Dynamic Body VSLAM with Semantic Constraints

N. Dinesh Reddy [1], Prateek Singhal[2], Visesh Chari[1,3] and K. Madhava Krishna[1]

***Abstract*— Image based reconstruction of urban environments is a challenging problem that deals with optimization of large number of variables, and has several sources of errors like the presence of dynamic objects. Since most large scale approaches make the assumption of observing static scenes, dynamic objects are relegated to the noise modelling section of such systems. This is an approach of convenience since the RANSAC based framework used to compute most multiview geometric quantities for static scenes naturally confine dynamic objects to the class of outlier measurements. However, reconstructing dynamic objects along with the static environment helps us get a complete picture of an urban environment. Such understanding can then be used for important robotic tasks like path planning for autonomous navigation, obstacle tracking and avoidance, and other areas.**

In this paper, we propose a system for robust SLAM that works in both static and dynamic environments. To overcome the challenge of dynamic objects in the scene, we propose a new model to incorporate *semantic constraints* into the reconstruction algorithm. While some of these constraints are based on multi-layered dense CRFs trained over appearance *as well as motion cues*, other proposed constraints can be expressed as additional terms in the bundle adjustment optimization process that does iterative refinement of 3D structure and camera / object motion trajectories. We show results on the challenging KITTI urban dataset for accuracy of motion segmentation and reconstruction of the trajectory and shape of moving objects relative to ground truth. We are able to show average relative error reduction by 41 % for moving object trajectory reconstruction relative to state-of-the-art methods like TriTrack[16], as well as on standard bundle adjustment algorithms with motion segmentation.

I. INTRODUCTION

Vision based SLAM (vSLAM) is becoming an increasingly widely researched problem, partly because of its ability to produce good quality reconstructions with affordable hardware, and partly because of increasing computational power that results in computational affordability of huge optimization problems. While vSLAM systems are maturing and getting progressively complicated, the two main components remain camera localization (or camera pose estimation) and 3D reconstruction. Generally, these two components precede an optimization based joint refinement of both camera pose and 3D structure, called bundle adjustment.

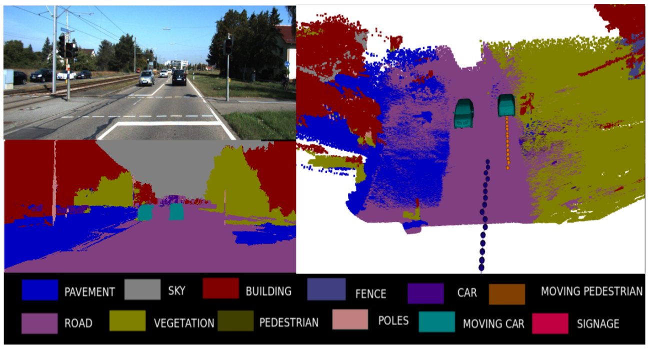


Fig. 1: **Overview of our approach**: **Top left** A frame from highway sequence of the KITTI dataset. **Bottom left** Semantic Motion Segmentation to provide result. **Right** 3d reconstruction with overlaid semantic map and trajectories of the moving objects and camera. (Best viewed in color)

In urban environments, vSLAM is challenging particularly because of the presence of *dynamic objects*. Indeed, it is difficult to capture videos of a city without observing moving objects like cars or people. However, dynamic objects are a source of error in vSLAM systems, since the basic components of such algorithms make the fundamental assumption that the world being observed is static. While optimization algorithms are designed to handle random noise in observations, dynamic objects are a source of *structured* noise since they do not conform to models of random noise distributions (like Gaussian distributions, for example). To overcome such difficulties, RANSAC based procedures for camera pose estimation and 3D reconstruction have been developed in the past, which treat dynamic objects as outliers and remove them from the reconstruction process.

While successful attempts have been made to isolate and discard dynamic objects from such reconstruction processes, there are several recent applications that *benefit* from *reconstructions* of such objects. For example, reconstructing dynamic urban traffic scenes are useful since traffic patterns can be studied to produce autonomous vehicles that can better navigate such situations. Reconstructing dynamic objects are also useful in indoor environments when robots need to identify and avoid moving obstacles in their path [5].

Reconstructing dynamic objects in videos present several challenges. Firstly, moving objects in images and videos have to be segmented and isolated, before they can be reconstructed. This in itself is a challenging problem in the presence of image noise and scene clutter. Degeneracies in camera motion also prevent accurate motion segmentation of such objects. Secondly, upon isolation, a separate vSLAM procedure must be initialized for *each* moving object, since objects like cars often move independent of each other and thus have to be treated as such. Often moving objects like cars occupy only a small portion of the image space in a video (Figure 1), because of which dense reconstructions are infeasible since getting long accurate feature correspondence tracks for such objects is difficult. Absence of large number of feature correspondences also hinders accurate estimation of the car's pose with respect to a world coordinate system. Finally, such objects cannot be reconstructed in isolation from the static scene, since optimization algorithms like bundle adjustment do not preserve contextual information

[1] with affiliation to International Institute of Technology, Hyderabad
[2] with affiliation to Georgia Institute of Technology, Atlanta
[3] with affiliation to the WILLOW group, INRIA, Paris

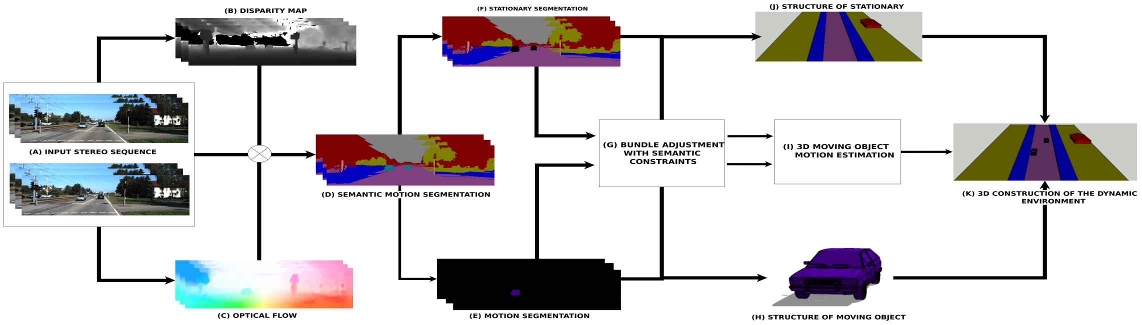

(A) INPUT STEREO SEQUENCE
(B) DISPARITY MAP
(C) OPTICAL FLOW
(D) SEMANTIC MOTION SEGMENTATION
(E) MOTION SEGMENTATION
(F) STATIONARY SEGMENTATION
(G) BUNDLE ADJUSTMENT WITH SEMANTIC CONSTRAINTS
(H) STRUCTURE OF MOVING OBJECT
(I) 3D MOVING OBJECT MOTION ESTIMATION
(J) STRUCTURE OF STATIONARY
(K) 3D CONSTRUCTION OF THE DYNAMIC ENVIRONMENT

Fig. 2: **Illustration of the proposed method**. The system takes a sequence of rectified stereo images (A). Our formulation computes the semantic motion segmentation (D) using the depth(B) and optical flow(C) information. We segment the moving objects (E) from the stationary background (F). We compute accurate structure of the static background (J) and the moving object (H) with the help of bundle adjustment (G). This leads to state-of-the-art 3d reconstruction of the dynamic environment(K) with the help of moving object trajectory estimation(I). (Best viewed in color.)

like the fact that the car must move along a direction perpendicular to the normal of the road surface.

In this paper, we look at the problem of dynamic scene reconstruction. We present an end-to-end system that takes a video, segments the scene into static and dynamic components and reconstructs *both* static and dynamic objects separately. Additionally, while reconstructing the dynamic object, we impose several novel constraints into the bundle adjustment refinement that deal with noisy feature correspondences, erroneous object pose estimation, and contextual information. To be precise, we propose the following contributions in this paper

- We use a *new semantic motion segmentation* algorithm using multi-layer dense CRF which provides state-of-the-art motion segmentation and object class labelling.
- For the first time to our knowledge, we incorporate *semantic contextual information* like support relations between the road surface and object motion, which helps better localize the moving object's pose vis-a-vis the world coordinate system, and also helps in reconstructing them.
- We describe a *novel random sampling strategy* that enables us to maintain the feasibility of the optimization problem in spite of the addition of a large number of variables. Using this approach we drastically reduce the size of our optimization problem *without* compromising on resultant accuracy.

We evaluate our system on 4 challenging KITTI Urban tracking datasets captured using a stereo camera. We are able to achieve average relative error reduction by 41.58 % based on Root Mean Square Error of Absolute Track Error relative to TriTrack [16], while we get an improvement of 13.89 % relative to traditional bundle adjustment after using our novel semantic motion segmentation.

This paper is organized as follows. We cover related work in Section II, and present a system overview in Section III. We describe process of motion segmentation using object class semantic constraints in Section IV. We track and initialize multiple moving bodies which we then optimize using a novel bundle adjustment in Section V. Finally we show experimental results on challenging datasets in Section VI, and conclude in Section VII.

II. RELATED WORK

Our system involves several components like semantic motion segmentation, dynamic body reconstruction using multi-body vSLAM, and trajectory optimization. Table I compares components of our approach with works in recent literature. In recent literature, TriTrack [16] is the closest approach to our method and we first explain it in detail as we compare our method to it in Section VI.

TriTrack [16] is an approach for scene reconstruction, when a moving camera is observing a dynamic scene. It proceeds by first isolating and reconstrucing the trajectory of the camera using an odometry algorithm called VISO2 [17], with dense feature matching and stereo computation as key components. The computed camera motion is then passed over to a sparse scene flow segmentation algorithm to do motion segmentation in 3D, followed by independent trajectory optimization of the segmented moving objects VISO2. Our approach improves over both motion segmentation and trajectory optimization using semantic constraints.

We now focus on each one of the components of our algorithm and draw references to relevant works in the literature in this section.

A. Dynamic body reconstruction

Dynamic body reconstruction is a relatively new development in 3D reconstruction with sparse literature on it. The few solutions in the literature can be categorized into decoupled and joint approaches. Joint approaches like [6] use monocular cameras to jointly estimate the depth maps, do motion segmentation and motion estimation of multiple bodies. Decoupled approaches like [7] [8] have a sequential pipeline where they segment motion and independently reconstruct the moving and static scenes. Our approach is a decoupled approach but essentially differs from other approaches, as we use a novel algorithm for semantic motion segmentation which is leveraged to obtain accurate localization of the moving objects through smoothness and planar constraints to give an accurate dynamic semantic map.

B. Semantic motion segmentation

Semantics have been used extensively for reconstruction [1] [4] [3] but haven't been exploited in motion segmentation till recently [18]. Generally, motion segmentation has been

Method	Outdoor	Stereo	MS	SR	MR	SBA
Sengupta *et al.*[1]	✓	✓		✓		
Hane *et al.*[3]	✓	✓		✓		
Jianxiong *et al.*[5]				✓		
kundu *et al.*[8]	✓		✓			
valentin *et al.*[4]	✓	✓		✓		
Vineet *et al.*[16]	✓	✓		✓	✓	
TriTrack [16]	✓	✓	✓		✓	
OURS	✓	✓	✓	✓	✓	✓

TABLE I: Comparison with related work. MS=Motion Segmentation, SR=Semantic Reconstruction, MR=Motion Reconstruction, SBA=Semantic Bundle Adjustment

approached using geometric constraints [8] or by using affine trajectory clustering into subspaces [9]. In our approach we use motion *along* with semantic cues to segment the scene into static and dynamic objects, which allows us to work with fast moving cars, occlusions and disparity failure. We show a typical result of the motion segmentation algorithm in (Figure 1)(bottom left) where each variable is labelled for both multi-variate semantic class and binary motion class.

C. Multi-body vSLAM

In dynamic scenes, decoupled approaches have motion segmentation followed by tracking each independently moving object to perform vSLAM. Traditional SLAM approaches with single motion model fail in such cases, as moving bodies cause reconstruction errors. Our approach employs Multi Body vSLAM framework [8] where we propose a novel trajectory optimization to with semantic constraints to show dense reconstruction results of moving objects.

D. Semantic constraints for reconstruction

Recent approaches to 3D reconstruction have either used semantic information in a qualitative manner [1], or have only proposed to reconstruct indoor scenes using such information [5]. Only Yuan *et al.* [7] propose to add semantic constraints for reconstruction. While our approach is similar to theirs, they use strict constraints for motion segmentation without regard to appearance information whereas our approach works for more general scenarios as it employs a more powerful inference engine in the CRF.

III. SYSTEM OVERVIEW

We give an illustration of our system in Figure 2. Given rectified input images from a stereo camera, we first compute low level features like SIFT descriptors, optical flow (using DeepFlow [14]) and stereo [20]. These are then used to compute semantic motion segmentation, as explained in Section IV. Once semantic segmentation is done per image, we isolate stationary objects from moving objects and reconstruct them independently. To do this, we connect moving objects across frames into tracks by computing SIFT matches on dense SIFT features [22]. Then we perform camera resectioning using EPnP [25] for stationary and ICP for moving objects, to register their 3D points across frames. This is then followed by bundle adjustment with semantic constraints (Section V), where we make use of the semantic and motion labels assigned to the segmented scene to obtain accurate 3D reconstruction. We then fuse the stationary and moving object reconstructions using an algorithm based on the truncated signed distance function (TSDF) [19]. Finally, we transfer labels from 2D images to 3D data by projecting 3D data onto the images, and using a winner-takes-it-all approach to assign labels to 3D data from the labels of the projected points.

IV. SEMANTIC MOTION SEGMENTATION

In this section, we deal with the first module of our system. A sample result of our segmentation algorithm is shown in Figure 1. With input images from a stereo camera, we give an overview on how we perform semantic segmentation [11] to first separate dynamic objects from the static scene. We combine classical semantic segmentation with a new set of motion constraints proposed in [18] to perform semantic motion segmentation, that *jointly* optimizes for semantic and motion segmentation. While we give an overview of the formulation in this section, for brevity, methodologies used for training, testing and the rationale behind using mean field approximations is outlined in [18].

We do joint estimation of motion and object labels by exploiting the fact that they are interrelated. We formulate the problem as a joint optimization problem of two parts, object class segmentation and motion segmentation. We define a dense CRF where the set of random variables $Z = \{Z_1, Z_2, ..., Z_N\}$ corresponds to the set of all image pixels $i \in \mathcal{V} = \{1, 2, ..., N\}$. Let $\mathcal{N}_i$ denote the neighbors of the variable Z_i in image space. Any possible assignment of labels to the random variables will be called a labelling and denoted by z. We define the energy of the joint CRF as

$$E^{\mathscr{J}}(z) = \sum_{i \in \mathcal{V}} \psi_i^{\mathscr{J}}(z_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{i,j}^{\mathscr{J}}(z_i, z_j) \tag{1}$$

where $\psi_i^{\mathscr{J}}$ is the joint unary potential and $\psi_{i,j}^{\mathscr{J}}$ represents the joint pairwise potential. We describe these terms in brief in the next two sections.

A. **Joint Unary Potential:**

The joint unary potential $\psi_i^{\mathscr{J}}$ is defined as an interactive potential term which incorporates a relationship between the object class and the corresponding motion likelihood for each pixel. Each random variable Z_i = $[X_i, Y_i]$ takes a label z_i = $[x_i, y_i]$, from the product space of object class and motion labels. The combined unary potential of the joint CRF is

$$\psi_{i,l,m}^{\mathscr{J}}([x_i, y_i]) = \psi_i^{O}(x_i) + \psi_i^{\mathscr{M}}(y_i) + \psi_{i,l,m}^{O\mathscr{M}}(x_i, y_i) \tag{2}$$

The object class unary potential $\psi_i^{O}(x_i)$ describes the cost of the pixel taking the corresponding label and is computed using pre-trained models of color, texture and location features for each object as in [2]. The new motion class unary potential $\psi_i^{\mathscr{M}}(y_i)$ is given by the motion likelihood of the pixel and is computed as the difference between the predicted and the measured optical flow. The measured flow is computed using dense optical flow. The predicted flow measures how much the object needs to move given its depth in the current image and assuming it is a stationary object. Objects deviating from the predicted flow are likely to be dynamic objects. It is computed as

$$\hat{X}' = KRK'X + KT/z \tag{3}$$

where K is the intrinsic camera matrix, R and T are the translation and rotation of the camera respectively and z is the depth [18]. X is the location of the pixel in image coordinates and $\hat{X}'$ is the predicted flow vector of the pixel given from the motion of the camera. Thus the unary potential is now computed as

$$\psi_i^{\mathscr{M}}(x_i) = ((\hat{X}' - X')^T \Sigma^{-1} (\hat{X}' - X')) \tag{4}$$

where Σ is the sum of the covariances of the predicted and measured flows as shown in [15], & $\hat{X}' - X'$ represents the difference of the predicted flow and measured flow. The object-motion unary potential $\psi_{i,l,m}^{O\mathscr{M}}(x_i,y_j)$ incorporates the object-motion class compatibility and can be expressed as

$$\psi_{i,l,m}^{O\mathscr{M}}(x_i,y_j) = \lambda(l,m) \tag{5}$$

where $\lambda(l,m) \in [-1,1]$ is a learnt correlation term between the motion and object class label. $\psi_{i,l,m}^{O\mathscr{M}}(x_i,y_j)$ helps in incorporating the relationship between an object class and its motion (for example, trees and roads are stationary, but cars move). We use a piecewise method for training the label and motion correlation matrices using the modified Adaboost framework [18], as described in [18].

B. Joint Pairwise Potential:

The joint pairwise potential $\psi_{ij}^{\mathscr{J}}(z_i,z_j)$ enforces the consistency of object and motion class between the neighboring pixels. We compute the joint pairwise potential as

$$\psi_{ij}^{\mathscr{J}}([x_i,y_i],[x_j,y_j]) = \psi_{ij}^{O}(x_i,x_j) + \psi_{ij}^{\mathscr{M}}(y_i,y_j) \tag{6}$$

where we disregard the joint pairwise term over the product space. The object class pairwise potential takes the form of a Potts model

$$\psi_{i,j}^{O}(x_i,x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ p(i,j) & \text{if } x_i \neq x_j \end{cases} \tag{7}$$

where $p(i,j)$ is given as the standard pairwise potential as given in [12].

The motion class pairwise potential $\psi_{i,j}^{\mathscr{M}}(y_i,y_j)$ is given as the relationship between neighboring pixels and encourages the adjacent pixels in the image to have similar motion label. The cost of the function is defined as

$$\psi_{ij}^{\mathscr{M}}(y_i,y_j) = \begin{cases} 0 & \text{if } y_i = y_j \\ g(i,j) & \text{if } y_i \neq y_j \end{cases} \tag{8}$$

where $g(i,j)$ is an edge feature based on the difference between the flow of the neighboring pixels ($g(i,j) = |f(y_i) - f(y_j)|$) & $f(\cdot)$ is returns the flow of the corresponding pixel.

C. Inference and learning:

We follow Krahenbuhl et al [12] to perform inference on this dense CRF using a mean field approximation. In this approach we try to find a mean field approximation $Q(z)$ that minimizes the KL-divergence $D(Q\|P)$ among all the distributions Q that can be expressed as a product of independent marginals, $Q(z) = \prod_i Q_i(z_i)$. We can further factorize Q into a product of marginals over multi-class object and binary motion segmentation layer by taking $Q_i(z_i) = Q_i^O(x_i)Q_i^{\mathscr{M}}(y_i)$. Here Q_i^O is a multi-class distribution over the object labels, and $Q_i^{\mathscr{M}}$ is a binary distribution over moving or stationary classes ($Q_i^{\mathscr{M}} \in \{0,1\}$). We compute inference separately for both the layers i.e object class layer and motion layer [18].

V. TRAJECTORY ESTIMATION

We isolate pixels belonging to moving objects from static objects in the motion segmented images which are the output of our semantic motion segmentation algorithm. Pixels belonging to each type of object (static or motion) are then used as input to localize and map each object independently. In this section, we propose a novel framework for trajectory computation for static or moving objects from a moving platform. The below process is carried out for all the moving objects and the camera mounted vehicle[1]. Let us introduce some preliminary notations for trajectory computation. The extrinsic parameters for frame $k = 1,2,3,4...n$ are the rotation matrix R_k and the camera center C_k relative to a world coordinate system. Then the translation vector between the world and the camera coordinate systems is T_k = -R_kC_k .

a) **Trajectory Initialization:** We initialize the motion of each object separately using SIFT feature points. SIFT feature points are tracked using dense optical flow between consecutive pair of frames. Key points with valid depth values are used in a 3-point-algorithm within a RANSAC framework to find the robust relative transformation between pairs of frames. We obtain pose estimates of the moving object in the world frame by chaining the relative transformations together in succession.

For moving objects the initial frame k where detection occurs is taken as the starting point. Trajectory estimates are then initialized for each object independently corresponding to the frame k assuming the camera is static.

b) **3D Object Motion Estimation:** Once 3D trajectories are estimated for each object independently, we need to map these trajectories onto the world coordinate system. Since, we are dealing with stereo data and for every frame we have 3D information, this mapping can be represented as simple coordinate transformations. Also, since we are not dealing with monocular images, the problem of relative scaling can be avoided.

Given the pose of the real camera in the k^{th} frame $((R_k^c, T_k^c))$ and virtual camera (R_k^v, T_k^v) [7] computed during trajectory initialization described earlier, we should be able to compute the pose of the b^{th} object (R_k^b, T_k^b) relative to its original position in the first frame in the world coordinate system. The object rotation R_k^b and translation T_k^b are given as

$$R_k^b = (R_k^c)^{-1} R_k^v, \qquad T_k^b = (R_k^c)^{-1}(T_k^v - T_k^c) \tag{9}$$

Thus we get the localization and sparse map of both the static and moving world. We found this approach to object motion estimation to be better on both small and long sequences than TriTrack [16].

A. Dynamic Object Trajectory Optimization

Once 3D object motion and structure initialization has been done, we need to refine the structure and motion using

[1] Henceforth referred as camera

bundle adjustment (BA). In this section, we describe our framework for BA to refine the trajectory and sparse 3D point reconstruction of dynamic objects along with *several novel* constraints added to BA that increase the accuracy of our trajectories and 3D points. We term these constraints *semantic or contextual* constraints since they represent our *understanding* of the world in a geometric language, which we use to effectively optimize 3D points and trajectories in the presence of noise and outliers. These semantic constraints are a consequence of the semantic motion labels acquired from the semantic motion segmentation algorithm (Section **??**). The assumptions underlying these constraints derive from *commonly* observed *shape* and *motion* traits of cars in urban scenarios. For example the normal constraints follow the logic that the motion of a dynamic object like a vehicle is always on a plane (the road surface) and hence constrained by its normal. Similarly, the 3D points on a dynamic object are constrained to lie within a 3D "box" since dynamic objects like cars cannot be infinitely large. Finally, our trajectory constraints encode the fact that dynamic objects have smooth trajectories, which is often true in urban scenarios. In summary, we try to minimize the following objective function

$$min\sum_{\text{i}}\sum_{p\in V(i)}\texttt{BA2D}+\lambda\texttt{BA3D}+\lambda\texttt{TC}+\texttt{NC}+\texttt{BC} \qquad (10)$$

where BA2D represents the 2D BA reprojection error ($\|\tilde{x}_p^i - K[R_i \mid T_i]X_p\|^2$), BA3D represents the 3D registration error common in optimization over stereo images ($\|\tilde{X}_p^i - [R_i \mid T_i]X_p\|^2$) and TC, NC, BC represent various optimization terms that can be seen as imposed constraints on the resulting shape and trajectories as explained below. Here i indexes into images, and ˜ represents variables in the camera coordinate system, with other quantities being expressed in the world coordinate system. Also, $p \in V(i)$ represents pixels visible in image i.

1) **Planar Constraint:** We constrain motion to be perpendicular to the ground plane where the ground plane normal is found from the initial 3D reconstruction of the ground.

$$\texttt{NC1}: \quad N_g\cdot(T_c^k - T_c^{k-1}) \qquad (11)$$

where N_g is the normal of the ground plane in the camera frame, T_c^k - T_c^{k-1} is the direction of camera motion in the local coordinate system. This local motion and normal estimation allows us to use the same constraint even on changing planes like up or down a slope. Since 3D reconstruction of the ground can be noisy, estimation of N_g is done using least squares. Alternatively, we could follow a RANSAC based framework of selecting m top hypotheses for the normal N_g^i ($i = 1\dots m$), and allow bundle adjustment to minimize an average error of the form

$$\texttt{NC2}: \quad \sum_{i=1}^{m} N_g^i\cdot(T_c^k - T_c^{k-1}) \qquad (12)$$

2) **Smooth Trajectory Constraints:** We enforce smoothness in trajectory, a valid assumption for urban scenes, by constraining camera translations in consecutive frames as

$$\texttt{TC1}: \quad \|(T_c^{k+1} - T_c^k)\times(T_c^k - T_c^{k-1})\| \qquad (13)$$

where $T_c^{k+1}, T_c^k, T_c^{k-1}$ are the 3d translations at frame k+1, k and k-1. Alternatively, we could also minimize the norm between two consecutive translations unlike TC1, which only penalizes direction deviations in translation.

$$\texttt{TC2}: \quad \|(T_c^{k+1} - 2*T_c^k + T_c^{k-1})\|^2 \qquad (14)$$

3) **Box Constraints:** Depth estimation of objects like cars are generally noisy because their surface is not typically Lambertian in nature, and hence violates the basic assumptions of brightness constancy across time and viewing angle. Furthermore, noise in depth infuses errors into the estimated trajectory through the trajectory initialization component. To improve the reconstruction accuracy in such cases, and to limit the destructive effect that noisy depth has on object trajectories, we introduce shape priors into the BA cost function that essentially constrains all the 3D points belonging to a moving object to remain with a "box".KITTI More specifically, let X_i^b & X_j^b be two 3D points on a moving object O^b. For every such pair of points on the object, we define the following constraint

$$\texttt{BC1}: \quad \sum_{\forall X_i^b, X_j^b \in O^b} \|X_i^b - X_j^b - B(i,j)\|^2 \qquad (15)$$
$$-\delta \le B(i,j) \le \delta$$

where $B(i,j)$ is a vector of bounds with individual components $(b_x(i,j), b_y(i,j), b_z(i,j))$ and δ is a vector of positive values.

Note that the above equation is defined for every pair of points on the object, which leads to a *quadratic explosion* of terms since $B(i,j)$ is a separate variable for each pair.

a) **Alternate Formulations:** One way to reduce the explosion would be to reduce the number of variables added because of the box constraints to BA. This could be done by alternatively minimizing the following terms instead of the constraint in equation (15)

$$\texttt{BC2}: \quad \sum_{\forall (X_i^b, X_j^b)\in O^b} \|X_i^b - X_j^b - b(i,j)\|^2, -\delta \le b(i,j) \le \delta \qquad (16)$$

$$\texttt{BC3}: \quad \sum_{\forall (X_i^b, X_j^b)\in O^b} \|X_i^b - X_j^b - B\|^2, -\delta \le B \le \delta \qquad (17)$$

$$\texttt{BC4}: \quad \sum_{\forall (X_i^b, X_j^b)\in O^b} \|X_i^b - X_j^b - b\|^2, -\delta \le b \le \delta \qquad (18)$$

where $b(i,j)$ in equation (16) is a scalar common to all 3 dimensions, B (equation (17)) is a 3×1 vector common to all point pairs, and b (equation (18)) is a scalar common to all pairs and dimensions.

b) **Alternate Minimization Strategies:** It is now known that a lot of information in terms like BC1,BC2,BC3,BC4 above are redundant in nature [21], and there is essentially a small "subset" of pairs which is sufficient to produce optimal or near-optimal results in such cases. However, it is not clear how to pick this small subset. Here, we take the help of the Johnson-Lindenstrauss theorem and its variants [23], [24], to select a random set of pairs from the ones available, such that we closely approximate the BC error when all the point pairs are used.

More specifically, the terms expressed in BC1,BC2,BC3,BC4 can all be expressed in the form

$$\texttt{BCLin}: \quad \|AX-B\|^2, \text{ such that } ,CB=D \qquad (19)$$

where X is a concatenation of all 3D points, and B is a collection of all box bounds. The matrix A is constructed in such a way that each row of A consists of only two non-zero elements at the i^{th} and j^{th} positions with values 1 and -1 respectively, and they represent the difference $X_i^b - X_j^b$. The linear constraint $CB = D$ is useful to represent the fact that some elements of vector B are equal to others. While this is useful to represent BC2,BC3,BC4 (BC1 can be exactly represnted without this constraint) we temporarily "relax" this constraint, and enforce it post-optimization by taking the average of duplicate variables. Note that the dimensions of A are of the order $3^nC_2 \times 3n$, where n is the number of 3D points. Notice that for $n = 3000$, ${}^nC_2$ is approximately 4.5 million, and is highly slow to optimize! To reduce this computational burden, we embed the above optimization problem in a randomly selected subspace of considerably lower dimension, with the guarantee that the solution obtained in the subspace is close to the original problem solution with high probability. To do this, we draw upon a slightly modified version of the *affine embedding* theorem presented in [24] which states

Theorem 5.1: For any minimization of the form $\|AX - B\|$, where A is of size $m \times n$ and $m \gg n$, there exists a *subspace embedding matrix* $S : \mathbb{R}^m \mapsto \mathbb{R}^t$ where $t = poly(n/\varepsilon)$ such that

$$\|SAX - SB\|_2 = (1 \pm \varepsilon)\|AX - B\|_2 \qquad (20)$$

Moreover, the matrix S of size $t \times m$ is designed such that each column of S has only 1 non-zero element at a randomly chosen location, with value 1 or -1 with equal probability.

Note that since elements of S are randomly assigned 1 or -1, the above transformation cannot be exactly interpreted as a random sampling of pairs of points. However for the sake of implementation simplicity, we "relax" S to a random selection matrix. As we show later, empirically we get very satisfying results.

Finally, there can be several strategies to select random pairs of points for box constraints. We experimented with the following in this paper.

- Strat1: Randomly select pairs from the available set.
- Strat2: Randomly select one point, and create its pair with the 3D point that is farthest from the selected point in terms of Euclidean distance.
- Strat3: Randomly select one point, and sort other points in descending order based on Euclidean distance with selected point. Pick the first point from the list that has not been part of any pair before.

Once the proper set of constraints are selected from the above choices, the final objective function in equation 10 is minimized with L_2 norm using CERES solver. [13].

VI. EXPERIMENTAL RESULTS

In this section we provide extensive evaluation of our algorithms on both synthetic and real data. For real datasets, we have used the KITTI tracking dataset for evaluation of the algorithm as the ground truth for localization of moving objects per camera frame is available. It consists of several sequences collected by a perspective car-mounted camera driving in urban, residential and highway environments, making it a varied and challenging real world dataset. We have taken four sequences consisting of 30, 212, 30 and 100 images for evaluating our algorithm. We choose these 4 sequences as they pose serious challenges to the motion segmentation algorithm as the moving cars lie in the same subspace as the camera. These sequences also have a mix of multiple cars visible for short duration along with cars visible for the entire sequence which allows us to test the robustness of our localization and reconstruction algorithms on both short and long sequences. While we show qualitative results for all the four sequences, we show extensive quantitative evaluation for the longest sequence of 212 frames called **KITTI1** sequence. We plan to release the code and dataset corresponding to our entire system.

We do extensive quantitative evaluation on synthetic dataset as well. We generated 1000 3D points on a cube attached to a planar ground to simulate a car and road. We then move the car over the road, while simultaneously mov-

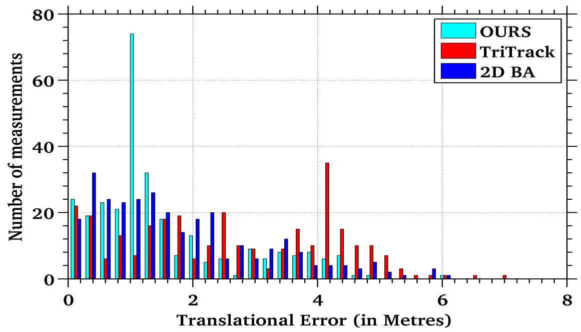
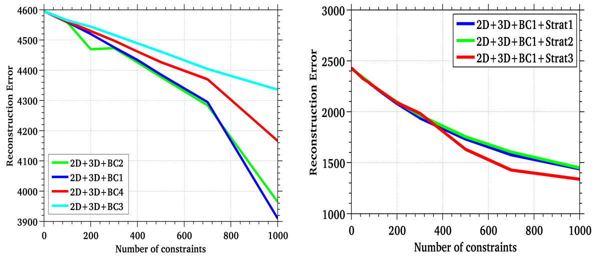


Fig. 3: Comparison of trajectory errors of our algorithm to TriTrack [16] and standard BA after motion segmentation. The histogram plots RMSE magnitude on the x axis, and number of pose measurements that fall in each bin on the y axis. Note that most of our errors are concentrated on the left (low error), while TriTrack [16] and BA are more evenly spread. The total summed error: 2D-BA - 1.79, TriTrack - 2.62, Ours - 1.54.



(a) BC terms

(b) Strat strategies

Fig. 4: Synthetic results for box constraints. Note that in the two experiments we added a large amount of noise and picked 1000 constraints from around 500000 pairs of points, which means we use 0.2% of all available constraints. We infer that BC1 in (a) and Strat3 in (b) are the best performers.

	Error Type	Without MS	MS	MS+NC1	MS+NC1 +TC1
BA2D	rmse	1.416246	1.001566	**0.941971**	0.958505
	mean	1.212164	0.826189	**0.764188**	0.779054
	median	1.088891	**0.677419**	0.690825	0.716546
BA3D	rmse	1.476649	**0.959499**	0.975747	0.978197
	mean	1.272985	**0.786729**	0.822169	0.824090
	median	1.279508	**0.712513**	0.773672	0.769680
BA23D	rmse	1.472399	**0.958505**	0.958541	0.958541
	mean	1.269541	**0.779054**	0.779132	0.779132
	median	1.269238	**0.716546**	0.716967	0.716967

TABLE II: Static scene of **KITTI** dataset. Note that adding Motion Segmentation (MS) drastically improves results, while normal constraints also help in some cases. BA23D = BA2D + BA3D

	Error Type	MS	MS+NC1	MS+NC1 +TC1	MS+NC1+TC1+ BC1 (1000 constr)
BA2D	rmse	2.425649	2.362224	2.351205	**2.302849**
	mean	1.989408	1.955466	1.969793	**1.937154**
	median	1.669304	1.616398	1.685272	**1.640389**
BA3D	rmse	3.627977	3.587194	3.352087	**3.270264**
	mean	2.544718	2.527314	2.398578	**2.367702**
	median	2.000463	1.997689	1.941246	**1.928450**
BA23D	rmse	2.357187	2.305733	2.296139	**2.254192**
	mean	2.035764	1.986784	1.971698	**1.881728**
	median	1.877257	1.759010	1.760857	**1.756554**

TABLE III: Dynamic scene of **KITTI** dataset of 212 frames. Note that adding box constraints over normal and trajectory lead to the best results. BA23D = BA2D + BA3D

ing the camera to generate moving images after projection of the 3D points. Finally we added Gaussian noise to both the 3D points on the car and the points on the road to simulate errors in measurement. Correspondences between frames are automatically known as a result of our dataset design.

A. Quantitative Evaluation of Object Trajectory Optimization

In this section, we do an extensive evaluation of the different terms proposed in Section V. Note that we tried all the different terms and strategies proposed here on real data as well, and in all cases conclusions derived from synthetic data experiments are consistent with real data.

1) **Evaluating Terms and Strategies:** In the following section we present the results for evaluation of various terms and strategies.

a) **Normal Constraint:** This constraint is a contextual constraint in the sense that it enforces the fact that the moving object is usually attached to a planar ground in urban settings, and so any deviation of the object trajectory along the direction of the normal of the ground plane should be penalized. While NC1 computes a least-squares estimate for the normal which is optimal under Gaussian noise, NC2 computes several normal hypotheses using a RANSAC framework. Figure (6a) shows the results comparing the two terms. We find that NC1 normally performs better.

b) **Trajectory Constraint:** The trajectory constraint enforces smoothness in moving object trajectories, by either enforcing that the direction of motion should not change significantly between consecutive frames (TC1) or enforcing that both direction and magnitude must be constrained (TC2). Figure (6b) plots comparative results, and we infer that TC1 performs better.

c) **Box Constraint:** Box constraints enforce that the 3D reconstruction of the moving object in consideration must be *compact*. This is a useful constraint since gross errors in the depth of the object as estimated by the stereo algorithm [20] normally are not corrected by BA since it settles into a local minima. Thus, to "focus" the BA towards better optimizing the 3D structure, we add these constraints.

d) **Box Sampling Strategies:** Since box constraints lead to an explosion of terms added to BA, we experiment with 4 strategies to reduce this computational burden by random sampling [24]. Figure (4) show results for various terms of box constraints, and various strategies to optimize. Normally we find that BC1 along with Strat3 performs best.

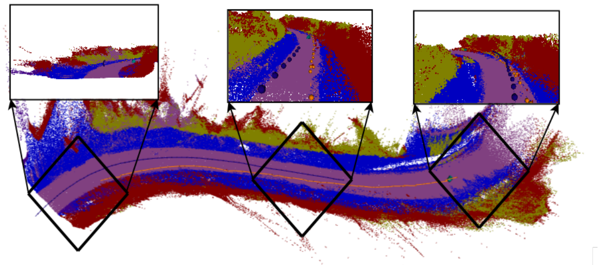
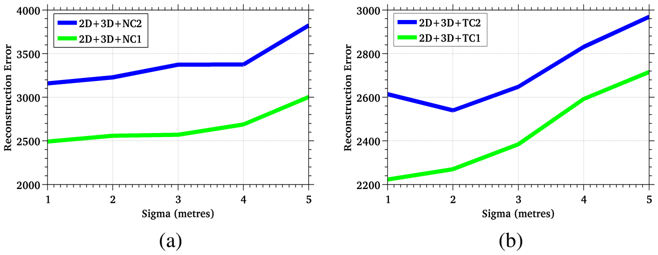


Fig. 5: Reconstruction result for **KITTI 4** sequence. Note the accurate reconstruction of trajectories and of the car and the camera, in spite of curvilinear motion. Please see supplementary video for further details.



(a)

(b)

Fig. 6: Synthetic results for Normal and trajectory constraints.

B. Trajectory Evaluation

We compare the estimated trajectories of the moving objects and the camera to the TriTrack (Stereo) [16] for camera and TriTrack for the moving object trajectories. VISO2 S(Stereo) [17] has reported error of 2.44 % on the KITTI odometry dataset, making it a good baseline algorithm to compare with. As proposed by Sturm et al. [10], the comparison methodology is based on ATE for root mean square error (RMSE), mean, median. We use their evaluation algorithm which aligns the 2 trajectories using SVD. We show all the three statistics, as mean and median are robust to outliers, while RMSE shows the exact deviation from the ground truth.

Table (II) shows results for trajectory error estimation for the static part of the KITTI1 sequence, with dynamic objects removed. As can be seen, we get significant improvement in camera trajectory estimation *after* motion segmentation. This reinforces our claim that motion segmentation is essential for

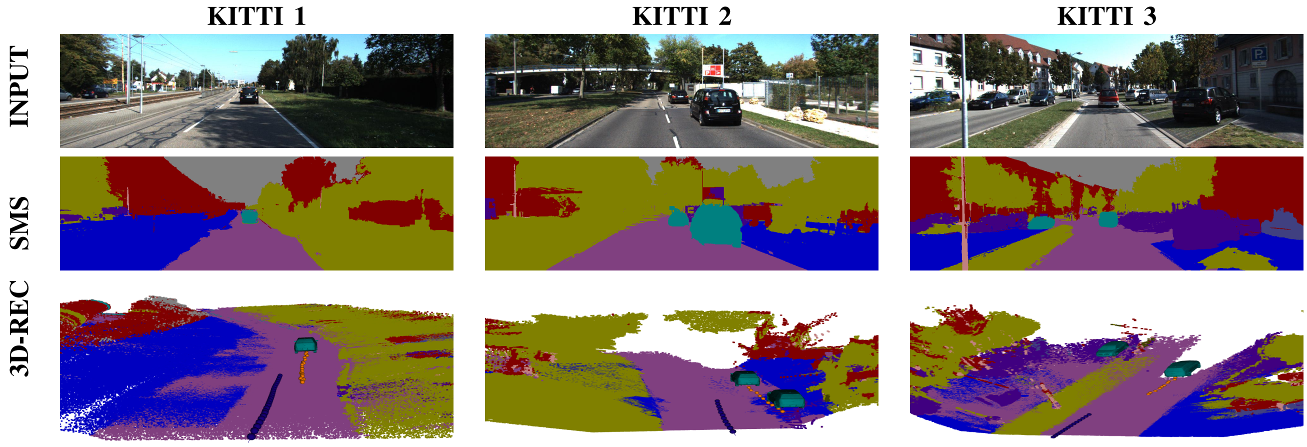


Fig. 7: We show the **(INPUT)** image sequences for which we compute the semantic motion segmentation (**SMS**). We have depicted the reconstruction of moving objects with their trajectories **(3D-REC)**. Blue trajectories represent the camera capturing the scene. All segmentation color labels are consistent with Figure 1. (best viewed in color)

trajectory estimation in dynamic scenes. Since semantic constraints are tailored to dynamic bodies the best improvement using them are seen (across all rows) in table III. Table (III) depicts the trajectory error for the moving object visible in all the 212 images of the sequence. We progressively show how each constraint on the motion of the moving object complements its trajectory computation and reconstruction in successive columns. This further enhances our claim that our semantic constraint on dynamic bodies allows us to localize and reconstruct them more accurately.

For quantitative evaluation of our method on the **KITTI1** sequence, we have computed the trajectories of all the moving objects. These trajectories are compared to their respective ground truth and the absolute position error of each pose is computed. We have done a histogram based evaluation of all the position error as depicted in Fig(3), where we compare the trajectories of our algorithm with TriTrack. We have evaluated the algorithm for a complete of 297 poses of moving objects and found that our approach outperforms TriTrack and standard 2D bundle adjustment. Qualitative results of the trajectories and reconstruction of some of the moving objects is depicted in the Fig(7, 5).

VII. CONCLUSION

In this paper, we have proposed a joint labelling framework for semantic motion segmentation and reconstruction in dynamic urban environments. We modelled the problem of creating a semantic dense map of moving objects in a urban environment using trajectory optimization. The experiments suggest that semantic segmentation provide good initial estimates to aid generalized bundle adjustment based approach. This helps in improving the localization of the moving objects and creates an accurate semantic map.

REFERENCES

[1] Sunando Sengupta and Eric Greveson and A. Shahrokni and Philip H.S. Torr, G. O. Young,Urban 3D Semantic Modelling Using Stereo Vision, In ICRA, 2013.

[2] J. Shotton, J. Winn, C. Rother, and A. Criminisi,Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation, In ECCV, 2006.

[3] C. Hane, C. Zach, A. Cohen, R. Angst, and M. Pollefeys, Joint 3D Scene Reconstruction and Class Segmentation. In CVPR, 2013.

[4] J. Valentin, S. Sengupta, J. Warrell, A. Shahrokni, and P. Torr, Mesh Based Semantic Modelling for Indoor and Outdoor Scenes.In CVPR 2013, pages 2067-2074.

[5] Jianxiong Xiao, Andrew Owens, Antonio Torralba, ”SUN3D: A Database of Big Spaces Reconstructed Using SfM and Object Labels”, in ICCV, 2013.

[6] A. Roussos, C. Russell, R. Garg and L. Agapito, Lourdes,Dense multi-body motion estimation and reconstruction from a handheld camera, in ISMAR, 2012

[7] Chang Yuan and Gerard Medioni,Reconstruction of Background and Objects Moving on Ground Plane Viewed from a Moving Camera, in CVPR, 2006.

[8] Kundu, Abhijit and Krishna, K. Madhava and Jawahar, C.V.,Realtime Multibody Visual SLAM with a Smoothly Moving Monocular Camera, in ICCV, 2011.

[9] E.Elhamifar and R.Vidal. Sparse subspace clustering, in CVPR, 2009.

[10] J. Sturm and N. Engelhard and F. Endres and W. Burgard and D. Cremers. A Benchmark for the Evaluation of RGB-D SLAM Systems,in IROS, 2012.

[11] Ladicky Lubor,Russell Christopher,Kohli Pushmeet and Torr Philip H.S , Associative hierarchical CRFs for object class image segmentation, in ICCV, 2009

[12] W. Choi and C. Pantofaru and S. Savarese,Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials,In NIPS,2011.

[13] Sameer Agarwal and Keir Mierle,Ceres Solver: Tutorial & Reference,Google

[14] Weinzaepfel, Philippe and Revaud, Jerome and Harchaoui, Zaid and Schmid, Cordelia,DeepFlow: Large displacement optical flow with deep matching, In ICCV, 2013.

[15] Victor Romero-Cano and Juan I. Nieto,Stereo-based motion detection and tracking from a moving platform,In IV,2013

[16] Philip Lenz and Julius Ziegler and Andreas Geiger and Martin Roser,Sparse Scene Flow Segmentation for Moving Object Detection in Urban Environments, In IV, 2011.

[17] Bernd Kitt and Andreas Geiger and Henning Lategahn, Visual Odometry based on Stereo Image Sequences with RANSAC based Outlier Rejection Scheme, In IV, 2010.

[18] N Dinesh Reddy, Prateek Singhal and K Madhava Krishna, Semantic Motion Segmentation Using Dense CRF Formulation, In ICVGIP,2014.

[19] Qian-Yi Zhou and Stephen Miller and Vladlen Koltun,Elastic Fragments for Dense Scene Reconstruction, In ICCV, 2013.

[20] K. Yamaguchi, D. McAllester and R. Urtasun, Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation, In ECCV, 2014.

[21] Siddharth Choudhary, Vadim Indelman, Henrik Christensen and Frank Dellaert, Information-based Reduced Landmark SLAM, In ICRA, 2015.

[22] A. Vedaldi and B. Fulkerson, VLFeat: An Open and Portable Library of Computer Vision Algorithms, 2008.

[23] Anirban Dasgupta, Maxim Gurevich, Kunal Punera, A Sparse Johnson-Lindenstrauss Transform, ACM STOC, 2010.

[24] Kenneth Clarkson, David Woodruff, Low Rank Approximation and Regression in Input Sparsity Time, ACM STOC, 2012.

[25] Vincent Lepetit, Moreno-Noguer Francesc and Pascal Fua, EPnP: An Accurate O(n) eolution to the PnP Problem, In IJCV, 2009.

[26] V. Vineet, O. Miksik, M. Lidegaard, *et al.*, Incremental Dense Semantic Stereo Fusion for Large-Scale Semantic Scene Reconstruction, ICRA 2015.