Static and Dynamic Objects Analysis as a 3D Vector Field

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

Static and Dynamic Objects Analysis as a 3D Vector Field

Cansen Jiang*, Yohan Fougerolle, David Fofi, Cedric Demonceaux
Le2i, FRE CNRS 2005, Arts et Metiers, Université Bourgogne Franche-Comté, France
cansen.jiang@u-bourgogne.fr

Danda Pani Paudel
Computer Vision Lab, ETH Zurich, Switzerland

Abstract

In the context of scene modelling, understanding, and landmark-based robot navigation, the knowledge of static scene parts and moving objects with their motion behaviours plays a vital role. We present a complete framework to detect and extract the moving objects to reconstruct a high quality static map. For a moving 3D camera setup, we propose a novel 3D Flow Field Analysis approach which accurately detects the moving objects using only 3D point cloud information. Further, we introduce a Sparse Flow Clustering approach to effectively and robustly group the motion flow vectors. Experiments show that the proposed Flow Field Analysis algorithm and Sparse Flow Clustering approach are highly effective for motion detection and segmentation, and yield high quality reconstructed static maps as well as rigidly moving objects of real-world scenarios.

1. Introduction

3D map reconstruction is one of the most active research topics in computer vision due to the numerous application requirements in robot localization [1, 2, 3], autonomous driving [4, 5, 6], and city map modelling [7, 8, 9, 10], etc. Benefiting from the emergence of affordable 2D and 3D cameras, such as Microsoft Kinect RGB-D camera and Velodyne 3D laser scanner, high quality 3D maps of both indoor and outdoor environments are obtained from nearly static environments [11, 10, 12]. However, high quality 3D map reconstruction remains a very challenging task for many practical scenarios such as streets or markets, mainly due to the numerous dynamic scene parts of the scene which yield significant "ghost" effects, see Fig. 1 Block 5. While detecting the dynamic scene parts in unknown environments, many practical difficulties, such as sudden illumination changes, night vision, and large field of view (FoV) requirement (*e.g.*360°) etc., lead current methods to fail [1, 13, 14, 15, 16, 17, 18, 19, 20]. These methods either rely on image information which is sensitive to illumination changes, or probabilistic models that require prior map knowledge, making them impractical for many real-world scenarios. Therefore, we propose a novel dynamic object detection method which only uses 3D point cloud information, making it robust to light changes, capable of night vision, and suitable for 360° FoV. Further, a complete framework for dynamic object detection, motion segmentation, and static map reconstruction is presented, see Fig. 1.

For a mobile camera system, both foreground and background observations are observed as moving objects due to the camera ego-motion. To (partly) compensate such phenomenon, registration techniques are applied so that the static parts of the scene coherently overlap while the motion trajectories of the moving objects are preserved, see Fig. 1 Block 1. Naturally, given accurate object-based point cloud segmentation with point correspondence knowledge, the moving objects are discriminated by their spatial displacements across frames. The precise establishment of point cloud correspondences and object segmentation are very challenging and are exhaustive methods [21, 22, 23]. In this work, we propose a method that establishes the feature correspondences using local flow consistency and performs the dynamic object segmentation on these (rather imprecise) correspondences. Our method is composed of 4 main steps described as follows.

Smooth Flow Vector (SFV) Estimation: The motion behaviour –either static or dynamic– of each point in a 3D scene is associated to a *Flow Vector* encoding its motion velocity and direction. The SFV is estimated by the subtraction of the corresponding points of consecutive frames. Such point correspondences can be quickly though roughly established by using nearest neighbour search, leading to noisy flow vector estimation. Therefore, under local motion consistency assumption, the smooth flow vector is estimated by the locally dominant flow vector within a small neighbourhood, which can be modelled and solved efficiently and optimally as an eigen-decomposition problem.

Motion Flow Identification: We identify the ones which correspond to the moving objects. The static objects coherently overlap while moving objects do not, which inspires the analysis of neighbour-points evolution along the flow vector. We propose a novel and efficient histogram analysis approach, see Fig. 1 Block 3.

*This research has been funded in part by ANR (Agence Nationale de la Recherche-France) International Project pLaTINUM (ANR-15-CE23-0010).

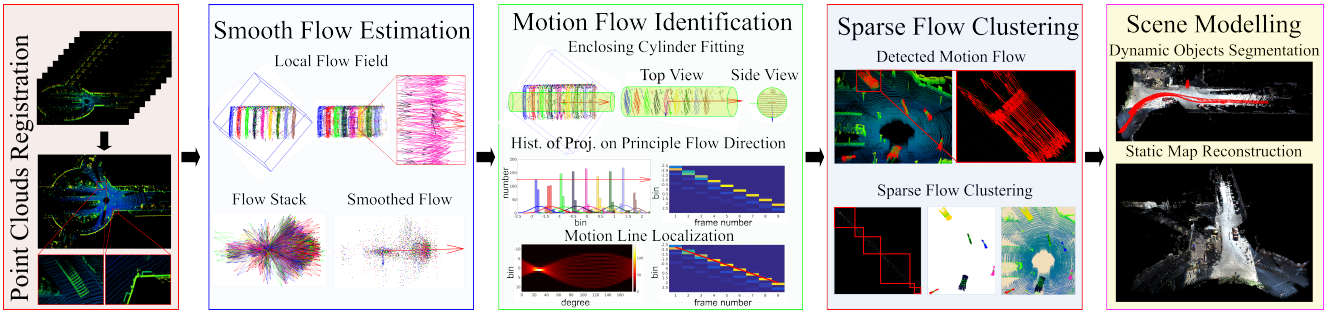

Figure 1. An overview of the proposed system to detect and segment the dynamic scene parts and to reconstruct the static map (Block 1-5 from left to right). Block 1: Given a short 3D point cloud sequence, an Iterative Closest Point algorithm is applied to register the point clouds to compensate the camera ego-motion. Block 2: For each point, we compute a Smooth Flow Vector (SFV) using the neighbourhood (local) flow field within a bounding box to represent its motion behaviour. The SFV is estimated by the Eigen decomposition of the centred neighbouring flows (the flow stack), as detailed in Sec. 4.1. Block 3: An enclosing cylinder (centred at the SFV) is determined to bound the inlier neighbourhood which are projected onto the SFV to build a set of 1D histograms. The shifting effect of the histograms is corresponding to the object motion. Such phenomenon can be studied via the properties of the motion line that located by the Radon transform on the 2D histogram, as detailed in Sec. 4.2. Block 4: The Sparse Flow Clustering algorithm regroups the detected motion flows into their independent motions, as detailed in Sec. 5. Block 5: Dynamic scene parts are identified and removed to build the static map.

Sparse Flow Clustering (SFC): To cluster the motion flows into their corresponding objects, we propose an algorithm which relies on the self-expressive property of motion flows' subspaces, as inspired by [18, 24, 25]. The SFC algorithm produces a sparse similarity graph which encodes the relations between the motion flows, from which we extract the corresponding motions using a spectral clustering.

Scene Modelling: We apply a 3D region growing approach [26] on the detected motion flows to densely segment the dynamic scene parts. The static scene reconstruction is achieved by registering only the static scene parts, while the rigidly moving objects, such as moving cars, are individually reconstructed from their dynamic scene parts.

Our major contributions are summarised as follows:
–We propose a novel algorithm using 3D Flow Field Analysis which outperforms the state-of-the-art methods for moving object detection.
–We introduce a new Sparse Flow Clustering model under the sparse subspace representation framework with spatial closeness constraints which also achieves significantly better performance than the literature approaches.
–We present an efficient framework for the detection and the segmentation of moving objects as well as the reconstruction of the static map of highly dynamic real-world scenes.

2. Related Work

Motion analysis of dynamic scenes is widely studied [18, 24, 25, 27, 28, 29, 30]. Among them, motion segmentation (MS)-based approaches are one of the most representative studies, such as the Generalized Principal Component Analysis [28], RANSAC-based MS [29], Agglomerative Subspace Clustering [30]. And more other methods are intensively studied in the reviews [27, 31]. Later, Elhamifar and Vidal [24] proposed a Sparse Subspace Clustering (2D-SSC) based on the self-representation property of motion subspace. Similarly, Hu *et al.* [25] proposed a SMooth Representation Clustering (2D-SMR) model, which achieves comparative results with a much higher computational efficiency, by enforcing the grouping effects of the motion subspaces of the image feature trajectories. Inspired by the 2D-SSC, Jiang *et al.* [18] proposed a 3D Sparse Subspace Clustering (3D-SSC) algorithm by analysing the feature trajectories directly in 3D Euclidean space. However, all the above algorithms rely on the consistency of the tracked feature trajectories and are therefore sensitive to tracking loss situations and partial occlusions.

Apart from the MS-based algorithms, Menze *et al.* [17] and Kochanov *et al.* [20] intended to detect and analyse the rigidly moving objects as Object Scene Flows (OSF) using stereo vision setup. However, because OSF relies on 2D image sequences, it is very sensitive to the environment changes as well as the sizes of the moving objects. Dewan *et al.* [32] proposed to detect and track the moving objects from a registered sequence using the SHOT [33] 3D feature descriptor. Unfortunately, such method remains very limited to object's motion speed and size and fails to detect such objects as fast moving cars or walking pedestrians.

In the context of Simultaneous Localization and Mapping with Moving Object Tracking (SLAM-MOT), [1, 13, 14, 15, 34, 35, 36] proposed to detect the moving objects by a probabilistic model which requires prior map information and relatively long term observations. Pomerleau *et al.* [11] and Ambrucs *et al.* [37] proposed to detect dynamic objects by comparing the current map with the known model. Yet, the initial clean reference model is required making these methods unsuitable for unknown environments. Zou

et al. [38] and Kundu *et al.* [39] proposed to use the epipolar constraints of images to discover moving objects, which relies on the feature matching and camera pose estimation accuracy, but is sensitive to illumination changes.

3. Background and Notations

Let $\mathsf{X} = \{\mathsf{x}_1, \cdots, \mathsf{x}_m\}$, where $\mathsf{x} \in \mathbb{R}^3$, be a 3D point set (cloud). And let $\mathsf{W} = \{\mathsf{w}_1, \cdots, \mathsf{w}_m\}$, where $\mathsf{w} \in \mathbb{R}^3$, 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 $\mathcal{X} = \{\mathsf{X}^t, t = 1, \ldots, n\}$ as the collection of multiple observed point sets that evolve over time t. Likewise, $\mathcal{W} = \{\mathsf{W}^t, t = 1, \ldots, n-1\}$ is the collection of flow vectors associated to $\mathcal{X}$.

For two point sets A and B, the operation $\mathsf{A} \ominus \mathsf{B}$ denotes the following element-wise subtraction:

$$\mathsf{A} \ominus \mathsf{B} = \{\mathsf{w} : \mathsf{w} = \mathsf{x} - \mathsf{y}, \mathsf{x} \in \mathsf{A}, \mathsf{y} = \mathscr{N}(\mathsf{x}, \mathsf{B})\}, \quad (1)$$

where x and y are the two corresponding points from A and B, respectively. w is the flow vector estimated by the subtraction of x and y. The nearest neighbourhood function $\mathscr{N}(\mathsf{x}, \mathsf{B})$ for efficient point correspondence establishment is defined as

$$\mathscr{N}(\mathsf{x}, \mathsf{B}) = \underset{\mathsf{y} \in \mathsf{B}}{\operatorname{argmin}} \|\mathsf{x} - \mathsf{y}\|_2. \quad (2)$$

Similarly, the nearest neighbourhood set of points within a radius r is given by

$$\mathscr{N}(\mathsf{x}, \mathsf{B}, r) = \{\mathsf{y} \in \mathsf{B} : \|\mathsf{x} - \mathsf{y}\|_2 \leq r\}. \quad (3)$$

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

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

Furthermore, let $\mathcal{C} \subset \mathsf{X}$ be the points within an enclosing cylinder centred at x_c, with axis w_c of radius r, which is denoted as:

$$\mathcal{C}(\mathsf{x}_\mathsf{c}, \mathsf{X}, \mathsf{w}_\mathsf{c}, r) = \{\mathsf{x} : \|\mathsf{x} - \mathsf{x}_\mathsf{c}\|^2 - \mathscr{P}(\mathsf{x}, \mathsf{w}_\mathsf{c})^2 \leq r^2, \mathsf{x} \in \mathsf{X}\}. \quad (5)$$

More notations: $\mathsf{A} = (a_{ij})$ is the element-wise representation of an $m \times n$ matrix. Its column-wise representation is $\mathsf{A} = [\mathsf{a}_1, \cdots \mathsf{a}_j, \ldots \mathsf{a}_n]$ where a_j is an m-dimensional vector. $\mathsf{A} \succeq 0$ means that A is a symmetric and positive semi-definite matrix.

4. Flow Field Analysis

Recall that, our first objective is to detect the moving objects inside a 3D point cloud sequence. In essence, the object motion is defined by its temporal displacement which can be described by a set of motion flows. To address, we propose the *3D Flow Field Analysis* model under the local motion consistency assumptions similar to the optical flow estimation [40] and the 3D scene flow estimation [41] where two assumptions are made: (i) the flows' motion behaviours are similar within a small neighbourhood and (ii) the local geometric structure does not change rapidly.

4.1. Smooth Flow Vector Estimation

Given n point sets $\mathcal{X} = \{\mathsf{X}^t, t = 1, \ldots, n\}$, for $t = 1, \ldots, n-1$, we compute the point-wise flow based on the evolution of points over time, as follows:

$$\mathsf{W}^t = \mathsf{X}^t \ominus \mathsf{X}^{t+1}. \quad (6)$$

In other words, we consider the difference between consecutive motions. Taking the locally homogeneous assumption of neighbouring flow vectors, we perform the smoothing of vector field by updating each $\mathsf{w}_i \in \mathsf{W}^t$ to,

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

where w_i^* is the returned desired smoothed flow vector of v. $\mathcal{N} = \mathscr{N}(\mathsf{x}_i, \mathsf{X}^t, r)$ is the small neighbourhood (within the radius r) that defined the local flow field $\Omega(\mathcal{N})$. In fact, the problem of Eq. (7) can be solved efficiently as an eigen-decomposition problem. Its solution can be obtained by computing the eigenvector of a matrix $\mathbf{W}$ whose rows are $\mathsf{w}^\intercal$ for all $\mathsf{w} \in \Omega(\mathcal{N})$. Note that, all the $\mathsf{w} \in \Omega(\mathcal{N})$ are normalized to unit vectors to obtain the optimal solution.

4.2. Static Point and Motion Flow Discrimination

Consider that the structure of the local point sets is preserved. Thus, $\mathcal{C}^t = \mathcal{C}(\mathsf{x}, \mathsf{X}^t, \mathsf{w}, r), t = 1, \ldots, n$ (the measurements of a local point set moving along w from Eq. 7) are homomorphic. Therefore, the projections $\mathscr{P}^t = \mathscr{P}(\mathcal{C}^t, \mathsf{w})$ remain unchanged for all $t = 1, \ldots, n$. Let $\mathcal{H}^t$ be a k-bin histogram of projections $\mathscr{P}^t$ at time t. The motion state of the point sets can be described by the following equation:

$$\mathcal{H}^{t+1}(b) = \mathcal{H}^t(b + \alpha(t)), \quad (8)$$

where b is a bin of the histogram, and $\alpha(t) = \beta t$ (with constant value β) is the displacement of the histogram (or projections) from t to $t+1$. Eq. (8) implies that the histogram is replicated from $t = 1, \ldots, n$ due to the temporal local structure and speed consistency.

Given histograms $\mathcal{H}^t(b), t = 1 \ldots, n$, our task is to estimate β and b that satisfy Eq. (8) for all t, which can be modelled as a minimization problem as:

$$\underset{\beta, b}{\operatorname{argmin}} \sum_{t=1}^{n-1} ||\mathcal{H}^{t+1}(b) - \mathcal{H}^t(b + \alpha(t))||_2. \quad (9)$$

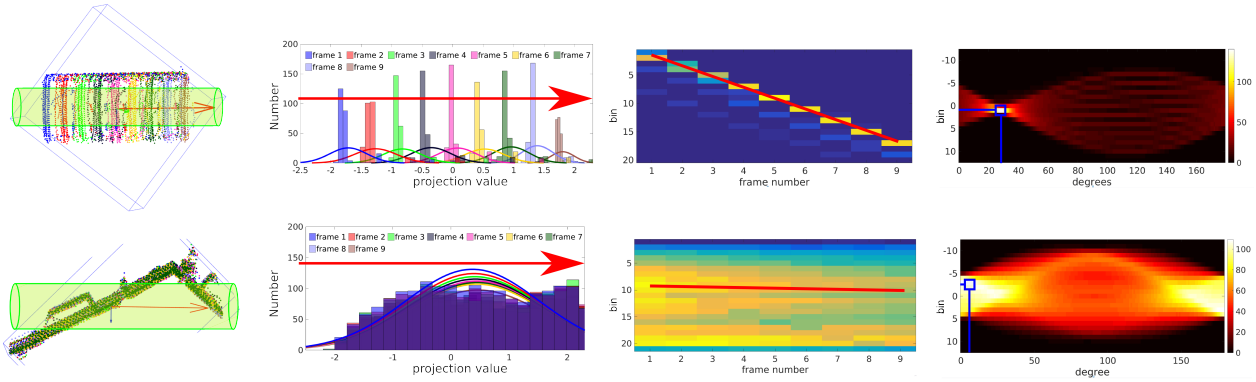

Figure 2. Motion and static flow analysis: Col. 1 shows the enclosing cylinder preserving the local structure. Col. 2 shows the histograms of cylinder-point projections of each frame. Remarkably, the histograms of motion flow (upper) are shifted along the flow direction, while the histograms of static 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).

To efficiently solve problem (9), the n 1D histograms $\mathcal{H}^t$ are concatenated into a 2D histogram $\mathsf{M} = [\mathcal{H}^1, \cdots, \mathcal{H}^n]$ of size $k \times n$, as illustrated in Fig. 2 middle. Let a line L in the 2D histogram be defined by $L(t) = \beta t + b$, for slope β and offset b. The optimal parameters β^* and b^* are obtained by

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

Problem (10) can be solved efficiently by applying Radon transform [42] on M, as illustrated in Fig. 2. Three measurements are made along the line L^* to categorize the point sets into static or dynamic. Since the slope β^* represents the magnitude of speed, β^* of a static point set is very small. Further, if $s_t = \mathcal{H}^t(L^*), t = 1, \ldots, n$ are 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.

4.3. Dynamic Neighbourhood Search

Practical scenarios, in which the sizes and the speeds of objects may significantly vary (from pedestrians to trucks), impose to analyse the scene in a dynamic manner. Our analysis algorithm is mostly driven by 3 parameters that are the size of bounding box (for fast neighbourhood search), its location, and the radius of the enclosing cylinder, which can be reduced to 2 parameters by considering a fixed size bounding box and the radius as a ratio of its size. We consider motions as being "slow" when the analysed point sets translated by the estimated motion remain within the 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 to follow the analysed object, and is updated as soon as consecutive frames have led to a coherent motion, as illustrated in Fig. 3. Our experiments show that it is sufficient to choose a radius smaller than 20% of the size of the bounding box then to dynamically adapt this radius proportionally to the object to camera distance.

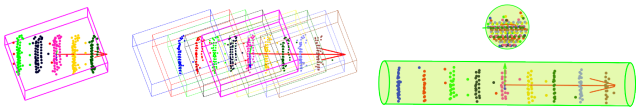

Figure 3. Dynamic local neighbourhood search of a fast moving object: left shows that the bounding box covers only 5 frames. Middle shows the translation of bounding box along the flow direction. Right shows the enclosing cylinder with full frames.

Precisely, 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. Initially, the bounding box (centred at x_o) covers $f < n$ frames, due to the high speed. 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 median values of projections of $\mathscr{P}^t$. The bounding box is translated to $\mathsf{x}_t = \mathsf{x}_o + \delta_t \mathsf{w}$, until all n frames are covered.

4.4. Implementation Details

Given n consecutive frames of point sets, an ICP-based registration algorithm [10] is applied to compensate the camera ego-motion. Starting form the registered point sets as input, Algo. 1 is applied to discriminate the static and dynamic points, and to estimate the motion flows of the dynamic points. For the sake of computational efficiency, 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 a relatively easy task. In step 4, the enclosing cylinder radius is defined as $r = 0.4(1 + d/D)$, where is d is the object to camera distance and D is the camera's maximum data acquisition distance (*e.g.* $D = 100$ for Velodyne 3D laser scanner). In step 7, τ_S is defined as 40% of the total number of neighbour points within the enclosing cylinder (sum 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 computes the volume integrations in different angles at different positions. Thus, its maximum response directly gives the desired solution of problem (10). In Fig. 2 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 lead to the different behaviours (refer to the above discussions in Section 4.2) of the motion line L^* of static and dynamic points.

Algorithm 1: Motion Flow Identification.

Data: Point sets $\mathcal{X} = \{\mathsf{X}^1, \cdots, \mathsf{X}^n\}$, centre point set $\mathsf{X}^{\bar{t}=\frac{n}{2}}$.
1 **Setting**: $n = 9$, $k = 20$, bounding box size $4 \times 4 \times 4$, $\tau_\beta = 0.175$, $\tau_S = 0.4\sum \mathsf{M}$, $\tau_E = 1.8$.
2 **for** $\mathsf{x}_i \in \mathsf{X}_{\bar{t}}$ **do**
3 Place a 3D bounding box at x_i for local flow field estimation ($\mathbf{W}$) using Eq. (6), and perform eigen-decomposition: $[\mathsf{V}, \mathsf{D}] = \text{eig}(\mathbf{W})$ to obtain the dominant flow $\mathsf{v} = \mathsf{V}(:, 3)$.
4 Fit an enclosing cylinder $\mathcal{C}(\mathsf{x}_i, \mathcal{X}, \mathsf{v}, r)$.
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.

5. Sparse Flow Clustering

We cluster the dynamic point set, obtained from the flow field analysis (discussed in Section 4), into similar subsets for objects' motion behaviour analysis. Our clustering process uses both the spatial and the motion vector information. On the one hand, we make the assumption that the vectors from one cluster are self-expressive. Thus, a flow vector can be approximated by the sparse 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 a predefined radius.

Let $\mathsf{X} = [\mathsf{x}_1, \ldots \mathsf{x}_j, \ldots \mathsf{x}_n]$ and $\mathsf{W} = [\mathsf{w}_1, \ldots \mathsf{w}_j, \ldots \mathsf{w}_n]$ be the $3 \times n$ matrices of the point set and the corresponding flow vectors of moving objects, the self-expressive sparse representation (as in [24]) can be written as

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

for a 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

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

This is a convex problem, whose optimal solution can be found using the second order cone programming [43]. In fact, its equivalent problem is the semi-definite programming given by

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

5.1. 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

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

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

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

where $\mathsf{E}_\mathsf{w} = \lambda_1 \sum_{j=1}^{n} \epsilon_\mathsf{w}$ and $\mathsf{E}_\mathsf{x} = \lambda_2 \sum_{j=1}^{n} \epsilon_\mathsf{x}$. X_t and W_t are the 3D points and their flow vectors at frame t, respectively. 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.Weight parameters λ_1 and λ_2 are simply set to 1.

5.2. Implementation Details

The Sparse Flow Clustering (SFC) algorithm consists of three major steps (see Algo. 2) which are implemented using CVX [44] optimization toolbox. In the sparse optimization step, a binary $n \times n$ connectivity graph $\mathsf{D} = [d_1, \cdots d_j, \cdots, \mathsf{d}_n]$ is used to enforce the spatial closeness constraint on the selected sparse representation elements, such that Eq. (17) becomes:

$$\mathsf{W} = \mathsf{W}(\mathsf{C} \cdot \mathsf{D}) + \mathsf{E}, \quad \forall d_{ij} > \tau_d, d_{ij} = 0, \text{ else } d_{ij} = 1, \quad (19)$$

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 computational efficiency.

In step 2 of Algo. 2, a sparse symmetric similarity graph $\mathsf{G} = |\mathsf{C}| + |\mathsf{C}|^\mathsf{T}$ is constructed. Since G encodes the connectivity information among the flows, a K-mean spectral clustering is employed to group the flow clusters.

Note that the proposed SFC does NOT rely on feature tracking and feature trajectory (unlike [18, 24, 25]), making it more practical 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.

Algorithm 2: Sparse Flow Clustering.

Data: 3D point sets $[\mathsf{X}_1, \cdots, \mathsf{X}_t]$ and flows $[\mathsf{W}_1, \cdots, \mathsf{W}_t]$.
Result: k clustered subspaces.
1 Sparse flow representation using Eq. (18).
2 Sparse similarity graph: $\mathsf{G} = |\mathsf{C}| + |\mathsf{C}|^\mathsf{T}$.
3 K-mean spectral clustering on G.

6. Experiments

We conduct extensive evaluations on the challenging real-world KITTI benchmark [5] that contains highly dynamic environment scenarios. Note that the proposed method only utilizes locally registered Velodyne 3D point clouds (*i.e.* using ICP-based algorithm [10]), and that GPS and IMU information are not used. Seven representative datasets were selected to cover different practical scenarios as listed: a large number of moving objects (pedestrians and cars in Market), fast motions (van in Junction), slow motions (pedestrians in Campus or Market), large objects (train in Station) and small objects (pedestrians), severe occlusions (van in Junction), static camera (Red Light, Campus, Pedestrian), and moving camera platforms (remaining others). The selected sequences have rather demonstrated the effectiveness and generality of the proposed methods. The detailed results are synthesised in Table 1 and Table 3. The performances with the state-of-the-art methods are assessed using the *Sensitivity* and *Specificity* metrics. For comparison with MS-based methods, the misclassification rate metric suggested by [24, 25] is adopted. All the experiments have been conducted on a machine with Intel Quad Core i7-2.7GHz, 32GB Memory using MATLAB.

6.1. Motion Detection Evaluation

Our Moving Object Detection algorithm (3D-MOD) is compared against four representative algorithms. We recall that the 2D-SMR, 2D-SSC and 3D-SSC are trajectory-based motion segmentation algorithms which group the feature trajectories into their corresponding motions. For the evaluation of moving object detection, we define: *True Positive* – as long as a motion trajectory is NOT classified as background motion, and *True Negative* – if a background trajectory is classified as background motion. When several motions are involved, a feature trajectory might not be correctly classified into its corresponding motion, and will be considered as a true positive.

Table 1 summarizes the performances of 2D-SMR-J1 [25], 2D-SMR-J2 [25], 2D-SSC [24], 3D-SSC [18] and 3D-MOD using sensitivity and specificity metrics. The main characteristics of the results are summed up as follows: a) The 3D-SSC has very similar performance to its 2D counterpart in terms of sensitivity, but a much higher specificity at the cost of lower computational efficiency. b) 3D-MOD achieves the best sensitivity and specificity in most cases. In average, the 3D-MOD shows a sensitivity that is slightly better than the other methods but with a significantly higher specificity. c) The 3D-based methods (3D-SSC and 3D-MOD) exhibit very stable performances and a much higher specificity, thanks to their robustness to perspective projection effects. d) Regarding the computational efficiency, our 3D-MOD approach can be seen as an intermediate method, although it can be easily parallelized if online motion detection application is required.

Table 2 adopts the mean and median Misdetection Error metrics ($\eta = \frac{\#\text{ False Positive} + \#\text{ False Negative}}{\#\text{ Features}}$) similar to [24, 25] for evaluation, with corresponding Whisker's boxplot [45] statistical comparisons in Fig. 4. Similar remarks from Table 1 can be observed: the 3D-SSC and 3D-MOD has significantly better performances than other methods due to their persistent high specificity. Fig. 4 also shows that the 3D-MOD outperforms the other methods with lower

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 1. Performance quantification on KITTI benchmark: Col. 1-3 are the sequence name, length and average moving object number, respectively. The rest columns show the Sensitivity, Specificity and Processing time (in second). Last row averages the overall performances.

Sequence	2D-SSC		3D-SSC		2D-SMR-J1		2D-SMR-J2		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 2. Quantitative evaluation on KITTI dataset: using Mean and Median values of Misdetection rate metric.

median misdetection rate as well as much higher robustness.

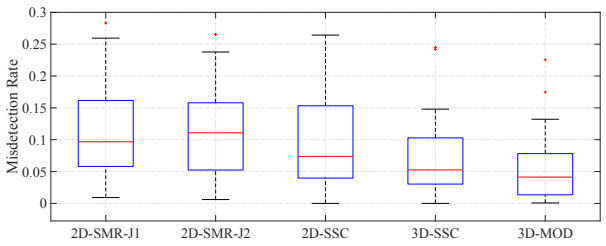

Figure 4. Performances analysis on KITTI dataset.

Table 3 concludes the comparisons between the 3D-MOD against the Object Scene Flow (OSF) [17] algorithm. Since the OSF method produces dense moving object detection and segmentation, for the purpose of fair comparison, a 3D Region Growing [26] is applied to the detected motion flows to densely segment the moving objects. Remarkably, the 3D-MOD is faster and consistently exhibits a much higher sensitivity with a slightly lower specificity.

The main reasons that 3D-MOD surpasses the state-of-the-art methods can be summarized as:

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

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

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

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 3. OSF and 3D-MOD quantitative evaluation: Col. 2-5 indicate the minimum and maximum object size (in pixel) and speed (m/s) of moving objects, respectively. Both sensitivity and specificity are computed using dense segmentation of 3D point cloud.

problem in essence.

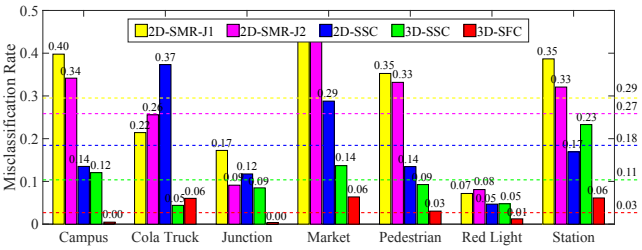

Figure 5. Motion segmentation quantification: dashed lines highlight the averaged misclassification rates.

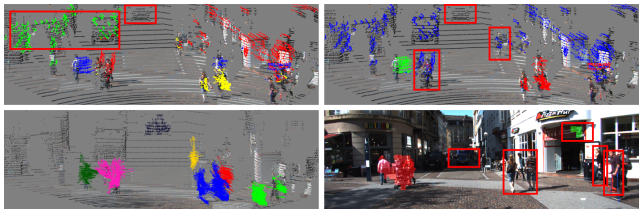

Figure 6. Motion segmentation results on Market sequence: using 2D-SMR (top-left), 3D-SSC (top-right), 3D-SFC (bottom-left), OSF(bottom-right). Red boxes highlight the wrong segmentations.

6.2. Motion Segmentation Evaluation

For motion segmentation quantification, we use the *Misclassification Rate* (same as [24, 25]) to compare the algo-

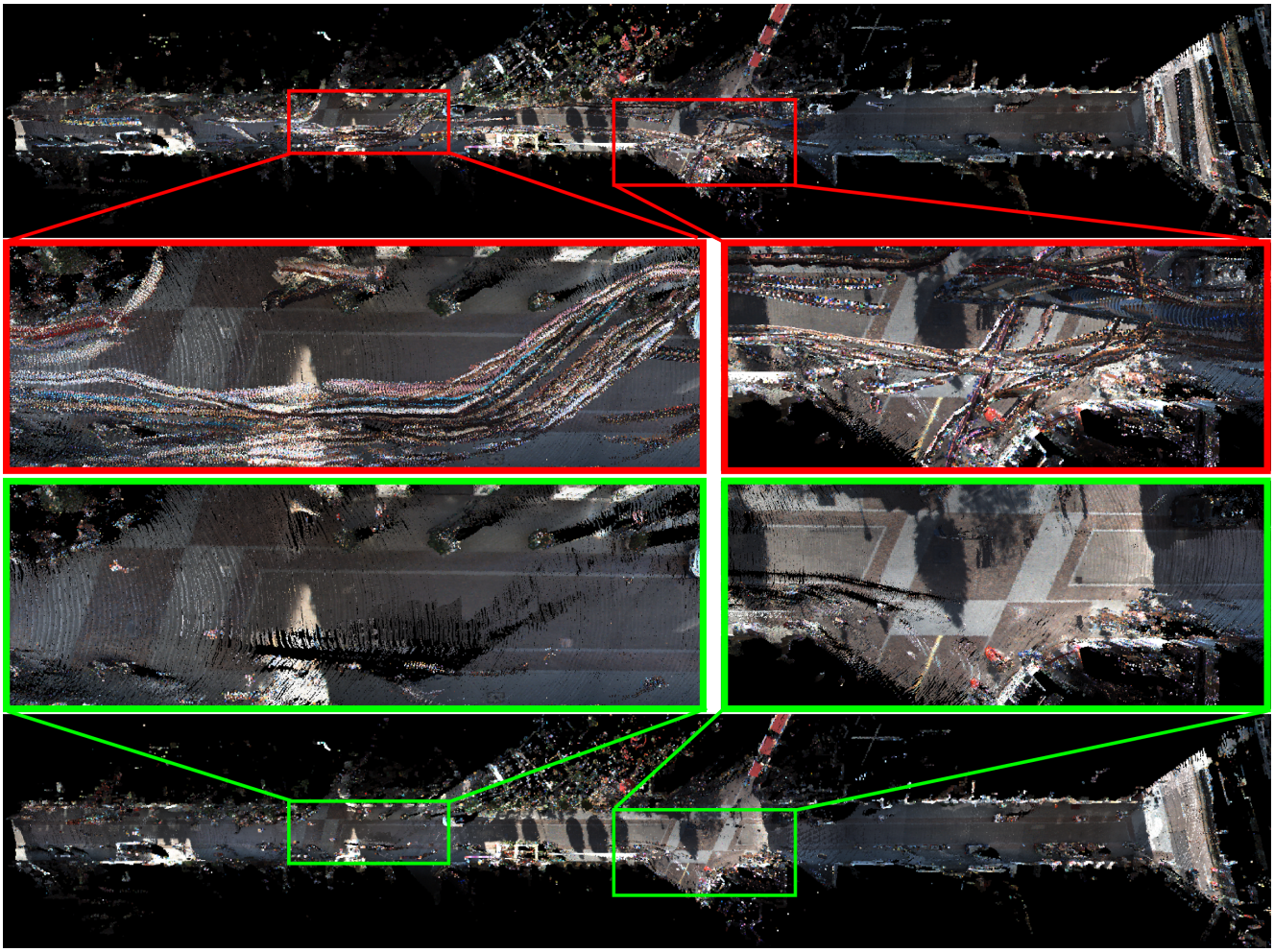

Figure 7. Top image is the scene 3D reconstruction using [10] of Market sequence where numerous moving objects exist. The zoom-in regions show the immense "ghost" effects from the moving objects. Bottom image is our static map which has significantly higher quality.

rithms' performances, see Fig. 5. Our 3D-SFC achieves the best results for the evaluated datasets. Fig. 6 shows the outstanding performance of the proposed 3D-SFC algorithm on MS. Note that, prior to the flow clustering, the detected static flows are removed (Fig. 6 bottom-left), which largely simplify the motion flow clustering problem. Moreover, 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.

6.3. Static Map and Rigid Object Reconstruction

We obtained very satisfactory static maps as well as reconstructed rigidly moving objects. Fig. 7 shows the challenging Market sequence (with 1200 frames) which contains a large amount of moving objects. The static map produced by our framework is of highly better quality because our framework is robust to light changes, occlusions, slow or very fast motions, etc. Getting the clustered motion trajectories, the 3D reconstruction of moving objects can be obtained by registering the sparse point clouds from different view ports during their motions, see our previous works [46, 47].

7. Conclusions and Future Perspectives

We have proposed an original 3D Moving Object Detection algorithm based on Flow Field Analysis under the local motion consistency assumptions. We have presented a novel 3D Sparse Flow Clustering approach relying on the self-representation property of flow subspaces and spatial closeness constraints. By integrating the proposed 3D-MOD and 3D-SFC algorithms, the proposed framework is not only robust, efficient and accurate, but also results in very high quality static map and dynamic object reconstructions using a 2D-3D moving camera setup. Our algorithms serve many applications such as accurate robot localization and autonomous driving in crowded environments. We also leave high-level tasks, such as semantic scene understanding and objects' behaviours analysis, as future perspectives.

References

[1] Chieh-Chih Wang, Charles Thorpe, Sebastian Thrun, Martial Hebert, and Hugh Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *Intern. Journal of Robotics Research (IJRR)*, 26(9):889–916, 2007. 1, 2

[2] Jose A Castellanos and Juan D Tardos. *Mobile robot localization and map building: A multisensor fusion approach*. Springer Science & Business Media, 2012. 1

[3] Philipp Koch, Stefan May, Michael Schmidpeter, Markus Kühn, Christian Pfitzner, Christian Merkl, Rainer Koch, Martin Fees, Jon Martin, Daniel Ammon, et al. Multi-robot localization and mapping based on signed distance functions. *Journal of Intelligent & Robotic Systems*, 83(3-4):409–428, 2016. 1

[4] Jaewoong Choi, Junyoung Lee, Dongwook Kim, Giacomo Soprani, Pietro Cerri, Alberto Broggi, and Kyongsu Yi. Environment-detection-and-mapping algorithm for autonomous driving in rural or off-road environment. *IEEE Transactions on Intell. Transportation Systems*, 13(2):974–982, 2012. 1

[5] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *Intern. Journal of Robotics Research (IJRR)*, 2013. 1, 6

[6] Christian Berger and Bernhard Rumpe. Autonomous driving-5 years after the urban challenge: The anticipatory vehicle as a cyber-physical system. *arXiv preprint arXiv:1409.0413*, 2014. 1

[7] Lingyun Liu and Ioannis Stamos. Automatic 3d to 2d registration for the photorealistic rendering of urban scenes. In *Proc. of Comp. Vis. and Pattern Rec. (CVPR)*, 2005. 1

[8] Danda Pani Paudel, Cédric Demonceaux, Adlane Habed, Pascal Vasseur, and In So Kweon. 2d-3d camera fusion for visual odometry in outdoor environments. In *Proc. of Intell. Ro. and Sys. (IROS)*, 2014. 1

[9] Vladyslav Usenko, Jakob Engel, Jörg Stückler, and Daniel Cremers. Reconstructing street-scenes in real-time from a driving car. In *Proc. of 3D Vision (3DV)*, 2015. 1

[10] Ji Zhang and Sanjiv Singh. Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, pages 1–16, 2016. 1, 4, 6, 8

[11] Rareş Ambruş, Nils Bore, John Folkesson, and Patric Jensfelt. Meta-rooms: Building and maintaining long term spatial models in a dynamic world. In *Proc. of Intell. Ro. and Sys. (IROS)*, 2014. 1, 2

[12] Lingni Ma, Christian Kerl, Jörg Stückler, and Daniel Cremers. Cpa-slam: Consistent plane-model alignment for direct rgb-d slam. In *Proc. of Int. Conf. on Rob. and Auto. (ICRA)*, 2016. 1

[13] Shu-Yun Chung and Han-Pang Huang. Slammot-sp: simultaneous slammot and scene prediction. *Advanced Robotics*, 24(7):979–1002, 2010. 1, 2

[14] Paschalis Panteleris and Antonis A Argyros. Vision-based slam and moving objects tracking for the perceptual support of a smart walker platform. In *Proc. of Euro. Conf. on Com. Vis. (ECCV)*, 2014. 1, 2

[15] Ren C Luo and Chun Chi Lai. Multisensor fusion-based concurrent environment mapping and moving object detection for intelligent service robotics. *IEEE Trans. on Industrial Electronics*, 61(8):4043–4051, 2014. 1, 2

[16] Jakob Engel, Jörg Stückler, and Daniel Cremers. Large-scale direct slam with stereo cameras. In *Proc. of Intell. Ro. and Sys. (IROS)*, 2015. 1

[17] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proc. of Comp. Vis. and Pattern Rec. (CVPR)*, 2015. 1, 2, 7

[18] 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, 2016. 1, 2, 6

[19] C. Jiang, D. P. Paudel, Y. Fougerolle, D. Fofi, and C. Demonceaux. Reconstruction 3d de scènes dynamiques par segmentation au sens du mouvemen. In *In Le 20èmé Congrès National sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA)*. Clermont-Ferrand, France, Jun. 2016. 1

[20] Deyvid Kochanov, Aljosa Osep, Jörg Stückler, and Bastian Leibe. Scene flow propagation for semantic mapping and object discovery in dynamic street scenes. In *Proc. of Intell. Ro. and Sys. (IROS)*, 2016. 1, 2

[21] Shin Miyake, Yuichiro Toda, Naoyuki Kubota, Naoyuki Takesue, and Kazuyoshi Wada. Intensity histogram based segmentation of 3d point cloud using growing neural gas. In *Proc. of Intell. Robotics and Applications*, 2016. 1

[22] George Sithole and WT Mapurisa. 3d object segmentation of point clouds using profiling techniques. *South African Journal of Geomatics*, 1(1):60–76. 1

[23] M Lindstrom and J-O Eklundh. Detecting and tracking moving objects from a mobile platform using a laser range scanner. In *Proc. of Intell. Ro. and Sys. (IROS)*, 2001. 1

[24] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2013. 2, 5, 6, 7

[25] Han Hu, Zhouchen Lin, Jianjiang Feng, and Jie Zhou. Smooth representation clustering. In *Proc. of Comp. Vis. and Pattern Rec. (CVPR)*, 2014. 2, 6, 7

[26] Georg Mühlenbruch, Marco Das, Christian Hohl, Joachim E Wildberger, Daniel Rinck, Thomas G Flohr, Ralf Koos, Christian Knackstedt, Rolf W Günther, and Andreas 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. 2, 7

[27] Roberto Tron and René Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *Proc. of Comp. Vis. and Pattern Rec. (CVPR)*, 2007. 2

[28] Rene Vidal, Yi Ma, and Shankar Sastry. Generalized principal component analysis. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2005. 2

[29] Jingyu Yan and Marc Pollefeys. Articulated motion segmentation using ransac with priors. In *Dynamical Vision*, pages 75–85. Springer, 2007. 2

[30] Shankar Rao, Roberto Tron, Rene Vidal, and Yi Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2010. 2

[31] Luca Zappella, Xavier Lladó, and Joaquim Salvi. Motion segmentation: A review. In *Proc. of Artificial Intell. Research and Development*, 2008. 2

[32] Ayush Dewan, Tim Caselitz, Gian Diego Tipaldi, and Wolfram Burgard. Motion-based detection and tracking in 3d lidar scans. In *Proc. of Int. Conf. on Rob. and Auto. (ICRA)*, 2016. 2

[33] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *Proc. of Euro. Conf. on Com. Vis. (ECCV)*, 2010. 2

[34] Kuen-Han Lin and Chieh-Chih Wang. Stereo-based simultaneous localization, mapping and moving object tracking. In *Proc. of Intell. Ro. and Sys. (IROS)*, 2010. 2

[35] Asma Azim and Olivier Aycard. Detection, classification and tracking of moving objects in a 3d environment. In *Proc. of Intell. Vehicles Symposium (IV)*, 2012. 2

[36] Guillaume Trehard, Evangeline Pollard, Benazouz Bradai, and Fawzi Nashashibi. Credibilist slam performances with different laser set-ups. In *Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on*, pages 589–594. IEEE, 2014. 2

[37] François Pomerleau, Philipp Krüsi, Francis Colas, Paul Furgale, and Roland Siegwart. Long-term 3d map maintenance in dynamic environments. In *Proc. of Int. Conf. on Rob. and Auto. (ICRA)*, 2014. 2

[38] Danping Zou and Ping Tan. Coslam: Collaborative visual slam in dynamic environments. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2013. 3

[39] Abhijit Kundu, CV Jawahar, and K Madhava Krishna. Realtime moving object detection from a freely moving monocular camera. In *Proc. of Robotics and Biomimetics (ROBIO)*, 2010. 3

[40] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 3

[41] Sundar Vedula, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2005. 3

[42] Stanley R. Deans. *The Radon transform and some of its applications*. Courier Corporation, 2007. 4

[43] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. 5

[44] Michael Grant and Stephen Boyd. Cvx: Matlab software for disciplined convex programming. 6

[45] Yoav Benjamini. Opening the box of a boxplot. *The American Statistician*, 1988. 6

[46] C. Jiang, Y. Fougerolle, D. Fofi, and C. Demonceaux. Dynamic 3d scene reconstruction and enhancement. In *International Conference Image Analysis and Processing*. Lecture Notes in Computer Science, Springer, Sept. 2017. 8

[47] C. Jiang, D. Christie, D.P. Paudel, and C. Demonceaux. High quality recontruction of dynamic objects using 2d-3d camera fusion. In *In Proc. of International Conference on Image Processing (ICIP)*. IEEE, Sept. 2017. 8