Motion Analysis for Dynamic 3D Scene Reconstruction and Understanding

Cansen Jiang

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE BOURGOGNE

Motion Analysis for Dynamic 3D Scene Reconstruction and Understanding

CANSEN JIANG

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE BOURGOGNE

N° X X X

THÈSE présentée par

CANSEN JIANG

pour obtenir le

Grade de Docteur de

l'Université de Bourgogne

Spécialité : **Instrumentation et Informatique de l'image**

Motion Analysis for Dynamic 3D Scene Reconstruction and Understanding

Unité de Recherche :

Laboratoire Électronique, Informatique et Image

Soutenue publiquement le 14 décembre 2017 devant le Jury composé de :

MARIE ODILE BERGER	Rapporteur	INRIA Nancy, France
PATRICK BOUTHEMY	Rapporteur	INRIA Rennes, France
PASCAL MONASSE	Examinateur	Ecole des Ponts ParisTech, France
ATILLA BASKURT	Examinateur	INSA Lyon, France
DAVID FOFI	Examinateur	Université de Bourgogne, France
CÉDRIC DEMONCEAUX	Directeur de thèse	Université de Bourgogne, France
YOHAN FOUGEROLLE	Co-encadrant	Université de Bourgogne, France

ACKNOWLEDGMENTS

"We are like dwarfs on the shoulders of giants, so that we can see more than they, and things at a greater distance, not by virtue of any sharpness of sight on our part, or any physical distinction, but because we are carried high and raised up by their giant size."

- Bernard de Chartres, *12th century AD*

First and foremost I would like to express my sincere gratitude to my supervisors Cédric Demonceaux, Yohan Fougerolle, and David Fofi for their great helps and inspirations through my Ph.D. study. Not only have I learned a great deal but it has also been an amazing experience working with them in our closely knit group. I am very thankful to Cedric for his patience, motivation, enthusiasm, immense knowledge, and continuous support which have been instrumental to the success of my Ph.D. I have also highly valued his friendship, trust and understanding. I am very grateful to Yohan for his support and encouragement. I always admire for his rigorous academic attitude and the critical thinking ability. He has been a friend as well as a supervisor to me. I would also like to convey my sincere thanks to David for his insightful comments and valuable advices in all our discussions.

I would also like to thank the jury members for their kindness of participating in my Ph.D. defence committee. Many thanks to Prof. Marie-Odile Berger and Prof. Patrick Bouthemy for their careful reading the thesis and their constructive remarks, and to Prof. Atilla Baskurt and Prof. Pascal Monasse for investing their precious time on playing the role of the president of the defence and the thesis examiners respectively. Their involvement has undoubtedly widened the understanding of this thesis and added new perspectives on our research problems.

Thanks to my visit to Korean Advance Institute of Technology (KAIST) as a visiting scholar in 2016, although only for a month, their cutting-edge research perspectives and solid theoretical foundation make me open-minded and inspire me to work harder. Further-

more, I would also like to thank all my colleagues (especially Francois) at RCV laboratory, KAIST, who made my stay in Korea very pleasant and unforgettable.

Besides the realm of research, I also had opportunities to interact with graduate and undergraduate students as a teacher, a tutor, or a thesis co-supervisor during the last three years. I really enjoyed the time working with Dennis as his thesis co-supervisor. Teaching the BSc/MsCV/Vibot/MAIA students was very happy time. I would really like to thank Ralph, Raphael, Yohan, Omar and Desire for their conscious and unconscious guidance for being a good teacher.

I very appreciate the MsCV programme team (especially Fabrice, David, and Herma) and the Le2i group (especially Nathalie, Ralph, and Christophe L.) who gave me tremendous support and kindness for my 5-year stay in Le Creusot.

I am very thankful to all my friends and colleagues for their kindness and friendship during the past few years. I especially like to thank DP for his guidance, ideas and encouragement in research and beyond. He has always been a role model who I follow. I also like to wholeheartedly thank Qinglin, Mojdeh, Guillaume, David S., Francois, and Desire for their taking good care of me. My life would have been much more difficult without their friendship and kindness (especially driving me to the supermarkets and inviting me for parties). Of course, there are many more friends who I would like to thank. To name a few: Zawawi, Abir, Nathan C., Nathan P., Mohammad, Ashvaany, Thomas, Ahlem, Sik, Mazen, Lee, Juan, Jilliam, Deepak, Jermie, Shaifali, Luis, Kristina, Ran, Yifei, Songyou, Ziyang, Chunxia, Peixi, Devesh, Suman, Taman, Lijia, Maya, Ajad, Chinmay, Priyanka, Usman, Osama, Vishnu ... Many thanks to them for given me many many precious memories.

Finally and above all, I would like to thank my family for giving me their love and support and making everything I have done possible. I also like to express my endless love to Yuanyuan for her kindness and love to me. Their love and continued kindness have always provided me the reasons to persevere.

CONTENTS

1

INTRODUCTION

“The three R’s of Vision: Recognition, Reconstruction, Reorganization."

- Jitendra Malik, *UC Berkeley*

1.1/ CONTEXT AND MOTIVATION

Since the 1960s, computer vision has been a very active research field. The ultimate goal of computer vision is to let the machine perceive the world as human. For this purpose, there are three major problems to solve: recognition, reconstruction and reorganization. This thesis tackles the challenge of reconstruction, particularly 3D scene reconstruction of dynamic environments. There are various purposes on 3D scene reconstruction, such as historical building preservation, film industry, city planning, map-based autonomous navigation, etc. As a general goal, the sought 3D map should be of high quality so that the distance measurement is precise (e.g. less than 10mm) and clean (e.g. only contains the desired objects). Early studies on the 3D reconstruction problem focus on the passive sensing approaches, e.g. Structure-from-Motion approaches, that estimate the scene’s 3D structure based on the multi-view geometric constraints. However, such approaches are very often imprecise and sensitive to noise, e.g. bottom-left image of Fig. 1.1. Recent advances rely on the active sensing techniques, e.g. laser scanners, that offer precise 3D point cloud measurement of the scene, e.g. bottom-right image of Fig. 1.1. Such accurate 3D data allow the possibility of reconstructing high quality 3D maps which provide reliable knowledge that plays a vital role in scene modelling, understanding, and landmark-based robot navigation.

When reconstructing the 3D maps, many approaches assume that the environment is nearly static with very few moving objects. However, in practice, some scenes can be highly dynamic, especially the uncontrolled outdoor scenes, e.g. a train station at rush hour. The reconstructed 3D maps of such cases are of low quality due to two reasons: (i) the dynamic parts provide very noisy information for the registration process. (ii) the moving object tracks also introduce numerous outliers all over the process. Therefore, it is necessary that the moving objects are detected, segmented and removed for accurate 3D map reconstruction.

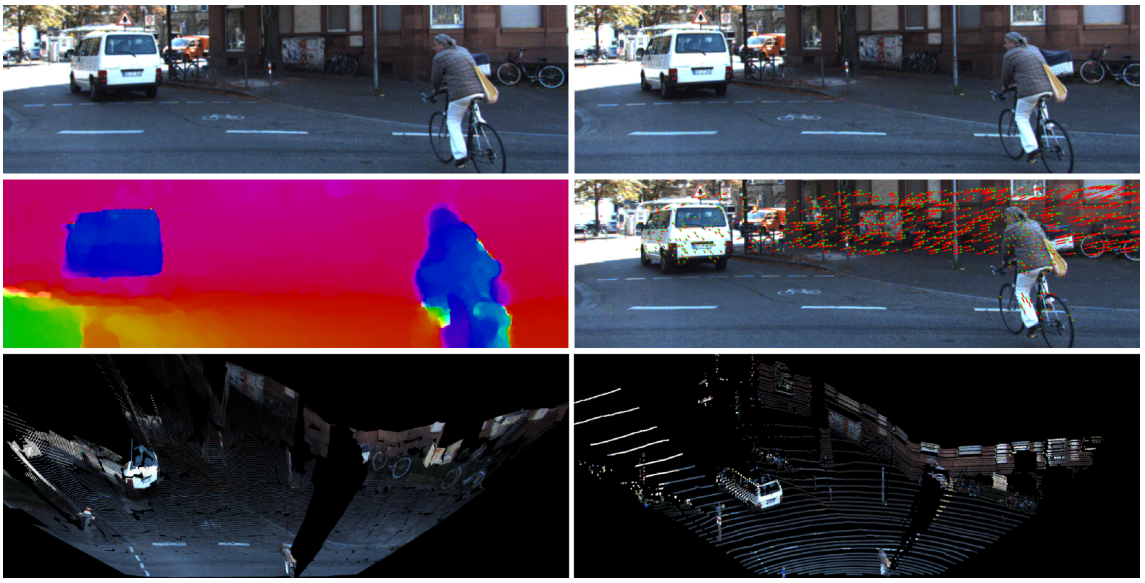


FIGURE 1.1 – Examples of moving objects in two consecutive frames. Middle-left is the respective dense optical flows where the colors encode the flow directions. Middle-right shows some detected features (green dots) with their motion trajectories (red lines). The bottom images are the reconstructed 3D point clouds using stereo vision (left) and Velodyne 3D laser scanner (right), respectively.

Identifying moving objects is a natural ability for humans. For example, Fig. 1.1 shows two consecutive frames of a video. It is effortless for a human to point out the moving objects (e.g. the van and the cyclist) and separate the feature trajectories with respect to their motions. However, these are very difficult tasks for machines. To address this difficulties, this thesis focuses on the study of motion detection and segmentation and the reconstruction of high quality 3D maps. In particular, motion detection discovers the moving objects from a dynamic scene, while the motion segmentation separates them according to their distinctive motion behaviours. These are important problems because their outputs are beneficial for many related fields in computer vision, such as video analysis, scene understanding, and multi-body 3D reconstruction.

1.2/ SCOPE AND CHALLENGES

This thesis addresses the problem of dynamic scene 3D reconstruction via the analysis of moving objects in terms of their detection and segmentation. By priorly removing the detected moving objects, the static map is reconstructed by registering the static scene parts of the dynamic sequence, while the multiple rigidly moving object reconstructions are obtained from the registration of the segmented motions. Furthermore, semantic information is learned using images and transferred to the reconstructed 3D maps. Specifically, we have investigated the motion detection and segmentation problems in two different scenarios, as well as solutions for precise sparse point cloud registration problem, as follows:

i **Unknown camera motion case:** we consider a 2D-and-3D camera system which is rigidly attached to a moving platform (e.g. a moving car), see Fig. 1.2 as an example. The 2D camera is one rgb camera (e.g. the Point-Grey-Flea-2 color camera) and the 3D camera is a laser scanner (e.g. the 64-layer Velodyne laser scanner). The 2D-3D camera system is assumed to be fully calibrated so that the intrinsic and extrinsic parameters of the sensors are known. Additionally, the data acquisition of both 2D and 3D cameras are assumed to be synchronized for immediate 3D to 2D correspondence.

ii **Known (or precisely estimated) camera motion case:** here, we only make use of the point cloud data acquired from 3D laser scanners. Given a point cloud se-

FIGURE 1.2 – Example of a 2D-and-3D camera system [1] rigidly mounted on a mobile platform. Such system contains a Flir color camera (red bounding box) and a Velodyne 3D laser scanner (blue bounding box)

quence, the camera ego-motions can be recovered precisely by using the traditional Simultaneous Localization and Mapping (SLAM) approaches. The estimated 3D camera motions are then compensated prior to the detection and segmentation of the moving objects.

iii **3D Map Reconstruction and Enhancement:** to register a set of sparse point clouds, we take advantage of well calibrated and synchronized 2D-3D camera systems. Once the 2D and 3D correspondences are established, the 3D to 3D feature correspondences can be inferred from their 2D to 2D feature matching pairs, which allows a fast (but rough) point cloud registration using minimal 3-point RANSAC algorithm. Afterwards, the precise registration is achieved by using Iterative Closest Point (ICP)-based approaches.

1.2.1/ UNKNOWN CAMERA MOTION CASE

In such cases, it is very challenging to detect and segment the moving objects since both the static and dynamic scene parts appear to be moving. To address this problem, the motion segmentation methods which deal with feature tracks obtained by tracking (e.g. the Kanade-Lucas-Tomasi (KLT) feature tracker and optical flow tracking) are very effective. Most of these methods rely on the affine projection model which assumes that the scene is planar. Moreover, when both the camera and the dynamic objects move along the same direction with similar velocity, the resulting feature tracks are numerically unstable due to the perspective projection effects. To overcome such difficulties, our motion segmentation directly segments the raw 3D feature trajectories. Since the feature tracks are analysed in a 3D space, our method does not rely on the affine projection or the perspective projection assumptions.

Furthermore, due to the drifting and occlusion problems in feature tracking, the feature trajectories inevitably contain noise, outliers, or missing entries. We design a robust motion segmentation algorithm in the presence of noise and outliers, as well as a novel feature trajectory construction approach to handle the problem of feature tracking loss. We also consider the practical scenarios that feature trajectories from dynamic objects and static scene parts are imbalanced. In this regard, we propose a flow-likelihood-based sampling approach to balance the number of feature trajectories from both static and dynamic scene parts.

1.2.2/ KNOWN CAMERA MOTION CASE

The moving object detection problem can be simplified as a change detection problem after compensating the camera ego-motion. Still, it is a challenging problem due to the lack of knowledge, e.g. the size, the velocity, and the number of moving objects. Traditional image-based approaches focus on camera motion compensation using image homography registration by assuming that the scene is nearly planar. The dynamic scene parts are then detected based on pixels or patches differences. For non-planar scenes where parallax effects arise, the "Planes + Parallax" Decomposition-based approaches are proposed. However, such approaches work well only for slow camera motions in which consecutive images are largely overlapped. The probabilistic model-based methods are very popular when point cloud data are involved. For example, local occupancy grid maps are applied to record and predict the states of the occupancy grids. Such approaches highly rely on the prior knowledge of the map.

While detecting the dynamic 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 the current methods to fail because they either rely on image information which is sensitive to illumination changes or probabilistic models that require prior map knowledge. We therefore seek a robust algorithm which detects the moving objects solely relying on non-textured 3D point clouds. To this end, a novel Flow Field Analysis (FFA) approach is introduced to detect the motion flows of moving objects based on (but not limited to) 3D point clouds acquired from laser scanners. The FFA approach analyses the spatial and temporal displacement of objects, which addresses the motion detection problem in essence. The detected motion flows can be further clustered into their respective motions using the Sparse Flow Clustering algorithm.

1.2.3/ 3D MAP RECONSTRUCTION AND ENHANCEMENT

Once the moving objects are detected and segmented, a forward step is to obtain high quality 3D reconstruction. The 3D static map reconstruction is conducted by only registering the static scene parts, while the multi-body reconstruction of the rigidly moving objects is achieved in a similar manner. Iterative Closest Point (ICP) is one of the most commonly used algorithm due to its simplicity and robustness. However, the convergence

of ICP algorithm requires a careful initialization and rich geometric structures of the point clouds. To overcome these problems, we find out that initialization using the 3-point RANSAC registration algorithm is very effective. Further, a Dual-Weight ICP (DW-ICP) algorithm is employed to iteratively estimate the rigid transformation by assigning different weights to the RANSAC inlier point pairs and the ICP correspondences.

Due to noise, the registered point clouds from multiple observations suffer from multi-layered artefact which is addressed by employing a 3D Thin Plane Spline algorithm. Furthermore, a ball pivoting algorithm is applied to construct 3D meshes of the smoothed point clouds. The textures of meshes are mapped from the color images and then refined by using mutual information. Finally, thanks to the recent advances in deep learning, we now are able to obtain faithful semantic labels using image information. To semantically understand the static and the dynamic objects, we learn their semantic labels using image information and transfer those labels to the 3D point clouds by using a max-pooling strategy, which is a significant step towards scene understanding.

1.3/ CONTRIBUTIONS

Our contributions have been published in several articles [2, 3, 4, 5, 6, 7, 8]. Our works dealing with unknown camera motion cases have been published in [2, 3, 5], and the contributions are summarized as follows:

- **i** We propose a novel framework for motion segmentation using a 2D-3D camera system attached on a mobile platform. The proposed framework clusters the raw 3D feature trajectories using the Sparse Subspace Clustering (3D-SSC) algorithm and the SMooth Representation clustering (3D-SMR) approach, which outperforms the state-of-the-art motion segmentation methods.
- **ii** We present a simple but effective scheme for incomplete trajectory construction to handle the practical problem of feature tracking loss.
- **iii** We introduce an effective flow-likelihood model which samples the feature trajectories based on their optical flow values, and balances the number of trajectory samples from the static and dynamic scene parts.

The corresponding papers are listed below:

- *Cansen Jiang, Danda Pani Paudel, Yohan Fougerolle, David Fofi, and Cedric Demonceaux. Static map and dynamic object reconstruction in outdoor scenes using 3-d motion segmentation. IEEE Robotics and Automation Letters (RAL), 1(1):324–331, Jan. 2016 (Invited presentation at ICRA'16, Stockholm, Sweden* $\sim 35\%$ *acceptance rate),* paper link, video link;
- *Cansen Jiang, Danda Pani Paudel, Yohan Fougerolle, David Fofi, and Cedric Demonceaux. Reconstruction 3d de scènes dynamiques par segmentation au sens du mouvement. In Le 20èmè congrès national sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA). Clermont-Ferrand, France, Jun. 2016*, paper link;
- *Cansen Jiang, Danda Pani Paudel, Yohan Fougerolle, David Fofi, and Cedric Demonceaux. Incomplete 3d motion trajectory segmentation and 2d-to-3d label transfer for dynamic scene analysis. In IEEE International Conference on Intelligent Robot and System (IROS). Vancouver, Canada, Sept. 2017 (*$\sim 45\%$ *acceptance rate),* paper link, video link.

In cases of known camera motions, our contributions have been reported in [8], and are:

i We present a novel algorithm for moving object detection by using the 3D vector flow analysis. Our algorithm efficiently and accurately detects the motion flows via Radon transform, and outperforms the state-of-the-art methods.

ii We further propose a new Sparse Flow Clustering (SFC) model under the sparse subspace self-representation framework with improved performances due to the introduction of a spatial closeness constraint which significantly outperforms the state-of-the-art approaches.

The corresponding paper is noted below:

- *Cansen Jiang, Danda Pani Paudel, Yohan Fougerolle, David Fofi, and Cedric Demonceaux. Static and dynamic objects analysis as a 3d vector field. In IEEE International Conference on 3D Vision (3DV). Qingdao, China, Oct. 2017 (Oral presentation* $\sim 7\%$ *acceptance rate)*, paper link, video link.

Finally, our contributions in high quality 3D reconstruction of static map and multi-rigid bodies are detailed in [4, 6, 7], as follows:

i We propose a robust and accurate optimization framework for sparse point cloud registration. Our formulation is leveraged from the closest-point and consensus-

based methods, while complementing each other in their unfavourable conditions.

ii We introduce a dynamic scene understanding framework for simultaneous dynamic object extraction, static map reconstruction, and semantic labels assignment.

The corresponding papers are indexed below:

- *Cansen Jiang, Dennis Christie, Danda Pani Paudel, and Cedric Demonceaux. High quality reconstruction of dynamic objects using 2d-3d camera fusion. In IEEE International Conference on Image Processing (ICIP). Beijing, China, Sept. 2017 (*$\sim 45\%$ *acceptance rate)*, paper link, video link;
- *Cansen Jiang, Yohan Fougerolle, David Fofi, and Cedric Demonceaux. Dynamic 3d scene reconstruction and enhancement. In IAPR International Conference on Image Analysis and Processing (ICIAP), pages 469–479. Catania, Italy, Sept. 2017 (Oral presentation* $\sim 10\%$ *acceptance rate)*, paper link.
- *Dennis Christie, Cansen Jiang, Danda Pani Paudel, and Cedric Demonceaux. 3D reconstruction of dynamic vehicles using sparse 3D-laser-scanner and 2D image fusion. In IEEE International Conference on Informatics and Computing (ICIC), pages 61–65. Lombok, Indonesia, Oct. 2016*, paper link.

1.4/ ORGANIZATION

This thesis is divided into seven chapters. Chapter 2 comprehensively studies the related works on moving object detection and segmentation. In Chapter 3, we introduce the fundamental knowledge in Subspace Clustering approaches, Robust Estimation methods, and Convex Optimization algorithms. The proposed 3D-SSC and 3D-SMR algorithms for motion segmentation problem with unknown camera motion are presented in Chapter 4. When camera motion is known, a Flow Field Analysis-based approach is employed to detect the motion flows which are further clustered using the proposed Sparse Flow Clustering algorithm, as detailed in Chapter 5. Afterwards, a DW-ICP algorithm is suggested for accurate point clouds registration in Chapter 6. This chapter also provides a 2D-to-3D label transfer strategy for 3D scene labelling and understanding. Chapter 7 concludes our work and summarizes some future perspectives.

2

Literature Review

“Know how to solve every problem that has ever been solved."

- Richard Feynman, *The Feynman Lectures on Physics*

Moving object detection and segmentation has been a popular research field over the last few decades. Date back to 1975, Limb and Murphy [9] proposed to estimate the velocity of moving objects of images from television stream, which introduced the interesting problem of *Moving Object Detection* (MOD). Later, such techniques were intensively driven by the second-generation coding [10] which aims to detect, segment, and remove the moving object to achieve very low bit rate video streams. For decades, the MOD problem has evolved from simple scenarios, i.e. static camera in planar scene, to more complicated cases, i.e., moving camera in non-planar scenes. Meanwhile, the quality and functionality of cameras have been profoundly improved, i.e. from noisy, low-resolution 2D cameras to high quality 3D cameras. Despite the significant amount of static-camera-based approaches, this thesis focuses on the moving camera cases. In other words, only the algorithms that are applicable to moving camera setup are discussed.

In details, most of the methods are grouped into two main categories, namely 2D-based (or image-based) MOD and 3D-based MOD. The 2D-based MOD approaches are applicable to the detection of moving objects using 2D data (images), while the 3D-based methods are dedicated to the motion discovery using 3D data. Note that the 3D data are generally considered as 3D point clouds acquired from 3D sensors, such as 3D laser scanners, RGBD cameras, stereo cameras etc. Within the scope of 2D-based MOD, there are two main sub-categories as *Planar Scene* and *Non-Planar Scene*. According to Irani and Anandan [11], planar (or 2D) scenes contain the scenes in which depth variations

are negligible compared to the camera-to-object distance. On the contrary, non-planar (or 3D) scenes have significant depth variations. Note that due to the non-negligible depth variations, the changes of camera pose result in strong parallax effects [1]. Therefore, dedicated algorithms are required to effectively perform MOD on both the planar and the non-planar scenes. In terms of camera types, the 3D-based methods are classified into three sub-categories, namely stereo camera, RGBD camera and laser scanner. In this chapter, we briefly discuss the most representative approaches in terms of their key techniques, performances, strengths and limitations.

2.1/ 2D-BASED MOVING OBJECT DETECTION

Early moving object detection approaches [13, 14, 15, 16] relied on the maintenance of background models, i.e. the median background model [17]. These background models are learned and updated over a period. By subtracting the background model, the temporal changes are detected as moving objects. However, such approaches are limited to the MOD of static camera cases. In dealing with moving camera cases, more advanced algorithms are required for both planar scenes and non-planar scenes.

On the one hand, since the planar scenes rarely suffer from parallax effect, camera motions can be relatively easily compensated by applying the image homography [2] [18]. More concretely, given a set of feature correspondences between two images, the projective transformation matrix can be estimated to register the two images. Thus, algorithms developed for static cameras have been able to adapt to the moving camera scenarios [19, 20, 21].

On the other hand, with strong parallax effects, the motion compensation using a single projective transformation matrix is insufficient. In this regard, multi-layer homography approaches [22, 23, 24] were proposed to segment the scene into multiple planes and register them respectively. Besides, to independently analyse the parallax areas, parallax decomposition [11, 25, 26, 27] was introduced. More recently, the motion segmentation techniques [28, 29, 30] which do not require camera motion compensation are applied

1. Parallax is a displacement or difference in the apparent position of an object viewed along two different lines of sight, and is measured by the angle or semi-angle of inclination between those two lines [12].
2. The operations of 2D transformation, e.g. affine transformation and projective transformation, of images are generally called *Image Homography* in computer vision.

to detect and segment moving objects through raw feature trajectories clustering. Some other methods, e.g. low-rank minimization [31, 32], belief propagation [33, 34, 35], split-and-merge mechanism [36, 37], are also employed to handle the difficult problem of MOD in 3D scenes.

Being one of the most demanding problem in computer vision, the MOD problem received considerable attention over a long time. Although there are significant contributions in literature, the MOD is still an unsolved problem. The following sections discussed the most classical approaches with their strength and limitations.

2.1.1/ MOD Approaches for Planar Scene

We can consider three major types of planar scenes, namely the Pan-Tilt-Zoom (PTZ) images [38, 39, 40, 41], the Unmanned Aerial Vehicle (UAV) [42, 43, 44, 45, 46, 47] images, and the remotely shot images [48, 49, 24, 50]. For instance, images taken from a UAV are typical 2D scenes where the heights of ground objects are far smaller than the height of the UAV. Ideally, such planar scenes are free from parallax effects due to the small depth difference of the scene structures. Thus, by simply applying the image homography, the scene changes due to camera ego-motion can be eliminated. Afterwards, solving the MOD problem becomes finding the changed area (or the foreground area) of the images.

2.1.1.1/ Frame difference-based approaches

Traditional MOD approaches [19, 20, 51, 52, 53, 54, 46, 55, 56, 57] focus on the measurement of pixel-wise or block-wise inter-frame image difference. Then the thresholding techniques are employed to obtain a binary mask which defines the moving object region. In the consideration of outliers, different robust estimation schemes are employed to improve their performances. For examples, MOrphological Processing (MOP) [52, 46] yields closed and continuous object-regions segmentation. Statistical robust estimation schemes, e.g. M-EStimator (MES) [51] and RANdom SAmple Consensus (RANSAC) algorithm [55, 56], have been included to penalize the impacts of the outliers. Assuming planar backgrounds, Cast Shadow Detector (CSD) [14, 53] using a physics-based signal model has been proposed to reduce the false alarms due to moving shadows.

The frame difference-based approaches are very efficient and easy to implement. However, such methods are very sensitive to noise and fail to detect moving objects with homogeneous textures. Although some robust estimation techniques have been incorporated in their frameworks, these methods still require the scene to be specified and empirically predefined thresholds, making them difficult to be generalized. (e.g. a dynamic background [3]).

2.1.1.2/ PROBABILISTIC-BASED APPROACHES

In a more sophisticated manner, statistical approaches, e.g. Single Gaussian Model (SGM), Gaussian Mixture Model (GMM), and Maximum-A-Posterior (MAP) criterion, maximize the probability of each pixel in the foreground or the background. In this context, the SGM-based methods [58, 59, 60, 61] model the background pixel with a zero-mean Gaussian. Given a new observation, the Gaussian model predicts the probability of the pixel being a background pixel (or being a foreground pixel). Similarly, some thresholding techniques have also been employed to separate the background and the moving objects. Unfortunately, these SGM models are not able to handle periodically changing areas (i.e. the waving trees and the moving stairs) which should be defined as dynamic background rather than moving objects.

To address the dynamic background problem, the GMM-based approaches [62, 63, 64, 65, 26, 66, 67, 68, 69, 70, 71, 72] have been introduced to model each pixel with multiple Gaussians. Each Gaussian has its corresponding mean and covariance which represent the specific behaviour of the data distribution. For instance, for a waving leaf, three Gaussians can be used to model the measurement noise, the light change and the leaf's motion. Since the GMM (also the SGM) models the background areas, it is trivial to obtain the corresponding background model of the scene. Considering the image registration error, the Spatial Distribution Gaussian (SDG) model [61, 63, 73] has been used to optimize the pixel matches. On one side, the matching of background pixels after homography registration can be more precise. One the other side, the evolution of object appearance is learned for better prediction.

Generally, the Expectation Maximization (EM) [74] algorithm determines the best mixture

3. A dynamic background refers to the background containing periodic changicng areas, such as grass lands, waving tree leaves, and water surface etc.

model and updates the mixture model over a long term observation. The EM algorithm computes the expectation of the log-likelihood and then maximizes the expected log-likelihood, which essentially is under the framework of MAP. There are more MAP-based approaches [58, 21, 75, 76, 59, 47] which maximize the posterior distribution of the learned probability model using the Bayes' theorem [77]. In fact, such probability model learns the temporal changing behaviours of each pixel using the previous observations. Thus, such approaches requires a long term observations as initialization. When the camera is moving very fast, i.e. a camera mounted on a driving car, the EM-based methods are usually inappropriate due to the lack of data observations.

To overcome the difficulty of small data samples due to short term observation, the Kernel-based Density Estimation (KDE) [16, 78, 59, 79, 80] has been applied to find a density function which best fits the current data distribution. The KDE [4], which is more reliable when only a few data samples are available [81], aims to extrapolate the measured data into a regular density function (e.g. the Epanechnikov Probability Density Function [78, 82]). Unlike the parametric fitting of a mixture of Gaussians, the kernel density estimation is a more general approach that does not assume any specific shape for the density function. In fact, since the KDE does not assume any specific underlying distribution, it can converge to any density shape with sufficient samples. However, the KDE relatively requires more computation time.

Belief Propagation Algorithm (BPA) [26, 83, 84, 85, 86, 35] which is invented to calculate the marginals in Bayes nets is a marginal probability (so-called Belief) estimation mechanism for graph optimization. Given an image sequence, each image is considered as one layer where each pixel (or component) has a marginal probability describing its state. The BPA maximizes the likelihood by iteratively updating the marginal probability between the connected components. Since the BPA aims to optimize the marginal probability through the image sequence, it is generally considered as marginal-MAP. Similar to other MAP algorithms, the BPA algorithms rely on the long term observations, which makes such method not suitable for fast moving camera scenarios.

4. The KDE has two major components, namely the bandwidth function and the kernel function. A kernel function is a positive and unit-variance Probabilistic Density Function (PDF) centred at the sample point. The bandwidth is a function that describes the relationship of the sample point and its neighbourhood. The Gaussian is a widely used kernel function with its variance as the bandwidth.

2.1.1.3/ SPATIAL-TEMPORAL MECHANISM-BASED APPROACHES

When the camera is moving fast, the conventional probabilistic models (e.g. the Gaussian model), however, are not sufficient to solve the accumulated errors in image registration. To address this problem, the Spatial-Temporal Mechanism (STM) [87, 88, 89, 90, 91, 55, 92, 79, 49, 61] has been proposed. The STM constrains both the motion smoothness of the foreground object (the spatial domain) and the preservation of object appearance (the temporal domain). Building on top of the SGM or GMM model, the STM is imposed by adding spatial and temporal terms to the mean(s) and the (co)-variance [87, 88, 55, 49]. In other words, the Gaussian evolves according to the background or object motion as time passes. Thus, the accumulated registration errors due to camera motion are penalized.

Moreover, given the initially detected sparse and discontinued foreground object segments, the STM maintains the spatio-temporal object appearance. Based on the traditional object segmentation approaches (e.g. the Markov Random Field segmentation [79]), the STM contributes to the refinement of foreground object segmentation [89, 91, 55].

Taking the advantages of motion spatio-temporal consistency, the Detection And Tracking (DAT) approaches [25, 93, 94, 46, 95, 96, 71] have also been proposed. In the MOD stage, moving objects are detected using the traditional approaches such as inter-frame difference method [25, 93, 46], Markov Random Field [97], implicit shape model object detector [94], hierarchical image segmentation [96], motion flows [95], or background subtraction [71], etc. In the second stage, the detected objects are tracked through out the following images. Thanks to the prior knowledge of the moving objects from the history, the DAT approaches perform more robustly, especially when the camera itself moves smoothly.

2.1.1.4/ LOW-RANK REPRESENTATION-BASED APPROACHES

Low-Rank Representation (LRR) algorithms [98, 99, 70, 31, 100, 101] are inspired by the compressive sensing techniques. When the camera is stationary or moves slowly, the contents of the observed sequence are highly repeated and redundant. Thus, the concatenation of these data forms a low-rank matrix or tensor[5]. To analyse such a low-rank

5. The one-dimensional, two-dimensional, and three-dimensional data collections are called vector, matrix, and tensor, respectively.

matrix, Principal Component Analysis (PCA) [99, 31] techniques are employed to discover the principal components of the data. For instance, the combination of the principal components of an image sequence forms a background model. Specifically, LRR-based approaches are background modelling techniques based on self-learning dictionaries. Afterwards, the MOD is achieved by subtracting the background model.

The most classical LRR approach is the Robust PCA [99] which addresses the background modelling problem as a low-rank constrained optimization problem. The Robust PCA also models the sparse corrupted entries, making it robust to noise and outliers. Following this direction, a unified framework named DEtecting Contiguous Outliers in the LOw-rank Representation (DECOLOR) [31] was proposed as a ℓ_0-penalty regularized RPCA which is capable of modelling static and dynamic backgrounds from slowly moving cameras. On top of DECOLOR, the 3D Total Variation (3DTV) circulant sampling method [102] in compressive sensing is employed to detect and segment the foreground objects.

In a more sophisticated manner, a 3-Term Decomposition (3TD) algorithm [70] systematically models the decomposition of the background, the turbulence, and the moving objects as a low-rank optimization problem. In the 3TD algorithm, three types of norms, namely the nuclear-norm, the Frobenius norm and the ℓ_1, are adopted to model the background, the turbulence and the moving objects, respectively. Similarly, an Inexact Augmented Lagrange Multiplier (IALM) [100] is applied to decompose the video sequence into background, moving objects and camera-motion matrix between consecutive frames.

Note that the LRR-based approaches (RPCA, DECOLOR, 3TD, IALM, etc.) vectorize every frame as a single vector such that the image sequence is concatenated as a two-dimensional matrix. Unlike such methods, the Tensor-based Low-rank and Saliently Fused-Sparse Decomposition (TLSFSD) model [101] preserves the natural space-time structure of video sequences by representing them as tensors. The TLSFSD uses the tensor nuclear norm to exploit the spatio-temporal redundancy of background. In addition, a saliently-fused sparse regularizer is employed to adaptively constrain the foreground with spatio-temporal smoothness and geometric structure information. The TLSFSD is acclaimed to have state-of-the-art performance [101].

2.1.1.5/ LEARNING-BASED APPROACHES

There exist two major types of learning-based approaches [103, 104, 26, 105, 57, 106, 107, 108, 35], namely the STatistically Learning (STL) and the SEmantically Learning (SEL) approaches. The STL-based approaches, e.g. the GMM-based background modelling [109], learn the scene background through a spatial-temporal statistical analysis of the observed data. To overcome the weakness of background modelling of slow motion, the scene conditional background learning approach [35] is proposed with the awareness of slow motion.

Within the context of statistical learning, the two-layer Independent Component Analysis (ICA) model [26] learns the object motions based on the ICA bases encoding, as well as their joint distribution for co-activation analysis of the motion patterns. More straightforwardly, the Layered Motion Segmentation algorithm [105] is introduced by representing the scene as a composition of layers. The layer segments are combined to product the latent images for the representation of piecewise parametric motions. More recently, a sparse representation-based dictionary learning approach [57] is proposed assuming that the data are self-expressive. The sparse representation theory assumes that in any signal, there exists a sparse linear combination of atoms from a dictionary that approximates it well [110]. The dictionary can be learned from a training set and used to approximate the new observations. The MOD is performed by subtracting the approximated scene with the new observation. By nature, the sparse representation-based approach is very robust to noise and outliers.

Unlike the unsupervised learning approaches, the SEL approaches focus on the object-specific supervised learning for moving object detection, i.e. pedestrian detector using distance transform map [111], moving vehicle detection using template matching [112], road scene semantic information aided moving object segmentation [26], flying object detection by spatio-temporal cube representation-based AdaBoost classifier [106, 107], convolution network-based semantic motion signature for moving object segmentation [108].

In practice, both the STL-based and the SEL-based approaches require some kind of training dataset. The STL-based approaches are usually unsupervised learning techniques that adaptively learn and update a scene model from early observations. However, the

SEL-based approaches often rely on manually labelled training data making them difficult to generalize to unknown scenes.

2.1.2/ MOD Approaches for Non-Planar Scene

The non-planar scenes contain significant depth variation such that the parallax effect is non-negligible. For example, indoor environments and city street environments are typical non-planar scenes which contain objects located at different depth layers. Recall that image registration using homography transformation is valid to either planar scenes or pure-rotation camera motion. In other words, when the camera is freely moving, the ego-motion compensation cannot be fully addressed by applying a single homography transformation between successive frames. To tackle the MOD problem for images of non-planar scenes, three major types of approaches were proposed, namely the *Plane+Parallax Decomposition*, the *Trajectory Analysis*, and the *Motion Segmentation*.

2.1.2.1/ Plane+parallax decomposition

Since the non-planar scenes have significant depth variation, when the camera moves, such depth variation leads to the parallax motions. Early approaches [113, 114] suggested to decompose the scenes into multiple layers and fit multiple 2D planar surfaces. Although the scene registration using multi-layered homography partly addresses the non-planar scene problems, there remains ambiguity between the parallax motions and the object motions.

As discussed by Irani et al. [11], the effects of parallax are only due to the camera translation and the non-planar scene variations, unlike camera rotation or zoom. Therefore, a Planes + Parallax Decomposition (PPD) [11] breaks down the scenes into: the planes and the parallax. With the help of PPD, MOD of the decomposed planes is performed using the afore-discussed traditional algorithms. To identify the parallax motions, the parallax-based shape constraint and the parallax-based rigidity constraint are enforced to detect the moving objects from the parallax motions. The PPD-based methods [25, 115, 11, 26, 27] are general to different types of non-planar scenes. However, such methods require the knowledge of a consistent reference point or reference plane, which is not always pos-

sible. Also, the number of planes to fit is also difficult to specify.

2.1.2.2/ FEATURE TRAJECTORY ANALYSIS

The Feature Trajectory Analysis (FTA)-based approaches [116, 117, 118, 119, 120, 121] aim to detect and track the sparse features through multiple frames, and analyse those feature trajectories. Since feature detection for low-textured regions is relatively difficult, dense optical flow tracking [122] is usually employed. Getting feature trajectories classified as either background trajectory or motion trajectory, object segmentation algorithms, such as normalized graph-cut algorithm [123], are applied for dense moving object extraction.

Original FTA-based approach [116] focuses on the clustering of feature trajectories based on their motion velocities. The scene is segmented into blocks where the feature trajectories having similar motion velocities are clustered together. Under the affine motion assumption, the dominant motion cluster is considered as the background motion cluster, while other clusters are considered as foreground objects. Such approach requires very precise feature tracking and motion estimation and not robust to noise.

A Robust SIFT Trajectories (RST) analysis approach [120] has been proposed to detect the major foreground object of the scene. Based on the RST, a consensus foreground object template is generated and updated during the tracking of the moving object. Such methods is efficient and robust, but restricted to single moving object detection. Inspired by the RST approach, the Matched Regional Adjacency Graph (MRAG) algorithm [121] groups the super-pixel trajectories. The MRAG construction relies on regions' visual appearance and geometric properties with a multi-graph matching scheme. The MRAG approach is able to detect and track multiple moving objects, yet it is not able to handle occlusions.

Inspired by the motion boundary detector [124] which seeks motion discontinuities by detecting edges where motion cues aggregated from adjacent regions change abruptly, an Embedding Discontinuity Detector (EDD) [118] was proposed. The EDD localizes the object boundaries by detecting density discontinuities in a trajectory spectral embedding. By nature, such method is incapable to detect small moving objects.

Background Trajectory Subspace (BTS) analysis approaches [117, 119] have been pro-

posed to model the subspace of the background feature trajectories with certain rank constraints [6]. The bases of BTS are defined by a combination of 3 feature trajectories under the RANSAC framework. All trajectories which are the projections of stationary points (assuming the background is static) must lie in the BTS. However, if the BTS contains motion feature trajectories, the rank of BTS will be higher than 3. Thus, any feature trajectory which leads to a higher rank of BTS belongs to the moving objects.

2.1.2.3/ OPTICAL FLOW-BASED APPROACHES

Optical Flow (OF)-based approaches [126, 127, 128, 129, 130, 53, 131, 132, 133, 134, 135, 136, 137, 138, 37, 117, 139, 140, 95, 141] have been abundantly developed in literature. The OF was inspired by the physical 3D-to-2D projection model under the assumption of brightness consistency, and is resulted from the relative motion between the camera and the objects. Thus, different motions generally result in different OFs in terms of flow directions and amplitudes. The OF of each pixel can be estimated using Horn and Schunck's method [142], Lucas-Kanade method [143], or their variations [144]. There are three major usages of OF in MOD: pixel-level motion estimation, pixel-to-pixel matching, and spatial-temporal feature tracking.

Straightforwardly, pixel-wise or patch-wise Displaced Frame Difference (DFD) [139, 37] approaches have been proposed by taking the advantages of OF-guided pixel matching. The dense pixel (or feature) matching are established by compensating its OF motion between two consecutive frames. Afterwards, inter-frame difference-based approaches are applied to detect the changing pixels which are further grouped by using object based segmentation techniques for moving object extraction.

Since the OF encodes the relative motion between the object and the camera, it is natural to discover the moving objects by detecting the OFs which violate the physical model (such as the Motion Epipolar Constraint [126]) of the OF. Therefore, the Flow Violation Policy (FVP) [126, 132, 95] has been proposed to detect moving objects by comparing the true OFs with the estimate OFs. The true OFs (or the artificial flows) are generated based on the object's 3D motion model by priorly estimating the camera motion and the camera calibration parameters. Then, the estimated OFs are compared with the true

6. Under the orthographic projection assumption, the trajectory matrix, which is constructed by concatenating the vectorized feature trajectories (as detailed in Chapter 3.3.3), is a rank 3 matrix [125].

ones. Hereafter, the estimated OFs, which are strongly different from the supposedly true flows, are considered as coming from moving objects. However, such FVP-based approaches require precise camera ego-motion estimation and are sensitive to the scene depth variances.

Focus of Expansion (FOE)-based approaches [126, 133, 136] have been developed based on the FVP. The FOE is a point where all the OF of static objects meet at. In fact, OFs of each independent rigid motion meets at a distinctive point. Thus, any OF vector not passing through the FOE belongs to moving objects. The FOE-based approaches can detect very small moving objects with fast moving cameras, however, their performances highly rely on the precise estimation of OFs and the FOE.

Object Contour Tracking (OCT) approaches [126, 128, 129, 130, 53, 132, 134, 135, 138, 140, 141] using optical flow field segmentation have been widely used in literature for both planar and non-planar scenes. Since distinctive moving objects produce distinctive motion fields, the discontinuity of the optical flow field indeed corresponds to the contours of the moving objects. To obtain the complete segmentation, Level-set segmentations [145, 137, 146], Piecewise-smooth Flow Field segmentation [147, 148], hierarchical motion field segmentation [149] or Markov Random Field [128, 130, 140, 135] are applied after getting the initial object contours from the optical flow field. Afterwards, the object contours are tracked and updated along the motion of the moving objects. The OCT-based methods are very promising for simple scenes with few moving objects. However, when there are a lot of moving objects, or when the tracks of moving objects are intersecting, such methods usually produce many false alarms and even fail.

2.1.2.4/ EPIPOLAR CONSTRAINT-BASED APPROACHES

Epipolar Constraint-based methods [150, 151, 152, 153, 154, 155, 27, 156] belong to another major branch in MOD based on the camera geometry. The *Epipole* is the point of intersection of the line joining the optical centres (also called the baseline) with the image plane [18]. The epipole in one image is the mapping of the camera center of another image. Given a 3D point with two epipoles corresponding to two different cameras, the plane going through these three points intersects the two images at two lines – so-called the *epipolar plane* and the *epipolar lines*. Since all epipolar planes intersect both camera

centres, all epipolar lines will intersect at the epipoles. Therefore, for image points corresponding to the same 3D point, these image points, 3D point and optical centres are coplanar, which is called the *Epipolar Constraint*.

Accordingly, when a static scene is observed by moving cameras, the two epipolar lines by passing their respective matching points and epipoles are coplanar. For any epipolar lines of matching pairs that violate this condition, these matching pairs are originated from a moving object. In other words, when the matching pairs come from the moving objects, their respective epipolar lines will not lie on the same plane. Practically, feature matches are very often noisy, MOD using epipolar constraint can be unstable. Thus, thresholding techniques and other robust estimation schemes are also incorporated, such as Probabilistic model [150, 156], Trifocal Tensors [153], Multi-frame Affine Motion Constraint [151], Spatial-temporal Mechanism [152], Space-time Invariant Condition [154], Semantic Information [155], Parallax Decomposition [27], etc. Although the epipolar constraint-based methods can efficiently perform MOD for both planar and non-planar scenes, such methods are very sensitive to image noise.

2.1.2.5/ ENERGY MINIMIZATION-BASED APPROACHES

ENergy Minimization (ENM)-based approaches [129, 52, 91, 157, 80, 118, 72, 158] are also widely used, because the MOD problem is a pixel labelling problem which can be represented in terms of ENM. The ENM function usually has two terms: one data-driven energy term that penalizes the solutions that are inconsistent with the observed data, and one regularization term that enforces spatial and temporal coherences.

ENM-based approaches are usually applied to problems such as camera-ego motion estimation [52, 157], contour extraction of moving objects [129, 148], pixel-wise matching pairs searching [118], spatial-temporal directional energy minimization for moving object extraction [159, 91], MRF-based energy minimization for moving object segmentation [80, 160], total-variational energy minimization for moving object segmentation [72, 158]. The ENM-based approaches are usually robust to noise and produce satisfactory object extraction. However, such methods involve a time-consuming iterative refinement process and require sophisticated parameter tuning.

2.1.2.6/ TWO-FRAME MOTION SEGMENTATION

One of the most popular motion segmentation approach relies on FTA. The MS methods aim to cluster the feature trajectories into their corresponding motions. In other words, moving object detection and segmentation are simultaneously achieved. This section introduces some of the most representative MS approaches by categorizing them into: two-frame MS and multi-frame MS.

Two-frame MS approaches [147, 149, 161, 145, 137, 162, 146] mainly focus on the joint optical flow field estimation and the flow field segmentation. For example, a simultaneous motion estimation and segmentation approach [147] is proposed by using the MAP-based Gibbs Distribution Potential (GDP) function. Such GDP function jointly minimizes the displaced frame difference, the motion field residual and maximizes the priori probability of the segmentation. For the same purpose, a Dense Discontinuity Preserving (DDP) [149] motion estimation technique is introduced under a hierarchical constrained optimization framework which jointly recovers the dense estimation as well as a parametric representation of the motion field via a half-quadratic formulation of robust cost functions. Besides, variational approaches [145, 137, 161, 162, 146] consist of a data term (describing both the brightness and gradient constancy) and a regularization term for spatial-temporal smoothness. Eventually, a level-set segmentation is applied to densely segment the moving objects.

Similar to the FVP-based approaches, the Unique Epipolar Constraint (UEC) approach [163] has been proposed. The UEC approach defines a total cost energy function containing a data term measuring the fitness of fundamental matrix and a discontinuity penalty term enforcing the spatial smoothness. Afterwards, a region growing algorithm [164] is employed to segment the independently moving objects. As a drawback, the affine motion assumption requires the epipoles to be located at infinity, which is not always true.

These two-frame MS approaches usually rely on the accurate computation of optical flow. However, such optical flow estimation is difficult and imprecise in practical environments, especially for low textured objects. Some two-frame MS approaches depend on the sparse feature correspondences. For instance, Two Perspective View (TPV) approaches [165, 166] exploit the algebraic and geometric properties of the multi-body

epipolar constraint and its associated multi-body fundamental matrix for object motion estimation and segmentation. A rank constraint on a polynomial embedding of the correspondences is derived, which benefits to the independent motion estimation and the multi-body fundamental matrix. The feature points are then clustered using either the epipoles, the epipolar lines, or the individual fundamental matrices. As mentioned by the authors, such methods are mainly designed for noise-free correspondences. In a similar manner, the TPV epipolar constraint is used to formulate the sparse self-expressive subspace segmentation problem [167], which is inspired by the SSC [29]. Then a collaborative clustering step alongside with a mixed-norm optimization scheme is employed. Different from the affine motion assumption in [163, 29], the TPV-based approaches directly work on the perspective camera model. Following this direction, a Branch-aNd-Bound (BNB) [168, 169] combinatorial optimization technique was incorporated to solve the chicken-and-egg dilemma –estimating and fitting unknown number of fundamental matrix for unknown motions. More recently, a Randomized Voting Scheme (RVS) [170] was proposed for rigidly moving object segmentation. However, in the presence of outliers, such approaches are not reliable.

A bottom-up frame-to-frame motion segmentation is proposed by using a multi-scale Motion Split-And-Merge (MSAM) [171, 172] clustering on the SIFT key-point matches. The MSAM initially splits the key-points into consistent segments using the J-Linkage [173], then the neighbouring segments are merged until converged. Remarkably, instead of feature tracking, the key-points are detected and matched across frames, which helps the BFF motion segmentation with significant missing data.

In an algebraic manner, a Hybrid Quadratic Surface Analysis (HQSA) [174] was proposed by casting the general MS problem of segmenting data samples drawn from a mixture of linear subspaces and quadratic surfaces. The proposed HQSA used both the derivatives and Hessians of fitting polynomials to separate the linear data samples from the quadratic data samples. Since the HQSA makes no affine motion assumption, the natural perspective motion cases can be elegantly solved.

Few other methods require a system initialization which is considered as the prior knowledge to assist the MS in new observations. For example, the MS is formulated as a manifold separation problem with a Dynamic Label Propagation (DLP) mechanism [175]. Such approach relies on a fixed number of frames for initialization, and the label history predicts

the new observation using a dynamically changing graph. Similarly, Sparse Background Model (SBM) [176] is constructed using the training data. Afterwards, the MAP framework is employed to maximize the probabilities of both foreground and background labels taking the prior knowledge of the SBM.

At last, the two-frame MS approaches incur that the motion between two frames are relatively small, which often yields to ambiguous segmentations. Especially when fast moving camera meets slowly moving objects, the two-frame MS approaches are usually not recommended.

2.1.2.7/ MULTI-FRAME MOTION SEGMENTATION

Multi-frame MS approaches are more popular than the two-frame-based methods due to the fact that longer observations provide more reliable information. Thus, in general, the multi-frame MS approaches are usually more robust. Most of the recent studies follow a standard procedure as feature trajectory construction (e.g. dense optical flow tracking [122]), affinity matrix construction (e.g. sparse subspace representation [177]), and spectral clustering (e.g. k-means spectral clustering [178]). In the following contents, we roughly classify the multi-frame MS literature into several categories: *Matrix Factorization techniques, Algebraic methods, High-Order Clustering, Subspace Self-Representation methods, and other approaches*.

Matrix Factorization Techniques (MFT) [179, 180, 181, 182] initially aim to factorize the feature track matrix, in which each column corresponds to one feature trajectory, into multiple segments under low-rank constraints. During the MFT, the Singular Value Decomposition (SVD) [183] is applied to the track matrix. Because the number of non-zero singular value corresponds to the number of motions, the singular vectors with respect to the non-zero singular values are the bases of the different motions. A split-and-merge scheme is employed to cluster the tracks based on the singular vectors [179]. Differently, a Shape Interaction Matrix (SIM) [180, 181, 182] is built using the decomposed singular vectors. The SIM is permuted into a block-diagonalized matrix where each sub-block represents a motion cluster. Although being sensitive to noise, the groundbreaking work of SIM inspired many other methods. For example, a robust Space Separation and Model Selection (SSMS) algorithm [184] is proposed by incorporating techniques as dimension

correction, model selection using the geometric Akaike Information Criterion (AIC) [185], and least-median fitting. Different from SIM, the SSMS algorithm directly analyses the raw data rather than the shape matrix derived from them. As a follow up, a more robust Affine Space Separation (ASS) [186] is proposed due to the observations that, with weak perspective effects, the projective subspace clustering problem can be more effectively solved as an ASS problem. More recently, a Robust Shape Interaction Matrix (RSIM) [187] is proposed to handle corrupted and incomplete feature trajectories.

Also inspired by the SIM algorithm, the Orthogonal Subspace Decomposition (OSD) [188] approach decomposes the object shape spaces into signal subspaces and noise subspaces. Instead of using the SIM contaminated by noise, the orthogonal shape signal subspace distance matrix produces more robust shape space grouping. Some other methods are: Discriminant Criterion [189, 190] for feature similarity analysis is exploited for robust motion segmentation. Degeneracies and Dependencies Implications [191] approach segments articulate object motions. Although these methods achieved various improvements, they fail when the subspaces intersect arbitrarily [28].

Inspired by [192], the Non-Negative Matrix Factorization (NNMF) [193] algorithm decomposes velocity profiles of feature trajectories into different motion components with respective non-negative weights. The NNMF then segments the partial track data using weighted spectral clustering. In a similar manner, the Semi-Nonnegative Matrix Factorization (SNMF) [194] approach models optical flow velocities of the dense point tracks which are grouped into semantically meaningful motion components.

Algebraic Methods [195, 196, 28, 197, 198] do not need initialization and are very popular. Among them, the Generalized Principal Component Analysis (GPCA) [195, 196, 28] is the most representative approach that offers an algebro-geometric solution to the MS problem with no knowledge of number of subspaces and their dimensions. The GPCA represents the subspaces with a set of homogeneous polynomials whose degree is the number of subspaces and whose derivatives at a data point give normal vectors to the subspace passing through the point. Applying PCA to the normal vectors, the basis for the complement of each subspace is then recovered. As claimed by the authors, the GPCA also provides a robust initialization to iterative techniques such as K-subspaces [199] or EM [200] algorithms. However, the determination of number of clusters and their dimensions only works for noise free data.

As an improvement, the Robust Generalized Principal Component Analysis (RGPCA) algorithms [197] integrate 3 major techniques, namely the RANSAC algorithm, the influence function, and the MultiVariate Trimming (MVT). Although these robust estimation schemes contribute to the RGPCA be more robust against noise, such methods do not provide a convenient estimate of the outlier percentage and can not be easily scaled when the subspace dimensions are high. More recently, a Robust Algebraic Segmentation (RAS) [198] uses a hybrid perspective constraint to unify the representation of rigid body and planar motions. The RAS is an algebraic process that partitions the image correspondences, which can be determined by a set of $2K^{\text{th}}$-degree polynomials, into K individual 3D motions. By incorporating robust statistics, the polynomials can be estimated regardless of moderate image noise and outliers.

High-Order Clustering problems [201, 202, 203, 204, 205, 206] arise when data is drawn from multiple subspaces or when observations fit a higher-order parametric model. To address these scenarios, a Local Subspace Affinity (LSA) approach [201] is based on the geometric and the locality constraints of feature trajectory. Both the geometric constraint and the locality (after mapping to a unit sphere space) constraint show that the trajectories of the same motion lie in a low dimensional linear manifold. Thus, the MS problem can be cast as finding those linear manifolds, which derives the affinity matrix for spectral clustering. However, in the presence of noise, the local subspace fitting can be unreliable, especially for points lying near the subspace intersections.

Inspired by robust statistical model fitting, an Ordered Residual Kernel (ORK)-based approach [206] is proposed. The ORK elicits the potential of two point trajectories to have emerged from the same subspace. Random samples of trajectories are fitted with initial subspace hypotheses. The ORK (e.g. the Mercer kernel [207]) is then employed to model the subspace fitting residuals for subspace segmentation. The ORK-based method performs well under severe outliers arising from spurious trajectories or mistracks.

In a more natural manner, TeNsor Decomposition (TND)-based approaches [208, 202] have been proposed due to the fact that similarity measurement of an n-tuple of data points leads to a multi-way similarity tensor. The TND approach seeks a two-dimensional affinity matrix from its high-order tensorial representation of the data point tuples. Thus, the affinity matrix can be built by sampling columns from the flattened form of the similarity tensor. Eventually, spectral clustering is applied to the constructed affinity matrix.

Building on top of TND-based multi-way spectral, a Spectral Curvature Clustering (SCC) algorithm [203] presents an iterative sampling technique that significantly improves the performances of TND-based methods.

More recently, the Sparse Grassmann Clustering (SGC) [205] combines both the high-order similarity tensor decomposition and the low-rank matrix representation. In the high-order similarity tensor decomposition, the SGC clusters the data by directly finding a low dimensional representation without explicitly building a similarity matrix. By exploiting the online estimation of Grassmann manifold via gradient descent, the SGC is based on individual columns of similarities and partial observations, making it very efficient and scalable. Similarly, a Hyper-Graphs Clustering (HGC) [204] has been proposed based on the concept of Random Cluster Models for residuals fitting. The HGC relies on much larger samples for subspace fitting, which yields to large hyperedges of hypergraphs. To efficiently solve the large hyperedges problem, a guided sampling strategy was imposed for effective sampling.

High-order model-based methods are capable of solving general model fitting and clustering problems. However, model selection usually requires a priori. Moreover, such methods are often computationally expensive.

Statistical Approaches have also been proposed in literature. For example, a MAP probabilistic framework [209] is adopted to maintain the spatial consistency and smoothness. In this framework, the set of multi-view correspondences are modelled by an irregular Markov Random Field which encodes the relationships between the trajectories. Eventually, the similarity graph is segmented by a graph-cut algorithm. Similarly, a Minimum Cost Multicuts (MCM) algorithm [210] is proposed. The cost of MCM is defined by edge weights computed between synchronous and asynchronous trajectories. In a similar manner, the MS is achieved by applying a graph-cut algorithm.

In addition, EM-based algorithms [211, 212, 213] are proposed to segment multivariate mixed data with different strategies, such as Probabilistic PCA [212], k-Planes Clustering [214] and Lossy Data Coding and Compression [213]. There are two major steps in these algorithms: initial cluster assignment by fitting subspace or hyper plane to the clusters; and then iteratively update of the subspace fitting. In fact, such methods are very sensitive to the initialization and their performances are not guaranteed.

Subspace Self-Representation (SSR) [177, 29] property was discovered in the studies of compressive sensing [215] and widely used for motion segmentation recently. The pioneering works by Elhamifar and Vidal [177, 216, 29] utilizes the Sparse Subspace Clustering (SSC) for robust motion segmentation. We assume that the feature trajectories can be represented by other feature trajectories from the same motion subspace. By incorporating the sparsity constraint based on a relaxed ℓ_1 optimization, the SSC is very robust to outliers and achieves significantly better performances. However, the original SSC is proportional to the cubic of the problem size such that it is computationally expensive for large scale data. Accordingly, the Scalable Sparse Subspace Clustering (SSSC) [217] algorithm adopts the "sampling, clustering, coding, and classifying" strategy. The SSSC samples a small set of data and performs the SSC to obtain the sample clusters. The overall cluster assignment is then achieved by classifying the non-sampled data. Essentially, the SSSC is a extension of SSC with scalability.

Inspired by the SSC, numerous methods have been proposed. Around the core idea of SSR, a Subspace Segmentation via Quadratic Programming (SSQP) [218] algorithm, which seeks to express each datum as a linear combination of other data, is proposed. The SSQP employs a regularizer to the constraint of zero connection between different subspaces, such that a block-diagonalized affinity matrix is obtained after spectral clustering.

The Low Rank Representation (LRR) [219, 220] targets the lowest-rank representation of a collection of data. Ideally, the LRR also produces block-diagonal affinities, while a few inter-subspace connections exist in the presence of noise. In fact, the LRR better captures the global structure of data and is more robust to outliers. Note that fundamentally speaking, the LRR and SSC are both convex optimizations exploiting the intuition of "Self-Expressiveness". Although the SSC employs the ℓ_1-norm optimization while the LRR adopts the nuclear norm (denoted as ℓ_*-norm), both methods produce sparse solutions. By combining the SSC and the LRR, a general framework [221, 222] has also been proposed for subspace estimation and clustering in the presence of noise and outliers.

Following this direction, the Latent Low-Rank Representation (LatLRR) [223] algorithm is proposed to construct the dictionary by using both observed and unobserved data (latent data), where the data can be understood as a low-rank constrained SIM. The LatLRR algorithm integrates both subspace segmentation and feature extraction into a unified

framework. In addition, computational efficiency of LatLRR is gained by positive semi-definite programming.

By taking into account the latent subspace, the Latent Space SSC (LS3C) [224] is proposed for simultaneous dimensionality reduction and clustering of data which lie in a union of subspaces. The LS3C learns the projection of data and finds the sparse coefficients in the low-dimensional latent space. The SSC measures the smallest principal angle between data, which is sensitive to noise when the smallest angles are small. To address this problem, the LS3C projects the data to a Hilbert space, followed by an efficient linear and non-linear optimization based on the positive semi-definite Gram matrix. In addition, their extended work [225] generalizes the kernel selection problem. To this end, a normalized spectral clustering can be applied for final trajectories clustering. Similarly, a Low-Rank Kernel Subspace Clustering (LR-KSC) algorithm is proposed by integrating both the non-linear mapping of Hilbert space and the self-expressiveness of the subspace.

In consideration of spatial distribution of feature trajectories, a Weighted Sparse Subspace Cluster (W-SSC) [226] algorithm was proposed under the spatial closeness constraint, under the assumption that feature trajectories from the same moving object are close to each other. Such assumption yields a weight matrix encoding the spatial distance between trajectories. The W-SSC improves the sparse representations by an element-wise product with the weight matrix under low-rank constraint, making it more robust to noise and outliers. Similarly, a Least Squares Regression (LSR)[227, 228] approach is raised by introducing the grouping effect (GE) for subspace segmentation, where the GE tends to group highly correlated data together. The LSR leads to a block-diagonal matrix and a direct optimal solution can be obtained. The LSR experimentally shows the effectiveness of the GE constraints, while the theoretical proof can be found in [30]. The SMooth Representation clustering (SMR) [30] explicitly enforces the GE on the data self-representation model, thus leading to a GE-based affinity matrix construction. Similarly, the objective function of SMR is smooth and convex, so that it can be solved efficiently. Moreover, since the SMR relies on a derivable objective function, it can be easily scaled and fitted to large problems with dense feature trajectory segmentation.

Mixture of Gaussian Regression (MGR) [229] approach has been proposed by combining the SSR property, the GE, and the GMM model. The MGR firstly learns the self-representation matrix with a MGR which is able to handle various type of noises, followed

by a spectral clustering with GE constraint applied to cluster the affinity matrix constructed from the self-representation matrix.

In most cases, when the objective function is smooth and convex, the sought optimal solution is dense. In other words, such solution is more sensitive to noise. To overcome such problem, an Efficient Dense Subspace Clustering (EDSC) [230] approach seeks a clean block-diagonal and dense affinity matrix. Recall that SSC and LRR produce sparse block-diagonal affinity matrix. The EDSC pursues dense block-diagonal affinities with efficient solution by relaxing the objective function using the Augmented Lagrange Multiplier (ALM).

In spectral clustering, a block-diagonal structure of affinity matrix is preferred. A Block-Diagonal constrained SSC (BD-SSC) [231] and Block-Diagonal constrained LRR)(BD-LRR) [231] were proposed. The block-diagonality is imposed by a k-block-diagonal Laplacian Matrix [231]. It is shown that both the BD-SSC and BD-LRR are relatively more robust than their original versions.

Some other interesting ideas are also introduced in literature. For instances, the Ordered Subspace Clustering (OSC) [232] takes into account the sequential occurring information of subspaces. The OSC segments the data drawn from a sequentially ordered union of subspaces. Similar to SSC, the OSC relies on a sparse representation with an additional penalty term on the sequential data. Besides, as inspired by the sparse and low-rank decomposition-based feature correspondences [233], a simultaneous motion segmentation and feature correspondences have been achieved. For this purpose, the Partial Permutation Matrices (PPMs) [234] aim to match feature descriptors while simultaneously encouraging point trajectories to satisfy subspace constraints. Moreover, a STructured Sparse Subspace Clustering (STSSC) [235] is proposed as a unified optimization framework for learning both the affinity and the segmentation simultaneously. The STSSC employs a special subspace structured norm which ensures consistency between the representation coefficients and the subspace segmentation. A Tree-Structured Coding (TSC) [236, 237] is proposed to hierarchically cluster the feature trajectories according to non-rigid motion components.

The subspace self-representation-based approaches, especially the SSC, LRR and their variations, are dominant approaches in the state-of-the-art. Thanks to the advanced robust estimation schemes, these methods achieve excellent performance.

Other methods also attempt to address the MS problem. For instance, dealing with the common problems of Geometric Structure Degeneracy [104] in video motion segmentation, a multi-stage unsupervised learning scheme is proposed by firstly fitting a degenerate motion model, and then fitting the general 3D motion model. To exploit the feature trajectory properties, Illumination Subspace Clustering (ISC) [238, 239] approaches consider that the changes of feature intensity can be locally approximated as a linear subspace corresponding to the feature motion. Besides, motion segmentation approaches based on velocity variance [240], or trajectory length difference [241, 242], or using higher order tuples of trajectories [243], are proposed. The ideas of these algorithms are interesting, yet they are still not as accurate and robust as the SSR-based approaches.

2.2/ 3D-BASED MOVING OBJECT DETECTION

Apart from the comprehensive studies on motion segmentation using 2D data, there are also many approaches in literature rely on 3D data. In this section, we consider that the 3D data can be obtained from different ways, such as Structure-from-Motion (SfM) [244], Stereo Vision [245], RGB-D Sensors [246], or Laser Scanners [247]. In detail, the SfM-based approaches produce sparse or semi-dense 3D point cloud with least accuracy. The stereo vision and RGBD cameras produce dense and moderate accuracy. Lastly, the 3D laser scanners usually have highest accuracy but provide sparse point clouds with large field of view. Due to the large variety in density and accuracy of data, the 3D-based approaches usually rely on the properties of the sensors. Thus, we review and categorize the literature methodologies according to their data acquisition setups.

2.2.1/ STRUCTURE FROM MOTION

Structure-from-Motion-based approaches are the oldest 3D-based MS techniques that usually work with sparse feature for motion segmentation. When the camera motion is small and the scene is mostly static, the dense 3D reconstruction can be obtained. For example, by considering a small camera motion which allows dense inter-frame correspondences, a Concurrent 3D motion segmentation (C3D) [248] has been proposed to integrate the depth information. The C3D is a variational approach which relies on image sequence which consists of dense recovery of 3D structure and motion. Under rigid

motion assumption, the C3D performs the motion segmentation based on level-set for contour curve evolution, depth by gradient descent, and least squares motion estimation within each region of segmentation.

Similarly, a two-stage dense SfM [33, 249] approach has been proposed. This method first detects and reconstructs sparse features for camera ego-motion estimation. Afterwards, the dense feature matches are established by solving an energy function similar to dense optical flow estimation [250]. Under the appearance and structure consistency assumption, the bilayer segmentation jointly optimizes the label assignment of background and foreground for dense and high quality motion segmentation.

In the context of Simultaneous Localization and Mapping (SLAM), a Mono camera-based approach, so-called Mono-SLAM [251], was proposed for real-time 3D reconstruction. A feature "visibility" is defined by the relative position of the camera, the feature, and the saved position of the camera from which the feature was initialized [251]. Whenever the feature visibility is lower than 50%, it is considered as an outlier. In fact, the Mono-SLAM predicts outliers rather than precisely detecting the moving objects. Similarly, a Collaborative SLAM (Co-SLAM) algorithm [252] is proposed for highly dynamic environment 3D reconstruction. The Co-SLAM uses the inter-camera mapping with a sophisticated point state classification. However, such approach is very sensitive to feature matching accuracy.

2.2.2/ STEREO VISION CAMERA

Stereo vision cameras are widely used in the assistance of MOD due to the richness of depth information. Recall that the 2D-based approaches [150] address the MOD problem by camera ego-motion compensation plus geometric constraints, such idea can also be extended to 3D scenarios. A GLobal Motion model (GLM) [253, 254] precisely recovers the camera motion in 3D space, then geometric and motion information are employed to discover the moving obstacles that violate the GLM model. The GLM still suffers the drawback of being sensitive to noise like its 2D counterpart.

Among the stereo-based approaches, the Object Scene Flow (OSF)-based approaches [255, 256, 257, 258] are the most classical due to their robust performances. The OSF aims to estimate the object motion between consecutive frames using the 3D

motion information. The OSF first estimates and compensates the camera motion, then the object displacement is estimated by its inter-frame 3D point cloud registration. The multi-frame OSF [259] is proposed to concurrently optimize both the optical flow estimation and the piece-wise object segmentation. Inspired by the OSF, a semantic information assisted Scene Flow Propagation (SFP) approach [260] is proposed for more reliable detection. The SFP approach focuses on the potentially moving objects (e.g. people or cars) with a recursive Bayesian probabilistic framework under the spatial-temporal consistency assumption. Although the OSF methods are very promising, they are still unreliable for light changing environments.

Spatial-Temporal Displacement (STD) [261] approach has been proposed by considering that features are tracked across multiple frames with associated depth information. Initially, the sparse features are detected and tracked with Kalman filtering techniques [262]. Then, the STD algorithm measures the displacement of features to infer the moving objects. In a more robust manner, a Object Detection-by-Tracking (ODT) scheme [263] is integrated. Instead of tracking the features, the objects such as pedestrians or cars are detected prior to object-based tracking. By precisely tracking the moving objects and estimating their centroid trajectory, MOD becomes relatively easy. Due to the robustness of object-based tracking against feature-based tracking, the ODT achieves very convincing performances. However, such approach relies on precise object detector and is not robust against occlusions.

By exploiting the color information, a Height-Color-Histogram (HCH) [264] has been proposed by integrating the local convex-concave shape and the color information of the surface for feature correspondences establishment. Getting the semi-dense feature correspondences, the 3D flow field can be estimated, thus the motion segmentation can be achieved using the flow field. However, this 3D flow-based motion segmentation approaches requires precise depth estimation, making such methods quite unreliable for practical applications especially when the camera is moving fast.

2.2.3/ RGB-D SENSOR

Many of the 2D-based MOD approaches can be extended to the RGB-D case. For examples, as inspired by the statistical model-based approaches [62, 63, 64] in 2D, a

real-time dense 3D motion segmentation approach [265] is proposed based on the EM algorithm. A RGB-D flow approach [266] extends the 2D-based object flow for moving object extraction. Graph-based approaches [267, 268] follows the image-based graph matching techniques [243, 242] for MOD. To improve the computational efficiency, the proposed method approximately infers the image labels and motion estimates via a variational mean-field inference and graph-cuts. By extending the super-pixel in 2D to super-voxel in 3D, a VOxel Segmentation (VOS) [269]-based approach is proposed. The VOS segments the 3D objects and tracks them with particle filter. Similarly, an object model learning approach [270] is proposed to fully reconstruct and track the objects. However, these methods are constrained to simple indoor environments. Inspired by the background modelling techniques, an Active Machine Learning (AML) [271] algorithm is proposed to automatically learn the background model, followed by a background subtraction for MOD. The AML also iteratively learns and updates the background model during the observation. As a common disadvantage, their performances highly rely on a reliable initialization. Also, these methods are not appropriate for fast camera motions.

Intuitively, benefiting from the depth discontinuity, the object segmentation becomes relatively easier and more precise for low texture scenes. Accordingly, a Depth Guided Segmentation (DGS) [272] approach is proposed. To segment the moving objects, the DGS measures the depth changes between consecutive frames. Similarly, a Time-of-Flight camera-based foreground object segmentation approach, so-called TofCut [273], was proposed by jointly maximizing the foreground pixel likelihood on both the color and the depth information using an adaptive weighting scheme. The TofCut obtains very robust and precise foreground object segmentation. However, it is very limited to the application of MOD because only one single foreground object can be segmented.

Under the SLAM framework, a groundbreaking work, called Kinect Fusion (KinFu) [274], reconstructs the 3D scenes with the help of RGB-D camera. The KinFu extracts a set of sparse features and estimates the camera motion which is further refined by an ICP algorithm. In the point cloud fusion, a Truncated Signed Distance Function (TSDF) is employed to represent the object surface. With the TSDF, any abrupt change of the state is considered as moving object which will be discarded. Inspired by the KinFu, a more robust MS algorithm [275] for TSDF volumes is presented. The segmentation problem is cast as Conditional Random Field-based MAP inference in the voxel space. In the MOD,

the sparse 3D point correspondences are used to determine the underlying motion groups by applying a RANSAC framework in a greedy manner. Although these KinFu-based approaches achieve very compelling results, such algorithms are particularly designed for indoor environment applications using RGB-D cameras.

2.2.4/ LASER SCANNER

Since the 3D laser scanners produce very precise measure of the scene, they are widely used in outdoor robot navigations. The most representative approaches are the Simultaneous Localization and Mapping with Moving Object Tracking (SLAM-MOT) family [276, 277, 278, 279] which rely on a horizontal single-layered 3D scanner. The SLAM-MOT simultaneously estimates the camera motion and constructs the map of the environments. In the presence of moving objects, the camera localization becomes very difficult. Therefore, MOD is one of the major objective in the SLAM-MOT framework. To detect the moving objects, the SLAM-MOT [278] constructs a statistical occupancy grid map which encodes the probability of the grids belonging to moving objects.

Inspired by the SLAM-MOT, various improvements are achieved. For instance, a spline model fitting approach, so-called SLAM-MOT-Sp [280], integrates the prediction of object motions under the motion consistency assumption. The SLAM-MOT approach is also extended to the usage of a multi-layered 3D laser scanner [281]. Instead of a 2D occupancy map, a probabilistic-based 3D-voxel map is constructed and MOD is achieved in a similar manner. Intuitively, with more observation data, SLAM-MOT based on the 3D-voxel map achieves relatively better results. Similar to the occupancy map, a ray-tracing technique [282] is proposed. The spatial changes are measured in the built map which is obtained from the odometry sensors refined with Iterative Closest Point (ICP) algorithm. In a different manner, a Covariance Area Intersection (CAI)-based approach [283] is proposed to record the states (the assigned probability) of the observed objects.

However, such probabilistic model-based approaches require the prior map information and a relatively long term observation. Moreover, slowly moving objects and small size objects are usually not detected.

Given the initial model of the 3D scene, Point Cloud Subtraction (PCS) approaches [284] detect the dynamic objects by comparing the current map with the known static map. The

key difficulty of such methods comes from the point cloud registration step which is solved by the Normal Distribution Transform Registration. In fact, such PCS approaches are very similar to the image-based background subtraction techniques. However, as a common drawback, the initial clean reference model is required, which makes these methods unsuitable for unknown dynamic environments.

More recently, a 3D feature displacement-based approach [285] is proposed to detect and track the moving objects from a registered sequence using the 3D feature descriptor. Unfortunately, such a method remains very limited by the object motion speed and size and fails to detect objects such as fast moving cars or walking pedestrians.

2.3/ SUMMARY

In this chapter, we have comprehensively reviewed the approaches within the scope of moving object detection and segmentation. The main ideas of the representative works have been revisited and analysed in terms of their strengths and limitations. A quick summary of these approaches is provided in Tables 2.1– 2.4.

As discussed, we consider the Subspace Self-Rpresentation (SSR)-based approaches, especially the Sparse Subspace Clustering (SSC) and the SMooth Rpresentation clustering (SMR), are the most powerful and promising. Moreover, 3D-based approaches relatively have less constraints (e.g. an affine project motion assumption) than the 2D-based approaches. Therefore, we can conclude that:

i Motion segmentation on 3D feature trajectories should be more practical, accurate and robust.

ii By properly incorporating the SSC and SMR approaches to 3D trajectories segmentation, robust and efficient performances should be expected.

iii Current methods mainly rely on image (color or intensity) information. When only point cloud data are available, more efficient and effective algorithms are desired.

	Representative method	Scene type	Frame length	Cam. motion	Background modelling	Related literature
Inter-frame Difference	Pixel/block-wise difference	planar	two	✓	✗	[19, 20, 54, 57]
	Morphological Processing	planar	two	✓	✗	[52, 46]
	Robust Estimation	planar	two	✓	✗	[51, 55, 56]
	Cast Shadow Detector	planar	two	✓	✗	[14, 53]
Statistical Method	Single Gaussian Model	planar	multi	✓	✓	[58, 59, 60, 61]
	Gaussian Mixture Model	planar	multi	✓	✓	[62, 63, 64, 65, 26, 66, 67, 68, 69, 70, 71, 72]
	Maximum-A-Posterior	planar	multi	✓	✓	[58, 21, 75, 76, 59, 47]
	Kernel Density Estimation	planar	multi	✓	✓	[16, 78, 59, 79, 80]
	Belief Propagation	planar	multi	✓	✓	[26, 83, 84, 85, 86, 35]
Spatio-temporal Mechanism	Registration Error Modelling	planar	multi	✓	✗	[89, 90, 91, 92, 79, 61]
	Object Motion and Appearance Modelling	planar	multi	✓	✗	[87, 88, 55, 49]
	Foreground Segmentation	planar	multi	✓	✗	[89, 91, 55]
	Detection and Tracking	planar	multi	✓	✗	[25, 97, 93, 94, 46, 95, 96, 71]
Rank Minimization	Low-Rank Representation	planar	multi	✓	✓	[98, 99, 70, 31, 100, 101]
	PCA	planar	multi	✓	✓	[99, 31]
	3-Term / Tensor Decomposition	planar	multi	✗	✓	[70, 100, 101]
Learning	Statistical Learning	planar	multi	✓	✓	[26, 105, 57, 110]
	Semantic Learning	planar	multi	✗	✗	[111, 112, 26, 106, 107, 108]

TABLE 2.1 – Summary of image-based moving object detection approaches.

Representative method		Scene type	Frame length	Cam. motion	Background modelling	Related literature
Planes + Parallax Decomposition		non-planar	two	✓	✗	[25, 115, 113, 11, 114, 26, 27]
Optical Flow	Displaced Frame Difference	non-planar	two	✗	✗	[139, 37]
	Flow Violation Policy	non-planar	two	✓	✗	[126, 132, 95]
	Focus of Expansion	non-planar	two	✓	✗	[126, 133, 136]
	Object Contour Tracking	non-planar	two	✗	✗	[126, 128, 129, 130, 53, 132, 134, 135, 138, 140, 141]
	Level-set Segmentation	non-planar	two	✗	✗	[145, 137, 146]
	Piecewise Segmentation	non-planar	two	✗	✗	[147, 148]
	Hierarchical Motion Field Seg.	non-planar	two	✗	✗	[149]
	Markov Random Field Seg.	non-planar	two	✗	✗	[128, 130, 140, 135]
Epipolar Constraint	Direct Epipolar Constraint	non-planar	two	✓	✗	[150, 151, 152, 153, 154, 155, 27, 156]
	Robustification Schemes	non-planar	two / multi	✓	✗	[150, 156, 151, 152, 154, 155, 27]
Energy Minimization	Pixel Labelling	non-planar	two / multi	✗	✗	[129, 52, 91, 157, 80, 118, 72, 158]
	Moving Object Extraction	non-planar	two / multi	✗	✗	[129, 148, 159, 91, 80, 160, 72, 158]
Trajectory Analysis	Motion Velocity Analysis	non-planar	multi	✗	✗	[116, 240]
	Object Template Matching	non-planar	multi	✗	✗	[120, 121]
	Trajectory Length Analysis	non-planar	multi	✗	✗	[241, 242, 243]

TABLE 2.2 – Summary of image-based moving object detection approaches.

Representative method		Scene type	Frame length	Cam. motion	Background modelling	Related literature
Trajectory Analysis	Embedding Discontinuity Detector	non-planar	multi	✗	✗	[118]
	Trajectory Subspace	non-planar	multi	✗	✗	[117, 119, 238, 239]
Two-frame Motion Segmentation	Variational Approach	non-planar	two	✓	✗	[145, 137, 161, 162, 146]
	Geometric Constraints	non-planar	two	✓	✗	[163, 165, 166, 167, 168, 169, 170]
	Split-and-Merge	non-planar	two	✓	✗	[171, 172, 173]
	Other Approaches	non-planar	two	✓	✗	[149, 174, 175, 176]
Multi-frame Motion Segmentation	Matrix Factorization	non-planar	multi	✓	✗	[179, 180, 181, 182, 192, 193, 194]
	Shape Interaction Matrix	non-planar	multi	✓	✗	[180, 181, 182, 188, 189, 190, 191]
	Algebraic Methods	non-planar	multi	✓	✗	[195, 196, 28, 197, 198]
	High-Order Clustering	non-planar	multi	✓	✗	[201, 202, 206, 203, 204, 205, 206]
	Tensor Decomposition	non-planar	multi	✓	✗	[208, 202, 203, 205, 204]
	Statistical Approaches	non-planar	multi	✓	✗	[209, 210, 211, 212, 213, 214]
Subspace Self-Representation MS	Sparse Subspace Clustering	non-planar	multi	✗	✗	[177, 216, 29, 217, 218, 224, 225, 226, 231, 235]
	Low Rank Representation	non-planar	multi	✗	✗	[219, 220, 221, 222, 223, 231, 233]
	Least Squares Regression	non-planar	multi	✗	✗	[227, 228, 30]
	Other SSR Approaches	non-planar	multi	✗	✗	[229, 232, 234]

TABLE 2.3 – Summary of image-based moving object detection approaches.

	Representative method	Scene type	Frame length	Cam. motion	Background modelling	Related literature
Structure from Motion	Concurrent 3D Motion Seg.	non-planar	two	✓	✗	[248]
	Two-stage Dense SfM	non-planar	two	✓	✗	[33, 249]
	SLAM-based Approaches	non-planar	two / multi	✓	✗	[251, 252]
Stereo Vision	Global Motion Model	non-planar	two	✓	✗	[253, 254]
	Object Scene Flow	non-planar	two / multi	✓	✗	[255, 256, 257, 258, 259, 260]
	Spatio-temporal Analysis	non-planar	multi	✗	✗	[261, 262, 263]
	Height Color Histogram	non-planar	multi	✗	✗	[264]
RGB-D Sensor	2D-to-3D Extension	non-planar	two / multi	✓	✗ / ✓	[265, 269, 270, 271, 266, 267, 268]
	Depth Guided Segmentation	non-planar	two	✓	✓	[272, 273]
	Kinect Fusion	non-planar	two	✓	✓	[274, 275]
Laser scanner	SLAM-MOT	non-planar	two	✓	✗ / ✓	[276, 277, 278, 279, 280, 281]
	Probabilistic 3D-Voxel Map	non-planar	multi	✓	✓	[282, 283]
	Point Cloud Subtraction	non-planar	two	✓	✓	[284]
	3D Feature Displacement	non-planar	two	✓	✗ / ✓	[285]

TABLE 2.4 – Summary of 3D-based moving object detection methods.

3

PRELIMINARY

“If I have seen further, it is by standing on the shoulders of giants"

- Isaac Newton, *Historical Society of Pennsylvania*

This chapter presents the notations used throughout this document. Some essential concepts and their properties in linear algebra are stated. Among them, the different mathematical spaces with their properties are reviewed.

We briefly revisit the standard formulation of motion segmentation problem based on feature trajectories clustering. To begin with, being the common assumption of subspace self-representation algorithms, the affine projection model and its rank constraints are introduced. Later, given a set of feature trajectories, we construct the two-dimensional data matrix in which each column is a vectorized feature trajectory of multiple moving objects tracked across multiple frames. We also show that, in theory, this data matrix can be decomposed into a block-diagonal matrix where each sub-block represents one independent motion subspace. Thus, the sought motion segmentation problem turns out to be a subspace clustering problem. To this end, two major techniques that inspired our contributions are discussed in details. Specifically, the subspace self-expressiveness property and the well-known Sparse Subspace Clustering formulation are recalled. In addition, the general form of subspace self-representation model for motion segmentation is also pointed out. Moreover, the Smooth Representation Clustering approach is detailed. Lastly, some widely used spectral clustering techniques are listed and analysed. We also introduce some robust estimation techniques which are incorporated in the development of our algorithms. Three major involved techniques, namely the Random Sample Consensus algorithm, the M-Estimator, and the Principal Component Analysis, are dis-

cussed. Finally, some optimization approaches, particularly in convex optimization, are presented. Since we formulate our problems in a convex optimization manner, we intentionally focus on the ℓ_p-norm optimization and analyse the sparsity of their solutions.

3.1/ BASIC NOTATIONS

We use the notations of Table 3.1.

Objects	Notations	Examples
vector	bold lower case letter	$\mathbf{a} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$
matrix	bold upper case letter	$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
transposition	$[\cdot]^\mathsf{T}$	If $\mathbf{a} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$, then $\mathbf{a}^\mathsf{T} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$.
vectorization	$\mathrm{vec}(\cdot)$	If $\mathbf{A} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$, then $\mathrm{vec}(\mathbf{A}) = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}^\mathsf{T}$.
matrix elements	subscript a_{ij}	If $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, then $a_{11} = 1,\ a_{12} = 2$.
column-wise representation	bold lower case with subscript $\mathbf{a}_i$	if $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_i, \cdots \mathbf{a}_n]$, each $\mathbf{a}_i$ is an m-dimensional vector.
diagonal elements	$\mathrm{diag}(\cdot)$	If $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, then $\mathrm{diag}(\mathbf{A}) = \begin{bmatrix} 1 & \\ & 4 \end{bmatrix}$.
trace of matrix	$\mathrm{tr}(\cdot)$	If $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, then $\mathrm{tr}(\mathbf{A}) = \mathbf{A}_{11} + \mathbf{A}_{22} = 1 + 4 = 5$.
positive semi-definite	$\succeq 0$	$\mathbf{A} \succeq 0$ means that the symmetric matrix $\mathbf{A}$ is positive semi-definite.
vector dimension	$\mathbb{R}^m$	$\mathbf{a} \in \mathbb{R}^3$ is a column vector that consists of 3 real-valued elements.
matrix dimension	$\mathbb{R}^{n \times m}$	$\mathbf{A} \in \mathbb{R}^{3 \times 2}$ is a matrix with 3 rows and 2 columns of real-valued elements.

TABLE 3.1 – Notations.

Objects	Notations	Examples
real-positive number	$\mathbb{R}^{+N}$	$\mathbf{a} \in \mathbb{R}^{+N}$ is a vector that only consists of positive real-valued elements.
identity matrix	$\mathbf{I}_m$	$\mathbf{I}_3 \in \mathbb{R}^{3\times3}$ is a diagonal matrix whose diagonal entries are equal to one.
absolute value	$\lvert\cdot\rvert$	If $\mathbf{a} = \begin{bmatrix} -1 & 2 & -3 \end{bmatrix}$, then $\lvert\mathbf{a}\rvert = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$.
ℓ_p-norm	$\lVert\cdot\rVert_p$	$\lVert\mathbf{a}\rVert_2$ denotes the ℓ_2-norm of vector $\mathbf{a}$.
defined by	$:=$	For a vector $\mathbf{a} \in \mathbb{R}^m$, its ℓ_2-norm is defined by $\lVert\mathbf{a}\rVert_2 := \sqrt{a_1^2 + a_2^2 + \cdots + a_m^2}$.
optimal solution	superscript *	Let a linear system be $\mathbf{Ax} = \mathbf{b}$, the optimal solution to this problem is denoted as $\mathbf{x}^*$.
derivative operation	∇	Let $f(x, y) = x^2 + y^2$, the first ordered derivative on x is denoted as $\nabla_x f = 2x$.
space mapping operation	$\rightarrow$	$f(\mathbf{x}) : \mathbb{R}^m \rightarrow \mathbb{R}$ means that the function has variable $\mathbf{x} \in \mathbb{R}^m$ and result $f(x) \in \mathbb{R}$.

TABLE 3.2 – Notations.

3.2/ SPACES

In mathematics, a space is a group of data with some specific structure. Such spaces often form a hierarchy, i.e., a subspace may inherit all the properties of its parent space. There are various types of spaces that are defined by their specific properties, e.g. Euclidean space or Minkowski space. This section introduces some related spaces.

3.2.1/ VECTOR SPACE, AFFINE SPACE, AND SUBSPACES

Definition 1 : Vector Space

A vector space is a non-empty set $\mathbf{V} \subset \mathbb{R}^m$ of objects, called vectors, on which are defined two operations, called addition and multiplication by scalars (real numbers), subject to the axioms (or rules) listed below. The axioms must hold for all vectors $\mathbf{u}, \mathbf{v} \in \mathbf{V}$ and for all scalars c [286]:

1 . If $\mathbf{u}, \mathbf{v} \in \mathbf{V}$, then $\mathbf{u} + \mathbf{v} \in \mathbf{V}$.

2 . If $\mathbf{u}, \mathbf{v} \in \mathbf{V}$ with a scalar c, then $c\mathbf{u}, c\mathbf{v}, c(\mathbf{u} + \mathbf{v}) \in \mathbf{V}$.

3 . There is a **zero** vector $\mathbf{0} \in \mathbf{V}$ such that $\mathbf{v} + \mathbf{0} \in \mathbf{V}$.

For different vectors from the same vector space, axiom 1 in definition 1 implies that their linear combination also belongs to the same vector space. In addition, axiom 2 indicates the linear property of vector space where vectors multiplied by scalars remain in the same vector space. In other words, vectors can be reproduced by the linear combination of other vectors from the same vector space. Therefore, a vector space is closed under addition and scalar multiplication. Moreover, a vector space must contain the **zero** vector (axiom 3), such that a vector space passes through the origin. On the contrary, if a “vector space" contains no **zero** vector, it is called an *Affine Space*. The affine space preserves same properties of axiom 1 and axiom 2 when $c \neq 0$. Fig. 3.1 illustrates different examples of a vector space and an affine space. Noticeably, plane Π_2 is an affine space since it does not go through the origin.

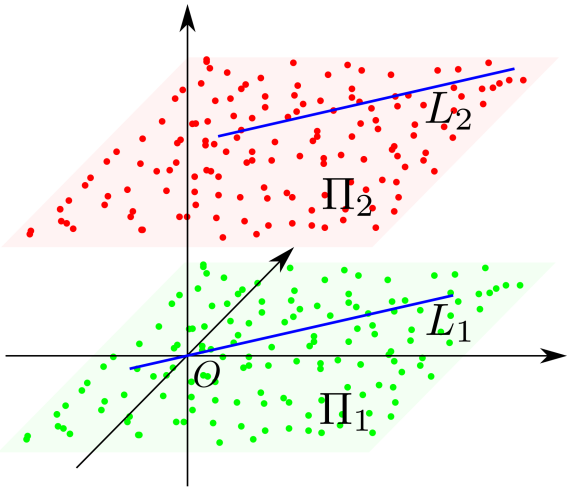


FIGURE 3.1 – Examples of vector space and affine space: The two planes represent a vector space (Π_1) and an affine space (Π_2), respectively. L_1, which lies in Π_1 and passes the origin, is a subspace of Π_1. L_2, which belongs to Π_2, is an affine subspace of Π_2.

Definition 2 : Vector Subspace

A vector subspace (or linear subspace) of $\mathbb{R}^n$ is any set $\mathbf{S} \subset \mathbb{R}^n$ that verifies the following three properties [286]:

1 . For $\mathbf{u}, \mathbf{v} \in \mathbf{S}$, the sum $\mathbf{u} + \mathbf{v}$ is also in $\mathbf{S}$.

2 . For $\mathbf{u} \in \mathbf{S}$, the vector $c\mathbf{u}$ is also in $\mathbf{S}$.

3 . The zero vector is in $\mathbf{S}$.

The above definition 2 shows that a *Vector Subspace* is a vector space, and also a subset of a higher-dimensional space. Similarly, an *Affine Subspace* can be defined as a subset of an affine space. To illustrate, Fig. 3.1 shows an example of the vector subspace L_1

inside a vector space Π_1, as well as an example of affine subspace L_2 corresponding to affine space Π_2. In fact, both the plane Π_1 and the line L_1 are linear subspaces of the 3D Euclidean space.

3.2.2/ COLUMN SPACE, ROW SPACE AND NULL SPACE

Let's first introduce the concept of *Spanning* a space. If a vector space $\mathbf{V} \subset \mathbb{R}^{m\times n}$ consists of all linear combinations of $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n$, then these vectors span the space. The *Column Space* of $\mathbf{V}$ is spanned by the columns. Similarly, the *Row Space* of $\mathbf{V}$ is the span of the rows. The *Null Space* is the span of vectors that are perpendicular [1] to the row space.

Consider an over-determined linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{3.1}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ $(m > n)$ $\mathbf{x} \in \mathbb{R}^n$, and $\mathbf{b} \in \mathbb{R}^n$. When $\mathbf{b} = \mathbf{0}$, the sought solutions of the linear system are the span of vectors perpendicular to the row space of $\mathbf{A}$. [2] Therefore, solving linear system 3.1 is, in essence, finding the null space of $\mathbf{A}$. In this regard, *Singular Value Decomposition* (SVD) is an efficient way to solve such a problem. In general, the SVD operation has the following form:

$$SVD(\mathbf{A}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}, \tag{3.2}$$

where Σ is a diagonal matrix whose entries are the eigenvalues of $\mathbf{A}^\mathsf{T}\mathbf{A}$. Note that all the singular values (entries of Σ) are no-less than zero because $\mathbf{A}^\mathsf{T}\mathbf{A}$ is positive definite. The columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{A}\mathbf{A}^\mathsf{T}$, while the columns of $\mathbf{V}$ are the eigenvectors of $\mathbf{A}^\mathsf{T}\mathbf{A}$. The solution space of problem 3.1 is the span of the singular vectors in $\mathbf{V}^\mathsf{T}$ corresponding to the zero singular values.

1. For two vector spaces $\mathbf{U} \in \mathbb{R}^{m\times n}$, $\mathbf{V} \in \mathbb{R}^{n\times m}$, if $\mathbf{U}\mathbf{V} = \mathbf{0}$, then $\mathbf{U}$ and $\mathbf{V}$ are orthogonal to each other.
2. When $\mathbf{b} \neq \mathbf{0}$, one can rewrite the linear system as $\mathbf{A}'\mathbf{x}' = \mathbf{0}$. Where $\mathbf{A}' \in \mathbb{R}^{m\times(n+1)}$; $\mathbf{x}' \in \mathbb{R}^{n+1}$; the last column of $\mathbf{A}'$ equals to $\mathbf{b}$; and the last element of $\mathbf{x}$ equals to -1.

3.2.3/ SUBSPACE CLUSTERING

Given a set of data point $\{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n\} \in \mathbb{R}^m$ which belong to K different subspaces $\{\mathbf{S}_i\}_{i=1}^K$, the subspace clustering aims to group those elements $\mathbf{a}_i$ into their respective subspaces $\mathbf{S}_i$. Fig. 3.2 shows that the 3D data point set is from 3 different subspaces, namely the red line L_1, the blue line L_2, and the plane Π. Clearly, the subspace clustering intends to classify the 3D points into L_1, L_2 and Π.

The subspace clustering is applied to both linear subspace and affine subspace. In fact, the affine subspace can be considered as lying on a higher-dimensional linear subspace. For instance, a line L, which does not go through the origin, is an affine subspace. This affine subspace is lying on the plane which passes through both the origin and the line L. Therefore, both linear subspace and affine subspace are within the scope of subspace clustering.

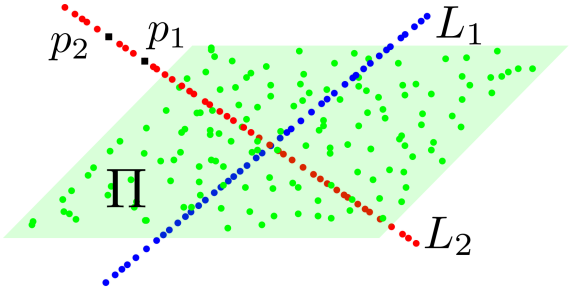


FIGURE 3.2 – Subspace clustering example: the linear subspaces, namely L_1, L_2 and Π, are intersecting at the origin. p_1, p_2 are two elements in L_2. The objective of subspace clustering is to group the elements to their corresponding subspaces.

3.3/ SUBSPACE FORMULATION FOR MOTION SEGMENTATION

The motion segmentation problem is a fundamental problem in computer vision [25]. Given an image sequence, it is recommended to detect and analyse the moving objects (or motions) based on the feature trajectories. For rigidly moving objects, an independent motion has a unique motion track which is determined by its velocity, direction, and spatial position. Mathematically, an independent motion can be modelled as a unique motion space where each feature trajectory associated to this moving object is a subspace. This section introduces the fundamentals of the popular subspace clustering approach for motion segmentation.

3.3.1/ AFFINE PROJECTION MODEL

Camera modelling is another fundamental problem in computer vision. Geometrically, the camera models are the mapping of data from 3D space to 2D image space. Among the different camera models, the perspective projection model is the ideal and the most accurate model for a wide range of cameras [287]. However, the resulting equations from perspective projection model are often complicated and non-linear due to the unknown scale factor [288]. For simplicity, there are various approximation models, namely weak-perspective projection model, orthographic projection model, and para-perspective projection model, which are generalized as *Affine Projection Model* [18]. The affine camera projection model is relatively simpler compared to the projective camera model. Especially, when the depth variation of objects is small compared to the camera-to-object distance, the affine projection model is valid as a proper approximation to perspective projection model.

Given a 3D point $[X, Y, Z]^{\mathsf{T}}$ projected onto the 2D space (or image plane) as $[x, y]^{\mathsf{T}}$, the affine projection model has the following form:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\mathbf{K} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}}_{\mathbf{P}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{3.3}$$

where $\mathbf{K} \in \mathbb{R}^{3\times3}$ consists of the intrinsic camera parameters. $\mathbf{R} \in \mathbb{R}^{3\times3}$ and $\mathbf{t} \in \mathbb{R}^3$ are the rotation and translation, respectively. The combination $\mathbf{P} \in \mathbb{R}^{3\times4}$ is called the camera matrix of an affine camera. It is straight forward that $\mathbf{P}$ has the form of

$$\mathbf{P} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_1 \\ a_{21} & a_{22} & a_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.4}$$

where t_1, t_2 are the translations in X and Y directions, respectively. Taking the first two rows of $\mathbf{P}$, we define the affine motion matrix $\mathbf{A} \in \mathbb{R}^{2\times4}$ as

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_1 \\ a_{21} & a_{22} & a_{23} & t_2 \end{bmatrix}. \tag{3.5}$$

Finally, the affine projection model can be simplified as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{A} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{3.6}$$

3.3.2/ FEATURE TRAJECTORY SUBSPACE

Suppose a moving camera observes a 3D point $[X, Y, Z]^{\mathsf{T}}$ over F frames, then we have F image points $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_F$ and F affine camera matrices $\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_F$. Stacking the image points $\mathbf{x}_i = [x_i, y_i]^{\mathsf{T}}$ into a feature trajectory vector leads to

$$\begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_F \\ y_F \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_F \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{3.7}$$

Now, let's consider N 3D points $[X_i, Y_i, Z_i]^{\mathsf{T}}$ $(i = 1, 2, \ldots, N)$ are observed in the i^{th} frame under the affine motion $\mathbf{A}_i$, we have

$$\begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{iN} \\ y_{i1} & y_{i2} & \cdots & y_{iN} \end{bmatrix} = \mathbf{A}_i \begin{bmatrix} X_1 & X_2 & \cdots & X_N \\ Y_1 & Y_2 & \cdots & Y_N \\ Z_1 & Z_2 & \cdots & Z_N \\ 1 & 1 & \cdots & 1 \end{bmatrix}. \tag{3.8}$$

Combining Eq. (3.7) and Eq. (3.8), for N points observed in F frames, we have N feature

trajectories under F affine motions:

$$\underbrace{\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ y_{11} & y_{12} & \cdots & y_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{F1} & x_{F2} & \cdots & x_{FN} \\ y_{F1} & y_{F2} & \cdots & y_{FN} \end{bmatrix}}_{\mathsf{X}\in\mathbb{R}^{2F\times N}} = \underbrace{\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_F \end{bmatrix}}_{\mathbf{M}\in\mathbb{R}^{2F\times 4}} \underbrace{\begin{bmatrix} X_1 & X_2 & \cdots & X_N \\ Y_1 & Y_2 & \cdots & Y_N \\ Z_1 & Z_2 & \cdots & Z_N \\ 1 & 1 & \cdots & 1 \end{bmatrix}}_{\mathbf{S}\in\mathbb{R}^{4\times N}}, \tag{3.9}$$

where X is the data matrix encoding the set of image feature trajectories, $\mathbf{M}$ is the combination of all affine motion matrices, and $\mathbf{S}$ is the shape matrix containing all the 3D features in homogeneous coordinate. Consider the four columns of $\mathbf{M}$ as the four basis of a linear subspace with intrinsic dimension of 4 and ambient dimension of $2F$, and the elements of $\mathbf{S}$ are the scale factors, then the columns of X become points in this subspace. Therefore, the assembly of N feature trajectories naturally forms a linear subspace.

Note that all these N features in Eq. (3.9) are under the same motion $\mathbf{M}$. In cases of multiple independent motions, say K different motions, Eq. (3.9) can be extended and factorized as

$$\begin{aligned} \mathbf{X} &= [\mathsf{X}_1, \mathsf{X}_2, \cdots, \mathsf{X}_K] \\ &= [\mathbf{M}_1\mathbf{S}_1, \mathbf{M}_2\mathbf{S}_2, \cdots, \mathbf{M}_K\mathbf{S}_K] \\ &= [\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_K] \begin{bmatrix} \mathbf{S}_1 & & & \\ & \mathbf{S}_2 & & \\ & & \ddots & \\ & & & \mathbf{S}_K \end{bmatrix}. \end{aligned} \tag{3.10}$$

where $\mathbf{M}_i, \mathbf{S}_i$ are the i^{th} motion matrix and the i^{th} shape matrix for K rigidly moving objects, respectively. $\mathbf{X}$ is the assembly of all feature trajectories from K different motions. In practice, the trajectories are not sorted according to their motions, which raises the problem of feature trajectories clustering. Naturally, the span of feature trajectories forms an independent and unique affine motion space where each element (or feature trajectory) is an affine subspace. Therefore, clustering the feature trajectories is essentially a subspace clustering problem.

3.3.3/ SUBSPACE SELF-REPRESENTATION MODEL

Given that $\mathbf{X}$ is disorganized, factorization of Eq. (3.10) compromises to a certain "permutation" matrix[3] $\mathbf{C}$ of $\mathbf{X}$. In other words, the subspace clustering problem raised by problem (3.10) points to finding the permutation matrix $\mathbf{C}$. Such a permutation matrix is used to construct the *Affinity Matrix* which encodes the connectivity (or the similarity) between the feature trajectories. More specifically, two feature trajectories from the same subspace should have high affinity while two feature trajectories from different subspaces have low affinity.

To this end, the SSC [29] algorithm introduced a very important concept which is called the *Subspace Self-expressiveness Property*:

Definition 3 : Subspace Self-expressiveness Property [29]

One element can be represented (or approximated) by the linear combination of other elements from the same subspace, so called Subspace Self-expressiveness.

Formally, suppose there exists a collection of data $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]$ where the columns lie in multiple subspaces, then for all $i = 1, \cdots, N$, there exists a vector $\mathbf{r}_i \in \mathbb{R}^N$, such that $\mathbf{x}_i = \mathbf{X}\mathbf{r}_i$. The non-zero entries of $\mathbf{r}_i$ are other elements from the same subspace. To understand this concept, recall the linearity property of the affine subspace: the affine subspace is bound under addition and non-zero scalar multiplication. Thus, the subspace self-expressiveness property is induced from the linearity of subspace. For example, in Fig. 3.2, p_1 and p_2 are from the same subspace. It is manifest that p_1 can represent p_2 by multiplying a scale factor. An affinity matrix can then be constructed by using these linear combination coefficients.

Formally, in SSC, the self-expressiveness property is expressed as:

$$\mathbf{X} = \mathbf{X}\mathbf{C} \quad \text{subject to} \quad \text{diag}(\mathbf{C}) = \mathbf{0}, \tag{3.11}$$

where $\mathbf{X}$ is the data matrix (or the assembly of feature trajectories) and $\mathbf{C}$ is the coefficient (or the permutation) matrix. To avoid the trivial solution that $\mathbf{C}$ equals to identity, the

3. In this thesis, following [29], we refer the permutation matrix to a non-negative coefficient matrix. Different from the traditional definition of permutation matrix, such non-negative coefficients are not necessarily to be 0 or 1.

zero-diagonal constraint $\mathrm{diag}(\mathbf{C}) = \mathbf{0}$ is adopted. The non-empty coefficient matrix $\mathbf{C}$ is used to build the symmetric affinity matrix $\mathbf{Q} = \mathbf{C} + \mathbf{C}^{\mathsf{T}}$. Such affinity matrix $\mathbf{Q}$ encodes the intra- and inter-subspace relationship for the later spectral clustering. To solve Eq. (3.11), inspired by compressive sensing theory, the SSC aims to minimize the ℓ_0-norm of $\mathbf{C}$ to obtain a sparse solution:

$$\begin{aligned} &\underset{\mathbf{C}}{\text{minimize}} \quad \|\mathbf{C}\|_0 \\ &\text{subject to} \quad \mathbf{X} = \mathbf{X}\mathbf{C}, \quad \mathrm{diag}(\mathbf{C}) = \mathbf{0}. \end{aligned} \tag{3.12}$$

In practice, solving problem (3.12) is very difficult due to the non-convexity of ℓ_0-norm (this will be discussed in details in Section 3.5.1). Accordingly, problem (3.12) is relaxed as an ℓ_1-norm optimization problem which also produces sparse solution, such that

$$\begin{aligned} &\underset{\mathbf{C}}{\text{minimize}} \quad \|\mathbf{C}\|_1 \\ &\text{subject to} \quad \mathbf{X} = \mathbf{X}\mathbf{C}, \\ &\qquad\qquad\quad \mathbf{1}^{\mathsf{T}}\mathbf{C} = \mathbf{1}^{\mathsf{T}}, \\ &\qquad\qquad\quad \mathrm{diag}(\mathbf{C}) = \mathbf{0}. \end{aligned} \tag{3.13}$$

When dealing with affine subspace, in Eq. (3.13), the constraint $\mathbf{1}^{\mathsf{T}}\mathbf{C} = \mathbf{1}^{\mathsf{T}}$ is incorporated to enforce the sum of each column of $\mathbf{C}$ to be 1. Alternatively, one can add an all-one row to $\mathbf{X}$ so that the affine constraint is implicitly enforced.

Inspired by SSC, recent motion segmentation methods use such property to cluster the motion trajectories. In summary, the self-representation model for motion segmentation can be generalized as

$$\underset{\mathbf{C}}{\text{minimize}} \quad \|\mathbf{X} - \mathcal{D}(\mathbf{X})\mathbf{C}\|_\ell + \Omega(\mathbf{X}, \mathbf{C}) \quad \text{subject to} \quad \zeta(\mathbf{C}) \tag{3.14}$$

where $\mathcal{D}(\mathbf{X})$ is the dictionary learned from $\mathbf{X}$, and $\|\cdot\|_\ell$ denotes the proper norm. $\Omega(\mathbf{X}, \mathbf{Z})$ is the regularization term and $\zeta(\mathbf{C})$ are the constraints on $\mathbf{C}$. By solving Eq. (3.14), a desired self-representation matrix $\mathbf{C}^*$ is obtained to construct the affinity matrix. Table 3.3 summarizes some state-of-the-art algorithms based on the subspace self-expressiveness property. Note that, depending on the selection of ℓ_p-norm, the solutions of these methods are either sparse or dense, as summarized in Table 3.3.

Algorithm	$\Omega(\mathbf{X}, \mathbf{C})$	$\|\cdot\|_l$	$\zeta(\mathbf{C})$
CASS[228]	$\sum_i \|\mathbf{X}\text{diag}(\mathbf{C})\|_*$	$\|\cdot\|_F^2$	$\emptyset$
LRR[219, 220]	$\|\mathbf{C}\|_*$	$\|\cdot\|_{2,1}$	$\emptyset$
LatLRR[223, 221]	$\|\mathbf{C}\|_* + \|\mathbf{L}\|_* + \lambda\|\mathbf{E}\|_1$	$\|\cdot\|_{2,1}$	$\mathbf{C}^* = \mathbf{C} + \mathbf{E}, \mathbf{C} \geq 0$
LSR[227]	$\|\mathbf{C}\|_F^2$	$\|\cdot\|_F^2$	$\text{diag}(\mathbf{C}) = \mathbf{0}$
LSR-Z[227]	$\|\mathbf{C}\|_F^2$	$\|\cdot\|_F^2$	$\text{diag}(\mathbf{C}) = \mathbf{0}$
LS3C[224]	$\|\mathbf{C}\|_1 + \lambda_1\|\mathbf{PY} - \mathbf{PYC}\|_F^2 + \lambda_2\|\mathbf{PY} - \mathbf{P}^\mathsf{T}\mathbf{YC}\|_F^2$	$\|\cdot\|_F^2$	$\text{diag}(\mathbf{C}) = \mathbf{0}, \mathbf{PP}^\mathsf{T} = \mathbf{I}, \mathbf{C}^\mathsf{T}\mathbf{1} = 1$
MSR[289]	$\|\mathbf{C}\|_1 + \delta\|\mathbf{C}\|_*$	$\|\cdot\|_{2,1}$	$\text{diag}(\mathbf{C}) = \mathbf{0}$
SMR[30]	$\|\mathbf{C}\|_F^2$	$\|\cdot\|_F^2$	$\emptyset$
SSC[29, 177]	$\|\mathbf{C}\|_1$	$\|\cdot\|_1$	$\text{diag}(\mathbf{C}) = \mathbf{0}$
SSQP[218]	$\|\mathbf{C}^\mathsf{T}\mathbf{C}\|_1$	$\|\cdot\|_F^2$	$\mathbf{C} \in \mathbb{R}^+, \text{diag}(\mathbf{C}) = \mathbf{0}$

TABLE 3.3 – Subspace self-representation motion segmentation methods.

Different from the SSC which models the motion segmentation problem as a constrained optimization problem, SMR [30] addresses the problem as an unconstrained optimization problem by enforcing the *Grouping Effect*.

Definition 4 : Grouping Effect [30]

Given a set of d-dimensional data points $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N] \in \mathbb{R}^{d \times N}$, a self-representation matrix $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_N] \in \mathbb{R}^{N \times N}$ has grouping effect if $\|\mathbf{x}_i - \mathbf{x}_j\|_2 \to 0 \Rightarrow \|\mathbf{c}_i - \mathbf{c}_j\|_2 \to 0, \forall i \neq j$.

The grouping effect implies that if the difference between two points (or feature trajectories) $\mathbf{x}_i, \mathbf{x}_j$ is very small, then the difference between their respective self-representation coefficients $\mathbf{c}_i, \mathbf{c}_j$ is also very small. Geometrically, if two points are very close to each other, then their respective self-representation coefficients are very similar to each other. In other words, the grouping effect enforces the spatial closeness between the feature trajectories. Explicitly, if the feature trajectories belong to the same moving objects, these feature trajectories should be closely distributed and lie in the same subspace. Formally, the grouping effect constraint is enforced as a regularization term:

$$\begin{aligned} \Omega(\mathbf{X}, \mathbf{C}) &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\|\mathbf{C}_i - \mathbf{C}_j\|_2^2 \\ &= \text{tr}(\mathbf{CLC}^\mathsf{T}) \end{aligned}, \tag{3.15}$$

where $w_{ij} = \|\mathbf{X}_i - \mathbf{X}_j\|_2^2$ defines the spatial closeness of two feature trajectories. A symmetric spatial distance graph $\mathbf{W}$ can then be constructed as

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & \cdots & w_{NN} \end{bmatrix}. \tag{3.16}$$

In Eq. (3.15), $\mathbf{L}$ is the Laplacian matrix defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is a diagonal matrix with entry $d_{ii} = \sum_{j=1}^{N} w_{ij}$. By enforcing the grouping effect, Eq. (3.14) is then adapted as

$$\underset{\mathbf{C}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2 + \text{tr}(\mathbf{C}\mathbf{L}\mathbf{C}^\mathsf{T}). \tag{3.17}$$

3.3.4/ Spectral Clustering

For most of the motion segmentation methods [29, 30, 187], the spectral clustering is an important step to obtain the final segmentation. Getting the self-representation matrix $\mathbf{C}^*$, a symmetric affinity matrix $\mathbf{Q} = \mathbf{C} + \mathbf{C}^\mathsf{T}$ is constructed. By using the affinity matrix, a Laplacian matrix is built to perform spectral clustering. In this section, two major spectral clustering methods are presented.

3.3.4.1/ Unnormalized spectral clustering

Let $Q \in \mathbb{R}^{N \times N}$ be a non-negative symmetric affinity matrix which also forms an undirected weighted graph $\mathbf{G}$. The *Unnormalized Laplacian Matrix* is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{Q}, \tag{3.18}$$

where $\mathbf{D}$ is a diagonal matrix whose entries are the sum of rows (or columns) of $\mathbf{Q}$, denoted as $d_{ii} = \sum_{j=1}^{N} q_{ij}$. Note that $\mathbf{L}$ is positive semi-definite[4] because the following condition always holds true [290]:

4. A matrix $\mathbf{A}$ is positive semi-definite if, for any vector $\mathbf{x}$, $\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} \geq 0$.

$$\mathbf{x}^{\mathrm{T}}\mathbf{L}\mathbf{x} = \sum_{i=1}^{N}\sum_{j=1}^{N} q_{ij}(x_i - x_j)^2 \geq 0, \tag{3.19}$$

where q_{ij} are the entries of affinity matrix $\mathbf{Q}$, and x_i, x_j are the elements of an arbitrary vector $\mathbf{x} \in \mathbb{R}^N$. In addition, the smallest eigenvalue of $\mathbf{L}$ equals to zero with the corresponding constant-$\mathbf{1}$ eigenvector. Furthermore, $\mathbf{L}$ is a singular matrix due to the zero-summation of entries of every column.

Proposition 1 (Number of connected components and the spectrum of $\mathbf{L}$ [290])**.** *Let* $\mathbf{G}$ *be an undirected graph with non-negative weights. Then the multiplicity* K *of the eigenvalue 0 of* $\mathbf{L}$ *equals the number of connected components* $\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_K$ *in the graph. The eigenspace of eigenvalue 0 is spanned by the indicator vectors* $\mathbf{1}_{\mathbf{A}_1}, \mathbf{1}_{\mathbf{A}_2}, \cdots, \mathbf{1}_{\mathbf{A}_K}$ *of those components.*

Proposition 1 says that the number of clusters equals to the multiplicity of zero-eigenvalue of $\mathbf{L}$. Thus, $\mathbf{L}$ can be reorganized as a block-diagonal matrix corresponding to the K different clusters:

$$\mathbf{L} = \begin{bmatrix} \mathbf{A}_1 & & & \\ & \mathbf{A}_2 & & \\ & & \ddots & \\ & & & \mathbf{A}_K \end{bmatrix}. \tag{3.20}$$

Therefore, the objective of spectral clustering can be achieved by categorizing the Laplacian matrix $\mathbf{L}$ into a K block-diagonalized components. Such components are ideally intra-connected but inter-disconnected.

With proposition 1, the unnormalized spectral clustering algorithm is summarized as

Algorithm 1: Unnormalized Spectral Clustering [290]

Input : Affinity matrix $\mathbf{Q} \in \mathbb{R}^{N\times N}$, number of clusters K.
Output: Cluster labels $\mathcal{L}$.

1 Compute the unnormalized Laplacian $\mathbf{L}$ using Eq. 3.18.
2 Compute the smallest K eigenvectors $u_1, u_2, \cdots, u_K$ of $\mathbf{L}$.
3 Construct $\mathbf{U} \in \mathbb{R}^{N\times K}$ with $u_1, u_2, \cdots, u_K$ as columns, and each row of $\mathbf{U}$ is denoted as $\mathbf{r}_i \in \mathbb{R}^K, i = 1, 2, \ldots, N$.
4 Cluster the points $\{\mathbf{r}_i \in \mathbb{R}^K\}_{i=1}^N$ using K-means algorithm, and return the cluster labels $\mathcal{L} \in \{1, 2, \ldots, K\}^N$.

3.3.4.2/ NORMALIZED SPECTRAL CLUSTERING

To improve the clustering performance, there are two popular ways to normalize the Laplacian matrix, i.e. [178, 123]

$$\begin{aligned} \mathbf{L}_{\text{sym}} &:= \mathbf{D}^{\frac{1}{2}}\mathbf{L}\mathbf{D}^{\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{\frac{1}{2}}\mathbf{Q}\mathbf{D}^{\frac{1}{2}} \\ \mathbf{L}_{\text{rw}} &:= \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{Q} \end{aligned} . \tag{3.21}$$

The first notation $\mathbf{L}_{\text{sym}}$ denotes a symmetric matrix, while the second notation $\mathbf{L}_{\text{rw}}$ is closely related to the random walk algorithm.

Proposition 2 (Properties of $\mathbf{L}_{\text{sym}}$ and $\mathbf{L}_{\text{rw}}$ [290])**.** *The normalized Laplacians satisfy the following properties:*

i. For every $\mathbf{x} \in \mathbb{R}^N$, we have

$$\mathbf{x}^\top \mathbf{L}_{\text{sym}}\mathbf{x} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} q_{ij}\left(\frac{x_i}{\sqrt{d_{ii}}} - \frac{x_j}{\sqrt{d_{jj}}}\right)^2 \geq 0. \tag{3.22}$$

ii. 0 is an eigenvalue of $\mathbf{L}_{\text{rw}}$ with the constant one vector $\mathbf{1}$ as eigenvector. 0 is an eigenvalue of $\mathbf{L}_{\text{sym}}$ with eigenvector $\mathbf{D}^{\frac{1}{2}}\mathbf{1}$.

iii. $\mathbf{L}_{\text{sym}}$ and $\mathbf{L}_{\text{rw}}$ are positive semi-definite and have N non-negative real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \lambda_N$.

The above proposition 2 shows that the normalize Laplacian matrices $\mathbf{L}_{\text{sym}}$ and $\mathbf{L}_{\text{rw}}$ have very similar properties with the unnormalized Laplacian. Moreover, the multiplicity K of the zero-eigenvalue of $\mathbf{L}_{\text{sym}}$ (also apply for $\mathbf{L}_{\text{rw}}$) equals to the number of clusters. More specifically, the following proposition holds true:

Proposition 3 (Number of connected components and spectra of $\mathbf{L}_{\text{sym}}$ and $\mathbf{L}_{\text{rw}}$ [290])**.** *Let $\mathbf{G}$ be an undirected graph with non-negative weights. Then the multiplicity K of the eigenvalue 0 of both $\mathbf{L}_{\text{sym}}$ and $\mathbf{L}_{\text{rw}}$ equals the number of connected components $\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_K$ in the graph. For $\mathbf{L}_{\text{rw}}$, the eigenspace of 0 is spanned by the indicator vectors $\mathbf{1}_{\mathbf{A}_i}$ of those components. For $\mathbf{L}_{\text{sym}}$, the eigenspace of 0 is spanned by the vectors $\mathbf{D}^{\frac{1}{2}}\mathbf{1}_{\mathbf{A}_i}$.*

By incorporating the normalization of the Laplacian matrix, two different spectral clustering algorithms are proposed. Firstly, spectral clustering algorithm 2 using normalized symmetric Laplacian is quite similar to its unnormalized version. The only difference comes from the additional normalization step of algorithm 2.

Algorithm 2: Normalized Spectral Clustering using $\mathbf{L}_{\text{sym}}$ [178]

Input : Affinity matrix $\mathbf{Q} \in \mathbb{R}^{N \times N}$, number of clusters K.
Output: Cluster labels $\mathcal{L}$.

1 Compute the normalized Laplacian $\mathbf{L}_{\text{sym}}$ using Eq. 3.21.
2 Compute the smallest K eigenvectors $u_1, u_2, \cdots, u_K$ of $\mathbf{L}_{\text{sym}}$.
3 Construct $\mathbf{U} \in \mathbb{R}^{N \times K}$ with $u_1, u_2, \cdots, u_K$ as columns, and each row of $\mathbf{U}$ is denoted as $\mathbf{r}_i \in \mathbb{R}^K, i = 1, 2, \ldots, N$.
4 Normalize $\mathbf{r}_i$ as $\bar{\mathbf{r}}_i = \frac{\mathbf{r}_i}{\|\mathbf{r}_i\|}$ for $i = 1, 2, \ldots, N$.
5 Cluster the points $\{\bar{\mathbf{r}}_i \in \mathbb{R}^K\}_{i=1}^N$ using K-means algorithm, and return the cluster labels $\mathcal{L} \in \{1, 2, \ldots, K\}^N$.

The next algorithm 3 uses the asymmetric normalized Laplacian matrix $\mathbf{L}_{\text{rw}}$. Note that in algorithm 3 the generalized eigenvectors of $\mathbf{L}$ are the same as the eigenvectors of $\mathbf{L}_{\text{rw}}$. This can be interpreted from the perspective of normalized graph cuts, so it is also called the normalized cuts algorithm [123].

Algorithm 3: Normalized Spectral Clustering using $\mathbf{L}_{\text{rw}}$ [123]

Input : Affinity matrix $\mathbf{Q} \in \mathbb{R}^{N \times N}$, number of clusters K.
Output: Cluster labels $\mathcal{L}$.

1 Compute the normalized Laplacian $\mathbf{L}$ as in Eq. 3.18.
2 Compute the smallest K eigenvectors $u_1, u_2, \cdots, u_K$ of the generalized eigenproblem $\mathbf{Lu} = \lambda\mathbf{Du}$.
3 Construct $\mathbf{U} \in \mathbb{R}^{N \times K}$ with $u_1, u_2, \cdots, u_K$ as columns, and each row of $\mathbf{U}$ is denoted as $\bar{\mathbf{r}}_i \in \mathbb{R}^K, i = 1, 2, \ldots, N$.
4 Cluster the points $\{\bar{\mathbf{r}}_i \in \mathbb{R}^K\}_{i=1}^N$ using K-means algorithm, and return the cluster labels $\mathcal{L} \in \{1, 2, \ldots, K\}^N$.

To sum up, the three spectral clustering algorithms 1 2 3 are quite similar, except that their Laplacian matrices are different. All these algorithms seek a different representation of data point $\mathbf{x}_i \in \mathbf{R}^N$ to $\bar{\mathbf{r}}_i \in \mathbf{R}^K$. The new data representations are the eigenvectors of the Laplacian matrix, which yields to a lower-dimensional but more distinctive space. With such representations, we simply apply the K-means algorithm to separate the different clusters. As discussed in [290], the normalization-based methods usually give better results. Accordingly, evaluations of motion segmentation performance in this thesis mainly use the normalization-based spectral clustering algorithms.

3.4/ ROBUST ESTIMATION METHODS

The observed data can be corrupted or affected by noise. Hence, estimation techniques

are recommended. In this section, three major techniques, namely RANSAC algorithm, M-Estimator, and PCA algorithm, are discussed.

3.4.1/ RANDOM SAMPLE CONSENSUS ALGORITHM

Definition 5 : RANSAC Algorithm [291]

RANdom SAmple Consensus (RANSAC) is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers, when outliers are to be accorded no influence on the values of the estimates.

RANSAC algorithm is one of the most popular techniques in robust statistics, i.e. in Computer Vision [18], where it learns the best parameter fitting to the data. Intuitively, RANSAC algorithm iteratively solve an overdetermined system [5] by randomly selecting the best sub-samples which guarantee the largest percentage of inlier samples. A simple RANSAC algorithm of a line-fitting example can be summarized as:

Algorithm 4: RANSAC Algorithm of Line Fitting.

Data: A set of 2D points $\mathbf{x}$, inlier threshold τ, maximum iteration K.
Result: Optimal fitted line l^*.

$iter = 0,\ maxInlierNb = 0$;
for $iter < K$ **do**
    1. Randomly select two points;
    2. Compute sample line $f_i(x) : Ax = b$;
    3. Count inlier number $m : f(x_i) \leq \tau,\ \text{for} i = 1, \ldots, N$;
    if $m > maxInlierNb$ **then**
        $maxInlierNb \leftarrow m$;
        $l^* \leftarrow f_i(x)$;
    else
        go back to the beginning of current section;

In the above Algorithm 4, the computation time is determined by iteration number K which can be approximated as

$$K = \frac{log(1-p)}{log(1-w^N)}, \tag{3.23}$$

where p is the probability of getting at least one sample set which contains only inliers. w is the inlier ratio and N is the size of the sample set (or the number of parameters). Fig. 3.3

5. An overdetermined system has more equations (constraints) than the system unknowns, in contrast to an underdetermined which has fewer equations than the system unknowns.

shows that the iteration number is increasing exponentially as the number of parameters increases. Moreover, taking the same value of p, the iteration time boosts dramatically when inlier ratio w decreases.

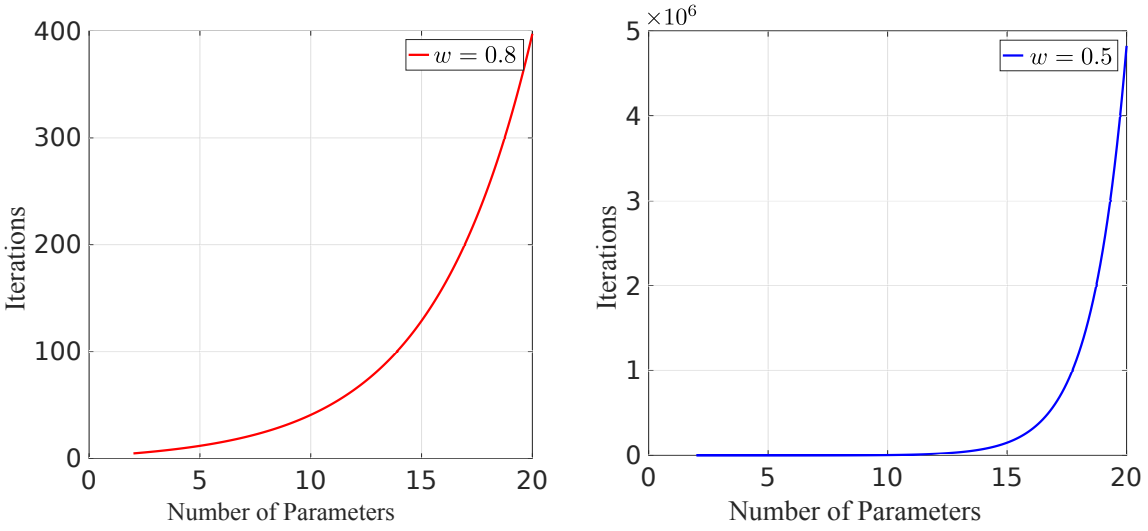


FIGURE 3.3 – RANSAC iteration estimation: Consider $p = 99\%$ of chance to have a solution, figures show the required iterations for specific inlier ratios, $w = 80\%$ and $w = 50\%$ respectively, with regard to different parameter sizes. Clearly, for lower inlier ratio, the algorithm requires significantly more iterations.

The main advantage of the RANSAC algorithm is its simplicity and generality in implementation as a robust estimation framework. Although relatively noisy data with significant amount of outliers are presented, RANSAC algorithm can still performs highly accurate parameter fitting. However, since the RANSAC algorithm is not a brute-force searching algorithm, such that it may not always find the optimal solution. Especially, when the inlier ratio is less than 50%, the RANSAC algorithm performs poorly. Moreover, there is no upper-bound of the computation time (see Fig. 3.3), which sometimes leads to non-optimal solution when maximum iteration is reached.

3.4.2/ M-ESTIMATOR

The standard least-squares method minimizes $\sum_{i=1}^{n}(r_i^2)$, where r_i is the residual error of the i^{th} datum. However, such method is not stable in the presence of outliers. Such method considers the outliers having the same weights as inliers, which results in the strong distortion in parameter fitting. To robustly estimate the parameters, the M-Estimator is introduced to penalize the influence (or the weight) of the outliers.

Definition 6 : M-Estimator

A maximum-likelihood estimator (or an M-estimator) is defined as the zero of the derivative of a statistical function. Thus, the M-estimator is often a critical point of the score function [292].

Mathematically, an M-Estimator has the general form as [293]:

$$\min \sum_{i=1}^{n} \rho(r_i), \tag{3.24}$$

where $\rho(\cdot)$ is a symmetric and positive-definite function with a unique minimum at zero. Instead of solving the function directly, an iterative re-weighted least-squares scheme is applied. In each iteration, sample data are assigned with individual weights which hinge upon the residual r_i.

Let $\mathbf{x} = [x_1, x_2, \cdots, x_n]^{\mathsf{T}}$ be the parameters to be estimated. The M-Estimator of $\mathbf{x}$ based on the kernel function $\rho(r_i)$ is the solution of the following m equations:

$$\sum_{i=1}^{n} \psi(r_i) \frac{\partial r_i}{\partial p_j} = 0, \quad \text{for } j = 1, \ldots, m, \tag{3.25}$$

where ψ is the derivative $\psi = \frac{\mathrm{d}\rho(x)}{\mathrm{d}x}$ is called the influence function which measures the influence of a datum on the parameter estimate. The weight function is then defined as:

$$w(x) = \frac{\psi(x)}{x}. \tag{3.26}$$

Substitute Eq. (3.26) to Eq. (3.25), we have

$$\sum_{i=1}^{m} w(r_i) r_i \frac{\partial r_i}{\partial p_j} = 0, \quad \text{for } j = 1, \ldots, m. \tag{3.27}$$

Eq. (3.27) is equivalent to solving an iterated reweighted least-squares problem

$$\min \sum_{i=1}^{n} w(r_i^{k-1}) r_i^2, \tag{3.28}$$

where superscript $^{k-1}$ indicates the iteration number. To guarantee the robustness of the M-Estimator, two constraints should be met:

i A bounded influence function $\psi(\cdot)$.

ii The kernel function $\rho(\cdot)$ is a strict convex function which has a unique minimum.

3.4.3/ ROBUST ESTIMATION USING PRINCIPAL COMPONENT ANALYSIS

Definition 7 : Principal Component Analysis

Principal Component Analysis (PCA) refers to the problem of fitting a low- dimensional affine subspace $\mathbf{A}$ of dimension $d \ll D$ to a set of points $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ in a high-dimensional space $\mathbb{R}^D$ [28].

Statically, the classical PCA [294] was first used to estimate the principal components of a multivariate random variable $\mathbf{x}$. Given a zero-mean multivariate random variable $\mathbf{x} \in \mathbb{R}^D$, we are seeking the d principal components $\mathbf{y} \in \mathbb{R}^d$, such that

$$\begin{aligned} &y_i = \mathbf{u}_i^{\mathsf{T}}\mathbf{x} \quad \text{subject to} \quad \mathbf{u}_i^{\mathsf{T}}\mathbf{u}_i = 1 \text{ and} \\ &\mathrm{Var}(\mathrm{y}_1) \geq \mathrm{Var}(\mathrm{y}_2) \geq \cdots \geq \mathrm{Var}(\mathrm{y}_\mathrm{d}) > 0, \end{aligned} \tag{3.29}$$

where $\mathbf{y} = [y_1, y_2, \cdots, y_d]$ are the d uncorrelated linear components of $\mathbf{x}$. More specifically, to find the first principal component y_i, we seek a vector $\mathbf{u}_i^* \in \mathbb{R}^D$, such that

$$\mathbf{u}_i^* = \arg\max_{\mathbf{u}_i \in \mathbb{R}^D} \mathrm{Var}(\mathbf{u}_i^{\mathsf{T}}\mathbf{x}) \quad \text{subject to} \quad \mathbf{u}_\mathrm{i}^{\mathsf{T}}\mathbf{u}_\mathrm{i} = 1. \tag{3.30}$$

The following theorem shows that the principal components of $\mathbf{x}$ can be computed from the eigenvectors of its covariance matrix $\Sigma_\mathbf{x}$ [6].

Theorem 3.4.1 (Principal Components of a Random Variable [294]). *Assume that rank*$(\Sigma_\mathbf{x}) \geq d$*. Then the first* d *principal components of a zero-mean multivariate random variable* $\mathbf{x}$*, denoted by* y_i *for* $i = 1, 2, \ldots, d$*, are given by*

$$y_i = \mathbf{u}_i^{\mathsf{T}}\mathbf{x}, \tag{3.31}$$

where $\{\mathbf{u}_\mathbf{i}\}_{i=1}^d$ *are the* d *orthonormal eigenvectors of* $\Sigma_\mathbf{x}$ *associated with its* d *largest eigenvalues* $\{\lambda_i\}_{i=1}^d$ *in which* $\lambda_i = \mathrm{Var}(y_i)$*.*

6. The covariance matrix of $\mathbf{x}$ is defined as $\Sigma_\mathbf{x} = \mathbb{E}[\mathbf{x}\mathbf{x}^{\mathsf{T}}]$, where $\mathbb{E}[\cdot]$ stands for the expectation operation.

In theorem 3.4.1, $\{\mathbf{u}_i\}_{i=1}^{d}$ are the set of orthonormal basis of the lower-dimensional affine subspace. Each basis $\mathbf{u}_i$ has an associated eigenvalue λ_i which measures the variance of the projection of data on this basis.

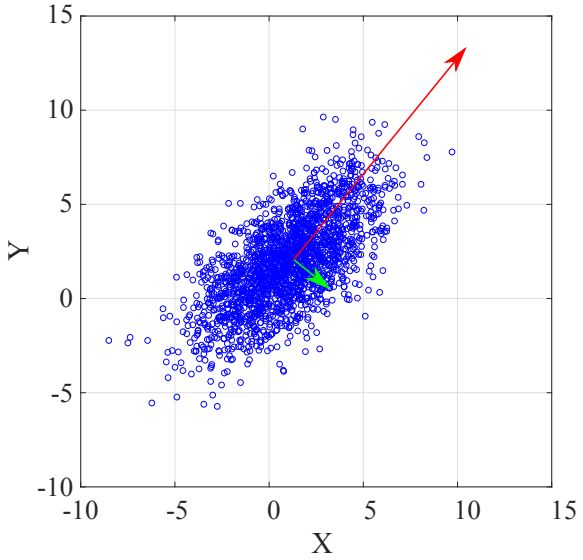


FIGURE 3.4 – principal Axis Estimation using SVD: the red and green axis are perpendicular to each other. The longer axes implies the larger associated singular value.

Geometrically, the PCA is closely related to the SVD. Given a set of points $\{\mathbf{x}_j\}_{j=1}^{N}$ in $\mathbb{R}^D$, we seek to find an affine subspace $\mathbf{S} \subset \mathbb{R}^D$ of dimension d that best fits these points. Each point $\mathbf{x}_j \in \mathbf{S}$ can be approximated as

$$\mathbf{x}_j = \mu + \mathbf{U}\mathbf{y}_j, \;\; j = 1, 2, \ldots, N, \tag{3.32}$$

where $\mu \in \mathbf{S}$ is a point in the subspace, $\mathbf{U}$ is a $D \times d$ matrix whos columns form a basis for the subspace, and $\mathbf{y}_j \in \mathbb{R}^d$ is the vector of new coordinates of $\mathbf{x}_j$ in the subspace. To solve Eq. (3.32), we can minimize the following equation

$$\min_{\mathbf{U}} \sum_{j=1}^{N} \|(\mathbf{x}_j - \mu_N) - \mathbf{U}\mathbf{U}^{\mathsf{T}}(\mathbf{x}_j - \mu_N)\|^2 \quad \text{subject to} \quad \mathbf{U}^{\mathsf{T}}\mathbf{U} = \mathbf{I}_d, \tag{3.33}$$

where μ_N is the mean of the data. Solving the above equation, $\mathbf{y}_j = \mathbf{U}^{\mathsf{T}}(\mathbf{x}_j - \mu_N)$ for $j = 1, 2, \ldots, N$ are the desired elements in the affine subspace.

Theorem 3.4.2 (PCA via SVD [294]). *Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ be the matrix formed by stacking the (zero-mean) data points as its column vectors. Let $\mathbf{X} = \mathbf{U}_\mathbf{X}\mathbf{\Sigma}_\mathbf{X}\mathbf{V}_\mathbf{X}^{\mathsf{T}}$ be the SVD of the matrix $\mathbf{X}$. Then for a given $d < D$, an optimal solution for $\mathbf{U}$ is given by the*

first d columns of $\mathbf{U}_\mathbf{X}$, an optimal solution for $\mathbf{y}_j$ is given by the j^{th} column of the top $d \times N$ submatrix of $\mathbf{\Sigma}_\mathbf{X}\mathbf{V}_\mathbf{X}^\mathsf{T}$, and the optimal objective value is given by $\sum_{i=d+1}^{D} \sigma_i^2$, where σ_i is the i^{th} singular value of $\mathbf{X}$.

Theorem 3.4.2 offers the optimal solution of problem (3.33) via SVD, which essentially leads to the same solution of problem (3.30). This equivalence contributes to the choice of PCA for dimensionality reduction, since the optimal solution can be interpreted either statistically or geometrically in different application contexts. Fig. 3.4 shows a simple application of PCA in finding the principal axis using SVD. Remarkably, the perpendicular red and green axis indicate the vectors which maximize the data distribution variance (Eq. (3.30)).

3.5/ OPTIMIZATION

This section introduces some optimization techniques involved in this thesis. For the sake of system efficiency and robustness, we formulate our problem as a Convex Optimization Problem. Accordingly, this section mainly discusses those related convex optimization skills.

3.5.1/ MATHEMATICAL OPTIMIZATION

Definition 8 : Mathematical Optimization Problem

Given the objective function $f : \mathbb{R}^n \to \mathbb{R}$ and the constraint functions $g_i : \mathbb{R}^n \to \mathbb{R}$, an optimization problem, on the variable vector $\mathbf{x} = (x_1, x_2, \cdots, x_n)$, is generally defined in the following form [295]

$$\text{minimize} \quad f(\mathbf{x}) \quad \text{subject to} \quad g_i(\mathbf{x}) \leq b_i, \quad i = 1, 2, \ldots, m. \tag{3.34}$$

Here the constraint functions are bounded by the constant set $(b_1, \cdots, b_m)$. A vector $\mathbf{x}^*$ is an optimum, or a solution of the problem 8, if it has the smallest objective value among all vectors that satisfy the constraints: for any $\mathbf{y}$ with $g_1(\mathbf{y}) \leq b_1, \cdots, g_m(\mathbf{y}) \leq b_m$, we have $f(\mathbf{y}) \geq f(\mathbf{x}^*)$. Concisely, solving an optimization problem 8 aims to find the optimal solution which has minimum cost (or maximum utility), among all candidates that meet the firm requirements.

The complexity of an optimization problem depends on many factors, such as the forms of the objective and the constraint functions, the numbers of variables and constraints, the structures of variables (like sparsity). Even when the objective and constraint functions are smooth (i.e. polynomials) the general optimization problem 8 is surprisingly difficult to solve [295]. Therefore, approaches to the general problem involve some compromise, such as very long computational time, or the possibility of not finding the optimal solution. For convex optimization problems, however, there exist very effective algorithms that can reliably solve even large problems, with hundreds or thousands of variables and constraints.

Definition 9 : Convex Optimization Problem

A convex optimization problem is a problem consisting of minimizing a convex function over a convex set. A set C is convex if the line segment between any two points in C lies in C; and a function is convex if and only if it is convex when restricted to any line that intersects its domain [295].

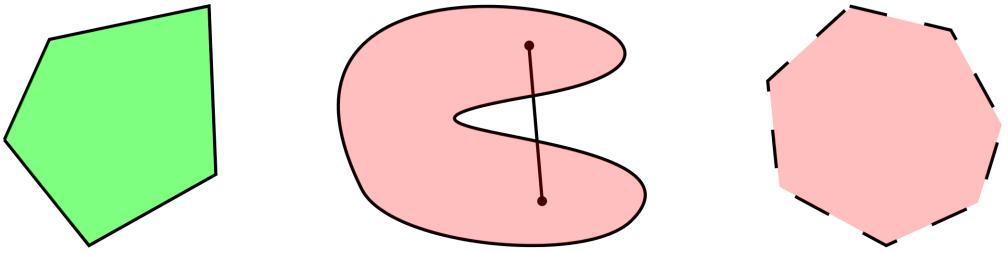


FIGURE 3.5 – Illustration of convex and non-convex sets. Left shape (including its boundary) is convex. The middle shape is non-convex because the line segment between two points in the set is not contained in the set. The right shape is non-convex because some boundary points do not belong to the set.

More formally, let C be a convex set, for any $x_1, x_2 \in C$ and $\alpha,\ \beta \in \mathbb{R}^+$ with $\alpha + \beta = 1$, we have:

$$\alpha x_1 + \beta x_2 \in C. \tag{3.35}$$

Figure 3.5 shows some simple examples of convex and non-convex sets.

Definition 9 also describes that a function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if $\mathbf{dom}\ f$ is a convex set, such that f satisfies the inequality

$$f(\alpha x + \beta y) \leq \alpha f(x) + \beta f(y) \tag{3.36}$$

for all $x,\ y \in \mathbb{R}^n$ and all $\alpha,\ \beta \in \mathbb{R}^+$ with $\alpha + \beta = 1$. In other words, a function is convex if and only if it is convex when restricted to any line that intersects its domain. Fig. 3.6 illustrates that a convex function fulfils Definition 9 while a non-convex function does not.

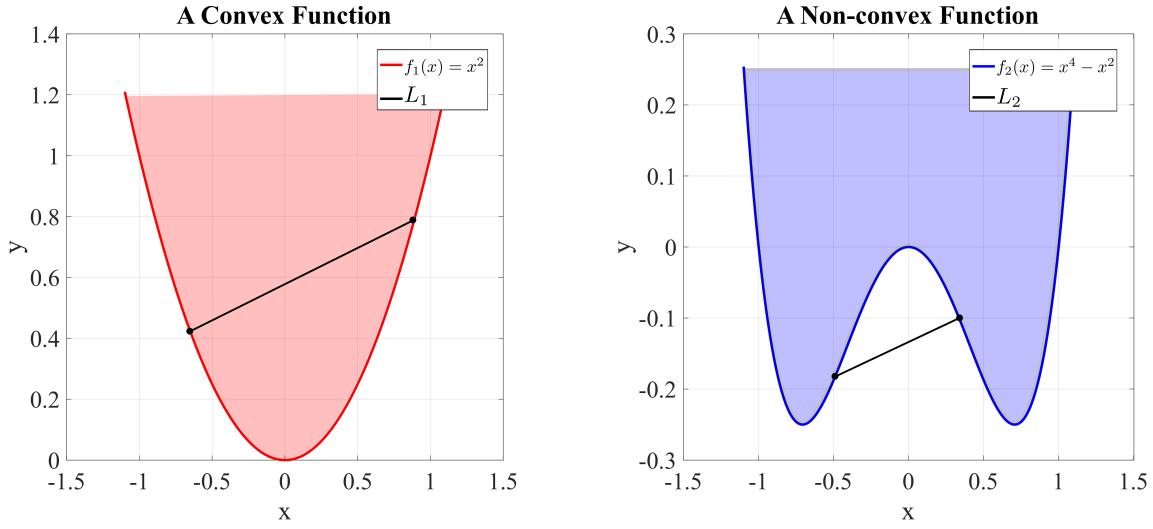


FIGURE 3.6 – Simple examples of convex and non-convex functions: shadowed areas are restricted to the functions' domain. $f_2(x)$ is not a convex function because L_2 does not intersect its domain.

The following sections introduce two well-known convex optimization problems, namely Least-Squares problems and Least-Norm Optimization problems, which are mainly used in the development of the proposed algorithms of this thesis.

3.5.2/ ℓ_p-NORM MINIMIZATION PROBLEM

The norm of a vector represents its length (or size) in a vector space, such interpretation is also applied to the norm of a matrix [7].

Definition 10 : ℓ_p-Norm General Form

Consider a vector $\mathbf{a} = [a_1, a_2, \cdots, a_m]^\mathsf{T} \in \mathbb{R}^m$ which consists of m real-valued elements, a_i. The ℓ_p-norm of $\mathbf{a}$ is defined as:

$$\|\mathbf{a}\|_p := \left(\sum_{i=1}^{m} |a_i|^p \right)^{\frac{1}{p}}, \tag{3.37}$$

where $p \in (0, \infty)$ induces the different properties of the ℓ_p-norm. When $p = 0$, strictly

7. Here, we consider the element-wise matrix norm (i.e. Frobenius norm) which is different from the induced norms. For more details refer to [296, pp. 71–73].

speaking, ℓ_0-norm is not actually a norm due to the presence of 0^{th}-root in Eq. (3.37). In *Information Theory*, more specifically in *Compressive Sensing*, a commonly used definition of ℓ_0-norm is denoted as [297]:

$$\|\mathbf{a}\|_0 := \sum_{i=1}^{m} |a_i|^0, \tag{3.38}$$

where $|a_i|^0 = 0$ for all zero entries. In other words, Eq. (3.38) implies that the ℓ_0-norm of a vector counts the number of non-zero elements. Solving Eq. (3.38) finds the sparsest solution for the under-determined linear system, where the sparse solution contains the minimum number of non-zero entries. The ℓ_0-norm optimization is widely used in many compressive sensing applications [298], i.e. the following classical optimization problem:

$$\text{minimize} \quad \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b}. \tag{3.39}$$

However, due to the extreme non-convexity of the ℓ_0-norm, solving Eq. (3.39) is an NP-hard problem [298]. Remarkably, Fig. 3.7 shows that, for $0 \leq p < 1$, the ℓ_p-norm is not a convex function. It is certain that the corresponding ℓ_p space (the domain of ℓ_p-ball) is not convex since it violate Definition 9. On the contrary, for $1 \leq p < \infty$, the ℓ_p-norm family is convex.

To efficiently solve Eq. (3.39), Donoho et al. [299] proved that the ℓ_1-norm optimization also produces a sparse solution which is a proper alternative. Thus, minimize the ℓ_0-norm can be relaxed as a ℓ_1-norm optimization problem.

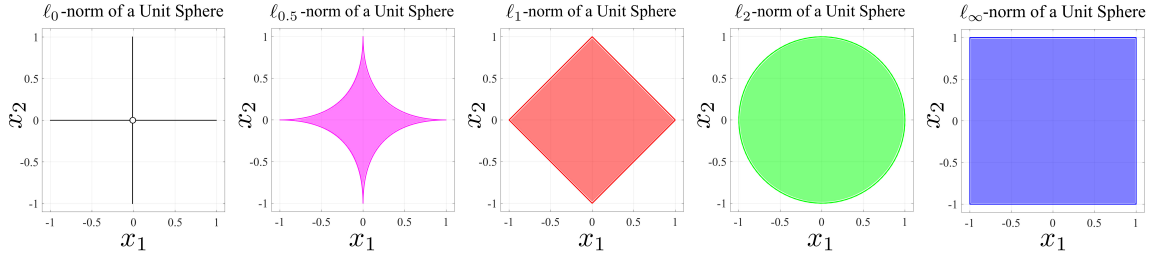


FIGURE 3.7 – ℓ_p-ball in two dimensions. As the value of p increases, the size of the corresponding ℓ_p space decreases, which can be visually observed.

Following the definition of Eq.(3.37), the ℓ_1-norm of a vector $\mathbf{a}$ is defined as:

$$\|\mathbf{a}\|_1 := \sum_{i=1}^{m} |a_i|. \tag{3.40}$$

The ℓ_1-norm, also called Manhattan norm [300], is widely used in computer vision as Sum of Absolute Difference (SAD). For instance, given two vectors $\mathbf{a} \in \mathbb{R}^m$ and $\mathbf{b} \in \mathbb{R}^m$, $SAD(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{m} |a_i - b_i|$. Fig. 3.7 shows the ℓ_1-norm of a unit sphere.

Relaxing the ℓ_0-norm minimization problem of Eq. (3.39) using ℓ_1-norm optimization, we have:

$$\text{minimize} \quad \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b}. \tag{3.41}$$

Although the ℓ_1-norm is a convex function, solving problem (3.41) is very difficult due to the non-smoothness of the function. Recent advances in convex optimization, however, are able to solve such problem efficiently using algorithms like Linear or Non-linear Programming. In this case, approximation of Eq. (3.41) can be cast as a Linear Programming (LP) problem [295]:

$$\text{minimize} \quad \mathbf{1}^{\mathsf{T}} t \quad \text{subject to} \quad -t \preceq \mathbf{A}\mathbf{x} - \mathbf{b} \preceq t, \tag{3.42}$$

where $\mathbf{1}^{\mathsf{T}}$ is the an all-ones vector, and t is the residual threshold.

The ℓ_2-norm, also known as Euclidean norm which is the most popular norm from the norm family, is defined as:

$$\|\mathbf{a}\|_2 := \sqrt{\sum_{i=1}^{m} |a_i|^2}. \tag{3.43}$$

By taking the square power of the ℓ_2-norm, it becomes the widely used Sum of Squared Difference (SSD) metric in computer vision. For example, give two vectors $\mathbf{a} \in \mathbb{R}^m$ and $\mathbf{b} \in \mathbb{R}^m$, $SSD(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{m} |a_i - b_i|^2$.

When dealing with matrix, the element-wise ℓ_2-norm, also called Frobenius norm, is defined as:

$$\begin{aligned} \|\mathbf{A}\|_F &:= \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} \\ &= \sqrt{\text{tr}(\mathbf{A}^{\mathsf{T}}\mathbf{A})} \ , \end{aligned} \tag{3.44}$$

where operator $\text{tr}(\cdot)$ denotes the trace of a matrix, i.e., the sum of its diagonal elements.

If ℓ_2-norm is applied to approximated problem 3.39, rather than taking the ℓ_1-norm ap-

proximation, it becomes the famous Least-norm optimization problem:

$$\text{minimize} \quad \|\mathbf{x}\|_2^2 \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b}. \tag{3.45}$$

Here, $\|\mathbf{x}\|_2^2 = \mathbf{x}^\mathsf{T}\mathbf{x}$ is a squared form of ℓ_2-norm. In fact, solving problem 3.45 is relatively easy thanks to the smoothness of the function. Let

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{x}^\mathsf{T}\mathbf{x} + \boldsymbol{\lambda}^\mathsf{T}(\mathbf{A}\mathbf{x} - \mathbf{y}) \tag{3.46}$$

be the Lagrange multipliers, the optimum locates at the zero-crossing point of the function's first-order derivative, which fulfils the following two conditions:

$$\nabla_\mathbf{x} L = 2\mathbf{x} + \mathbf{A}^\mathsf{T}\boldsymbol{\lambda} = 0, \tag{3.47}$$

$$\nabla_{\boldsymbol{\lambda}} L = \mathbf{A}\mathbf{x} - \mathbf{y} = 0. \tag{3.48}$$

Since condition 3.47 results in $\mathbf{x}^* = \frac{-\mathbf{A}^\mathsf{T}\boldsymbol{\lambda}}{2}$, substituting $\mathbf{x}$ in condition 3.48 leads to $\boldsymbol{\lambda} = -2(\mathbf{A}\mathbf{A}^\mathsf{T})^{-1}\mathbf{y}$. Hence, $\mathbf{x}^* = \mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{A}^\mathsf{T})^{-1}\mathbf{y}$. Note that the smooth and convex ℓ_2-norm function has a unique optimal but dense solution.

Lastly, when $p = \infty$, the ℓ_∞-norm is defined as

$$\|\mathbf{a}\|_\infty := \max\{|a_i| : i = 1, 2, \ldots, m\}. \tag{3.49}$$

The ℓ_∞-norm actually finds the maximum absolute value among all the elements of the vector.

3.5.3/ SPARSITY ANALYSIS

Definition 11 : Sparse Matrix

A sparse matrix (or a sparse vector) is a matrix in which most of the elements are zeros. By contrast, most of the elements of a dense matrix are non-zeros.

We have seen that the ℓ_p-norm optimization problem returns sparse or dense solutions by choosing the proper value of p. According to Definition 11, most of the entries of a sparse

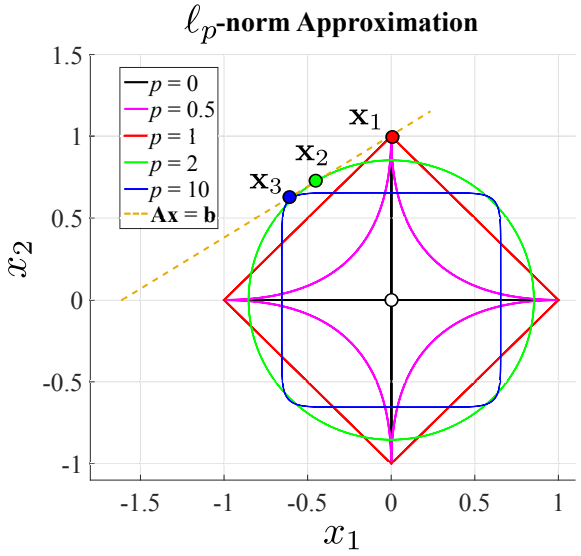


FIGURE 3.8 – Solving a linear system using ℓ_p-norm approximation: the dashed line is the desired solution to the linear system. The coloured-solid dots ($\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$) are the tangent point between the dashed line and the different ℓ_p-norm functions. The coloured-solid dots are also the solutions to the linear system using different ℓ_p-norms having the same error.

solution are zeros, which yields to a robust solution. In other words, the sparse solution suppresses most of the outliers due to the zero entries. The existence of sparse property of the ℓ_p-norm optimization can be understood by their geometric properties. Recall the classical optimization problem in solving a linear system using the ℓ_p-norm optimization:

$$\text{minimize} \quad \|\mathbf{x}\|_p \quad \text{subject to} \quad \mathbf{Ax} = \mathbf{b}. \tag{3.50}$$

Taking different p values, Fig. 3.8 sketches the boundaries of the two-dimension ℓ_p-balls where $p = 0, 0.5, 1, 2,$ and 10. In this figure, the coordinates of $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ are the solutions to the problem 3.50 using different ℓ_p-norm optimization. Clearly, $\ell_0, \ell_{0.5}, \ell_1$-norms' solution $\mathbf{x}_1$ is on the x_2-axis, which contributes to the sparsity of $\mathbf{x}_1$. However, since $\mathbf{x}_2$ and $\mathbf{x}_3$ are not on the axis, they have NO zero entry. Therefore, solution $\mathbf{x}_1$ is relatively more sparse comparing to solutions $\mathbf{x}_2$ and $\mathbf{x}_3$. Extending to a high dimensional system, optimization using $\ell_{0\leq p\leq 1}$-norm results in sparse solution while $\ell_{1<p<\infty}$-norm leads to dense solution.

4

Motion Segmentation with Unknown Camera Motion

"Motion is a powerful cue for image and scene segmentation in the human visual system."

- Philip H. S. Torr, *University of Oxford*

This chapter is dedicated to the motion segmentation problem with unknown camera motions. We seek for robust solutions that can simultaneously detect and segment the moving objects without the knowledge of the camera ego-motion. Since data from uncontrolled outdoor environments are usually noisy, the sought algorithm should be robust to noise and outliers. To this end, we proposed to use the Subspace Self-Representation (SSR)-based approaches.

As detailed in Chapter 2, we conclude from the comprehensive review of literature that the motion segmentation techniques using feature trajectories are good choices because no camera ego-motion compensation is required. Among the numerous works in literature, the Sparse Subspace Clustering (SSC) approach [29] is very promising due its robustness to noise and outliers. For the sake of robustness, a sparse solution is preferred, which can be achieved by optimization of some energy function with proper norm, e.g. the ℓ_1-norm. However, such function should be convex but not necessarily smooth, which leads to a high computational time. Therefore, the SMooth Representation (SMR) clustering approach [30], which relies on a smooth and convex energy function under the SSR framework, is a good alternative. The SMR is a quadratic cost function, such that the optimal solution can be directly obtained at the zero-crossing point of its first-order derivative.

Besides, since object motions occur in a three-dimensional world, it is more natural to directly perform the motion segmentation using 3D data. Moreover, 3D data, i.e. acquired from 3D laser scanner, are not restricted by camera projection models compared to 2D data (also say image data). Therefore, we proposed two algorithms which segment the object motions using their raw 3D feature trajectories. By extending the 2D-based Sparse Subspace Clustering (2D-SSC), we proposed a 3D-based Sparse Subspace Clustering (3D-SSC) algorithm which inherits the aforementioned merits. We also proposed a 3D-based SMooth Representation (3D-SMR) clustering algorithm which is a very efficient algorithm with comparable performances with 3D-SSC. The proposed algorithms have been validated by extensive synthetic and real data experiments.

This chapter is organized as follows. We briefly reintroduce our problem scenarios and the motivations in Section 4.1. Then, a feature tracking and matching-aided 3D trajectories construction architecture is presented in Section 4.5. The proposed 3D-SSC and 3D-SMR algorithms and their implementation details are introduced in Section 4.3 and Section 4.4, respectively. In Section 4.6, experiments with synthetic and real datasets are presented and discussed. Section 4.7 summarizes our work.

4.1/ INTRODUCTION

Recently, visual Simultaneous Localization and Mapping (vSLAM)-based autonomous robot navigation techniques have achieved great success in static environments. Yet, in dynamic scenes, the vSLAM remains a very challenging problem, mainly because the moving objects contribute to a poor localization accuracy and map artefacts. Under such circumstances, the localization estimates the camera motion by using either the features' motion consensus [301] or the weighted cost minimization [302]. The dynamic scene parts in both cases are treated as alien objects or outliers, and thus discarded. Such methods make the assumption of mostly static environments where only few moving objects exist. However, when a significant number of features belong to the dynamic scene parts, it becomes difficult to discard them, which leads to the degradation of localization accuracy [303]. Thus, precise robot navigation in dynamic environments requires the detection and the elimination of dynamic objects prior to the 3D map construction. By excluding the dynamic objects, we obtain the static map consisting of only the static scene parts, which

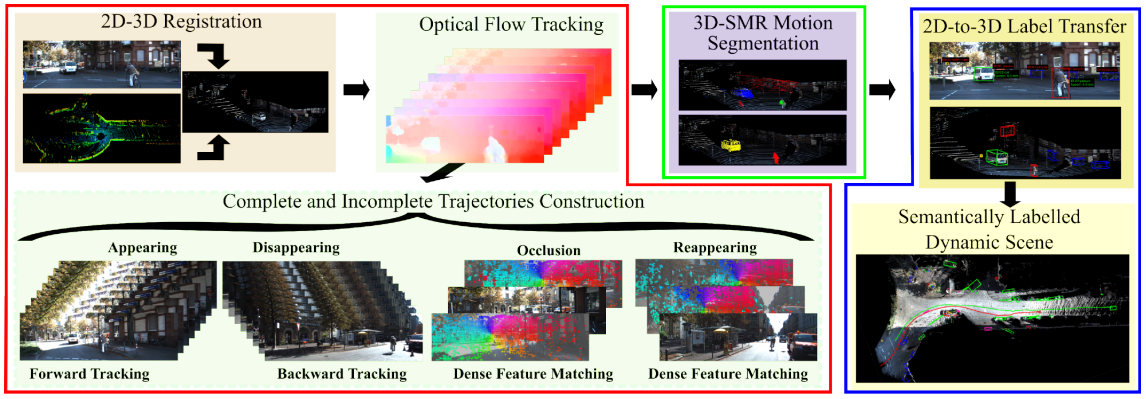


FIGURE 4.1 – Dynamic scene analysis pipeline. The red block shows the feature trajectory construction supported by forward and backward feature tracking and matching techniques, as detailed in Section 4.5. The green block depicts the moving object detection using motion segmentation on 3D feature trajectories, as detailed in Section 4.4. The blue block illustrates the 2D-to-3D label transfer for automatic semantic labelling of a dynamic scene, as detailed in Chapter 6.

in itself, is of primary interest for scene modelling [304]. It is also an important step towards scene understanding [305] and landmark-based navigation [306].

We aim to build the static map of a dynamic scene using a mobile robot equipped with a 2D-3D camera setup. Building the static map requires the categorization of the moving and the static objects. We propose the 3D-SSC and 3D-SMR motion segmentation methods that categorize the static scene parts and the multiple moving objects using their 3D motion trajectories. Our motion segmentation methods use the raw trajectory data without any projection model assumption. We also propose a complete pipeline (see Fig. 4.1) for static map building which estimates the inter-frame motion parameters by exploiting the minimal 3-Point RANSAC algorithm on the feature correspondences only from the static scene parts.

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 analyse 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. To do so, a complete pipeline for static map building is shown in Fig. 4.1 which involves three main stages: (i) 3D feature trajectories construction; (ii) feature trajectories segmentation; (iii) 3D scene registration and understanding.

When the robot is equipped with 3D sensors, it is natural to represent and segment the

features' trajectories directly in 3D space. In practice, such feature trajectories obtained by detecting and tracking the 3D feature points are very often noisy and imprecise. However, if both 2D cameras and 3D sensors are available, the 3D feature trajectories can be retrieved by tracking their corresponding 2D features. In this work, a 2D optical-flow-based method has been adopted to acquire the 2D feature trajectories which lead to the formation of 3D feature trajectories thanks to the 2D-to-3D correspondences. In many practical scenarios, many feature trajectories can be incomplete (or broken) due to the loss of tracking. To overcome this issue, we present a novel feature trajectory construction approach jointly benefiting from the feature tracking and matching techniques, as detailed in Section 4.5.

Moreover, the feature trajectories obtained using dense optical flow tracking yield numerical instabilities due to their non-uniform distribution on the static and the dynamic objects [1]. We tackle this problem by employing a Flow-Likelihood-based Sampling (FLS) technique, so that the number of trajectory samples of moving objects and static objects is balanced, making it more applicable for wider ranges of dynamic objects coverage. The FLS technique samples the features by using their median-suppressed optical flow, under the assumption that the median optical flow belongs to the scene background. A higher value implies that the feature is more different from the background flow, hence it is more likely to be originated from a moving object.

Using the 3D trajectories of sparse feature points, we propose the so-called 3D-SSC and 3D-SMR motion segmentation algorithms that categorize feature trajectories into their respecting motions. Recall that many motion segmentation methods provide some solutions for objects moving either in 2D space, and (or) in 3D space under specific camera projection model assumption. Contrastingly, the proposed methods performs motion segmentation using the raw 3D feature trajectories, which requires no projection model assumption. The 3D-SSC algorithm finds the minimal linear sparse subspaces that best represent the motion trajectories, while the 3D-SMR minimizes the subspace self-representation energy with strong regularization constraints. In this chapter, we show that both the 3D-SSC and the 3D-SMR approaches outperform their 2D-based counterparts.

1. In practical scenes, it is very likely that the static scene parts, such as walls and grounds, have larger coverage than the moving objects, such as walking pedestrians and cars.

4.2/ NOTATION AND BACKGROUND

Motion segmentation aims to determine different distinctive motions from the features' motion trajectories. 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 point cloud sequence to represent the features' motions. For K objects following distinct motions, there exist K subsets (or groups) of distinct trajectories, so called subspaces. Feature trajectories from the same subspace are linearly dependent under the rigid body motion assumption. In other words, all the feature trajectories lie in a union of K subspaces.

Let $\mathsf{X} \in \mathbb{R}^3$ and $\mathsf{Y} \in \mathbb{R}^3$ be two three-dimensional points in Cartesian coordinates. These two points are related by a rigid body motion – the rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ and the translation vector $\mathbf{t} \in \mathbb{R}^{3\times1}$, such that:

$$\mathsf{X} = \underbrace{\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}}_{\mathbf{T}\in\mathbb{R}^{3\times4}} \begin{bmatrix} \mathsf{Y} \\ 1 \end{bmatrix}, \tag{4.1}$$

where $\mathbf{T}$ represents the 3D-space rigid transformation matrix. Let $\{\mathsf{Y}_i\}_{i=1}^{P}$ represent a set of P points that belong to the k^{th} 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 $\{\mathbf{T}_j\}_{j=1}^{F}$, then all the 3D feature points X_{ji} (*i.e.* the i^{th} feature in the j^{th} frame) can be expressed as:

$$\underbrace{\begin{bmatrix} \mathsf{X}_{11} & \mathsf{X}_{12} & \cdots & \mathsf{X}_{1P} \\ \mathsf{X}_{21} & \mathsf{X}_{22} & \cdots & \mathsf{X}_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ \mathsf{X}_{F1} & \mathsf{X}_{F2} & \cdots & \mathsf{X}_{FP} \end{bmatrix}}_{\mathbf{X}_k\in\mathbb{R}^{3F\times P}} = \underbrace{\begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \\ \vdots \\ \mathbf{T}_F \end{bmatrix}}_{\mathbf{M}_k\in\mathbb{R}^{3F\times4}} \underbrace{\begin{bmatrix} \mathsf{Y}_1 & \mathsf{Y}_2 & \cdots & \mathsf{Y}_P \\ 1 & 1 & \cdots & 1 \end{bmatrix}}_{\mathbf{S}_k\in\mathbb{R}^{4\times P}}, \tag{4.2}$$

where $\mathbf{M}$ and $\mathbf{S}$ represent the motion and structure of a dynamic object, respectively. Each column, say J_i, of matrix $\mathbf{X}_k$ represents one 3D motion trajectory of a 3D feature point. Since all the entries of the last row of $\mathbf{S}$ are one, the feature trajectories of the same rigidly moving object (*i.e.* the columns of $\mathbf{X}$) lie in a subspace of $\mathbb{R}^{3F}$ of dimension at most three.

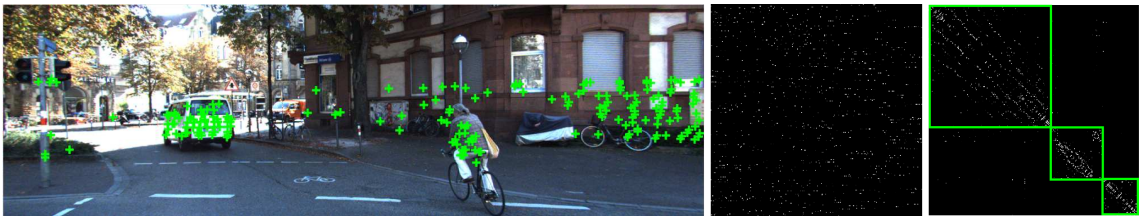


FIGURE 4.2 – 3D-SSC affinity matrix to block-diagonal matrix for motion segmentation: left image shows a set of features extracted from a scene containing three moving objects, namely the background, the van and the cyclist. Middle block shows the disorganized sparse affinity matrix constructed using Eq. 4.6, while right block is the block-diagonal matrix after spectral clustering. Each sub-block represents an independent motion.

Note that the rank of $\mathbf{S}$ can be at most 4.

In the cases of multiple motions, let $\{\mathbf{S}_k\}_{k=1}^K$ be a collection of K linear subspaces of $\mathbb{R}^{3F}$ with dimension $\{D_k\}_{k=1}^K$. If $\{\mathbf{X}_k\}_{k=1}^K$ correspond to K different unknown motions consisting of P_k trajectories, the measurement matrix, say $\mathbf{X}$, containing N measured trajectories $\mathrm{J}_1, \cdots, \mathrm{J}_N$ of F frames can be denoted as:

$$
\begin{aligned}
\mathbf{X} &= [\mathrm{J}_1, \mathrm{J}_2, \cdots, \mathrm{J}_N] \\
&= [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K] \\
&= [\mathbf{M}_1\mathbf{S}_1, \mathbf{M}_2\mathbf{S}_2, \cdots, \mathbf{M}_K\mathbf{S}_K] \\
&= [\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_K] \begin{bmatrix} \mathbf{S}_1 & & & \\ & \mathbf{S}_2 & & \\ & & \ddots & \\ & & & \mathbf{S}_K \end{bmatrix},
\end{aligned} \tag{4.3}
$$

where $\mathbf{X}_k = [\mathrm{J}_1, \cdots, \mathrm{J}_{P_k}] \in \mathbb{R}^{3F \times P_k}$ is a rank-D_k matrix of the P_k feature trajectories that lie in $\mathbf{S}_k$, with $N = \sum_{k=1}^K P_k$. In practice, these trajectory matrix $\mathbf{X}$ is randomly distributed rather than well-ordered. Therefore, the objective of motion segmentation is to classify these disordered observations into a block-diagonal matrix of K sub-blocks, where each sub-block corresponds to a distinctive motion.

Problem 4.3 is the so-called *Motion Segmentation* problem which can be modelled as an energy minimization problem under the subspace self-representation assumption. Such problem can be framed in either a constrained or unconstrained manner with different regularization terms. As inspired by Problem 3.14 in Chapter 3 where the data matrix $\mathbf{X}$ is constructed from image feature trajectories, the general SSR model for Problem 4.3 can

be defined in a similar manner as:

$$\underset{\mathbf{C}}{\text{minimize}} \quad \|\mathbf{X} - \mathcal{D}(\mathbf{X})\mathbf{C}\|_{\ell} + \Omega(\mathbf{X}, \mathbf{C}) \quad \text{subject to} \quad \zeta(\mathbf{C}), \tag{4.4}$$

where $\mathcal{D}(\mathbf{X})$ is the dictionary learned from $\mathbf{X}$, and $\|\cdot\|_{\ell}$ denotes a proper norm. $\Omega(\mathbf{X}, \mathbf{Z})$ is a regularization term and $\zeta(\mathbf{C})$ is a constraint set on $\mathbf{C}$. By solving Eq. (4.4), a desired self-representation matrix $\mathbf{C}^*$ is obtained to construct the affinity matrix. Note that the selection of the norm and regularization terms determines the property (e.g. function's smoothness and solution sparsity) of the energy function. For sparse solutions, ℓ_0-norm, ℓ_1-norm or nuclear norm can be chosen, while ℓ_2-norm and ℓ_F-norm are preferred for computational efficiency. Besides, the regularization terms are some constraints, e.g. spatial closeness constraint or motion smoothness constraint. These constraints usually yield a more sophisticated cost function but a better overall performance.

4.3/ 3D-SSC Motion Segmentation

Given a set of 3D feature trajectories from K different motions, our objective is to cluster those trajectories into K different groups where each group stands for one independent motion. We assume that these motions belong to rigidly moving objects such that the feature trajectories from the same moving objects are very similar to each other. Thus, a trajectory can be approximated by taking a linear combination of other trajectories from the same motion. More formally, each motion can be considered as a linear subspace or affine subspace where each element can be represented by other elements, so-called SSR property. To address this challenge, we formulate the 3D trajectories-based motion segmentation problem under the SSR framework with sparsity constraint on the solution set. There are two reasons for this selection: a) The SSR property allows the direct representation of data, which solves the subspace clustering problem in a more natural manner (see Fig. 4.3 for an illustration). b) The sparsity constraint yields the minimal number of elements used in the expression of the current element. In other words, the chance of having outliers in the subspace representation matrix is minimized, making the system robust to outliers.

FIGURE 4.3 – Illustration of 2D-SSC and 3D-SSC for motion segmentation: the 2D-SSC approach (left) wrongly clusters the 2D feature trajectories of the road sign into the motion of the van. On the contrary, the 3D-SSC (right) is able to correctly cluster the 3D feature trajectories of the three moving objects, namely the background, the van and the cyclist.

4.3.1/ SPARSE SUBSPACE REPRESENTATION AND RECOVERY

Referring to Equation (4.3), one can observe that the problem of 3D motion segmentation reduces to that of decomposing $\mathbf{X} = [\mathrm{J}_1, \cdots, \mathrm{J}_N]$ into K subspaces $\{\mathbf{X}_k\}_{k=1}^{K}$ and the SSR matrix C. This problem is addressed in [29] by solving a relaxed optimization problem, using the self-expressiveness property of the data. The solution is obtained under the assumption that every column J_i can be represented as a combination of other columns in $\mathbf{X}$. To make the representation least ambiguous, the combination coefficients are kept as sparse as possible. Therefore, the general SSR model for 3D feature trajectories segmentation problem 4.4 can be reformulated as:

$$\begin{aligned} \underset{\mathbf{C}}{\text{minimize}} \quad & \|\mathbf{C}\|_1 \\ \text{subject to} \quad & \mathbf{X} = \mathbf{X}\mathbf{C}, \\ & \operatorname{diag}(\mathbf{C}) = \mathbf{0}. \end{aligned} \tag{4.5}$$

where $\mathbf{C}$ is the subspace self-representation matrix that encodes the relationships between the elements. Each column of $\mathbf{C}$, say $\mathbf{c}_i$, is a sparse vector whose non-zero entries correspond to the selected elements from the same subspace. The values of those non-zero entries are the scale factors of the linear combination of the selected elements for the representation of the current element. By enforcing the constraint $\operatorname{diag}(\mathbf{C}) = \mathbf{0}$, the trivial solution of identity matrix $\mathbf{I}_N$ is avoided. Moreover, since Eq. (4.5) is a ℓ_1-norm optimization problem, a sparse solution is granted.

Although this optimization problem is solved as in [29], 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 is computationally more expensive, it helps to avoid an extra step of post-processing.

Based on our empirical evaluations over several approaches to handle noisy data (also discussed in [29] Theorem 2), optimization problem of Eq. (4.5) yields the following optimization program:

$$\begin{aligned} \underset{\mathbf{C}}{\text{minimize}} \quad & \left\| \begin{bmatrix} \mathbf{C} \\ \mathbf{C}_{-} \end{bmatrix} \right\|_1 \\ \text{subject to} \quad & \mathbf{X} = [\mathbf{X} \;\; \mathbf{I}_d] \begin{bmatrix} \mathbf{C} \\ \mathbf{C}_{-} \end{bmatrix}, \\ & \text{diag}(\mathbf{C}) = \mathbf{0}, \\ & c_{ij} \geq 0. \end{aligned} \tag{4.6}$$

where $\mathbf{I}_d$ is a $3F \times 3F$ identity matrix and c_{ij} are the entries of $\mathbf{C}$. Problem (4.6) recovers a subspace-sparse representation with the non-zero $\mathbf{C}^*$ and the supposedly zero matrix $\mathbf{C}^*_{-}$, which leads to a more restrictive model of disjoint subspace arrangement. Once the sparse representation matrix $\mathbf{C}^*$ is computed, a weighted graph $\mathbf{G}$ with weights $\mathbf{Q} = |\mathbf{C}^*| + |\mathbf{C}^*|^{\mathsf{T}}$ is built. The segmentation of trajectories into different subspaces is obtained by applying spectral clustering methods, e.g. Unnormalized Spectral Clustering Algorithm 1, Normalized Spectral Clustering Algorithm 2 or Random Walk Clustering Algorithm 3, on the Laplacian of graph $\mathbf{G}$.

4.3.2/ Implementation details

The proposed 3D-SSC algorithm is an extension of the image-based 2D-SSC algorithm and is summarized in Algorithm 5. We refer [177] for its theoretical derivations. Our system is based on the 2D-SSC [29] and CVX optimization toolbox [295], with the following critical modifications: a) A modified system with 3D-SSC; b) Non-negative constraint in sparse representation; c) Diagonal identity constraint (see Equation (4.6)) for corrupted data recovery. Although he proposed system requires 3D data acquisition, it offers the following advantages:

i Direct 3D space motion analysis: perspective projection effects produced by the affine projection assumption is avoided.

ii More precise motion behaviour analysis: the rotation and translation can be precisely recovered from the segmented 3D motion trajectories.

iii Better perception of the scene structure: because the 3D data provide more meaningful information, e.g. geometric structures, continuity or discontinuity, for better scene understanding.

Algorithm 5: 3D-SSC Motion Segmentation.

Data: 3D feature trajectories $\mathbf{X} \in \mathbb{R}^{3F \times N}$.
Result: K clustered subspaces.
1 Sparse subspace recovery using Eq. (4.6).
2 Construct similarity graph $\mathbf{G}$ with $\mathbf{Q} = |\mathbf{C}| + |\mathbf{C}|^{\mathsf{T}}$.
3 Spectral clustering on $\mathbf{Q}$ using Algorithms 1 2 3.

4.4/ 3D-SMR MOTION SEGMENTATION

Motion segmentation using 3D-SSC is very computationally expensive and not easy to scale to large problems (e.g. more than 1000 feature trajectories). Thus, we seek an alternative solution with comparable performances and a much better efficiency. As inspired by [30], we propose a second 3D-based motion segmentation algorithm using SMooth Representation clustering, so-called 3D-SMR. Different from [30], our method performs the MS in 3D space using raw motion trajectories, with spatial regularization constraints (linear and angular motion consistency energy) to improve the trajectory clustering performances. Alongside with the subspace self-expressive property, the 3D-SMR algorithm intends to separate the motion subspaces by enforcing the grouping effects as well as the motion consistency of their subspaces. Remarkably, our Grouping Effect (GE) constraint describes the feature trajectories' closeness (distances) in the 3D Euclidean space. Doing so, 3D-SMR avoids the perspective effects that appear on the image measurements. The GE constraint is enforced as a regularization term:

$$\begin{aligned} \Omega(\mathbf{X}, \mathbf{C}) &= \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \|\mathsf{C}_i - \mathsf{C}_j\|_2^2 \\ &= \mathrm{tr}(\mathbf{C}\mathbf{L}\mathbf{C}^{\mathsf{T}}), \end{aligned} \tag{4.7}$$

where $\mathbf{C} = [\mathsf{C}_i, \cdots, \mathsf{C}_N]$ is the $P \times P$ square-sized self-representation matrix. $W = (w_{ij})$ with $w_{ij} = \|\mathsf{J}_i - \mathsf{J}_j\|_2^2$ is the weight matrix defined by the spatial closeness (e.g. the Euclidean distance) of feature trajectories, and $\mathbf{L}$ is the Laplacian matrix. To construct the weight matrix W, a $0-1$ weighted k-Nearest Neighbour (kNN) graph is used. Combining Eq. (4.4) and Eq. (4.7), the 3D-SMR model is obtained:

$$\underset{\mathbf{C}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2 + \text{tr}(\mathbf{C}\mathbf{L}\mathbf{C}^\mathsf{T}), \tag{4.8}$$

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

4.4.1/ Motion Consistency Constraints

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

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

$$\begin{aligned} \Omega(\mathbf{X}, \mathbf{V}, \boldsymbol{\theta}, \mathbf{C}) &= \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \widetilde{w}_{ij} \|\mathsf{C}_i - \mathsf{C}_j\|_2^2 \\ &= \text{tr}(\mathbf{C}\widetilde{\mathbf{L}}\mathbf{C}^\mathsf{T}), \end{aligned} \tag{4.9}$$

where

$$\widetilde{w}_{ij} = \mathcal{E}(\mathsf{J}_i, \mathsf{J}_j) + \varphi(\mathsf{V}_i, \mathsf{V}_j) + \psi(\vartheta_i, \vartheta_j). \tag{4.10}$$

Recall Eq. (4.7), the weight component $\mathcal{E}(\mathsf{J}_i, \mathsf{J}_j) = \|\mathsf{J}_i - \mathsf{J}_j\|_2^2$ describes the spatial closeness of the feature trajectories. $\varphi(\mathsf{V}_i, \mathsf{V}_j) = \alpha\|\bar{v}_i - \bar{v}_j\|_2^2$ measures the consistency of the motion velocity, where $\bar{v}_i$ and $\bar{v}_j$ are the median speeds of the feature trajectories V_i and V_j in 3D space. $\psi(\vartheta_i, \vartheta_j) = \beta\text{atan2}(\vartheta_i \times \vartheta_j, \vartheta_i \cdot \vartheta_j)$ computes the directional difference between the feature trajectories, where $\text{atan2}(\cdot)$ function calculates the angle between the motion vectors ϑ_i and ϑ_j within the appropriate quadrant. α and β are the constant values controlling the weights of the regularization terms.

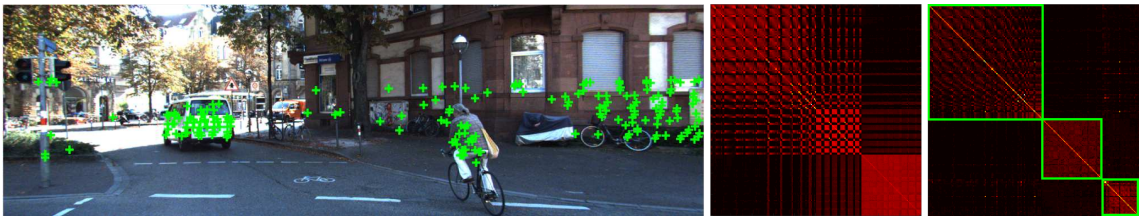


FIGURE 4.4 – 3D-SMR affinity matrix to block-diagonal matrix for motion segmentation: left image shows a set of features extracted from a scene containing three moving objects, namely the background, the van and the cyclist. Middle block shows the disorganized semi-dense affinity matrix (compared to Fig. 4.2) constructed using Eq. 4.13, while right block is the block-diagonal matrix after spectral clustering. Each sub-block represents an independent motion.

The Laplacian matrix in Eq. (4.9) can be written as: $\widetilde{\mathbf{L}} = \widetilde{\mathbf{D}} - \widetilde{\mathbf{W}}$, where $\widetilde{d}_{ii} = \sum_{i=1}^{N} \widetilde{w}_{ij}$ and the weight function $\widetilde{\mathbf{W}} = (\widetilde{w}_{ij})$. Replacing the regularization term of Eq. (4.8) with Eq. (4.9), a more practical 3D-SMR model is proposed as:

$$\underset{\mathbf{C}}{\text{minimize}} \quad \alpha\|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2 + \text{tr}(\mathbf{C}\widetilde{\mathbf{L}}\mathbf{C}^\top). \tag{4.11}$$

Since solving Eq. (4.11) is a smooth and convex problem, the desired optimal solution $\mathbf{C}^*$ can be obtained by taking the first order derivative, such that:

$$\mathbf{X}^\top\mathbf{X}\mathbf{C}^* + \mathbf{C}^*\mathbf{L} = \mathbf{X}^\top\mathbf{X}. \tag{4.12}$$

Equation (4.12) is a Sylvester equation [307] having a unique optimal solution which can be solved efficiently by the Bartels-Stewart algorithm [307] with computational complexity of $O(n^3)$.

Following [29] and [30], we employ two different affinity matrices which are defined as:

$$\mathbf{Q}_1 = |\mathbf{C}^*| + |\mathbf{C}^*|^\top \tag{4.13}$$

and

$$\mathbf{Q}_2 = \left(\left| \frac{\lambda \mathsf{C}_i^{*\top} \mathsf{C}_j^*}{\|\mathsf{X}_i\|_2 \|\mathsf{X}_j\|_2} \right|^\gamma \right), \tag{4.14}$$

where $\lambda > 0$ and $\gamma > 0$ are scale factors to control the affinity variances. The effectiveness of both affinity matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ benefits from the block-diagonal property of $\mathbf{C}^*$. Moreover, $\mathbf{Q}_2$ measures the inner product of the normalized trajectories by the norms of

their original features. Such normalization reduces the bias arose by the variation of the features' amplitudes [30]. According to our experiments, $\gamma = 1$ generally yields accurate motion segmentation results, while $\lambda = max(\|\mathrm{X}_i\|_2 \|\mathrm{X}_j\|_2)$ is set to avoid the numerical instability. Finally, a spectral clustering algorithm is applied to the affinity matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ to segment the feature trajectories into their corresponding motions, see Fig. 4.4. These procedurals are summarized in Algorithm 6.

Algorithm 6: 3D-SMR Motion Segmentation.

Data: 3D feature trajectories $\mathbf{X} \in \mathbb{R}^{3F \times N}$.
Result: K clustered subspaces.
1 Smooth representation-based affinity matrix construction using Eq. (4.12).
2 Construct similarity graph $\mathbf{G}$ with $\mathbf{Q}_1$ using Eq. (4.13) or $\mathbf{Q}_2$ using Eq. (4.14).
3 Spectral clustering on $\mathbf{Q}_1$ or $\mathbf{Q}_2$ using Algorithms 1 2 3.

4.4.2/ DISCUSSION

In essence, both the 3D-SSC and the 3D-SMR algorithms are based on the subspace self-representation theory. Such approaches do not requires the prior knowledge of camera ego-motion and object information. The 3D-SSC finds a sparse solution which guarantees its robustness to noise. Although the optimization problem of 3D-SSC can be efficiently (compared to the brute-force search) solved by using the Alternating Direction Method of Multipliers (ADMM) [308] algorithm, obtaining the global optimal solution is not guaranteed. In another way, the 3D-SMR seeks a direct optimal and dense solution which is more sensitive to noise. To improve its robustness, additional regularization terms, e.g. the spatial closeness constraint, are required. Such regularization terms are more meaningful for 3D points than image pixels. Notably, the dense solution leads to the connections between different subspaces, which makes the spectral clustering problem more difficult. In particular, for the subspaces having very few elements, the spectral cluster performance can be highly degraded. Therefore, the 3D-SMR requires relatively more feature trajectories in each motion subspace compared to the 3D-SSC.

4.5/ FEATURE TRAJECTORY CONSTRUCTION

Prior to motion segmentation, the feature trajectories are acquired by feature tracking across multiple consecutive frames. We use both 2D and 3D measurements to construct the feature trajectories in 3D space. For a calibrated 2D-3D camera setup [2] the 3D scene points are projected onto the 2D image to establish the 2D-to-3D correspondences. More formally, let an image point $\mathbf{x} = [x, y]^\mathsf{T}$ and a 3D point $\mathrm{X} = [X, Y, Z]^\mathsf{T}$ be a correspondence pair, denoted as $\mathbf{x} \leftrightarrow \mathrm{X}$, we have:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\mathbf{K}\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}}_{\mathbf{P}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{4.15}$$

where $\mathbf{K} \in \mathbb{R}^{3\times3}$, $\mathbf{R} \in \mathbb{R}^{3\times3}$, and $\mathbf{t} \in \mathbb{R}^3$ are the intrinsic parameter matrix, the rotation matrix, and the translation matrix, respectively. $\mathbf{P} \in \mathbb{R}^{3\times4}$ is the so-called projection matrix which encodes the camera parameters. All these matrices are assumed to be known and correctly estimated after camera calibration. Therefore, 3D-to-2D correspondences can be built efficiently by applying Eq. (4.15).

Initially, the 3D points (i.e. acquired by a 3D laser scanner) are projected onto the image space, and all these projections are considered as 2D feature points and tracked across the sequence using a dense optical flow method. Note that there exists no one-to-one correspondences between the 3D points and the 2D image pixels due to their differences in data density and cameras' field-of-view. Therefore, the 2D-to-3D correspondences are established only for the overlapping field-of-view areas. To cover a wide range of speeds, a large displacement dense optical flow [309] tracking algorithm has been adopted. To reject the incorrectly tracked features, we utilize the forward and backward validation of optical flow tracking, similar to [310]. The 3D feature trajectories are then retrieved thanks to the 2D-to-3D correspondences.

Practically, some feature trajectories can be incomplete due to occlusions or loss of feature tracking. Most of the literature approaches [177, 195, 219] simply discard such incom-

2. A calibrated 2D-3D camera setup means that both the cameras' intrinsic parameters and the relative poses between the 2D and 3D cameras are known.

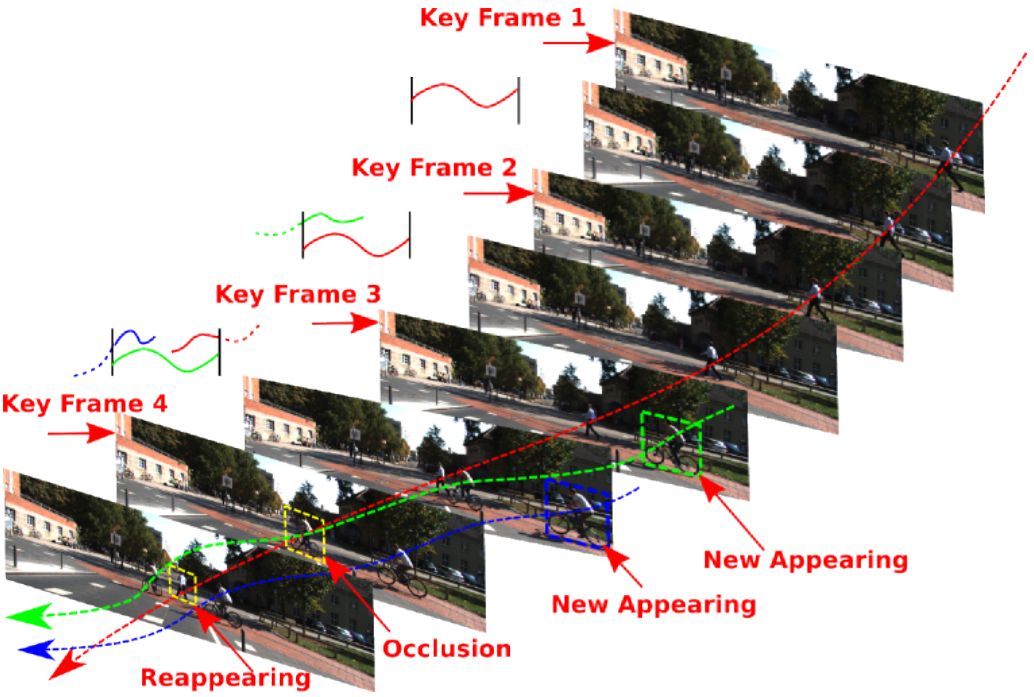


FIGURE 4.5 – Incomplete feature trajectories construction: the red, green and blue dashed lines represent the trajectories of the pedestrian and two cyclists, respectively. The green and blue rectangles highlight the appearing of the two cyclists, while the yellow rectangles highlight the pedestrian being occluded and reappearing. The solid lines (on top) represent the feature trajectories within two key frames, while the connected dashed lines are the forward or backward extended trajectories.

plete feature trajectories leaving some potential moving objects undiscovered. To address this problem, we define a trajectory to be complete if its feature is detected and tracked throughout the frames of interest (i.e. between two key frames), whereas, an incomplete trajectory is only partially detected and tracked between two key frames. The incomplete trajectories mainly come from the failure of feature tracking due to occlusions or object disappearances. Therefore, we propose the following simple but effective incomplete feature trajectory completion approach.

4.5.1/ FEATURE TRAJECTORY CONNECTION

Let $\mathrm{J} = [\mathrm{X}_1, \cdots, \mathrm{X}_F]^\mathsf{T} \in \mathbb{R}^{3F}$ be a complete 3D feature trajectory vector of F frames, and $\mathbf{J} = \{\mathrm{J}_i\}_{i=1}^{P} \in \mathbb{R}^{3F\times P}$ is the combination of P complete feature trajectories. Denote $\hat{\mathrm{J}} = [\mathrm{X}_1, \cdots, \mathrm{X}_{\hat{F}}]^\mathsf{T} \in \mathbb{R}^{3\hat{F}}$ as an incomplete feature trajectory of $\hat{F}$ frames ($\hat{F} < F$), and $\hat{\mathbf{J}} = \{\hat{\mathrm{J}}_i\}_{i=1}^{\hat{P}} \in \mathbb{R}^{3\hat{F}\times\hat{P}}$ as the collection of $\hat{P}$ incomplete feature trajectories. Since trajectories of $\mathbf{J}$ and $\hat{\mathbf{J}}$ have different number of elements, motion segmentation cannot be performed

altogether. In other words, the row dimensions (trajectory length) of $\mathbf{J}$ and $\hat{\mathbf{J}}$ must be the same, while their column dimensions (feature numbers) are unconstrained. Thus, the incomplete trajectories are required to be extended so that the length of $\hat{\mathrm{J}}$ is $3F$ (same as J), and the size of $\hat{\mathbf{J}}$ is $3F \times \hat{P}$ accordingly.

In practice, the incomplete feature trajectories mainly come from four different scenarios: new object appearance (denoted as $+$), tracked object disappearance (denoted as $-$), object going under occlusion (denoted as o), and previous object reappearance (denoted as $++$) as follows:

- **i** *Newly appearing objects* are detected if new features are tracked through a minimum number of required frames for motion analysis.
- **ii** *Disappearing tracked objects* are detected using a feature tracking failure detection method [310].
- **iii** *Objects under occlusion* refer to a partial occlusion, where the object's features have both complete and incomplete trajectories.
- **iv** *Reappearing objects* are detected using the Deep-matching [311] between the features in key frames.

If a feature is untracked throughout two key frames, a forward or backward tracking is activated, which yields to the extended incomplete trajectory having the same dimension as a complete trajectory, denoted as $\dim(\hat{\mathrm{J}}) = \dim(\mathrm{J})$. A forward feature tracking implies that the feature is tracked from frame t to frame $t+1$. On the contrary, the feature is tracked from frame t to frame $t-1$ is backward feature tracking. The forward/backward feature tracking is carried out until the extended incomplete feature trajectory has the same length as the complete trajectories.

The feature trajectory completion algorithm is illustrated in Figure 4.5. In this figure, there are only two moving objects (a walking pedestrian and the background) between key frames 1 and 2, and two new objects (the cyclists) appear in between key frames 3 and 4. Accordingly, the feature trajectories on the moving objects between key frames 1 and 2 are complete, while incomplete trajectories occur due to the newly appearing objects or occlusions between key frames 3 and 4.

Figure 4.6 shows the constructed complete and incomplete trajectories with MS results. In this figure, the walking pedestrian was completely occluded by the passing cyclist,

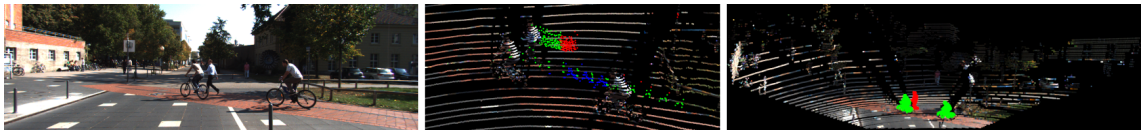


FIGURE 4.6 – Feature trajectories' completion for MS: left image shows the cyclist crossing the walking pedestrian. The green trajectories in the middle image are tracked features between two key frames, while the red and blue trajectories are acquired from backward and forward feature tracking, respectively. The right image shows the MS results.

leading to incomplete trajectories of the pedestrians. Thus, the backward feature tracking is activated to extend the incomplete trajectories, see the red trajectories of the middle image. Besides, the newly appearing cyclist requires a forward feature tracking to extend the incomplete trajectories, see the blue trajectories of the middle image. Doing so, both the complete and extended incomplete trajectories are now represented with vectors of same lengths, which allows the MS to overcome the loss of feature tracking. Although the incomplete trajectories are extended by simple and direct extrapolation, the incomplete feature trajectory construction offers the following advantages: (a) The lost tracked objects are rediscovered and re-tracked. (b). The simultaneous motion segmentation on complete and incomplete feature trajectories now becomes possible.

4.5.2/ FEATURE TRAJECTORY SAMPLING

Our primary interest is to perform robust motion segmentation while addressing a wide range of the dynamic coverages and speeds. We define dynamic coverage as the area that the dynamic objects cover in an image. 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 belongs 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 $\{\mathbf{v}_i\}_{i=1}^{N}$ be the measured optical flow speeds corresponding to the trajectories $\{\mathrm{J}_i\}_{i=1}^{N}$ with $N = P + \hat{P}$. Let $\{\mathcal{L}_i\}_{i=1}^{N}$ be the flow likelihood which is assigned to each trajectory, the likelihood function is defined as:

$$\mathcal{L}(\mathrm{J}_i|\mathbf{X}) \propto e^{\|\mathbf{v}_i-\bar{\mathbf{v}}\|^2/\sigma^2}, \tag{4.16}$$

where $\bar{\mathbf{v}}$ and σ are the median flow speed and standard deviation of the starting image,

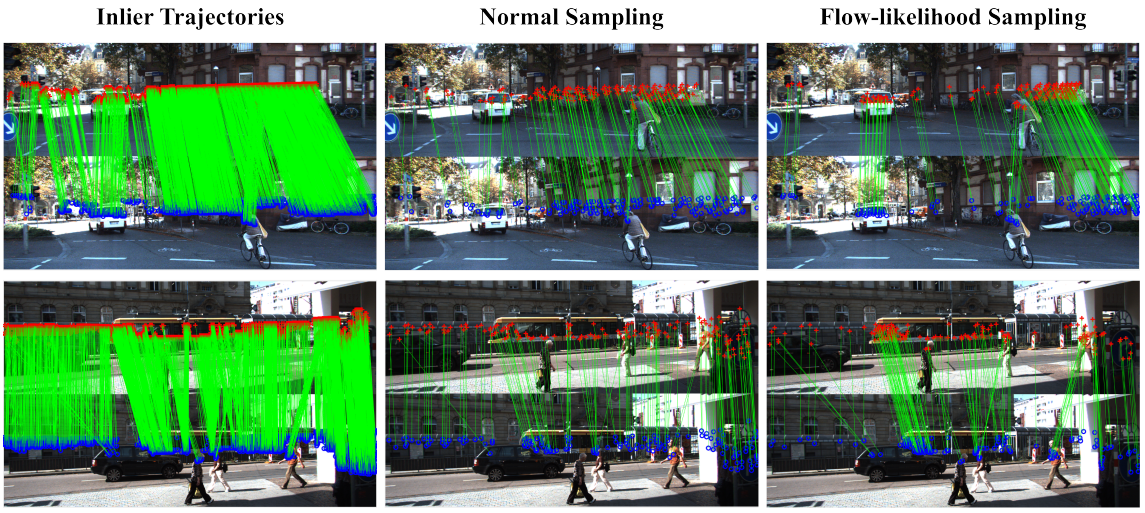


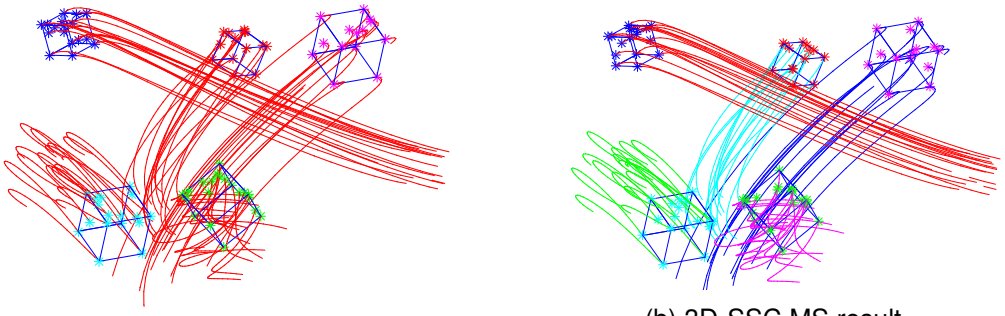
FIGURE 4.7 – Results of uniform sampling vs. the proposed flow-likelihood-based sampling: the green lines show the tracked features from the first frame to the last frame. The last column shows that more features are sampled from moving objects.

respectively. A subset of feature trajectories for motion segmentation is selected based on the likelihood measure of Equation (4.16). In fact, if a feature has very distinctive optical flow speed compared to the median flow (a background flow), it is more likely to be from a moving object. This sampling avoids the problem of having too many samples from the background, hence we balance the data for the optimization problem of Equation (4.4). During this process, we also reject all the trajectories that do not follow the smooth motion. Fig. 4.7 exemplifies the proposed flow-likelihood-based sampling approach, as we can observe that 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. 4.7).

4.6/ EXPERIMENTS

We conducted comprehensive experiments with both synthetic and real data. For synthetic experiments, we simulate different number of independently moving objects with different noise level to study the robustness of the algorithms. For real data evaluation, we rely on the benchmark KITTI dataset [1] which contains a large amount of data from real-world outdoor environments. We recall that the 2D-3D camera system of KITTI dataset are fully calibrated. Our experiments show the feasibility of the proposed 3D-SSC and 3D-SMR 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) Initial uncategorised motions.

(b) 3D-SSC MS result.

FIGURE 4.8 – 3D-SSC MS on synthetic 3D data: (a) randomly generated 5 rigid motions with uncategorised trajectories. (b) 3D-SSC segmented motions are labeled with different colors.

4.6.1/ SYNTHETIC DATA

We build a system that contains multiple moving objects under different noise conditions. More specifically, a set of synthetic data is generated with K moving cubes with different sizes, positions, orientations, and motion behaviours. The motion feature trajectories are randomly selected to generalize the algorithm evaluation. To quantify the robustness of the algorithm under different Gaussian noise levels, the misclassification rate index used is defined as

$$\eta = \frac{\text{number of misclassified features}}{\text{total number of features}}. \tag{4.17}$$

To test the performance of the algorithm under different noise levels and number of motions, various levels of white Gaussian noise are added to the feature trajectories. The noise level ς is defined as

$$\varsigma = \frac{\text{nosie amplitude}}{\text{signal amplitude}}, \tag{4.18}$$

where the signal amplitude is the maximum distance among the features from the same cube. Fig. 4.9 shows that the 3D-SSC behaves very robustly under 12% of noise for at least up to 10 moving objects. Similarly, the 3D-SMR achieves very robust performances even for 16% noise level. However, the 3D-SMR algorithm is relatively less robust than the 3D-SSC when the noise level is lower than 12%. In practical scenarios, data obtain

from uncontrolled outdoor environments are relatively noisy, the 3D-SMR algorithm might be more appropriate.

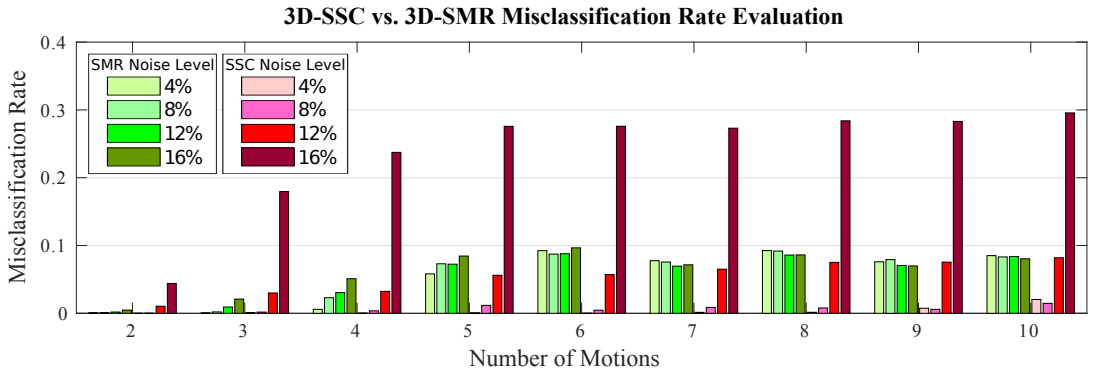


FIGURE 4.9 – Averaged motion segmentation performances of 3D-SSC and 3D-SMR on synthetic data over 50 tests.

4.6.2/ EVALUATION ON KITTI DATASET

To evaluate our system with realistic outdoor scenes, we conduct extensive experiments on the KITTI dataset. The experiments are conducted with seven different datasets, namely Highway, Junction, Station, and Market. These datasets have been selected to cover a wide range of moving objects in terms of quantity, size, speed, shape, occlusion, etc. More specifically, the 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 datasets). The selected sequences have rather demonstrated the effectiveness and the generality of the proposed methods. The details of the evaluation datasets are provided in Tables 4.1 and 4.2. Table 4.1 shows the detailed evaluation of Pedestrian sequence where every 10 frames were considered as one subsequence. Note that all the seven sequences were evaluated in the same way as Table 4.1 and summarized as one row in Table 4.2. 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.

Trajectory Construction Evaluation: The feature trajectories are constructed using the dense optical flow tracking approach and sub-sampled based on the flow-likelihood sam-

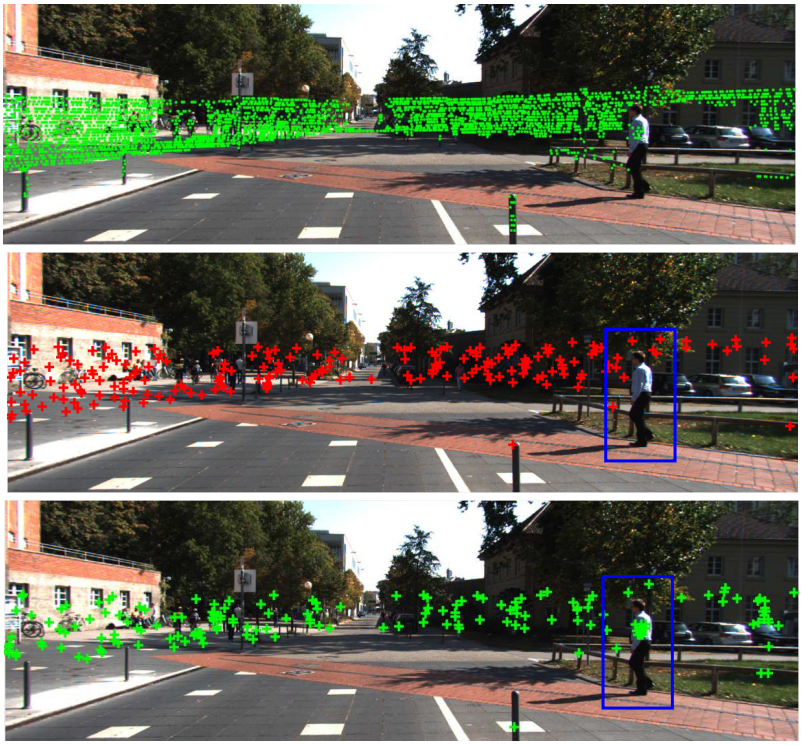


FIGURE 4.10 – Feature trajectory sampling qualitative evaluation: from top to bottom are the densely tracked features over 15 frames, 200 features after uniform sampling, and 200 features after applying FLS, respectively. The blue bounding boxes highlight that: there is no feature sampled from the walking pedestrian using the uniform sampling, while the FLS is able to sample features from the walking pedestrian.

pling technique. In this regard, we aim to have balanced trajectory samples from both the background and the foreground objects. To quantify, column 3 and 4 of Table 4.1 show the number of feature trajectories from the moving objects and the background, respectively. There are two major remarks: first, the more moving objects involve, the more feature trajectories of moving objects are sampled. Secondly, the averaged ratio of foreground trajectories over background trajectories is around 87%, which implies that the background trajectory samples are slightly more numerous than those from the foreground. Such feature distribution helps to balance the data for the subspace representation, thanks to the likelihood-based sampling.

Some qualitative results can be seen from Fig. 4.7 where a significant number of features belong to the dynamic parts, although they cover relatively small regions. Moreover, in Fig. 4.10, it is clear that features belonging to background parts are far more than the features belonging to the moving objects. Thus, when a uniform sampling approach is applied, the features from moving objects, e.g. the walking pedestrian within the blue

bounding box, are barely sampled. In contract, the proposed FLS approach is able to sample most of the features from the moving objects.

Tables 4.1 and 4.2 also show the percentages of feature trajectories which are recovered by the proposed trajectory completion approach. Those incomplete feature trajectories mainly come from the loss of feature tracking due to the different issues discussed in Section 4.5.1. Specifically, columns *Mot. State* show the changes of motion status within the sub-sequences (interval of every 10 frames), where symbols $+, -, \mathrm{o}, ++$ denote the four different motion scenarios, namely new appearing, disappearing, occlusion, and reappearing. These scenarios lead to the existence of incomplete trajectories and thus the loss of tracked objects. Thanks to the proposed incomplete trajectory construction approach, these problems can be mostly addressed and moving objects can be more sensitively detected and segmented. Accordingly, our complete and incomplete trajectory construction architecture presented in Section 4.5 is essential to address such tracking failures.

Motion Segmentation Quantification: We compare different state-of-the-art methods in the evaluation of moving object detection and segmentation using the representative seven datasets. These approaches are 2D-SSC [29], 2D-SMR-Q1 [30], 2D-SMR-Q2 [30], Object Scene Flow (OSF) [256], our 3D-SSC and 3D-SMR [3]. The evaluations of their performances are summarized in Tables 4.1 and 4.2. The segmentation performances are assessed using the popular *Sensitivity* and *Specificity* metrics as in [29]. The sensitivity and specificity metrics are respectively defined as

$$\text{Sensitivity} = \frac{\text{number of true positives}}{\text{number of true positives} \ + \ \text{number of false negatives}} , \tag{4.19}$$

and

$$\text{Specificity} = \frac{\text{number of true negatives}}{\text{number of true negatives} \ + \ \text{number of false positives}} . \tag{4.20}$$

The quantitative results of Tables 4.1 and 4.2 show the effectiveness of the proposed

3. In this thesis, the performances of the compared algorithms are achieved by using the codes provided by the authors with suggested parameter settings. In the spectral clustering procedural, the K clusters is predefined to obtain the most favourable performances of the algorithms. The source codes can be downloaded from the following links: 2D-SSC http://www.vision.jhu.edu/code/; 2D-SMR https://sites.google.com/site/hanhushomepage/pyu; OSF http://www.cvlibs.net/projects/objectsceneflow/

Sub-seq.	# Mot.	# Feat.		η	Mot. State				3D-SSC		2D-SMR-Q_1		2D-SMR-Q_2		3D-SMR-Q_1		3D-SMR-Q_2	
		# Dyn.	# Stat.		+	-	o	++	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
1	3	113	219	0.10	x	x	x	x	0.991	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2	3	115	230	0.13	x	x	x	x	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3	3	122	246	0.18	x	x	x	x	1.000	0.988	1.000	0.996	1.000	1.000	0.992	1.000	1.000	0.996
4	3	78	251	0.07	x	x	x	x	0.603	0.996	0.962	0.769	0.962	0.765	0.974	0.777	0.987	0.996
5	4	54	270	0.15	v	x	v	x	0.593	0.685	0.611	0.696	0.685	0.689	0.667	0.704	1.000	0.993
6	4	82	271	0.27	v	x	x	x	0.817	0.727	1.000	0.815	1.000	0.815	1.000	0.838	1.000	0.993
7	7	237	173	0.23	x	x	x	x	0.873	0.983	1.000	0.526	1.000	0.526	1.000	0.711	1.000	0.838
8	7	255	156	0.20	x	x	v	x	0.973	0.974	1.000	0.532	1.000	0.526	1.000	0.904	1.000	0.929
9	7	225	166	0.22	x	x	v	v	0.964	0.994	0.996	0.801	0.996	0.801	0.987	0.982	0.996	0.994
10	8	206	167	0.19	v	x	x	x	0.956	0.820	0.961	0.497	0.961	0.503	0.995	0.976	1.000	0.994
11	9	236	141	0.20	v	x	v	v	0.932	0.986	1.000	0.532	1.000	0.532	1.000	0.745	0.996	0.979
12	9	247	139	0.22	x	x	x	x	0.973	0.971	0.976	0.921	0.976	0.921	0.968	0.950	0.968	0.906
13	9	200	169	0.19	v	v	x	x	0.810	0.781	1.000	0.793	1.000	0.793	1.000	0.817	1.000	0.988
14	8	233	175	0.26	x	x	v	v	1	0.983	0.906	0.691	0.906	0.691	1.000	0.857	1.000	0.926
Average	6	172	198	0.18	/	/	/	/	0.892	0.921	0.956	0.755	0.963	0.754	0.970	0.876	**0.996**	**0.967**
Time(s)	/	/	/	/	/	/	/	/	35.122		0.054		**0.0378**		0.613		0.608	

TABLE 4.1 – Performance quantification on Pedestrian dataset. Columns |Sub-seq.|, |# Mot.|, and |# Feat.| show the sub-sequences index, moving objects number, dynamic features number, and static features number, respectively. $\eta = \frac{\text{\# incomplete trajectories}}{\text{\# total features}}$ represents the percentage of extended incomplete trajectories. Columns |Mot. State| show the motion states with symbols $+, -, \text{o}, ++$ denoting new appearance, disappearance, occlusion, and reappearance scenarios discussed in Section 4.5. Symbols v and x mean that the motion states occur or do NOT occur, respectively. The last columns compare the Sensitivity and Specificity of algorithms 3D-SSC, 2D-SMR [30] and the proposed 3D-SMR.

Sequence	# Frms.	# Objs.	# Feats.	η	OSF		3D-SSC		2D-SMR-Q_1		2D-SMR-Q_2		3D-SMR-Q_1		3D-SMR-Q_2	
					Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
Campus	60	4	341	0.23	0.404	0.988	0.920	0.888	0.944	0.658	0.947	0.621	**0.987**	0.816	0.970	**0.997**
Highway	50	2	392	0.24	0.579	0.994	**0.978**	0.625	0.609	0.963	0.613	0.962	0.825	**0.999**	0.962	0.995
Junction	90	3	416	0.24	0.613	0.966	0.892	0.943	0.968	0.998	0.971	0.997	0.973	**0.999**	**0.976**	0.997
Market	100	6	402	0.25	0.506	0.962	0.882	**0.852**	0.933	0.640	0.920	0.637	0.959	0.747	**0.989**	0.817
Pedestrian	140	6	370	0.18	0.519	**0.983**	0.892	0.921	0.956	0.755	0.963	0.754	0.970	0.876	**0.996**	0.967
Red Light	120	4	371	0.19	0.578	**0.987**	0.868	0.830	0.880	0.633	0.886	0.682	0.964	0.906	**0.998**	0.976
Station	50	5	417	0.28	0.164	**0.996**	0.901	0.631	0.942	0.486	**0.958**	0.437	0.918	0.703	0.929	0.875
Average	87	5	387	0.23	0.480	**0.982**	0.905	0.813	0.890	0.733	0.894	0.727	0.942	0.864	**0.974**	0.946

TABLE 4.2 – Quantification of OSF [256], 3D-SSC, 2D-SMR [30] and our 3D-SMR in motion segmentation: this table is a summary of performances of the 7 representative datasets, and the dataset notations are detailed in Table 4.1.

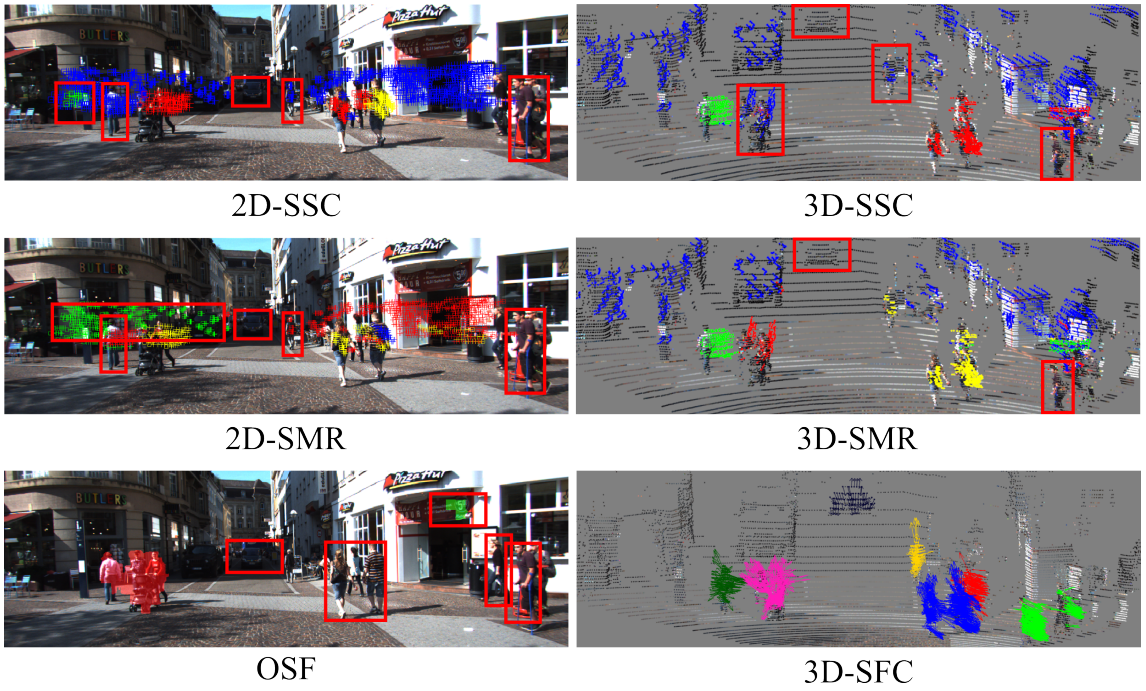


FIGURE 4.11 – Qualitative comparison of different motion segmentation approaches, namely 2D-SSC [29], 2D-SMR [30], OSF [256], and the proposed 3D-based approaches (the 3D-SFC algorithm refers to Chapter 5.4). Red boxes highlight the undetected or incorrectly segmented motions.

3D-SSC and the 3D-SMR algorithms. In general, both the 3D-SSC and 3D-SMR achieve much better performances than their 2D counterparts. Note that this significant improvement mainly comes from the direct clustering of the 3D data space which makes no camera projection model assumption. Remarkably, both the 2D-SSC and the 2D-SSC make the affine projection model assumption which is not always valid. Particularly, the evaluated datasets are mainly scenarios for city-modelling and autonomous driving, such scenes have strong perspective effects which violate the affine projection assumption.

Moreover, the 3D-SMR not only has better performances comparing to the 3D-SSC, but also has much less computational time. The spatial closeness and motion consistency constraints of 3D-SMR make the subspace representation more robust than the 3D-SSC. Specifically, the $0-1$ spatial distance graph constrains that only the neighbouring trajectories can be used for the SSR. In addition, the additional 3D motion consistency regularization term improves the robustness of the 3D-SMR. Thanks to the efficient Bartels-Stewart algorithm, the 3D-SMR requires much less computational time comparing to the 3D-SSC. Although the OSF achieves slightly better averaged specificity as the expense of very low sensitivity, it has much lower overall performances compared to the proposed 3D-SSC

and 3D-SMR. Figure 4.11 shows some qualitative results of the proposed algorithms with the state-of-the-art methods. Remarkably, the proposed 3D-based approaches achieve notable improvements.

4.7/ SUMMARY

This chapter introduces the proposed novel framework for 3D motion segmentation using the 3D-based sparse subspace clustering algorithm and smooth representation clustering approach that categorize the static scene parts and multiple moving objects. The proposed methods have been tested using the comprehensive real-world KITTI datasets and outperform their 2D-based counterparts. Our approach of sampling sparse feature trajectories based on their flow likelihood allows the proposed motion segmentation approach to handle wide range of motions, both in terms of magnitude, speed and coverage. Furthermore, the effectiveness of the incomplete trajectory construction, which is essential in many practical scenarios, is demonstrated. Finally, a complete framework for static scene parts construction and dynamic object reconstruction with semantic labelling are validated, which will be elaborated in Chapter 6.

5

Motion Segmentation with Known Camera Motion

"Each system is trying to anticipate change in the environment."

- Kevin Kelly, *Editor of Wired Magazine*

In this chapter, we introduce an effective algorithm to detect the moving objects from a set of registered point clouds. Compared to the previous chapter in which the camera motion was unknown, we now suppose that the camera motion can be precisely recovered after applying the point cloud registration techniques. We refer such scenarios as *Known Camera Motion* cases where the precise camera motion can be obtained via various approaches, i.e. Visual Odometry or ICP point cloud registration methods. In other words, the 3D point sets are roughly registered by compensating the camera ego-motion. As a result, given F frames of registered point clouds, there exist continuous displacements of point sets of moving objects, while the point sets of static scene parts have no displacement. Therefore, the static scene parts should overlay together while the dynamic scene parts should not.

By connecting the points of moving objects according to their temporal and spatial displacement, they become a set of motion vectors. In this regard, we propose a *3D Vector Field Analysis* approach which identifies the static points and the motion flows. After compensating the camera ego-motion, for every point in the previous frame, a flow vector is established by subtracting its nearest neighbour in the current frame. The flow vector encodes the motion direction and velocity of the objects. By exploiting these properties, the flow vectors of moving objects, so-called the motion flows, can be detected and classified

into their independent motions.

In the following, we introduce the complete pipeline of dynamic objects detection based on 3D point clouds. The 3D flow field analysis algorithm and the flow field segmentation algorithm are also presented and will be discussed in details. Similarly, the performances of such framework and the proposed algorithms are evaluated using extensive experiments of the real-world uncontrolled KITTI dataset.

5.1/ INTRODUCTION

3D map reconstruction is one of the most active research topics in computer vision due to the numerous application requirements in robot localization [278, 312, 313], autonomous driving [314, 1, 306], city map modelling [315, 316, 317, 318]. Benefiting from the emergence of affordable 2D and 3D cameras, high quality 3D maps of both indoor and outdoor environments can be obtained from nearly static environments [319, 320, 321, 284, 318, 322, 323, 324]. 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 parts of the scene which yield significant "ghost" effects.

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 etc., lead current methods to fail [278, 280, 279, 283, 325, 256, 260]. 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, and suitable for 360° FoV. Further, a complete framework for dynamic object detection, motion segmentation, and static map reconstruction is presented, see Fig. 5.1.

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. 5.1 Block 1. Naturally, given an accurate object-based point cloud segmentation with

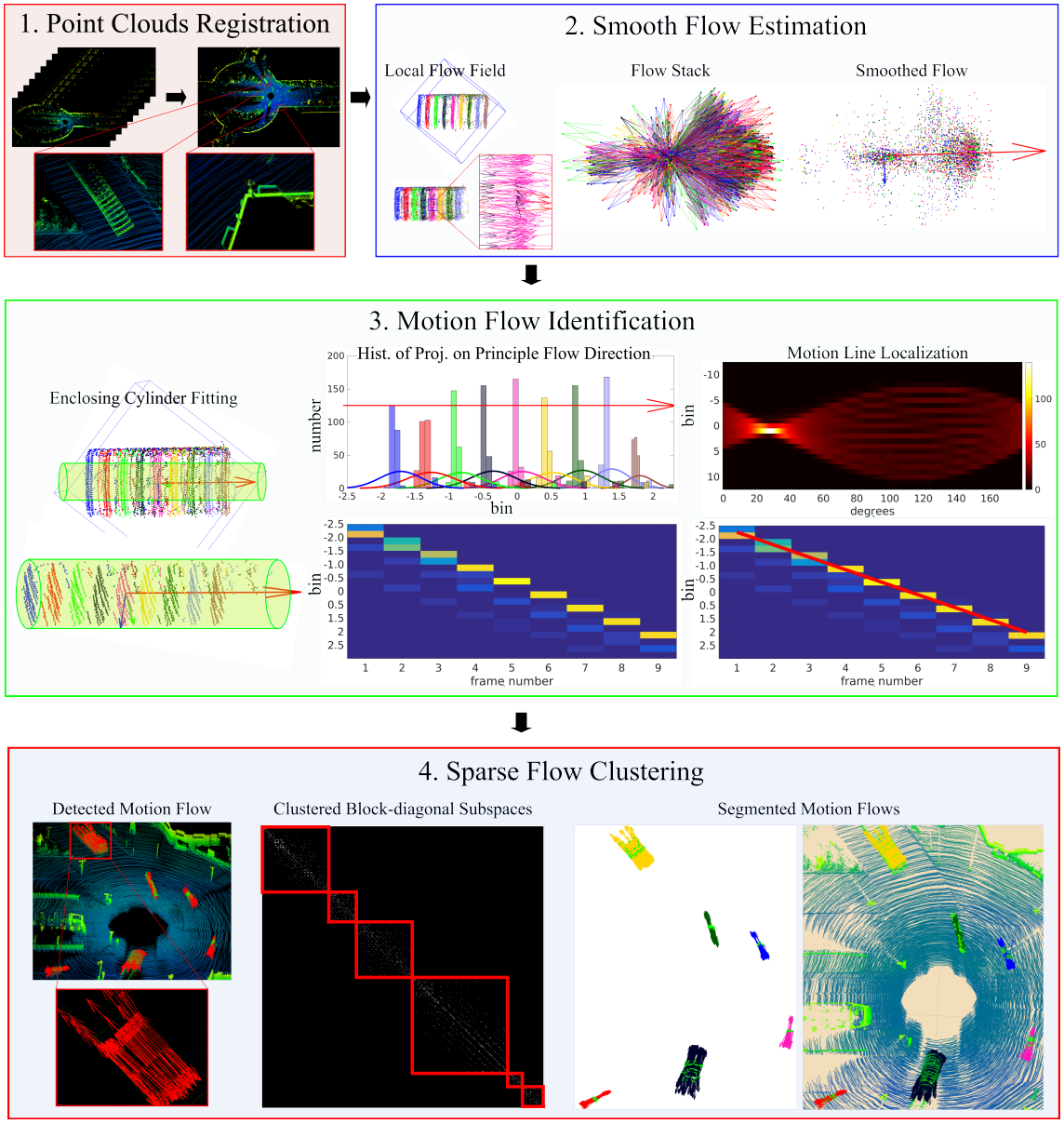


FIGURE 5.1 – Overview of the pipeline to detect and segment the motion flows. Block 1: Given a short 3D point cloud sequence, an Iterative Closest Point algorithm is applied to register the point clouds to compensate the camera ego-motion. Block 2: For each point, we compute a Smooth Flow Vector (SFV) using the neighbourhood (local) flow field within a bounding box. The SFV is estimated by the Eigen decomposition of the centred neighbouring flows (the flow stack), as detailed in Section 5.3.1. 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 correspondence to the object motion. Such phenomenon can be studied via the properties of the motion line located by the Radon transform on the 2D histogram, as detailed in Section 5.3.2. Block 4: The Sparse Flow Clustering algorithm regroups the detected motion flows into their independent motions, as detailed in Section 5.4.

point correspondence knowledge, the moving objects are discriminated by their displacements across frames. The precise establishment of point cloud correspondences and object segmentation are very challenging [326, 327, 328]. 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 three 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 which are registered by compensating the camera ego-motion. Such point correspondences are quickly established by using nearest neighbour search, although such correspondences may not be precise. Under local motion consistency assumption, the smooth flow vector is estimated as the local 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 flows 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. 5.1 Block 3. For each flow vector, an enclosing cylinder is adapted to select the most representative neighbour points which preserve a persistent geometric structure. The projections of those points onto the current flow vector are stored in a histogram. The motion flows can be identified by detecting shifts within the concatenated histogram from all the frames, as detailed in Section 5.3.4.

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 [29, 30]. 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.

5.2/ Background and Notations

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

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

$$\mathbf{A} \ominus \mathbf{B} = \{\mathbf{w}_i := \mathbf{y}_i - \mathbf{x}_i, \quad \forall \mathbf{x}_i \in \mathbf{A}\}, \tag{5.1}$$

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

$$\mathscr{N}(\mathbf{x}, \mathbf{B}) = \underset{\mathbf{y} \in \mathbf{B}}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{y}\|. \tag{5.2}$$

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

$$\mathscr{N}(\mathbf{x}, \mathbf{B}, r) = \{\mathbf{y} \in \mathbf{B} : \|\mathbf{x} - \mathbf{y}\| \leq r\}. \tag{5.3}$$

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

$$\mathscr{P}(\mathbf{S}, \mathbf{w}) = \{p : p = \mathbf{w}^\mathsf{T}\mathbf{x}, \mathbf{x} \in \mathbf{S}\}. \tag{5.4}$$

We refer an illustrative example of Eq. (5.4) to Fig. 5.2 which shows that a set of 3D points are projected onto the given 3D vector space. Note that the projection result of a three dimensional point to the given 3D vector axis is a 3D point, but we only take into account its scalar abscissa p on the axis. The origin of the axis is a specified 3D point, e.g. the

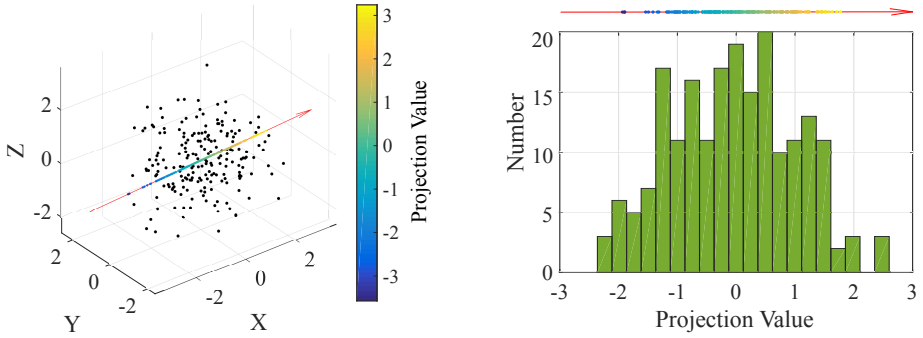


FIGURE 5.2 – Illustration of projections of 3D point set on a 3D vector. Left image contains a set of 3D points $\mathbf{S}$ (the black dots) and a 3D vector $\mathbf{w}$ (the red arrow). The color-coded dots highlight the projection positions of the 3D points onto the 3D vector space. Right image is the constructed 20-bin 1D histogram by using the projection values $\mathscr{P}(\mathbf{S}, \mathbf{w})$.

mean values of the 3D point set.

Furthermore, $\Theta \subset \mathbf{S}$ are the points within an infinite cylinder centred at $\mathbf{x}_c$, of radius r and axis $\mathbf{w}_c$, is given by

$$\Theta(\mathbf{x}_c, \mathbf{S}, \mathbf{w}_c, r) = \{\mathbf{x} : \|\mathbf{x} - \mathbf{x}_c\|^2 - \mathscr{P}(\mathbf{x}, \mathbf{w}_c)^2 \leq r^2, \mathbf{x} \in \mathbf{S}\}. \tag{5.5}$$

In other words, Eq. (5.5) rejects the points which have point-to-axis distances larger than the cylinder radius r, see Fig. 5.3.

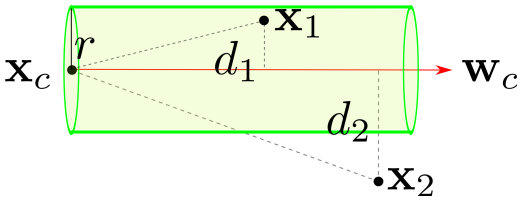


FIGURE 5.3 – Interpretation of an enclosing cylinder centred at $\mathbf{x}_c$ and axis $\mathbf{w}_c$. The distances from points $\mathbf{x}_1, \mathbf{x}_2$ to the cylinder axis $\mathbf{w}_c$ are notated as d_1 and d_2, respectively. Since ${d_1}^2 = \|\mathbf{x}_c - \mathbf{x}_1\|^2 - \mathscr{P}(\mathbf{x}_1, \mathbf{w}_c)^2 \leq r^2$, $\mathbf{x}_1 \in \Theta$ is considered as inside the cylinder. In contrast, $\mathbf{x}_2 \notin \Theta$ is outside the cylinder.

5.3/ FLOW FIELD ANALYSIS

Our 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. We propose the *3D Flow Field Analysis* model under the local motion consistency assumptions similar to the optical flow estimation [142] and the 3D

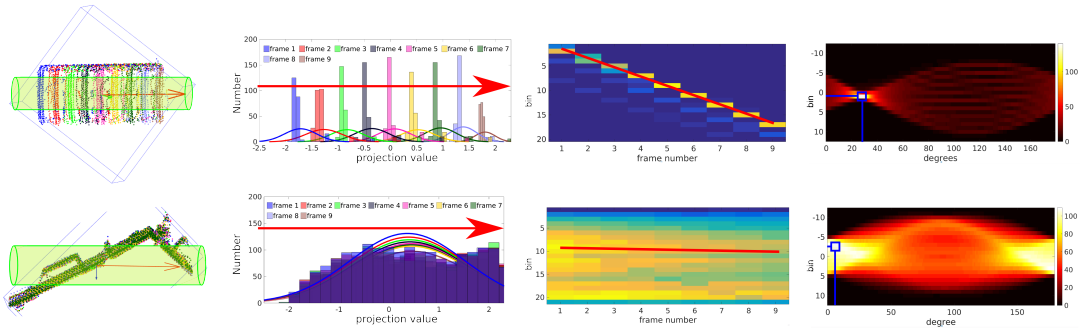


FIGURE 5.4 – Motion and static flow analysis: Row 1 and Row 2 are the graphical representations of the flow field analysis of a moving object and a static object, respectively. In comparison, Col. 1 shows the enclosing cylinder preserving the local structure. Col. 2 shows the 20-bin 1D histograms of cylinder-point projections of each frame. Remarkably, the histograms of motion flow (upper) are shifted along the flow direction, while the histograms of static flow (lower) are overlaid together. In Col. 3 are the concatenated all-frame histograms from Col. 2. The motion line L^* (solid red line) is estimated using the Radon transform in Col. 4 according to the criteria of Eq. (5.11).

scene flow estimation [329] where two assumptions are made: (i) the motion behaviours of optical flows are similar within a small neighbourhood and (ii) the local geometric structure does not change rapidly.

5.3.1/ SMOOTH FLOW VECTOR ESTIMATION

Given n point sets $\mathbf{S} = \{\mathbf{X}_t, t = 1, \ldots, n\}$. For $t = 1, \ldots, n-1$, we compute the point-wise flow which represents the evolution of points over time, as follows:

$$\mathbf{W}_t = \mathbf{X}_{t+1} \ominus \mathbf{X}_t. \tag{5.6}$$

In other words, we consider the difference between consecutive positions. Taking the locally homogeneous assumption of neighbouring flow vectors, we perform the smoothing of vector field by updating each $\mathbf{w}_i \in \mathbf{W}_t$ as

$$\mathbf{v}_i^* = \underset{\mathbf{v}\in\mathbb{R}^3}{\operatorname{argmax}} \sum_{\mathbf{w}\in\Omega(\mathcal{N})} \mathbf{w}^\mathsf{T}\mathbf{v} \quad \text{s.t.} \quad \|\mathbf{v}\| = 1 \tag{5.7}$$

where $\mathbf{v}_i^*$ is the returned desired smoothed flow vector to replace $\mathbf{w}_i$. $\mathcal{N} = \mathscr{N}(\mathbf{x}_i, \mathbf{X}_t, r)$ is the neighbourhood (within the radius r) that defines the local flow field $\Omega(\mathcal{N})$. Eq. (5.7) finds the consensus flow $\mathbf{v}_i^*$ which minimizes the overall distances between $\mathbf{v}_i^*$ and the flows within $\Omega(\mathcal{N})$. The problem of Eq. (5.7) can be solved efficiently as an eigen-decomposition

problem. Its solution can be obtained by computing the eigenvectors of the covariance matrix $\mathbf{W}^{\mathsf{T}}\mathbf{W}$, where the rows of $\mathbf{W}$ are $\mathbf{w}^{\mathsf{T}}$ for all $\mathbf{w} \in \Omega(\mathcal{N})$. The desired smoothed flow vector corresponds to the eigenvector of the largest eigenvalue. Note that, all the $\mathbf{w} \in \Omega(\mathcal{N})$ are normalized to unit vectors to obtain the optimal solution.

5.3.2/ Static Point and Motion Flow Discrimination

Consider that the structure of the local point sets is preserved. Thus, $\Theta_t = \Theta(\mathrm{x}, \mathrm{X}_t, \mathrm{w}, r), t = 1, \ldots, n$ (the measurements of a local point set moving along $\mathbf{w}$ from Eq. 5.7) are homomorphic. Therefore, the shape of distribution of projections $\mathscr{P}_t = \mathscr{P}(\Theta_t, \mathbf{w})$ remain unchanged over time interval $[1, t]$. Let $\mathcal{H}_t$ be a k-bin 1D histogram of projections $\mathscr{P}_t$ at time t. The motion state of the point sets can be described by the following equation:

$$\mathcal{H}_{t+1}(b) = \mathcal{H}_t(b + \alpha(t)), \tag{5.8}$$

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. (5.8) 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. (5.8) for all t, which can be modelled as a minimization problem as:

$$\underset{\beta,b}{\operatorname{argmin}} \sum_{t=1}^{n-1} \|\mathcal{H}_{t+1}(b) - \mathcal{H}_t(b + \beta t)\|_2. \tag{5.9}$$

To efficiently solve problem (5.9), the n 1D histograms $\mathcal{H}_t$ are concatenated into a 2D histogram $\mathbf{M} = [\mathcal{H}_1, \cdots, \mathcal{H}_n]$ of size $k \times n$, as illustrated in Fig. 5.4 middle columns. 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

$$L^* = \underset{\beta,b}{\operatorname{argmax}} \int \mathcal{H}_t(L(t))\, dt. \tag{5.10}$$

Since the Radon transform [330] computes the volume integration in different angles at different positions in a continuous manner, problem (5.10) can be solved efficiently and globally by applying Radon transform on $\mathbf{M}$, as illustrated in Fig. 5.4. 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:

$$S = \sum_{t=1}^{n} s_t \quad \text{and} \quad E = -\sum_{t=1}^{n} s_t log(s_t). \tag{5.11}$$

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.

5.3.3/ 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 in Eq. (5.7) on local flow field estimation), 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. 5.5. 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.

Precisely, we use a dynamic searching strategy along the flow direction. Let $\mathcal{B} = \{\mathbf{B}_t, t = 1, \ldots, f\}$ be the assembly of f frames of point sets within a local bounding box. Initially, the bounding box (centred at $\mathbf{x}_c$) covers $f < n$ frames, due to the high speed. Let $\mathscr{P}_t(\mathbf{B}_t, \mathbf{w}), t = 1, \ldots, f$ be the projections of $\mathcal{B}$ along the motion direction $\mathbf{w}$, 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 $\mathbf{x}_t = \mathbf{x}_c + \delta_t \mathbf{w}$, until all n frames are covered.

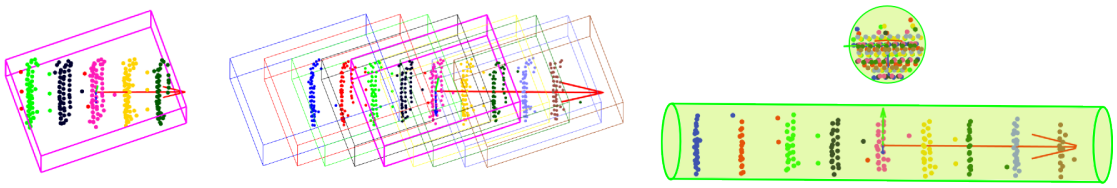


FIGURE 5.5 – Dynamic local neighbourhood search of a fast moving object: left shows that the bounding box covers only 5 frames. Middle shows the translation of bounding box along the flow direction. Right shows the enclosing cylinder with full frames.

5.3.4/ IMPLEMENTATION DETAILS

Given n consecutive frames of point sets, an ICP-based registration algorithm is applied to compensate the camera ego-motion. Notably, robust ICP algorithms [331, 318] are preferred to obtain precise camera motion estimation. According to our expertise, ICP registration on edge and plane feature points generally yields satisfactory results, similarly to [318]. Starting from the registered point sets as input, Algo. 7 is applied to discriminate the static and the dynamic points, and to estimate the motion flows of the dynamic points. For the sake of computational efficiency, the points from ground plane are detected and removed beforehand. Note that the detection of ground plane for the data acquired by a ground-vehicle is 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 neighbours within the enclosing cylinder (sum of the 2D histogram $\mathbf{M}$). $\tau_\beta = 0.175$ 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 integration in different angles at different positions. Thus, its maximum response directly gives the desired solution of problem (5.10). In Fig. 5.4 Col. 2, the 1D histograms from dynamic scene part have shifting effects along the flow direction, as expected. Differently, these histograms tend to overlap with each other for the static scene parts. These phenomena lead to the different behaviours (refer to the above discussions in Section 5.3.2) of the motion line L^* of static and dynamic points.

Algorithm 7: Motion Flow Identification.

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

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

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

- Place a 3D bounding box at $\mathbf{x}_i$ for local flow field estimation ($\mathbf{W}$) using Eq. (5.6), and perform eigen-decomposition: $[\mathbf{V}, \mathbf{D}] = \text{eigen}(\mathbf{W})$ to obtain the dominant flow $\mathbf{v} = \mathbf{V}(:, 3)$.
- Fit an enclosing cylinder $\Theta(\mathbf{x}_i, \mathbf{X}, \mathbf{v}, r)$.
- Project cylinder points to axis $\mathbf{v}$ using Eq. (5.4), and compute histograms $\mathcal{H}_t, t = 1, \ldots, n$ to construct $\mathbf{M}$.
- Compute the slope β^* of L^* using Radon transform on $\mathbf{M}$, motion strength S and stability E using Eq. (5.11).
- If $\beta^* < \tau_\beta$, $S < \tau_S$ and $E < \tau_E$, reject static point $\mathbf{x}_i$.

Result: Detected motion flow set Ω.

5.4/ SPARSE FLOW CLUSTERING

We cluster the dynamic point set, obtained from the flow field analysis (discussed in Section 5.3), 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 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 $\mathbf{X} = [\mathbf{x}_1, \ldots \mathbf{x}_j, \ldots \mathbf{x}_n]$ and $\mathbf{W} = [\mathbf{w}_1, \ldots \mathbf{w}_j, \ldots \mathbf{w}_n]$ be the $3 \times n$ matrices of the point set and the corresponding flow vectors of the moving objects, the self-expressive sparse representation (similar to [29]) can be written as

$$\mathbf{W} = \mathbf{WC}, \tag{5.12}$$

where the sparse $n \times n$ matrix $\mathbf{C} = [\mathbf{c}_1, \ldots \mathbf{c}_j, \ldots \mathbf{c}_n]$ 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

$$\left\|\mathbf{x}_j - \mathbf{X}\mathbf{c}_j\right\|_2^2 \leq \epsilon_r, \quad \forall j. \tag{5.13}$$

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

$$\begin{aligned} \underset{\mathbf{C}}{\text{minimize}} \quad & \|\mathbf{C}\|_{1,1}\,, \\ \text{subject to} \quad & \mathbf{W} = \mathbf{WC}, \qquad \text{diag}(\mathbf{C}) = 0, \\ & \left\|\mathbf{x}_j - \mathbf{X}\mathbf{c}_j\right\|_2^2 \leq \epsilon_r, \qquad \forall j. \end{aligned} \tag{5.14}$$

This is a convex problem, whose optimal solution can be found using second order cone programming [295]. Its equivalent problem is the semi-definite programming given by

$$\begin{aligned} \underset{\mathbf{C},\mathbf{S}}{\text{minimize}} \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij} \\ \text{subject to} \quad & \mathbf{W} = \mathbf{WC}, \qquad \text{diag}(\mathbf{C}) = 0, \\ & -s_{ij} \leq c_{ij} \leq s_{ij}, \qquad \forall\{i,j\}, \\ & \begin{pmatrix} \mathbf{I} & \mathbf{x}_j - \mathbf{X}\mathbf{c}_j \\ (\mathbf{x}_j - \mathbf{X}\mathbf{c}_j)^\mathsf{T} & \epsilon_r \end{pmatrix} \geq 0, \qquad \forall j, \end{aligned} \tag{5.15}$$

where s_{ij} are the elements of $\mathbf{S}$.

5.4.1/ INFLUENCE OF NOISE AND OUTLIERS

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

$$\mathbf{w}_j = \mathbf{w}_j^0 + e_j, \tag{5.16}$$

where $e_j \in \mathbb{R}^3$ is the noise or outlier entry of noise free data $\mathbf{w}_j^0$. Replacing Eq. (5.12) with Eq. (5.16), we have

$$\mathbf{W} = \mathbf{WC} + \mathbf{E}. \tag{5.17}$$

Due to the local structure persistence and temporal flow speed consistency assumptions, the sought sparse representation from the current frame is valid for the neighbour frames. Therefore, the sparse clustering problem of Eq. (5.15) can be reformulated as:

$$\begin{aligned} \underset{\mathbf{C},\mathbf{E}_\mathrm{x},\mathbf{E}_\mathrm{w}}{\text{minimize}} \quad & \|\mathbf{C}\|_{1,1} + \mathbf{E}_\mathrm{w} + \mathbf{E}_\mathrm{x}, \\ \text{subject to} \quad & \left\|\mathbf{w}_j - \mathbf{W}_t\mathbf{c}_j\right\|_2^2 \leq \epsilon_\mathrm{w}, \quad \forall j, \quad c_{jj} = 0, \quad t = 1,\ldots,n, \\ & \left\|\mathbf{x}_j - \mathbf{X}_t\mathbf{c}_j\right\|_2^2 \leq \epsilon_\mathrm{x}, \quad \forall j, \quad c_{jj} = 0, \quad t = 1,\ldots,n, \end{aligned} \tag{5.18}$$

where $\mathbf{E}_\mathrm{w} = \lambda_1 \sum_{j=1}^{n} \epsilon_\mathrm{w}$ and $\mathbf{E}_\mathrm{x} = \lambda_2 \sum_{j=1}^{n} \epsilon_\mathrm{x}$. $\mathbf{X}_t$ and $\mathbf{W}_t$ 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. (5.18) can be solved as a semi-definite programming problem. Weight parameters λ_1 and λ_2 are simply set to 1.

5.4.2/ Implementation Details

The Sparse Flow Clustering (SFC) algorithm consists of three major steps (see Algo. 8) which are implemented using CVX [332] optimization toolbox. In the sparse optimization step, a binary $n \times n$ connectivity graph $\mathbf{D} = [\mathbf{d}_1, \cdots \mathbf{d}_j, \cdots, \mathbf{d}_n]$ is used to enforce the spatial closeness constraint on the selected sparse representation elements, such that Eq. (5.17) becomes:

$$\mathbf{W} = \mathbf{W}(\mathbf{C} \cdot \mathbf{D}) + \mathbf{E}, \quad \forall d_{ij} > \tau_d, d_{ij} = 0, \quad \text{else} \quad d_{ij} = 1, \tag{5.19}$$

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 $\mathbf{C}$ improves the algorithm's computational efficiency.

In step 2 of Algo. 8, a sparse symmetric similarity graph $\mathbf{G} = |\mathbf{C}^*| + |\mathbf{C}^*|^\mathsf{T}$ is constructed. Since $\mathbf{G}$ encodes the connectivity information among the flows, a K-mean spectral clustering is employed to group the flow clusters. In fact, K can be determined by finding the number of graph components, which can be obtained by analysing the eigenspectrum of the Laplacian matrix of $\mathbf{G}$ [333]. However, other model selection techniques [241] should be employed when there are connections between points in different subspaces. In the following experiments, we provide the number of motions as an input to all the algorithms for fair comparison.

Note that the proposed SFC does NOT rely on feature tracking and feature trajectory (unlike [29, 30]), making it more practical for highly dynamic environment motion analysis. Moreover, the SFC algorithm, which is proposed under the robust sparse subspace

Sequence	# Frms.	# Objs.	2D-SSC			3D-SSC			2D-SMR-Q1			2D-SMR-Q2			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 5.1 – 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.

representation framework, offers new research perspectives for vector field analysis.

Algorithm 8: Sparse Flow Clustering.

Data: 3D point sets $[\mathbf{X}_1, \cdots, \mathbf{X}_t]$ and flows $[\mathbf{W}_1, \cdots, \mathbf{W}_t]$.
Result: K clustered subspaces.
1 Sparse flow representation using Eq. (5.18).
2 Sparse similarity graph: $\mathbf{G} = |\mathbf{C}^*| + |\mathbf{C}^*|^\mathsf{T}$.
3 K-mean spectral clustering on $\mathbf{G}$.

5.5/ EXPERIMENTS

We conduct extensive evaluations on the challenging real-world KITTI benchmark [1] 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 [318]), and that GPS and IMU information are not used. Recall that the 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 5.1 and Table 5.3. 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 [29, 30] is adopted.

Sequence	2D-SSC		3D-SSC		2D-SMR-Q1		2D-SMR-Q2		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 5.2 – Quantitative evaluation on KITTI dataset: using Mean and Median values of Misdetection rate metric.

5.5.1/ MOTION DETECTION EVALUATION

Our 3D-based Moving Object Detection (3D-MOD) algorithm is compared against four representative algorithms. We recall that the 2D-SMR, 2D-SSC, 3D-SSC, and 3D-SMR 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 5.1 summarizes the performances of 2D-SMR-Q1 [30], 2D-SMR-Q2 [30], 2D-SSC [29], 3D-SSC 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.

Similar to Eq. (4.17), table 5.2 adopts the mean and median *Misdetection Rate* error metric defined as

$$\eta = \frac{\text{number of false positive } + \text{ number of false negative}}{\text{number of features}}. \tag{5.20}$$

For a more compact illustration, the corresponding Whisker's box-plot [334] statistical comparison is summarized in Fig. 5.6. Similar remarks from Table 5.1 can be observed: the 3D-SSC and 3D-MOD has significantly better performances than other methods due to their persistent high specificity. Fig. 5.6 also shows that the 3D-MOD outperforms the other methods with lower median misdetection rate as well as much higher robustness.

Table 5.3 concludes the comparisons between the 3D-MOD against the Object Scene Flow (OSF) [256] algorithm. Since the OSF method produces dense moving object detection and segmentation, for the purpose of fair comparison, a 3D Region Growing [335]

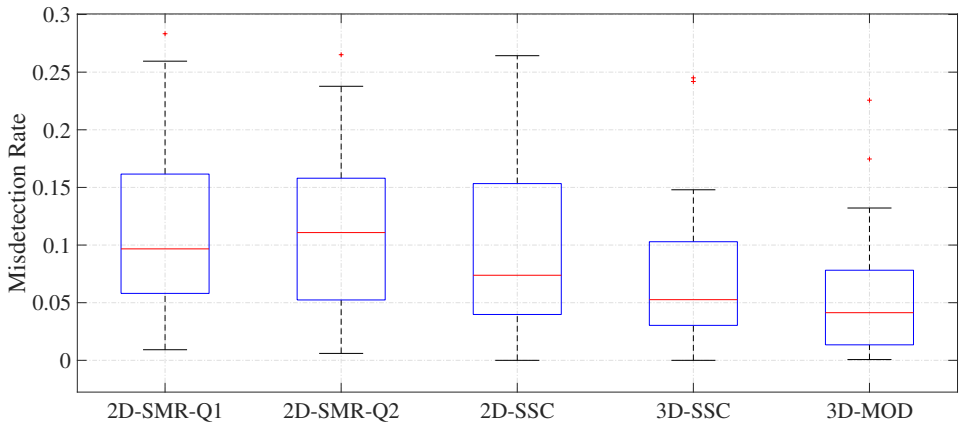


FIGURE 5.6 – Misdetection rate comparison on KITTI dataset.

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:

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

ii 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.

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

5.5.2/ MOTION SEGMENTATION EVALUATION

For motion segmentation quantification, we use the *Misclassification Rate* (same as [29, 30]) to compare the algorithms' performances, see Fig. 5.7. Overall, our 3D-SFC achieves the best results for the evaluated datasets. Fig. 4.11 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. 4.11 bottom-right image), which largely simplifies the motion flow clustering problem. Moreover, the 3D-SFC is proposed under the sparse

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

representation framework with extra spatial closeness constraint, which produces a very reliable similarity graph for spectral clustering.

5.6/ SUMMARY

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 robust, efficient and accurate. In many aspects, both the 3D-MOD and 3D-SFC algorithms outperform the state-of-the-art

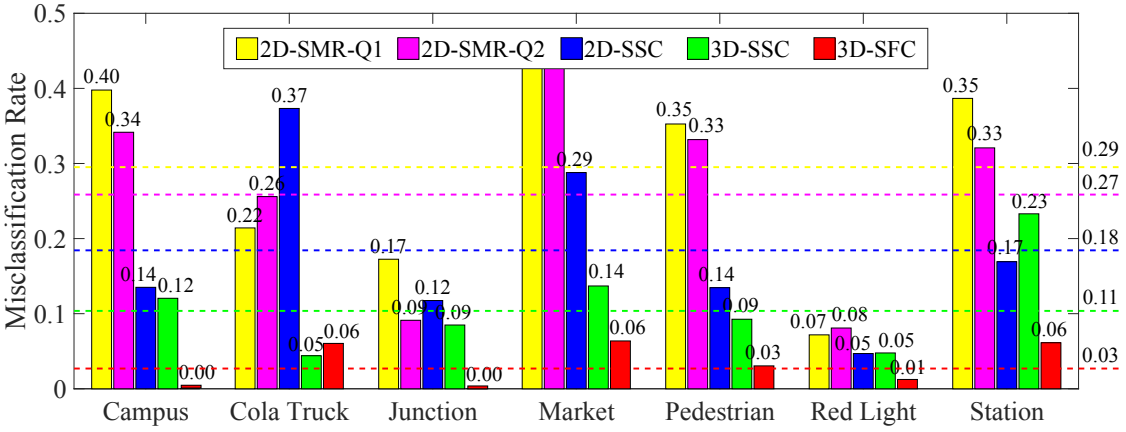


FIGURE 5.7 – Motion segmentation quantification: dashed lines highlight the averaged misclassification rates.

methods since we have compared all these techniques on comprehensive highly dynamic real-world KITTI datasets, for which they consistently exhibit a better accuracy, lower misclassication and misdetection rates.

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.

6

Scene Reconstruction and Understanding

"When you have all these traces of trash moving around, you can ask yourself how can we make the system more efficient. Then we can make better decisions. "

- Carlo Ratti, *Massachusetts Institute of Technology*

The previous chapters address the problems of moving object detection and segmentation for both known and unknown camera motion cases. A step forward is the reconstruction and the understanding of the 3D scenes. In this chapter, we present a novel two-step 3D point cloud registration scheme which consists of the minimal 3-Point Random Sample Consensus (3P-RANSAC) algorithm and the Dual-Weight Iterative Closest Point (DW-ICP) approach. The point cloud registration is initialized using the 3P-RANSAC algorithm which exploits the properties of Gibbs 3D rotation representation and the Cayley transform. Such algorithm solves the 3D transformation problem in a linear manner under the RANSAC framework for the seek of robust estimation. The initial registration is refined via the proposed DW-ICP which contains two energy terms, namely the matching consensus energy and the closest-point energy. The DW-ICP iteratively minimizes the registration error by incorporating the robust estimation techniques (e.g. M-Estimators). The proposed point cloud registration scheme is applied to the reconstruction of both static scene parts and the rigidly moving objects.

To understand the 3D scene, we propose to use a 2D-to-3D semantic label transfer approach which learns semantic knowledge of the scene using image information and as-

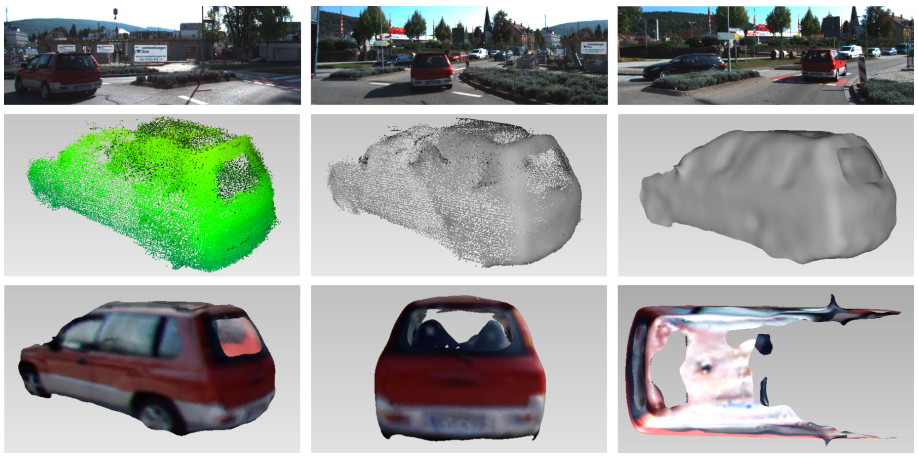


FIGURE 6.1 – Moving car reconstruction from a mobile platform: on top row are selected frames of a moving car. The middle row shows the registered sparse point cloud, the smoothed point cloud, and the reconstructed mesh of the point cloud, respectively. The bottom row shows the fine reconstruction in different views.

signs the corresponding labels to the 3D objects. The proposed approaches are valid using the afore-discussed seven representative datasets to produce high quality static map reconstruction, rigidly moving object reconstruction and semantic labelling of the reconstructed scenes.

6.1/ INTRODUCTION

Scene reconstruction and understanding are two major tasks in 3D Computer Vision. The reconstruction offers the exact observation of the 3-dimensional world of its size, shape and geometric structure intuitively, whereas the semantic scene information allows the understanding of the world. Both have always been active areas of research due to their wide range of potential applications, such as scene representation, understanding, and robot navigation [336].

For a moving 2D-3D camera setup, the 3D reconstruction of the scene can be obtained by registering a sequence of point clouds with the help of Visual Odometry (VO) measurements [316, 337]. However, the VO-based registration is valid only for the static scene parts. Therefore, such reconstruction suffers from several visual artefacts due to the dynamic parts. In this regard, 3D-SSC and 3D-SMR motion segmentation approaches categorize the scene into static and dynamic parts before performing VO. Moreover, although camera ego-motion can be roughly estimated using ICP-based approaches as discussed

in Chapter 5, such algorithms are rather preliminary and naive. To this end, we propose a pipeline for the high quality reconstruction of both static scene parts and dynamic objects, making them dense, coherent, and complete, see Fig. 6.1 for instance.

Given multiple sparse and partial point clouds observed from different view ports, the proposed pipeline (see Fig. 6.2) aims to produce high quality reconstructions by exploiting both the 2D and the 3D observations. The point clouds are registered roughly with the help of 3P-RANSAC on the point cloud correspondences. Since the 3-point RANSAC algorithm estimates the 3D-to-3D rigid transformation between two point sets, the accuracy of the registration highly relies on the quality of the correspondence set. We recall that the 3D-to-3D feature correspondences are established by the tracking of their associated 2D features, which is sensitive to noise. Moreover, point cloud registration from long term observations inherently suffers from multi-layered problems due to the multiple scans of the same area. This can largely decrease the quality of the registration while increasing the memory consumption. To address these problems, a more robust and effective algorithm, called Dual-Weighted Iterative Closest Point (DW-ICP) algorithm, is proposed. A 3D reconstruction enhancement framework is presented to produce photographic quality results of real outdoor scenes. Finally, the semantic information of the 3D scene is assigned using the 2D-to-3D label transfer strategy.

The DW-ICP Algorithm: Iterative Closest Point (ICP) is one of the most commonly used algorithm due to its simplicity and robustness. However, the convergence of ICP algorithm requires a good initialization and rich geometric structures. To overcome these problems, an initialization using 3-point RANSAC registration algorithm is recommended. Moreover, a DW-ICP algorithm is introduced to iteratively estimate the rigid transformation by assigning different weights to the RANSAC inlier point pairs and the ICP correspondences, as detailed in Section 6.2.3.

3D Reconstruction Enhancement: Due to the noise of data, the 3D registration from

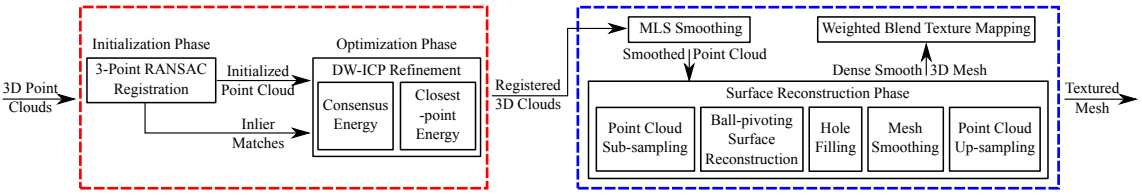


FIGURE 6.2 – Framework for high quality rigid object reconstruction: the point clouds registration and mesh reconstruction.

multiple observations has multi-layered artefacts. To address this problem, we employ a 3D Thin Plane Spline algorithm which smooths and combines the multi-layered object surface into a single layer. Furthermore, a ball pivoting surface triangulation approach is applied to construct 3D meshes of the smoothed point clouds. Finally, the textures of the 3D meshes are mapped and refined using mutual information, as detailed in Section 6.3.

The 2D-to-3D Label Transfer: Thanks to the recent advancement in deep learning, it is now possible to obtain faithful semantic labels using image information. To associate the semantic labels with static and dynamic objects, we transfer their labels obtained by using object detector (e.g. the Yolo [338] detector) on the corresponding images. The transfer of these labels is carried out by the max-pooling over multiple detections. We argue that the semantic understanding of dynamic 3D scenes has obtained very little attention in literature: Geiger et al. [305] propose a 3D traffic scene understanding framework which predicts the motions of vehicle tracklets by fusing semantic (Sky, Road, and Traffic Lane) and 3D scene flow information. Different from [305], our method results in the motion trajectories of generic objects alongside with their labels (e.g. pedestrian, cyclist, car), as well as the labels of static parts (e.g. traffic lights).

6.2/ Robust Point Clouds Registration

To register a sequence of sparse point clouds, we formulate an optimization problem supported by a set of noisy feature trajectories. The accurate registration is obtained by jointly optimizing the registration of feature matching pairs and the closest-points correspondences, which is the key prior to obtain high quality textured surface reconstructions.

6.2.1/ Linearized Rigid Motion Formulation

Given a set of correspondences between two 3D point clouds, the exact solution for rigid motion parameters, e.g. $\mathbf{R}$ and $\mathbf{t}$, can be obtained in a linear manner. Let $\mathrm{X} = [x, y, z]^{\mathsf{T}}$ and $\mathrm{Y} = [x', y', z']^{\mathsf{T}}$ be two corresponding 3D points under rigid transformation, denoted as

$$\mathrm{X} = \mathbf{R}\mathrm{Y} + \mathbf{t}, \tag{6.1}$$

where $\mathbf{R}$ is the 3×3 rotation matrix and $\mathbf{t}$ is the 3×1 translation matrix. Let g be the Gibbs representation [339] of a rotation matrix $\mathbf{R}$, we have $\mathsf{G} = [\mathsf{g}]_\times$ is a 3×3 skew-symmetric matrix, where $\mathsf{g} = \mathsf{e}\, tan\frac{\theta}{2} = [g_x, g_y, g_z]^\mathsf{T}$ with $\mathsf{e} = [e_x, e_y, e_z]^\mathsf{T}$ is the Euler rotation axis and rotation angle θ. Applying the Cayley Transformation [340], $\mathbf{R}$ can be represented as:

$$\mathbf{R} = (\mathsf{I}_3 + \mathsf{G})^{-1}(\mathsf{I}_3 - \mathsf{G}), \tag{6.2}$$

where I_3 is a 3×3 identity matrix. Replacing Eq. 6.1 using Eq. 6.2, we have:

$$\mathsf{X} = (\mathsf{I}_3 + \mathsf{G})^{-1}(\mathsf{I}_3 - \mathsf{G})\,\mathsf{Y} + \mathbf{t}, \tag{6.3}$$

multiplying $(\mathsf{I}_3 + \mathsf{G})$ on both sides, we have:

$$(\mathsf{I}_3 + \mathsf{G})\mathsf{X} = (\mathsf{I}_3 - \mathsf{G})\,\mathsf{Y} + (\mathsf{I}_3 + \mathsf{G})\mathbf{t}. \tag{6.4}$$

Notate $\tilde{\mathbf{t}} = (\mathsf{I}_3 + \mathsf{G})\mathbf{t} = [\tilde{t}_x, \tilde{t}_y, \tilde{t}_z]^\mathsf{T}$ and reorganize Eq. 6.4, we have:

$$(\mathsf{X} - \mathsf{Y}) = -\mathsf{G}(\mathsf{X} + \mathsf{Y}) + \tilde{\mathbf{t}}. \tag{6.5}$$

To linearise Eq. 6.5, we fill in the elements of variables as follows:

$$\begin{bmatrix} x - x' \\ y - y' \\ z - z' \end{bmatrix} = - \begin{bmatrix} 0 & g_z & -g_y \\ -g_z & 0 & g_x \\ g_y & -g_x & 0 \end{bmatrix} \begin{bmatrix} x + x' \\ y + y' \\ z + z' \end{bmatrix} + \begin{bmatrix} \tilde{t}_x \\ \tilde{t}_y \\ \tilde{t}_z \end{bmatrix}. \tag{6.6}$$

Expand Eq. (6.6), we have:

$$\begin{bmatrix} x - x' \\ y - y' \\ z - z' \end{bmatrix} = \begin{bmatrix} 0 & z + z' & -(y + y') & 1 & 0 & 0 \\ -(z + z') & 0 & (x + x') & 0 & 1 & 0 \\ y + y' & -(x + x') & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} g_x \\ g_y \\ g_z \\ \tilde{t}_x \\ \tilde{t}_y \\ \tilde{t}_z \end{bmatrix}. \tag{6.7}$$

Eq. 6.7 is a linear system such that the parameters can be estimated using a Linear Least Square approximation. Since the skew-symmetric matrix has rank 2, each matching pair provides 2 independent equations, the linear system requires minimum 3 matching pairs

to be solved. To maximize the number of inliers, a Random Sample Consensus (RANSAC) framework is adopted. However, in the presence of inaccurate correspondences, obtained from noisy motion trajectories, the quality of RANSAC registration is usually not very satisfactory. Therefore, we refine the registration by minimizing the dual-weighted closet-point energy.

6.2.2/ Robust Closest-Point Energy Minimization

When two overlapping point clouds of the same rigid object are given, the transformation is generally obtained by minimizing the energy derived from the closest-points distance. In most of the cases, this energy is minimized using an iterative method – also known as Iterative Closest Point (ICP) algorithm [341, 342]. In every step, the ICP algorithm considers the closest points across two point clouds, say the reference and the model, to be the corresponding ones. Let $\mathbf{X} = \{\mathsf{X}_1, ..., \mathsf{X}_n\}$ be the reference point cloud, and $\mathbf{Y} = \{\mathsf{Y}_1, ..., \mathsf{Y}_m\}$ be the new model, the robust method of ICP iteratively minimizes the following energy:

$$\mathcal{E}_I(\hat{\mathbf{T}}) = \min_{\mathbf{T}} \sum_{i=1}^{n} \rho(\min_{j\in\{1,\cdots,m\}} |\mathsf{X}_i - \mathbf{T}\mathsf{Y}_j|), \tag{6.8}$$

where $\hat{\mathbf{T}}$ is the desired transformation matrix. Note that the energy term $\mathcal{E}_I$ includes a robust cost function to handle noisy and partial data. Our choice of robust cost, say $\rho(x)$, is the Tukey's biweight function [343]:

$$\rho(x) = \begin{cases} (\tau^2/6)(1 - [1 - (x/\tau)^2]^3) & \text{if } |x| \le \tau \\ (\tau^2/6) & \text{if } |x| > \tau \end{cases}, \tag{6.9}$$

and the weight of each corresponding pair is defined by:

$$w(x) = \frac{1}{x}\frac{\mathrm{d}\rho(x)}{\mathrm{d}x} = \begin{cases} [1 - (x/\tau)^2]^2 & \text{if } |x| \le \tau \\ 0 & \text{if } |x| > \tau \end{cases}, \tag{6.10}$$

where τ is the inlier threshold, such that outliers ($|x| > \tau$) are assigned with zero weights.

6.2.3/ Modified Closest-Point Energy Minimization

While the consensus-based registration method requires a subset of accurate correspondences, the closest-point-based method requires rich geometric structures of the point clouds. This prohibits to make a choice of one method over another. Therefore, we propose to minimize a combined energy function – one from consensus, noted $\mathcal{E}_R$, and the other from closest-point, say $\mathcal{E}_I$. We minimize the energy function in an iterative manner, hence use the terminology Dual-Weighted Iterative Closest Point (DW-ICP).

First, we define an energy function that measures the quality of the inlier set obtained from 3-point RANSAC. Note that due to the sparsity and the noisiness of feature points, the estimated transformation matrix obtained from RANSAC can be imprecise. Let $\{\mathsf{X}_i \leftrightarrow \mathsf{Y}_i\}, i = 1, \ldots, k$ be the inlier set, the energy $\mathcal{E}_R$ for matching consensus is expressed as:

$$\mathcal{E}_R(\hat{\mathbf{T}}) = \min_{\mathbf{T}} \sum_{i=1}^{k} \tilde{\rho}(|\mathsf{X}_i - \mathbf{T}\mathsf{Y}_i|), \tag{6.11}$$

where $k \leq m, n$, and $\tilde{\rho}(\cdot)$ is the Huber's weight function:

$$\tilde{\rho}(x) = \begin{cases} (x^2/2) & \text{if } |x| \leq \tilde{\tau} \\ \tilde{\tau}[|x| - (\tilde{\tau}/2)] & \text{if } |x| > \tilde{\tau} \end{cases}, \tag{6.12}$$

$$\tilde{w}(x) = \frac{1}{x}\frac{\mathrm{d}\tilde{\rho}(x)}{\mathrm{d}x} = \begin{cases} 1 & \text{if } |x| \leq \tilde{\tau} \\ (\tilde{\tau}/|x|) & \text{if } |x| > \tilde{\tau} \end{cases}, \tag{6.13}$$

where $\tilde{\tau}$ is the threshold for inlier matches. The Huber loss function is selected under the assumption that the provided inlier set is noisy without severe outliers that need to be completely discarded. In the spirit of Eq. (6.8) and Eq. (6.11), we formulate our combined energy function as follows:

$$\begin{aligned} \mathcal{E}(\hat{\mathbf{T}}) = \min_{\hat{\mathbf{T}}} \Bigg\{ & \alpha \sqrt{\frac{1}{n}\sum_{i=1}^{n} \rho(\min_{j \in \{1,\cdots,m\}} |\mathsf{X}_i - \mathbf{T}\mathsf{Y}_j|)} \; + \\ & (1-\alpha)\sqrt{\frac{1}{k}\sum_{i=1}^{k} \tilde{\rho}(|\mathsf{X}_i - \mathbf{T}\mathsf{Y}_i|)} \Bigg\}, \end{aligned} \tag{6.14}$$

where α is the regularization term to control the influence of the $\mathcal{E}_I$ and $\mathcal{E}_R$ energy terms. Rather than optimizing the closest-point energy $\mathcal{E}_I$ or matching consensus energy $\mathcal{E}_R$ independently, the DW-ICP aims to iteratively and simultaneously optimize the joint energy $\mathcal{E}$ of Eq. (6.14).

6.2.4/ DISCUSSIONS

The proposed method uses the 3D motion trajectories of a sequence of segmented point clouds obtained from 3D-SSC and 3D-SMR as input. First, we use the 3-Point RANSAC registration to roughly register the point clouds as initialization. Afterwards, the DW-ICP is applied to refine the registration. Note that (also refer to Eq. (6.14)) the DW-ICP iteratively minimizes a combined energy term, one from consensus $\mathcal{E}_R$ and other from closest-point $\mathcal{E}_I$, during the optimization process. On the one hand, $\mathcal{E}_I$ minimizes the overall registration error of the whole 3D point clouds. On the other hand, $\mathcal{E}_R$ minimizes the registration error of the inlier obtained from RANSAC. These two terms are usually complementary to each other, which is the key to the success of the proposed optimization framework.

On top of traditional ICP, there are two main advantages of DW-ICP: (a) The feature matching constraint promises a proper registration regardless of the poor geometry structure of the point clouds. (b) Robust estimation framework is preserved such that the algorithm is generic and robust to outliers during a long term registration.

6.3/ 3D MESH GENERATION

The complete pipeline for high quality 3D reconstruction of rigidly moving objects, using 2D-3D camera setup attached to a moving vehicle, is shown in Fig. 6.2. This section details the reconstruction of photo-realistic high quality 3D models, which serves for the important topic in computer graphic –3D meshes generation [344, 345, 346]. A full pipeline is presented in Fig. 6.2 (blue box), which consists of three major steps, namely Moving Least Square (MLS) [347] point cloud smoothing, 3D Mesh Reconstruction, and Weighted Blend Texture Mapping [348].

Point Cloud Smoothing: Due to the measurement noise of the laser scanner and imperfect registrations, any point cloud registered over long sequences suffers from outlier

and multi-layer effects. The reconstructed meshes of such point cloud suffer from many visual artefacts, such as spiky mesh and holes. Thus, a MLS algorithm, which smooths an unorganized point could using a polynomial fitting, is applied.

Surface Reconstruction: Prior to the surface reconstruction, a poisson-disk distribution [349]-based sub-sampling removes the redundant points (overlapped points) due to the multiple observations of the same scene. Later, a Ball Pivoting Triangulation algorithm [350] is used to establish the neighbour-points relationships, followed by a dilation operation for hole filling. Next, a Taubin Surface Smoothing method is adopted to smooth the reconstructed surface while preserving the sharp edges. Finally, a Least Square Subdivision approach [351] is performed to up-sample and is followed by the re-meshing of the point cloud to produce high quality meshes.

Texture Mapping: We use the 2D images acquired by the 2D-3D camera setup for texture mapping. During this process, a photographic alignment between the 3D mesh and the images is required. Since the 2D-3D camera setup is calibrated, and the motion of the camera is known, all the images are aligned with respect to the mesh reconstructed frame. The camera poses (between the cameras and the reconstructed mesh) are estimated by computing the inverse of the transformation matrices (obtained from registration) and using the camera calibration parameters. Furthermore, the blurring effect during the texture fusion from multiple images is reduced by using a Weighted Blending algorithm.

6.4/ 2D-TO-3D LABEL TRANSFER

We consider that the semantic scene understanding should answer two questions: What is the object? And what is it doing? In other words, the object of interest should be discovered and recognized with semantic labels. Further, the object behaviour, such as a moving or parked cars, should be understood. In this context, semantic scene understanding has been partially addressed in [305] for moving vehicle motion prediction. We focus on the fusion of knowledge from 2D and 3D data to fully address the semantic scene understanding problem.

Since 2D image-based semantic labelling achieves very satisfactory performances [338], we propose to transfer the retrieved 2D object labels to their corresponding point clouds. Recall that the 2D-3D correspondences are established using a projective projection mo-

del: $\mathbf{x} \sim \mathbf{KPX}$ where $\mathbf{x}$ is the 2D projections of the 3D points $\mathbf{X}$. $\mathbf{K}$ and $\mathbf{P}$ are the intrinsic and extrinsic parameters obtained from camera calibration. Thus, the label of $\mathbf{x}$ can be transferred to $\mathbf{X}$. Let Γ be the semantic label assigned to a 3D object. $\mathcal{S}_\Gamma$ is the real-world-averaged size of object class Γ, and $\mathcal{S}_i$ is the object size (volume) measured from its 3D point cloud. To accurately transfer the 2D labels over m different observations, a max-pooling strategy is applied to obtain the desired label Γ^* for the given 3D object, such that:

$$\Gamma^* = \underset{\Gamma}{\operatorname{argmax}}\ \eta_i \rho_i,\ i = 1, \ldots, m, \tag{6.15}$$

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

6.5/ EXPERIMENTS

We evaluate the performances of the registration algorithms using both synthetic and real data. Since it is difficult to obtain the true motions of the camera and the rigidly moving objects, we focus on the synthetic data which simulate the motion behaviour of the true object motions. The DW-ICP algorithm parameters were set as $\alpha = 0.8$, $\tau = 0.08m$ and $\tilde{\tau} = 0.03m$. The stopping condition of the DW-ICP iteration is defined as: rotation tolerance $\epsilon_\mathbf{R} = 10e\text{-}6$, translation tolerance $\epsilon_\mathbf{t} = 10e\text{-}6$, and max iteration as 100. In addition, we evaluate the static map reconstruction performances using the KITTI dataset with manually labelled groundtruth. Further, we qualitatively evaluate the 3D reconstruction of large-scale city scenes with and without moving object removal.

6.5.1/ POINT CLOUD REGISTRATION QUANTITATIVE EVALUATION

Synthetic Datasets: The synthetic datasets were generated from three different objects, namely the Van, Red Car, and Cola Truck, see Fig. 6.3 for example. We simulate the motion behaviours of rigidly moving objects with smooth rotation and translation for 100

frames. Practical scenarios, such as partial overlaps, occlusions, and poor 3D geometric structures, are also simulated. We applied 10 levels of Gaussian noise, from 0.005 to 0.050 in meters. The maximum noise level is chosen as 2.5 times higher than the expected accuracy (0.02m) of the Velodyne laser scanner. We compare the performances of the algorithms using the averaged absolute rotation and translation errors.

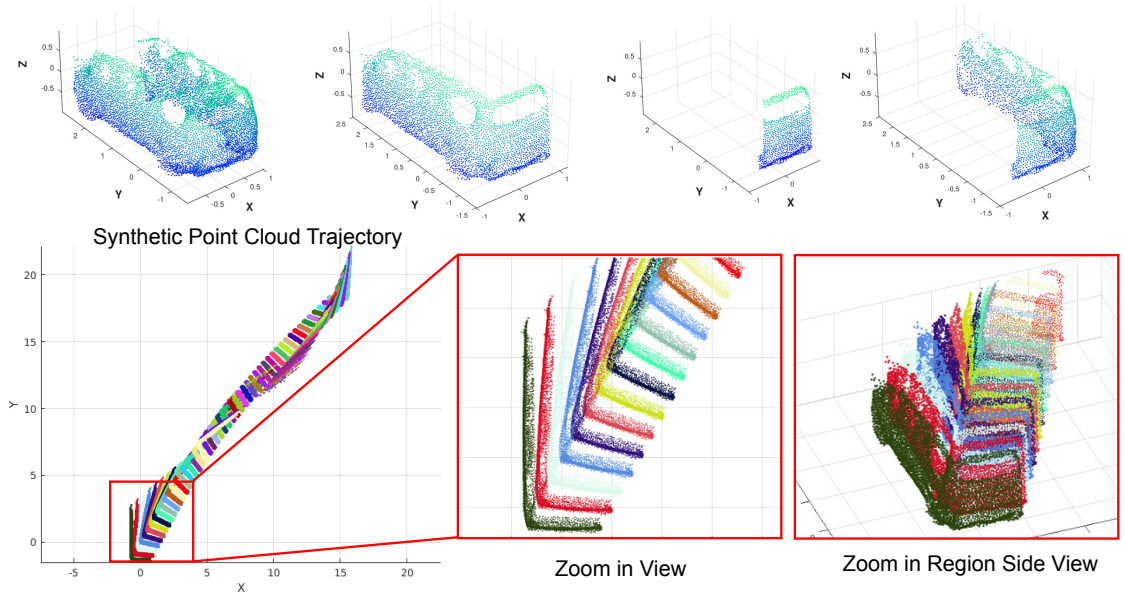


FIGURE 6.3 – Synthetic Trajectory of Van Object: first row shows the complete van object model with different side views. Second row shows the synthetic trajectory of the van object with various view ports.

Fig. 6.4 shows the performances of 4 different algorithms, namely 3-Point RANSAC [316], RANSAC+ICP refinement [341], RANSAC+Robust-ICP [331] and RANSAC+DW-ICP. The overall performance of the algorithms are ranked (from top to down) as: DW-ICP, Robust-ICP, RANSAC+ICP and RANSAC. The Robust-ICP (using M-Estimator) has significantly better performance against that of traditional ICP. Most importantly, the proposed DW-ICP consistently outperforms the other approaches, regardless of rotation or translation.

Real Datasets: Table 6.1 depicts the dataset information, where the 3D Error (averaged Leave-One-Out Error) metric is used to quantify the registration performance. The registration error of our method is consistently lower than 3P-RANSAC [316], although we have slightly more computational time due to the DW-ICP refinement process. Moreover, the high quality reconstructions of Fig. 6.1 and Fig. 6.5 are obtained using the proposed framework of Fig. 6.2. Note that the objects are reconstructed from long-term and long-distance observations (see column *Distance* of Table 6.1), under the situations that

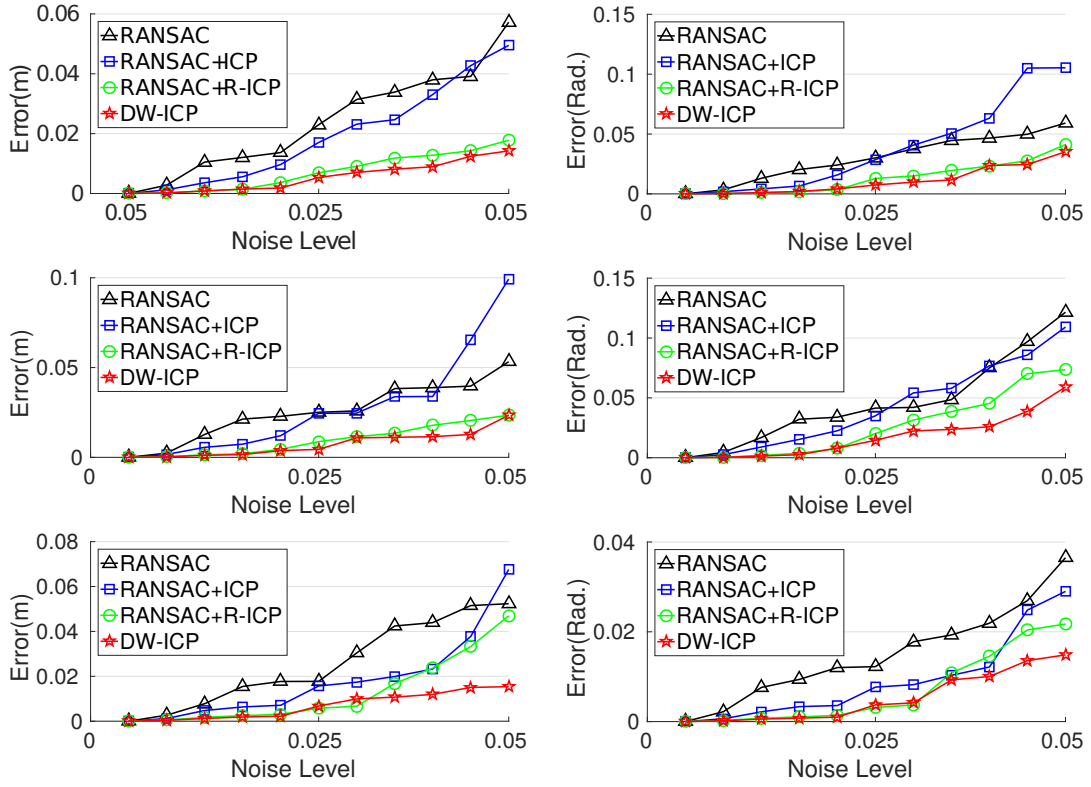


FIGURE 6.4 – Quantification of point cloud registration using synthetic data: the top to bottom rows are translation and rotation errors on Van, Red Car, and Cola Truck dataset, respectively.

both target objects and camera system are moving in high speeds. Remarkably, the 3D reconstruction using 3P-RANSAC is not only sparse and noisy, but also has multi-layered problems. On the contrary, the framework effectively overcomes the accumulation errors during the registration process and produces easily recognizable results. Figure 6.5 demonstrates that significantly more satisfactory results of our method are achieved compared to that of [316].

Object	# Frame	Sides	Distance (m)	3P-RANSAC		Ours	
				Error (m)	Time (s)	Error (m)	Time (s)
Van	44	3	16.5	0.0150	3.1	**0.0131**	4.6
Red Car	60	3	10.8	0.0084	2.8	**0.0080**	4.3
Cola Truck	48	2	30.0	0.0234	3.7	**0.0229**	4.1

TABLE 6.1 – Rigidly moving object dataset information: *Col. Sides* is the number of object sides (left, right, back, and front) being captured. *Col. Dist.* is the averaged distance from the camera to the object. *Col. 3P-RANSAC [316]* and *Col. Ours* show their respective averaged 3D error and computational time.

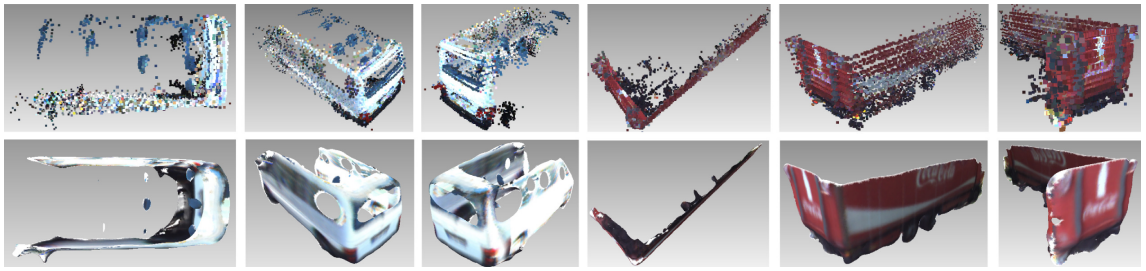
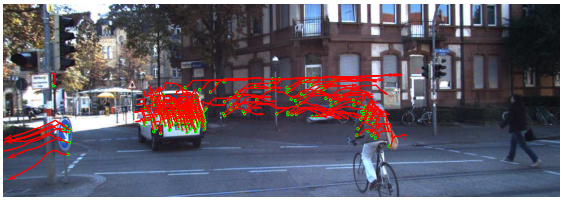
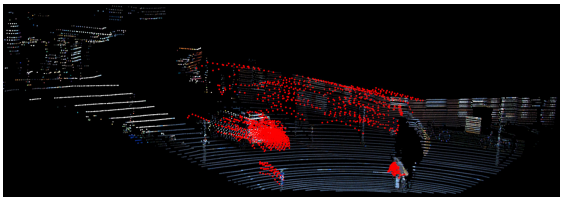


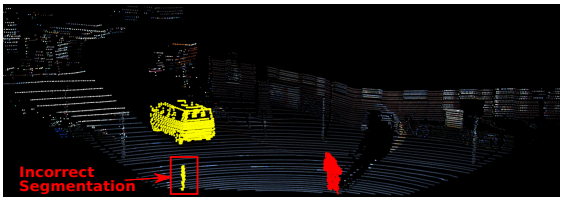
FIGURE 6.5 – Qualitative comparison of reconstructed Van and Cola Truck: the top row images are the 3D reconstruction of rigidly moving Van and Cola Truck using 3P-RANSAC [316]; the bottom row images are the obtained high quality 3D meshes using the proposed framework.



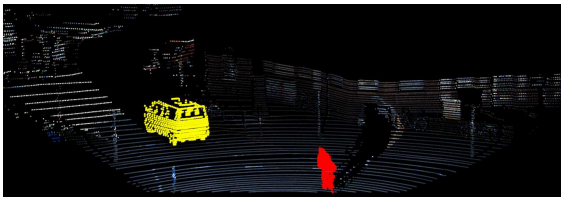
(a) 2D feature trajectories.



(b) 3D feature trajectories.



(c) Dense segmentation based on 2D-SSC [177].

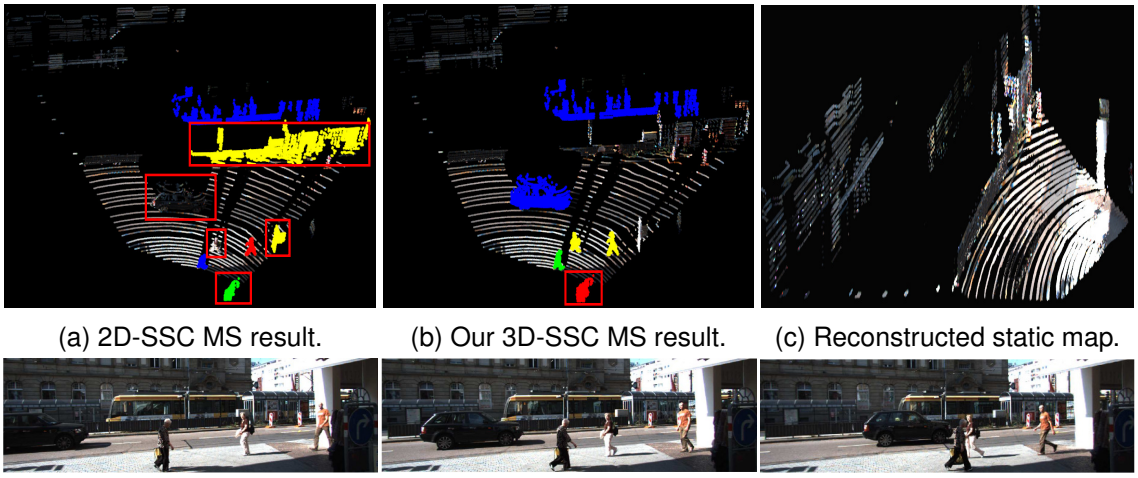


(d) Dense segmentation based on 3D-SSC.

FIGURE 6.6 – Qualitative comparison of 2D-SSC vs. 3D-SSC in motion segmentation: (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.

6.5.2/ STATIC MAP RECONSTRUCTION EVALUATION

Benefiting from the effectiveness of the proposed 3D-SSC, 3D-SMR and 3D-MOD motion segmentation methods, the static maps of all the representative datasets are reconstructed. Although the camera motion is unknown, in most cases, we can safely assume that the major cluster or the most widely distributed cluster of the trajectories corresponds to the background objects. The remained clusters are considered as moving entities which should be detached. Thus, as a prior step to the static map reconstruction, we densely segment the moving objects in each frame. In this step, we apply the 3D Region Growing [335] technique, which takes the sparse feature points (trajectories) of the moving objects as initial seeds, for dense 3D point cloud segmentation, see Fig. 6.6 and Fig. 6.7.



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

FIGURE 6.7 – Train station sequence static map reconstruction results: (a) and (b) show the 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.

The first row of Fig. 6.6 shows both the 2D and the 3D feature trajectories, and the second rows are the dense segmentation results using 2D-SSC and 3D-SSC, respectively. Note that the traffic pole in Fig. 6.6c is wrongly segmented as the same motion of the moving Van, which is due to the incorrect motion segmentation of the 2D-SSC.

A more challenging Station dataset (shown in Fig. 6.7) contains a fast moving car, three slowly moving pedestrians, and a intermittently occluded train by moving objects. Interestingly, by applying the 3D-SSC, all moving objects: pedestrians, fast driving car, and occluded train are detected and removed correctly in the reconstructed static map (see Fig. 6.7c). 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. 6.7b), so as the two pedestrians (yellow objects in Fig. 6.7b).

6.5.2.1/ STATIC MAP RECONSTRUCTION QUANTIFICATION

Intuitively, the static map can be obtained by registering the static scene parts from multiple observations. Thus, the quality of the overall static map relies on the dense segmentation of each frame. To quantify, we manually segment and label the dynamic scene parts of three sequences, namely Cola Truck, Junction and Station sequence. To this end, Table 6.2 summarizes the quantification results of the static map reconstruction. Starting from the second column, the columns represent the number of moving objects, the num-

ber of correctly and incorrectly removed objects, and accuracies in removing the dynamic objects and maintaining the static scene parts. The Dynamic Accuracy is defined by

$$\text{Dyn. Acc.} = \frac{\text{number of points segmented from dynamic objects}}{\text{total number of points from dynamic objects}}, \tag{6.16}$$

and the Static Accuracy is defined in a similar manner. Specifically, if the moving objects are over segmented –some parts of the static scene are removed– then the Stc. Acc. will be lower than 100%. Note that these measurements are made on the densely segmented point clouds, unlike in the motion segmentation evaluations which are mainly based on the sparse feature trajectories. A higher dynamic accuracy (Dyn. Acc.) means a better removal of dynamic objects. Similarly, the higher static accuracy (Stc. Acc.) stands for a more complete static map. Results show that the dynamic objects are removed correctly with very high accuracy, meanwhile, the static scene parts are maintained very well. The reported computational time includes the time for both MS and static map reconstruction.

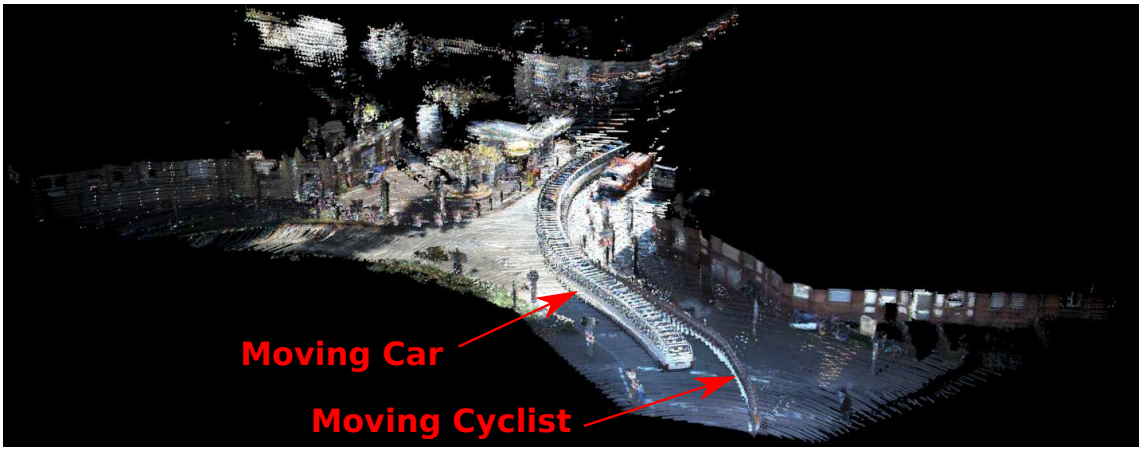
Sequence	# Objs.	Corr.	Incorr.	Dyn. Acc.(%)	Stc. Acc.(%)	Time (min.)
Cola Truck	1	1	0	97.55	100	6.00
Junction	2	2	0	91.02	100	13.40
Station	5	5	1	91.60	92.47	3.16

TABLE 6.2 – Static map reconstruction quantification based on 3D-SSC motion segmentation results.

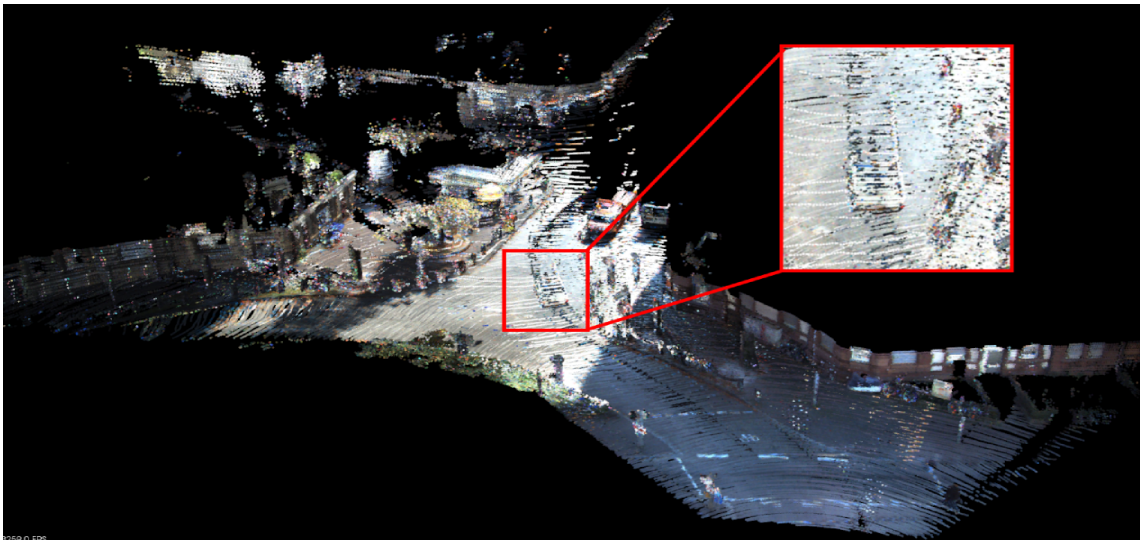
6.5.2.2/ STATIC MAP RECONSTRUCTION QUALIFICATION

To show the distinctive performances of the proposed approaches, we compare the static map 3D reconstruction after removing the dynamic objects using the proposed 3D-SSC and 3D-SMR. Moreover, we compare the static map reconstruction before and after feature trajectories completion. Finally, we show the improvement of 3D registration using the proposed DW-ICP.

Figure 6.8 shows the full scene 3D reconstruction and the static scene reconstruction of the Junction sequence. As can be seen from Fig. 6.8a, the "ghost" artefacts from trajectories of moving Van and Cyclist significantly degrade the quality of the reconstructed 3D map. By removing the moving objects, a much higher quality static map is achieved, see Fig. 6.8b. However, as highlighted by the red bounding box, part of the moving van



(a) Reconstructed full scene using [316].



(b) Static map reconstruction without incomplete trajectory completion.

FIGURE 6.8 – Junction sequence results: (a) shows the full scene 3D reconstruction using 80 frames. (b) shows the reconstructed static map without moving objects based on the 3D-SSC motion segmentation results. Last row images show the corresponding image sequence for every 15 frames.

trajectory still remains, which due to the existence of incomplete feature trajectories. In other words, the moving van is partially occluded by a cyclist, which leads to the failure of pixel-level feature tracking. Likewise, Fig. 6.11a contains the new appearing cyclist, which occludes the pedestrian (recall Fig. 4.5). In such cases, discarding the incomplete

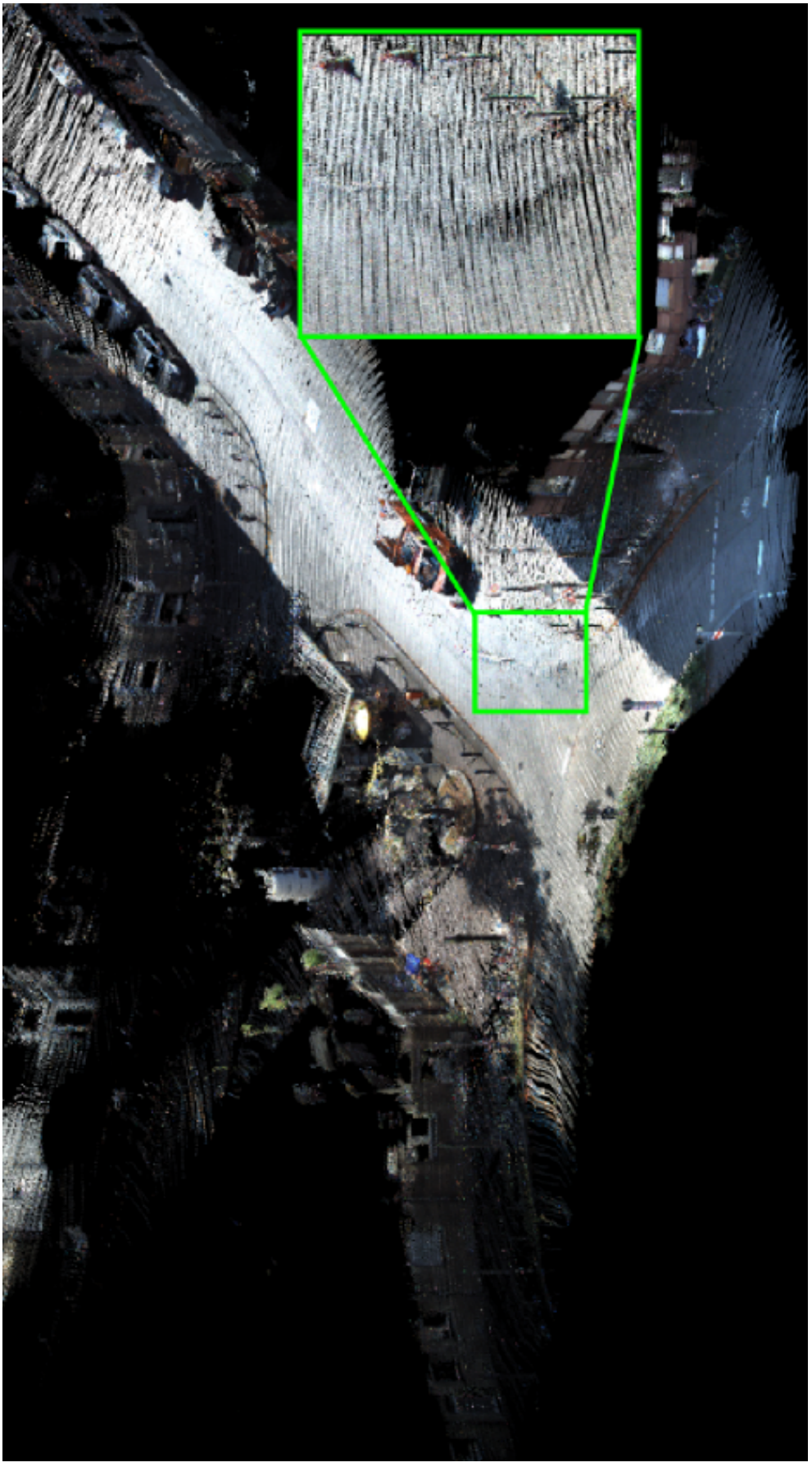
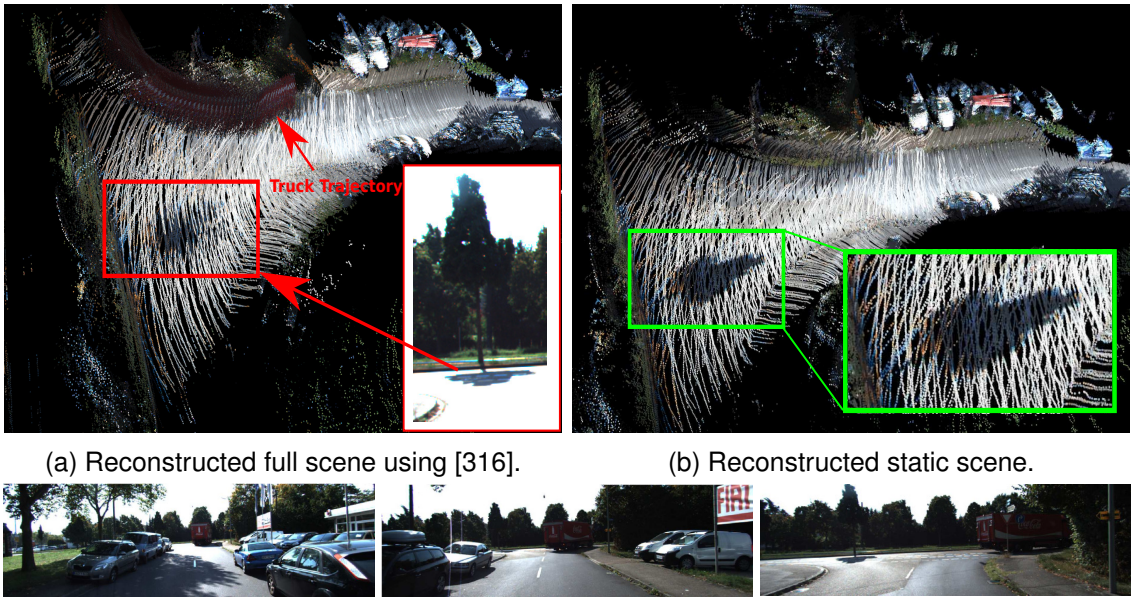
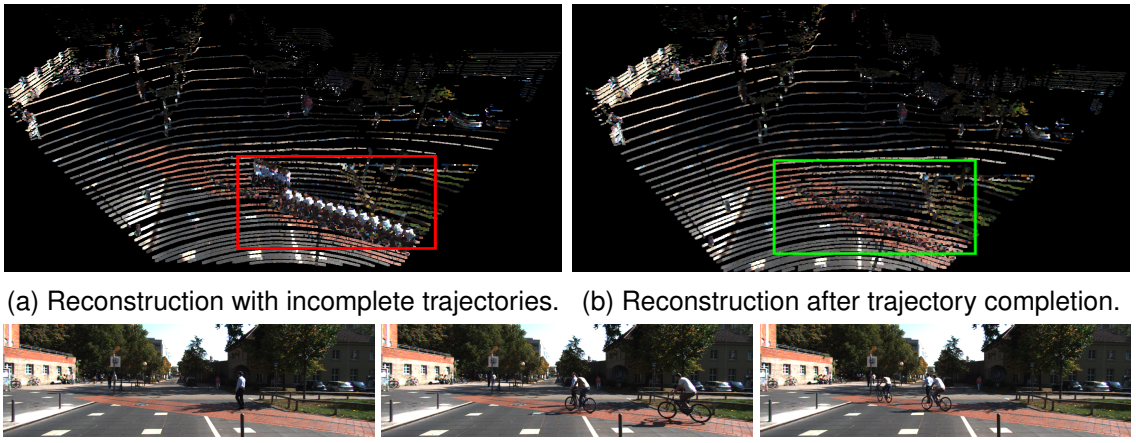


FIGURE 6.9 – High quality 3D reconstruction of Junction sequence by incorporating the proposed 3D-SMR motion segmentation and the proposed DW-ICP refinement.



(a) Reconstructed full scene using [316]. (b) Reconstructed static scene.

FIGURE 6.10 – Cola Truck sequence static map reconstruction based on 3D-SSC motion segmentation results: (a) shows the full scene 3D reconstruction where the red rectangle highlight the reconstruction of the tree shadow. (b) shows the reconstructed static map without moving objects. Last row images show some images of the sequence.



(a) Reconstruction with incomplete trajectories. (b) Reconstruction after trajectory completion.

FIGURE 6.11 – Incomplete trajectory recovery assisted static maps reconstruction based on 3D-SMR motion segmentation results: (a) shows that the reconstructed static map contain some neglected moving objects due to the loss of feature tracking. With the help of incomplete feature trajectory completion, finer static maps of (b) is achieved by removing those loss-tracked moving objects.

trajectories might lead to the misdetection of some moving objects. Differently, rather than discarding those incomplete feature trajectories, we extend them to have the same feature number as of the complete feature trajectories, which allows the concurrent motion segmentation on both complete and incomplete trajectories. Fig. 6.9 and 6.11b show that higher quality static maps are obtained by taking into account the incomplete feature

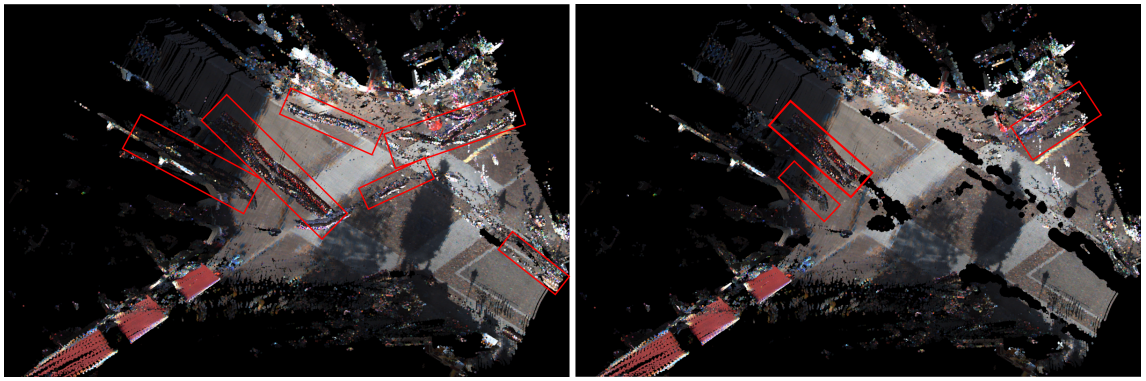


FIGURE 6.12 – 3D reconstruction of Market sequence based on 3D-SMR motion segmentation: left and right images show the reconstructed static map with and without incomplete trajectory completion, respectively. The red bounding boxes highlight the failures of moving object removal. Right image shows that major parts of the moving objects are removed after the trajectory completion.

trajectories recovery.

Furthermore, Fig. 6.9 illustrates the high quality 3D reconstruction of Junction sequence by integrating the feature trajectories completion algorithm, the 3D-SMR motion segmentation method, and the DW-ICP registration approach. It is noteworthy that the small objects (e.g. the traffic poles) are also well registered, as remarked in the green bounding box. More interestingly, Fig. 6.10 demonstrates the significant improvement of the obtained reconstruction result using the proposed methods on the Cola Truck sequence. For instance, the red rectangle region in Fig. 6.10a highlights the tree shadow which is barely recognizable. On the contrary, the same shadow in Fig. 6.10b has been recovered more realistically. In the close-up view of all the built maps, similar differences are abundant. The superior performance of our method is mainly due to two reasons: i) point cloud registration using only the static scene parts yields more robust results; ii) the proposed DW-ICP algorithm is more effective than the 3P-RANSAC algorithm.

6.5.2.3/ STATIC MAP WITH KNOWN CAMERA MOTION

The previous sections show the remarkably better results of static map reconstruction compared to the full scene reconstruction. However, these reconstructed scenes are relatively simple with not many (less than 5) moving objects involved. Practically, there exist much more complicated scenes where the trajectories of moving objects are intersecting with severe occlusions. In such cases, motion segmentation approaches like 3D-SSC or

3D-SMR are not effective due to two main reasons: (i) feature trajectory construction is very difficult; (ii) the number of moving objects is changing at all time. Fig. 6.12 illustrates that major parts of the moving objects are detected and removed by applying the feature trajectory completion algorithm. Yet, some untracked moving objects still remain.

Recall that the proposed flow field analysis approach does not rely on feature trajectory construction, and is able to detect motions without prior knowledge. After compensating the camera ego-motion estimated using the lidar odometry approach [352], through the flow field analysis, the proposed 3D-MOD algorithm is able to detect and remove the moving objects in highly dynamic scenes, e.g. the Market sequence of Fig. 6.13 and Fig. 6.14. Note that there are many challenges raised by the Market sequence, namely unknown number of motions, slow motions, small-size objects, severe occlusions, intersecting trajectories, unconstrained motions and sudden illumination changes.

The 3D reconstructions of Market sequence are shown in Fig. 6.13 using Lidar-Visual Odometry [318], Fig. 6.12 left using 3D-SSC, Fig. 6.12 right using 3D-SMR and Fig. 6.14 using 3D-MOD. Despite these difficulties, the static map produced by our framework is of very high quality because our framework is not sensitive to light changes, occlusions, slow or very fast motions, etc. To conclude, when the camera ego-motion is unknown or cannot be estimated precisely, 3D-SSC and 3D-SMR approaches are adequate options to obtain high quality static maps, especially when the scene is simple. In cases of known camera motion (or precisely recovered camera motion), the 3D-MOD approach is an optimal option which sensitively detects wide range of moving objects.

6.5.3/ LABEL TRANSFER EVALUATION

Figure 6.15 presents the automatically labelled 3D map of Junction sequence with the proposed 2D-to-3D label transfer strategy. In this figure, the semantic information of the 3D objects is accurately discovered using the proposed max-pooling strategy, which avoids multi-labelling from different observations. Furthermore, the accurate object motion velocities are estimated using 3-point RANSAC and ICP point cloud registration. Objects are categorized as either the static or the moving objects. Then the accurate online motion information (*e.g.* motion direction, linear and angular speed, etc.) is obtained thanks to the precisely recovered odometry from the proposed framework.

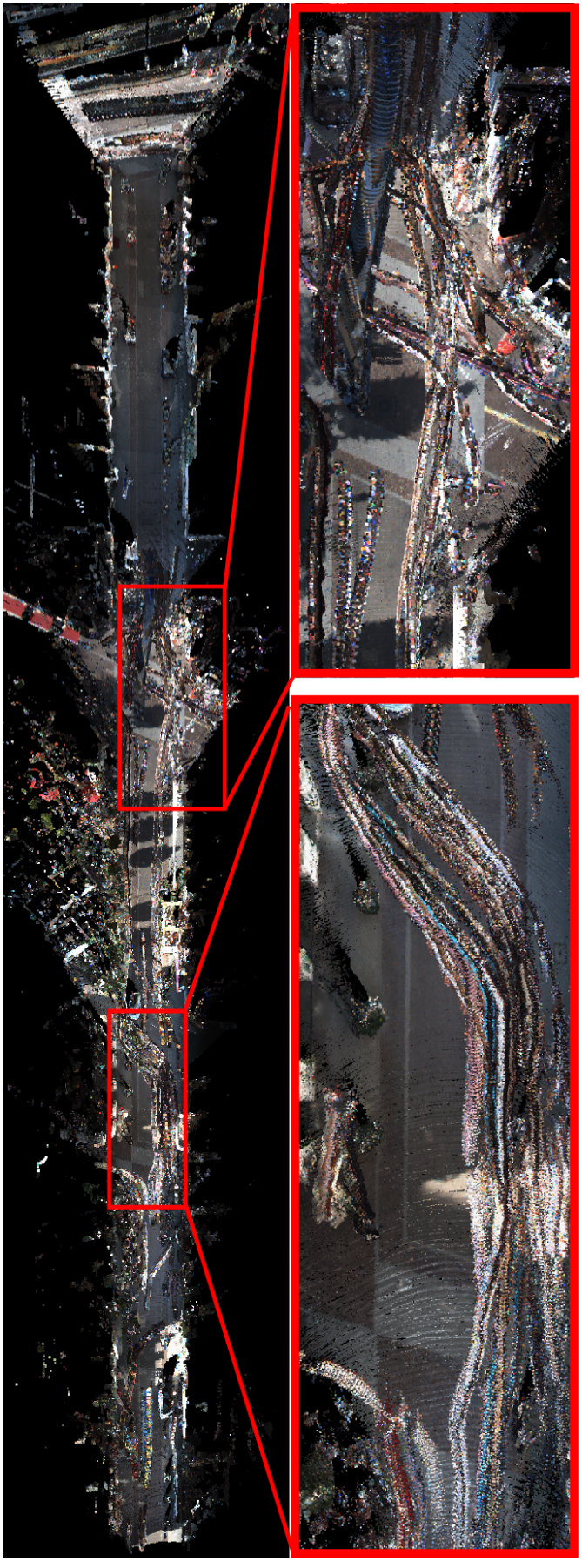


FIGURE 6.13 – Full scene 3D reconstruction [318] of Market sequence with numerous moving objects. The zoom-in regions show the immense artefacts from the walking pedestrians and moving cars.

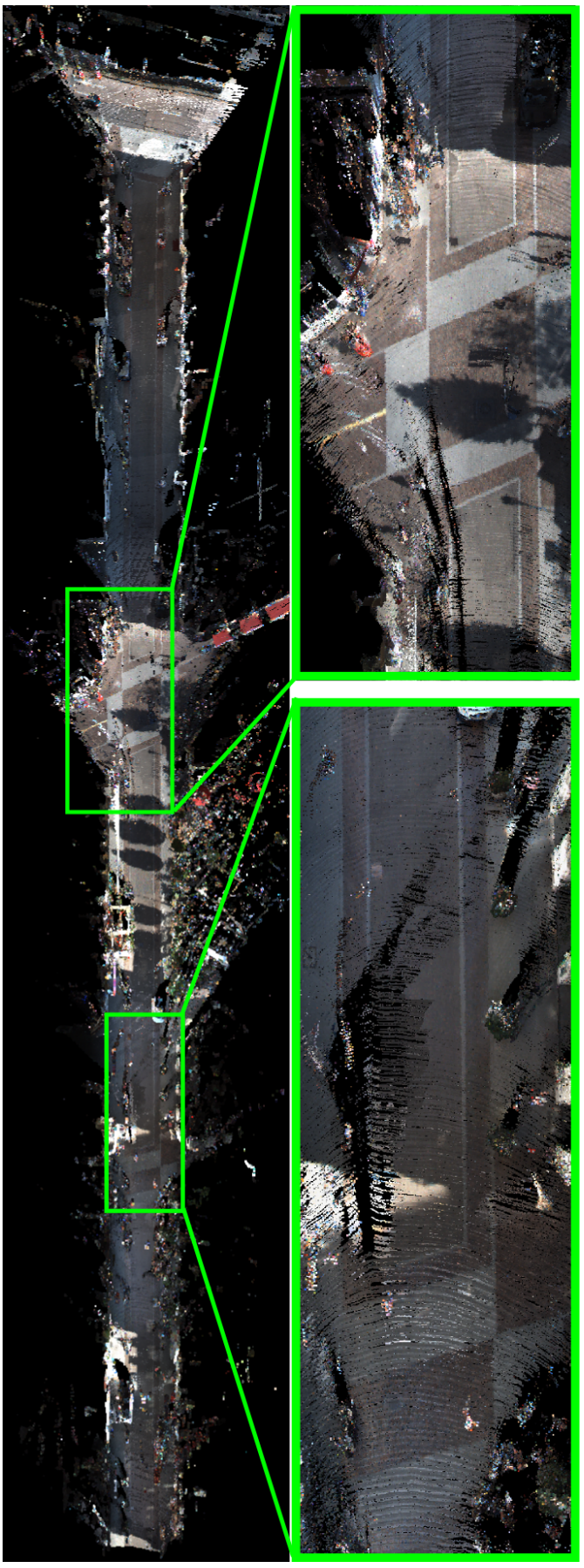


FIGURE 6.14 – Static scene 3D reconstruction of Market sequence using 3D-MOD: the static map is of significantly higher quality compared to Fig. 6.13.

6.6/ SUMMARY

We have proposed a complete pipeline for high quality reconstruction of dynamic objects using 2D-3D camera setup attached to a moving vehicle. Starting from the segmented motion trajectories of individual objects, we compute their precise motion parameters, register multiple sparse point clouds to increase their density, and develop a smooth and textured surface from the dense (but scattered) point cloud. The success of our method relies on the proposed optimization framework for accurate motion estimation between two sparse point clouds. Our formulation for fusing *closest-point* and *consensus*-based motion estimations, in the absence and the presence of motion trajectories respectively, is the key to obtain such accuracy.

Moreover, thanks to the successful motion detection and segmentation, moving objects are densely segmented using 3D region growing technique. After removing these dynamic objects, the static scene parts are used to reconstruct the static maps. Afterwards, high quality static maps are obtained by registering those static scene parts using the DW-ICP algorithm. Extensive experiments on both synthetic and real datasets have shown the efficiency of the proposed methods.

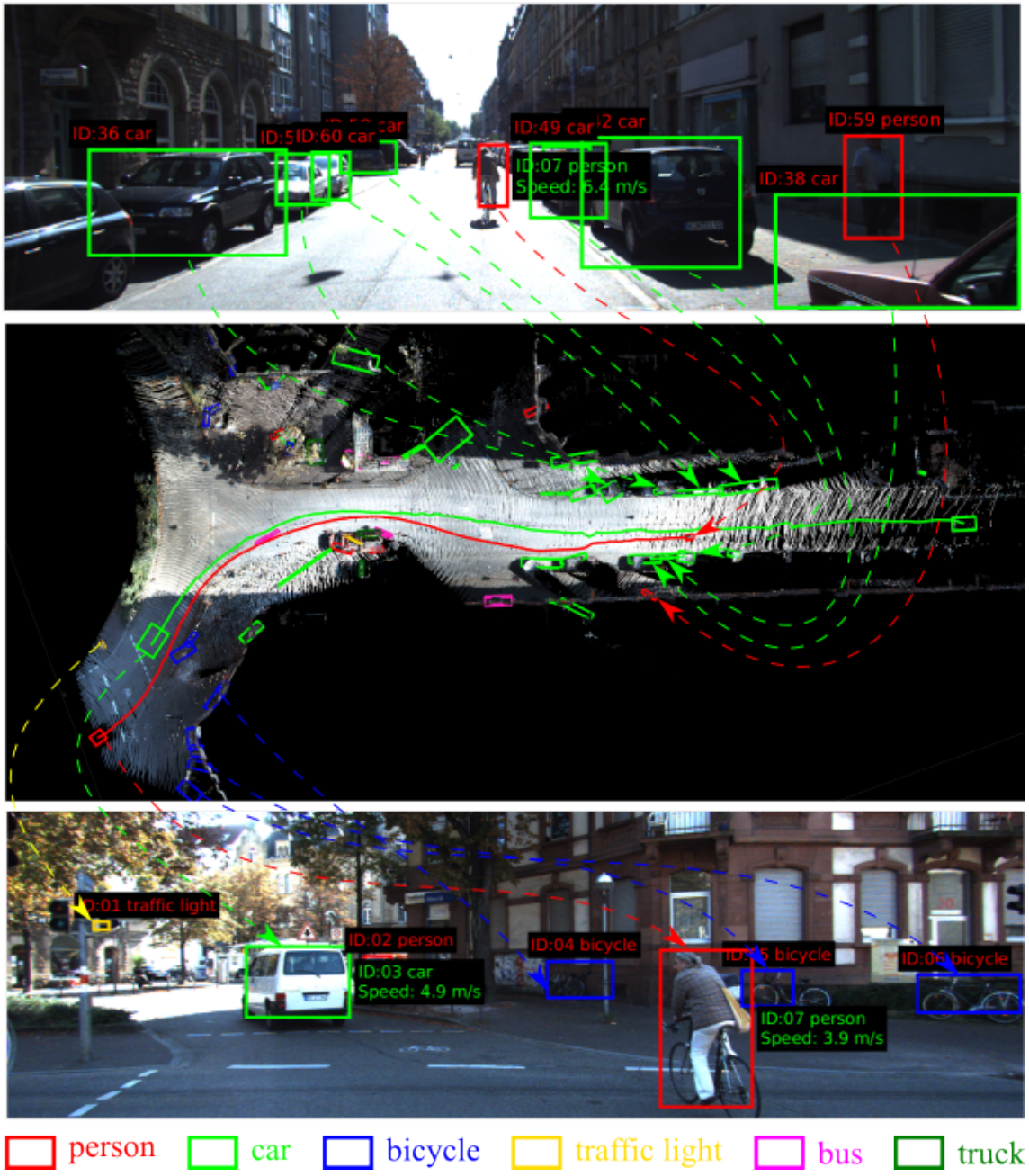


FIGURE 6.15 – Semantically labelled dynamic scene using 2D-to-3D label transfer: The top and bottom images are the last and the first semantically labelled images of Junction sequence. Middle image is the top view of our reconstructed dynamic scene with semantic labels. Dashed lines connect the objects in 3D map and 2D image. The solid red and green curves are the trajectories of the cyclist and the van, respectively (the remaining objects are static).

7

Conclusion and Future Work

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

- Alan Turing, *Computing Machinery and Intelligence*

In this thesis, we have studied two different cases of moving object detection and segmentation. When the camera motion is unknown, we show that the motion segmentation problem can be solved effectively by analysing the 3D feature tracks. Such problem is solved as a subspace clustering problem under the subspace self-representation assumption. In general, the subspace clustering problem can be formulated as an energy minimization problem in either a constrained or unconstrained manner with different regularization terms. In this regard, we propose the 3D Sparse Subspace Clustering (3D-SSC) approach under the subspace self-representation framework. Experiments confirm that the proposed 3D-SSC approach is accurate and robust. By incorporating the motion consistency regularization terms, the 3D SMooth Representation (3D-SMR) clustering method achieves comparative performances with significantly better computational efficiency. Moreover, our sampling of the sparse feature trajectories based on their flow likelihood allows the proposed motion segmentation to handle wide ranges of motions in terms of magnitude, speed and coverage. Our experiments illustrate the effectiveness of the incomplete trajectory construction, which is essential in many practical scenarios.

Although the proposed 3D-SSC and 3D-SMR achieve very satisfactory results, there still remain inherent drawbacks of such methods: (i) Feature trajectory construction is sensitive to the environment changes and noise. Feature tracking in itself is a very challenging problem in outdoor environments, therefore, a good compromise between trajectory

length and quality should be made. (ii) The adopted spectral clustering approach requires the number of moving objects as input, however, this is impractical for many real-world scenarios. Thus, it would be interesting to investigate an automatic spectral clustering method.

Regarding the cases of known camera motion, we illustrate that both the static and the dynamic objects can be analysed as a 3D Vector Field. Under the local motion consistency assumption, the motion flows are detected by locally analysing the spatial and the temporal displacement of the point clouds via Radon transform. We then present a novel 3D Sparse Flow Clustering approach relying on the self-representation property of flow subspaces and the spatial closeness constraints. By integrating the proposed algorithms, we introduce a robust, efficient and accurate framework for static map reconstruction. In many aspects, the proposed algorithms outperform the state-of-the-art methods on the comprehensive highly dynamic real-world KITTI datasets, in which they consistently exhibit better accuracies, lower misclassification and misdetection rates, and consequently offer great potential for very high quality 3D reconstructions of static maps as well as moving objects.

While the proposed algorithms are proved to be very effective, there remain gaps to be filled: (i) The flow field analysis algorithm relies on the linear local motion consistency assumption. Therefore, when the object has a pure rotation as well as small translation, it is very difficult to be detected. A step forward is to relax such a strong assumption to adapt the algorithm to more general cases. (ii) For the extreme cases, such as a partially observed planar object moving perpendicularly to the camera's principal axis, it becomes an ill-posed problem to detect such motions. Therefore, more information, such as texture, is required to overcome these difficulties.

Finally, we suggest a complete pipeline for high quality reconstruction of dynamic objects using a 2D-3D camera setup attached to a moving vehicle. Starting from the segmented motion trajectories of individual objects, we compute their precise motion parameters, register multiple sparse point clouds to increase the density, and develop a smooth and textured surface from the dense (but scattered) point cloud. The success of our method relies on the proposed optimization framework for accurate motion estimation between two sparse point clouds. Our formulation for fusing *closest-point* and *consensus*-based motion estimations, respectively in the absence and presence of motion trajectories, is

the key to obtain a high accuracy. Remarkably, thanks to the success of motion detection and segmentation, the moving objects are densely segmented using 3D region growing technique. On one hand, by removing these objects, only the static scene parts are used to reconstruct the static maps. On the other hand, photo-realistic multi-body 3D reconstructions are achieved by registering the independent moving object tracks. At last, the semantic labels are attached to the reconstructed static map for further scene understanding.

We have demonstrated the effectiveness of the proposed framework for high quality 3D map reconstruction with basic semantic labelling. As a future perspective, we aim to investigate a higher level scene analysis and understanding, e.g. interactive motion understanding and motion prediction of moving objects. Moreover, although the temporal static maps are reconstructed, the maintenance of the reconstructed maps over a long term observation (e.g. a few months or years) is still an open problem.

BIBLIOGRAPHIE

[1] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, page 0278364913491297, 2013.

[2] Cansen Jiang, Danda Pani Paudel, Yohan Fougerolle, David Fofi, and Cedric Demonceaux. Static-map and dynamic object reconstruction in outdoor scenes using 3-d motion segmentation. *IEEE Robotics and Automation Letters (RAL) (Invited presentation at ICRA'16, Stockholm, Swedden)*, 1(1):324–331, Jan. 2016.

[3] Cansen Jiang, Danda Pani Paudel, Yohan Fougerolle, David Fofi, and Cedric Demonceaux. Reconstruction 3d de scènes dynamiques par segmentation au sens du mouvement. In *Le 20èmé congrès national sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA)*. Clermont-Ferrand, France, Jun. 2016.

[4] Dennis Christie, Cansen Jiang, Danda Paudel, and Cédric Demonceaux. 3d reconstruction of dynamic vehicles using sparse 3d-laser-scanner and 2d image fusion. In *Informatics and Computing (ICIC), International Conference on*, pages 61–65. IEEE, 2016.

[5] Cansen Jiang, Danda Pani Paudel, Yohan Fougerolle, David Fofi, and Cedric Demonceaux. Incomplete 3d motion trajectory segmentation and 2d-to-3d label transfer for dynamic scene analysis. In *IEEE Internatioal Conferenece on Inteligent Robot and System (IROS)*. Vancouver, Canada, Sept. 2017.

[6] Cansen Jiang, Dennis Christie, Danda Pani Paudel, and Cedric Demonceaux. High quality reconstruction of dynamic objects using 2d-3d camera fusion. In *IEEE International Conference on Image Processing (ICIP)*. Beijing, China, Sept. 2017.

[7] Cansen Jiang, Yohan Fougerolle, David Fofi, and Cedric Demonceaux. Dynamic 3d scene reconstruction and enhancement. In *IAPR International Conference on Image Analysis and Processing (ICIAP)*, pages 469–479. Catania, Italy, Sept. 2017.

[8] Cansen Jiang, Danda Pani Paudel, Yohan Fougerolle, David Fofi, and Cedric Demonceaux. Static and dynamic objects analysis as a 3d vector field. In *IEEE International Conference on 3D Vision (3DV)*. Qingdao, China, Oct. 2017.

[9] J.O. Limb and J.A. Murphy. Estimating the velocity of moving images in television signals. *Computer Graphics and Image Processing*, 4(4):311 – 327, 1975.

[10] Murat Kunt, Athanassios Ikonomopoulos, and Michel Kocher. Second-generation image-coding techniques. *Proceedings of the IEEE*, 73(4):549–574, 1985.

[11] Michal Irani and P Anandan. A unified approach to moving object detection in 2d and 3d scenes. *IEEE transactions on pattern analysis and machine intelligence*, 20(6):577–589, 1998.

[12] Oxford English Dictionary. Oxford english dictionary, 2003.

[13] Julian FY Cheung, Michael C Wicks, Gerard J Genello, and Ludwik Kurz. A statistical theory for optimal detection of moving objects in variable corruptive noise. *IEEE transactions on image processing*, 8(12):1772–1787, 1999.

[14] Thanarat Horprasert, David Harwood, and Larry S Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *IEEE ICCV*, volume 99, pages 1–19. Citeseer, 1999.

[15] Liyuan Li, Weimin Huang, Irene Yu-Hua Gu, and Qi Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004.

[16] Ahmed Elgammal, Ramani Duraiswami, David Harwood, and Larry S Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, 2002.

[17] Massimo Piccardi. Background subtraction techniques: a review. In *Systems, man and cybernetics, 2004 IEEE international conference on*, volume 4, pages 3099–3104. IEEE, 2004.

[18] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[19] Norbert Diehl. Object-oriented motion estimation and segmentation in image sequences. *Signal processing: Image communication*, 3(1):23–56, 1991.

[20] Michael Hötter and Robert Thoma. Image segmentation based on object oriented mapping parameter estimation. *Signal processing*, 15(3):315–334, 1988.

[21] Roland Mech and Michael Wollborn. A noise robust method for 2d shape estimation of moving objects in video sequences considering a moving camera. *Signal Processing*, 66(2):203–217, 1998.

[22] John YA Wang and Edward H Adelson. Layered representation for motion analysis. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*, pages 361–366. IEEE, 1993.

[23] Edward H Adelson. *Layered representations for image coding*. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1991.

[24] Yuxin Jin, Linmi Tao, Huijun Di, Naveed I Rao, and Guangyou Xu. Background modeling from a free-moving camera by multi-layer homography algorithm. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1572–1575. IEEE, 2008.

[25] Michal Irani, Benny Rousso, and Shmuel Peleg. Detecting and tracking multiple moving objects using temporal integration. In *European Conference on Computer Vision*, pages 282–287. Springer, 1992.

[26] AG Amitha Perera, Glen Brooksby, Anthony Hoogs, and Gianfranco Doretto. Moving object segmentation using scene understanding. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 201–201. IEEE, 2006.

[27] Chang Yuan, Gerard Medioni, Jinman Kang, and Isaac Cohen. Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints. *IEEE transactions on pattern analysis and machine intelligence*, 29(9), 2007.

[28] Rene Vidal, Yi Ma, and Shankar Sastry. Generalized principal component analysis (gpca). *IEEE transactions on pattern analysis and machine intelligence*, 27(12):1945–1959, 2005.

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

[30] Han Hu, Zhouchen Lin, Jianjiang Feng, and Jie Zhou. Smooth representation clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3834–3841, 2014.

[31] Xiaowei Zhou, Can Yang, and Weichuan Yu. Moving object detection by detecting contiguous outliers in the low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):597–610, 2013.

[32] Moein Shakeri and Hong Zhang. Corola: a sequential solution to moving object detection using low-rank approximation. *Computer Vision and Image Understanding*, 146:27–39, 2016.

[33] Guofeng Zhang, Jiaya Jia, Wei Xiong, Tien-Tsin Wong, Pheng-Ann Heng, and Hujun Bao. Moving object extraction with a hand-held camera. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[34] Brian L Price, Bryan S Morse, and Scott Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 779–786. IEEE, 2009.

[35] Kimin Yun, Jongin Lim, and Jin Young Choi. Scene conditional background update for moving object detection in a moving camera. *Pattern Recognition Letters*, 88:57–63, 2017.

[36] AG Amitha Perera, Chukka Srinivas, Anthony Hoogs, Glen Brooksby, and Wensheng Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 666–673. IEEE, 2006.

[37] Ninad Thakoor and Jean X Gao. Automatic video object extraction with camera in motion. *International Journal of Image and Graphics*, 8(04):573–600, 2008.

[38] Isaac Cohen and Gerard Medioni. Detecting and tracking moving objects for video surveillance. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 319–325. IEEE, 1999.

[39] Pietro Azzari, Luigi Di Stefano, and Alessandro Bevilacqua. An effective real-time mosaicing algorithm apt to detect motion through background subtraction using a ptz camera. In *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, pages 511–516. IEEE, 2005.

[40] Seon Joo Kim, Gianfranco Doretto, Jens Rittscher, Peter Tu, Nils Krahnstoever, and Marc Pollefeys. A model change detection approach to dynamic scene modeling. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 490–495. IEEE, 2009.

[41] Constant Guillot, Maxime Taron, Patrick Sayd, Quoc Cuong Pham, Christophe Tilmant, and Jean-Marc Lavest. Background subtraction adapted to ptz cameras by keypoint density estimation. In *Proceedings of the British Machine Vision Conference*, pages 34–1, 2010.

[42] Isaac Cohen and Gérard Medioni. Detecting and tracking moving objects in video from an airborne observer. In *Proc. IEEE Image Understanding Workshop*, volume 1, pages 217–222, 1998.

[43] Pablo O Arambel, Jeffrey Silver, Matthew Antone, and Thomas Strat. Signature-aided air-to-ground video tracking. In *Information Fusion, 2006 9th International Conference on*, pages 1–8. IEEE, 2006.

[44] Andrew P Brown, Kevin J Sullivan, and David J Miller. Feature-aided multiple target tracking in the image plane. In *Proc. SPIE*, volume 6229, 2006.

[45] Yu-chia Chung and Zhihai He. Low-complexity and reliable moving objects detection and tracking for aerial video surveillance with small uavs. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pages 2670–2673. IEEE, 2007.

[46] Chung-Hsien Huang, Yi-Ta Wu, Jau-Hong Kao, Ming-Yu Shih, and Cheng-Chuan Chou. A hybrid moving object detection method for aerial images. *Advances in Multimedia Information Processing-PCM 2010*, pages 357–368, 2010.

[47] Boyoon Jung and Gaurav S Sukhatme. Real-time motion tracking from a mobile robot. *International Journal of Social Robotics*, 2(1):63–78, 2010.

[48] Slim Amri, Walid Barhoumi, and Ezzeddine Zagrouba. A robust framework for joint background/foreground segmentation of complex video scenes filmed with freely moving camera. *Multimedia Tools and Applications*, 46(2-3):175–205, 2010.

[49] Soo Wan Kim, Kimin Yun, Kwang Moo Yi, Sun Jung Kim, and Jin Young Choi. Detection of moving objects with a moving camera using non-panoramic background model. *Machine vision and applications*, 24(5):1015–1028, 2013.

[50] Tomasz Kryjak, Mateusz Komorkiewicz, and Marek Gorgon. Real-time implementation of foreground object detection from a moving camera using the vibe algorithm. *Computer Science and Information Systems*, 11(4):1617–1637, 2014.

[51] Serge Ayer, Philippe Schroeter, and Josef Bigün. Segmentation of moving objects by robust motion parameter estimation over multiple frames. *Computer Vision—ECCV'94*, pages 316–327, 1994.

[52] Munchurl Kim, Jae Gark Choi, Daehee Kim, Hyung Lee, Myoung Ho Lee, Chieteuk Ahn, and Yo-Sung Ho. A vop generation tool: automatic segmentation of moving objects in image sequences based on spatio-temporal information. *IEEE transactions on circuits and systems for video technology*, 9(8):1216–1226, 1999.

[53] Jurgen Stander, Roland Mech, and Jörn Ostermann. Detection of moving cast shadows for object segmentation. *IEEE Transactions on multimedia*, 1(1):65–76, 1999.

[54] Ross Cutler and Larry Davis. Real-time periodic motion detection, analysis, and applications. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 326–332. IEEE, 1999.

[55] Malavika Bhaskaranand and Sitaram Bhagavathy. Motion-based object segmentation using frame alignment and consensus filtering. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2245–2248. IEEE, 2010.

[56] Xu Zhang, Shengjin Wang, and Xiaoqing Ding. Beyond dominant plane assumption: Moving objects detection in severe dynamic scenes with multi-classes ransac. In *Audio, Language and Image Processing (ICALIP), 2012 International Conference on*, pages 822–827. IEEE, 2012.

[57] Huaxin Xiao, Yu Liu, Wei Wang, and Maojun Zhang. Motion detection algorithm for unmanned aerial vehicle nighttime surveillance. *IEICE TRANSACTIONS on Information and Systems*, 97(12):3248–3251, 2014.

[58] Til Aach, André Kaup, and Rudolf Mester. Statistical model-based change detection in moving video. *Signal processing*, 31(2):165–180, 1993.

[59] Yaser Sheikh and Mubarak Shah. Bayesian object detection in dynamic scenes. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 74–79. IEEE, 2005.

[60] Kwang Moo Yi, Kimin Yun, Soo Wan Kim, Hyung Jin Chang, and Jin Young Choi. Detection of moving objects with non-stationary cameras in 5.8 ms: Bringing motion detection to your mobile device. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 27–34, 2013.

[61] Amitha Viswanath, Reena Kumari Behera, Vinuchackravarthy Senthamilarasu, and Krishnan Kutty. Background modelling from a moving camera. *Procedia Computer Science*, 58:289–296, 2015.

[62] Anurag Mittal and Dan Huttenlocher. Scene modeling for wide area surveillance and image synthesis. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 160–167. IEEE, 2000.

[63] Ying Ren, Chin-Seng Chua, and Yeong-Khing Ho. Motion detection with nonstationary background. *Machine Vision and Applications*, 13(5-6):332–343, 2003.

[64] Naveed Iqbal Rao, Huijun Di, and Guangyou Xu. Joint correspondence and background modeling based on tree dynamic programming. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 425–428. IEEE, 2006.

[65] Rozenn Dahyot. Unsupervised camera motion estimation and moving object detection in videos. 2006.

[66] Naveed I Rao, Huijun Di, and GuangYou Xu. Panoramic background model under free moving camera. In *Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on*, volume 1, pages 639–643. IEEE, 2007.

[67] Aryo Wiman Nur Ibrahim, Pang Wee Ching, GL Gerald Seet, WS Michael Lau, and Witold Czajewski. Moving objects detection and tracking framework for uav-based surveillance. In *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, pages 456–461. IEEE, 2010.

[68] Xiaoyong Zhang, Masahide Abe, and Masayuki Kawamata. Motion detection in old film sequences using adaptive gaussian mixture model. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2337–2340. IEEE, 2011.

[69] Yuki Hishinuma, Tomoyuki Suzuki, Kazuki Nakagami, and Takao Nishitani. Transformed domain gmm foreground segmentation for mobile video camera. In *Image*

Processing (ICIP), 2010 17th IEEE International Conference on, pages 2217–2220. IEEE, 2010.

[70] Omar Oreifej, Xin Li, and Mubarak Shah. Simultaneous video stabilization and moving object detection in turbulence. *IEEE transactions on pattern analysis and machine intelligence*, 35(2):450–462, 2013.

[71] Moein Shakeri and Hong Zhang. Detection of small moving objects using a moving camera. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2777–2782. IEEE, 2014.

[72] Yinhui Zhang, Mohamed Abdel-Mottaleb, and Zifen He. Unsupervised segmentation of highly dynamic scenes through global optimization of multiscale cues. *Pattern Recognition*, 48(11):3477–3487, 2015.

[73] Kimin Yun and Jin Young Choi. Robust and fast moving object detection in a non-stationary camera via foreground probability based sampling. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 4897–4901. IEEE, 2015.

[74] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[75] Rozenn Dahyot, Pierre Charbonnier, and Fabrice Heitz. Unsupervised statistical detection of changing objects in camera-in-motion video. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 1, pages 638–641. IEEE, 2001.

[76] Boyoon Jung and Gaurav S Sukhatme. Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In *International Conference on Intelligent Autonomous Systems*, pages 980–987, 2004.

[77] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[78] Anurag Mittal and Nikos Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. Ieee, 2004.

[79] Daniel Szolgay, Jenny Benois-Pineau, Rémi Mégret, Yann Gaëstel, and J-F Dartigues. Detection of moving foreground objects in videos with strong camera motion. *Pattern Analysis and Applications*, 14(3):311–328, 2011.

[80] Jihong Min, Hyeongwoo Kim, Jongwon Choi, and In So Kweon. A superpixel mrf approach using high-order likelihood for moving object detection. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 266–269. IEEE, 2012.

[81] Cédric Archambeau, Maurizio Valle, Alex Assenza, and Michel Verleysen. Assessment of probability density estimation methods: Parzen window and finite gaussian mixtures. In *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, pages 4–pp. IEEE, 2006.

[82] Vassiliy A Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, 14(1):153–158, 1969.

[83] Zhaozheng Yin and Robert Collins. Belief propagation in a 3d spatio-temporal mrf for moving object detection. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[84] Xun Xu and Thomas S Huang. A loopy belief propagation approach for robust background estimation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7. IEEE, 2008.

[85] Suha Kwak, Taegyu Lim, Woonhyun Nam, Bohyung Han, and Joon Hee Han. Generalized background subtraction based on hybrid inference by belief propagation and bayesian filtering. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2174–2181. IEEE, 2011.

[86] WonTaek Chung, YongHyun Kim, Yong-Joong Kim, and DaiJin Kim. A two-stage foreground propagation for moving object detection in a non-stationary. In *Advanced Video and Signal Based Surveillance (AVSS), 2016 13th IEEE International Conference on*, pages 187–193. IEEE, 2016.

[87] Shih-Ping Liou and Ramesh C Jain. Motion detection in spatio-temporal space. *Computer Vision, Graphics, and Image Processing*, 45(2):227–250, 1989.

[88] Frederic Dufaux, Fabrice Moscheni, and Andrew Lippman. Spatio-temporal segmentation based on motion and static segmentation. In *Image Processing, 1995. Proceedings., International Conference on*, volume 1, pages 306–309. IEEE, 1995.

[89] Liang-Hua Chen, Yu-Chun Lai, Chih-Wen Su, and Hong-Yuan Mark Liao. Extraction of video object with complex motion. *Pattern Recognition Letters*, 25(11):1285–1291, 2004.

[90] Chang Liu, Pong C Yuen, and Guoping Qiu. Object motion detection using information theoretic spatio-temporal saliency. *Pattern Recognition*, 42(11):2897–2906, 2009.

[91] Konstantinos G Derpanis and Richard P Wildes. Early spatiotemporal grouping with a distributed oriented energy representation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 232–239. IEEE, 2009.

[92] Vijay Mahadevan and Nuno Vasconcelos. Spatiotemporal saliency in dynamic scenes. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):171–177, 2010.

[93] Saad Ali and Mubarak Shah. Cocoa: tracking in aerial imagery. In *Proceedings of SPIE*, volume 6209, pages 118–123, 2006.

[94] Bastian Leibe, Konrad Schindler, Nico Cornelis, and Luc Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *IEEE transactions on pattern analysis and machine intelligence*, 30(10):1683–1698, 2008.

[95] Gonzalo R Rodríguez-Canosa, Stephen Thomas, Jaime del Cerro, Antonio Barrientos, and Bruce MacDonald. A real-time method to detect and track moving objects (datmo) from unmanned aerial vehicles (uavs) using a single camera. *Remote Sensing*, 4(4):1090–1111, 2012.

[96] Hao Shen, Shuxiao Li, Jinglan Zhang, and Hongxing Chang. Tracking-based moving object detection. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 3093–3097. IEEE, 2013.

[97] Patrick Bouthemy and Edouard François. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *International Journal of Computer Vision*, 10(2):157–182, 1993.

[98] Yu Liu, Huaxin Xiao, Zheng Zhang, Wei Xu, Maojun Zhang, and Jianguo Zhang. Data separation of l1-minimization for real-time motion detection. *Sciences*, 1(1):143–168, 2008.

[99] John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex

optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009.

[100] Agwad ElTantawy and Mohamed S Shehata. Moving object detection from moving platforms using lagrange multiplier. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2586–2590. IEEE, 2015.

[101] Wenrui Hu, Yehui Yang, Wensheng Zhang, and Yuan Xie. Moving object detection using tensor-based low-rank and saliently fused-sparse decomposition. *IEEE Transactions on Image Processing*, 26(2):724–737, 2017.

[102] Xianbiao Shu and Narendra Ahuja. Imaging via three-dimensional compressive sampling (3dcs). In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 439–446. IEEE, 2011.

[103] J-C Lee, Bing J Sheu, W-C Fang, and Rama Chellappa. Vlsi neuroprocessors for video motion detection. *IEEE Transactions on Neural Networks*, 4(2):178–191, 1993.

[104] Yasuyuki Sugaya and Ken-ichi Kanatani. Geometric structure of degeneracy for multi-body motion segmentation. In *ECCV Workshop SMVP*, pages 13–25. Springer, 2004.

[105] M Pawan Kumar, Philip HS Torr, and Andrew Zisserman. Learning layered motion segmentations of video. *International Journal of Computer Vision*, 76(3):301–319, 2008.

[106] Artem Rozantsev, Vincent Lepetit, and Pascal Fua. Flying objects detection from a single moving camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4128–4136, 2015.

[107] Artem Rozantsev, Vincent Lepetit, and Pascal Fua. Detecting flying objects using a single moving camera. *IEEE transactions on pattern analysis and machine intelligence*, 39(5):879–892, 2017.

[108] Yinhui Zhang and Zifen He. Semantic motion signature for segmentation of high speed large displacement objects. *IEICE TRANSACTIONS on Information and Systems*, 100(1):220–224, 2017.

[109] Feng Liu and Michael Gleicher. Learning color and locality cues for moving object detection and segmentation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 320–327. IEEE, 2009.

[110] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69, 2008.

[111] Dariu M Gavrila. Pedestrian detection from a moving vehicle. In *European conference on computer vision*, pages 37–49. Springer, 2000.

[112] Margrit Betke, Esin Haritaoglu, and Larry S Davis. Real-time multiple vehicle detection and tracking from a moving vehicle. *Machine vision and applications*, 12(2):69–83, 2000.

[113] Simon Baker, Richard Szeliski, and P Anandan. A layered approach to stereo reconstruction. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 434–441. IEEE, 1998.

[114] Edward H Adelson. Layered representation for image coding, January 6 1998. US Patent 5,706,417.

[115] Michal Irani, Benny Rousso, and Shmuel Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, 1994.

[116] Christian Micheloni, Gian Luca Foresti, and Flavio Alberti. A new feature clustering method for object detection with an active camera. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 4, pages 2587–2590. IEEE, 2004.

[117] Yaser Sheikh, Omar Javed, and Takeo Kanade. Background subtraction for freely moving cameras. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1219–1225. IEEE, 2009.

[118] Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1846–1853. IEEE, 2012.

[119] Ali Elqursh and Ahmed Elgammal. Online moving camera background subtraction. *Computer Vision–ECCV 2012*, pages 228–241, 2012.

[120] Shih-Wei Sun, Yu-Chiang Frank Wang, Fay Huang, and Hong-Yuan Mark Liao. Moving foreground object detection via robust sift trajectories. *Journal of Visual Communication and Image Representation*, 24(3):232–243, 2013.

[121] Bahareh Kalantar, Shattri Bin Mansor, Alfian Abdul Halin, Helmi Zulhaidi Mohd Shafri, and Mohsen Zand. Multiple moving object detection from uav videos using tra-

jectories of matched regional adjacency graphs. *IEEE Transactions on Geoscience and Remote Sensing*, 2017.

[122] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011.

[123] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[124] Patrik Sundberg, Thomas Brox, Michael Maire, Pablo Arbeláez, and Jitendra Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2233–2240. IEEE, 2011.

[125] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[126] William B Thompson and Ting-Chuen Pong. Detecting moving objects. *International journal of computer vision*, 4(1):39–57, 1990.

[127] Randal C Nelson. Qualitative detection of motion by a moving observer. *International journal of computer vision*, 7(1):33–46, 1991.

[128] Jean-Marc Odobez and Patrick Bouthemy. Detection of multiple moving objects using multiscale mrf with camera motion compensation. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 2, pages 257–261. IEEE, 1994.

[129] Shoichi Araki, Takashi Matsuoka, Haruo Takemura, and Naokazu Yokoya. Real-time tracking of multiple moving objects in moving camera image sequences using robust statistics. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 2, pages 1433–1435. IEEE, 1998.

[130] Thomas Meier and King Ngi Ngan. Automatic segmentation of moving objects for video object plane generation. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):525–538, 1998.

[131] Marc Gelgon and Patrick Bouthemy. A region-level motion-based graph representation and labeling for tracking a spatial image partition. *Pattern Recognition*, 33(4):725–740, 2000.

[132] Ashit Talukder, S Goldberg, Larry Matthies, and Adnan Ansar. Real-time detection of moving objects in a dynamic scene from moving robotic vehicles. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1308–1313. IEEE, 2003.

[133] Felix Woelk and Reinhard Koch. Fast monocular bayesian detection of independently moving objects by a moving observer. In *Joint Pattern Recognition Symposium*, pages 27–35. Springer, 2004.

[134] Michael Kellner and Tobias Hanning. Motion detection based on contour strings. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 4, pages 2599–2602. IEEE, 2004.

[135] Thomas Veit, Frédéric Cao, and Patrick Bouthemy. An a contrario decision framework for region-based motion detection. *International journal of computer vision*, 68(2):163–178, 2006.

[136] Yan Zhang, Stephen J Kiselewich, William A Bauson, and Riad Hammoud. Robust moving object detection at distance in the visible spectrum and beyond using a moving camera. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 131–131. IEEE, 2006.

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

[138] Vishal Jain, Benjamin B Kimia, and Joseph L Mundy. Segregation of moving objects using elastic matching. *Computer Vision and Image Understanding*, 108(3):230–242, 2007.

[139] Jinsong Wang, Nilesh V Patel, William I Grosky, and Farshad Fotouhi. Moving camera moving object segmentation in compressed video sequences. *International Journal of Image and Graphics*, 9(04):609–627, 2009.

[140] Konstantinos G Derpanis and Richard P Wildes. Detecting spatiotemporal structure boundaries: Beyond motion discontinuities. In *Asian Conference on Computer Vision*, pages 301–312. Springer, 2009.

[141] Chiranjoy Chattopadhyay and Sukhendu Das. Prominent moving object segmentation from moving camera video shots using iterative energy minimization. *Signal, Image and Video Processing*, 9(8):1927–1934, 2015.

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

[143] John L Barron, David J Fleet, and Steven S Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994.

[144] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[145] Hicham Sekkati and Amar Mitiche. Joint optical flow estimation, segmentation, and 3d interpretation with level sets. *Computer Vision and Image Understanding*, 103(2):89–100, 2006.

[146] Thomas Brox and Joachim Weickert. Level set segmentation with multiple regions. *IEEE Transactions on Image Processing*, 15(10):3213–3218, 2006.

[147] Michael M Chang, A Murat Tekalp, and M Ibrahim Sezan. Simultaneous motion estimation and segmentation. *IEEE transactions on image processing*, 6(9):1326–1333, 1997.

[148] Tomer Amiaz and Nahum Kiryati. Dense discontinuous optical flow via contour-based segmentation. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages III–1264. IEEE, 2005.

[149] Etienne Mémin and Patrick Pérez. Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, 46(2):129–155, 2002.

[150] Philip HS Torr and David W Murray. Stochastic motion clustering. In *European Conference on Computer Vision*, pages 328–337. Springer, 1994.

[151] Yasuyuki Sugaya and Kenichi Kanatani. Extracting moving objects from a moving camera video sequence. *Memoirs of the Faculty of Engineering, Okayama University*, 39(1):56–62, 2005.

[152] Jinman Kang, Isaac Cohen, Gérard Medioni, and Chang Yuan. Detection and tracking of moving objects from a moving platform in presence of strong parallax. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 10–17. IEEE, 2005.

[153] Jing Zhang, Fanhuai Shi, and Yuncai Liu. Motion segmentation by multibody trifocal tensor using line correspondence. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 599–602. IEEE, 2006.

[154] Ying Piao and Jun Sato. Space-time invariants for 3d motions from projective cameras. In *Asian Conference on Computer Vision*, pages 811–821. Springer, 2006.

[155] Koichiro Yamaguchi, Takeo Kato, and Yoshiki Ninomiya. Vehicle ego-motion estimation and moving object detection using a monocular camera. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, pages 610–613. IEEE, 2006.

[156] Abhijit Kundu, K Madhava Krishna, and Jayanthi Sivaswamy. Moving object detection by multi-view geometric techniques from a single camera mounted robot. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4306–4312. IEEE, 2009.

[157] Jérémy Huart, Guillaume Foret, and Pascal Bertolino. Moving object extraction with a localized pyramid. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 783–786. IEEE, 2004.

[158] Bin Kang and Wei-Ping Zhu. Robust moving object detection using compressed sensing. *IET Image Processing*, 9(9):811–819, 2015.

[159] Edouard Francois and Patrick Bouthemy. Multiframe-based identification of mobile components of a scene with a moving camera. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 166–172. IEEE, 1991.

[160] Josh Wills, Sameer Agarwal, and Serge Belongie. What went where [motion segmentation]. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, volume 1. IEEE, 2003.

[161] Daniel Cremers and Stefano Soatto. Motion competition: A variational approach to piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, 2005.

[162] Thomas Brox, Andrés Bruhn, and Joachim Weickert. Variational motion segmentation with level sets. *Computer Vision–ECCV 2006*, pages 471–483, 2006.

[163] Joseph Weber and Jitendra Malik. Rigid body segmentation and shape description from dense optical flow under weak perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):139–143, 1997.

[164] Joseph Weber and J Malik. *Scene partitioning via statistic-based region growing*. Computer Science Division (EECS), University of California, 1994.

[165] René Vidal, Yi Ma, Stefano Soatto, and Shankar Sastry. Two-view multibody structure from motion. *International Journal of Computer Vision*, 68(1):7–25, 2006.

[166] Abhijit Kundu, CV Jawahar, and K Madhava Krishna. Realtime moving object detection from a freely moving monocular camera. In *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*, pages 1635–1640. IEEE, 2010.

[167] Zhuwen Li, Jiaming Guo, Loong-Fah Cheong, and Steven Zhiying Zhou. Perspective motion segmentation via collaborative clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1369–1376, 2013.

[168] Ninad Thakoor and Jean Gao. Branch-and-bound hypothesis selection for two-view multiple structure and motion segmentation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–6. IEEE, 2008.

[169] Ninad Thakoor, Jean Gao, and Venkat Devarajan. Multibody structure-and-motion segmentation by branch-and-bound model selection. *IEEE Transactions on Image Processing*, 19(6):1393–1402, 2010.

[170] Heechul Jung, Jeongwoo Ju, and Junmo Kim. Rigid motion segmentation using randomized voting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1210–1217, 2014.

[171] Ralf Dragon, Bodo Rosenhahn, and Jörn Ostermann. Multi-scale clustering of frame-to-frame correspondences for motion segmentation. *Computer Vision–ECCV 2012*, pages 445–458, 2012.

[172] Ralf Dragon, Jörn Ostermann, and Luc Van Gool. Robust realtime motion-split-and-merge for motion segmentation. In *German Conference on Pattern Recognition*, pages 425–434. Springer, 2013.

[173] Roberto Toldo and Andrea Fusiello. Robust multiple structures estimation with j-linkage. *Computer Vision–ECCV 2008*, pages 537–547, 2008.

[174] Shankar R Rao, Allen Y Yang, Andrew W Wagner, and Yi Ma. Segmentation of hybrid motions via hybrid quadratic surface analysis. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 2–9. IEEE, 2005.

[175] Ali Elqursh and Ahmed Elgammal. Online motion segmentation using dynamic label propagation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2008–2015, 2013.

[176] Dong Zhang and Ping Li. Motion detection for rapidly moving cameras in fully 3d scenes. In *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, pages 444–449. IEEE, 2010.

[177] Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2790–2797. IEEE, 2009.

[178] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[179] Terrance E Boult and L Gottesfeld Brown. Factorization-based segmentation of motions. In *Visual Motion, 1991., Proceedings of the IEEE Workshop on*, pages 179–186. IEEE, 1991.

[180] Joao Costeira and Takeo Kanade. A multi-body factorization method for motion analysis. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 1071–1076. IEEE, 1995.

[181] João Paulo Costeira and Takeo Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[182] C William Gear. Multibody grouping from motion images. *International Journal of Computer Vision*, 29(2):133–150, 1998.

[183] Gilbert Strang, Gilbert Strang, Gilbert Strang, and Gilbert Strang. *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.

[184] Kenichi Kanatani. Evaluation and selection of models for motion segmentation. *Computer Vision—ECCV 2002*, pages 33–42, 2002.

[185] Kenichi Kanatani. Geometric information criterion for model selection. *International Journal of Computer Vision*, 26(3):171–189, 1998.

[186] Ken-ichi Kanatani. Motion segmentation by subspace separation and model selection. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 586–591. IEEE, 2001.

[187] Pan Ji, Mathieu Salzmann, and Hongdong Li. Shape interaction matrix revisited and robustified: Efficient subspace clustering with corrupted and incomplete data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4687–4695, 2015.

[188] Ying Wu, Zhengyou Zhang, Thomas S Huang, and John Y Lin. Multibody grouping via orthogonal subspace decomposition. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2001.

[189] Naoyuki Ichimura. Motion segmentation based on factorization method and discriminant criterion. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 600–605. IEEE, 1999.

[190] Naoyuki Ichimura. Motion segmentation using feature selection and subspace method based on shape space. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 3, pages 850–856. IEEE, 2000.

[191] Lihi Zelnik-Manor and Michal Irani. Degeneracies, dependencies and their implications in multi-body and multi-sequence factorizations. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–287. IEEE, 2003.

[192] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

[193] Anil M Cheriyadat and Richard J Radke. Non-negative matrix factorization of partial track data for motion segmentation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 865–872. IEEE, 2009.

[194] Quanyi Mo and Bruce A Draper. Semi-nonnegative matrix factorization for motion segmentation with missing data. In *European Conference on Computer Vision*, pages 402–415. Springer, 2012.

[195] René Vidal, Yi Ma, and Shankar Sastry. Generalized principal component analysis (gpca). In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, volume 1. IEEE, 2003.

[196] René Vidal, Yi Ma, and Jacopo Piazzi. A new gpca algorithm for clustering subspaces by fitting, differentiating and dividing polynomials. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, volume 1. IEEE, 2004.

[197] Allen Y Yang, Shankar R Rao, and Yi Ma. Robust statistical estimation and segmentation of multiple subspaces. In *Computer Vision and Pattern Recognition Workshop, Conference on*, pages 99–107. IEEE, 2006.

[198] Shankar R Rao, Allen Y Yang, S Shankar Sastry, and Yi Ma. Robust algebraic segmentation of mixed rigid-body and planar motions from two views. *International journal of computer vision*, 88(3):425–446, 2010.

[199] Jeffrey Ho, Ming-Husang Yang, Jongwoo Lim, Kuang-Chih Lee, and David Kriegman. Clustering appearances of objects under varying illumination conditions. In *Computer vision and pattern recognition, IEEE computer society conference on*, volume 1. IEEE, 2003.

[200] Kenichi Kanatani and Yasuyuki Sugaya. Multi-stage optimization for multi-body motion segmentation. In *Australia-Japan Advanced Workshop on Computer Vision*, volume 2, page 7, 2003.

[201] Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European conference on computer vision*, pages 94–106. Springer, 2006.

[202] Amnon Shashua, Ron Zass, and Tamir Hazan. Multi-way clustering using super-symmetric non-negative tensor factorization. *Computer Vision–ECCV 2006*, pages 595–608, 2006.

[203] Guangliang Chen and Gilad Lerman. Spectral curvature clustering (scc). *International Journal of Computer Vision*, 81(3):317–330, 2009.

[204] Pulak Purkait, Tat-Jun Chin, Alireza Sadri, and David Suter. Clustering with hypergraphs: the case for large hyperedges. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1697–1711, 2017.

[205] Suraj Jain and Venu Madhav Govindu. Efficient higher-order clustering on the grassmann manifold. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3511–3518, 2013.

[206] Tat-Jun Chin, Hanzi Wang, and David Suter. The ordered residual kernel for robust motion subspace clustering. In *Advances in neural information processing systems*, pages 333–341, 2009.

[207] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[208] Venu Madhav Govindu. A tensor decomposition for geometric grouping and segmentation. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 1150–1157. IEEE, 2005.

[209] Konrad Schindler. Spatially consistent 3d motion segmentation. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages III–409. IEEE, 2005.

[210] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicuts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3271–3279, 2015.

[211] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[212] Michael E Tipping and Christopher M Bishop. Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482, 1999.

[213] Yi Ma, Harm Derksen, Wei Hong, and John Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE transactions on pattern analysis and machine intelligence*, 29(9), 2007.

[214] Paul S Bradley and Olvi L Mangasarian. K-plane clustering. *Journal of Global Optimization*, 16(1):23–32, 2000.

[215] David L. Donoho, Martin Vetterli, Ronald A. DeVore, and Ingrid Daubechies. Data compression and harmonic analysis. *IEEE Transactions on Information Theory*, 44(6):2435–2476, 1998.

[216] Ehsan Elhamifar and René Vidal. Clustering disjoint subspaces via sparse representation. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 1926–1929. IEEE, 2010.

[217] Xi Peng, Lei Zhang, and Zhang Yi. Scalable sparse subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 430–437, 2013.

[218] Shusen Wang, Xiaotong Yuan, Tiansheng Yao, Shuicheng Yan, and Jialie Shen. Efficient subspace segmentation via quadratic programming. In *AAAI*, volume 1, pages 519–524, 2011.

[219] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 663–670, 2010.

[220] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

[221] Paolo Favaro, René Vidal, and Avinash Ravichandran. A closed form solution to robust subspace estimation and clustering. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1801–1807. IEEE, 2011.

[222] Yu-Xiang Wang, Huan Xu, and Chenlei Leng. Provable subspace clustering: When lrr meets ssc. In *Advances in Neural Information Processing Systems*, pages 64–72, 2013.

[223] Guangcan Liu and Shuicheng Yan. Latent low-rank representation for subspace segmentation and feature extraction. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1615–1622. IEEE, 2011.

[224] Vishal M Patel, Hien Van Nguyen, and René Vidal. Latent space sparse subspace clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 225–232, 2013.

[225] Vishal M Patel and René Vidal. Kernel sparse subspace clustering. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 2849–2853. IEEE, 2014.

[226] Duc-Son Pham, Saha Budhaditya, Dinh Phung, and Svetha Venkatesh. Improved subspace clustering via exploitation of spatial constraints. In *Computer Vision and*

Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 550–557. IEEE, 2012.

[227] Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. Robust and efficient subspace segmentation via least squares regression. *Computer Vision–ECCV 2012*, pages 347–360, 2012.

[228] Canyi Lu, Jiashi Feng, Zhouchen Lin, and Shuicheng Yan. Correlation adaptive subspace segmentation by trace lasso. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1345–1352, 2013.

[229] Chun-Guang Li and René Vidal. Structured sparse subspace clustering: A unified optimization framework. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 277–286, 2015.

[230] Pan Ji, Mathieu Salzmann, and Hongdong Li. Efficient dense subspace clustering. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 461–468. IEEE, 2014.

[231] Jiashi Feng, Zhouchen Lin, Huan Xu, and Shuicheng Yan. Robust subspace segmentation with block-diagonal prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3818–3825, 2014.

[232] Stephen Tierney, Junbin Gao, and Yi Guo. Subspace clustering for sequential data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1019–1026, 2014.

[233] Zinan Zeng, Tsung-Han Chan, Kui Jia, and Dong Xu. Finding correspondence from multiple images via sparse and low-rank decomposition. *Computer Vision–ECCV 2012*, pages 325–339, 2012.

[234] Pan Ji, Hongdong Li, Mathieu Salzmann, and Yuchao Dai. Robust motion segmentation with unknown correspondences. In *European conference on computer vision*, pages 204–219. Springer, 2014.

[235] Baohua Li, Ying Zhang, Zhouchen Lin, and Huchuan Lu. Subspace clustering by mixture of gaussian regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2094–2102, 2015.

[236] Juan-Manuel Pérez-Rúa, Tomas Crivelli, Patrick Pérez, and Patrick Bouthemy. Discovering motion hierarchies via tree-structured coding of trajectories. In *27th British Machine Vision Conference (BMVC 2016)*, 2016.

[237] Juan-Manuel Pérez-Rúa, Tomas Crivelli, Patrick Pérez, and Patrick Bouthemy. Hierarchical motion decomposition for dynamic scene parsing. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3952–3956. IEEE, 2016.

[238] Atsuto Maki and Hiroshi Hattori. Illumination subspace for multibody motion segmentation. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2001.

[239] Hua Yang, Greg Welch, Jan-Michael Frahm, and Marc Pollefeys. 3d motion segmentation using intensity trajectory. In *Asian Conference on Computer Vision*, pages 157–168. Springer, 2009.

[240] Liangjing Ding, Adrian Barbu, and Anke Meyer-Baese. Motion segmentation by velocity clustering with estimation of subspace dimension. In *Asian Conference on Computer Vision*, pages 491–505. Springer, 2012.

[241] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. *Computer Vision–ECCV 2010*, pages 282–295, 2010.

[242] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1187–1200, 2014.

[243] Peter Ochs and Thomas Brox. Higher order motion models and spectral clustering. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 614–621. IEEE, 2012.

[244] Robert C Bolles, H Harlyn Baker, and David H Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.

[245] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[246] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.

[247] Velodyne LiDAR. Hdl-64e, 2014.

[248] Hicham Sekkati and Amar Mitiche. Concurrent 3-d motion segmentation and 3-d interpretation of temporal sequences of monocular images. *IEEE Transactions on Image Processing*, 15(3):641–653, 2006.

[249] Guofeng Zhang, Jiaya Jia, Wei Hua, and Hujun Bao. Robust bilayer segmentation and motion/depth estimation with a handheld camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):603–617, 2011.

[250] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):500–513, 2011.

[251] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.

[252] Danping Zou and Ping Tan. Coslam: Collaborative visual slam in dynamic environments. *IEEE transactions on pattern analysis and machine intelligence*, 35(2):354–366, 2013.

[253] Zhencheng Hu, Keiichi Uchimura, and Jia Wang. Moving obstacles extraction with stereo global motion model. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 79–83. IEEE, 2006.

[254] Jia Wang, Zhencheng Hu, Hanqing Lu, and Keiichi Uchimura. Motion detection in driving environment using uv-disparity. *Computer Vision–ACCV 2006*, pages 307–316, 2006.

[255] Christoph Vogel, Konrad Schindler, and Stefan Roth. Piecewise rigid scene flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1377–1384, 2013.

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

[257] Andreas Wedel, Annemarie Meißner, Clemens Rabe, Uwe Franke, and Daniel Cremers. Detection and segmentation of independently moving objects from dense scene flow. In *Energy minimization methods in computer vision and pattern recognition*, pages 14–27. Springer, 2009.

[258] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Kitti object scene flow benchmark, 2015.

[259] Tatsunori Taniai, Sudipta N Sinha, and Yoichi Sato. Fast multi-frame stereo scene flow with motion segmentation. *arXiv preprint arXiv:1707.01307*, 2017.

[260] Deyvid Kochanov, Aljosa Osep, Jörg Stückler, and Bastian Leibe. Scene flow propagation for semantic mapping and object discovery in dynamic street scenes. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2016.

[261] Jens Klappstein, Tobi Vaudrey, Clemens Rabe, Andreas Wedel, and Reinhard Klette. Moving object segmentation using optical flow and depth information. *Advances in Image and Video Technology*, pages 611–623, 2009.

[262] Simon J Julier and Jeffrey K Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Int. symp. aerospace/defense sensing, simul. and controls*, volume 3, pages 182–193. Orlando, FL, 1997.

[263] Bastian Leibe, Nico Cornelis, Kurt Cornelis, and Luc Van Gool. Dynamic 3d scene analysis from a moving vehicle. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[264] Kimiya Aoki and Hiroyasu Koshimizu. Detection of 3d-flow by characteristic of convex-concave and color. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 75–78. IEEE, 2006.

[265] Jörg Stückler and Sven Behnke. Efficient dense rigid-body motion segmentation and estimation in rgb-d video. *International Journal of Computer Vision*, 113(3):233–245, 2015.

[266] Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d flow: Dense 3-d motion estimation using color and depth. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2276–2282. IEEE, 2013.

[267] S. Perera and N. Barnes. A simple and practical solution to the rigid body motion segmentation problem using a rgb-d camera. In *2011 International Conference on Digital Image Computing: Techniques and Applications*, pages 494–500, Dec 2011.

[268] Samunda Perera and Nick Barnes. *Maximal Cliques Based Rigid Body Motion Segmentation with a RGB-D Camera*, pages 120–133. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[269] Jeremie Papon, Tomas Kulvicius, Eren Erdal Aksoy, and Florentin Wörgötter. Point cloud video object segmentation using a persistent supervoxel world-model. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3712–3718. IEEE, 2013.

[270] Seongyong Koo, Dongheui Lee, and Dong-Soo Kwon. Incremental object learning and robust tracking of multiple objects from rgb-d point set data. *Journal of Visual Communication and Image Representation*, 25(1):108–121, 2014.

[271] Yerry Sofer, Tal Hassner, and Andrei Sharf. Interactive learning for point-cloud motion segmentation. In *Computer Graphics Forum*, volume 32, pages 51–60. Wiley Online Library, 2013.

[272] Jaime S Cardoso, Jorge CS Cardoso, and Luis Corte-Real. Object-based spatial segmentation of video guided by depth and motion information. In *Motion and Video Computing, IEEE Workshop on*, pages 1–7. IEEE, 2007.

[273] Liang Wang, Chenxi Zhang, Ruigang Yang, and Cha Zhang. Tofcut: Towards robust real-time foreground extraction using a time-of-flight camera. In *Proc. of 3DPVT*, pages 1–8, 2010.

[274] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[275] Samunda Perera, Nick Barnes, Xuming He, Shahram Izadi, Pushmeet Kohli, and Ben Glocker. Motion segmentation of truncated signed distance function based volumetric surfaces. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 1046–1053. IEEE, 2015.

[276] Chieh-Chih Wang and Chuck Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 2918–2924. IEEE, 2002.

[277] Chieh-Chih Wang, Charles Thorpe, and Sebastian Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 842–849. IEEE, 2003.

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

[279] 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.

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

[281] Asma Azim and Olivier Aycard. Detection, classification and tracking of moving objects in a 3d environment. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 802–807. IEEE, 2012.

[282] François Pomerleau, Philipp Krüsi, Francis Colas, Paul Furgale, and Roland Siegwart. Long-term 3d map maintenance in dynamic environments. In *IEEE International Conference on Robotics and Automation*, pages 3712–3719. IEEE, 2014.

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

[284] Rareş Ambruş, Nils Bore, John Folkesson, and Patric Jensfelt. Meta-rooms: Building and maintaining long term spatial models in a dynamic world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1854–1861. IEEE, 2014.

[285] Ayush Dewan, Tim Caselitz, Gian Diego Tipaldi, and Wolfram Burgard. Motion-based detection and tracking in 3d lidar scans. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4508–4513. IEEE, 2016.

[286] David C Lay. Linear algebra and its applications, 2016.

[287] Guanghui Wang and QM Jonathan Wu. *Guide to three dimensional structure and motion factorization*. Springer, 2011.

[288] John Oliensis and Richard Hartley. Iterative extensions of the sturm/triggs algorithm: Convergence and nonconvergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2217–2233, 2007.

[289] Dijun Luo, Feiping Nie, Chris Ding, and Heng Huang. Multi-subspace representation and discovery. *Machine Learning and Knowledge Discovery in Databases*, pages 405–420, 2011.

[290] Daniel A Spielman. Spectral graph theory and its applications. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 29–38. IEEE, 2007.

[291] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[292] Peter J Huber. Robust statistics. In *International Encyclopedia of Statistical Science*, pages 1248–1251. Springer, 2011.

[293] Zhengyou Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and vision Computing*, 15(1):59–76, 1997.

[294] René Vidal, Yi Ma, and S Shankar Sastry. *Generalized Principal Component Analysis*. Springer, 2016.

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

[296] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. John Hopkins University Press, 2012.

[297] Richard G Baraniuk. Compressive sensing [lecture notes]. *IEEE signal processing magazine*, 24(4):118–121, 2007.

[298] David L Donoho and Jared Tanner. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Proceedings of the National Academy of Sciences of the United States of America*, 102(27):9446–9451, 2005.

[299] David L Donoho. For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. *Communications on pure and applied mathematics*, 59(6):797–829, 2006.

[300] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *ICDT*, volume 1, pages 420–434. Springer, 2001.

[301] Javier Civera, Oscar G Grasa, Andrew J Davison, and JMM Montiel. 1-point ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, 27(5):609–631, 2010.

[302] Georg Klein and David Murray. Improving the agility of keyframe-based slam. *Computer Vision–ECCV 2008*, pages 802–815, 2008.

[303] Wolfram Burgard, Cyrill Stachniss, and Dirk Hähnel. Mobile robot map learning from range data in dynamic environments. *Autonomous Navigation in Dynamic Environments*, pages 3–28, 2007.

[304] Florent Lafarge and Clément Mallet. Creating large-scale city models from 3d-point clouds: a robust approach with hybrid representation. *International journal of computer vision*, 99(1):69–85, 2012.

[305] Andreas Geiger, Martin Lauer, Christian Wojek, Christoph Stiller, and Raquel Urtasun. 3d traffic scene understanding from movable platforms. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):1012–1025, 2014.

[306] 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.

[307] Richard H. Bartels and George W Stewart. Solution of the matrix equation ax+ xb= c [f4]. *Communications of the ACM*, 15(9):820–826, 1972.

[308] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[309] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1385–1392, 2013.

[310] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Forward-backward error: Automatic detection of tracking failures. In *Pattern recognition (ICPR), 2010 20th international conference on*, pages 2756–2759. IEEE, 2010.

[311] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Deepmatching: Hierarchical deformable dense matching. *International Journal of Computer Vision*, 120(3):300–323, 2016.

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

[313] 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.

[314] 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.

[315] 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.

[316] Danda Pani Paudel, Cédric Demonceaux, Adlane Habed, Pascal Vasseur, and In So Kweon. 2d-3d camera fusion for visual odometry in outdoor environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 157–162. IEEE, 2014.

[317] 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.

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

[319] Edmond Boyer and Marie-Odile Berger. 3d surface reconstruction using occluding contours. *International Journal of Computer Vision*, 22(3):219–233, 1997.

[320] Evren Imre and Marie-Odile Berger. A 3-component inverse depth parameterization for particle filter slam. In *DAGM-Symposium*, pages 1–10. Springer, 2009.

[321] Evren Imre, M-O Berger, and Nicolas Noury. Improved inverse-depth parameterization for monocular simultaneous localization and mapping. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 381–386. IEEE, 2009.

[322] 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.

[323] Yohann Salaün, Renaud Marlet, and Pascal Monasse. Multiscale line segment detector for robust and accurate sfm. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 2000–2005. IEEE, 2016.

[324] Yohann Salaun, Renaud Marlet, and Pascal Monasse. Robust sfm with little image overlap. In *IEEE International Conference on 3D Vision (3DV)*. Qingdao, China, Oct. 2017.

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

[326] Shin Miyake, Yuichiro Toda, Naoyuki Kubota, Naoyuki Takesue, and Kazuyoshi Wada. Intensity histogram based segmentation of 3d point cloud using growing neural gas. In *International Conference on Intelligent Robotics and Applications*, pages 335–345. Springer, 2016.

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

[328] M Lindstrom and J-O Eklundh. Detecting and tracking moving objects from a mobile platform using a laser range scanner. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 3, pages 1364–1369. IEEE, 2001.

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

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

[331] Andrew W. Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21(13):1145–1153, 2003.

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

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

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

[335] 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.

[336] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

[337] Ji Zhang and Sanjiv Singh. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2174–2181. IEEE, 2015.

[338] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[339] Josiah Willard Gibbs. *Elements of vector analysis: arranged for the use of students in physics*. Tuttle, Morehouse & Taylor, 1884.

[340] Arthur Cayley. About the algebraic structure of the orthogonal group and the other classical groups in a field of characteristic zero or a prime characteristic. *Reine Angewandte Mathematik*, 32:1846, 1846.

[341] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[342] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.

[343] Noel Cressie and Douglas M Hawkins. Robust estimation of the variogram: I. *Journal of the International Association for Mathematical Geology*, 12(2):115–125, 1980.

[344] Kai Wang, Guillaume Lavoué, Florence Denis, and Atilla Baskurt. A comprehensive survey on three-dimensional mesh watermarking. *IEEE Transactions on Multimedia*, 10(8):1513–1527, 2008.

[345] Kai Wang, Guillaume Lavoué, Florence Denis, Atilla Baskurt, and Xiyan He. A benchmark for 3d mesh watermarking. In *Shape Modeling International Conference (SMI), 2010*, pages 231–235. IEEE, 2010.

[346] Jinjiang Guo, Vincent Vidal, Irene Cheng, Anup Basu, Atilla Baskurt, and Guillaume Lavoue. Subjective and objective visual quality assessment of textured 3d meshes. *ACM Transactions on Applied Perception (TAP)*, 14(2):11, 2016.

[347] Peter Lancaster and Kes Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of computation*, 37(155):141–158, 1981.

[348] Marco Callieri, Paolo Cignoni, Massimiliano Corsini, and Roberto Scopigno. Masked photo blending: Mapping dense photographic data set on high-resolution sampled 3d models. *Computers & Graphics*, 32(4):464–473, 2008.

[349] Massimiliano Corsini, Paolo Cignoni, and Roberto Scopigno. Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):914–924, 2012.

[350] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4):349–359, 1999.

[351] Simon Boyé, Gael Guennebaud, and Christophe Schlick. Least squares subdivision surfaces. In *Computer Graphics Forum*, volume 29, pages 2021–2028. Wiley Online Library, 2010.

[352] Pan Ji, Ian Reid, Ravi Garg, Hongdong Li, and Mathieu Salzmann. Low-rank kernel subspace clustering. *arXiv preprint arXiv:1707.04974*, 2017.

List of Figures

List of Tables

List of Definitions

Document réalisé avec LaTeX et :
le style LaTeX pour Thèse de Doctorat créé par S. Galland — http://www.multiagent.fr/ThesisStyle
la collection de paquets `tex-upmethodology` — http://www.arakhne.org/tex-upmethodology/

Abstract:

This thesis studies the problem of dynamic scene 3D reconstruction and understanding using a calibrated 2D-3D camera setup mounted on a mobile platform via the analysis of objects' motions. For static scenes, the sought 3D map reconstruction can be obtained by registering the point cloud sequence. However, with dynamic scenes, we require a prior step of moving object elimination, which yields to the motion detection and segmentation problems. We provide solutions for the two practical scenarios, namely the known and unknown camera motion cases, respectively. When camera motion is unknown, our 3D-SSC and 3D-SMR algorithms segment the moving objects by analysing their 3D feature trajectories. In contrast, by compensating the known camera motion, our 3D Flow Field Analysis algorithm inspects the spatio-temporal property of the object's motion. By removing the dynamic objects, we attain the high quality 3D background and multi-body reconstruction by using our DW-ICP point cloud registration algorithm. In the context of scene understanding, semantic object information is learned from images and transferred to the reconstructed static map via our 2D-to-3D label transfer scheme. All the proposed algorithms have been quantitatively and qualitatively evaluated and validated by using extensive experiments of real outdoor scenes.

Keywords: Moving Object Detection, Motion Segmentation, 3D Map Reconstruction, Dynamic Scene Analysis

SPIM

■ École doctorale SPIM - Université de Bourgogne/UFR ST BP 47870 F - 21078 Dijon cedex
■ tél. +33 (0)3 80 39 59 10 ■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr

UNIVERSITÉ DE BOURGOGNE