Realtime Multibody Visual SLAM with a Smoothly Moving Monocular Camera

Abhijit Kundu, K Madhava Krishna and C. V. Jawahar
International Institute of Information Technology, Hyderabad, India
abhijit.kundu@gatech.edu, {mkrishna,jawahar}@iiit.ac.in

Abstract

This paper presents a realtime, incremental multibody visual SLAM system that allows choosing between full 3D reconstruction or simply tracking of the moving objects. Motion reconstruction of dynamic points or objects from a monocular camera is considered very hard due to well known problems of observability. We attempt to solve the problem with a Bearing only Tracking (BOT) and by integrating multiple cues to avoid observability issues. The BOT is accomplished through a particle filter, and by integrating multiple cues from the reconstruction pipeline. With the help of these cues, many real world scenarios which are considered unobservable with a monocular camera is solved to reasonable accuracy. This enables building of a unified dynamic 3D map of scenes involving multiple moving objects. Tracking and reconstruction is preceded by motion segmentation and detection which makes use of efficient geometric constraints to avoid difficult degenerate motions, where objects move in the epipolar plane. Results reported on multiple challenging real world image sequences verify the efficacy of the proposed framework.

1. Introduction

Vision based SLAM [4, 7, 9, 14, 23] and SfM systems [6] have been the subject of much research and are finding applications in many areas like robotics, augmented reality, city mapping. But almost all these approaches assume a static environment, containing only rigid, non-moving objects. Moving objects are treated the same way as outliers and filtered out using robust statistics like RANSAC. Though this may be a feasible solution in less dynamic environments, but it soon fails as the environment becomes more and more dynamic. Also accounting for both the static and moving objects provides richer information about the environment. A robust solution to the SLAM problem in dynamic environments will expand the potential for robotic applications, like in applications which are in close proximity to human beings and other robots. Robots will be able to work not only for people but also with people.

The last decade saw lot of developments in the "multibody" extension [18, 20, 24] to multi-view geometry. These methods are natural generalization of classical structure from motion theory [6] to the challenging case of dynamic scenes involving multiple rigid-body motions. Thus given a set of feature trajectories belonging to different independently moving bodies, multibody SfM estimates the number of moving objects in the scene, cluster the trajectories on basis of motion, and then estimate the model as in relative camera pose and 3D structure w.r.t. each body. However all of them have focused only on theoretical and mathematical aspects of the problem and have experimented on very short sequences, with either manually extracted or noise-free feature trajectories. Also the high computation cost, frequent non-convergence of the solutions and highly demanding assumptions; all have prevented them from being applied to real-world sequences. Only recently Ozden *et al*. [16] discussed some of the practical issues, that comes up in multibody SfM. In contrast, we propose a multibody visual SLAM system, which is a realtime, incremental adaptation of the multibody SfM. However the proposed framework still offers the flexibility of choosing the objects that needs to be reconstructed. Objects, not chosen for reconstruction are simply tracked. This is helpful, since certain applications may just need to know the presence of moving objects rather than its full 3D structure or there may not be enough computational resource for realtime reconstruction of all moving objects in the scene. The proposed system is a tightly coupled integration of various modules of feature tracking, motion segmentation, visual SLAM, and moving object tracking while exploring various feed-backs in between these modules. Fig. 1 illustrates system pipeline and outputs of each different modules.

Reconstructing 3D trajectory of a moving point from monocular camera is ill-posed: it is impossible, without making some assumptions about the way it moves. However object motions are not random, and can be parameterised by different motion models. Typical assumptions have been either that a point moves along a line or a conic or on a plane [1] or more recently as a linear combination of basis trajectories [17]. Target tracking from bearings-only

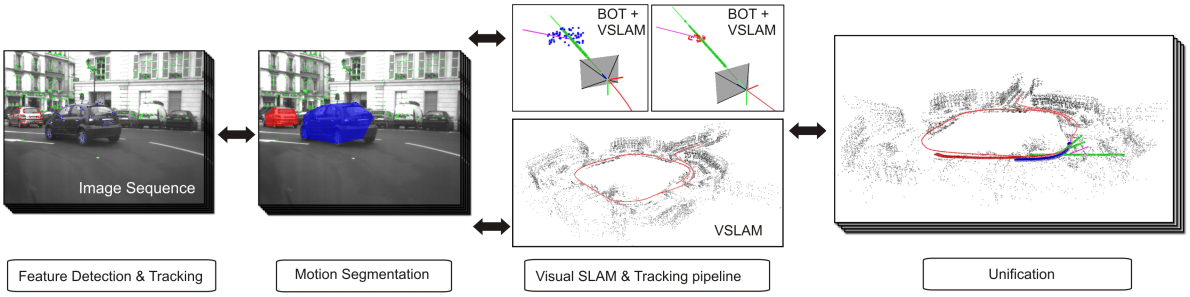

Figure 1. The input to our system is monocular image sequence. Various modules of feature tracking, motion segmentation, visual SLAM and Moving Object tracking are interleaved and running online. The final result is an integrated dynamic map of the scene including 3D structure and 3D trajectory of the camera, static world and moving objects.

sensors (which is also the case for a monocular camera) has also been studied extensively in "Bearings-only Tracking" (BOT) literature [2, 10] where statistical filters seems to be the method of choice. This same monocular observability problem gives rise to the so called "relative scale problem" [5, 16] in multibody SfM. In other words since each independently moving body has its 3D structure and camera motion estimated in its own scale, it results in a one-parameter family of possible, relative trajectories per moving object w.r.t. static world. This needs to resolved for a realistic, unified reconstruction of the static and moving parts together. Ozden *et al*. [5] exploited the increased coupling between camera and object translations that tends to appear at false scales and the resulting non-accidentalness of object trajectory. However their approach is mostly batch processing, wherein trajectory data over time is reconstructed for all possible scales, and the trajectory which for say is most planar is chosen by the virtue of it being unlikely to occur accidentally. Instead, we take a different approach by making use of a particle filter based bearing only tracker to estimate the correct scale and the associated uncertainty.

In realtime visual SLAM systems, moving objects have not yet been dealt properly. In [25], a 3D model based tracker runs parallel with the MonoSLAM [4] for tracking a previously modelled moving object. This prevents the visual SLAM framework from incorporating moving features lying on that moving object. But the proposed approach does not perform moving object detection; so moving features apart from those lying on the tracked moving object can still corrupt the SLAM estimation. Sola [22] does an observability analysis of detecting and tracking moving objects with monocular vision. To bypass the observability issues with mono-vision, he proposes a BiCamSLAM [22] solution with stereo cameras. A similar stereo solution has also been proposed recently by [12]. All these methods [12, 22, 25] have a common framework in which a single filtering based SLAM [4] on static parts is combined with moving object tracking (MOT), which is often termed as SLAMMOT [12]. Unlike SLAMMOT, we adopted multibody SfM kind approach where multiple moving objects are also fully reconstructed simultaneously, but our framework still allows simple tracking if full 3D structure estimation of moving object is not needed.

We propose a realtime incremental multibody visual SLAM algorithm. The final system integrates feature tracking, motion segmentation, visual SLAM and moving object tracking. We introduce several feedback paths among these modules, which enables them to mutually benefit each other. We describe motion cues coming from SfM estimate done on that moving object and several geometric cues imposing constraints on possible depth and velocities, made possible due to reconstruction of static world. Integration of multiple cues reduces immensely the space of possible trajectories and provides an online solution for the relative scale problem. This enables a unified representation of the scene containing 3D structure of static world, moving objects, 3D trajectory of the camera and moving objects along with associated uncertainty. As concluded in Sec.4.2 of [17] and also in BOT literature, dynamic reconstruction with mono-vision is good only when object and camera motion are non-correlated. To avoid this, existing methods resorted to spiral camera motions [12], multiple photographers [17] or uncorrelated camera-object motion [5]. We do not have any restrictive assumptions on the camera motion or environment. Instead, we extract more information from reconstruction pipeline in form of cues. We assume a calibrated monocular camera moving arbitrarily through a dynamic scene. Objects need to be rigid for reconstruction, otherwise they can simply be tracked. Estimation of ground-plane and the assumption of objects moving over it improves results but is not essential. We briefly discuss the motion segmentation framework in Sec. 2, followed by our visual SLAM framework (Sec. 3) and Moving Object Tracking (Sec. 4) using efficient Lie Group theory. We present various cues and the process of integrating them to tracking framework. We then describe the Unification (Sec. 5) module wherein we show how the BOT solves the relative scale problem, which is then used to build a unified representation of a dynamic scene. Results of the proposed system on multiple real image datasets are shown in Sec. 6.

2. Feature Tracking & Motion Segmentation

Feature tracking module tracks existing feature points, while new features are instantiated. The motion segmentation module segments these feature tracks belonging to different motion bodies and maintains it with time. We discuss them briefly (see [8] for in-depth discussion) as following.

2.1. Feature Tracking

In order to detect moving objects, we should be able to get feature tracks on the moving bodies also. Thus contrary to conventional SLAM, where features belonging moving objects are not important, we need to pay extra caution to feature tracking in this scenario. In each image, a number of salient features (FAST corners [19]) are detected while ensuring the features are sufficiently spread all over the image. A subset of these, detected in keyframes of the visual SLAM are made into 3D points. The extra set of tracks helps in detecting independent motion. In order to preserve feature tracks belonging to independent motions, we do not perform restrictive matching initially. Instead the feature matching is performed in two stages. In the 1st stage, features are matched over a large range so as to allow matches belonging to moving objects. A preliminary segmentation and motion estimate is made using this coarse matching. Finally when the camera motion estimate is available, we resort to guided matching which yields a larger number of features. In this stage, we make full use of the camera motion knowledge, while matching features.

2.2. Multibody Motion Segmentation

The input to motion segmentation framework are feature tracks, the camera relative motion in reference to each reconstructed body from the visual SLAM module, and the previous segmentation. The task of motion segmentation module is that of model selection so as to assign these feature tracks to one of the reconstructed bodies or some unmodelled independent motion. Efficient geometric constraints are used to form a probabilistic fitness score for each reconstructed object. With each new frame, existing features are tested for model-fitness and unexplained features belong to either unmodeled moving object, possibly new object or outliers. The geometric constraints are a combination of Epipolar constraint and Flow Vector Bound (FVB) [8]. The FVB constraint helps detecting difficult degenerate independent motions, where points move along the epipolar plane and thus making the epipolar constraint useless. This is important as they are very common in real world (*e.g.* motion of other moving cars as seen from a camera-mounted car). A recursive Bayes filter continuously updates the probability of a feature moving independently w.r.t. a body based on these geometric constraints. More details can be found in [8]. Computation of the 3D structure, in the form of depth bound (DB) detailed later in Sec. 4.2 helps in setting a tighter bound in FVB constraint, which results in more accurate independent motion detection.

3. Visual SLAM Framework

Visual SLAM or SfM estimates the camera pose denoted as g^t_{CW} and map points $X_W \in \mathbb{R}^3$ w.r.t. a certain world frame W, at a time instant t. The structure coordinates X_W are assumed to be constant *i.e.* static in this world frame evident from the absence of time t in its notation. In the multibody VSLAM scenario, the world frame W can be either the static world frame S or a rigid moving object O, chosen for reconstruction. The 4×4 matrix g_{CW} contains a rotation and a translation and transforms a map point from world coordinate frame to camera-centred frame C by the equation $X_C = g_{CW}X_W$. It belongs to the Lie group of Special Euclidean transformations, $SE(3)$. The tangent space of an element of $SE(3)$ is its corresponding Lie algebra $se(3)$, so any rigid transformation is minimally parameterised as a 6-vector in the tangent space of the identity element. We denote this minimal 6-vector as $\xi := (v^T\omega^T)^T \in \mathbb{R}^6$, where the first three elements is an axis-angle representation of rotation, while the later three represents the translation. The $\xi \in \mathbb{R}^6$ represents the twist coordinates for the twist matrix $\widehat{\xi} \in se(3)$. Thus a particular twist is a linear combination of the generators of the $SE(3)$ group, *i.e.*

$$\widehat{\xi} = \sum_{i=1}^{6} \xi_i G_i = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix} \mid \widehat{\omega} \in so(3), v \in \mathbb{R}^3 \qquad (1)$$

Here ξ_i are individual elements of ξ and G_i are the 4×4 generator matrices which forms the basis for the tangent space to SE(3). And $\widehat{\omega}$ is a skew-symmetric matrix obtained from the 3-vector ω. The exponential map $\exp : se(3) \rightarrow SE(3)$ maps a twist matrix to its corresponding transformation matrix in $SE(3)$ and can be computed efficiently in closed form. Changes in the camera pose g_{CW} is obtained by premultiplying with a 4×4 transformation matrix in SE(3). Thus the camera pose evolves with time as:

$$g^{t+1}_{CW} = \Delta g^t g^t_{CW} = \exp(\widehat{\xi}) g^t_{CW} \qquad (2)$$

The world points X_W are first transformed to camera frame and then projected in the image plane using a calibrated camera projection model $CamProj(.)$. This defines our measurement function $\hat{z}$ as:

$$\hat{z} = \begin{pmatrix} u \\ v \end{pmatrix} = CamProj(g_{CW}X_W) \qquad (3)$$

In each visual SLAM, the state vector $\hat{x}$ consists of a set of camera poses and reconstructed 3D world points. The optimization process iteratively refines this state vector $\hat{x}$ by minimizing a sum of square errors between current $\hat{x}$ estimate and observed data $\hat{z}$. The incremental updates

in optimization are calculated as in Eq. 2 at the tangent space around identity $se(3)$ and mapped back onto manifold. This enables minimal representation during optimization and avoids singularities. Also the Jacobians of the above equations needed in the optimization process can be readily obtained in closed form. Due to this advantages, the Lie theory based representation of rigid body motion is becoming popular among recent VSLAM solutions [9, 23]. We again use this Lie group formulation in tracking of the moving object described in section 4.

The monocular visual SLAM framework is that of a standard bundle adjustment visual SLAM [7, 13, 23]. A 5-point algorithm with RANSAC is used to estimate the initial epipolar geometry, and subsequent pose is determined by camera resection. Some of the frames are selected as keyframes, which are used to triangulate 3D points. The set of 3D points and the corresponding keyframes are used in by the bundle adjustment process to iteratively minimize reprojection error. The bundle adjustment is initially performed over the most recent keyframes, before attempting a global optimization. Our implementation closely follows to that of [7, 13]. While one thread performs tasks like camera pose estimation, keyframe decision and addition, another back-end thread optimizes this estimate by bundle adjustment. But there are couple of important differences with the existing SLAM methods, namely its interplay with the motion segmentation, bearing only object and feature tracking module, reconstruction of small moving objects. They are discussed next.

3.1. Feedback from Motion Segmentation

The motion segmentation prevents independent motion from entering the VSLAM computation, which could have otherwise resulted in incorrect initial SfM estimate and lead the bundle adjustment to converge to local minima. The feedback results in less number of outliers in the SfM process of a particular object. Thus the SfM estimate is better conditioned and less number of RANSAC iterations is needed. Apart from improvement in the camera motion estimate, the knowledge of the independent foreground objects coming from motion segmentation helps in the data association of the features, which is currently being occluded by that object. For the foreground independent motions, we form a convex-hull around the tracked points clustered as an independently moving entity. Existing 3D points lying inside this region is marked as not visible and is not searched for a match. This prevents 3D features from unnecessary deletion and re-initialization, just because it was occluded by an independent motion for some time.

3.2. Dealing Degenerate Configurations

In dynamic scenes, moving objects are often small compared to the field of view, and often appear planar or has very less perspective effects. Then both relative pose estimation and camera resection faces ambiguity and results in significant instability. During relative pose estimation from two views, coplanar world points can cause at most a two-fold ambiguity. So we use 5-point algorithm from 3 views to resolve this planar degeneracy, exactly as described in [15]. Though theoretically, calibrated camera resection from a coplanar set of points has a unique solution unlike its uncalibrated counterpart, it still suffers from ambiguity and instability as shown in [21]. So for seemingly small and planar objects we modified the EPnP code as in Sec. 3.4 of [11] to initialize the resection process, which is then refined by bundle adjustment.

4. Moving Object Tracking

A monocular camera is a projective sensor that only provides the bearing information of the scene. So moving object tracking with mono-vision is a bearings-only tracking (BOT) which aims to estimate the state of a moving target comprising of its 3D position and velocity. A single BOT filter is employed on each independently moving objects. At any time instant t, the camera only observes the bearing of tracked feature on the moving object. We consider the moving object state vector as $g^t_{OS} \in SE(3)$, representing 3D rigid body transformation of the moving object O in the static world frame S. Due to inherent non-linearity and observability issues, particle filter has been the preferred approach [2] for BOT. In this section we develop a formulation of the particle filter based BOT that integrates multiple cues from static world reconstruction.

4.1. Particle Filter based BOT

The uncertainty in pose of the object is represented by the poses of set of particles g_{iS} and their associated weights. Each particle's state denoted by $g^t_{iS} \in SE(3)$ represents its pose w.r.t. S at a time instant t. We continue with Lie group preliminaries discussed in Sec. 3. We assume an instantaneous constant velocity (CV) motion model, which is considered the best bet and most generic for modeling an unknown motion. Mean velocity between two intervals is represented by the mean twist matrix $\widetilde{\widehat{\xi}}^t_i = \frac{1}{\Delta t} ln(g^t_i (g^{t-1}_i)^{-1})$, where $\widetilde{\widehat{\xi}} \in se(3)$ is the mean twist matrix associated with the mean six dimensional velocity vector $\widetilde{\xi} \in \mathbb{R}^6$. The motion model of the particle then generates samples according to the pdf *i.e.* probability distribution function $p(g^{t+1}_{iS} | g^t_{iS}, \widehat{\xi}^t_i)$. Each component of the mean velocity vector has a gaussian error with a standard deviation σ_j, $j \in \{1, \dots 6\}$. To transform this gaussian distribution in $\mathbb{R}^6$ to $SE(3)$ space the following procedure is used. We define a vector $\alpha \in \mathbb{R}^6$, whose each component α_j is sampled from the Gaussian $\mathcal{N}(0, \sigma^2_j)$, then $\widehat{\alpha}$ is the twist matrix

associated with α_j. Then $\widehat{\xi}_i^t = e^{\widehat{\alpha}}\widetilde{\widehat{\xi}_i^t}$ generates samples in the twist matrix space of $\mathbb{R}^{4X4}$ corresponding to the gaussian errors centered at the mean velocity vector. Then the dynamic model of the particle generates samples that approximate the pdf given before as

$$g_{iS}^{t+1} = \exp(\widehat{\alpha})\exp(\widetilde{\widehat{\xi}_i^t}\Delta t)g_{iS}^t \tag{4}$$

The measurement model that predicts the location of a particle with $SE(3)$ pose g_i^{t+1} in the image as

$$\hat{z}_i^{t+1} = \begin{pmatrix} u_i^{t+1} \\ v_i^{t+1} \end{pmatrix} = CamProj(Trans(g_{CS}^{t+1} g_i^{t+1})) \tag{5}$$

Here $Trans(.)$ operator extracts the translation vector associated with the $SE(3)$ pose of the particle and $CamProj(.)$ is the camera projection Eq. 3. The weight w_i of the particle is updated as $w_i^{t+1} = \frac{1}{\sqrt{(2\pi)\eta}}\exp(\frac{(z-\hat{z}_i)'(z-\hat{z}_i)}{2\eta^2})$, where z is the actual image coordinate of the feature being tracked. The particles then undergo resampling in the usual particle filter way: particles with a higher weight have higher probability of getting resampled.

4.2. Ground Plane, Depth Bound & Size Bound

The structure estimation of the static world from visual SLAM module helps in reducing the possible bound in depth. Instead of setting the maximum depth to infinity, known depth of the background allows to limit the depth of a foreground moving object. The depth bound (DB) is adjusted on the basis of depth distribution of static world map points along the particular frustum of the ray. This bound gets updated as the camera moves around in the static world. The 3D point cloud of the static world is used to estimate the ground-plane (GP). Using the fact that most real world objects move over the ground plane, we can add constraints to the velocity vector such that its height above the ground plane is constant. Both the above cues ignored that we are able to track multiple features of the object. At wrong depths, this points may be reconstructed to lie below the ground-plane or too much above it. This criteria of size and unrealistic reconstructions is used to get an additional depth constraint. All these cues constraints the possible depth or velocity space. Integration of these depth and velocity constraints into the BOT filter is discussed in Sec. 4.4.

4.3. Initialization

Initialization is an important step for performance of particle filter in BOT. For a moving object which enters the scene for the first time, particles are initialized all along the ray starting from the camera through the image point which is the projection of a point on the dynamic object being considered. A uniform sampling is then used to initialize the particles at various depths inside that bound $[d_{min}, d_{max}]$ computed from the depth bound cue described previously in Sec. 4.2. The velocity components are initialized in a similar manner. At each depth, number of particles with various velocities are uniformly sampled so that the speeds lie inside a predetermined range $[s_{min}, s_{max}]$ along all possible directions. When a previously static object starts moving independently, we can do better initialization than uniform sampling: we initialize the depth as normal distribution $\mathcal{N}(\widehat{d}, \sigma^2)$, where $\widehat{d}$ is the depth estimate obtained from the point's reconstruction as part of the original body.

4.4. Integrating Depth and Velocity Constraints

Depth and velocity constraints play a very important role in improving the tracker performance, even in scenarios which are otherwise unobservable for a bearing only tracker. This reduces the space of state vectors to some constrained set of state vectors denoted as ψ. This can be implemented as the motion model, by sampling from a truncated density function p_s, defined as:

$$p_s = \begin{cases} p(g_{iS}^{t+1}|g_{iS}^t, \widehat{\xi}_i^t) & g_{iS}^{t+1} \in \psi \\ 0 & otherwise \end{cases} \tag{6}$$

Here, non-truncated pdf over motion model, $p(g_{iS}^{t+1}|g_{iS}^t, \widehat{\xi}_i^t)$ is evaluated from Eq. 4. To draw samples from this truncated distribution, we use rejection sampling over the distribution, until the condition $g_{iS} \in \psi$ is satisfied. This method of rejection sampling is sometimes inefficient. So in our implementation, we restrict the number of trials and if it still does not lie inside ψ, we flag those particles for lower weight in the measurement update step.

5. Unification

The output of the system can only be fully explored when we have a unified 3D dynamic map of the scene containing structure and trajectory of both static and moving objects. As discussed next, this requires solving some challenges but it also enables a new cue. The unified output can be used to generate dynamic 3D occupancy maps, which also takes into account the most likely space to be occupied by a moving object in the next instant.

5.1. Relative Scale Problem

From visual SLAM on rigid moving objects, we obtain camera pose $g_{CO}^t \in SE(3)$ and object points $X_O \in \mathbb{R}^3$ with respect to the object frame O. We also have the camera pose g_{CS} in the static world frame S. Thus configuration of the moving object O w.r.t. static world S can be obtained as $g_{OS} = g_{CO}^{-1} g_{CS}$. Expanding this equation in the homogenous representation we obtain:

$$\begin{bmatrix} \mathbf{R}_{OS} & \mathbf{t}_{OS} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{CS}^T & -\mathbf{R}_{CS}^T \mathbf{t}_{CS} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{CO} & \mathbf{t}_{CO} \\ 0 & 1 \end{bmatrix} \tag{7}$$

Equating the rotation and translation parts of Eq. 7, we obtain $\mathbf{R}_{OS} = \mathbf{R}_{CS}^T\mathbf{R}_{CO}$ & $\mathbf{t}_{OS} = \mathbf{R}_{CS}^T\mathbf{t}_{CO} - \mathbf{R}_{CS}^T\mathbf{t}_{CS}$. We can obtain the exact $\mathbf{R}_{OS}$, but from monocular SfM, we can only obtain $\mathbf{t}_{CO}$ and $\mathbf{t}_{CS}$ up to some unknown scales [6]. We can fix the scale for $\mathbf{t}_{CS}$, *i.e.* for the static background as 1, and denote the scale for $\mathbf{t}_{CO}$ as the unknown relative scale parameter s. Then the trajectory of the moving object is 1-parameter family of possible trajectories given by

$$\mathbf{t}_{OS} = s\mathbf{R}_{CS}^T\mathbf{t}_{CO} - \mathbf{R}_{CS}^T\mathbf{t}_{CS} \tag{8}$$

All of these trajectories satisfy the image observations, *i.e.* projection of the world points on the moving object are same for all the above trajectories. This is a direct consequence of the depth unobservability problem of monocular camera. Thus even after reconstructing a moving car, we are not able to say whether it is a toy car moving in front of the camera, or a standard car moving over the road. So we need to estimate this relative scale, and only when the estimated scale is close to the true scale, the reconstruction will be meaningful.

Unlike Ozden *et al.* [5], we take a different approach by employing the particle filter based BOT on a point of the moving object to solve the relative scale problem. The state of the moving object (*i.e.* position and velocity) and the associated uncertainty is continuously estimated by the tracker and is completely represented by its set of particles. The mean of the particles is thus the best estimate of the moving point from the filtering point of view and with the assumptions (state transition model) made in design of the filter. When the BOT is able to estimate the depth of a moving point up to a reasonable certainty, we can use this depth to fix the relative scale, and get a realistic multibody reconstruction. Apart from the online nature of the solution, the BOT can also estimate the state of an object, for which reconstruction is not possible. Denoting the posterior depth estimate as obtained by BOT of a point on the moving object by d_{BOT}, and d_{SFM} as depth of the same point as computed by the visual SLAM on that object. The map points X_O and camera poses g_{CO} are scaled by $s = d_{BOT}/d_{SFM}$, before being added to the integrated map.

5.2. Feedback from SfM to BOT

For the objects chosen for reconstruction, a successful reconstruction of the moving object from the visual SLAM module can help to improve the bearing only tracking (BOT). As described in Sec. 5.1, there exist a 1-parameter family of possible solutions for the trajectory of a moving point. Let d_{SFM} denote the depth of the tracked moving point from the camera in the object frame, and d_{iS} be the depth of ith particle from the camera pose in the static world frame. Using Eq. 8, the

$$\mathbf{t}_{iS} = s_i\mathbf{R}_{CS}^T\mathbf{t}_{CO} - \mathbf{R}_{CS}^T\mathbf{t}_{CS} \tag{9}$$

where $s_i = d_{iS}/d_{SFM}$. Thus for a particle at particular depth, SfM on the moving object gives a unique estimate of the particle translation. This information can be used during measurement update, and also to set the motion model for the next state transition. Thus when SfM estimates are available this can act as a secondary observation. The observation function is then given by Eq. 9. The measurement update computes a distance measure between the particle positions estimated from Eq. 9 and the predicted position of the particle by motion model. Thus particles having different velocity than that estimated by the SfM, but still lying on the projected ray can now be assigned lower weights or rejected. For the particles which survived the resampling after this measurement update, the motion models of the particles are set in accordance with that estimated by Eq. 9. Let the twist matrix corresponding to this transformation estimate given by SfM for a particle i be denoted as $\widehat{\xi^t_{i,SFM}}$. The particle i is then sampled based on the motion model given by the pdf $p(g_{iS}^{t+1}|g_{iS}^t, \xi^t_{i,SFM})$, which essentially generates a particle with mean

$$g_{iS}^{t+1} = \exp(\widehat{\xi^t_{i,SFM}}\Delta t)g_{iS}^t \tag{10}$$

Between two views, SfM estimate obtained from Visual SLAM module reduces the set of possible trajectories from all possible trajectories lying along the two projection rays, to an one-parameter family of trajectories as given by Eq. 8.

6. Experiments

6.1. System Results

The system has been tested on a number of publicly available real image datasets[1], with varying number and type of moving entities. Figures are best viewed on screen.

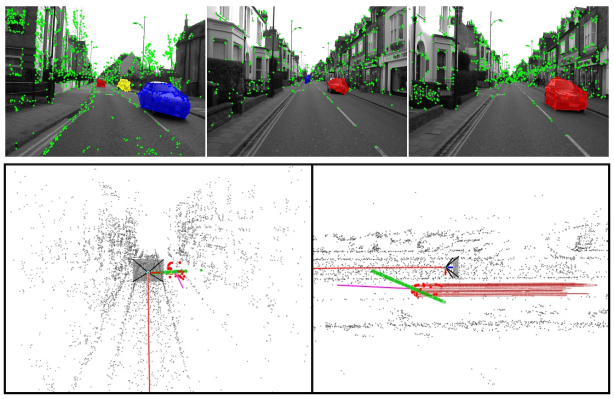

Figure 2. Results on the CamVid dataset. The top image shows output of motion segmentation. The bottom left image shows the reconstruction of the static world and a moving car at certain instant. Particles of the BOT are shown in green and the trajectory of camera is colored red. Bottom right image also shows the estimated 3D trajectory of the moving car.

[1] Additional results available at project website

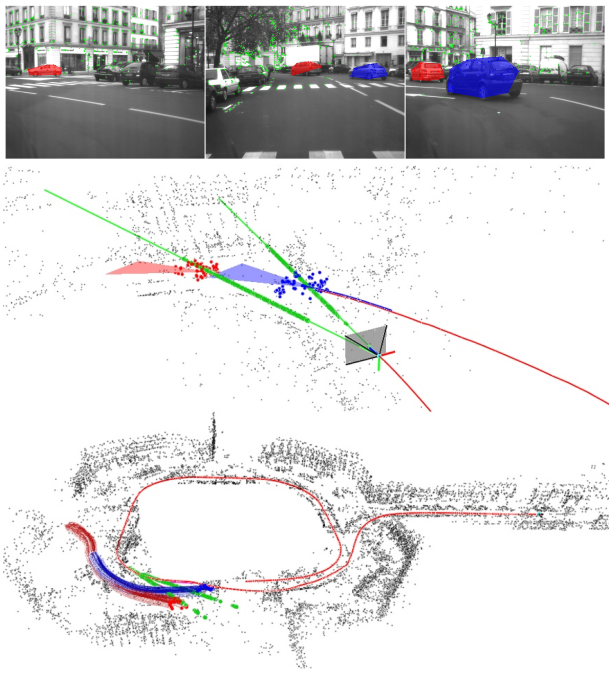

Figure 3. Results on the Versailles Rond Sequence. Top image samples some segmentation results from the sequence. The middle image shows an instance of the online occupancy map. Shaded region shows the most likely space to be occupied in next 16 frames (around 1s). Bottom image demonstrates the reconstruction and trajectories of two moving cars.

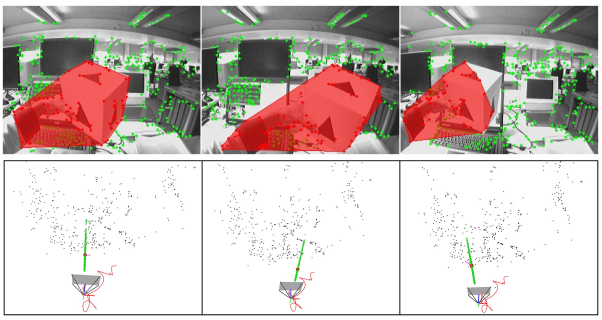

Figure 4. Results for the Moving Box sequence

Camvid Sequence: We tested our system on some dynamic parts of the CamVid dataset [3]. This a road sequence involving a camera mounted on a moving car. The results on this sequence is highlighted in Fig. 2. It shows the camera trajectory and 3D structure of static background. Reconstruction and the 3D trajectory of a moving car in the scene as produced by the system are also shown. Note the high degree of correlation between camera and the car trajectory, which makes it challenging for both motion segmentation and relative scale estimation.

Versailles Rond Sequence: This is an urban outdoor sequence taken from a fast moving car, with multiple numbers of moving objects appearing and leaving the scene. Only left of the stereo image pairs has been used. Fig. 3 shows the results of the integrated map produced by the algorithm. The middle image shows an instance of the online occupancy map, consisting of the 3D reconstruction of two moving cars, corresponding BOT tracker and most likely occupancy map of the moving objects in next instant. Bottom of Fig. 3 demonstrate the reconstructed trajectory of two moving cars, shown in red and blue.

Moving Box Sequence: This is same sequence as used in [25]. A previously static box is being moved in front of the camera which is also moving arbitrarily. However unlike [25], our method does not use any 3D model, and thus can work for any previously unseen object. As shown in Fig. 4 our algorithm reliably detects the moving object just on the basis of motion constraints. However, the foreground moving box is nearly white and thus provides very less features for reconstruction. This sequence also highlights the detection of previously static moving objects. Upon detection, 3D map points lying on the moving box are deleted and their 3D coordinates are used to initialize the BOT as described in Sec. 4.3.

6.2. Discussion

We have shown results for multibody visual SLAM under highly correlated camera-object motion, degenerate motion, arbitrary camera trajectory and changing number of moving entities. Also the algorithm is online (causal) in nature and also scales to arbitrary long sequences.

Smooth Camera Motions: Moving object reconstruction and tracking from a smoothly moving camera is very challenging. It becomes unobservable for a naive BOT, and results in very high correlation and thus rendering the methods of [17, 5] unsuitable. Left of Fig. 5 shows the trajectory of a single point on the 5th moving car of Versailles Rond sequence for three different scales. Contrary to [5], the trajectories at wrong scales does not show any accidentalness or violation of heading constraint, which proves its ineffectiveness for relative scale estimation from smoothly moving cameras. Our tracking framework coupled with the various feedbacks is able to provide a realistic estimate and also captures the high uncertainty present in such cases. Typical road scenes also involves frequent degenerate motions, making them hard even for detection. Right image of Fig. 5 shows an example of degenerate motion detection, as the flow vectors on the moving person almost move along epipolar lines, but they are being detected due to usage the FVB constraint (Sec. 2.2) which gets improved by incorporating feedback from static world reconstruction.

Comparison of different cues to BOT: Fig. 6 shows improvement in bearing only tracking for different cues. Left graph shows the depth variance obtained for a moving car in CamVid sequence. Since it is only tracked through BOT, no SFM cue is available. Whereas the right figure compares the performance for 3rd moving car in Versilles Rond sequence. As seen in Fig. 6, feedback from SfM has the high-

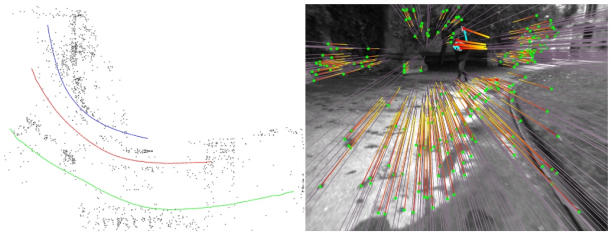

Figure 5. LEFT: Moving object trajectory for three different scales of 0.04, 0.11 and 0.18, where 0.11(red) being the correct scale. RIGHT: Degenerate motion detection. Epipolar lines in Grey, flow vectors after rotation compensation is shown in orange. Cyan lines show the distance to epipolar line. Moving Features detected are shown as red dots.

est effect in decreasing the uncertainty among all cues. For a particular particle of BOT filter, the ground-plane (GP) cue constraints possible velocities to lie parallel to the plane. Whereas SFM cue to BOT restricts it to an unique velocity vector for each depth of the particle. Depth and Size bounds can perform well even for highly correlated motions.

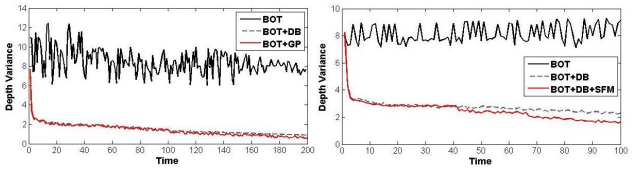

Figure 6. Comparison of different cues to the BOT namely Depth Bound (DB), Ground Plane (GP) and SFM feedback.

System Details: The system is implemented as threaded processes in C++. The open source libraries of TooN, OpenCV and SBA (for bundle adjustment) are used throughout the system. Runtime of the algorithm depends on lot of factors like the number of bodies being reconstructed, total number of independent motions being tracked by the BOT, image resolution and bundle adjustment rules. The system runs in realtime at the average of 14Hz (more dataset specific runtime details in supplementary material) in a standard laptop (Intel Core i7) compared to 1 minute per frame of [16], with up to two moving objects being simultaneously tracked and reconstructed.

7. Conclusions

We presented a multibody visual SLAM system which is an adaptation of multibody SfM theory in similar lines as visual SLAM is for standard offline batch SfM theory. We believe the proposed algorithm is one of the first systems to obtain a fast incremental multibody reconstruction across long real-world sequences involving difficult degenerate and highly correlated motions, arising from a smoothly moving monocular camera. The different modules of motion segmentation, visual SLAM and moving object tracking were integrated and we presented, how each module helps the other one. We present a particle filter based BOT algorithm, which integrates multiple cues from the reconstruction pipeline. The integrated system can simultaneously perform realtime multibody visual SLAM, tracking of multiple moving objects and unified representation of them using only a single monocular camera.

References

[1] S. Avidan and A. Shashua. Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence. *PAMI*, 22(4):348–357, 2002. 1

[2] T. Brehard and J. Le Cadre. Hierarchical particle filter for bearings-only tracking. *IEEE TAES*, 43(4):1567–1585, 2008. 2, 4

[3] G. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *PRL*, 30(2):88–97, 2009. 7

[4] A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *PAMI*, 29(6):1052–1067, 2007. 1, 2

[5] K. Egemen Ozden, K. Cornelis, L. Van Eycken, and L. Van Gool. Reconstructing 3D trajectories of independently moving objects using generic constraints. *CVIU*, 96(3):453–471, 2004. 2, 6, 7

[6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 1, 6

[7] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *ISMAR*, 2007. 1, 4

[8] A. Kundu, K. M. Krishna, and C. V. Jawahar. Realtime motion segmentation based multibody visual slam. In *ICVGIP*, 2010. 3

[9] J. Kwon and K. Lee. Monocular SLAM with Locally Planar Landmarks via Geometric Rao-Blackwellized Particle Filtering on Lie Groups. In *CVPR*, 2010. 1, 4

[10] J.-P. Le Cadre and O. Tremois. Bearings-only tracking for maneuvering sources. *IEEE TAES*, 34(1):179 –193, 1998. 2

[11] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the pnp problem. *IJCV*, 81(2):155–166, 2009. 4

[12] K. Lin and C. Wang. Stereo-based Simultaneous Localization, Mapping and Moving Object Tracking. In *IROS*, 2010. 2

[13] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3d reconstruction. In *CVPR*, 2006. 4

[14] J. Neira, A. Davison, and J. Leonard. Guest editorial, special issue in visual slam. *IEEE T-RO*, 24(5):929–931, 2008. 1

[15] D. Nister. An efficient solution to the five-point relative pose problem. *PAMI*, 26(6):756–770, 2004. 4

[16] K. E. Ozden, K. Schindler, and L. V. Gool. Multibody structure-from-motion in practice. *PAMI*, 32:1134–1141, 2010. 1, 2, 8

[17] H. S. Park, I. Matthews, and Y. Sheikh. 3d reconstruction of a moving point from a series of 2d projections. In *ECCV*, 2010. 1, 2, 7

[18] S. Rao, A. Yang, S. Sastry, and Y. Ma. Robust Algebraic Segmentation of Mixed Rigid-Body and Planar Motions from Two Views. *IJCV*, 2010. 1

[19] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *PAMI*, 32:105–119, 2010. 3

[20] K. Schindler and D. Suter. Two-view multibody structure-and-motion with outliers through model selection. *PAMI*, 28(6):983–995, 2006. 1

[21] G. Schweighofer and A. Pinz. Robust pose estimation from a planar target. *PAMI*, pages 2024–2030, 2006. 4

[22] J. Sola. *Towards visual localization, mapping and moving objects tracking by a mobile robot: a geometric and probabilistic approach*. PhD thesis, LAAS, 2007. 2

[23] H. Strasdat, J. Montiel, and A. Davison. Scale Drift-Aware Large Scale Monocular SLAM. In *RSS*, 2010. 1, 4

[24] R. Vidal, Y. Ma, S. Soatto, and S. Sastry. Two-view multibody structure from motion. *IJCV*, 68(1):7–25, 2006. 1

[25] S. Wangsirpitak and D. Murray. Avoiding moving outliers in visual slam by tracking moving objects. In *ICRA*, 2009. 2, 7