Multi-frame Motion Segmentation for Dynamic Scene Modelling

Jun Zhang and Viorela Ila
Australian Centre for Robotic Vision
Australian National University
{jun.zhang2,viorela.ila}@anu.edu.au

Abstract

Motion segmentation is a fundamental problem in dynamic scene understanding. Although it has been a long-term studied topic, motion segmentation is still not successfully applied in real-life challenging scenarios, such as autonomous or assistant driving. Based on such consideration, this paper proposes a robust solution to motion segmentation for urban driving scenes, by addressing two main challenges which are not fully explored in previously proposed algorithms: (1) detect, segment and track motions simultaneously without prior knowledge of feature trajectories or number of motions but with the help of semantic scene labeling (2) provide a robust solutions in strong perspective scenes with occlusion, by formulating the problem using a novel motion model of dynamic points used in the re-projection error minimization within a stereo/RGB-D setup. The proposed approach is carefully evaluated on the virtual KITTI dataset with ground truth, and tested on the real KITTI dataset.

1 Introduction

Motion segmentation remains as one of the fundamental challenges in computer vision and robotics. It is a key component for many robotic tasks such as dynamic scene reconstruction, navigation and path planning in dynamic environments and simultaneous localization and tracking of multiple objects. Applications such as autonomous vehicles, augmented reality and assistive driving can benefit from accurate and robust motion segmentation. In particular, motion segmentation in autonomous driving must be robust, accurate and work properly in a wide range from close to far.

Significant research effort has been made in motion segmentation in past two decades. Earlier proposed methods in the literature can be divided into two categories, methods based on affine assumption and those based on Epipolar geometry. Affine methods [1–3] assume that each individual rigid motion across multiple frames lies in an affine or linear subspace. This type of problem can be effectively solved by factorization or subspace clustering frameworks. Nevertheless, affine-based algorithms have several issues, among which is the inability to deal with perspective effects. This drawback heavily restricts their performance in motion segmentation in outdoor scenarios, in particular in autonomous or assistant driving applications. Another drawback is their requirement of full-length feature trajectories, which impedes them to handle real scenarios with objects entering into or leaving the field of view, and temporary occlusion. Finally, in this methods, the number of motions is normally assumed to be known as prior. A fixed number of motions is an incorrect assumption in real scenarios where the number of motions is changing over time. Epipolar geometry-based methods [4, 5] are generally designed for two-frame segmentation, as they are based on Epipolar constraints of two-perspective-view to model different motions, and are thereby able to deal with perspective effects. However, they become invalid when the object motion is degenerate, such as a moving planar object or object that performs a pure rotation relative to the camera centre.

In this paper, we propose a novel motion segmentation framework that is based on a visual odometry (VO) system able to identify moving objects in real world scenarios. Our framework is built upon Epipolar geometry-based methods, but formulated in a 3D to 2D projective geometry. For that, we propose a novel motion model of dynamic points in the scene that can be used in the projection of moving points in subsequent frames. The motion of each object is then identified through a combination of the structure flow and instance level semantic segmentation based on Mask-RCNN [6]. The proposed method not only can overcome the problem of degenerate motions (i.e., translation along the camera view), but also effectively handle the partial occlusions present in real-life scenarios. The proposed solution ensures a continuous motion tracking and segmentation across multiple frames.

2 Related Works

2.1 Motion Segmentation

Considering the pros and cons of both affine-based and Epipolar geometry-based methods, current state-of-the-art al-

gorithms all seek to combine them into a single framework that leverages their advantages and avoid their weaknesses. Li *et al.* [7] propose a multi-frame motion segmentation framework, which combines sparse subspace clustering (SSC) [8] approach with Epipolar constraints to formulates the segmentation problem as a graph partitioning problem based on an affinity matrix. To make it applicable in multi-frame case, an aggregated affinity matrix from multiple frames is derived to find a joint sparse coefficient recovery across the frames. Besides that, a robust model selection with outlier rejection is introduced, by first over-segmenting data into groups, then merging with loose grouping. More recently, a multi-frame spectral clustering [9] framework is introduced in [10] with jointly integration of affine model, homography model and fundamental matrix. Specifically, affinity matrix is first constructed for different models. The affinities between feature points are then encapsulated in the ORK kernel [11] to handle problems of inlier determination and sampling imbalance, which are finally fused as subset constraint and integrated into the spectral clustering problem that can be solved in an alternate optimization pattern. Similar to [7], [12] also construct the formulation under Epipolar constraint and solve it using SSC algorithm. The difference is that the latter introduces semantic affinity matrix and combine it with geometric affinity matrix in the formulation. Making use of semantic information helps to deal with issues of indistinguishable motions and partial occlusion, and as a result increases the segmentation accuracy. Therefore, the proposed framework leverages object semantic information to increase the robustness of the results.

2.2 Scene Flow

Scene flow estimation has been an active research topics in recent years. Attractive results have been delivered in state-of-the-art works. Early works, such as [13, 14], denote scene flow as a vector combining optical flow and disparity. Wedel et al. [13] proposed to decouple the 3D position and scene flow estimation separately and estimate dense 3D flow using a variational approach. In [14], a seed growing algorithm is presented to jointly and efficiently estimate semi-dense disparity and optical flow. The basic idea of it is that correspondences can be found in a small neighboring region around an initial set of seed correspondences. In stead of the above denotation, [15] introduce a novel representation of the dynamic 3D scene by a collection of piecewise planar, rigidly moving regions. In this case, the estimation of scene flow includes 3D position, normal vector and rigid motion of a plane for each segment, which is formulated as a discrete, non-submodular energy function and optimized in an alternative way. To improve the efficiency, [16] assumes a fine superpixel segmentation as fixed prior, and proposes a purely continuous factor graph formulation. This decomposes the problem into photometric, geometric and smoothing constraints and solves independently. In the end, a global non-linear refinement is performed to get an optimal result. Inspired by [15], we try to group features belonging to different rigid motion models in dynamic scene, and describe moving objects with corresponding models. By doing this, we not just achieve the goal of motion segmentation, but also modelling the dynamic scene in a continual and effective fashion.

3 Background Methodology

This section introduces the notation and geometrically formulates the problem of motion segmentation from stereo/ RGB-D images.

Given a stereo setup, we assume a set of 3D map points represented in the world coordinate frame $\{o\}$; ${}^{o}\boldsymbol{m} = \{{}^{o}\boldsymbol{m}^{i} \in \mathbb{R}^{4} | \, , i = 1, \ldots, n\}$, where n is the total number of observed 3D points and ${}^{o}\boldsymbol{m}^{i} = {}^{o}[m_x^i,\ m_y^i,\ m_z^i, 1]^{\top}$ is a 3D point in homogeneous coordinates. Let ${}^{k-1}\boldsymbol{p}^{i} \in \mathbb{R}^{4}$ be the stereo feature associated with the i^{th} point observed in the image frame $k-1$, where ${}^{k-1}\boldsymbol{p}^{i} = {}^{k-1}[u_l^i,\ u_r^i,\ v^i\ , 1]^{\top}$ are the rectified stereo image coordinates of the feature. For each stereo feature in frame $k-1$ we can find its correspondence in the subsequent frame k: ${}^{k}\boldsymbol{p}^{i} \in \mathbb{R}^{4}$. Note that for the RGB-D setup, we can use the same formulation as stereo but assuming a preset virtual baseline.

3.1 Camera Pose Estimation

To estimate the camera pose at time k in the world frame $\{o\}$, ${}^{o}\boldsymbol{T}_k$, we construct an optimization problem. Specifically, each static landmark ${}^{o}\boldsymbol{m}^{i}$ observed at time $k-1$ is projected onto the image plane k as shown in Figure 1a using the projection function:

$$ {}^{k}\hat{\boldsymbol{p}}^{i} = \pi({}^{k}\boldsymbol{m}^{i}) = \pi({}^{o}\boldsymbol{T}_k^{-1}\ {}^{o}\boldsymbol{m}^{i}) \tag{1} $$

where $\pi(\cdot)$ is defined as follows:

$$ \pi(\begin{bmatrix} m_x \\ m_y \\ m_z \\ 1 \end{bmatrix}) = \begin{bmatrix} u_l \\ u_r \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f\frac{m_x}{m_z} + c_u \\ f\frac{m_y - b}{m_z} + c_u \\ f\frac{m_y}{m_z} + c_v \\ 1 \end{bmatrix} \tag{2} $$

with f the focal length of the cameras , (c_u, c_v) the principal point (assuming both cameras have the same focal length and principal point) and b is the baseline of the stereo system.

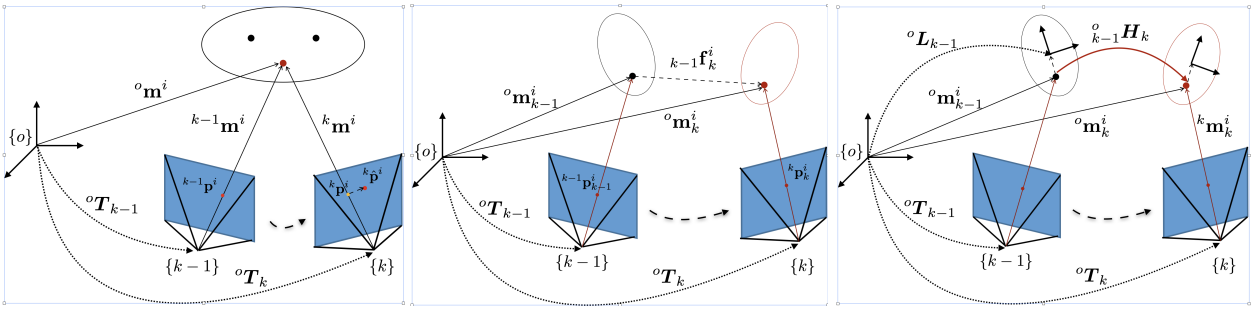
Finally, ${}^{o}\boldsymbol{T}_k$ is found by minimizing the following reprojection error for all the visible static points $\{i \in 1 \ldots n_s\}$ in frame k:

$$ {}^{o}\boldsymbol{T}_k^{*} = \underset{{}^{o}\boldsymbol{T}_k}{argmin} \sum_{i=1}^{n_s} ||{}^{k}\boldsymbol{p}^{i} - {}^{k}\hat{\boldsymbol{p}}^{i}||^2 \tag{3} $$

The optimal result is obtained using Levenberg-Marquardt optimization. The ego-motion is obtained from:

$$ {}^{k-1}_{k-1}\boldsymbol{T}_k = {}^{o}\boldsymbol{T}_{k-1}^{-1}\ {}^{o}\boldsymbol{T}_k \tag{4} $$

To be robust against outliers, we wrap the above estimation into a RANSAC model fitting framework.



(a) Ego motion. (b) Structure flow. (c) Motion of dynamic point.

Figure 1: Coordinate frames, points and motion.

3.2 Sparse Structure Flow

The location of a static 3D point in the scene should always remain the same in the reference frame $\{o\}$. For points on moving objects, their absolute location changes with time. Figure 1b shows an example of a 3D point on a moving object at two time steps: $\boldsymbol{m}^i_{k-1}$ and $\boldsymbol{m}^i_k$. In order to describe the motion, in the following derivations we introduce the time step in our notation as a right subindex.

In a stereo set-up, the scene flow is defined as a vector combining optical flow and disparity, and has been used in the literature in the context of motion segmentation. In order to overcome the problem of degenerate motions, in this paper we define the *structure flow* as the 3D motion field of a scene, seen as a displacement vector of each surface point. The structure flow vector associated to the i^{th} 3D point is computed as a difference of the two vectors:

$$ {}_{k-1}\boldsymbol{f}^i_k = {}^o\boldsymbol{m}^i_k - {}^o\boldsymbol{m}^i_{k-1} \tag{5} $$

For static 3D points the structure flow vector should be close to zero.

3.3 Motion Model of Points on a Rigid Body

In this section we derive the motion model of a point on a rigid body. We assume that the rigid body transformation of an object in motion from time $k-1$ to time k is given by ${}^{k-1}_{k-1}\boldsymbol{H}_k \in SE(3)$ and it is represented in the coordinate frame of the object at time $k-1$. If the pose of the object at time $k-1$ is given by ${}^o\boldsymbol{L}_{k-1} \in SE(3)$ in the reference frame o, according to Chirikjian et al. [17], the frame change of the object pose transformation is given by:

$$ {}^o_{k-1}\boldsymbol{H}_k = {}^o\boldsymbol{L}_{k-1}\ {}^{k-1}_{k-1}\boldsymbol{H}_k\ {}^o\boldsymbol{L}^{-1}_{k-1} \in SE(3) \tag{6} $$

According to [18], the motion of a point on a rigid body can be expressed using the *rigid body pose transformation in inertial frame* given by ${}^o_{k-1}\boldsymbol{H}_k$, with the following relation:

$$ {}^o\boldsymbol{m}^i_k = {}^o_{k-1}\boldsymbol{H}_k\ {}^o\boldsymbol{m}^i_{k-1} \tag{7} $$

If a point at time k in the camera coordinates is given by ${}^k\boldsymbol{m}^i_k = {}^o\boldsymbol{T}^{-1}_k\ {}^o\boldsymbol{m}^i_k$ and using (7), we obtain:

$$ {}^k\boldsymbol{m}^i_k = {}^o\boldsymbol{T}^{-1}_k\ {}^o_{k-1}\boldsymbol{H}_k\ {}^o\boldsymbol{m}^i_{k-1} \tag{8} $$

Given a set of image features corresponding to moving points on a rigid body $\{i \in 1 \dots n_d\}$, the motion of the points on the rigid body ${}^o_{k-1}\boldsymbol{H}_k$ can be estimated by minimizing the following re-projection error:

$$ {}^o_{k-1}\boldsymbol{H}^*_k = \underset{{}^o_{k-1}\boldsymbol{H}_k}{argmin} \sum_{i=1}^{n_d} ||{}^k\boldsymbol{p}^i_k - \pi({}^o\boldsymbol{T}^{-1}_k\ {}^o_{k-1}\boldsymbol{H}_k\ {}^o\boldsymbol{m}^i_{k-1})||^2 \tag{9} $$

where ${}^o\boldsymbol{T}_k$ is known and obtained using (3). In practice, it is convenient to estimate for the transformation ${}^k\boldsymbol{R}_k = {}^o\boldsymbol{T}^{-1}_k\ {}^o_{k-1}\boldsymbol{H}_k \in SE(3)$ and recover the motion afterwards.

This novel formulation allows modelling the motion of all the points pertaining to the same object by an $SE(3)$ transformation that can be estimated in the same way as ego-motion.

4 Implementation

Our proposed method of motion segmentation is embedded in a Visual Odometry framework, in which the ego-motion of camera and object motions estimation can benefit from a good feature points tracking and at the same time, the tracking can benefit from better estimation of those motions. The proposed pipeline is summarized in Figure 2. In the pipeline illustration, $\boldsymbol{S}^m$ refers to semantic mask of the image and $\boldsymbol{L}$ means labels to the features obtained after the classification. Note that $\boldsymbol{L}$ contains 3 types of labels: dynamic objects $(1, 2, ...)$, static (0) and outliers (-1).

4.1 Ego-motion Estimation

The goal of visual odometry is to incrementally estimate the motion of a camera/robot (ego-motion) from visual information of the environment. To achieve this, salient features on each new image frame are first detected, and discriminative descriptors are matchesextracted. SIFT [19] technique is chosen in our implementation, due to its powerful discrimination and strong invariance to scale, rotation, illumination change and noise.

After the feature extraction is done, correspondences with the previous frame are found. In this paper, an effective matching approach based on motion prior is intro-

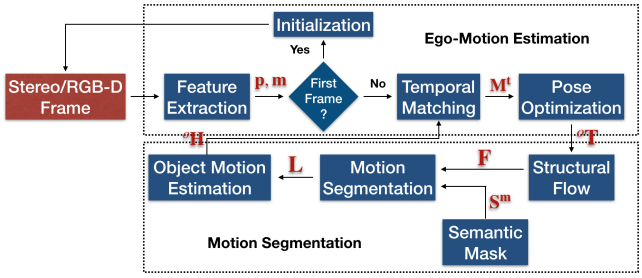


Figure 2: The proposed system composes by two main blocks: the ego-motion estimation and the motion segmentation. Letters in red color refer to output for each small blocks.

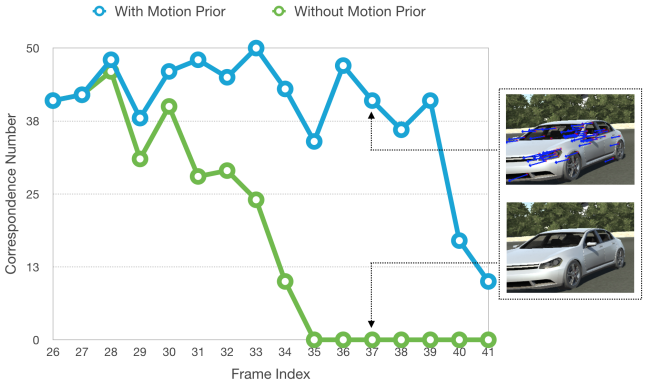


Figure 3: Comparison of matching result with and without motion priors in sequence 0001 of virtual KITTI dataset. The blue arrows with red points on the sedan are projection of the structure flow vectors.

duced to tackle this problem. As illustrated in Algorithm 1, assuming we have an initial estimate of the ego-motion ${}^{k-1}_{k-1}\tilde{\boldsymbol{T}}_k = {}^{k-2}_{k-2}\boldsymbol{T}_{k-1}$ and object motions ${}^{o}_{k-1}\tilde{\boldsymbol{H}}^j_k = {}^{o}_{k-2}\boldsymbol{H}^j_{k-1}$ for each individual j^{th} object, the features can be categorised in three groups: static, dynamic and unknown. Note that the initial estimates can be obtained by assuming constant motion models of the camera and objects in the scene. For the static or dynamic features, their correspondences in the current frame are searched by first projecting their 3D map point into the current frame, using camera motion model (static) or object motion model (dynamic) together with intrinsic parameters, and then performing a local search ($\boldsymbol{S}_{lo}$) by comparing descriptor distance ($\boldsymbol{D}_{dist}$).

The static/dynamic labels (motion models) are propagated through feature matching process. In order to keep a high number of features to track, we also consider the features that have not yet been assigned a motion model. For those, we perform a neighbourhood search in order to obtain the adequate motion model to be used in the matching process. For each candidate feature, its top K nearest neighbours sorted by Euclidean distance are obtained. The distance matrix is precomputed using FLANN [20]. The K nearest neighbours are

Algorithm 1 Feature Matching via Multiple Motion Models

Require:
${}^{o}\boldsymbol{m}_{k-1} = \{{}^{o}\boldsymbol{m}^i_{k-1}, i = 1, ..., n_{k-1}\}$ and $\boldsymbol{p}_k = \{\boldsymbol{p}^j_k, j = 1, ..., n_k\}$: map points in frame $k-1$ and features in frame k;
${}^{o}\tilde{\boldsymbol{T}}_k$ and ${}^{o}_{k-1}\tilde{\boldsymbol{H}}_k = \{{}^{o}_{k-1}\tilde{\boldsymbol{H}}^i_k, i = 1, ..., m\}$: ego-motion and dynamic points motion of m objects in frame k;
Ensure: $\boldsymbol{M}^t_{k-1,k} = \{\{\boldsymbol{m}^i_{k-1}, \boldsymbol{p}^i_k\}, i = 1, ..., n_t\}$: Temporal matches between $k-1$ and k frames;
for each ${}^{o}\boldsymbol{m}^i_{k-1} \in {}^{o}\boldsymbol{m}_{k-1}$ **do**
  if ${}^{o}\boldsymbol{m}^i_{k-1}$ is static, **then**
    $\hat{\boldsymbol{p}}^i_k = \pi({}^{o}\tilde{\boldsymbol{T}}^{-1}_k \, {}^{o}\boldsymbol{m}^i_{k-1})$;
  else if ${}^{o}\boldsymbol{m}^i_{k-1}$ is dynamic, **then**
    $\hat{\boldsymbol{p}}^i_k = \pi({}^{o}\tilde{\boldsymbol{T}}^{-1}_k \, {}^{o}_{k-1}\tilde{\boldsymbol{H}}^i_k \, {}^{o}\boldsymbol{m}^i_{k-1})$;
  end if
  $\boldsymbol{p}_{lo} = \mathcal{S}_{lo}\{\boldsymbol{p}_k\}$;
  ${}^{*}\boldsymbol{p}_k = \{\boldsymbol{p}^j_k \in \boldsymbol{p}_{lo} \mid \boldsymbol{D}_{dist}(\hat{\boldsymbol{p}}^i_k, \boldsymbol{p}^j_k)$ is min$\}$;
  $\boldsymbol{M}^t_{k-1,k} \leftarrow \{{}^{o}\boldsymbol{m}^i_{k-1}, {}^{*}\boldsymbol{p}_k\}$;
end for
return $\boldsymbol{M}^t_{k-1,k}$ **;**

thresholded within a reasonable distance. Furthermore, the algorithm checks the ratio between the first and the second neighbour in the list. If this ratio is larger than a threshold, the motion of the first candidate is assigned to the feature. Otherwise, the algorithm assign the motion model with the highest occurrence within the thresholded K nearest neighbours. For the initialisation of the first two frames, we simply performe a wider but fixed radius search to find the correspondences.

Figure 3 illustrates a comparison of matching result with and without motion priors in a short clip, where the silver sedan is passing by the agent car. As the sedan is coming closer, the matching without motion priors fails to find any correspondence, while the other still keeps track on the sedan.

The 3D static landmarks in the previous frame and their associated 2D features in the current frame are used to calculate the ego-motion as described in section 3.1.

4.2 Motion Segmentation

In many applications ego-motion is used to separate foreground features from the background using a RANSAC approach. But ego-motion can be very imprecise and features on moving objects have high possibility to be selected as background points. Therefore, in our pipeline, the magnitudes of the structure flow vectors are used as a motion cue to decide whether a certain feature is static or dynamic. For all the matched pairs between previous and current frame, their 3D structure flow vectors are computed as described in section 3.2. This strategy avoids relying only on the ego-motion to separate the points on dynamic objects from the background points. Figure 4 shows how the result is greatly improved, the top figure shows that all the features on the

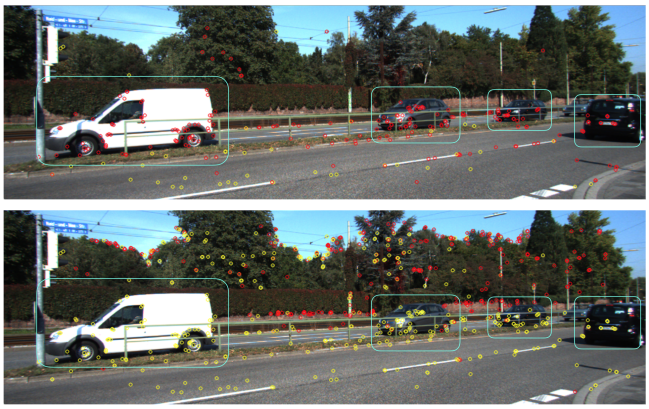


Figure 4: Sample frame from sequence 0004 in KITTI Dataset, showing foreground extraction using structure flow (upper) or inliers/outliers threshold after Ego-motion estimation (bottom). Yellow circles refer to background and red foreground.

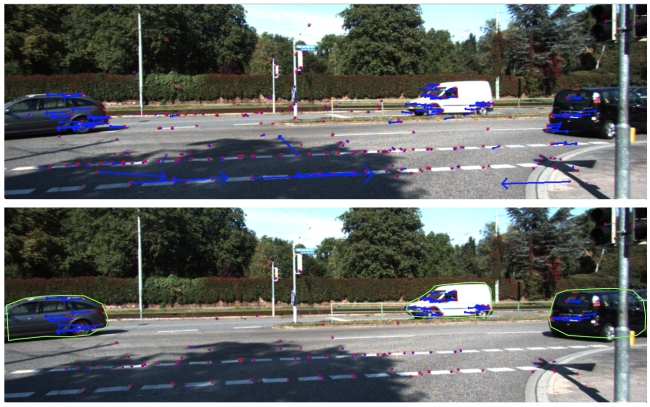


Figure 5: Sample frame from sequence 0004 in KITTI Dataset, demonstrating projection (blue arrows) of structural flow vector distribution before (upper) and after (bottom) combining semantic labeling.

moving objects are labeled as dynamic (red color) whereas the bottom figure shows features on the moving vehicles are mostly recognised as static (yellow colour) when only using ego-motion.

Moreover, depending only on the structure flow to identify dynamic features is not reliable enough, outliers can also have large flow values. As is indicated in Figure 5 (top), outliers are commonly distributed on the background scene. Therefore we propose the use of semantic labelling produced by MASK R-CNN [6] to help accomplish motion segmentation. Concretely, the model of MASK R-CNN is pre-trained on the COCO dataset [21], and it is directly applied to segment objects in our pipeline without fine-tuning. We select a set of object classes that can possibly be in motion (e.g. cars, trucks, bicycles, pedestrians, etc.), and only threshold the flow vector of points pertaining to objects in those classes. In this case, dynamic features are clustered based on different objects they belong to. This combination is straightforward but very effective in obtaining good segmentation results. Figure 5 (bottom) shows the satisfactory result after combination.

The last step of the pipeline estimate the motions of the points on the moving objects using the re-projection error described in section 3.3. A RANSAC approach is used to eliminate the outliers of each motion model present in the scene. In order to insure a large number of features tracked on moving objects, the technique make use of the instance level segmentation to propagate the motion models (labels) to all the features detected on the segmented objects. This approach is based on the fact that instance level segmentation has high accuracy, and imperfections of this will not drastically affect the real applications. Note that the motion models are calculated only using valid points. The resulting motions are then used to predict the correspondences in the next frame and insure a robust tracking of the points.

5 Experimental Results

To show the effectiveness of our proposed algorithm, we first quantitatively evaluate it on the virtual KITTI dataset [22](vKITTI, RGB-D), which provides the ground truth labeling of moving objects. Then we demonstrate qualitative results by testing the pipeline on the KITTI dataset [23](KITTI, stereo) using MASK R-CNN segmentation. For both datasets, 3000 SIFT features are detected per frame (each image in stereo data). As ground truth depth is provided in vKITTI, we add Gaussian noise with variance of 0.04 meters, which is a normal measuring error in most commercial depth sensors. The processing time of our proposed system is around 0.24 second per frame on average when run on an i7 quad-core 2.5Ghz laptop. Note that the semantic segmentation part is performed off-line using MASK R-CNN.

5.1 Quantitative Study

Virtual KITTI dataset includes 5 image sequences. From each of the sequences, we only test a selected short clip, which contains multiple moving objects. The motion segmentation results are evaluated using a Classification Error (c_e) for each frame, which is defined as the proportion of misclassified features (f_m) among the total number of features (f_t):

$$c_e = 100\% \times \frac{f_m}{f_t} \tag{10}$$

The total classification error is in fact a combination of three errors: false positive error (F/P), false negative error (F/N) and not-detected error (N/D). False positive error refers to classifying features as foreground but they are actually on the background or outliers, while false negative error refers to classifying features as background or wrong object when they are actually a different moving object. Not-detected error means the error of certain moving objects not being detected in the scene. In this case, all the detected features on

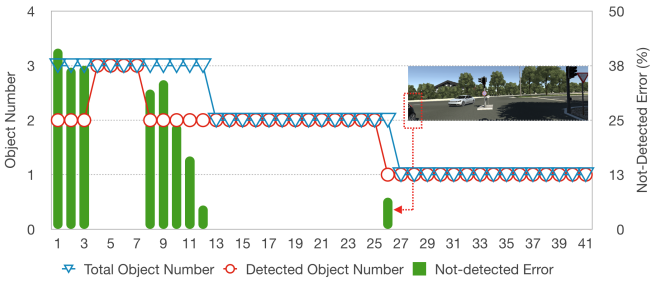


Figure 6: Distribution of object detection number and N/D error across frames of sequence 0001 in virtual KITTI Dataset.

the corresponding objects are counted into error.

Table. 1 shows the average value of the above four error metrics, as well as re-projection error (R/P), as obtained using the computed motion models of each moving object. The number of motions means the total number of moving objects that appear in the specific number of frames of the corresponding sequence. We can see that, the total error is mainly contributed by the not-detected error. This is reasonable, as it is difficult to keep continuous feature tracks on the moving objects due to distance, view or light condition change, occlusion, etc. For instance, Figure 6 demonstrates object detection number against total number and N/D error distribution across frames. It can be seen that one vehicle appears in the scene for 12 frames, but it is only been detected between frame $4 - 7$ due to the cover of tree shadow (light change). In frame 26, another vehicle is about disappearing and only a small part of it is shown in the scene, therefore the system fails to detect it. When the moving vehicles are distant from the agent vehicle motion is hard to be detected. For that reason, sequence 0200 performs poorly. As shown in Figure 7, most of the moving cars are further in the scene. The proposed system is able to accurately handle situations where the moving objects are relatively close to the moving camera, those situations being of high interest in real applications. Handling distant objects can be achieved by integrating the result of the proposed pipeline into a multi-body SLAM algorithm, and this will constitute our future work. Note that the R/P error in 0200 is zero, because R/P is only computed on the moving features, and when an object is detected, only a few features (say 4) are detected on certain object. In this case all these features have high probability of being recognized as inliers when estimating the motion model. Similarly, the F/P and F/N errors are both zeros, as only one object is detected in this sequence and all the features on this moving object are correctly classified.

5.2 Qualitative Study

For the real KITTI dataset, similarly to vKITTI, only short clips with multiple moving objects are tested in our experiments. Here we select several challenging situations for illustration. For instance, Figure 8 shows a segmentation result

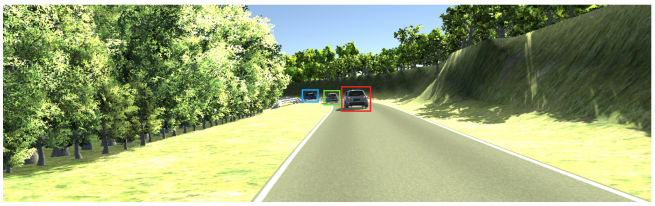


Figure 7: Sample frame from sequence 0020 in virtual KITTI Dataset, where most of the frames contain small size objects (highlighted by bounding boxes) in the scene. This increases the difficulty in detecting features in the objects.

Figure 8: Segmentation result of strong perspective scene with occlusion. Sample frame from sequence 0004 in KITTI Dataset.

of a crossroad, where the scene has strong perspective view. Besides that, there are several vehicles crossing, and some of them are intersected with occlusions. Despite that, the proposed algorithm can still overcome these challenges and deliver a satisfactory segmentation result.

Another example of difficult detection and recovery of the motion is when the scene contains degenerate motions, such as objects moving on the Epipolar plane. This situation is very frequent in real scenarios, e.g. following the car ahead on the road, as can be seen in Figure 9. The result indicates that the proposed method is still able to detect and segment this type of motion consistently, and continuously track the object across frames.

6 Conclusions

In this paper, we introduced a robust and effective system that can simultaneously detect, segment and track rigid moving objects in multiple frames. The proposed system is able to deal with real challenging scenarios with strong perspective, occlusion and degenerate motion. Experimental results indicate that our system helps to achieve excellent performance in segmenting and tracking motions in urban scene and obtain satisfactory results in the virtual KITTI dataset, as well as the real KITTI dataset.

Nevertheless, there are problems that remain to be solved. One of them is how to improve the feature tracking ability on moving objects consistently over frames, which ensures the continuous tracking of objects. This is an unavoidable challenge in real scenes due to light condition change and occlusion. As mentioned before, we plan to integrate the proposed motion segmentation technique into a multi-body

Table 1: AVERAGE MISCLASSIFICATION RATE AND REPROJECTION ERROR FOR THE SEQUENCES FROM THE VIRTUAL KITTI DATASET.

Sequence	Total Error	F/P Error	F/N Error	N/D Error	R/P Error	Num of Motions	Num of Frames
0001	6.2214%	0.0362%	0.7089%	5.4762%	0.7285	3	42
0002	21.0520%	0.1933%	4.3047%	16.5540%	0.6262	5	51
0006	14.0884%	0.0000%	0.9996%	13.0889%	0.5523	6	40
0018	14.1468%	0.1035%	0.0763%	13.9669%	0.5843	4	43
0020	44.7693%	0.0000%	0.0000%	44.7693%	0.0000	3	56

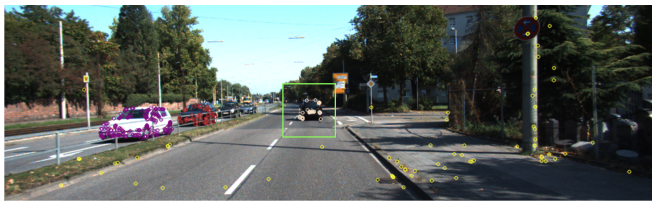


Figure 9: Example of difficult to be detected degenerate motion, which is marked in the green bounding box. Sample frame from sequence 0004 in KITTI Dataset.

SLAM system able to localise the agent camera, track the moving objects and build a dynamic map of the environment. The final solution will aim to improve robustness in situations including distant objects, occlusions, degenerate motions and continuous tracking.

ACKNOWLEDGMENT

This research is supported by the Australian Research Council through the Australian Centre of Excellence for Robotic Vision (CE140100016), and the Discovery Early Career Researcher Award (DECRA) Program (DE150101365). The authors would like to thank Mr. Gerard Kennedy and Mr. Mina Henein for providing help in preparing the testing datasets.

References

[1] K.-i. Kanatani, "Motion segmentation by subspace separation and model selection," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 586–591.

[2] A. Gruber and Y. Weiss, "Multibody factorization with uncertainty and missing data using the em algorithm," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, pp. I–I.

[3] R. Vidal and R. Hartley, "Motion segmentation with missing data using power factorization and gpca," in *null*. IEEE, 2004, pp. 310–316.

[4] H. Li, "Two-view motion segmentation from linear programming relaxation," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[5] S. R. Rao, A. Y. Yang, S. S. Sastry, and Y. Ma, "Robust algebraic segmentation of mixed rigid-body and planar motions from two views," *International journal of computer vision*, vol. 88, no. 3, pp. 425–446, 2010.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[7] Z. Li, J. Guo, L.-F. Cheong, and S. Zhiying Zhou, "Perspective motion segmentation via collaborative clustering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1369–1376.

[8] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2790–2797.

[9] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[10] X. Xu, L. F. Cheong, and Z. Li, "Motion segmentation by exploiting complementary geometric models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2859–2867.

[11] T. Lai, H. Wang, Y. Yan, T.-J. Chin, and W.-L. Zhao, "Motion segmentation via a sparsity constraint," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 973–983, 2017.

[12] C. Rubino, A. D. Bue, and T.-J. Chin, "Practical motion segmentation for urban street view scenes," in *Robotics and Automation (ICRA), 2018 IEEE International Conference on*. IEEE, 2018.

[13] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers, "Efficient dense scene flow from sparse or dense stereo data," in *European conference on computer vision*. Springer, 2008, pp. 739–751.

[14] J. Čech, J. Sanchez-Riera, and R. Horaud, "Scene flow estimation by growing correspondence seeds," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3129–3136.

[15] C. Vogel, K. Schindler, and S. Roth, "Piecewise rigid scene flow," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1377–1384.

[16] Z. Lv, C. Beall, P. F. Alcantarilla, F. Li, Z. Kira, and F. Dellaert, "A continuous optimization approach for efficient and accurate scene flow," in *European Conference on Computer Vision*. Springer, 2016, pp. 757–773.

[17] G. S. Chirikjian, R. Mahony, S. Ruan, and J. Trumpf, "Pose changes from a different point of view," in *Proceedings of the ASME International Design Engineering Technical Conferences (IDETC) 2017*. ASME, 2017.

[18] M. Henein, G. Kennedy, V. Ila, and R. Mahony, "Simultaneous localization and mapping with dynamic rigid objects," *arXiv preprint arXiv:1805.03800*, 2018.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[20] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, 2014.

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[22] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *CVPR*, 2016.

[23] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.