports.

VI. 2D-to-3D Label Transfer

We consider that semantic scene understanding should answer two questions: What is the object? And what is it doing? In other words, the object of interest (such as cars and pedestrians) should be discovered and recognized with semantic labels. Further, the object behaviour, such as a moving or parked cars, should be understood. In this context, semantic scene understanding has been partially addressed in [15] for moving vehicle motion prediction. We focus on the fusion of knowledge from 2D and 3D data to fully address the semantic scene understanding problem.

Since 2D image-based semantic labelling achieves very satisfactory performances [16], we propose to transfer the retrieved 2D object labels to their corresponding point clouds. For this purpose, the 2D-3D correspondences are established using a projective projection model: $\mathbf{x} \sim \mathsf{KP}\mathbf{X}$ where $\mathbf{x}$ is the 2D projections of the 3D points $\mathbf{X}$. K and P are the intrinsic and extrinsic parameters obtained from camera calibration. Thus, the label of $\mathbf{x}$ can be transferred to $\mathbf{X}$. Let $\mathcal{L}$ be the semantic label assigned to a 3D object. $S_{\mathcal{L}}$ is the real-world-averaged size of object class $\mathcal{L}$, and S_i is the object size (volume) measured from its 3D point cloud. To accurately transfer the 2D labels over m different observations, a max-pooling strategy is applied to obtain the desired label $\mathcal{L}^*$ for the given 3D object, such that:

$$\mathcal{L}^* = \underset{\mathcal{L}}{\operatorname{argmax}}\ \eta_i \rho_i,\ i = 1, \ldots, m. \tag{7}$$

Where $\eta_i = \frac{1}{e^{|S_i - S_{\mathcal{L}}|/S_{\mathcal{L}}}}$ is the 3D size similarity, and $\rho_i \in [0, 1]$ is the confidence score of the 2D labels obtained from the detector. Beside the objects labels, the motion status are also assigned to them as either static or dynamic with their motion trajectories. To sum up, there are two layers of semantic understanding in our framework: 1). Precise object localizations in both 2D image and 3D maps. 2). Motion behaviour analysis of moving objects, serving for higher level scene understanding.

VII. Experiments

We conducted extensive experiments on the real-outdoor KITTI benchmark [33] to evaluate the performances of the proposed algorithms. Three representative state-of-the-art methods, namely 3D Sparse Subspace Clustering (3D-SSC) [6], SMooth Representation Clustering (2D-SMR) [9] and Object Scene Flow (OSF) [27], are compared using Sensitivity (Sens.) and Specificity (Spec.) metrics. To obtain 2D semantic labels, we selected the Yolo [16] object detector

Sub-seq.	# Mot.	# Feat.		η	Mot. State				3D-SSC		2D-SMR $\mathcal{W}_1$		2D-SMR $\mathcal{W}_2$		3D-SMR $\mathcal{W}_1$		3D-SMR $\mathcal{W}_2$	
		# Dyn.	# Stat.		+	-	o	++	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
1	3	113	219	0.10	x	x	x	x	0.991	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2	3	115	230	0.13	x	x	x	x	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3	3	122	246	0.18	x	x	x	x	1.000	0.988	1.000	0.996	1.000	1.000	0.992	1.000	1.000	0.996
4	3	78	251	0.07	x	x	x	x	0.603	0.996	0.962	0.769	0.962	0.765	0.974	0.777	0.987	0.996
5	4	54	270	0.15	v	x	v	x	0.593	0.685	0.611	0.696	0.685	0.689	0.667	0.704	1.000	0.993
6	4	82	271	0.27	v	x	x	x	0.817	0.727	1.000	0.815	1.000	0.815	1.000	0.838	1.000	0.993
7	7	237	173	0.23	x	x	x	x	0.873	0.983	1.000	0.526	1.000	0.526	1.000	0.711	1.000	0.838
8	7	255	156	0.20	x	x	v	x	0.973	0.974	1.000	0.532	1.000	0.526	1.000	0.904	1.000	0.929
9	7	225	166	0.22	x	x	v	v	0.964	0.994	0.996	0.801	0.996	0.801	0.987	0.982	0.996	0.994
10	8	206	167	0.19	v	x	x	x	0.956	0.820	0.961	0.497	0.961	0.503	0.995	0.976	1.000	0.994
11	9	236	141	0.20	v	x	v	v	0.932	0.986	1.000	0.532	1.000	0.532	1.000	0.745	0.996	0.979
12	9	247	139	0.22	x	x	x	x	0.973	0.971	0.976	0.921	0.976	0.921	0.968	0.950	0.968	0.906
13	9	200	169	0.19	v	v	x	x	0.810	0.781	1.000	0.793	1.000	0.793	1.000	0.817	1.000	0.988
14	8	233	175	0.26	x	x	v	v	1	0.983	0.906	0.691	0.906	0.691	1.000	0.857	1.000	0.926
Average	6	172	198	0.18	/	/	/	/	0.892	0.921	0.956	0.755	0.963	0.754	0.970	0.876	**0.996**	**0.967**
Time(s)	/	/	/	/	/	/	/	/	35.122		0.054		**0.0378**		0.613		0.608	

TABLE I: Performance quantification on Pedestrian dataset. Columns |Sub-seq.|, |# Mot.|, and |# Feat.| show the sub-sequences index, moving objects number, dynamic features number, and static features number, respectively. $\eta = \frac{\#\text{ incomplete trajectories}}{\#\text{ total features}}$ represents the percentage of extended incomplete trajectories. Columns |Mot. State| show the motion states with symbols $+, -, \mathsf{o}, ++$ denoting new appearance, disappearance, occlusion, and reappearance scenarios discussed in Section IV. Symbols v and x mean that the motion states occur or do NOT occur, respectively. The last columns compare the Sensitivity and Specificity of algorithms 3D-SSC [6], 2D-SMR [9] and the proposed 3D-SMR.

which accurately detects and recognizes multi-class objects using image information in real-time. All the experiments are conducted on a computer with Intel Quad Core i7-2.7GHz, 32GB Memory using MATLAB.

A. Quantitative Evaluation

We select seven representative datasets, namely Campus, Highway, Junction, Market, Pedestrian, Red Light and Station, that have wide dynamic ranges regarding to frame lengths, number of moving objects, number of feature trajectories and status of motion changes. Note that the Highway, Junction, Market and Station sequences are evaluated in [6] by discarding some parts of the sequences where incomplete feature trajectories occur. Thanks to the proposed incomplete trajectory construction approach, those sequences are re-evaluated by including the scenarios of incomplete feature trajectories. Therefore, the sensitivity and specificity of 3D-SSC reported in this paper is relatively lower than in [6]. Newly evaluated challenging sequences (Campus, Pedestrian and Red Light), where incomplete trajectories occur very often, are selected on purpose to show the effectiveness of the proposed method.

Among all the tested sequences, we report the detailed descriptions of Pedestrian dataset (as example) in Table I. In this table, columns *Mot. State* show the changes of motion status within the sub-sequences (interval of every 10 frames), where symbols $+, -, \mathsf{o}, ++$ denote 4 different motion scenarios, namely new appearing, disappearing, occlusion, and reappearing. The number of motions are changing during the observations, resulting into mainly incomplete feature trajectories. Accordingly, our complete and incomplete trajectory construction architecture presented in Section IV is essential to address such tracking failures. Table I shows that the proposed 3D-SMR achieves significantly better performance against 3D-SSC and 2D-SMR in both sensitivity and specificity. Moreover, the computation time of 3D-SMR is about 0.6 second per frame, which is two-order magnitude faster than 3D-SSC.

The summary of evaluations on all representative datasets is shown in Table II. Note that all seven datasets are evaluated in the same manner as in Table I. In average, the proposed 3D-SMR outperforms the state-of-the-art 3D-SSC and 2D-SMR motion segmentation methods. Sacrificing the sensitivity, the OSF [27] achieves slightly better specificity than our 3D-SMR. Overall, the proposed 3D-SMR performs better than the representative state-of-the-art methods.

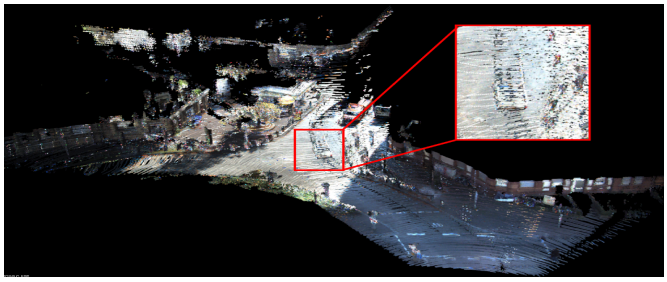
The quantitative results of Table I and II show the effectiveness of the proposed 3D-SMR algorithm. Firstly, the spatial closeness and motion consistency constraints of 3D-SMR make the subspace representation more robust than the 3D-SSC. Secondly, the spatial closeness constraint of 2D-SMR in image space under affine projection model suffers from perspective effects, which makes it insensitive to motions towards and outwards the camera. On the contrary, the direct MS on 3D Euclidean space of 3D-SMR is free from perspective effects. Moreover, the additional 3D motion consistency regularization term improves the robustness of the 3D-SMR. Thirdly, the proposed incomplete feature trajectory completion approach effectively recovers the loss of tracked objects, which tackles the challenges of complicated uncontrolled outdoor environments. Lastly, the computational efficiency of the 3D-SMR algorithm, which takes 0.6 second with MATLAB implementation, provides great potential of future real-time implementation.

B. Qualitative Evaluation

Thanks to the effectiveness of the proposed framework, the static-maps of all sequences are correctly reconstructed. The reconstructions for Junction and Campus sequences are shown in Fig. 4. In Fig. 4a, the moving van is partially occluded by a cyclist, which leads to the failure of pixel-level feature tracking. Likewise, Fig. 4c contains new ap-

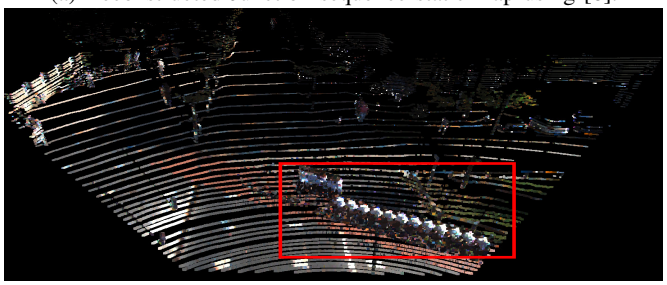
Sequence	# Frms.	# Objs.	# Feats.	η	OSF		3D-SSC		2D-SMR $\mathcal{W}_1$		2D-SMR $\mathcal{W}_2$		3D-SMR $\mathcal{W}_1$		3D-SMR $\mathcal{W}_2$	
					Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
Campus	60	4	341	0.23	0.404	0.988	0.920	0.888	0.944	0.658	0.947	0.621	**0.987**	0.816	0.970	**0.997**
Highway	50	2	392	0.24	0.579	0.994	**0.978**	0.625	0.609	0.963	0.613	0.962	0.825	**0.999**	0.962	0.995
Junction	90	3	416	0.24	0.613	0.966	0.892	0.943	0.968	0.998	0.971	0.997	0.973	**0.999**	**0.976**	0.997
Market	100	6	402	0.25	0.506	0.962	0.882	**0.852**	0.933	0.640	0.920	0.637	0.959	0.747	**0.989**	0.817
Pedestrian	140	6	370	0.18	0.519	**0.983**	0.892	0.921	0.956	0.755	0.963	0.754	0.970	0.876	**0.996**	0.967
Red Light	120	4	371	0.19	0.578	**0.987**	0.868	0.830	0.880	0.633	0.886	0.682	0.964	0.906	**0.998**	0.976
Station	50	5	417	0.28	0.164	**0.996**	0.901	0.631	0.942	0.486	**0.958**	0.437	0.918	0.703	0.929	0.875
Average	87	5	387	0.23	0.480	**0.982**	0.905	0.813	0.890	0.733	0.894	0.727	0.942	0.864	**0.974**	0.946

TABLE II: Quantification of OSF [27], 3D-SSC [6] , 2D-SMR [9] and our 3D-SMR in motion segmentation: this table is a summary of performances of the 7 representative datasets, and the dataset notations are detailed in Table I.

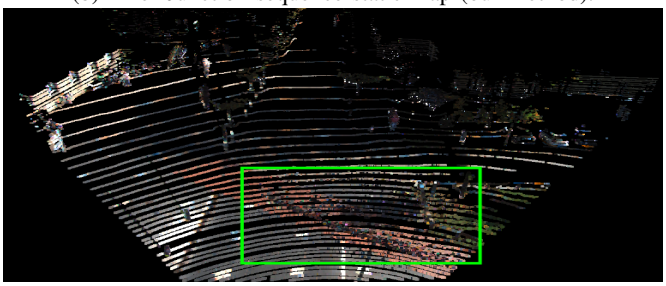


(a) Reconstructed Junction sequence static-map using [6].

(b) Finer Junction sequence static-map (our method).



(c) Reconstructed Campus sequence using [6].



(d) Finer Campus sequence static-scene (our method).

Fig. 4: Incomplete trajectory recovery assisted static maps reconstruction: (a) and (c) show that the reconstructed static maps using [6] contain some neglected moving objects due to the loss of feature tracking. With the help of incomplete feature trajectory completion, finer static maps of (b) and (d) are achieved by removing those loss-tracked moving objects.

pearing cyclist, which occludes the pedestrian (reappeared after cyclist's passing) during the observation (recall Fig. 2). In such cases, the occurrence of incomplete trajectories leads to the failure of [6]. Differently, rather than discarding those incomplete feature trajectories, we extend them to have the same length as of the complete feature trajectories, which allows the concurrent MS on both complete and incomplete trajectories. Fig. 4b and 4d show that higher quality static maps are obtained taking into account the incomplete feature trajectories recovery.

Qualitative results of moving object detection using the state-of-the-art methods on Market sequence are illustrated in Fig. 5. Figure shows that the OSF method is insensitive to detect those slow motions of pedestrians. Besides, the proposed 3D-SMR clearly achieves better motion segmentation than 3D-SSC and 2D-SMR.

Figure 6 presents the automatically labelled 3D map of Junction sequence with the proposed 2D-to-3D label transfer strategy. In this figure, the semantic information of the 3D objects are accurately discovered using the proposed max-pooling strategy, which avoids multi-labelling from different observations. Furthermore, the accurate object motion velocities are estimated using 3-point RANSAC and ICP point cloud registration. At first, objects are categorized as either static or moving objects. Then the accurate online motion information (*e.g.* motion direction, linear and angular speed, etc.) is obtained thanks to the precisely recovered odometry knowledge from the proposed framework.

VIII. Conclusion and Future Work

We proposed a 3D-based Smooth Representation Clustering (3D-SMR) algorithm with motion consistency regularization for motion segmentation and scene understanding. The proposed 3D-SMR algorithm is proved to be more efficient and accurate than the state-of-the-art methods using comprehensive real-world KITTI datasets. The effectiveness of the incomplete trajectory construction, which is essential in many practical scenarios, is demonstrated. Finally, a complete framework for dynamic scene analysis using 3D motion segmentation and 2D-to-3D label transfer is proposed. The

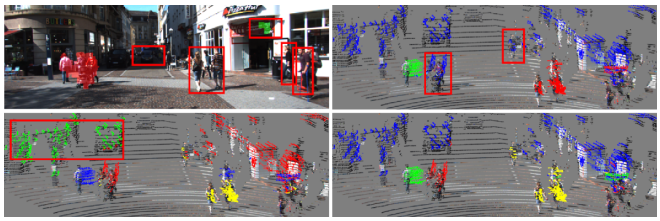


Fig. 5: Illustration of moving object detection using OSF [27] (top left), 3D-SSC [6] (top right), 2D-SMR (bottom left) and our 3D-SMR (bottom right). The red boxes highlight the undetected motions and incorrectly segmented motions.

knowledge of object motion behaviours and their semantic information allows a high level 3D scene understanding. The results show the prospects of the proposed 2D-to-3D label transfer idea, yet, 3D labelling accuracy quantification is remained as a future work.

REFERENCES

[1] Lafarge, F., & Mallet, C. Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation. In IJCV, 2012.

[2] Berger, C., & Rumpe, B.. Autonomous driving-5 years after the urban challenge: The anticipatory vehicle as a cyber-physical system. In arXiv, 2014.

[3] Civera, J., Grasa, O., Davison, A. J., & Montiel, J. 1-Point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry. In Journal of Field Robotics, 2010.

[4] Paudel, D. P., Demonceaux, C., Habed, A., Vasseur, P., & Kweon, I. S. 2D-3D camera fusion for visual odometry in outdoor environments. In IROS, 2014.

[5] Burgard, W., Stachniss, C., & Hhnel, D. Mobile robot map learning from range data in dynamic environments. In Auto. Nav. in Dyn. Env., 2007.

[6] Jiang, C., Paudel, D. P., Fougerolle, Y., Fofi, D., & Demonceaux, C. Static-map and Dynamic Object Reconstruction in Outdoor Scenes using 3D Motion Segmentation. In RAL, 2016.

[7] Jiang, C., Fougerolle Y., Fofi D., and Demonceaux C., Dynamic 3d scene reconstruction and enhancement, in International Conference Image Analysis and Processing, Catania, Italy, Sept. 2017.

[8] Elhamifar, E., & Vidal, R. Sparse subspace clustering: Algorithm, theory, and applications. In TPAMI, 2013.

[9] Hu, H., Lin, Z., Feng J., & Zhou J.. Smooth Representation Clustering. In CVPR, 2014.

[10] Elqursh, A., & Elgammal, A. Online motion segmentation using dynamic label propagation. In ICCV, 2013.

[11] Brooksby, G., Hoogs, A., & Doretto, G. Moving object segmentation using scene understanding. CVPR-Workshop, 2006.

[12] Saleemi, I., Hartung, L., & Shah, M. Scene understanding by statistical modeling of motion patterns. In CVPR, 2010.

[13] Han, J., Shao L., Xu D., & Shotton J. Enhanced computer vision with microsoft kinect sensor: A review. In IEEE Trans. on Cybern., 2013.

[14] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., & Schiele, B. The cityscapes dataset for semantic urban scene understanding. In CVPR, 2016.

[15] Geiger, A., Lauer, M., Wojek, C., Stiller, C., & Urtasun, R. 3d traffic scene understanding from movable platforms. In TPAMI, 2014.

[16] Joseph, R., Divvala, S., Girshick, R., & Farhadi, A. You only look once: Unified, real-time object detection. In CVPR, 2016.

[17] Vidal, R., & Hartley, R. Motion segmentation with missing data using powerfactorization and gpca. In CVPR 2004.

[18] Yan, J., & Pollefeys, M. Articulated motion segmentation using ransac with priors. In Dynamical Vision, 2007.

[19] Rao, S., Tron, R., Vidal, R., & Ma, Y. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. In TPAMI, 2010.

[20] Tron, R., & Vidal, R.. A benchmark for the comparison of 3-d motion segmentation algorithms. In CVPR, 2007.

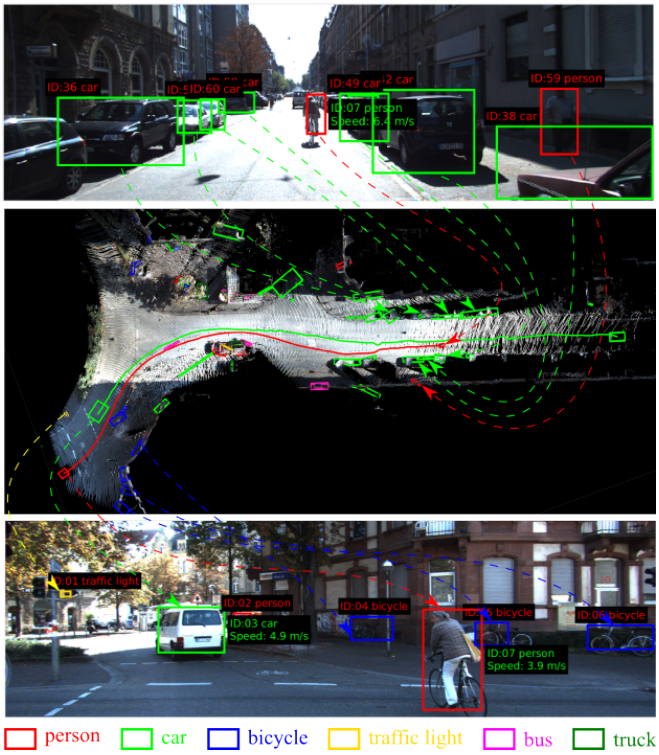


Fig. 6: Semantically labelled dynamic scene using 2D-to-3D label transfer: Top and bottom are the last and the first semantically labelled images of Junction sequence. Middle image is the top view of our reconstructed dynamic scene with semantic labels. Dashed lines connect the objects in 3D map and 2D image. The solid red and green curves are the trajectories of the cyclist and the van, respectively (the remaining objects are static).

[21] Stückler, J., & Behnke, S. Efficient Dense Rigid-Body Motion Segmentation and Estimation in RGB-D Video. In IJCV, 2015.

[22] Sofer, Y., Hassner, T., & Sharf, A. Interactive Learning for Point Cloud Motion Segmentation. In Computer Graphics Forum, 2013.

[23] Wang, C., Thorpe, C., & Thrun, S. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In ICRA, 2003.

[24] Pomerleau, F., Krusi, P., Colas, F., Furgale, P., & Siegwart, R. Long-term 3D map maintenance in dynamic environments. In ICRA, 2014.

[25] Ambrus, R., Bore, N., Folkesson, J., & Jensfelt, P. Meta-rooms: Building and maintaining long term spatial models in a dynamic world. In IROS, 2014.

[26] Kochanov, D., Osep, A., Stückler, J., & Leibe, B. Scene flow propagation for semantic mapping and object discovery in dynamic street scenes.In IROS, 2016.

[27] Menze, M., & Geiger, A. Object Scene Flow for Autonomous Vehicles. In CVPR, 2015.

[28] Bartels R., & Stewart G. Solution of the matrix equation AX + XB = C. In Communications of the ACM, 1972.

[29] Weinzaepfel, P., Revaud, J., Harchaoui, Z., & Schmid, C. Deepflow: Large displacement optical flow with deep matching. In ICCV, 2013.

[30] Zdenek, K., Mikolajczyk, K., & Matas, J. Forward-backward error: Automatic detection of tracking failures. In ICPR, 2010.

[31] Revaud, J., Weinzaepfel, P., Harchaoui, Z., & Schmid, C. DeepMatching: Hierarchical Deformable Dense Matching. In IJCV, 2015.

[32] Vosselman, G., Gorte, B. G., Sithole, G., & Rabbani, T. Recognising structure in laser scanner point clouds. In ISPRS Archives, 2004.

[33] Geiger, A., Lenz, P., & Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In CVPR, 2012.