Monocular Reconstruction of Vehicles: Combining SLAM with Shape Priors

Falak Chhaya[1], Dinesh Reddy[1], Sarthak Upadhyay[1], Visesh Chari[1], M. Zeeshan Zia[2] and K. Madhava Krishna[1]

***Abstract*— Reasoning about objects in images and videos using 3D representations is re-emerging as a popular paradigm in computer vision. Specifically, in the context of scene understanding for roads, 3D vehicle detection and tracking from monocular videos still needs a lot of attention to enable practical applications.**

Current approaches leverage two kinds of information to deal with the vehicle detection and tracking problem: (1) 3D representations (eg. wireframe models or voxel based or CAD models) for diverse vehicle skeletal structures learnt from data, and (2) classifiers trained to detect vehicles or vehicle parts in single images built on top of a basic feature extraction step. In this paper, we propose to extend current approaches in two ways. First, we extend detection to a multiple view setting. We show that leveraging information given by feature or part detectors in multiple images can lead to more accurate detection results than single image detection. Secondly, we show that given multiple images of a vehicle, we can also leverage 3D information from the scene generated using a unique structure from motion algorithm. This helps us localize the vehicle in 3D, and constrain the parameters of optimization for fitting the 3D model to image data. We show results on the KITTI dataset, and demonstrate superior results compared with recent state-of-the-art methods, with upto 14.64 % improvement in localization error.

I. Introduction

Recent advances in Simultaneous Localization and Mapping (SLAM) and Structure from Motion (SfM[1]) have resulted in mature technologies, that are beginning to appear in commercial products, from Google Project Tango and Microsoft Hololens to the Dyson 360 Eye and advanced driver assistance systems. While there have been many advances in semantic recognition [10], [14] as well, the state-of-the-art lacks far behind the robustness needed for most applications of robotic perception and scene understanding. Notably though, a couple of recent works have attempted to improve robustness of visual recognition by leveraging known geometry [23], [21] and shown promising results.

Simultaneously, recent research [29], [17], [30], [31], [15], [26] in computer vision has revived detailed 3D geometric representations of object classes from decades earlier [3], when they had not been effective due to lack of computational resources, and inference and learning algorithms. This revival aided by modern discriminative classification, description, and probabilistic inference techniques has shown success albeit limited only to single image understanding. In the present paper, we tightly integrate these deformable 3D object models with state-of-the-art multibody SfM methods, to introduce a system that can outperform the latest results [30], [31] in the domain of highly detailed object modeling and tracking in video. This system allows the recognition and the reconstruction modules to help each other produce better overall results: SfM methods fail on moving, specular vehicles, whereas a single view is often not enough to disambiguate object shape from background clutter. In addition to improving 3D location estimates, extracting accurate 3D shape and pose open the possibility for more sophisticated planning and control downstream in the autonomous vehicle's processing pipeline.

Specifically, we integrate deformable wireframe models of object classes (here, vehicles) that represent object geometry at a finer level than 2D bounding boxes [21] and in a more general "intra-class invariant" manner than instance-specific 3D CAD models [23], into a multibody SLAM framework [18], [24]. Approximating the visible surfaces of a vehicle by planar segments, supported by discriminative part detectors allows us to obtain more stable and accurate 3D reconstruction of moving objects as compared to state-of-the-art SLAM pipelines [8], [19] which are not robust in the face of specular, moving objects. This is because the feature tracks on segmented sequences of specular, moving objects are very sparse and the fundamental matrices [16] used to represent camera motion are often degenerate. By segmenting the car into its constituent planes modeled by homographies and filtered by RANSAC, we obtain superior reconstruction of the camera trajectory in dynamic road scenes. We upgrade the single-view object class formulation of [30], [31] to multiple views. This multi-view deformable wireframe fitting is posed as stochastic hill climbing in the space of vehicle shape and pose parameters (in block coordinate descent iterations) that maximizes part likelihood averaged over multiple images. The projection onto multiple images is enabled by the camera trajectory obtained relative to the moving object in the multibody SLAM framework. Thus we have a pipeline which tightly couples recognition using a rich geometric object model with estimation of camera trajectory for highly dynamic scenes.

Summarily, we list the contributions of the present paper in the following:

1) We propose a novel piece-wise planar approximation to vehicle surfaces and use it for robust camera trajectory estimation. The object presents itself as a plane to the moving camera. By segmenting the car into its constituent planes by RANSAC with Homography as the model we obtain superior reconstruction of the moving object.
2) We extend the single-view deformable wireframe model fitting [30], [31] (inference) to multiple views, which stabilizes the estimation of object location and

[1] International Institute of Information Technology, Hyderabad, India
[2] Imperial College London, UK

[1]We use SLAM and SfM interchangeably, since the most successful methods for both problems are essentially based on pose graph optimization.

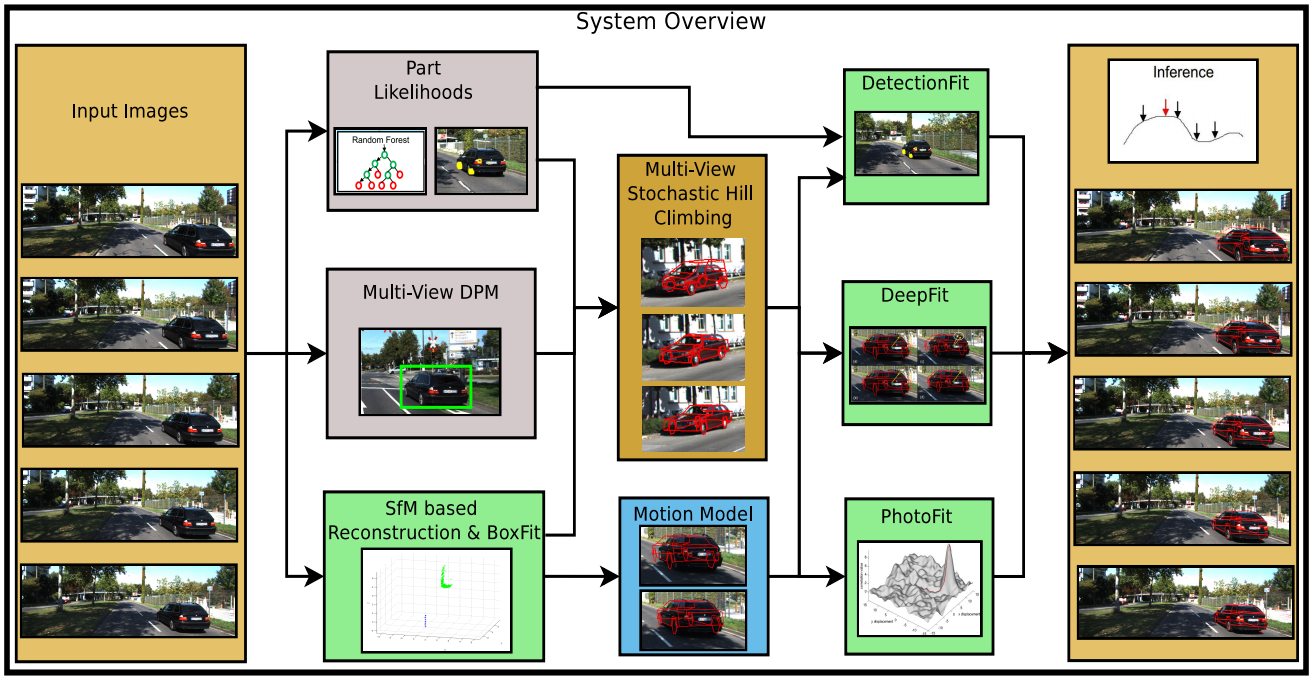

Fig. 1: Our full pipeline. Input image sequence is the input to the pipeline. Part likelihoods are obtained from a Random forest classifier as per [30]. Cars are detected using multi-view DPM which is a bank of DPM detectors(discrete viewpoint estimate and 2D bounding box per frame) and reconstruction is performed using a novel SfM pipeline. All this information is exploited in Multi-View Stochastic Hill Climbing algorithm to optimize for the shape of the car as described in Algorithm 1. Once the shape is optimized, using this shape we optimize for pose, for which, we perturb the pose of 3D deformable wireframe model of car into the next image using motion model. We use different terms like `detectionfit`, `deepfit`, `photofit` to optimize for pose as described in III(C).

shape.

3) We experimentally demonstrate improvements in 3D shape estimation and localization on several sequences in KITTI dataset [13] resulting from the the tight integration between SfM cues and object shape modeling.

II. Related Work

The lack of robustness is the biggest problem facing robotic perception today, with high-level semantic recognition and geometry estimation being the most important components of any sophisticated perception system. We first note that geometry estimation pipelines [19], [8] have matured over the recent years, so much so, that there are workshops organized at prominent robotics conferences (e.g. RSS 2015) to discuss whether SLAM is solved. On the other hand, the last decade observed a plethora of work in the areas of visual recognition aided by several advances at the level of formulating novel features [7] along with more pronounced and efficient use of classifiers [10], inference methods [2], and more recently end-to-end methods [14]. Unfortunately, while these advances have steadily improved performance in various computer vision tasks, they are still far from being robust enough for robotic applications. The use of geometry to reason about scenes while simultaneously performing recognition in computer vision can be traced to the works of [6], [28], [20] in recent times. These works consistently demonstrate the superior performance of systems that combine geometric reasoning with coarse-grained semantic recognition, as compared to isolated recognition approaches.

Inspired by these former works, more recent approaches have investigated fine-grained semantic modeling to output not just bounding boxes but 2D parts, 3D shapes and poses such as in [29], [17], [30], [31], [15], [26]. Unfortunately, while these methods yield further improvements in performance, they largely focus on the challenging problem of single-image scene understanding, whereas in robotics multiple views of a scene are often available. With this in mind, some recent works have approached the problem from a more practical, robotics perspective [11], [9], [23], [12], [24], [21]. While [11] extracts planar regions in SfM point cloud, which are fairly restrictive; whereas [23], [21] combine SfM and multi-view object recognition. Unfortunately, [23] is restricted to a handful of particular object instances (five types of chairs), because one 3D CAD model cannot represent the visual appearance of an entire object category whereas [21] supports object class recognition but does not allow recognition to feed back into improving geometry estimates and has only coarse bounding box level object representations. In comparison, we incorporate a finer-grained deformable geometric model (similar to [30]) that can represent entire object classes, such as the *car* class - and maintain a closed-loop collaboration between SfM and recognition.

Starting with the seminal work of [9], which coupled coarse-grained object modeling and tracking with SfM and ground plane estimation for road scene understanding, more recent works [12], [24] have attempted a stronger coupling between the semantic and geometric components of the system. The work of [27] termed Deep-Matching and DeepFlow

extracts feature information and is aggregated from fine to coarse using sparse convolutions and max-pooling.

This paper while retaining philosophical similarities to [9], [12], [24] contrasts with them by not just optimizing and localizing 3D bounding boxes but 3D shapes and parts thereby advancing the state of the art results showcased in [30], [31]. Thus it allows precise estimation of object shape and pose, using a deformable 3D wireframe model for the vehicles, which opens up the possibility for the perception module giving a superior input to the planning and control modules in an autonomous vehicle (which we do not cover) - since it can allow precise prediction of the future trajectory and fine-grained interactions of the vehicles in sight.

III. Our Approach

Our approach to detect the shape of a object, and track its 3D pose over several frames is formulated as a joint minimization over shape and pose space defined over multiple frames. We get camera pose estimates w.r.t. object(car) using proposed plane segmentation for initial 3-5 frames and utilize it to optimize the shape of deformable wireframe model of car. After we converge on a local minimum for shape parameters, we apply our motion model to the deformable wireframe and optimize for pose. In terms of cues, we utilize part detection likelihoods from a multi-class Random Forest [30], sparse 3D reconstruction estimates, as well as deep matches [27]. In addition, we employ photometric constraints to guide the minimization when the object is too small for either reconstruction or deep matching to reliably work.

A. Notation

For each image $I(t)$ in a video sequence, let x_t^i be a deep matched feature in that frame, corresponding to the i^{th} track. Thus, $\chi^i = \{x^i_{s(i)}, \ldots, x^i_t, \ldots, x^i_{e(i)}\}$ represents the feature track with index i, starting from frame $s(i)$ and ending at frame $e(i)$. Similarly, let X_t^i represent the 3D point with index i, with the subscript t indicating that the 3D point is *visible* in frame t. As we will explain later, X_t^i is the output of our SfM based reconstruction pipeline. Also, let $L_k(t)$ represent the detector confidences for the k^{th} object part for image $I(t)$. Further we represent the (perspective) projection function as $\mathcal{P}$,

$$\mathcal{P}(S_j(\alpha), P(t)) = K[R(t)\ T(t)]S_j(\alpha) \tag{1}$$

where K is the intrinsic matrix of the camera, while $S_j(\alpha)$ represents the j^{th} 3D coordinate on the object wireframe.

B. Deformable Wireframe Object Model

We utilize a deformable wireframe object class model [5], [30] to represent object instances (vehicles). The model is learnt (offline, once) on 3D CAD data manually annotated with pre-defined landmarks (also called object "parts"). It is based on a dimensionality reduction algorithm called Principal Components Analysis (PCA), with an object shape represented by the sum of a mean wireframe μ plus the m principal component directions p_j and corresponding standard deviations s_j , where $1 \leq j \leq m$. Any 3D wireframe X following a similar geometric topology as the object class, can thus be represented up to some residual ϵ, as a linear combination of r principal components with geometry parameters α, where α_k is the weight of the k^{th} principal component.

Specifically, let $S(\alpha)$ represent the shape and $P(t) = \begin{bmatrix} R(t) & T(t) \end{bmatrix}$ the pose at time t of the object, where R and T represent rotation and translation respectively. Here, $S(\alpha)$ is the list of 3D points that represent that locations of landmarks (parts) of the deformable wireframe model, parameterized by shape parameter vector α. Thus,

$$S(\alpha) = \mu + \sum_{i=1}^{r} \alpha_k s_k p_k + \epsilon \tag{2}$$

In practice, object pose is represented by three translation parameters (t_x, t_y, t_z) and two rotation parameters $(\theta_{az}, \theta_{el})$, where az and el represent azimuth and elevation respectively. The camera/object relation is assumed to be such that the in-plane rotation is fixed and does not have to be modeled.

In order to match this geometric model to real-world images, [30] compute synthetic renderings to generate training data, encoded as shape context descriptors, and train a multiclass Random Forest classifier to detect and score the object parts which in turn goes into the objective function described by equation 7.

C. Multi-view, Multi-cue Objective Function

As mentioned earlier, our objective function is defined as a joint minimization over shape and pose space, defined over multiple frames and having four different terms that optimize different aspects of shape and pose. In this section, we describe each individual term, and then combine them to arrive at our objective function. Several of the terms occurring in this section is further explained in detail with figures in the supplementary material [4].

Minimal volume term: `boxfit` This function tries to fit the reconstructed 3D points on the vehicle in such a way that it encapsulates, in a *minimum cuboidal volume*, all the 3D points X_t in *each* frame. Formally, we define the `boxfit` function as follows:

$$B(\pi(t), X_t) = \sum_i d_\perp(\pi(t), X_t^i) \tag{3}$$

where $d_\perp$ is a function that calculates the perpendicular distance of the i^{th} 3D point from the respective planes $\pi(t)$ of the object in the t^{th} frame. The overall purpose is to minimize the distance of the reconstruted 3D points from thier respective planes to fit a cuboidal volume.

Part likelihood term: `detectionfit` This function measures how well the projection of the current estimate of object and pose parameters explain the part detection likelihoods as obtained from the multi-class Random Forest [30]. It is specified as

$$F(S(\alpha), P(t), L_k(t)) = -\frac{1}{\sum_{i=1}^{m} o_i(S(\alpha))} o_j(S(\alpha)) log(\frac{L_k(\mathcal{P}(S(\alpha), P(t)))}{L_b(\mathcal{P}(S(\alpha), P(t)))}) \tag{4}$$

where L_b represents the background likelihood in the given image. The above formulation is a direct derivative

of Zia *et al.* [30], with the only difference being that the occlusion function $o(\cdot)$ only includes self-occlusion in our case. A sample part likelihood (L_k) is shown in Figure 2.

Deep match term: `deepfit` As we will describe later, dynamic objects like vehicles in videos are not suitable for obtaining accurate feature tracks over long time. Thus, traditional reconstruction methods like bundle adjustment (BA), and even state-of-the-art methods like ORB SLAM fail to work in such scenes (Table II). To circumvent this, we use deep match [27] correspondences, which are accurate over short distance, along with an optimization term that tries to preserve the *relative* location of the projection of the vehicle wireframe, w.r.t. tracked features.

We measure the *shearing* in 2D correspondences produced using deep match [27], when the object undergoes motion along a video, as described in equation 7. When the object is close to the camera, such a function might only approximate the motion of these correspondences in space-time. However, in practice we found it a good approach when the object is either moderately sized (few meters away from the camera), or distant.

$$D\Big(S_j(\alpha), P(t), P(t+1), \chi^i\Big) = \\ \|\|\mathcal{P}(S_j(\alpha), P(t)) - x_t^i| - |\mathcal{P}(S_j(\alpha), P(t+1)) - x_{t+1}^i|\|^2 \tag{5}$$

Note that the above function only measures the magnitude of deviation of a feature track from the projection of a point on the wireframe model. Also note that penalizing the magnitude is sufficient, since any deviation in feature points is captured by increasing/decreasing magnitude w.r.t at least one wireframe corner.

Photometric term: `photofit` When the object is observed to be far away from the camera, even deep matches are not abundant. In such cases, this term finds itself becoming useful. Because of the size of the object, obtaining correspondences becomes a tough job, and this reconstructing them to produce 3D points X_t^i, becomes improbable. Instead we leverage on the fact that distant objects more or less undergo affine transformations of their textured surfaces, and hence the immediate texture surrounding the corners of the wireframe model's projection tend to remain intact.

$$\Phi\Big(t, S(\alpha), P(t), P(t+1)\Big) = \\ \|I(t, \mathcal{B}(\mathcal{P}(S(\alpha), P(t)))) - I(t+1, \mathcal{B}(\mathcal{P}(S(\alpha), P(t+1))))\| \tag{6}$$

where $\mathcal{B}(\cdot)$ denotes the immediate neighborhood in image space. We can now formulate our objective function, factored into the above four terms: *boxfit*, *detectionfit*, *deepfit*, and *photofit* as,

$$\underset{S(\alpha),P(t)}{\operatorname{argmin}} \underbrace{\sum_i B\Big(S(\alpha), P(t), X_t^i\Big)}_{boxfit} + \underbrace{\sum_k F\Big(S(\alpha), P(t), L_k(t)\Big)}_{detectionfit} \\ + \underbrace{\sum_j \sum_i D\Big(S_j(\alpha), P(t), P(t+1), \chi^i\Big)}_{deepfit} + \underbrace{\Phi\Big(t, S(\alpha), P(t), P(t+1)\Big)}_{photofit} \tag{7}$$

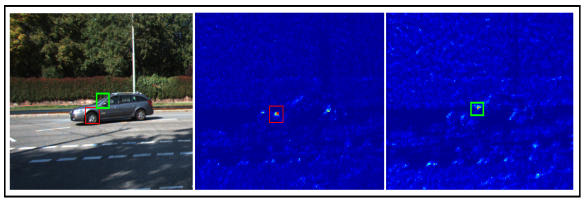

Fig. 2: Output of the Random Forest (RF) part detector on an image, for two example parts. Parts are not large enough to have high discriminative power on their own, thus the global wireframe model acts as a strong regularizer.

D. Optimizing the objective function

Our objective function, described in equation 7, is highly non-linear and as such cannot be minimized easily. One way to approach this problem, is to minimize shape and pose separately, in an iterative EM-like procedure. However, we see empirically, that having inaccurate estimates of pose leads to wrong shape fitting, since the shape of the vehicle changes to accommodate detection and other evidences, in the absence of accurate pose. We also note, that once shape is recovered with reasonable accuracy, pose estimation can be done fairly independently.

Keeping both these aspects in mind, we split the minimization process for equation 7 into two parts. In the first part, we compute an initial estimate of the *relative pose* between a few initial frames, which is a side product of our SfM pipeline described in section IV. Thus our pose estimation problem just reduces to finding an estimate of just *one* transformation between the coordinate system of the deformable wireframe, and that of the SfM based reconstruction. We then resort to a stochastic hill climbing based approach, similar to Zia *et al.* [30], to fit *both* shape and coordinate transformation (also represented as pose) to multi-view data. The difference with Zia *et al.* [30] is that we fit one set of pose parameters to detector and other evidence from *multiple views*. Specifically, we use the `boxfit` and the `detectionfit` functions for this purpose. Our multi-view deformable wireframe based stochastic hill climbing approach is described in Algorithm 1 with more details in supplementary material [4].

In the second part, we fix the shape parameters optimized in the first part, and only optimize over pose using the same stochastic hill climbing approach described earlier. This is primarily because optimization algorithms like BA work best in pose estimation scenarios only when feature correspondences are dense and tractable over several frames. In our case, as we will show later, even state-of-the-art algorithms like LSD and ORB SLAM fail to initialize and track interest points on objects (and dynamic objects in general). Thus we resort to particle based approaches, that are more capable of handling both inaccuracies in correspondences and variability in their strength over time.

E. Motion model

With the optimization approach described above, there are around 6 pose parameters to be estimated per frame, along with shape parameters for each object. While relative pose computation might reduce the search space in stochastic hill

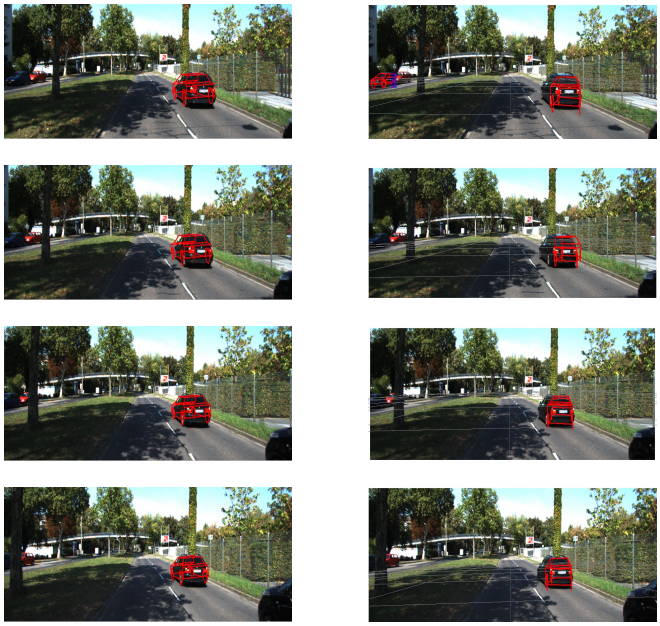

Fig. 3: Visual comparison of our results (left column) and [31] (right column) for a sequence. Notice how our multi-view model produces better 2D fits. The misalignment in 2D for [31] implies a large localization error in 3D and the pose.

climbing, lack of good long term correspondences ensure that relative pose estimation in SfM eventually "drifts" away from the ground truth. Thus, it is useful to enforce a motion model to further restrict our search space, and not overly depend on our reconstruction capabilities. In the absence of any specific information about vehicle movement, we use a generic motion model that defines the current pose based on the two previously observed poses as

$$P(t) = P(t-1) + \underbrace{(P(t-1) - P(t-2))}_{Previous\ Motion} + \mathcal{N}(0, \Sigma)$$

where $\mathcal{N}(0, \Sigma)$ represents a zero mean Gaussian with variance Σ.

IV. Shape Reconstruction as Initialization

In this section, we describe our SfM based procedure to reconstruct a few 3D points on the surface of the object, represented by the variable X_t in equation 7. We first describe our plane-based modeling of the shape of the object, which in our experience, leads to a very robust pose estimation algorithm. We follow this up with our BA based formulation for global optimization of 3D points and relative pose.

A. Vehicle Reconstruction Modeling

We leverage a piece-wise planar model for the vehicle, which allow utilizing homographies to represent each side of the vehicle, in turn robustifying the multi-body SfM estimation. This together with coarse bounding box level object detections [10] feeds into fitting a deformable wireframe object model [30] to multiple views of the scene. This pipeline also outputs auxiliary *relative pose* information, which is used to reduce the number of pose parameters fitted to the data. Since this reconstruction is in different scale as compared to deformable wireframe, this *relative pose* has to be scaled.

B. Vehicle Reconstruction

1) Plane Segmentation: We model an image of a vehicle as a combination of two planar regions. In Fig. 3, for example, the side and the back of the car is visible. Given deep matches [27], we randomly sample two feature points. The line joining these two points acts as a prior in the segmentation of the car region into two planes. A homography matrix [16] is fit to each set of features from the above two planes and the inliers are computed. We iteratively sample the planes and move towards the region that gives the maximum number of inliers for the set of tracked points on the car.

2) Sparse Reconstruction and Camera Localization: The planes detected from the above step help obtain a good initialization of the vehicle motion. The detected planes are tracked to consecutive planes using the optical flow based tracking. We compute the homography matrices for each of the planes H_1 and H_2, and decompose both the homography matrices to compute the Rotation and Translation(R and T). These decompositions provide a total of 16 possible combinations. We discard 12 of the candidates using standard methods [16], and exploit the perpendicularity constraint of the car planes for the selection of the correct R and T from the remaining 4 candidates. This provides us with the correct combination of R, T, which contain normals perpendicular to each other. The reconstruction of the car is computed from the above planes using triangulation of the tracked points. We denote the R_{ij} and T_{ij} as the rotation and translation matrix from the i frame to the j frame. We compute the 3D points on the car using the triangulation of $R_{a(a+1)}$ and $T_{a(a+1)}$. To solve for the scale problem, we compute the extrinsic matrix $R_{a(a+2)}$ and $T_{a(a+2)}$ using resectioning of the earlier triangulated points. The extrinsic matrix $R_{(a+1)(a+2)}$, $T_{(a+1)(a+2)}$ is obtained by matrix transformations and a is updated to $a+1$. Here a is an observation in the SfM pipeline.

The initialization of the car motion and reconstruction is optimized using bundle adjustment. We solve the reprojection error of each 3D reconstructed point. The objective function of the reprojection error is minimized using Levenberg-Marquardt algorithm from Ceres solver [1].

V. Experimental Evaluation

In this section, we do a thorough qualitative and quantitative evaluation of each block from our method on the challenging real-world KITTI Tracking Dataset [13]. Compared to the other publicly available outdoor datasets, KITTI provides ground truth 3D location of each individual objects providing for a good comparison platform. We evaluate the algorithm on portions of 6 sequences (02,03,07,09,11,15) of the KITTI tracking dataset. Each of the sequence contains cars with multiple motion and occlusions, making it a challenging experimental setup. We have consciously chosen the sequences to make sure that they posses diverse attributes. i.e. depth range of the target cars varies from 4 meter to 25 meter. Also these sequences are chosen such that they cover different scenarios like traffic points, cars with dense trees

Algorithm-1: Multi-view Stochastic Hill-Climbing

procedure FIT MODEL(H,σ_1, σ_2,I)
  for each iteration $l \in L$ **do** ▷ L = 20
    for each particle $i \in N$ **do** ▷ N = 250
      $h_i \in \mathcal{N}(H, \sigma_1)$ ▷ H is the mean model
      $prevscore = -10000$
      for each candidate $j \in R$ **do** ▷ R = 400
        $h_i^j \in \mathcal{N}(h_i, \sigma_2)$
        for each image $k \in I$ **do**
          $\mathcal{P}(S_{h_i^j}(\alpha), P(t)) = K[R(t)\ T(t)]S_{h_i^j}(\alpha)$
          for each visible part $p \in Parts$ **do**
            $score = score + R_k(S_{h_i^j}(\alpha), P(t), R_p(t))$ ▷ part likelihood
        if $score > prevscore$ **then**
          $\overline{h_i} = h_i^j, prevscore = score$
      $h_i = \overline{h_i}, M_i = prevscore$
    $h_{best} = h_{\arg\max_i(M_i)}$
  return h_{best}

	F decomposition			F, H decomposition		
	RMSE	Mean	Med	RMSE	Mean	Med
S1	**0.46**	**0.41**	**0.41**	0.54	0.47	0.54
S2	2.36	2.03	1.96	**0.54**	**0.48**	**0.48**
S3	1.7	1.5	1.5	**1.03**	**0.90**	**1.05**
S4	0.98	1.04	1.16	**0.71**	**0.62**	**0.73**
S5	12.61	8.67	6.12	**0.50**	**0.44**	**0.46**
S6	1.07	0.85	0.85	**0.11**	**0.1**	**0.1**

TABLE I: Results comparing different Intialziation methods for our SLAM system. We Intialize the SLAM with F Decomposition based method and our novel F,H Decomposition method and show an improvement in overall SLAM pipeline.

in background, cars moving in shadows and light, cars that are moving and stationary, cars of different colors.

We perform two types of evaluation: (1) Sparse reconstruction and camera localization which corresponds to BOXFIT. (2) Object/Car localization. We have compared our results with corresponding state of the art systems like [8], [19], [31].

A. Implementation Details

In this section, we outline critical implementation details. Our particle generation strategy is along similar lines of Zia *et al.* [30], with only our evaluation function being different.

We obtain predicted object bounding boxes for a few initial frames by repeatedly applying the deformable part model based vehicle detector [10]. We generate an initial set of 250 particles randomly sampled from a uniform distribution for the unknown shape parameters, whereas the parameters for pose are based on the initialization from a collection of viewpoint-dependent part configurations. We only choose those locations which project back to the vehicle bounding box. Approximate depth estimate of the vehicle is also computed using the bounding box height and real-world average vehicle height. For each particle, we generate 400 candidates by sampling from a Gaussian distribution with mean at the particle value. Likelihood is computed for each candidate and the one with the highest likelihood is set as the new particle. This process is repeated for 20 iterations.

For calculating the part likelihood, we use a viewpoint-invariant classifier, meaning that one class label includes views of a part over all poses in which the part is visible [30]. This marginalization over viewpoints speeds up the part detection. Additionally, the classifier also has a background class, which will be used for normalizing eq. 4. We train a single random forest classifier for each object class (here only vehicles), distinguishing between the parts of interest (36 for vehicles) and background.

The particle with the highest likelihood across all images is selected as the final result. We observe that the location of the particle is within 2m of the ground truth for most of the cases within 6-7 iterations, but the shape and pose needs 8-10 iterations for optimization.

B. Sparse reconstruction and camera localization

Monocular SLAM systems [8], [19] are robust only for static scenes with plenty of texture. But due to significantly small size of cars and their motion with respect to camera, considerable deficiency of feature tracks exists even for a small motion, such that these systems fail to provide tangible initial estimates of camera relative motion in a way useful for BA based optimization. Thus many state-of-the-art VSLAM pipelines[8], [19] are unable to handle moving objects. To showcase the superior performance of our method, we compare the trajectory of cars with ground truth using Absolute Trajectory Error (ATE) as proposed in [25], [22]. ATE directly measures the difference between points of the ground-truth and the estimated trajectory.

Table I compares initialization of different methods for obtaining the camera trajectory. We compare trajectory estimates with routine F matrix decomposition vis-a-vis the current approach based on a combination of F and H decompositions. As it can be observed from the table our method performs better than the baseline approaches that rely on F matrix decomposition. Table II demonstrates the comparison of the proposed method with LSD-SLAM [8] and ORB-SLAM [19]. As it can be observed from the table that due to lack of enough number of unique features ORB-SLAM fails to initialize on any of the sequences we tested on. This non initialization of ORB SLAM is reported as NI in Table II. We show a better performance compared to LSD-SLAM due to our proposed initialization method.

After the initial 7-8 frames, we typically stop our reconstruction pipeline based on Homography combined with BA, and only use our multi-view stochastic hill climbing method to generate pose estimates thereafter. These results that extend to 30 frames and more as obtained through Hill Climbing are portrayed in Tables III and IV wherein instead of ATE we use object pose localization (location + orientation) to characterize the performance. Comparison with standard SLAM pipelines such as LSD SLAM cannot be performed over longer sequences, for such systems break down after initial frames.

	LSD SLAM			ORB-SLAM	Ours		
	RMSE	Mean	Med		RMSE	Mean	Med
S1	0.80	0.69	0.79	NI	**0.54**	**0.47**	**0.54**
S2	1.15	0.99	0.96	NI	**0.54**	**0.48**	**0.48**
S3	1.17	1.47	1.71	NI	**1.03**	**0.90**	**1.05**
S4	1.21	1.04	1.20	NI	**0.71**	**0.62**	**0.73**
S5	1.27	1.09	1.23	NI	**0.50**	**0.44**	**0.46**
S6	0.66	0.58	0.58	NI	**0.11**	**0.1**	**0.1**

TABLE II: Results comparing our reconstruction approach to LSD and ORB SLAM approaches on 6 sequences. Note that when both the vehicle and camera move, ORB SLAM completely fails to initialize, while LSD Slam gives inferior results in almost all cases. For each type of metric based on trajectory error, we highlight the best result in each row.

Method	<1 m(%)	<1.5 m(%)	<2 m(%)
[31]	55.2	76.24	89.38
OURS	**70.44**	**95.08**	**98.36**

TABLE III: Results comparing our object localization estimation w.r.t that of Zia *et al.* [31]. We show a clear improvement in the localization compared to the other state-of-the-art algorithm.

C. Vehicle Localization and Pose Detection

While in the previous section, we presented results for camera localization and sparse reconstruction, in this section we focus on the localization and pose estimation of the vehicle w.r.t the camea. We do 3D localization comparison of each detected vehicle with respect to the ground-truth. We measure the 3D localization by the fraction of detected object centroids that are correctly localized up to deviations of 1, 1.5 and 2 meter. The pose accuracy evaluation for each individual vehicle is computed by measuring the percentage of vehicles localized with pose error less than 5^o and 10^o. We have compared the localization accuracy with respect to the 3D localization of Zia et al. [31]. The comparative study of the 3D object localization is depicted in Table III and that of orientation is depicted in Table IV. We show a significant improvement in localization and orientation estimation of vehicles, when compared to [30]. A noteable and expected observation here is that, accuracy of localization within 1 meter is better when the vehicle is within range of 15 meter from camera. Nonetheless, even till 25 meters of depth, our localization within 2 meter is as high as 98.36%. Also, there is remarkable improvement in pose detection shown in Table IV. This might lead to better input for planning and control as pose of the vehicle provides crucial information about direction of motion. the efficacy of the algorithm.

The Need for `photofit`: In this subsection we illustrate the need for our `photofit` term in equation 7. Figure 4 shows the number of deep matches obtained on average in the KITTI [13] dataset as a function of the height of the vehicle in pixels. Since we model vehicles using multiple homographies for reconstruction purposes, small vehicles are difficult to reconstruct accurately. Even the `deepfit` term allows for large variation of pose in such cases. Hence we resort to the `photofit` term for accurate localization and pose estimation.

Method	<5 deg (%)	<10 deg (%)	Average VP error(deg)	Median VP error(deg)
[31]	51.26	65.72	19.34	5.5
OURS	**88.86**	**96.73**	**2.33**	**1.87**

TABLE IV: Results comparing our object orientation with respect to the object orientation of the Zia *et al.* [31].We show a clear improvement in the pose compared to the other state-of-the-art algorithm.

Fig. 4: Plot of the number of deep matches found per vehicle, as a function of the height of the vehicle, averaged over several vehicles in the KITTI [13] dataset. Notice how small vehicles are tracked sparsely.

The Role of `deepfit`: In this subsection we portray the advantages of using RF based part model projections to enhance object localization. Figure 5(a) depicts a situation where particles are sampled without considering the `detectionfit` while computing their weights. The 3D pose of the vehicle and consequently its projection onto the image are erroneous as the parts have drifted from their actual locations. However by incorporating the `detectionfit` term along with other cost terms tangibly improves the localization and pose of the vehicle in 3D and consequently in the image as well as shown in Figure 5(b). The projected vehicle parts overlap the true parts in 5b. Figure 5(c) shows over 26 frames that `detectionfit` prevents 3D model from drifting away.

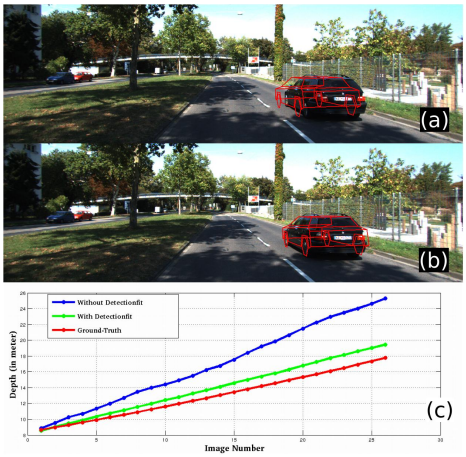

Fig. 5: Comparison for with and without detectionfit. (a)Selected particle from deepfit alone(without detectionfit). (b)Selected particle from deepfit + detectionfit which fits the model better. (c) Graphical representation demonstrates that detectionfit prevents drifting of 3D model

TABLE V: Visual results comparing Zia et al. [31](Middle row) with multi-view(ours)(third row) fitting for 4 input sequences (First Row) of the KITTI sequence. The multi-view deformable object model provides better shape estimates of the object model in most of the scenarios.

VI. Discussion and Conclusion

We have approached the problem of multi-view object detection from a novel SfM based deformable wireframe alignment perspective. We have proposed a unique object reconstruction pipeline, which outperforms state-of-the-art algorithms. Through the proposed method we show significant improvement over current state-of-the-art object localization methods by almost 15 %. We also show qualitatively superior object shape estimation, when projected onto the images. Moreover we have proposed plane segmentation based initialization of camera poses that outputs superior trajectories relative to moving cars when compared with current monocular SLAM pipelines. More information including supplementary material can be found here [4]. We plan to work on making the pipeline work in realtime for our future work.

References

[1] S. Agarwal, K. Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[2] M. Blaschko and C. Lampert. Learning to localize objects with structured output regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

[3] R. A. Brooks. Symbolic reasoning among 3-d models and 2-d images. *Artificial Intelligence*, 1981.

[4] F. Chhaya et al. Supplementary material. `http://robotics.iiit.ac.in/people/falak.chhaya/Monocular_Reconstruction_of_Vehicles.html`

[5] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active Shape Models - Their Training and Application. *Computer Vision and Image Understanding (CVIU)*, 61(1):38–59, 1995.

[6] A. E. D. Hoiem and M. Hebert. Putting objects in perspective. In *IJCV*, 2008.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[8] J. Engel, T. Schops, and D. Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *European Conf on Computer Vision (ECCV)*, 2014.

[9] A. Ess, B. Leibe, K. Schindler, and L. V. Gool. Moving obstacle detection in highly dynamic scenes. In *ICRA*, 2009.

[10] P. F. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 2009.

[11] A. Gee, D. Chekhlov, A. Calway, and W. Mayol-Cuevas. Discovering Higher Level Structure in Visual SLAM. *IEEE Transactions on Robotics*, 24(5):980–990, 2008.

[12] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3d traffic scene understanding from movable platforms. *TPAMI*, 2014.

[13] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[15] F. Güney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *CVPR*, 2015.

[16] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[17] M. Hejrati and D. Ramanan. Analyzing 3d objects in cluttered images. In *NIPS*, 2012.

[18] A. Kundu, K. M. Krishna, and C. V. Jawahar. Realtime multibody visual SLAM with a smoothly moving monocular camera. In *ICCV*, 2011.

[19] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. ORB-SLAM: a Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 2015.

[20] D. H. P. K. N. Silberman and R. Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*, 2012.

[21] S. Pillai and J. Leonard. Monocular SLAM Supported Object Recognition. In *RSS*, 2015.

[22] N. D. Reddy, P. Singhal, V. Chari, and K. M. Krishna. Dynamic Body VSLAM with Semantic Constraints. In *IROS*, 2015.

[23] R. Salas-Moreno, R. Newcombe, H. Strasdat, P. Kelly, and A. Davison. SLAM++: Simultaneous localization and mapping at the level of objects. In *CVPR*, 2013.

[24] S. Song and M. Chndraker. Joint SFM and Detection Cues for Monocular 3D Localization in Road Scenes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[25] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[26] S. Wang, S. Fidler, and R. Urtasun. Holistic 3d scene understanding from a single geo-tagged image. In *CVPR*, 2015.

[27] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *IEEE International Conference on Computer Vision (ICCV)*, Sydney, Australia, Dec. 2013.

[28] T. H. X. Wang and S. Yan. An HOG-LBP human detector with partial occlusion handling. In *ICCV*, 2009.

[29] S. S. Y. Xiang. Estimating the aspect layout of object categories. In *CVPR*, 2012.

[30] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler. Detailed 3d representations for object modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2608–2623, 2013.

[31] M. Z. Zia, M. Stark, and K. Schindler. Towards scene understanding with detailed 3d object representations. *IJCV*, 112(2):188–203, 2015.