Moving Object Detection by 3D Flow Field Analysis

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

Moving Object Detection by 3D Flow Field Analysis

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

***Abstract*—Map-based localization and sensing are one of the key components in autonomous driving technologies, where high quality 3D map reconstruction is fundamentally utmost important. However, due to the highly dynamic and uncontrollable properties of real world environment, building a high quality 3D map is not straightforward and requires several strong assumptions. To address this challenge, we present a complete framework, which detects and extracts the moving objects from a sequence of unordered and texture-less point clouds, to build high quality static maps. To accurately detect the moving objects from data acquired with a possibly fast moving platform, we propose a novel 3D Flow Field Analysis approach in which we inspect the motion behaviour of the registered point sets. The proposed algorithm elegantly models the temporal and spatial displacement of the moving objects. Thus, both small moving objects (*e.g.* walking pedestrians) and large moving objects (*e.g.* moving trucks) can be detected effectively. Further, by incorporating the Sparse Subspace Clustering framework, we propose a Sparse Flow Clustering algorithm to group the 3D motion flows under both the constraints of motion similarity and spatial closeness. To this end, the static scene parts and the moving objects can be independently processed to achieve photo-realistic 3D reconstructions. Finally, we show that the proposed 3D Flow Field Analysis algorithm and the Sparse Flow Clustering approach are highly effective for motion detection and segmentation, as exemplified on the KITTI benchmark, and yield high quality reconstructed static-maps as well as rigidly moving objects.**

***Index Terms*—Motion Flow Detection, Motion Segmentation, Dynamic Scene Analysis, 3D Map Reconstruction**

MAP-based localization and sensing are one of the key components in autonomous driving technologies [Levinson et al., 2007; Lu et al., 2019; Magnusson et al., 2007], where high quality 3D map reconstruction is fundamentally utmost important [Seif and Hu, 2016]. However, due to the highly dynamic and uncontrollable properties of real world environment, building a high quality 3D map is never easy. In literature, there are significant amount of research on 3D map reconstruction problems, representatively, the traditional image-based Structure-from-Motion [Pollefeys et al., 2008] technique, the depth-image-based Truncated Signed Distance Function [Newcombe et al., 2011] approach, and the lidar-based Localization and Mapping [Wen et al., 2019; Zhang and Singh, 2018] method. Generally, such approaches achieve very nice results for nearly static environments, while the 3D reconstruction quality significantly degrades when facing highly dynamic and crowded environments, see Fig. 1 for example.

C. Jiang, Y. Fougerolle, D. Fofi, C. Demonceaux are with ImViA, VIBOT ERL CNRS 6000, Université Bourgogne Franche-Comté, Dijon, France
D.P. Paudel is with Computer Vision Lab, ETH Zurich, Zurich, Switzerland.
Manuscript received xxxx, 2020; revised xxxx, 2021.

Unfortunately, practical scenarios (*e.g.* streets or markets) are very often highly dynamic, the scene modelling and the camera localization can become very challenging tasks, mainly due to the numerous dynamic scene parts which yield artefacts and poor localization. Our previous work Jiang et al. [2016] shows that high quality scene modelling and precise camera localization can be achieved by detecting and removing the dynamic parts. We therefore introduce our method for the robust, accurate and efficient detection of dynamic objects with high quality reconstruction of the static scene.

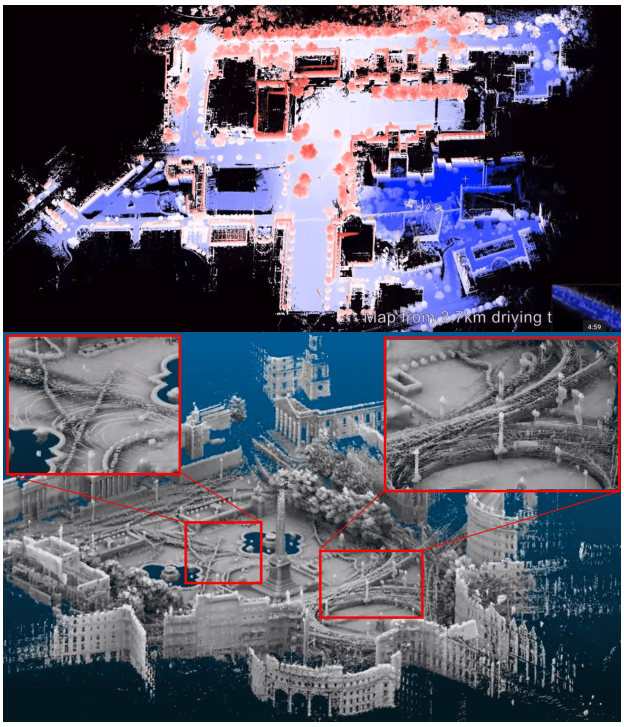

Fig. 1. 3D reconstruction using Lidar Odometery and Mapping (LOAM) [Zhang and Singh, 2016] technique: top image shows a decent quality 3D map of a static campus environment, while the mapping result (bottom image) of the plaza is quite unsatisfactory due to the "ghost" artefacts (see the zoom-in area of red boxes) caused by moving objects.

Given a mobile camera-lidar platform, both the foreground and the background observations are observed as moving due to the camera's ego-motion. It is natural for human beings to identify the real moving objects due to their capability of visual object segmentation and tracking, such task is especially complex for machines and often relies on strong assumptions (sizes and velocities of the moving objects, for instance). To tackle this challenge, the Background Modelling and Subtraction-based methods Jung and Sukhatme [2004];

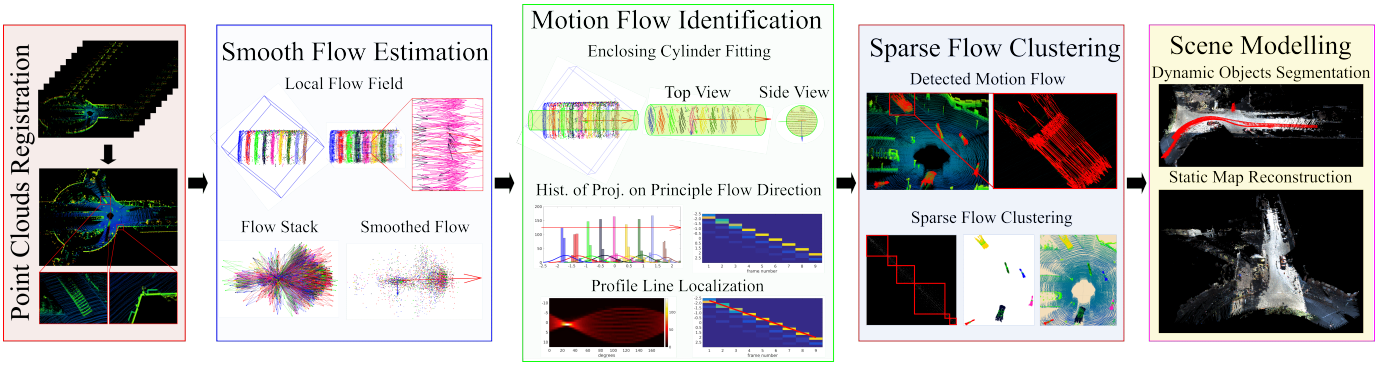

Fig. 2. Overview of the proposed system to detect and segment the dynamic scene parts and to reconstruct the static scene (Block 5). Given a short registered 3D point cloud sequence (Block 1), for each point of the center frame, we compute a Smooth Flow Vector (SFV) representing its motion behaviour (Block 2, Section III-A). Then, an Infinite Enclosing Cylinder is determined to bound the inlier neighbourhood with similar motions (Block 3 up, Section III-B). The drift effect that corresponds to the principal motion is obtained from the analysis of the histogram of the projections of the cloud of points onto the SFV (Block 3 down, Section III). The Sparse Flow Clustering algorithm then regroups the motion flows, as detailed in Section IV.

Sheikh et al. [2009]; Yun et al. [2017]; Zhou et al. [2012] are proposed by compensating the camera ego-motion and thus, the moving objects can be detected by applying the background subtraction operation. Such methods are highly relying on the accuracy of ego-motion estimation and the lighting consistency. In a more robust manner, the Object-based Detection, Segmentation and Tracking approaches [Cho et al., 2014; Leibe et al., 2008; Menze et al., 2018; Rashed et al., 2019; Ray and Chakraborty, 2019] are object-level motion detection by using super-pixels or object models. The moving objects are discriminated by comparing their motion trajectories versus the camera ego-motion trajectory. Such approaches are usually accurate and not sensitive to noise, while they require precise object appearance modelling and accurate object trajectory estimation. Instead, the Motion Trajectory Analysis-based techniques Brox and Malik [2010]; Elhamifar and Vidal [2013]; Vidal et al. [2008] directly segment the object's feature trajectory according to their motion subspaces, which is mathematically more elegant. Nevertheless, such approaches usually prefer continuous feature tracking and is not robust to occlusions.

In real-world scenarios, the situations are more complicated due to the lack of prior knowledge of the objects, such as their sizes, their appearances, their positions, their velocities, etc. The above mentioned approaches are insufficient to perform correct and accurate moving object detection and segmentation, especially in crowded or night-time environments. Facing this challenge, we propose a solely 3D point cloud-based moving object detection approach taking into account the merits of 3D lidar (*e.g.* large field of view, precise measurement, night vision ability, etc).

Therefore, to address the problems of dynamic object detection and motion behaviour analysis, we propose a novel framework by using the 3D Flow Field Analysis, see Fig. 2. Firstly, by compensating the sensor ego-motion (*e.g.* by using LOAM [Zhang and Singh, 2016], LO-Net [Li et al., 2019]), there exist continuous displacements of point sets of moving objects, while the point sets of static scene parts have no displacement. Therefore, the static scene parts should overlay together while the dynamic scene parts should not. By connecting the points of moving objects according to their temporal and spatial displacement, they become a set of motion vectors. In this regard, we propose a 3D Vector Field Analysis approach which identifies the static flows and the motion flows. After compensating the camera ego-motion, for every point in the previous frame, a flow vector is established by subtracting its nearest neighbour in the current frame. The flow vector encodes the motion direction and velocity of the objects. By exploiting these properties, the flow vectors of moving objects, so-called the motion flows, can be detected and classified into their independent motions. Moreover, a 3D-based Sparse Flow Clustering (3D-SFC) algorithm is proposed to cluster the detected motion flows. Such 3D-SFC can robustly group the motion flows by measuring their temporal motion similarity and spatial closeness. To this end, both the static scene parts and the moving rigid objects can be reconstructed independently. In brief, the proposed framework for motion discrimination and scene reconstruction consists of four main steps, namely the Smooth Flow Vector Estimation, the Motion Flow Estimation, the Sparse Flow Clustering, and the Scene Modelling.

Firstly, the *Smooth Flow Vector Estimation* step aims to compute the motion of each 3d point between two successive frames. Fundamentally, a 3d point's motion can be expressed as a *3d Flow Vector* representing its motion speed and direction, and estimated by the subtraction of the corresponding points between two consecutive frames. Roughly, these 3d point correspondences can be efficiently established by applying naive nearest neighbour search from the current frame to the next frame. Unavoidably, the estimated 3d flow vectors can easily be contaminated by the noisy observations. Therefore, under the local motion consistency assumption, the smooth flow vector is simply estimated as the locally dominant flow vector within a local neighbourhood, such as a 3D bounding box for instance.

Secondly, the *Motion Flow Identification* step identifies the flow vectors corresponding to the moving objects by analysing the objects' temporal and spatial displacement along their motion directions. For each flow vector, an enclosing cylinder is adapted to select the most representative neighbour points

which preserve a persistent geometric structure. The projections of those points onto the current flow vector are stored in a histogram because the motion flows can be identified by detecting the shifts within the concatenated histogram from all the frames, as detailed in Section III-D.

Thirdly, the *Sparse Flow Field Clustering* step groups the detected motion flows into their motion subspaces. Our approach is based on the distinctiveness of both motion directions and spatial distributions. We seek for the sparse self-representation of motion flows from their motion subspace, which forms a sparse similarity graph. The motion flows can then be separated into independent motions by applying spectral clustering on the similarity graph.

Lastly, the *3D Scene Modelling* step builds the photo-realistic 3d models of the dynamic outdoor environments. To densely segment the dynamic scene parts, the 3D region growing approach is applied by taking the detected motion flows as seeds. To this end, the reconstruction of the static scene is achieved by registering only the static scene parts, while the rigidly moving objects, such as moving cars, are individually reconstructed from their registered dynamic parts.

This article is an extended version of our previous work [Jiang et al., 2017c] and our contribution can be summarized as follows:

- We propose a robust and efficient framework for the detection and the segmentation of moving objects as well as the reconstruction of the static map from highly dynamic outdoor scenes.
- We present a novel algorithm for moving object detection using 3D vector flow analysis which outperforms the state-of-the-art methods.
- We propose a new Sparse Flow Clustering model based on sparse subspace self-representation and spatial closeness of the flow vectors.

I. Literature Review

Moving Object Detection (MOD) has been a long-lasting open problem raised by [Limb and Murphy, 1975] who aimed to estimate the velocity of moving objects in images from television stream. From then on, MOD becomes a very popular research field over the past few decades due to the wide ranges of applications, such as video surveillance [Reilly et al., 2010], object discovery [Pont-Tuset et al., 2017], scene modeling [Heikkila and Pietikainen, 2006], etc. Considering the system setup, there are two major branches of research: the *Stationary Camera-based MOD* approaches as extensively reviewed in [Benezeth et al., 2010; Elhabian et al., 2008; Joshi and Thakore, 2012], and the *Moving Camera-based MOD* techniques being profoundly discussed by [Jiang, 2017; Yazdi and Bouwmans, 2018]. In this article, we focus on the most related research work using moving camera setups, and discuss their positive/negative aspects comparing to the proposed algorithms.

A. Image-based MOD

Feature Trajectory Analysis-based Approaches: feature trajectories are the one of the most important clues of object motions. In this context, *Motion Segmentation* (MS) techniques, such as the Generalized Principal Component Analysis (GPCA) [Vidal et al., 2005], RANSAC-based MS [Yan and Pollefeys, 2007], and Agglomerative Subspace Clustering [Rao et al., 2010], are proposed to group the feature trajectories according to the objects' motion subspaces. The GPCA is the representative approach that offers an algebro-geometric solution to the MS problem without the knowledge of subspace number and dimension, by representing the subspaces with a set of homogeneous polynomials. As claimed by the authors, the GPCA also provides a robust initialization to iterative techniques such as K-subspaces or Expectation Maximization algorithms. However, the determination of number of clusters and their dimensions only works for noise free data in practice.

Differently, [Elhamifar and Vidal, 2013] proposed the groundbreaking Sparse Subspace Clustering (2D-SSC) algorithm relying on the self-representation property of the affine motion subspace. The 2D-SSC assumes that one feature trajectory can be represented by other feature trajectories from the same motion subspace. By incorporating the sparsity constraint on the relaxed ℓ_1 optimization, the 2D-SSC offers a robust solution to MS with outliers and achieves significantly better performances. However, the computational complexity of the 2D-SSC is proportional to the cubic of the problem size, and is therefore expensive for large scale data. As inspired, [Hu et al., 2014] proposed a SMooth Representation (2D-SMR) clustering model which outperforms the existing methods in literature by enforcing the grouping effects of the motion subspaces from image feature trajectories. To overcome the perspective projection problem of the image feature trajectories, [Jiang et al., 2016] proposed a 3D data-based Sparse Subspace Clustering (3D-SSC) algorithm which achieves comparative performances against its 2D counterparts without affine motion constraint. This algorithm relies on the consistency of the tracked trajectories and is therefore sensitive to lost tracking situations and partial occlusions. To improve its robustness, [Jiang et al., 2017b] proposed a 3D-SMR algorithm which jointly benefits from the 2D-SMR in scalable feature size and tracking correspondence. In a more sophisticated manner, [Keuper et al., 2018] incorporate the low-level feature trajectory and high-level object recognition cues to achieve better performance. Inherently, feature trajectory construction is sensitive to image noise and environment change, making such approaches limited to slow camera motion and temporally consistent lighting conditions.

Motion Flow Analysis-based Approaches: the motion flow, which encodes the motion magnitude and direction of the image pixel, is widely used in moving object discovery by analyzing the flow field discontinuities, such as [Huang et al., 2018; Mémin and Pérez, 2002; Yokoyama and Poggio, 2005]. The piecewise-smooth flow field are segmented by using Hierarchical [Mémin and Pérez, 2002], Level-set [Mitiche and Sekkati, 2006], or Graph-cut [Wedel et al., 2009] segmentation algorithms. More recently, [Ma et al., 2019; Menze et al., 2018] intended to detect and analyze the rigidly moving objects as Object Scene Flow (OSF) using stereo vision set-up. Inspired by OSF [Menze and Geiger, 2015], Kochanov et al. [Kochanov et al., 2016] proposed to detect and segment the

moving objects by propagating the OSF output to construct the static-map. The OSF-based approaches usually achieve more precise results, however, they are sensitive to the environment changes and require precise object motion model estimations.

B. Lidar-based MOD

2D Lidar-based Approaches: 2D lidar sensors (or single layer laser scanner) are widely used in industrial robots [Wang et al., 2015] or ADAS applications [Ziebinski et al., 2017] for object detection and tracking. Traditional approaches [Mertz et al., 2013; Wang et al., 2007, 2015] proposed to detect and track the moving objects along side with the simultaneous localization and mapping (SLAM) framework. The moving objects are detected by discriminating their temporal and spatial displacement. However, such method is limited to 2D laser scanner with the assumption of flat ground plane and a specific height range of moving object.

3D Lidar-based Approaches: 3D lidar (or multi-layered laser scanner) are very popular in autonomous driving applications [Levinson et al., 2011] nowadays. Taking the advantage of precise 3D point clouds, [Steinhauser et al., 2008; Sualeh and Kim, 2019; Wang et al., 2012; Ye et al., 2016] proposed to detect the possible moving objects (*e.g.* cars or pedestrians) and track them as motion candidates. Such approaches are trivial but require precise classifiers or feature descriptors ([Dewan et al., 2016]) for object recognition, which is usually impractical due to the sparse point cloud and object occlusions. Apart from the above geometrical analysis-based approaches, [Börcs et al., 2017; Engelcke et al., 2017] applied the deep learning techniques resulting in more precise object recognition capability. Without relying on the object knowledge, [Asvadi et al., 2015; Azim and Aycard, 2012] utilized the occupancy grid map to statically predict and track the moving objects. Nonetheless, designing the occupancy grid size and the selection of statistical model are empirically difficult.

Recently, Deep Learning -based approaches [Behl et al., 2019; Fan and Yang, 2019; Liu et al., 2019a,b] presented interesting results on 3D scene flow estimation thanks to the recent advances in computational resources and large-scale training dataset. In particular, [Fan and Yang, 2019] proposed a series of point-based recurrent neural networks, *i.e.* the PointRNN, the PointGRU, and the PointLSTM, for dynamic point cloud forecasting via flow predicting. Such approaches adopt the spatio-temporally-local correlation to aggregate the point features and their states according to the point coordinates. [Liu et al., 2019a] focus on 3D action recognition, dynamic point cloud segmentation, and scene flow estimation with multiple frames. The proposed FlowNet-based methods apply an end-to-end model for both point feature association and flow estimation. Other approaches [Cho et al., 2014; Takabe et al., 2016] intended to fuse the image and lidar observations for MOD using photometric and depth consistencies. [Rashed et al., 2019] made use of both the camera and lidar for robust MOD in low-light autonomous driving environments. Although the deep learning-based approaches show promising results, it is difficult to collect massive training data and to have heavy computational resources during the training process.

To summarize, unlike the above discussed methods, the proposed algorithms neither rely on feature tracking across the frame sequence contributing to their robustness to occlusions, nor require exhaustive machine learning training process making them being handy and easy to implement. Our MOD algorithm directly detects and segments the motion flows using raw 3D point cloud sequence without texture information. Although a rough 3D point cloud registration step is required for ego-motion compensation, the traditional ICP-based registration techniques [Fitzgibbon, 2003] are sufficient. Moreover, our method is very generic in detecting moving objects in terms of size, speed and direction.

II. Fundamental Definitions and Notations

Let $\mathsf{X} = \{\mathsf{x}_1, \cdots, \mathsf{x}_m\}$, where $\mathsf{x}_i \in \mathbb{R}^3$, be a 3D point set (cloud). And let $\mathsf{W} = \{\mathsf{w}_1, \cdots, \mathsf{w}_m\}$, where $\mathsf{w}_i \in \mathbb{R}^3$, be the set of flow vectors associated to X. The 3D vector field Ω defined by X and W is notated as $\Omega : \mathsf{X} \rightarrow \mathsf{W}$. Given a sequence of point sets from a dynamic scene, we define $\mathsf{S} = \{\mathsf{X}_t, t = 1, \ldots, n\}$ as the collection of multiple observed point sets that evolve over time t. Likewise, $\mathsf{Z} = \{\mathsf{W}_t, t = 1, \ldots, n-1\}$ is the collection of flow vectors associated to S.

For two 3D point sets A and B, the vector field $\Omega : \mathsf{A} \rightarrow \mathsf{W}$ can be obtained by the element-wise subtraction between the two point sets. We define the element-wise subtraction operation $\mathsf{A} \ominus \mathsf{B}$ as

$$\mathsf{A} \ominus \mathsf{B} = \{\mathsf{w}_i := \mathsf{x}_i - \mathsf{y}_i, \quad \forall \mathsf{x}_i \in \mathsf{A}\}, \tag{1}$$

where x_i is an element of A, and $\mathsf{y}_i = \mathscr{N}(\mathsf{x}_i, \mathsf{B})$ is the closest point of x_i in B. The subtraction $\mathsf{x}_i - \mathsf{y}_i$ defines the flow vector w_i. The closest point function $\mathscr{N}(\mathsf{x}, \mathsf{B})$ is defined as

$$\mathscr{N}(\mathsf{x}, \mathsf{B}) = \underset{\mathsf{y} \in \mathsf{B}}{\operatorname{argmin}} \|\mathsf{x} - \mathsf{y}\|. \tag{2}$$

In a similar manner, the nearest neighbourhood set of points centred at x within a radius r is given by

$$\mathscr{N}(\mathsf{x}, \mathsf{B}, r) = \{\mathsf{y} \in \mathsf{B} : \|\mathsf{x} - \mathsf{y}\| \leq r\}. \tag{3}$$

We also define $\mathscr{P}(\mathsf{S}, \mathsf{w})$, the projection of set S on the flow vector w (similarly, $\mathscr{P}(\mathsf{x}, \mathsf{w})$ for point x), such that

$$\mathscr{P}(\mathsf{S}, \mathsf{w}) = \{p : p = \mathsf{w}^\intercal \mathsf{x}, \mathsf{x} \in \mathsf{S}\}. \tag{4}$$

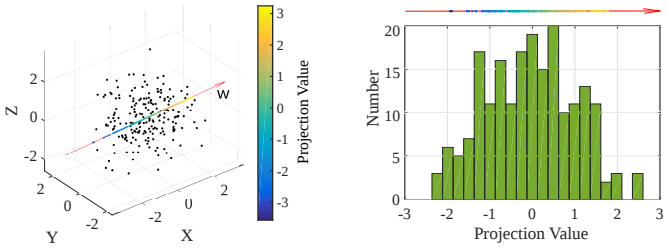

Fig. 3. Illustration of the histogram of 3D point projections: Left image shows a set of 3D points S (the black dots) and their projections $\mathscr{P}(\mathsf{S}, \mathsf{w})$ (the color-coded dots encoded by their projection values) along the 3D vector w (the red arrow which corresponds to the largest principle axis of S). A 20-bin 1D histogram is then constructed by using the projection values of $\mathscr{P}(\mathsf{S}, \mathsf{w})$.

We refer the illustrative examples of Eq. (4) to Fig. 3 and Fig. 4. In Fig. 3, a set of 3D points are projected onto the given 3D vector and the projection values are statistically represented as an n-bin one dimensional histogram. Similarly, in Fig. 4, two 3D points $\mathsf{x}_1, \mathsf{x}_2$ are projected onto w_c as two blue dots p_1, p_2. Note that the projection of a three dimensional point to the given 3D vector axis corresponds to its foot of perpendicular to the 3D vector, but we only take into account its scalar abscissa p on the axis. The origin of the axis is a specified 3D point, e.g. the mean values of the 3D point set. Since the mathematical representation of a 3D point is similar to a 3D vector, the projection of one 3D vector to the given 3D vector can be performed in a similar manner.

Furthermore, let $\Theta \subset \mathsf{S}$ be the points within an infinite cylinder centred at x_c, of radius r and axis w_c, given by

$$\Theta(\mathsf{x}_c, \mathsf{S}, \mathsf{w}_c, r) = \{\mathsf{x} : \|\mathsf{x} - \mathsf{x}_c\|^2 - \mathscr{P}(\mathsf{x}, \mathsf{w}_c)^2 \leq r^2, \mathsf{x} \in \mathsf{S}\}. \tag{5}$$

In other words, Eq. (5) rejects the points which have point-to-axis distances larger than the cylinder radius r, see Fig. 4 as an example.

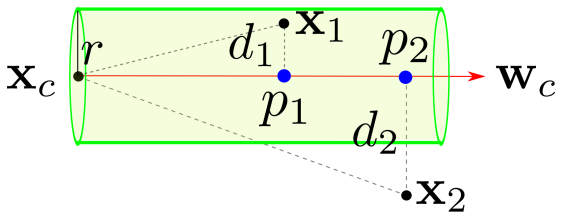

Fig. 4. Interpretation of an enclosing cylinder centred at x_c and axis w_c. Two 3D points $\mathsf{x}_1, \mathsf{x}_2$ are projected onto the cylinder axis w_c as p_1 and p_2. And the distances from points $\mathsf{x}_1, \mathsf{x}_2$ to p_1, p_2 are notated as d_1 and d_2, respectively. Since ${d_1}^2 = \|\mathsf{x}_c - \mathsf{x}_1\|^2 - \mathscr{P}(\mathsf{x}_1, \mathsf{w}_c)^2 \leq r^2$, $\mathsf{x}_1 \in \Theta$ is considered as inside the cylinder. In contrast, $\mathsf{x}_2 \notin \Theta$ is outside the cylinder.

Hereafter, we define some notations for matrix operation. Let $\mathsf{A} = (a_{ij})$ be the element-wise representation of an $m \times n$ -sized matrix. Its column-wise representation is notated as $\mathsf{A} = [\mathsf{a}_1, \cdots \mathsf{a}_j, \ldots \mathsf{a}_n]$ where a_j is an m-dimensional vector. $\mathsf{A} \succeq 0$ means that A is a symmetric and positive semi-definite matrix.

III. Flow Field Analysis

We intend to identify the moving objects, *i.e.* moving cars and cyclists, from a sequence of 3D point clouds. Essentially, a moving object should fulfil the criteria that a certain spatial displacement occurs within a certain time period, which can be described by a set of motion flows. To analyse, we propose the *3D Flow Field Analysis* model based on the local motion consistency assumptions. Refer to the optical flow estimation Horn and Schunck [1981] and the 3D scene flow estimation Vedula et al. [2005], two assumptions are made:

i. the motion behaviours of the optical flows within a small neighbourhood are similar;
ii. the local geometric structure does not change rapidly.

A. Smooth Flow Vector Estimation

Let a collection of n-consecutive point sets be $\mathsf{S} = \{\mathsf{X}_t, t = 1, \ldots, n\}$. For $t = 1, \ldots, n-1$, we compute the point-wise flow sets $\mathsf{Z} = \{\mathsf{W}_t, t = 1, \ldots, n-1\}$ which represent the motion of points over time t, as follow:

$$\mathsf{W}_t = \mathsf{X}_{t+1} \ominus \mathsf{X}_t. \tag{6}$$

Recall the definition of Eq. (1) and Eq. (2), the element-wise subtraction is performed to estimate the motion flows between frame t and frame $t+1$. Due to the noisy observation of point set and the incorrect point-pair association, see Fig. 2 Block 2, the estimated motion flow using Eq. (6) can be incorrect. Therefore, by taking the locally homogeneous assumption of neighbouring flow vectors, we perform the smoothing of vector field by updating each $\mathsf{w}_i \in \mathsf{W}_t$ as

$$\mathsf{v}_i^* = \underset{\mathsf{v} \in \mathbb{R}^3}{\operatorname{argmax}} \sum_{\mathsf{w} \in \Omega(\mathcal{N})} \mathsf{w}^\mathsf{T}\mathsf{v} \quad \text{s.t.} \quad \|\mathsf{v}\| = 1\ , \tag{7}$$

where v_i^* is the desired smooth flow vector to replace w_i. Refer to Eq. (2), $\mathcal{N} = \mathscr{N}(\mathsf{x}_i, \mathsf{X}_t, r)$ is the neighbourhood (within the radius r) that defines the local flow field $\Omega(\mathcal{N})$. Actually, Eq. (7) finds the consensus flow v_i^* which minimizes the overall distances between v_i^* and the flows within $\Omega(\mathcal{N})$. The problem of Eq. (7) can be solved efficiently as an eigen-decomposition problem. Its solution can be obtained by computing the eigenvectors of the covariance matrix $\mathsf{W}^\mathsf{T}\mathsf{W}$, where the rows of W are w^T for all $\mathsf{w} \in \Omega(\mathcal{N})$. The desired smoothed flow vector corresponds to the eigenvector of the largest eigenvalue. Note that, all the $\mathsf{w} \in \Omega(\mathcal{N})$ are normalized to unit vectors to obtain the optimal solution.

B. Motion Flow Discrimination

Recall the second assumption that the structure of the local point sets $(\Theta_t = \Theta(\mathsf{x}, \mathsf{X}_t, \mathsf{w}, r), t = 1, \ldots, n)$ is preserved within a short time period t. Thus, the measurements of a local point set Θ_t moving along w from Eq. 7 are homomorphic. Therefore, the shape of distribution of projections $\mathscr{P}_t = \mathscr{P}(\Theta_t, \mathsf{w})$ remain unchanged over time interval $[1, t]$. Let $\mathcal{H}_t$ be a k-bin 1D histogram of projections $\mathscr{P}_t$ at time t. The motion state of the point sets can be described by the following equation:

$$\mathcal{H}_{t+1}(b) = \mathcal{H}_t(b + \alpha(t)), \tag{8}$$

where b is one bin of the histogram, and $\alpha(t) = \beta t$ in which β corresponds to the displacement of the histogram (or projections) from t to $t+1$. Eq. (8) implies that the histogram is replicated from $t = 1, \ldots, n$ thanks to the temporal local structure and velocity consistency.

Given a sequence of histograms $\mathcal{H}_t(b), t = 1 \ldots, n$, our task is to estimate β and b such that Eq. (8) is satisfied for all t. Mathematically, it can be modelled as the followed minimization problem:

$$\underset{\beta, b}{\operatorname{argmin}} \sum_{t=1}^{n-1} ||\mathcal{H}_{t+1}(b) - \mathcal{H}_t(b + \beta t)||. \tag{9}$$

To efficiently solve problem (9), the n-frame 1D histograms $\mathcal{H}_t$ are sequentially concatenated into a 2D histogram $\mathsf{M} = [\mathcal{H}_1, \cdots, \mathcal{H}_n]$ with size $k \times n$, as illustrated in the third column of Fig. 5. Let a line L in the 2D histogram be defined by

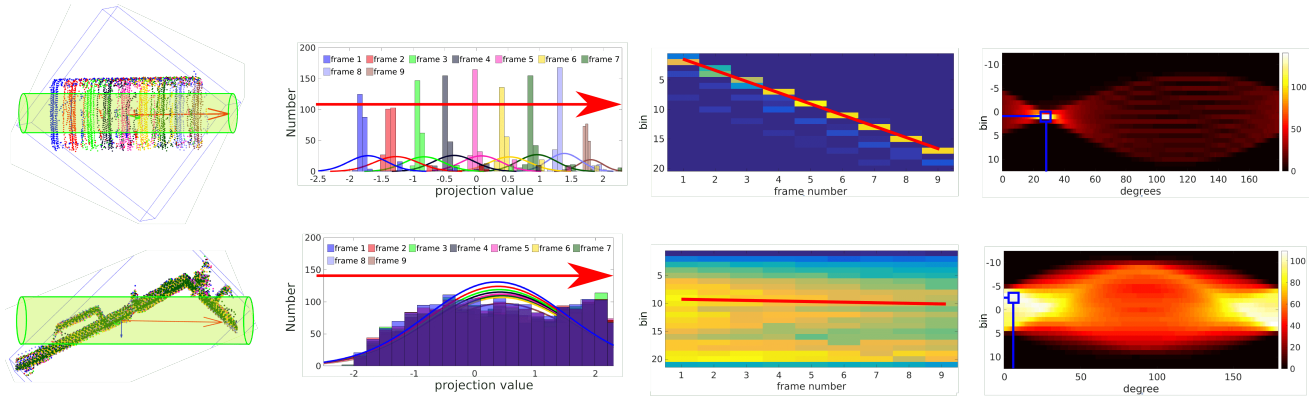

Fig. 5. Motion and static flow analysis: Row 1 and Row 2 are the graphical representations of the flow field analysis of a moving object and a static object, respectively. In comparison, Col. 1 shows the enclosing cylinder preserving the local structure. Col. 2 shows the 20-bin 1D histograms of cylinder-point projections of each frame. Remarkably, the histograms of motion flow (upper) are shifted along the flow direction, while the histograms of static flow (lower) are overlaid together. In Col. 3 are the concatenated all-frame histograms from Col. 2. The motion line L^* (solid red line) is estimated using the Radon transform in Col. 4 according to the criteria of Eq. (11).

$L(t) = \beta t + b$, with slope β and offset b. Note that the sought line L goes through the centres of the n-frame 1D histograms. In this regards, the optimal parameters β^* and b^* are obtained by

$$L^* = \underset{\beta,b}{\operatorname{argmax}} \int \mathcal{H}_t(L(t))\, dt. \tag{10}$$

Thanks to the Radon transform Deans [2007], which computes the volumetric integration in different angles at different positions in a continuous manner, problem (10) can be solved efficiently and globally by applying the Radon transform on M, as illustrated in the last column of Fig. 5. Three measurements are made along the line L^* to categorize the point sets as static or dynamic. Firstly, the slope β^* represents the magnitude of the motion speed, β^* of a static point set is very small accordingly. Further, let $s_t = \mathcal{H}_t(L^*), t = 1, \ldots, n$ be the values $\mathcal{H}_t(b)$ on the line L^*, two measurements are defined:

$$S = \sum_{t=1}^{n} s_t \quad \text{and} \quad E = -\sum_{t=1}^{n} s_t log(s_t). \tag{11}$$

Where S and E measure the strength and distribution homogeneity, respectively. A point set is considered to be static, if β^*, S and E values are below their respective thresholds. Otherwise, the point set is assumed to be dynamic.

C. Dynamic Neighbourhood Search

Practical scenarios, in which the sizes and the speeds of objects may significantly vary (*i.e.* from pedestrians to trucks), impose to the scene analysis in a dynamic manner. A default size of the local bounding box may cover only a small (or respectively too large) part of the object. This problem can be effectively addressed by taking a relatively large bounding box with a radius-variance enclosing cylinder, where the cylinder radius is inversely proportional to the number of points within the enclosing cylinder.

Our analysis algorithm is mostly driven by three parameters, namely the size of bounding box, its location, and the radius of the enclosing cylinder. To point out, we apply the bounding box (rather than an ellipsoid) for fast neighbourhood searching of Eq. (7) on the local flow field estimation. These three parameters can be reduced to two by taking a fix-sized bounding box with the radius as a ratio of its size. A motion is considered as "slow" when the sequential point sets S are totally bounded by the pre-defined bounding box. Consequently, the slow motions are not problematic because the corresponding point sets remain in the same bounding box. Otherwise, the bounding box is translated along the motion flow direction to obtain a larger coverage. As soon as the consecutive frames have led to a coherent motion, the local neighbourhood is updated, as illustrated in Fig. 6. In this figure, the bounding box is supposed to cover 9 consecutive frames for the object's motion analysis, however, only 5 consecutive frames are covered within the given sized bounding box due to the large displacement of the moving object. In order to achieve a larger coverage, the bounding box is translated along the motion flow direction. Followed by, the enclosing cylinder is applied for the object's motion behaviour study. Regarding to the parameter reduction, it is sufficient to choose a radius that is 20% smaller than the size of the bounding box according to our experiments. Moreover, this radius is proportionally adapted to the distance between the object and the camera.

Formally, we use a dynamic searching strategy along the flow direction. Let $\mathcal{B} = \{\mathsf{B}_t, t = 1, \ldots, f\}$ be the assembly of f frames of point sets within a local bounding box. When a fast motion occurs, the bounding box (centred at x_c) covers f frames with $f < n$, where n is the objective frame length for motion analysis. Let $\mathscr{P}_t(\mathsf{B}_t, \mathsf{w}), t = 1, \ldots, f$ be the projections of $\mathcal{B}$ along the motion direction w, and $\delta_t = median(\mathscr{P}_t), t = 1, \ldots, f$ be the median values of projections of $\mathscr{P}_t$. The bounding box is translated to $\mathsf{x}_t = \mathsf{x}_c + \delta_t \mathsf{w}$, until all n frames are covered.

D. Implementation Details and Discussions

Starting with the camera ego-motion compensation, the ICP-based point cloud registration algorithms are applied to register the given n consecutive frames of point sets. Notably, robust

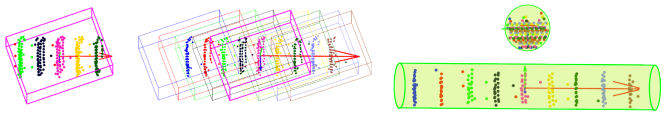

Fig. 6. Dynamic local neighbourhood search of a fast moving object: Left image shows that the fix-sized bounding box covers only 5 consecutive frames. By translating the bounding box along the flow direction, 9 consecutive frames are covered, see the middle image. To this end, the enclosing cylinder with all neighbouring frames are applied for accurate motion analysis, as shown in the right image.

ICP algorithms Fitzgibbon [2003]; Zhang and Singh [2016] are preferred to obtain precise camera motion estimation. According to our expertise, ICP registration on edge and plane feature points generally yields satisfactory results, similarly to Zhang and Singh [2016]. Taking the registered point sets as input, Algo. 1 is applied to discriminate the static and the dynamic points, and to estimate the motion flows of the dynamic points. For the sake of computational efficiency, the points from ground plane are detected and removed beforehand. Note that the detection of ground plane for the data acquired by a ground-vehicle is an almost solved problem [Douillard et al., 2011; Zermas et al., 2017]. In step 4, the enclosing cylinder radius is defined as $r = 0.4(1 + d/D)$, where d is the object to camera distance and D is the camera's maximum data acquisition distance (*e.g.* $D = 100$ meters for Velodyne HDL-64 [Lidar, 2016] 3D laser scanner). In step 7, τ_S is defined as 40% of the total number of neighbours within the enclosing cylinder (also known as the sum value of the 2D histogram M). $\tau_\beta = 0.175$ denotes that the slope of L^* is 10 degree. $\tau_E = 1.8$ is empirically studied and used for all our experiments.

We recall that the Radon transform calculates the volumetric integration in both angular and positional domains. Thus, its maximum response complies to the sought optimal solution of problem (10). In Fig. 5 Col. 2, the 1D histograms from dynamic scene part have shifting effects along the flow direction, as expected. Differently, these histograms tend to overlap with each other for the static scene parts. These phenomena lead to the different properties (refer to the above discussions in Section III-B) of the motion line L^* of static and dynamic points.

Algorithm 1: Motion Flow Identification.

Data: Point sets $\mathsf{S} = \{\mathsf{X}_1, \cdots, \mathsf{X}_n\}$, where the centre point set at $t = \frac{n}{2}$ is noted as $\bar{\mathsf{X}}$. The size of local neighbourhood is notated as $\mathcal{N}$.

1 **Setting**: $n = 9$, $k = 20$, bounding box size $4m \times 4m \times 4m$, $\tau_\beta = 0.175$, $\tau_S = 0.4\mathcal{N}$, $\tau_E = 1.8$.

2 **for** $\mathsf{x}_i \in \bar{\mathsf{X}}$ **do**

3 Place a 3D bounding box at x_i for local flow field estimation (W) using Eq. (6), and perform eigen-decomposition: $[\mathsf{V}, \mathsf{D}] = \text{eigen}(\mathsf{W}^\mathsf{T}\mathsf{W})$ to obtain the dominant flow $\mathsf{v} = \mathsf{V}(:, 3)$.

4 Fit an enclosing cylinder $\Theta(\mathsf{x}_i, \mathsf{X}, \mathsf{v}, r)$.

5 Project cylinder points to axis v using Eq. (4), and compute histograms $\mathcal{H}_t, t = 1, \ldots, n$ to construct M.

6 Compute the slope β^* of L^* using Radon transform on M, motion strength S and stability E using Eq. (11).

7 If $\beta^* < \tau_\beta$, $S < \tau_S$ and $E < \tau_E$, reject static point x_i.

Result: Detected motion flow set Ω.

IV. 3D Sparse Flow Clustering

In this section, we introduce our 3D SFC algorithm to further analyse the object's motion behaviour. By obtaining a set of dynamic points and their corresponding motion flows, as discussed in the above Section III, the 3D SFC intends to cluster them into multiple subsets w.r.t. their motion properties, *i.e.* similar motion speed, alike motion direction, and small spatial distance. Our clustering process uses the information from space subset S as well as their corresponding assignment vectors Z. On the one hand, we rely on the assumption that the vectors from one cluster are self-expressive. In other words, a flow vector can be closely approximated by the linear combination of the other flow vectors from the same cluster. On the other hand, we ensure that the clustered vector fields have bounded space subset within the predefined radius.

Let $\mathsf{X} = [\mathsf{x}_1, \ldots \mathsf{x}_j, \ldots \mathsf{x}_n]$ and $\mathsf{W} = [\mathsf{w}_1, \ldots \mathsf{w}_j, \ldots \mathsf{w}_n]$ are $3 \times n$ matrices of the point set and the corresponding flow vectors, the self-expressive sparse representation (similar to Elhamifar and Vidal [2013]) can be written as

$$\mathsf{W} = \mathsf{W}\mathsf{C}, \tag{12}$$

where the sparse $n \times n$ matrix $\mathsf{C} = [\mathsf{c}_1, \ldots \mathsf{c}_j, \ldots \mathsf{c}_n]$ with $c_{jj} = 0$ to avoid trivial solutions, for all $j = 1, \ldots n$. Similarly, for a predefined squared radius bound ϵ_r (where the sparsity comes from), the bounded space subset is ensured by enforcing the constraint

$$\|\mathsf{x}_j - \mathsf{X}\mathsf{c}_j\|_2^2 \leq \epsilon_r, \forall j. \tag{13}$$

Therefore, the sparsity-constraint relaxed optimization problem for flow clustering can be written as

$$\begin{aligned} \underset{\mathsf{C}}{\text{minimize}} \quad & \|\mathsf{C}\|_{1,1}\,, \\ \text{subject to} \quad & \mathsf{W} = \mathsf{W}\mathsf{C}, \qquad \text{diag}(\mathsf{C}) = 0, \\ & \|\mathsf{x}_j - \mathsf{X}\mathsf{c}_j\|_2^2 \leq \epsilon_r, \qquad \forall j. \end{aligned} \tag{14}$$

This is a convex problem, whose optimal solution can be found by using the second order cone programming Boyd and Vandenberghe [2004]. In fact, its equivalent problem as the semi-definite programming is given by

$$\begin{aligned} \underset{\mathsf{C},\mathsf{S}}{\text{minimize}} \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij} \\ \text{subject to} \quad & \mathsf{W} = \mathsf{W}\mathsf{C}, && \text{diag}(\mathsf{C}) = 0, \\ & -s_{ij} \leq c_{ij} \leq s_{ij}, && \forall\{i,j\}, \\ & \begin{pmatrix} \mathsf{I} & \mathsf{x}_j - \mathsf{X}\mathsf{c}_j \\ (\mathsf{x}_j - \mathsf{X}\mathsf{c}_j)^\mathsf{T} & \epsilon_r \end{pmatrix} \succeq 0, && \forall j, \end{aligned} \tag{15}$$

where s_{ij} are the elements of S.

A. Influence of Noise and Outliers

In practical scenarios, the flow data might be contaminated by noise or outliers. Let

$$\mathsf{w}_j = \mathsf{w}_j^0 + e_j, \tag{16}$$

where $e_j \in \mathbb{R}^3$ is the noise or outlier entry of noise free data w_j^0. Replacing Eq. (12) with Eq. (16), we have

$$\mathsf{W} = \mathsf{W}\mathsf{C} + \mathsf{E}. \tag{17}$$

Due to the local structure persistence and temporal flow speed consistency assumptions, the sought sparse representation from the current frame is valid for the neighbor frames. Therefore, the sparse subspace clustering problem of Eq. (15) can be reformulated as:

$$\begin{aligned} \underset{\mathsf{C},\mathsf{E}_\mathsf{x},\mathsf{E}_\mathsf{w},}{\text{minimize}} \quad & \|\mathsf{C}\|_{1,1} + \mathsf{E}_\mathsf{w} + \mathsf{E}_\mathsf{x}, \\ \text{subject to} \quad & \|\mathsf{w}_j - \mathsf{W}_t\mathsf{c}_j\|_2^2 \le \epsilon_\mathsf{w}, \quad \forall j,\ c_{jj} = 0, t = 1, \cdots, n, \\ & \|\mathsf{x}_j - \mathsf{X}_t\mathsf{c}_j\|_2^2 \le \epsilon_\mathsf{x}, \quad \forall j,\ c_{jj} = 0, t = 1, \cdots, n, \end{aligned} \tag{18}$$

where X_t and W_t are the 3D points and their flow vectors at frame t, respectively. In Eq. (18), $\mathsf{E}_\mathsf{w} = \lambda_1 \sum_{j=1}^n e_\mathsf{w}$ and $\mathsf{E}_\mathsf{x} = \lambda_2 \sum_{j=1}^n e_\mathsf{x}$ are energy terms with weight parameters λ_1 and λ_2 to control the influence of spatial (*i.e.* the 3D point coordinate) and temporal (*i.e.* the motion flow) factors, and we simply set $\lambda_1 = 1$ and $\lambda_2 = 1$. Note that the squared radius bound ϵ_w and ϵ_x are constrained to be non-negative, but not predefined. Similarly, Eq. (18) can be solved as a semi-definite programming problem.

Algorithm 2: Sparse Flow Clustering.

Data: 3D point sets $\bigcup_{t=1}^n \mathsf{X}_t$ and flows $\bigcup_{t=1}^n \mathsf{W}_t$.
Result: k clustered subspaces.
1 Sparse flow representation using Equation (18).
2 Sparse similarity graph construction: $\mathcal{G} = |\mathsf{C}^*| + |\mathsf{C}^*|^\mathsf{T}$.
3 K-mean spectral clustering on $\mathcal{G}$.

B. Spectral Clustering

Getting the sparse subspace representation matrix C, a sparse symmetric similarity graph, which stands for the connectivity among the flows, can be constructed as $\mathcal{G} = |\mathsf{C}| + |\mathsf{C}|^\mathsf{T}$. To group the flows into their corresponding motions, a spectral clustering approach [Atev et al., 2010; Ng et al., 2002] can be applied on $\mathcal{G}$ to segment the motion flows into their individual groups, see Fig. 7.

In this figure, the top-left image shows a sequence of registered 3D point clouds where several moving objects exist, specifically two moving cars, three walking pedestrians, and a cyclist. The bottom-left image is the zoom-in view of the detected motion flows on a moving car. The middle image demonstrates a clustered connectivity graph $\mathcal{G}$ which reveals the relationship between each 3d flow. Note that $\mathcal{G}$ is derived from the sparse representation matrix C, it is expected that each independent motion forms one diagonal block with sparse non-zero entries. The size of the diagonal block is determined by the cluster's element number relating to the object's size and the density of point set. To this end, the right image shows the color-coded motion cluster in a more illustrative manner.

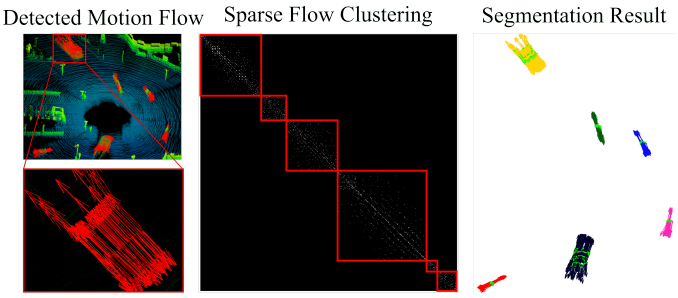

Fig. 7. Sparse Flow Clustering Illustration: the left image show the detected motion flows of 6 different moving objects of various sizes and velocities; the middle image shows the block-diagonal clusters corresponding to their motion subspaces; the right image are the color-coded motion flows.

C. Implementation Details and Discussions

The SFC algorithm consists of three major steps (see Algo. 2) which are implemented based on the CVX Grant and Boyd [2008] optimization toolbox. In the sparse optimization step, a point to point distance graph $\mathcal{G}_s = [\mathsf{g}_1, \cdots \mathsf{g}_j, \cdots, \mathsf{g}_n] \in \mathbb{R}^{n \times n}$ is applied to enforce the spatial closeness of the selected sparse representation elements, such that Eq. (18) becomes

$$\mathsf{W} = \mathsf{W}(\mathsf{C} \cdot \mathcal{G}) + \mathsf{E}, \quad \forall \mathcal{G}_{ij} > \tau_d, \mathcal{G}_{ij} = 1, \text{else } \mathcal{G}_{ij} = 0. \tag{19}$$

Where operator $(\cdot)$ stands for the dot product, and τ_d is the point-to-point spatial distance threshold. Two major remarks on spatial distance constraint can be made: a) It is more meaningful to use sparse representation only on the local neighbourhood. b) Exploiting the sparsity of C improves the algorithm's robustness and computational efficiency.

In step 2 of Algo. 2, a sparse symmetric similarity graph $\mathcal{G} = |\mathsf{C}^*| + |\mathsf{C}^*|^\mathsf{T}$ is constructed. Since $\mathcal{G}$ encodes the connectivity information among the flows, a K-mean spectral clustering is employed to group the flow clusters. In fact, K can be determined by finding the number of graph components via the analysis of the eigenspectrum of the Laplacian matrix of $\mathcal{G}$ [Von Luxburg, 2007]. However, other model selection techniques [Brox and Malik, 2010] should be employed when there are connections between points in different subspaces. In the following experiments, we provide the number of motions as an input to all the algorithms for fair comparison.

To emphasize, the proposed SFC does not rely on feature tracking and feature trajectory construction (unlike [Elhamifar and Vidal, 2013; Hu et al., 2014; Jiang et al., 2016]), making it more appropriate for highly dynamic environment motion analysis. Moreover, the SFC algorithm, which is proposed under the robust sparse subspace representation framework, offers new research perspectives for vector field analysis.

V. Experiments

We evaluate the proposed algorithms by conducting extensive experiments on the challenging real-world KITTI benchmark Geiger et al. [2013] with rapidly changing environments. The seven representative datasets (namely Campus, Cola Truck, Junction, Market, Pedestrian, Red Light, and Station) have been carefully selected to cover a wide range of moving

Sequence	# Frms.	# Objs.	2D-SSC			3D-SSC			2D-SMR J1			2D-SMR J2			3D-MOD		
			Sens.	Spec.	Time	Sens.	Spec.	Time	Sens.	Spec.	Time	Sens.	Spec.	Time	Sens.	Spec.	Time
Campus	60	4	0.858	**0.994**	31.84	0.871	0.947	33.02	0.854	0.986	0.032	0.856	0.991	0.036	**0.914**	0.982	5.43
ColaTruck	50	2	**0.940**	0.306	21.93	0.845	0.949	52.39	0.356	0.808	0.032	0.360	0.749	0.038	0.798	**0.966**	5.05
Junction	90	3	0.908	0.820	24.08	0.892	0.943	38.40	0.768	0.937	0.039	0.774	0.920	0.042	**0.983**	**0.997**	5.68
Market	100	6	0.735	0.929	21.33	0.770	0.920	37.31	0.861	0.823	0.053	0.826	0.883	0.043	**0.913**	**0.994**	5.07
Pedestrian	140	6	0.900	0.896	32.57	0.927	0.918	35.12	0.908	0.905	0.039	0.870	0.914	0.047	**0.928**	**0.974**	6.01
Red Light	120	4	0.937	**0.999**	33.25	**0.941**	0.985	31.40	0.928	0.921	0.036	0.918	0.976	0.042	0.916	0.985	5.22
Station	50	5	**0.866**	0.963	39.50	0.850	0.964	45.09	0.916	0.814	0.041	0.908	0.847	0.051	0.862	**0.993**	6.50
Average	87	4	0.878	0.893	29.32	0.874	0.949	38.79	0.799	0.876	0.039	0.793	0.897	0.043	**0.901**	**0.985**	5.57

TABLE I
MOTION OBJECT DETECTION PERFORMANCE QUANTIFICATION ON THE KITTI BENCHMARK: COL. 1-3 ARE THE SEQUENCE NAME, FRAME LENGTH AND AVERAGE MOVING OBJECT NUMBER, RESPECTIVELY. THE REST COLUMNS SUMMARIZE THE SENSITIVITY, SPECIFICITY AND PROCESSING TIME (IN SECOND UNIT) OF THE COMPARED ALGORITHMS ON SEVEN DIFFERENT DATASETS, WHILE THE LAST ROW AVERAGES THEIR OVERALL PERFORMANCES. THE HIGHLIGHTED VALUES ARE THE BEST PERFORMANCES.

objects in terms of quantity, size, speed, shape, occlusion, etc. To test the flexibility of the algorithms in terms of camera motion, the Campus, Pedestrian and Station sequences are acquired from a static camera set-up, while the other sequences are acquired when the camera is moving. The detailed results are synthesized in Table I Col. 2-3 and Table III Col. 2-5. The performances with the state-of-the-art methods are assessed by using the *Sensitivity* and *Specificity* metrics [Fawcett, 2006], defined as follows:

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}, \tag{20}$$

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives + False Positives}}. \tag{21}$$

For comparison with MS-based methods, the misclassification rate metric suggested by Elhamifar and Vidal [2013]; Hu et al. [2014] is adopted. All the experiments have been conducted on a machine with Intel Quad Core i7-2.7GHz, 32GB Memory using MATLAB.

A. Motion Detection Evaluation

We compare the proposed 3D-based Moving Object Detection algorithm (3D-MOD) against the four representative algorithms available in the literature. Remind that the 2D-SMR, 2D-SSC and 3D-SSC are feature-based motion segmentation algorithms cluster the feature trajectories into their corresponding motions. We define: *True Positive* – if only a motion trajectory is NOT classified as background motion, and *True Negative* – when a background trajectory is classified as background motion. Here, we consider the feature trajectories belong to the static scene parts as background motion. When several motions are involved, although a feature trajectory might not be correctly classified into its corresponding motion, it is yet considered as a true positive.

Table I summarizes the quantitative evaluation of 2D-SMR-J1 Hu et al. [2014], 2D-SMR-J2 Hu et al. [2014], 2D-SSC Elhamifar and Vidal [2013], 3D-SSC Jiang et al. [2016] and 3D-MOD on the seven representative datasets by using the *Sensitivity* and *Specificity* metrics. There are several remarks listed as follows:

a. Both the 2D-SSC and the 3D-SSC algorithms achieve quite good performances in terms of sensitivity. Meanwhile, the 3D-SSC has much better specificity but rather low computational efficiency.
b. Although the 2D-SMR based approaches have the worst performance in both sensitivity and specificity, such approaches have the best time performance which offers great potential in real-time applications.
c. Overall, the 3D-MOD accomplishes quite decent performances in both sensitivity and specificity. Precisely, the 3D-MOD has a more superior averaged sensitivity, as well as a remarkably higher averaged specificity.
d. The 3D-based methods (*i.e.* 3D-SSC and 3D-MOD) exhibit very stable performances, especially much higher specificity, thanks to their insensitivity to perspective projection effects.
e. Regarding the computational efficiency, our 3D-MOD approach provides a compromised solution. In addition, it can be easily parallelized and boosted if online MOD is required.

Further, we adopt the mean and median *Misdetection Error* metrics defined by

$$\eta = \frac{\text{\# False Positive + \# False Negative}}{\text{\# Features}}, \tag{22}$$

as in Elhamifar and Vidal [2013]; Hu et al. [2014] for MOD performance evaluation, refer to Table II. Illustratively, the corresponding box-plot statistical comparisons are provided as in Fig. 8. Similarly, the 3D-SSC and 3D-MOD have noteworthy better performances than other methods due to their persistent high specificity. To point out, the 3D-MOD outperforms the other methods with clearly lower median misdetection rate as well as much higher robustness, as shown in Fig. 8.

Moreover, the 3D-MOD is compared against the Object Scene Flow (OSF) Menze et al. [2018] algorithm, as concluded in Table III. Since the OSF method produces pixel-level dense moving object detection and segmentation, it is more appropriate to compare their performances in a dense manner. In this regards, the 3D Region Growing Mühlenbruch et al. [2006] algorithm, seeded at the motion flows detected by the

Sequence	2D-SSC		3D-SSC		2D-SMR-J_1		2D-SMR-J_2		3D-MOD	
	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
Campus	0.067	0.063	0.096	0.067	0.071	0.066	0.067	0.064	**0.055**	**0.037**
ColaTruck	0.506	0.545	**0.092**	0.103	0.341	0.373	0.385	0.340	0.095	**0.097**
Junction	0.116	0.081	0.077	0.050	0.136	0.155	0.148	0.155	**0.008**	**0.007**
Market	0.174	0.162	0.139	0.124	0.175	0.148	0.146	0.152	**0.032**	**0.023**
Pedestrian	0.114	0.113	0.086	0.044	0.099	0.112	0.125	0.127	**0.038**	**0.033**
Red Light	0.037	0.032	**0.036**	0.033	0.087	0.046	0.064	0.044	0.052	**0.014**
Station	0.097	0.079	**0.086**	0.093	0.150	0.167	0.140	0.151	0.102	**0.045**

TABLE II
QUANTITATIVE EVALUATION ON KITTI DATASET: USING THE MEAN AND MEDIAN VALUES OF MISDETECTION RATE METRICS.

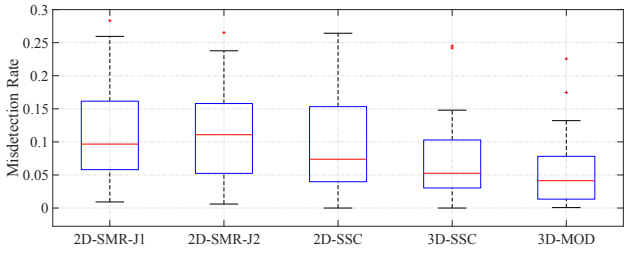

Fig. 8. Box-plot analysis of Misdetection Rate on KITTI dataset.

Sequence	Object Size		Speed		OSF			3D-MOD		
	Min.	Max.	Min.	Max.	Sens.	Spec.	Time	Sens.	Spec.	Time
Campus	527	17483	0.35	5.56	0.404	0.988	60.8	**0.928**	**0.993**	9.31
ColaTruck	3339	29795	4.87	7.22	0.579	**0.994**	66.1	**0.772**	0.936	28.8
Junction	1397	10479	3.50	16.7	0.613	0.966	73.9	**0.933**	**0.980**	27.2
Market	148	8310	0.35	1.34	0.506	**0.962**	72.2	**0.954**	0.944	26.2
Pedestrian	291	15344	0.35	5.56	0.519	**0.983**	69.5	**0.933**	0.982	11.6
Red Light	1149	3977	0.36	8.33	0.578	**0.987**	84.5	**0.937**	**0.987**	14.0
Station	4010	45473	0.35	7.12	0.164	**0.996**	71.3	**0.882**	0.972	29.2
Average	/	/	/	/	0.480	**0.982**	71.2	**0.906**	0.971	20.9

TABLE III
QUANTITATIVE EVALUATION ON OSF MENZE ET AL. [2018] AND 3D-MOD: COL. 2-5 INDICATE THE MINIMUM AND MAXIMUM SIZE (PIXEL) AND SPEED (m/s) OF MOVING OBJECTS, RESPECTIVELY.

3D-MOD, is applied to densely segment the moving objects. Thus, both the Sensitivity and Specificity are computed by using dense segmentation of 3D point clouds. From this table, we observe that the 3D-MOD is not only faster, but also consistently exhibits a much higher sensitivity with just a slightly lower specificity.

To conclude, the main reasons that the 3D-MOD surpasses the state-of-the-art methods are:

a) The 3D-MOD relies on a pre-registration of point clouds, while the motion segmentation-based methods utilize the raw feature trajectories without ego-motion compensation.

b) The 3D-MOD interprets the motions by using high quality 3D data, while the OSF estimates a low-precision 3D scene structure by using stereo vision techniques.

c) The 3D-MOD analyses the 3D motion behaviours under local flows consistency assumption, which addresses the problem in essence.

B. Motion Segmentation Evaluation

Quantitatively, we utilize the *Misclassification Rate* (same as η) to compare the performances of the different algorithms, as shown in Fig. 9. In most cases, the 2D-based approaches

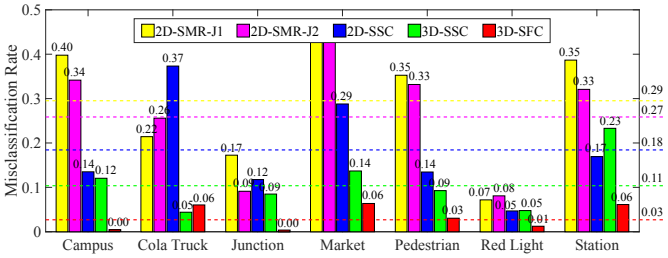

Fig. 9. Quantitative comparison of motion segmentation algorithms: dashed lines are their averaged misclassification rates.

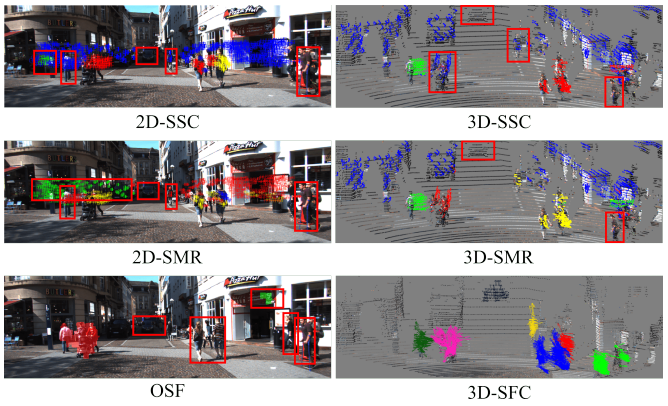

Fig. 10. Qualitative comparison of different motion segmentation approaches on Market sequence: left images are image-based motion segmentation results of 2D-SSC Elhamifar and Vidal [2013], 2D-SMR Hu et al. [2014], and OSF Menze et al. [2018], respectively. The right images are outcomes of 3D-based approaches, namely the 3D-SSC [Jiang et al., 2016], the 3D-SMR [Jiang et al., 2017b] and the proposed 3D-SFC. Red boxes highlight the undetected or incorrectly segmented motions.

achieve much higher misclassification rate, while the 3D-based algorithms obtain relatively lower misclassification rate. Furthermore, our 3D-SFC exceptionally outperforms the compared algorithms on the evaluated datasets.

Qualitatively, Fig. 10 illustrates the motion segmentation results of the compared algorithms on the Market sequence. Left column images exemplify that the 2D-based motion segmentation results are quite unsatisfactory, while the right column images of the 3D-based approaches manifest much better outcomes. Particularly, in the bottom-right image, all the moving objects are detected correctly, including the black car in the middle of the scene and the walking pedestrian on the left side. The excellent performance of the proposed 3D-

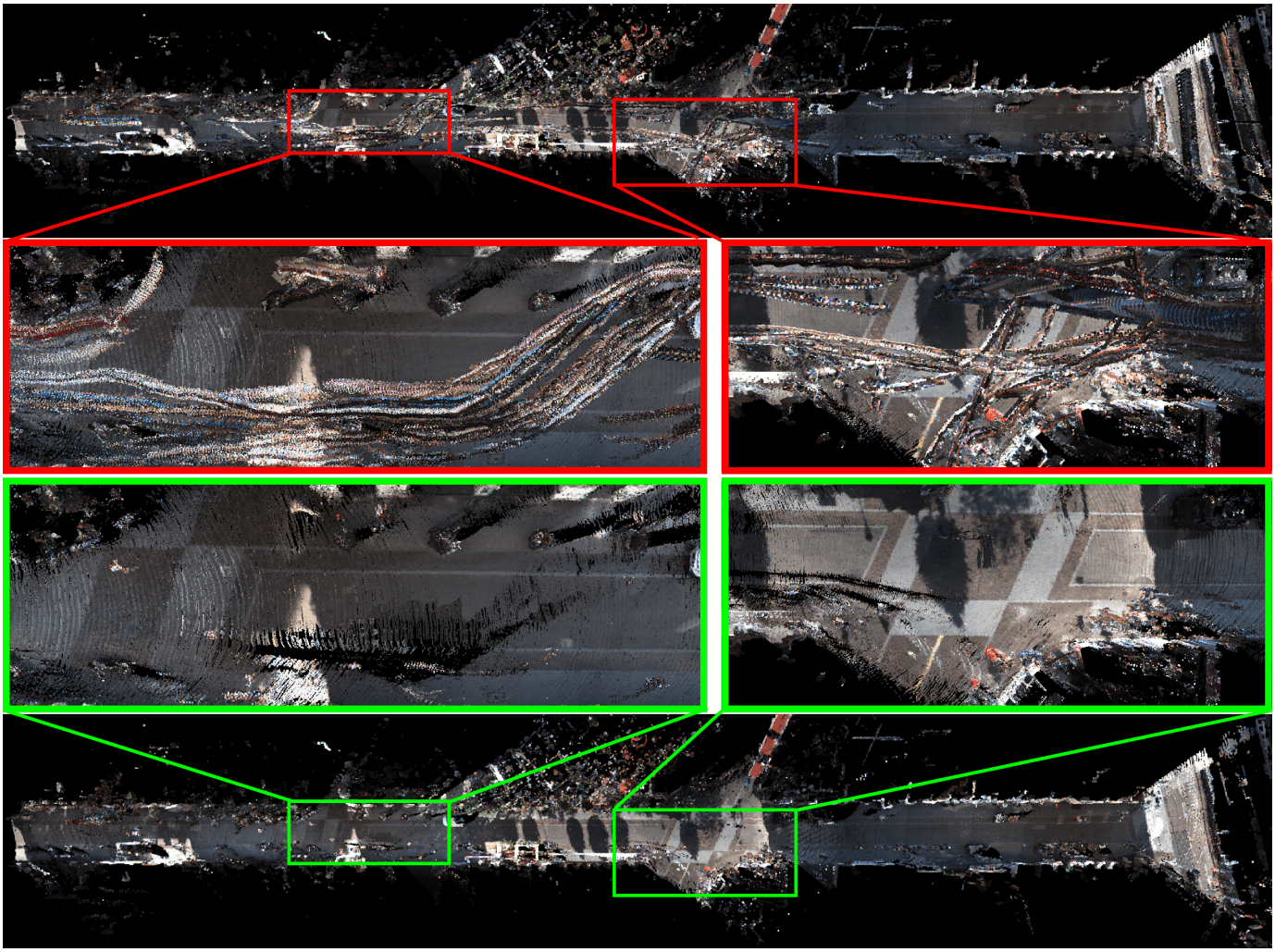

Fig. 11. Top image is the full scene 3D reconstruction by using Zhang and Singh [2016] of Market sequence where numerous moving objects occur. The zoom-in regions show the immense artifacts from the walking pedestrians. Bottom image is our static-map which has distinctively higher quality.

SFC mainly comes from:

a) The 3D-SFC classifies the detected motion flows from 3D-MOD, which contributes to the discard of most background features.

b)The 3D-SFC is proposed under the sparse representation framework with extra spatial closeness constraint, which produces a very reliable similarity graph for spectral clustering.

C. Static-map and Rigid Object Reconstruction

We conducted multiple experiments on the KITTI dataset and obtained quite better static-maps than the other approaches, as proved with the detailed tests and measures presented below. Fig. 11 shows the challenging Market sequence which contains a large amount of moving objects. The static-map produced by our framework is of significantly better quality because our framework is not sensitive to light changes, occlusions, slow or very fast motions, etc. Remarkably, top images in Fig. 11 contains serious "ghost" artefacts caused by the trajectories of moving objects, which not only degrades the visual quality but also defects the functionality of the reconstructed 3D map. For instance, these "ghost" artefacts occlude the ground areas and the lane markings in road surface, making the automatic or manual labelling in High Definition Map (HD-Map) production more difficult. Moreover, performances of map-based localization algorithms [Levinson et al., 2007; Magnusson et al., 2007; Wan et al., 2018] are expected to deteriorate due to the heavily-noise corrupted point cloud map.

By applying the proposed framework, the bottom images in Fig. 11 demonstrate that the 3D point cloud map contains only the stable objects, so-called static map. Such static map offers great potentials in applications such as city scene modelling [Babahajiani et al., 2017; Fan et al., 2009], automatic lane marking extraction in HD-Map production [Guan et al., 2015; Prochazka et al., 2019], landmark-based localization in autonomous driving [Lu et al., 2019], etc.

Moreover, the top-row images of Fig. 12 and 13 illustrate the superior quality of the synthetic images rendered by projecting the textured 3D point cloud of static maps onto a virtual camera coordinate. As can be seen from the bottom-row images of Fig. 12 and 13, many walking pedestrians are captured by the vehicle's camera. However, for some specific applications such as the Google Street View [Anguelov et al.,

Fig. 12. Synthetic image generation of market sequence by using the reconstructed static map. Top images illustrate the static scene imaginary of the market area, while the bottom images are "contaminated" by the moving car and the numerous walking pedestrians.

Fig. 13. Synthetic image generation of station sequence: top image is the rendered static scene imaginary by using the reconstructed static map. By comparing to the bottom real camera captured image, we notice that large moving objects (such as the moving car and train) are correctly detected and removed.

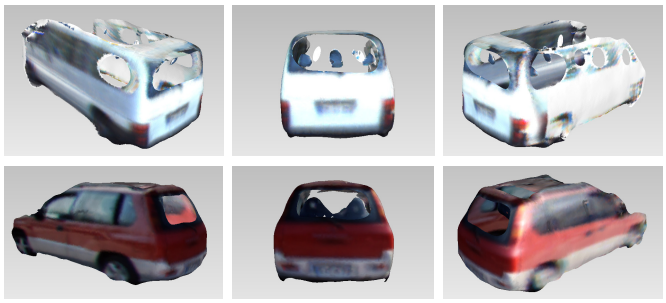

Fig. 14. Photo-realistic 3D reconstructions of individual moving objects.

2010; Mao et al., 2011], it is preferable to have a clean view of the city scene. Noteworthy, the synthetic image generation is, in essence, a key component of video inpainting technique [Newson et al., 2014; Zhang et al., 2019]. Moreover, these synthetic high quality images of static scenes can be used as reference images to label the moving objects, which makes the moving object annotation far more efficient and intelligent.

Apart from the static map reconstruction, getting the clustered motion trajectories from our framework, the 3D reconstruction of moving objects can be obtained by registering the observed sparse point clouds during their motions. Fig. 14 shows two reconstructed rigidly moving objects. Thanks to the proposed 3D-SFC, the detected moving objects can be separated according to their specific motion subspace. The detected moving objects are then individually registered with texture mapping to produce photo-realistic 3D modelling [Jiang et al., 2017a]. For more experimental results, readers are recommended to view this video (https://youtu.be/LewA8Lhn5Xo).

VI. Conclusion

We have proposed an original 3D Flow Field Analysis (3D-FFA) algorithm for 3D Moving Object Detection (3D-MOD) under the motion consistency assumption of a local neighbourhood. We further present a novel 3D Sparse Flow Clustering (3D-SFC) approach based on the self-expressiveness property of motion flow subspace as well as the spatial closeness constraint. By integrating the proposed 3D-MOD and 3D-SFC algorithms, we show that our framework is not merely robust, efficient and accurate, but also allows photo-realistic static-map and dynamic object reconstructions by using a 2D-3D moving camera system. In many aspects, both the 3D-MOD and 3D-SFC algorithms outperform the state-of-the-art methods since we have compared all these techniques on comprehensive highly dynamic real-world KITTI datasets, for which they consistently exhibit better accuracy, lower misclassication and misdetection rates, and consequently yield very high quality 3D reconstructions of static-maps as well as moving objects. In addition, the proposed framework offers great potentials in 3D city scene modelling, robot navigation and many other autonomous driving applications.

As for future perspectives, since the proposed 3D-FFA algorithm makes the assumption of linear motion, it may failed to detect pure rotation motions. Thus, a complementary algorithm in dealing with pure rotation motions is preferred. Moreover, it is interesting to produce higher resolution synthetic image sequence by incorporating the image inpainting techniques. Furthermore, recent advances, such as Point FlowNet3D [Behl et al., 2019; Liu et al., 2019a], achieve very interesting results in estimating 3D scene flows, which should benefit to a better performance in motion flow field analysis as long as higher computational cost is inessential.

REFERENCES

D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver. Google street view: Capturing the world at street level. *Computer*, 43(6):32–38, 2010.

A. Asvadi, P. Peixoto, and U. Nunes. Detection and tracking of moving objects using 2.5 d motion grids. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 788–793. IEEE, 2015.

S. Atev, G. Miller, and N. P. Papanikolopoulos. Clustering of vehicle trajectories. *IEEE transactions on intelligent transportation systems*, 11(3):647–657, 2010.

A. Azim and O. Aycard. Detection, classification and tracking of moving objects in a 3d environment. In *2012 IEEE Intelligent Vehicles Symposium*, pages 802–807. IEEE, 2012.

P. Babahajiani, L. Fan, J.-K. Kämäräinen, and M. Gabbouj. Urban 3d segmentation and modelling from street view images and lidar point clouds. *Machine Vision and Applications*, 28(7):679–694, 2017.

A. Behl, D. Paschalidou, S. Donné, and A. Geiger. Pointflownet: Learning representations for rigid motion estimation from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7962–7971, 2019.

Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Comparative study of background subtraction algorithms. *Journal of Electronic Imaging*, 19(3): 033003, 2010.

A. Börcs, B. Nagy, and C. Benedek. Instant object detection in lidar point clouds. *IEEE Geoscience and Remote Sensing Letters*, 14(7):992–996, 2017.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *European conference on computer vision*, pages 282–295. Springer, 2010.

H. Cho, Y.-W. Seo, B. V. Kumar, and R. R. Rajkumar. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1836–1843. IEEE, 2014.

S. R. Deans. *The Radon transform and some of its applications*. Courier Corporation, 2007.

A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard. Motion-based detection and tracking in 3d lidar scans. 2016.

B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel. On the segmentation of 3d lidar point clouds. In *2011 IEEE International Conference on Robotics and Automation*, pages 2798–2805. IEEE, 2011.

S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed. Moving object detection in spatial domain using background removal techniques-state-of-art. *Recent patents on computer science*, 1(1):32–54, 2008.

E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1355–1361. IEEE, 2017.

H. Fan and Y. Yang. Pointrnn: Point recurrent neural network for moving point cloud processing. *arXiv preprint arXiv:1910.08287*, 2019.

H. Fan, L. Meng, and M. Jahnke. Generalization of 3d buildings modelled by citygml. In *Advances in GIScience*, pages 387–405. Springer, 2009.

T. Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

A. W. Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and vision computing*, 21(13-14):1145–1153, 2003.

A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, pages 1–16, 2013.

M. Grant and S. Boyd. Cvx: Matlab software for disciplined convex programming, 2008.

H. Guan, J. Li, Y. Yu, Z. Ji, and C. Wang. Using mobile lidar data for rapidly updating road markings. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2457–2466, 2015.

M. Heikkila and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):657–662, 2006.

B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

H. Hu, Z. Lin, J. Feng, and J. Zhou. Smooth representation clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3834–3841, 2014.

J. Huang, W. Zou, J. Zhu, and Z. Zhu. Optical flow based real-time moving object detection in unconstrained scenes. *arXiv preprint arXiv:1807.04890*, 2018.

C. Jiang. *Motion Analysis for Dynamic 3D Scene Reconstruction and Understanding*. PhD thesis, 2017.

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 Robotics and Automation Letters*, 1(1):324–331, January 2016.

C. Jiang, D. Christie, D. P. Paudel, and C. Demonceaux. High quality reconstruction of dynamic objects using 2d-3d camera fusion. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2209–2213. IEEE, 2017a.

C. Jiang, D. P. Paudel, Y. Fougerolle, D. Fofi, and C. Demonceaux. Incomplete 3d motion trajectory segmentation and 2d-to-3d label transfer for dynamic scene analysis. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 606–613. IEEE, 2017b.

C. Jiang, D. P. Paudel, Y. Fougerolle, D. Fofi, and C. Demonceaux. Static and dynamic objects analysis as a 3d vector field. In *2017 International Conference on 3D Vision (3DV)*,

pages 234–243. IEEE, 2017c.

K. A. Joshi and D. G. Thakore. A survey on moving object detection and tracking in video surveillance system. *International Journal of Soft Computing and Engineering*, 2(3):44–48, 2012.

B. Jung and G. S. Sukhatme. Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In *International conference on intelligent autonomous systems*, pages 980–987. Citeseer, 2004.

M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele. Motion segmentation & multiple object tracking by correlation co-clustering. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):140–153, 2018.

D. Kochanov, A. Osep, J. Stückler, and B. Leibe. Scene flow propagation for semantic mapping and object discovery in dynamic street scenes. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2016.

B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *IEEE transactions on pattern analysis and machine intelligence*, 30(10):1683–1698, 2008.

J. Levinson, M. Montemerlo, and S. Thrun. Map-based precision vehicle localization in urban environments. In *Robotics: science and systems*, volume 4, page 1. Citeseer, 2007.

J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168. IEEE, 2011.

Q. Li, S. Chen, C. Wang, X. Li, C. Wen, M. Cheng, and J. Li. Lo-net: Deep real-time lidar odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8473–8482, 2019.

V. Lidar. Hdl-64e, 2016.

J. Limb and J. Murphy. Estimating the velocity of moving images in television signals. *Computer graphics and image processing*, 4(4):311–327, 1975.

X. Liu, C. R. Qi, and L. J. Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 529–537, 2019a.

X. Liu, M. Yan, and J. Bohg. Meteornet: Deep learning on dynamic 3d point cloud sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9246–9255, 2019b.

W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song. L3-net: Towards learning based lidar localization for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6389–6398, 2019.

W.-C. Ma, S. Wang, R. Hu, Y. Xiong, and R. Urtasun. Deep rigid instance scene flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3614–3622, 2019.

M. Magnusson, A. Lilienthal, and T. Duckett. Scan registration for autonomous mining vehicles using 3d-ndt. *Journal of Field Robotics*, 24(10):803–827, 2007.

B. Mao, L. Harrie, Y. Ban, and H. Fan. Real time visualisation of 3d city models in street view based on visual salience. *International Journal of Geographical Information Science*, 2011.

E. Mémin and P. Pérez. Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, 46(2):129–155, 2002.

M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015.

M. Menze, C. Heipke, and A. Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:60–76, 2018.

C. Mertz, L. E. Navarro-Serment, R. MacLachlan, P. Rybski, A. Steinfeld, A. Suppé, C. Urmson, N. Vandapel, M. Hebert, C. Thorpe, et al. Moving object detection with laser scanners. *Journal of Field Robotics*, 30(1):17–43, 2013.

A. Mitiche and H. Sekkati. Optical flow 3d segmentation and interpretation: A variational method with active curve evolution and level sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1818–1829, 2006.

G. Mühlenbruch, M. Das, C. Hohl, J. E. Wildberger, D. Rinck, T. G. Flohr, R. Koos, C. Knackstedt, R. W. Günther, and A. H. Mahnken. Global left ventricular function in cardiac ct. evaluation of an automated 3d region-growing segmentation algorithm. *European radiology*, 16(5):1117–1123, 2006.

R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136. IEEE, 2011.

A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez. Video inpainting of complex scenes. *SIAM Journal on Imaging Sciences*, 7(4):1993–2019, 2014.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, et al. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167, 2008.

J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

D. Prochazka, J. Prochazkova, and J. Landa. Automatic lane marking extraction from point cloud into polygon map layer. *European Journal of Remote Sensing*, 52(sup1):26–39, 2019.

S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1832–1845, 2010.

H. Rashed, M. Ramzy, V. Vaquero, A. El Sallab, G. Sistu,

and S. Yogamani. Fusemodnet: Real-time camera and lidar based moving object detection for robust low-light autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

K. S. Ray and S. Chakraborty. Object detection by spatio-temporal analysis and tracking of the detected objects in a video with variable background. *Journal of Visual Communication and Image Representation*, 58:662–674, 2019.

V. Reilly, H. Idrees, and M. Shah. Detection and tracking of large number of targets in wide area surveillance. In *European conference on computer vision*, pages 186–199. Springer, 2010.

H. G. Seif and X. Hu. Autonomous driving in the icity hd maps as a key challenge of the automotive industry. *Engineering*, 2(2):159–162, 2016.

Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1219–1225. IEEE, 2009.

D. Steinhauser, O. Ruepp, and D. Burschka. Motion segmentation and scene classification from 3d lidar data. In *2008 IEEE Intelligent Vehicles Symposium*, pages 398–403. IEEE, 2008.

M. Sualeh and G.-W. Kim. Dynamic multi-lidar based multiple object detection and tracking. *Sensors*, 19(6):1474, 2019.

A. Takabe, H. Takehara, N. Kawai, T. Sato, T. Machida, S. Nakanishi, and N. Yokoya. Moving object detection from a point cloud using photometric and depth consistencies. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 561–566. IEEE, 2016.

S. Vedula, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2005.

R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1945–1959, 2005.

R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using powerfactorization and gpca. *International Journal of Computer Vision*, 79(1):85–105, 2008.

U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

G. Wan, X. Yang, R. Cai, H. Li, Y. Zhou, H. Wang, and S. Song. Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4670–4677. IEEE, 2018.

C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, 26(9):889–916, 2007.

D. Z. Wang, I. Posner, and P. Newman. What could move? finding cars, pedestrians and bicyclists in 3d laser data. In *2012 IEEE International Conference on Robotics and Automation*, pages 4038–4044. IEEE, 2012.

D. Z. Wang, I. Posner, and P. Newman. Model-free detection and tracking of dynamic objects with 2d lidar. *The International Journal of Robotics Research*, 34(7):1039–1063, 2015.

A. Wedel, A. Meißner, C. Rabe, U. Franke, and D. Cremers. Detection and segmentation of independently moving objects from dense scene flow. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 14–27. Springer, 2009.

C. Wen, Y. Dai, Y. Xia, Y. Lian, J. Tan, C. Wang, and J. Li. Toward efficient 3-d colored mapping in gps-/gnss-denied environments. *IEEE Geoscience and Remote Sensing Letters*, 17(1):147–151, 2019.

J. Yan and M. Pollefeys. Articulated motion segmentation using ransac with priors. In *Dynamical Vision*, pages 75–85. Springer, 2007.

M. Yazdi and T. Bouwmans. New trends on moving object detection in video images captured by a moving camera: A survey. *Computer Science Review*, 28:157–177, 2018.

Y. Ye, L. Fu, and B. Li. Object detection and tracking using multi-layer laser for autonomous urban driving. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 259–264. IEEE, 2016.

M. Yokoyama and T. Poggio. A contour-based moving object detection and tracking. In *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 271–276. IEEE, 2005.

K. Yun, J. Lim, and J. Y. Choi. Scene conditional background update for moving object detection in a moving camera. *Pattern Recognition Letters*, 88:57–63, 2017.

D. Zermas, I. Izzat, and N. Papanikolopoulos. Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5067–5073. IEEE, 2017.

J. Zhang and S. Singh. Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, pages 1–16, 2016.

J. Zhang and S. Singh. Laser–visual–inertial odometry and mapping with high robustness and low drift. *Journal of Field Robotics*, 35(8):1242–1264, 2018.

R. Zhang, W. Li, P. Wang, C. Guan, J. Fang, Y. Song, J. Yu, B. Chen, W. Xu, and R. Yang. Autoremover: Automatic object removal for autonomous driving videos. *arXiv preprint arXiv:1911.12588*, 2019.

X. Zhou, C. Yang, and W. Yu. Moving object detection by detecting contiguous outliers in the low-rank representation. *IEEE transactions on pattern analysis and machine intelligence*, 35(3):597–610, 2012.

A. Ziebinski, R. Cupek, D. Grzechca, and L. Chruszczyk. Review of advanced driver assistance systems (adas). In *AIP Conference Proceedings*, volume 1906, page 120002. AIP Publishing LLC, 2017.