Static-map and Dynamic Object Reconstruction in Outdoor Scenes using 3D Motion Segmentation

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

Static-map and Dynamic Object Reconstruction in Outdoor Scenes using 3D Motion Segmentation

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

***Abstract*—This paper aims to build the static-map of a dynamic scene using a mobile robot equipped with 3D sensors. The sought static-map consists of only the static scene parts, which has a vital role in scene understanding and landmark based navigation. Building static-map requires the categorization of moving and static objects. In this work, we propose a Sparse Subspace Clustering-based Motion Segmentation method that categories the static scene parts and the multiple moving objects using their 3D motion trajectories. Our motion segmentation method uses the raw trajectory data, allowing the objects to move in direct 3D space, without any projection model assumption or whatsoever. We also propose a complete pipeline for static-map building which estimates the inter-frame motion parameters by exploiting the minimal 3-point Random Sample Consensus algorithm on the feature correspondences only from the static scene parts. The proposed method has been especially designed and tested for large scene in real outdoor environments. On one hand, our 3D Motion Segmentation approach outperforms its 2D based counterparts, for extensive experiments on KITTI dataset. On the other hand, separately reconstructed static-maps and moving objects for various dynamic scenes are very satisfactory.**

***Index Terms*—Mapping; Motion and Path Planning; SLAM**

I. INTRODUCTION

IN recent years, visual Simultaneous Localization and Mapping (vSLAM) based autonomous robot navigation techniques have achieved great success in static environments. Yet, in dynamic scenes, the navigation remains very challenging, mainly because the moving objects contribute to a poor localization accuracy and map artifacts. Under such circumstances, the localization is usually performed by estimating the camera motion based on either the features' motion consensus [1] or the weighted cost minimization [2]. Dynamic scene parts in both cases are treated as alien objects or outliers, and thus discarded. However, when a significant number of features belong to the dynamic scene parts, it can not only become difficult to discard them, but also degrade the localization accuracy [3]. Therefore, robot navigation in dynamic environments requires the detection and removal of moving objects, prior to the static-map building. A static-map of the dynamic scene consists of only the static scene parts, which in itself, is of primary interest for scene modelling. Furthermore, it is also an important step towards scene understanding and landmark based navigation. To do so, we propose a complete pipeline for static-map building, see Fig. 1, which involves three main stages: a) 3D feature trajectories construction; b) Motion Segmentation (MS); c) 3D scene registration.

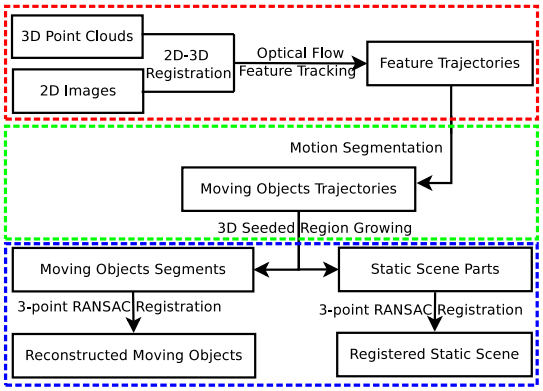


Fig. 1: Static-map and dynamic object building framework.

For mobile robots capturing dynamic scenes, both static and dynamic scene parts appear to be moving. Therefore, a straightforward approach to distinguish the dynamic and static parts would be to analyze their motion trajectories. In this regard, the scene parts that reciprocate the robot motion are considered to be static, whereas the remaining ones belong to the moving objects or outliers. Note that a common practice for the detection of the object motion is to segment their features' trajectories.

When the robot is equipped with 3D sensors, it is obvious to represent and segment the features' trajectories directly in 3D space. In practice, such feature trajectories can be obtained by detecting and tracking 3D feature points. If both 2D cameras and 3D sensors are available, the 3D feature tracking can be supported by their 2D feature descriptors, after projecting onto the image. In this work, a 2D optical-flow based method has been adopted to acquire the 3D feature trajectories. However, in many practical scenarios, the trajectories obtained in this manner yield numerical instabilities due to their non-uniform distribution on static and dynamic objects. We tackle this problem by employing a flow-likelihood-based feature sampling technique so that the feature distribution of moving and static objects is balanced, making it more switchable for wide range of dynamic objects coverage. The flow-likelihood-based sampling technique samples the features based on their median-flow-suppressed optical flow speed, under the assumption that median optical flow belongs to the scene background. A higher speed implies that the feature is

Manuscript received: August, 31, 2015; Revised November, 20, 2015; Accepted December, 16, 2015.

This paper was recommended for publication by Editor Cyrill Stachniss upon evaluation of the Associate Editor and Reviewers' comments.

The authors are with LE2I UMR 6306, CNRS, Arts et Métiers, Univ. Bourgogne Franche-Comté, F-21000 Dijon, France (e-mail: firstname.lastname@u-bourgogne.fr

Digital Object Identifier (DOI): see top of this page.

different from the background flow, hence it is more likely to belong to a moving object.

Using the 3D trajectories of sparse feature points, we propose a Sparse Subspace Clustering based 3D (3D-SSC) MS algorithm that categorizes multiple moving objects as well as the static scene parts. Although many MS methods are available in the context of video surveillance, object tracking and action recognition [4], they provide some solutions for objects moving either in 2D space, or in 3D space under the camera projection model assumption. The proposed method performs the motion segmentation in 3D space using the raw motion trajectories, without any projection model assumption or whatsoever. The 3D-SSC algorithm intends to find the minimal linear sparse subspaces that best represent the given motion trajectories. In this work, we show through several experiments that our 3D-SSC approach outperforms its 2D based counterparts.

While building the static-map, the dynamic objects must be segmented and removed from the scene, so that only the static parts remain in the resulting map. Thanks to MS, the sparse set of feature points can be divided into multiple subsets – each subset being assigned to an object with unique motion trajectory. Later, a dense segmentation of the 3D scene points (into static and dynamic) is obtained using these feature subsets, with the help of a Region Growing technique. We have developed two applications, namely, static-map building and moving object reconstruction upon the algorithm to detect the static parts and the moving objects. The static-map is built by registering multi-frame point clouds using minimal 3-point Random Sample Consensus (RANSAC) algorithm on the feature correspondences only from the static scene parts. The minimal 3-point RANSAC uses Cayley representation of the rotation matrix, which allows to obtain rigid transformation between two point clouds using linear solvers, similarly to [15]. The proposed static-map building algorithm performs very satisfactorily on realistic outdoor environments. The moving objects are densely reconstructed by registering their observations from different view-ports.

The main contributions of this paper are two-folded:

- A novel framework for 3D motion segmentation has been proposed. The proposed method groups the 3D feature trajectories using the Sparse Subspace Clustering algorithm which outperforms its 2D-based counterparts.
- A complete pipeline to build the static-map, by taking the advantage of the proposed MS method, is presented. Our system not only provides the static-map of a real outdoor dynamic scene, but also contributes to better 3D reconstruction of the moving objects.

II. Related Work

For decades, numerous works have been conducted in image-based motion segmentation [4]. Among the most representative approaches, Generalized Principle Component Analysis (GPCA) [7], RANSAC-based MS [10], Agglomerative Subspace Clustering (ASC) [6], and Sparse Subspace Clustering (SSC) [5] have been intensively studied in [4]. Usually, the problem of MS is addressed by separating the motions into subspaces such that every motion trajectory belongs to its corresponding subspace. In this regard, GPCA estimates the global linear subspaces for motion clustering, while the LSA does the same locally. Although these methods provide great insight for subspace-based motions clustering, their practical usage is limited either because of their high sensitivity towards noise/outliers or sharp increase in computational complexity with the increasing number of moving objects. ASC is a more robust method that combines the techniques of lossy compression, rank minimization, and sparse representation. Inspired by ASC, Elhamifar and Vidal [5] proposed an SSC algorithm that relies on the idea of self-expressive sparse representation. In fact, SSC is considered to be the leading MS method in the literature [8].

Apart from 2D-based MS, 2D-3D or 3D-based MS methods have also been developed. A recent work of Stuckler et al. [11] performs dense 3D motion segmentation on RGB-D data using an Expectation Maximization framework. Similarly, Papon et al. [25] and Koo et al. [26] track and segment the moving objects in a RGB-D sequences. These methods, however, are designed under the fixed camera assumptions and mostly tested in controlled environments. Perera et al. [14] use Truncated Signed Distance Function to segment the moving objects using volumetric surfaces representation. This algorithm is supported by the RANSAC-based MS [10] in a greedy manner, and therefore suffers from the aforementioned problems. Differently, Sofer et al. [12] performs 3D motion segmentation using Active Machine Learning (AML) [13] algorithm. Despite the fact that the AML algorithm provides high classification accuracy, its application specific training data requirement makes the method cumbersome.

Static-map building is a high interest topic in robotics and computer vision. Wang et al. [18] proposed methods that fulfils SLAM and Moving Objects Tracking (SLAM-MOT) simultaneously, using either map prior or motion consistency assumptions. However, they fail to handle the cases of slow motions and temporal stationary objects. Pomerleau et al. [19] proposed to detect the moving objects using ray-tracing technique. The spatial changes are measured in the built map which is obtained after using the motion from odometry sensors refined with ICP. This method also assumes that the dynamic parts only have a small scene coverage. Similarly, Ambrus et al. [20] proposed to maintain and update spatial models for Meta-room, using the Normal Distribution Transform Registration. A Meta-room is a reference static structure of an office to detect and update the dynamic objects. However, the initial requirement of clean reference model makes this method unsuitable for unknown dynamic environments.

III. 3D Motion Segmentation

Motion segmentation aims to determine different distinctive motions from the features' motion trajectories. In this context, we assume that a mobile robot captures a sequence of point clouds of a dynamic scene consisting of multiple moving objects. We also refer to the stationary objects or background as static scene parts. Similarly, the moving objects are called dynamic scene parts. Let a set of feature points be detected

and tracked across the points cloud sequence to represent the features' motion trajectories. Our objective is to group these trajectories into multiple subsets such that each subset represents a unique motion. More specifically, for n objects following distinct motions, there exist n subsets (or groups) of distinct trajectories, so called subspaces. All the trajectories from a subspace are linearly dependent among themselves under the rigid body motion assumption. In other words, all the feature trajectories lie in a union of n subspaces.

A. Motion in 3D-space

Considering X and Y as the Cartesian coordinate vectors of corresponding feature points in two point clouds related by a rigid body motion – rotation R and translation t, X and Y are related by:

$$\mathsf{Y} = \underbrace{\begin{bmatrix} \mathsf{R} & \mathsf{t} \end{bmatrix}}_{\mathsf{T}\in\mathbb{R}^{3\times 4}} \begin{bmatrix} \mathsf{X} \\ 1 \end{bmatrix}, \tag{1}$$

where T represents the 3D-space rigid transformation matrix. Let $\{\mathsf{X}\}_{i=1}^{P}$ represent a set of points that belong to a single rigid body in an arbitrary reference coordinate frame. If the moving coordinate frames $\{f_j\}_{j=1}^{F}$ are related to the reference by transformations $\{\mathsf{T}_j\}_{j=1}^{F}$, all the feature points Y_{ji} (*i.e.* j^{th} feature in i^{th} frame) can be expressed as:

$$\underbrace{\begin{bmatrix} \mathsf{Y}_{11} & \mathsf{Y}_{12} & \cdots & \mathsf{Y}_{1P} \\ \mathsf{Y}_{21} & \mathsf{Y}_{22} & \cdots & \mathsf{Y}_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ \mathsf{Y}_{F1} & \mathsf{Y}_{F2} & \cdots & \mathsf{Y}_{FP} \end{bmatrix}}_{Y\in\mathbb{R}^{3F\times P}} = \underbrace{\begin{bmatrix} \mathsf{T}_1 \\ \mathsf{T}_2 \\ \vdots \\ \mathsf{T}_F \end{bmatrix}}_{T\in\mathbb{R}^{3F\times 4}} \underbrace{\begin{bmatrix} \mathsf{X}_1 & \mathsf{X}_2 & \cdots & \mathsf{X}_P \\ 1 & 1 & \cdots & 1 \end{bmatrix}}_{X\in\mathbb{R}^{4\times P}}, \tag{2}$$

where T and X represent the motion and structure of a dynamic object, respectively. The columns of matrix Y contain the motion trajectory of feature points. Since all the entries of X's last row are one, the trajectory of feature points (*i.e.* the columns of Y) lie in a subspace of $\mathbb{R}^{3F}$ of dimension at most three. Note that the rank of Y can be at most of 4. In the case of multiple motions, let $\{S_\ell\}_{\ell=1}^{n}$ be a collection of n linear subspaces of $\mathbb{R}^{3F}$ with dimension $\{d_\ell\}_{\ell=1}^{n}$. If $\{Y_\ell\}_{\ell=1}^{n}$ correspond to n different unknown motions, the measurement matrix, say $\boldsymbol{Y}$, containing m measured trajectories can be denoted as:

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_m \end{bmatrix} = \begin{bmatrix} Y_1 & Y_2 & \cdots & Y_n \end{bmatrix} \mathsf{C}, \tag{3}$$

where $Y_\ell \in \mathbb{R}^{3F\times m_\ell}$ is a rank-d_ℓ matrix of the m_ℓ feature trajectories that lie in S_ℓ, and $\mathsf{C} \in \mathbb{R}^{m\times m}$ is an unknown permutation matrix. Equation (3) shows that the measured trajectories $\{\boldsymbol{y}_k\}_{k=1}^{m}$ lie in the union of n subspaces.

B. Sparse subspace representation and recovery

Referring to Equation (3), one can observe that the problem of 3D motion segmentation reduces to that of decomposing $\boldsymbol{Y}$ into $\{Y_\ell\}_{\ell=1}^{n}$ and C. This problem is addressed in [5] by solving a relaxed optimization problem, using the self-expressiveness property of the data. The solution is obtained under the assumption that every $\boldsymbol{y}_k$ can be represented as a combination of the columns of $\boldsymbol{Y}$ with $\boldsymbol{y}_k$ removed. To make the representation least ambiguous, the combination coefficients are kept as sparse as possible. Such solution is refereed as subspace-sparse representation (SSR). Therefore, a relaxed optimization problem for SSR can be written as:

$$\min \| \mathsf{C} \|_1, \text{ s.t. } \boldsymbol{Y} = \boldsymbol{Y}\mathsf{C}, \ \mathrm{diag}(\mathsf{C}) = 0. \tag{4}$$

Although this optimization problem is solved as in [5], our formulation includes a noteworthy modification that is critical to the problem at hand: the entries of C are forced to be non-negative so that similar motions in opposite directions are not considered to be the same. This happens especially (but not limited to) when the observed objects are moving along the robot's direction with twice speed. Such objects get categorized as background (because of the opposite relative motions), if the non-negativity constraint is not considered. Although the non-negative constrain brings more computational expense, it helps to avoid an extra step of post-processing. In the cases when the post-processing are reliable, this constrain automatically becomes optional. Furthermore, among many approaches for handling noisy data [5], we have adopted the most suitable technique based on our empirical evaluation. Consequently, for a $3F \times 3F$ identity matrix I_d and c_{ij} the entries of C, the final optimization problem is stated below:

$$\min \| \mathsf{C} \|_1, \text{ s.t. } \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{Y} & \mathsf{I}_d \end{bmatrix} \mathsf{C}, \mathrm{diag}(\mathsf{C}) = 0, c_{ij} \geq 0. \tag{5}$$

Once the sparse representation matrix C is computed, a weighted graph $\mathcal{G}$ with weights $\mathcal{W} = \mathsf{C} + \mathsf{C}^{\mathsf{T}}$ is built. The segmentation of trajectories into different subspaces is obtained by applying spectral clustering [9] method on the Laplacian of graph $\mathcal{G}$. Alternatively, any other clustering method can be applied on graph $\mathcal{G}$ for the same task.

C. Implementation details

The proposed 3D motion segmentation algorithm (3D-SSC) is an extension of the existing image based SSC (2D-SSC). The readers are strongly recommended to refer [5] for its theoretical derivations. For implementation aspects, our system is designed based on the 2D-SSC toolbox [5], with the following critical modifications: a) A modified system with 3D data based-SSC; b) Non-negative constraint in sparse representation to distinguish similar motions in opposite directions; c) Diagonal identity constraint (see Equation (5)) adoption for corrupted data recovery. The proposed system offers the following advantages:

- Direct 3D space motion analysis: perspective effects produced by the affine projection assumption is avoided.
- More precise motion behaviour analysis: object motion estimation, namely the rotation and translation, can be precisely recovered from the segmented 3D motion trajectories for each moving object.
- Better perception of scene structure: the 3D data provide more meaningful information, e.g. geometric structures, continuity or discontinuity, for better scene understanding.

IV. Static-map Building

For controlled environments, one can safely assume that the static-map can always be built after physically removing the dynamic object from the scene, or by restricting them to move

while building the map. However in the real-world outdoor scenes, it is very often impractical to restrict the objects not to move, such as driving cars and walking pedestrians, for the sake of map building. A common practice involves the selection of most appropriate time frame so that the number of moving objects can be minimized. Any act of leveraging from this restriction makes the presence of dynamic objects unavoidable. In such scenarios, the process of static-map reconstruction demands the detection and the removal of dynamic objects while building the map, or preferably before.

To reconstruct the static-map in an outdoor environment, we suggest to use a mobile robot equipped with both 2D camera and 3D sensor. Ideally, a 3D sensor alone is sufficient for the proposed static-map building framework. However in practice, construction of meaningful trajectories using only 3D data is undermined by the lack of robust 3D feature descriptors. Therefore, we also make use of the 2D camera – calibrated and synchronized with 3D sensor and sharing the same field of view. Although the RGB-D camera is one good example, any combination of 2D camera and 3D sensor would suffice as long as the aforementioned criteria are satisfied. Doing so allows us to associate 3D points to their 2D descriptors by projecting them onto the image plane. The process of feature trajectory construction is followed by our 3D MS method as proposed in Section III. The point clouds of the static scene parts are later obtained after performing region growing on the segmented motion trajectories. Finally, the static-map is built by registering these point clouds with the help of RANSAC. The complete pipeline of the proposed static-map building method is depicted in Fig. 1.

A. Feature trajectory construction and segmentation

The feature trajectories are constructed using both 2D and 3D measurements. First, we project all the 3D scene points of the reference frame onto its image. These projections are considered as 2D feature points and tracked across the image sequence using a dense optical flow method. To cover a wide speed range, coarse-to-fine dense Optical Flow [16] tracking algorithm has been adopted. The 3D feature trajectories are then retrieved from 2D feature trajectories after establishing 2D-to-3D correspondences similar to [22]. We define dynamic coverage as the area that the dynamic objects cover in an image. Our primary interest is to perform robust MS, while addressing a wide range of dynamic coverages and speeds. For example, if the dynamic object covers a small part of the image or quickly changes its appearance because of a high speed, only a small fraction of the tracked features belong to this object. This makes the data highly imbalanced, causing numerical instability during subspace-sparse representation. To address this problem, we introduce a flow-likelihood-based sampling of the trajectories. Let $\{\mathbf{v}_k\}_{k=1}^m$ be the measured speeds corresponding to the trajectories $\{\boldsymbol{y}_k\}_{k=1}^m$ (refer Equation (3)). If $\{c_k\}_{k=1}^m$ are the binary classes (dynamic=1, and static = 0) assigned to each trajectory, the likelihood function is defined as

$$\mathcal{L}(c_k = 1|Z) = e^{\|\mathbf{v}_k - \bar{\mathbf{v}}\|^2/\sigma^2}, \tag{6}$$

where $\bar{\mathbf{v}}$ and σ are the median speed and standard deviation respectively. A subset of the feature trajectories for MS is selected based on the likelihood measure of Equation (6). This sampling method avoids the problem of having too many samples from the background, hence balancing the data for the optimization problem of Equation (5). During this process, we also reject all the trajectories that do not follow the smooth motion, categorizing them as outliers. Fig. 2 demonstrates the effectiveness of the proposed flow-likelihood-based sampling approach, in which more features on moving objects (such as vans, train, cyclist and pedestrians) are sub-sampled using the flow-likelihood-based sampling method (last column in Fig. 2).

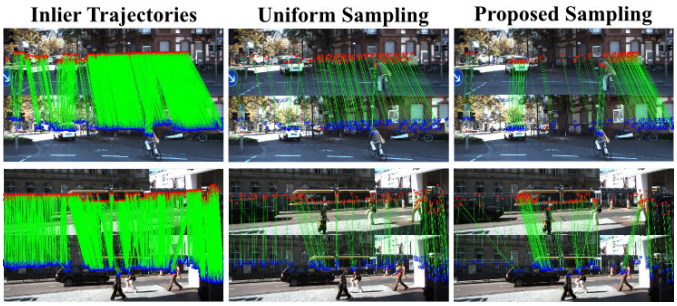


Fig. 2: Results of uniform sampling vs. the proposed flow-likelihood-based sampling: the green lines show the tracked features from the first frame to the last frame. The last column shows that more features are sampled from moving objects.

Once the segmented feature trajectories are obtained, a multi-seeded Region Growing [17] technique is applied on the point clouds to densely segment the moving objects. After segmenting all the dynamic objects, the remaining point clouds represent the static scene parts.

B. 3-point RANSAC registration

Given the segmented point clouds, the static-map is built by registering multiple point clouds of the static parts. The registration is performed using 3-point RANSAC on rigid transformation parameters. In fact, the segmented motion trajectories also allow us to obtain the dense reconstruction of dynamic objects in a very similar manner. Recall the rigid transformation of Equation (1). Let g be the Gibbs representation of a rotation matrix R. $\mathsf{G} = [\mathsf{g}]_\times$ and I_3 are 3×3 skew-symmetric and identity matrices, respectively. Using Cayley Transform [21], R can be expressed as:

$$\mathsf{R} = (\mathsf{I}_3 + \mathsf{G})^{-1}(\mathsf{I}_3 - \mathsf{G}). \tag{7}$$

Using Equation (7), Equation (1) can be rewritten as:

$$(\mathsf{I}_3 + \mathsf{G})\mathsf{Y} = (\mathsf{I}_3 - \mathsf{G})\mathsf{X} + (\mathsf{I}_3 + \mathsf{G})\mathsf{t}. \tag{8}$$

If the second term on right hand side of Equation (8) is replaced by a new vector $\tilde{\mathsf{t}}$, it can be written as

$$(\mathsf{Y} - \mathsf{X}) = -(\mathsf{Y} + \mathsf{X})\mathsf{G} + \tilde{\mathsf{t}}. \tag{9}$$

Note that the Equation (9) is linear in the entries of g and $\tilde{\mathsf{t}}$. Each pair of corresponding points provides 2 independent equations, for a system of 6 unknowns. Therefore, only 3

correspondences are required to solve this system linearly. It is straightforward to recover R and t from its solution.

Cayley transform based rotation matrix representation is well known in geometry, however, its usage in robotics is shadowed due to its inability to represent the rotation of $180°$. In fact, the Gibbs vector for rotation angle θ and axis $\hat{\mathrm{g}}$ is expressed as: $\mathrm{g} = tan(\theta/2)\hat{\mathrm{g}}$. The entries of g start behaving badly from $\theta > 90°$, due to the tangent nature. For $\theta < 90°$, it can be safely used to estimate the rigid motion. On the positive side, this representation offers a linear solution using minimal 3-point correspondences. More importantly, the over-determined system constructed from all inliers, at the refinement stage of RANSAC, can be solved using linear least-square method on exact 6 rigid motion parameters. Note that the drawback of Cayley representation is not really a problem for our application, because the angle between two consecutive views in any practical scenario is always smaller than $90°$.

V. Experiment

We conducted several experiments with both synthetic and real data. Two kinds of real data, one acquired using Microsoft Kinect RGB-D camera, and another from benchmark KITTI dataset [23] were used. Our experiments show the feasibility of the proposed 3D-SSC in segmenting the 3D trajectories. Furthermore, both quantitative and qualitative results of reconstructed static-maps using the proposed method are discussed in details. All the experiments are conducted in a computer with Intel Quad Core i7-2.7GHz, 32GB Memory.

A. 3D-SSC motion segmentation simulation

We build a system that contains multiple moving objects under different noise conditions to verify the robustness of the algorithm. More specifically, a set of synthetic data is generated with *n* moving cubes with different sizes, positions, orientations, and motions. The motion feature trajectories are randomly selected to generalize the algorithm evaluation. To quantify the robustness of the algorithm under different noise levels, the miss-classification rate is defined as $\eta = \frac{\#\text{ miss-classified features}}{\#\text{ total features}}$. To test the performance of the algorithm under different noise levels and multiple motions, various levels of white Gaussian noise (from $0\%,\ 4\%,\ \cdots,\ 16\%$) are introduced to feature locations. Fig. 3 shows that the 3D-SSC behaves very robustly under 12% of noise for at least up to 10 moving objects.

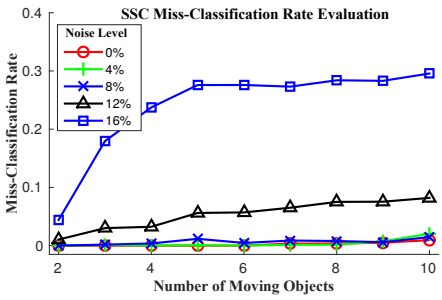


Fig. 3: Averaged 3D-SSC MS performances on 50 tests.

B. Evaluation using Kinect data

To evaluate the performance of the algorithm on real 3D data, a set of RGB-D sequences using Microsoft Kinect is recorded, see Fig. 4. In the experiment, 5 moving objects with different shapes are involved, namely the book, bottle, mug, lamp, and box. All the moving objects are attached with a chessboard pattern for the ease of feature selection and annotation. The details of the experiments are summarized in Table I, where the columns represent the frame length, the number of features, and the segmentation accuracy $(1 - \eta)$, respectively. As can be seen from Table I, the trajectories' length for different objects are different, thus representing the incomplete trajectory cases. These results show that the 3D-SSC algorithm is able to correctly and precisely segment the 3D feature trajectories in a controlled indoor environment.

Fig. 4: Kinect data.

Objects	Len.	Feat.	Acc.(%)
Mug	15	18	100
Bottle	32	12	100
Lamp	24	18	100
Box	22	18	100
Book	20	16	93.75

TABLE I: Kinect MS results.

C. Evaluation on KITTI dataset

To evaluate our system with realistic outdoor scenes, we conduct extensive experiments on the KITTI dataset [23]. The experiments are conducted with four different datasets, namely Highway, Junction, Station, and Market. These datasets have been selected with different frame lengths, number of moving objects, and number of feature trajectories. The details of all four datasets are provided in Table III. In this table, the speed indicates the relative speed of the moving objects with respect to the camera. Note that the dynamic objects cover a wide range of speeds, representing both fast and slow motions.

1) MS Evaluation: The feature trajectories are constructed using the dense optical flow tracking approach and sub-sampled based on the flow-likelihood sampling technique. As can be observed in Fig. 5a, a significant number of features belong to the dynamic parts, although they cover relatively small region. Such feature distribution helps to balance the data for the sparse representation, thanks to the likelihood-based sampling. Fig. 5b shows the 3D feature trajectories. Fig. 5c and Fig. 5d show segmentation results obtained by 2D-SSC [5] and 3D-SSC, respectively. Note that the 2D-SSC MS fails to categorize the road sign as a static scene part.

The results obtained using 2D-SSC MS as well as 3D-SSC MS for all four datasets are summarized in Table III. The segmentation performances are assessed by the popular *Sensitivity* and *Specificity* metrics [24]. We also report another measurement, reported as Seg. $\geq 50\%$, counting the number of objects with more than half of the feature points correctly classified. Finally, the eigenvalue ratios are computed by $\rho = \frac{\sum_{i=1}^{n} \lambda_i}{\sum_{j=1}^{m} \lambda_j}$, where n and m are the number of motions and the total number of trajectories, respectively. A higher value of ρ denotes a better representation of the motion subspaces.

Seq.	# Objs.	Corr.	Incorr.	Dyn. Acc.(%)	Stc. Acc.(%)	Time (min.)
Highway	1	1	0	97.55	100	6.00
Junction	2	2	0	91.02	100	13.40
Station	5	5	1	91.60	92.47	3.16

TABLE II: Static-map quantification.

Table III also reports the computation time for both methods (for the software developed in MATLAB).

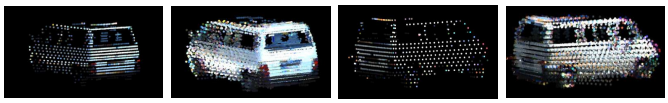
Three main observations should be noted: a) 2D-SSC has very high sensitivity with less motions, while its performance decreases significantly as the motion number increases. In contrast, the proposed 3D-SSC algorithm remains robust against abundant motions. b) The 3D-SSC results are more meaningful in the sense that even when the algorithm cannot perfectly classify all the trajectories, the motions can still be correctly categorized based on the trajectories' voting. c) The 3D-SSC performs superior to 2D-SSC due to the fact that the subspace representation on direct 3D space is more compact than that of 2D-SSC. This can be observed from the eigen ratio column of Table III.

2) Static-map Evaluation: Thanks to the effectiveness of the proposed MS method, the static-maps of three dynamic scenes are reconstructed, namely the Junction, Highway, and Train station, see Fig. 6, 7, 8. To illustrate the quality of the reconstructed static-maps, the full scene reconstructions using state-of-the-art method [22] are also shown sidewise. Few 2D frames from the sequence are displayed for the motion visualization. In details, the Fig. 6 (Junction) shows the reconstructed static-scene in a long sequence, the moving car and the cyclist are detected and segmented correctly. Though there are few frames rejected due to the loss of tracked features, the proposed system is robust enough to reconstruct a long sequence with significantly changing lighting conditions.

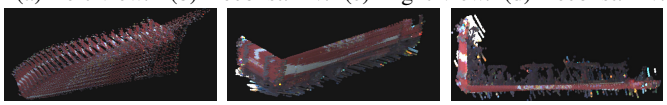
In the Highway sequence, the qualitative analysis between Fig. 7a and Fig. 7b show that the static scene part of our map is significantly better than that of [22]. For instance, the red rectangle region in Fig. 7a highlights the tree shadow which is barely recognized. On the contrary, the same shadow in Fig. 7b has been recovered more realistically. In the close-up view of all the built maps, similar differences are abundant. A more challenging dataset shown in Fig. 8 (Train Station) contains fast moving car and slowly moving pedestrians, with intermittently occluded train by moving objects. Interestingly, all moving objects: pedestrians, fast driving car, and occluded train are detected and removed correctly in the reconstructed static-map (see Fig. 8c). Recall that the objects moving in the same direction with similar speed share the same motion subspace. Therefore, the car and the train are grouped together (blue objects in Fig. 8b), so as the two pedestrians (yellow objects in Fig. 8b). In fact, such motion grouping simplifies the complexity of scene understanding based on the motion behaviours.

Table II summarizes the quantification results of the static-map reconstruction. Starting from the second column, they represent the number of moving objects, the number of correctly and incorrectly removed objects, and accuracies in removing the dynamic objects and maintaining the static scene parts. The metric Dynamic Accuracy is defined by $Dyn.\ Acc. = \frac{\#\ of\ Points\ Segmented\ from\ Dynamic\ Objects}{Total\ \#\ of\ Points\ from\ Dynamic\ Objects}$, and the Static Accuracy is defined in a similar manner. Note that these measurements are made on the densely segmented point clouds, unlike in Table III. A higher dynamic accuracy (Dyn. Acc.) means a better removal of dynamic objects. Similarly, the higher static accuracy (Stc. Acc.) stands for a better maintenance of the static scene parts. Results show that the dynamic objects are removed correctly with very high accuracy, meanwhile, the static scene parts are maintained very well. The reported computation time includes the time for both MS and static-map reconstruction.

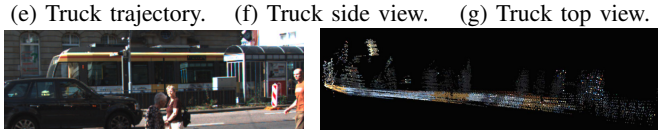
As offered by our static-map building framework, dense reconstructions of the dynamic objects are recovered using multiple frame measurements, as shown in Fig. 9. Firstly, Fig. 9a-9d show two views of the denser reconstruction of a car along with their single frame representations. Secondly, Fig. 9e shows the multi-frame grouping of the truck's point clouds in a common coordinate frame, obtained using [22]. It goes without saying that this representation can hardly be identified as a truck. On the contrary, the reconstructed truck using our method has very high quality, see Fig. 9f-9g. Thirdly, the full reconstruction of the moving train is shown in Fig. 9i obtained from its partial measurements due to dynamic occlusions.



(a) Left view. (b) Reconst. LV. (c) Right view. (d) Reconst. RV.



(e) Truck trajectory. (f) Truck side view. (g) Truck top view.



(h) Train in crowded scene. (i) Reconstructed train.

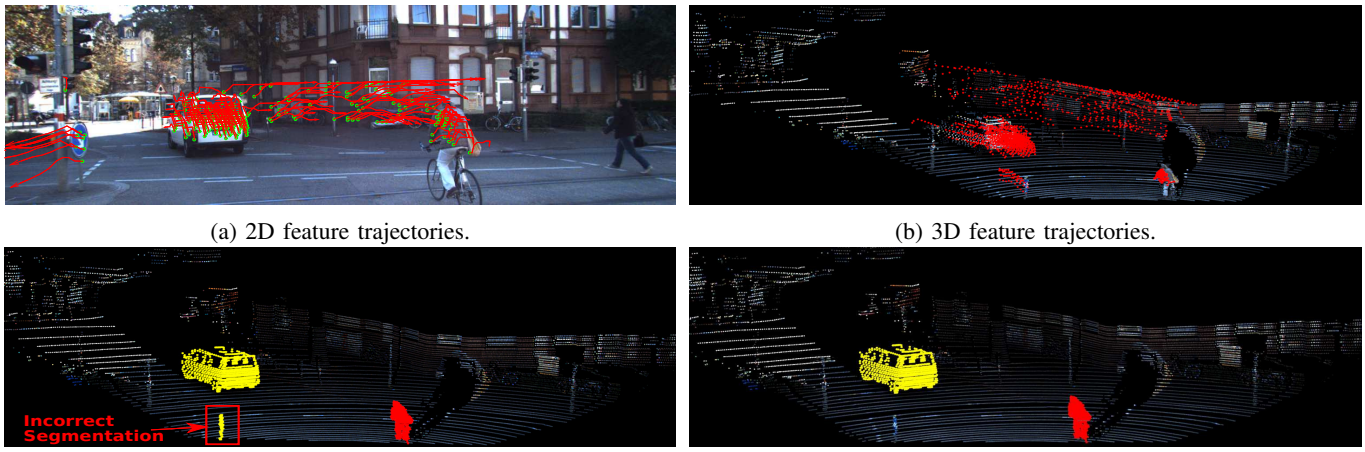
Fig. 9: Reconstructed moving objects. (a) and (c) show the left and right side view of the car in one frame, respectively. (b) and (d) show the denser reconstruction of left and right side view of the car, respectively. (e) shows the trajectory of the moving truck. (f) and (g) show the side view and top view of the reconstructed truck, respectively. (h) shows the train in a crowded environment, occluded by foreground moving objects. (i) shows the reconstructed running train from 9 frames.

VI. Conclusion and Future Work

We have proposed a novel framework for 3D motion segmentation using Sparse Subspace Clustering algorithm that categories the static scene parts and multiple moving objects. The proposed method has been tested with extensive experiments and outperforms its 2D based counterpart, especially when rich moving objects are involved. Our approach of sampling sparse feature trajectories based on their flow likelihood, and the proposed motion segmentation approach

Seq.	# Frames	# Objs.	# Feat.	Speed (m/s)		Sensitivity		Specificity		Seg. $\geq 50\%$		Eigen. Ratio ρ		Time(min.)	
				Min.	Max.	2D	3D	2D	3D	2D	3D	2D	3D	2D	3D
Highway	45	2	122	4.87	7.22	**1.0**	0.95	**1.0**	**1.0**	2	2	0.0250	0.0264	3.54	4.83
Junction	70	3	83	0.50	5.15	**1.0**	0.98	0.77	**0.99**	3	3	0.0398	0.0399	9.61	12.85
Station	9	6	77	0.35	7.12	0.62	**0.95**	0.31	**0.66**	3	**6**	0.0789	0.0979	1.39	1.68
Market	13	9	50	0.39	1.34	0.88	**1.0**	0.68	**0.98**	6	**9**	0.0666	0.1907	1.61	2.09

TABLE III: 2D-SSC vs. 3D-SSC in MS on KITTI dataset: *Seg.* $\geq 50\%$ counts the objects with more than 50% feature points correctly classified. $\rho = \frac{\sum_{i=1}^{n} \lambda_i}{\sum_{j=1}^{m} \lambda_j}$, where n and m are the number of motions and the total number of trajectories, respectively.



(a) 2D feature trajectories.

(b) 3D feature trajectories.

(c) 3D Region growing segmentation based on 2D MS [5].

(d) 3D Region growing segmentation based on our method.

Fig. 5: 2D vs. 3D MS results: (a) and (b) show the 2D and 3D feature trajectories for 10 frames, respectively. Arrows in (a) represent the direction of the feature motions. (c) and (d) show the 3D region growing segmentation based on the segmented feature trajectories using 2D-SSC and our 3D-SSC algorithm, respectively. Must view in color.

(a) Reconstructed full scene using [22].

(b) Reconstructed static-map.

Fig. 6: Junction sequence results: (a) shows the full scene 3D reconstruction using 80 frames. (b) shows the reconstructed static-map without moving objects. Last row images show the corresponding image sequence for every 15 frames.

can handle wide range of motions, both in terms of magnitude, speed and coverage. Furthermore, the proposed static-map building pipeline reconstructs photo-realistic maps, both for static and dynamic scene parts in an uncontrolled outdoor environment. The proposed framework was tested with off-line data, yet, it can be adapted to build a SLAM-like on-line system, where data are acquired and processed piece by piece. In the future, more robust feature tracking algorithm, such as cross-frame optical flow, will be implemented to handle short-term occlusion problem. Also, multi-object trackers initialized with the detected moving objects should be developed to better understand the motions.

REFERENCES

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

[2] Klein, G., & Murray, D. Improving the agility of keyframe-based SLAM. In ECCV, 2008.

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

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

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

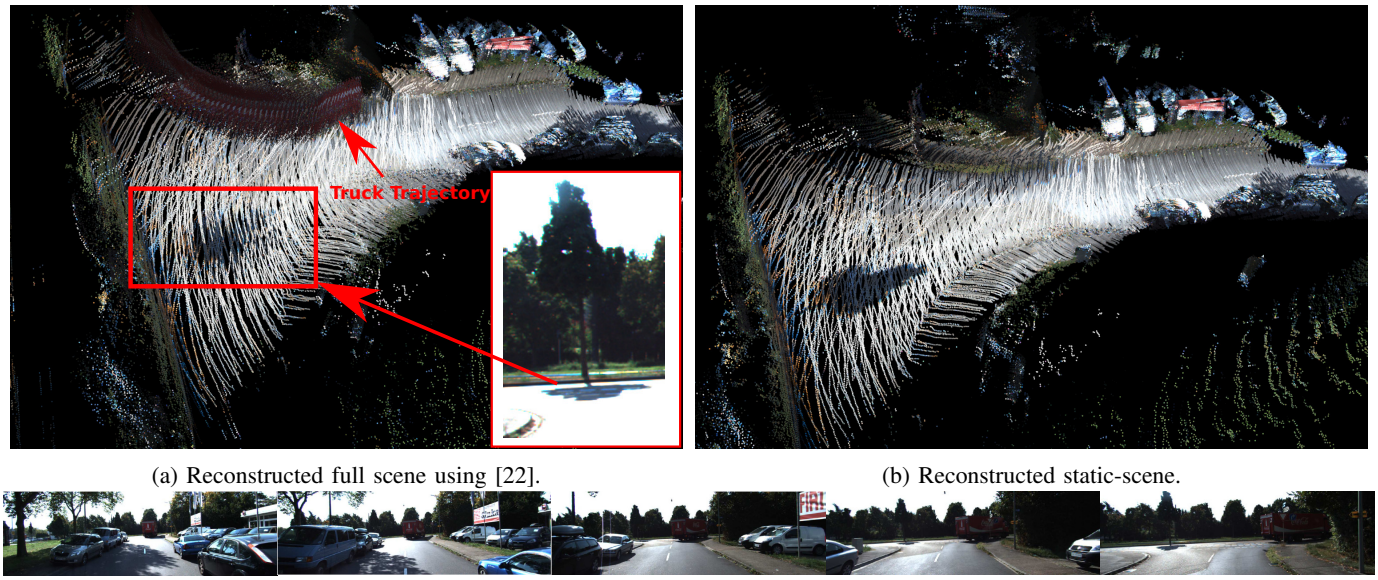
[6] Rao, S., Tron, R., Vidal, R., & Ma, Y. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. In TPAMI, 2010.

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

[8] Michael G. & Stephen B., CVX: Matlab software for disciplined convex programming, version 2.0 beta, 2013.

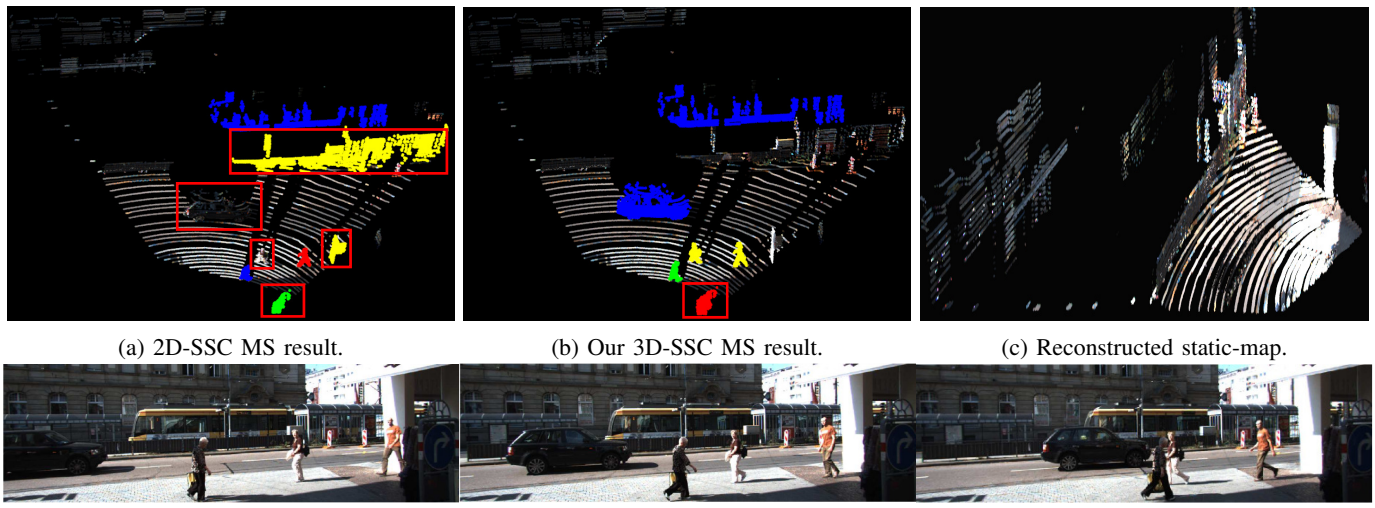
[9] Ng, A. Y., Jordan, M. I., & Weiss, Y. On spectral clustering: Analysis and an algorithm. In Adv. neural inf. proc. systems, 2002.

[10] Yan, J., & Pollefeys, M. Articulated motion segmentation using ransac



(a) Reconstructed full scene using [22]. (b) Reconstructed static-scene.

Fig. 7: Highway sequence static-map reconstruction results: (a) shows the full scene 3D reconstruction using 45 frames. The red rectangle shows the reconstruction of the tree shadow. (b) shows the reconstructed static-map without moving objects. Last row images show the corresponding image sequence every 10 frames.



(a) 2D-SSC MS result. (b) Our 3D-SSC MS result. (c) Reconstructed static-map.

Fig. 8: Train station sequence static-map reconstruction results: (a) and (b) show 2D-SSC and 3D-SSC MS results, respectively. Incorrect segmentations are highlighted with red rectangles. (c) shows the reconstructed static-map without moving objects from 9 frames. Last row images show some selected corresponding sequential images. Must view in color.

with priors. In Dynamical Vision, 2007.
[11] Stckler, J., & Behnke, S. Efficient Dense Rigid-Body Motion Segmentation and Estimation in RGB-D Video. In IJCV, 2015.
[12] Sofer, Y., Hassner, T., & Sharf, A. Interactive Learning for PointCloud Motion Segmentation. In Computer Graphics Forum, 2013.
[13] Settles, B. Active learning literature survey. Tech. Report, 2010.
[14] Perera, S., Barnes, N., Xuming He, Izadi, S., Kohli, P., & Glocker, B. Motion Segmentation of Truncated Signed Distance Function Based Volumetric Surfaces. In WACV, 2015.
[15] Cayley, A. About the algebraic structure of the orthogonal group and the other classical groups in a field of characteristic zero or a prime characteristic. In Reine Angewandte Mathematik, 1846.
[16] Liu, C., Yuen, J., Torralba, A., Sivic, J., & Freeman, W. T. Sift flow: Dense correspondence across different scenes. In ECCV, 2008.
[17] Vosselman, G., Gorte, B. G., Sithole, G., & Rabbani, T. Recognising structure in laser scanner point clouds. In Int. archives of photogrammetry, remote sensing and spatial information sciences, 2004.
[18] Wang, C., Thorpe, C., & Thrun, S. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In ICRA, 2003.
[19] Pomerleau, F., Krusi, P., Colas, F., Furgale, P., & Siegwart, R. Long-term 3D map maintenance in dynamic environments. In ICRA, 2014.
[20] Ambrus, R., Bore, N., Folkesson, J., & Jensfelt, P. Meta-rooms: Building and maintaining long term spatial models in a dynamic world. In IROS, 2014.
[21] Alismail, H., Baker, L. D., & Browning, B. Continuous trajectory estimation for 3D SLAM from actuated lidar. In ICRA, 2014.
[22] Paudel, D. P., Demonceaux, C., Habed, A., Vasseur, P., & Kweon, I. S. 2D-3D camera fusion for visual odometry in outdoor environments. In IROS, 2014.
[23] Geiger, A., Lenz, P., & Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In CVPR, 2012.
[24] Fawcett, T. An introduction to ROC analysis. In Pattern recognition letters, 2006.
[25] Papon, J., Kulvicius, T., Aksoy, E. E., & Worgotter, F.. Point cloud video object segmentation using a persistent supervoxel world-model. In IROS, 2013.
[26] Koo, S., Lee, D., & Kwon, D. S.. Incremental object learning and robust tracking of multiple objects from RGB-D point set data. In J. Vis. Commun., 2013.