Joint SFM and Detection Cues for Monocular 3D Localization in Road Scenes

Shiyu Song Manmohan Chandraker
NEC Labs America, Cupertino, CA

Abstract

We present a system for fast and highly accurate 3D localization of objects like cars in autonomous driving applications, using a single camera. Our localization framework jointly uses information from complementary modalities such as structure from motion (SFM) and object detection to achieve high localization accuracy in both near and far fields. This is in contrast to prior works that rely purely on detector outputs, or motion segmentation based on sparse feature tracks. Rather than completely commit to tracklets generated by a 2D tracker, we make novel use of raw detection scores to allow our 3D bounding boxes to adapt to better quality 3D cues. To extract SFM cues, we demonstrate the advantages of dense tracking over sparse mechanisms in autonomous driving scenarios. In contrast to complex scene understanding, our formulation for 3D localization is efficient and can be regarded as an extension of sparse bundle adjustment to incorporate object detection cues. Experiments on the KITTI dataset show the efficacy of our cues, as well as the accuracy and robustness of our 3D object localization relative to ground truth and prior works.

1. Introduction

The rapid advent of autonomous driving technologies has introduced the need for accurate 3D localization of objects such as cars in real-world driving scenarios. The applications of real-time object localization range from driver safety, to danger prediction, to better understanding of traffic scenes. This paper presents a framework for 3D object localization that combines cues from structure from motion (SFM), object detection and ground plane estimation to achieve high accuracy, using only monocular video as input.

The key to our accurate 3D object localization is the joint optimization framework of Section 4 that accounts for SFM and object detection as complementary modalities for scene understanding. Monocular SFM cues consist of 3D points on the object and a per-frame estimate of the ground plane, while detection cues include 2D bounding boxes and detector scores. The mutual interactions of these cues is governed by our joint optimization to exploit their relative strengths.

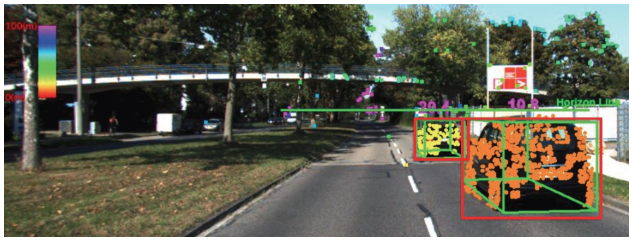

Figure 1. We demonstrate a 3D object localization framework that combines cues from SFM and object detection. Red denotes 2D bounding boxes, the horizontal line is the horizon from estimated ground plane, green denotes estimated 3D localization for far and near objects, with distances in magenta. Notice that for the closest object, the 3D bounding box is accurate even though the 2D one is not. This shows the effectiveness of our joint optimization, that incorporates SFM cues, raw detection scores and 3D priors.

Intuitively, SFM can estimate accurate 3D points on nearby objects, but suffers due to the low resolution of those far away. On the other hand, bounding boxes from object detection are obtainable for distant objects, but are often inconsistent with the 3D scene in the near field. Thus, we seek 3D bounding boxes that are most consistent with 2D tracked ones, while also maximizing the alignment of estimated object pose with tracked 3D points (Figure 1).

Besides detected 2D bounding boxes, we also make novel use of object detection scores. Our system is designed for a real-time application, so cannot afford complex scene understanding approaches. Rather, SFM, object detection, object tracking and 3D localization are sequential operations in our system, so inaccuracies in earlier stages must be compensated. The input to 3D object localization are 2D tracklets from tracking-by-detection, which are often noisy and poorly localized. Section 4 proposes a method to incorporate raw detection scores in our joint optimization, while avoiding the prohibitive cost of evaluating the detector model for every object pose configuration. This allows us to efficiently use detection bounding boxes with high enough scores that are more consistent with 3D geometry, to effectively undo any poor localization of the 2D object tracks.

An important aspect of our object localization is the use of 3D points as SFM cues, for which we use dense tracking that exploits intensity-aligned pose optimization [9]. Section

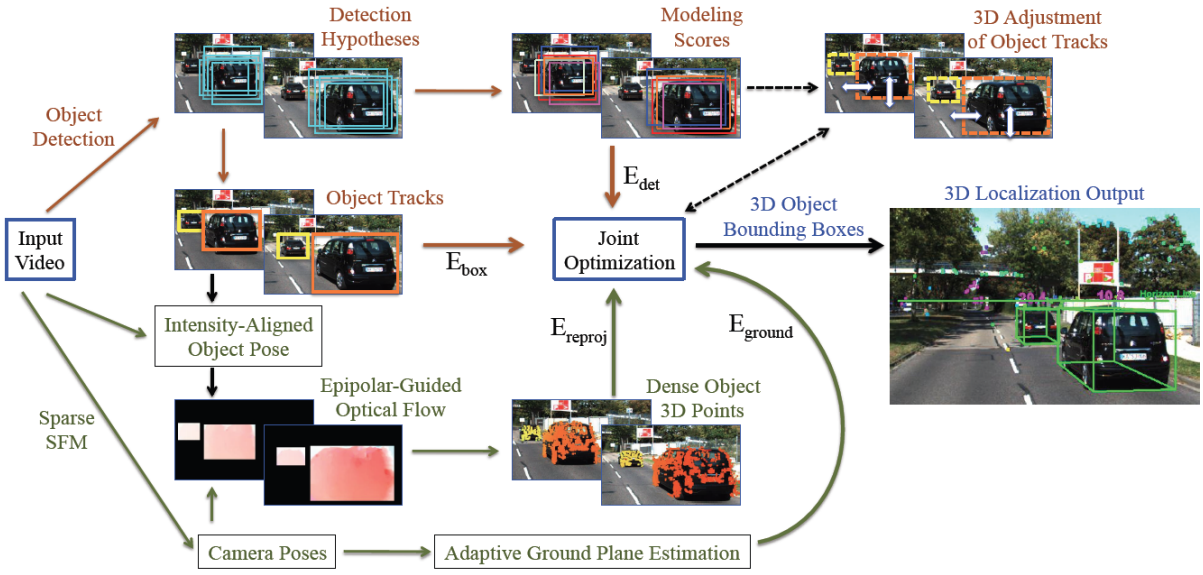

Figure 2. Overview of our system that estimates 3D object localization by combining SFM cues (green) with object detection cues (brown). Given monocular video input, camera poses and ground plane are estimated by SFM, while a dense tracking framework is used to obtain 3D points on objects. These are combined with cues from object detection hypotheses and object tracks in a joint optimization framework that allows for soft adjustment of track positions to maximize consistency with 3D cues, bounding boxes and detection scores. Details on the object cues are presented in Section 4, while object SFM is further elaborated in Section 5.

5 makes a crucial observation that such an approach has distinct advantages over a pose estimation based on sparse feature matching, which is severely limited in autonomous driving scenarios. The intensity-aligned pose provides epipolar constraints to guide a TV-L1 optical flow, which leads to improved accuracy [16]. Unlike PnP-based pose estimation, these epipolar constraints are not derived from feature tracks, so can instead be used to improve the quality of dense tracking. Further, accurate dense tracks added through our mechanism prevent the system from catastrophic breakdown, as would happen if too few sparse features are available for a stable PnP-based pose estimation.

We show in Section 6 that combining cues from SFM and object detection significantly improves 3D localization for both near and distant objects. The benefit of our cue combination is available even for more comprehensive monocular scene understanding frameworks like [3, 20]. We demonstrate this experimentally by using the object tracks of [3] within our joint optimization framework, to achieve a significant improvement in object localization accuracy.

To summarize, our main contributions are:

- A joint optimization framework for 3D object localization that combines SFM cues such as ground plane and 3D points on the object, with object cues such as 2D bounding boxes and detection scores, to achieve high accuracy in both near and far fields.
- Incorporation of raw detection scores to allow 3D bounding boxes to "undo" tracking errors, that is, achieve consistency with both 3D geometry and detection scores.
- A dense tracking framework for challenging objects like cars in driving scenarios that can compensate for unstable detection outputs, to reliably estimate 3D object pose.

2. Related Work

Multibody SFM has been proposed in the past to simlutaneously localize a moving camera and moving objects [11, 12]. In addition, Schindler et al. [15] propose a model selection for segmentation. However, multibody SFM for moving object localization has been demonstrated only for short sequences. Indeed, in real driving scenarios, it is challenging to obtain reliable feature tracks in sufficient numbers for multibody SFM by itself to be robust, due to small object size, fast speeds and lack of texture.

To localize moving objects, Ozden et al. [12] and Kundu et al. [10] use joint motion segmentation and SFM. Brox et al. [2] use a combination of sparse SFM and dense optical flow for joint tracking and segmentation. In practice, it is difficult to obtain stable feature tracks on low-textured objects like cars when they are not close and segmentation is challenging in actual driving videos where camera and various object motions are often correlated. A different approach is that of multi-target tracking frameworks that combine object detection with stereo [4] or monocular SFM [3, 20]. Detection can handle farther objects, decouples feature tracking for individual objects and together with the ground plane, provides a cue to estimate object scales that

are difficult to resolve for traditional monocular SFM even with multiple segmented motions [13].

The utility of an adaptively estimated ground plane is shown in [17], however, localization is performed only as a triangulation against a common ground plane. In contrast, this paper proposes a joint optimization based on cues from object detection and dense 3D points, while also allowing variations in the ground plane among objects. Multi-target tracking works like [3, 4, 20] do not show 3D localization results, although it plays a role in their frameworks. Also related are works on 3D object detection in a single image [6, 14]. Similarly, systems like [7] use a stereo setup to infer the layout of urban traffic junctions. Recent scene understanding frameworks also reason about object relationships to handle occlusions in monocular input [23]. Unlike those works, the sequential design in our system is governed by efficiency requirements, but we do handle object tracks in a soft fashion to achieve a localization most consistent with both 3D cues and object detection. Our contributions are complementary to the above scene understanding and multi-target tracking frameworks, which can also benefit from our novel use of 3D points and detection cues. To demonstrate this benefit, we incorporate object tracks from [3] in our framework and obtain a significant improvement in localization.

Our input is monocular video, so a robust mechanism is needed to estimate 3D points despite challenges in driving scenarios. As our experiments show, traditional sparse SFM does not suffice. Instead, we use a combination of dense intensity-aligned pose estimation [9], an epipolar-guided extension [16] to TV-L1 optical flow of [22] and consistency checks in the spirit of sparse SFM, to extract 3D points for our joint optimization framework. Unlike [19], we do not use a fundamental matrix to constrain flow vectors, since sparse feature matches on objects are not reliable. While more accurate optical flow methods are available [1, 21], we choose TV-L1 for its balance of accuracy and speed.

3. Background

Notation A vector in $\mathbb{R}^n$ is denoted $\mathbf{x} = (x_1, \cdots, x_n)^\top$. A matrix is denoted as $\mathbf{X}$. The homogeneous representation of vector $\mathbf{x}$ is $\widetilde{\mathbf{x}} = (\mathbf{x}^\top, 1)^\top$. A variable x in frame t of a sequence is represented as x_t or $x(t)$. A set of variables indexed by i is denoted $\{\mathbf{x}^i\}$.

Ground plane geometry As shown in Figure 3, the camera height (also called ground height) h is defined as the distance from the principal center to the ground plane. Usually, the camera is not perfectly parallel to the ground plane and there exists a non-zero pitch angle θ. The ground height h and the unit normal vector $\mathbf{n} = (n_1, n_2, n_3)^\top$ define the ground plane. For a 3D point $(x, y, z)^\top$ on the ground plane, we have $h = y\cos\theta - z\sin\theta$.

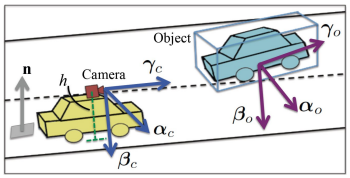

Figure 3. Coordinate system definitions for 3D object localization. The SFM ground plane is $(\mathbf{n}^\top, h)^\top$.

Object localization through ground plane Accurate estimation of both ground height and orientation is crucial for 3D object localization. Let $\mathbf{K}$ be the camera intrinsic calibration matrix. As [3, 4, 20], the bottom of a 2D bounding box, $\mathbf{b} = (x, y, 1)^\top$ in homogeneous coordinates, can be back-projected to 3D through the ground plane $\{h, \mathbf{n}\}$:

$$\mathbf{c} = (c_x, c_y, c_z)^\top = -\frac{h\mathbf{K}^{-1}\mathbf{b}}{\mathbf{n}^\top\mathbf{K}^{-1}\mathbf{b}}, \quad (1)$$

Similarly, the object height can also be obtained using the estimated ground plane and the 2D bounding box height.

Monocular SFM and ground plane estimation For the background SFM (camera self-localization), we use the monocular system of [17], with scale drift corrected using an adaptive ground plane estimation. This is an important choice – as shown in our experiments, using an inaccurate or fixed ground plane from calibration cannot be an option for reliable 3D localization over long sequences.

4. Joint Use of SFM and Detection Cues

As discussed in Section 1, SFM and 2D object bounding boxes offer inherently complementary cues for scene understanding. We now present a framework that combines SFM cues (3D points and ground plane) with those from object detection (2D bounding boxes and detection scores), to localize both near and far objects in 3D. We formulate the problelm in an energy minimization framework consisting of SFM and object costs, with additional terms to enforce consistency with prior knowledge. We begin by defining the 3D coordinate system and our representation of object pose. Figure 2 illustrates an overview of the system.

4.1. 3D Coordinate System

Consider camera coordinate system $\mathcal{C}$ with orthonormal axes $(\boldsymbol{\alpha}_c, \boldsymbol{\beta}_c, \boldsymbol{\gamma}_c)$ and an object coordinate system $\mathcal{O}$ with axes $(\boldsymbol{\alpha}_o, \boldsymbol{\beta}_o, \boldsymbol{\gamma}_o)$. Let the origin of object coordinates be the 3D point $\mathbf{c}_o = (x_c, y_c, z_c)^\top$, expressed in camera coordinates, corresponding to center of the line segment where the back plane of the object intersects the ground. Let the ground plane be parameterized as $\mathbf{g} = (\mathbf{n}^\top, h)^\top$, where $\mathbf{n} = (\cos\theta\cos\phi, \cos\theta\sin\phi, \sin\theta)^\top = (n_1, n_2, n_3)^\top$ and $h = -\mathbf{c}_c^T\mathbf{n}$. We assume that the object lies on the ground plane and is free to rotate in-plane with yaw angle ψ. Thus,

the pose of the object i is completely determined by a six-parameter vector $\mathbf{\Omega}^i = (x_c^i, z_c^i, \psi^i, \theta^i, \phi^i, h^i)^\top$. The coordinate system definitions are visualized in Figure 3.

With the above definitions, one may transform between object and camera systems using the ground plane, object yaw angle and object position. Define $\mathbf{N} = [\mathbf{n}_\alpha, \mathbf{n}_\beta, \mathbf{n}_\gamma]$, where $\mathbf{n}_\gamma = (-n_1, n_3, -n_2)^\top$, $\mathbf{n}_\beta = -\mathbf{n}$ and $\mathbf{n}_\alpha = \mathbf{n}_\beta \times \mathbf{n}_\gamma$. Then, given a homogeneous 3D point $\widetilde{\mathbf{x}}_o$ in the object coordinate system, the transformation from object to camera coordinates is given by $\widetilde{\mathbf{x}}_c = \mathbf{P}_\pi \mathbf{P}_\psi \widetilde{\mathbf{x}}_o$, with:

$$\mathbf{P}_\pi = \begin{bmatrix} \mathbf{N} & \mathbf{c}_o \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad \mathbf{P}_\psi = \begin{bmatrix} \exp\left([\boldsymbol{\omega}_\psi]_\times\right) & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad (2)$$

where $\boldsymbol{\omega}_\psi = (0, \psi, 0)^\top$ and $[\cdot]_\times$ is the cross product matrix. The projection function for a 3D point $\mathbf{x}_o$ in object coordinates to the 2D point $\mathbf{u}$ on the image plane is denoted $\mathbf{u} = \pi_{\mathbf{\Omega}}(\mathbf{x}_o)$, which is the inhomogeneous version of

$$\lambda \widetilde{\mathbf{u}} = \mathbf{K} \left[\mathbf{I} \mid \mathbf{0} \right] \mathbf{P}_\pi \mathbf{P}_\psi \widetilde{\mathbf{x}}_o. \quad (3)$$

4.2. SFM Cues

The pathways for incorporating 3D cues in our system are illustrated in green in Figure 2.

Tracked 3D points Let N objects be tracked in the scene over T frames, with object i being tracked from frames s_i to e_i. During this interval, suppose M_i feature points are triangulated by the object SFM mechanism (detailed in Section 5). In the object coordinates, we denote this set of 3D points as $\mathbf{X}_o^i = [\mathbf{x}_1^i, \cdots, \mathbf{x}_{M_i}^i]$. Since the object is rigid, note that the location of each $\mathbf{x}_j^i$ does not depend on time. Let $\mathbf{u}_j^i(t) = (u_j^i(t), v_j^i(t))^\top$ be the 2D point corresponding to $\mathbf{x}_j^i$ in frame t. Then, the first component of the SFM cost favors the object poses $\{\mathbf{\Omega}^i(t)\}$, for $i = 1, \cdots, N$, that minimize the reprojection error:

$$\mathcal{E}_{reproj}\left(\{\mathbf{\Omega}^i(t)\}\right) = \sum_{i=1}^{N} \sum_{t=s_i}^{e_i} \sum_{j=1}^{M_i} \|\mathbf{u}_j^i(t) - \pi_{\mathbf{\Omega}^i(t)}(\mathbf{x}_j^i)\|^2. \quad (4)$$

Note that there is an overall ambiguity in the origin of $\mathcal{O}$ with respect to $\mathcal{C}$ that cannot be resolved by SFM alone. To do so, we require input from object bounding boxes.

Ground plane Recall that the background monocular SFM outputs a ground plane estimate at every frame. The object pose defined in Section 4.1 also depends on the ground plane. Unlike prior works, we do not impose a shared ground plane for all objects. Rather each object resides on its own ground plane, which is useful in practical situations where the ground plane variation within the field of view is high. The background SFM ground plane, $\bar{\mathbf{g}}(t)$, is now used as a prior for the object ground plane, $\mathbf{g}^i(t)$:

$$\mathcal{E}_{ground}\left(\{\mathbf{\Omega}^i(t)\}\right) = \sum_{i=1}^{N} \sum_{t=s_i}^{e_i} \|\mathbf{g}^i(t) - \bar{\mathbf{g}}(t)\|^2. \quad (5)$$

The combined cost from the SFM cues (point tracks and ground plane) is now a weighted sum:

$$\mathcal{E}_{sfm} = \mathcal{E}_{reproj} + \lambda_g \mathcal{E}_{ground}. \quad (6)$$

4.3. Object Cues

The framework for incorporating object detection cues in our system is shown in brown in Figure 2.

Object bounding box Let the dimensions of the object 3D bounding box (to be estimated) be $l_\alpha, l_\beta, l_\gamma$ along the $\boldsymbol{\alpha}_o, \boldsymbol{\beta}_o, \boldsymbol{\gamma}_o$ axes. Then, locations of the 3D bounding box vertices, in object coordinates, are $\mathbf{B} = [\mathbf{v}_1, \cdots, \mathbf{v}_8]$, where $\mathbf{v}_1 = (-l_\alpha/2, 0, 0)^\top, \cdots, \mathbf{v}_8 = (l_\alpha/2, -l_\beta, l_\gamma)^\top$. Note that the $\mathbf{B}$ is in object coordinates and does not vary over time. Let the projected edges of the 3D bounding box at frame t be $\mathbf{b}(t) \in \mathbb{R}^4$, which are the extrema of the projected 3D vertices along both the image axes. With a mild abuse of notation, we will denote $\mathbf{b}(t) = \pi_{\mathbf{\Omega}(t)}(\mathbf{B})$. Let $\mathbf{d}(t)$ be the corresponding four sides of the tracked 2D bounding box in frame t. Then, we define an object bounding box error:

$$\mathcal{E}_{box}\left(\{\mathbf{\Omega}^i(t)\}, \{\mathbf{B}^i\}\right) = \sum_{i=1}^{N} \sum_{t=s_i}^{e_i} \|\pi_{\mathbf{\Omega}^i(t)}(\mathbf{B}^i) - \mathbf{d}^i(t)\|^2. \quad (7)$$

Object detection scores While we use the output of an object tracker, we must also be aware that tracked bounding boxes are not always accurate. There are two issues that 3D localization must address:

- A 2D tracker might not always pick the detection bounding box with the highest score, rather it would pick one with a high score that is most consistent with priors like smoothness and track length. However, these constraints are imposed in 2D by most trackers, so our 3D localization must be provided an opportunity to undo any suboptimal choices made by the 2D tracker.
- Further, many tracking-by-detection frameworks use a discrete set of 2D bounding boxes obtained after nonmaximal suppression of the detector output. Our localization framework considers a continuous model of the raw detection output, which is crucial for 3D consistency.

To address the first issue, a straightforward approach would be to simply attempt to find the 3D bounding box $\mathbf{B}$ whose projection maximizes the detection score. However, note that this requires the detector model to be evaluated at every function evaluation of the 3D localization, which can be too expensive for real-time operation. So, we adopt an alternative approach that approximates the detection scores with a model that is easy to evaluate. In particular, given the set of 2D bounding box hypothesis from an object detector [5] over the four-dimensional space of image position, length and width, we model the corresponding detection scores as a sum of Gaussians, fitted for the amplitude, mean and full

covariance matrix. Note that the number of objects at each frame is already known from the 2D tracking output, thus, the estimation of the means and 4×4 convariances of the Gaussians is a straightforward non-linear minimization (for which we use a Levenberg-Marquardt routine).

Let the estimated model of detection scores at time t be denoted $\mathcal{S}_t$, which yields a score $\mathcal{S}_t(\mathbf{b})$ for a 2D bounding box $\mathbf{b}$ at frame t. As a result of the above modeling, during every function evaluation for 3D localization, for each putative 3D bounding box $\mathbf{B}$ and object pose $\mathbf{\Omega}$, we can estimate the detector score $\mathcal{S}_t(\pi_{\mathbf{\Omega}}(\mathbf{B}))$ without having to evaluate the detection model. We are now in a position to propose an efficient detection cost for 3D localization, which simply attempts to find the 3D bounding box and object pose that achieves the best detection score:

$$\mathcal{E}_{det}\left(\{\mathbf{\Omega}^i(t)\},\{\mathbf{B}^i\}\right) = \sum_{i=1}^{N}\sum_{t=s_i}^{e_i}\left(\frac{1}{\mathcal{S}_t(\pi_{\mathbf{\Omega}^i(t)}(\mathbf{B}^i))}\right)^2. \quad (8)$$

Thus, we use the raw detection output to allow the estimated 3D bounding box to overcome any suboptimal choices made by the 2D tracker in assignment of bounding boxes, while avoiding the cost of too many detector evaluations for every putative 3D bounding box and object pose.

The total object cost is a weighted sum of the bounding box and detection costs:

$$\mathcal{E}_{obj} = \mathcal{E}_{box} + \lambda_d \mathcal{E}_{det}. \quad (9)$$

4.4. Priors

We impose two priors for 3D localization: object size and trajectory smoothness. Let $\mathbf{x}_w(t)$ be the 3D position of the object, in world coordinates, at frame t. Then the trajectory smoothness prior constitutes an energy given by

$$\mathcal{E}_{smooth} = \sum_{i=1}^{N}\sum_{t=s_i-1}^{e_i+1} \|\mathbf{x}_w(t-1) - 2\mathbf{x}_w(t) + \mathbf{x}_w(t+1)\|^2. \quad (10)$$

Let $\{\bar{l}_\alpha, \bar{l}_\beta, \bar{l}_\gamma\}$ be the priors on the object dimensions, obtained from the KITTI dataset [7]. Then, the size energy is

$$\mathcal{E}_{size} = \sum_{i=1}^{N}\sum_{j\in\{\alpha,\beta,\gamma\}} (l_j^i - \bar{l}_j)^2. \quad (11)$$

The total energy from imposing priors on 3D localization is then given by a weighted sum:

$$\mathcal{E}_{prior} = \mathcal{E}_{smooth} + \lambda_s \mathcal{E}_{size}. \quad (12)$$

4.5. Joint Optimization

With the above definitions of the various cues, we define the combined energy function to be minimized over the set of object poses $\{\mathbf{\Omega}^i(t)\}$, 3D bounding box dimensions $\mathbf{B}^i$ and the set of tracked 3D points on each object $\mathbf{X}_o^i$, for objects $i = 1, \cdots, N$, each of which is visible in frames s_i to e_i:

$$\mathcal{E}\left(\{\mathbf{\Omega}^i(t)\},\{\mathbf{B}^i\},\{\mathbf{X}_o^i\}\right) = \mathcal{E}_{sfm} + \lambda_o \mathcal{E}_{obj} + \lambda_p \mathcal{E}_{prior}, \quad (13)$$

where $\mathcal{E}_{sfm}$, $\mathcal{E}_{obj}$ and $\mathcal{E}_{prior}$ are defined in (6), (9) and (12), respectively. The optimization in (13) may be regarded as an extension of traditional bundle adjustment to incorporate object cues, since it is defined over a set of variables $\{\mathbf{\Omega}^i(t)\}$ that constitutes "poses" and another set given by $\{\mathbf{B}^i, \mathbf{X}_o^i\}$ that constitutes "3D points". Thus, it can be solved efficiently using a sparse Levenberg-Marquardt algorithm and is fast enough to match the real-time monocular SFM.

To maintain computational efficiency over long sequences, we perform the above joint optimization over a sliding window of maximum size $T = 50$ previous frames. Note that the computation is online and output is produced instantaneously. In all our experiments, the parameter values are empirically set to the following fixed values: $\lambda_0 = 0.7$, $\lambda_p = 2.7$, $\lambda_g = 2.7$, $\lambda_d = 0.03$ and $\lambda_s = 0.03$.

4.6. Initialization

The success of a local minimization framework as defined in (13) is contingent on a good initialization. We rely on the ground plane along with cues from both 2D bounding boxes and SFM to initialize the variables in (13), as follows.

Object Poses, $\{\mathbf{\Omega}^i\}$: The initial position of an object, $\widehat{\mathbf{c}}_o = (\widehat{x}_c, \widehat{y}_c, \widehat{z}_c)^\top$, is computed from the object bounding box and ground plane using (1). The initial yaw can be estimated from initial object positions in two frames $\widehat{\mathbf{c}}_o^{t-1}$ and $\widehat{\mathbf{c}}_o^{t+1}$. The object's ground height and orientation are initialized to the SFM ground plane. The initial pose of the object is now available as $\widehat{\mathbf{\Omega}} = (\widehat{x}_c, \widehat{z}_c, \widehat{\psi}, \theta, \phi, h)^\top$.

3D Bounding Boxes, $\{\mathbf{B}^i\}$: The initial object dimensions $(\widehat{l}_\alpha, \widehat{l}_\beta, \widehat{l}_\gamma)$ are computed as the optimal alignment to the 2D bounding box, by fixing $\widehat{l}_\gamma = \eta \widehat{l}_\alpha$ and minimizing the cost $\mathcal{E}_{box}$ over $\widehat{l}_\alpha$ and $\widehat{l}_\beta$, with a prior that encourages the ratio of bounding box sizes along $\boldsymbol{\gamma}_o$ and $\boldsymbol{\alpha}_o$ to be η. The practical reason for this regularization is that the camera motion is largely forward and most other vehicles in the scene are similarly oriented, thus, the localization uncertainty along $\boldsymbol{\gamma}_o$ is expected to be higher. By training on ground truth 3D bounding boxes in the KITTI dataset, we set $\eta = 2.5$.

3D Points, $\{\mathbf{X}^i\}$: For initialization, each tracked 2D feature point $\mathbf{u}$ is assumed to lie on the plane $\mathbf{n}_\gamma$, orthogonal to the ground. Its position in camera coordinates is

$$\mathbf{x}_c = -(\mathbf{n}_\gamma^\top \mathbf{c}_c)(\mathbf{n}_\gamma^\top \mathbf{K}^{-1}\widetilde{\mathbf{u}})^{-1}\mathbf{K}^{-1}\widetilde{\mathbf{u}}, \quad (14)$$

thus, from (2), the initial 3D point in object coordinates $\mathcal{O}$ is $\widetilde{\mathbf{x}}_o = \mathbf{P}_\pi^{-1}\mathbf{P}_\psi^{-1}\widetilde{\mathbf{x}}_c$.

5. Object Structure from Motion

In this section, we describe how we overcome practical challenges for extracting SFM cues on challenging objects

like cars. Since the background SFM relies on sparse feature matching, it is natural to initially consider a similar mechanism to estimate 3D points on objects. However, the PnP-based pose computation of the sparse pipeline requires prior knowledge of feature tracks, which are not plentiful on objects like cars. Instead, we use a dense pose estimation based on image intensity alignment similar to Kerl et al. in [9]. This has several advantages as discussed below and our experiments also demonstrate that the quality of pose estimated by intensity alignment is better in our application than obtainable by a sparse framework similar to the background SFM. An illustration of the system is shown in Figure 4.

Pose Estimation by Intensity Alignment Recall our definition of the object pose, $\mathbf{\Omega} = (x_c, z_c, \psi, \theta, \phi, h)^\top$, presented in Section 4.1. Suppose the object pose $\mathbf{\Omega}(t)$ in frame t is known, along with a set of reliable 3D points, $\{\mathbf{x}\}$. Then the object pose $\mathbf{\Omega}(t+1)$ can be estimated by minimizing the intensity difference between the projections of the 3D points in two neighboring frames:

$$\min_{\mathbf{\Omega}(t+1)} \sum_{i=1}^{N} \left[\mathbf{I}_t\left(\pi_{\mathbf{\Omega}(t)}(\mathbf{x}_i)\right) - \mathbf{I}_{t+1}\left(\pi_{\mathbf{\Omega}(t+1)}(\mathbf{x}_i)\right)\right]^2 \quad (15)$$

As this intensity alignment is only valid for a small motion between $\mathbf{\Omega}(t)$ and $\mathbf{\Omega}(t+1)$, the above optimization is embedded into an iterative warping approach to handle large motions. Note that at every frame, this estimated object pose undergoes a refinement akin to bundle adjustment through the joint optimization framework of Section 4, taking into account all the cues including 3D points and object bounding boxes. We are now in a position to use this pose for improving the quality of 3D points obtained by dense tracking.

Epipolar Guided TV-L1 Optical Flow Having access to the object pose computed from intensity alignement, rather than from sparse feature tracks, allows us to use epipolar constraints from the computed pose to guide an optical flow process that generates dense features tracks. Similar to [16], this reduces the optical flow from a 2D search on the image plane to a 1D search on the epipolar line, enhancing the accuracy of feature tracks since flow vectors are now constrained to be consistent with epipolar geometry. We use an implementation similar the 1-D stereo case in [22].

Dense Feature Tracking The TV-L1 optical flow between neighboring frames t and $t+1$ is used as input to the dense feature tracking. To maximize efficiency, we only compute optical flow within the small sub-image defined by the object bounding box. To ensure high quality for the tracks, we use the feature selection mechanism of [18] and divide each object region into 8×8 buckets, with only the pixels having the highest Harris corner responses selected to be tracked.

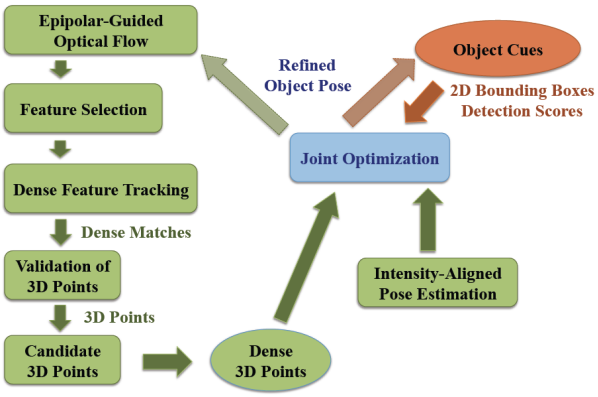

Figure 4. System overview for obtaining SFM cues on objects, depicted in green. An intensity-based pose alignment allows epipolar guidance for optical flow-based dense feature tracking. A validation step yields candidate 3D points that are added to the main thread when needed. The dense 3D points as well as intensity-aligned pose estimates undergo bundle adjustment together with object cues, using the framework of Section 4, to yield refined camera poses for use in the next time step.

Validation for 3D Points The tracks obtained by epipolar-guided optical flow are triangulated to obtain 3D points. However, to eliminate errors due to dense tracking drift, the 3D points must be validated before use in object localization. Each triangulated 3D point is reprojected into the past few images where it is visible. As a consistency check, only those points are retained for whom the NCC scores corresponding to all the reprojections are above a threshold. Now we have a new set of 3D points ready to be added to the main thread, on which the joint optimization of Section 4 resides, when required. It may be noted that the 3D points themselves are also refined by the joint optimization framework of Section 4 that incorporates other cues too.

6. Experiments

We present evaluation on the KITTI dataset [8], which contains real-world driving sequences. Since KITTI doesn't provide a benchmark for object localization and the ground truth 3D labels are not public for test sequences, we evaluate our 3D localization with the training sequences of the tracking benchmark. In particular, KITTI tracking training sequences 00–05, 10, 14, 15 and 18 that have moving cars are used for testing, while others are used for parameter tuning of SFM and 3D localization. We demonstrate 3D localization on ground truth bounding boxes, as well as tracked bounding boxes computed using [7] and [3]. Our system has been extensively tested on real-world driving scenarios.

The joint framework for 3D localization presented in Section 4 uses several cues to achieve high accuracy. First, it adaptively estimates the ground plane at every frame, instead

Method	Ground truth tracks						Tracked bounding boxes [7]					
	Near Obj			Far Obj			Near Obj			Far Obj		
	Z(%)	X(m)	Size(%)	Z(%)	X(m)	Size(%)	Z(%)	X(m)	Size(%)	Z(%)	X(m)	Size(%)
`CalibGround`	10.2	0.53	14.8	25.3	0.79	12.3	13.9	0.58	16.1	26.9	0.75	12.0
`AdaptiveGround`	9.0	0.38	14.8	9.8	0.35	12.3	13.3	0.50	16.1	10.2	0.33	**12.0**
`Ground+Opt`	6.4	0.26	9.3	8.9	0.35	13.3	9.5	0.33	13.5	**9.4**	0.34	13.6
`Ground+Opt+Det`	6.1	0.25	9.1	8.6	0.33	12.1	9.4	0.32	12.4	9.5	**0.33**	12.5
`Ground+Opt+Det+PnP`	5.9	0.24	8.1	8.5	0.34	**11.8**	9.4	0.30	10.9	11.2	0.37	14.2
`Ground+Opt+Det+Align`	**5.5**	**0.24**	**7.3**	**8.3**	**0.33**	12.0	**8.3**	**0.28**	**8.0**	10.4	0.36	13.9

Table 1. Comparison of 3D object localization errors for various cues used in our joint optimization framework, with bounding boxes from ground truth as well as the tracking output of [7]. The benefits of each of adaptive ground plane, object bounding boxes, detection scores and 3D points are clearly visible, as is the performance benefit from our dense tracking.

of relying on a fixed one. Second, it estimates and tracks 3D bounding boxes that are consistent with 2D tracks. Next, it uses raw detection scores to allow the 3D localization to recover from possibly suboptimal choices made by the 2D tracker. Finally, it incorporates SFM cues in the form of epipolar-guided dense feature tracks.

To demonstrate the effectiveness of each of the above contributions, we show the object localization accuracy with different methods in Table 1 using 2D bounding boxes from ground truth, as well as the tracking output of [7]. For each table, the left column lists different methods as various cues are added in the localization framework. The most important evaluation metric for our autonomous driving application is the percentage error in depth. We also list the horizontal localization accuracy in meters to give an idea of absolute errors and the percentage size error.

We differentiate between near and far objects in evaluating the results (although our localization method does not make any such distinction). This is to show the effectiveness of different cues at various distance ranges. For instance, we expect SFM cues to be more effective in the near range, while we expect the ground plane estimation to have a significant impact on far objects. We consider objects up to 15 meters away to be near.

`CalibGround` denotes the baseline method where localization is performed by directly back-projecting the bottom of the tracked 2D bounding box to 3D using (1), with a fixed ground plane. The calibration ground plane is $(\mathbf{n}^\top, h)^\top = (0, -\cos\theta, \sin\theta, 1.7)^\top$, with $\theta = -0.03$ for the KITTI dataset. Note that the localization errors are very high – clearly this is not suitable for autonomous driving.

`AdaptiveGround` uses the same back-projection of (1) to estimate the location of the 3D bounding box, however, the ground plane used is adaptively estimated at every frame, replicating the method of [17]. It is observed that the localization accuracy is especially improved for far objects, since small errors in ground plane orientation can have a large impact on 3D error over longer distances. A good ground plane also has a role in the stability of the joint optimization framework, since the object size, ground plane and the object distance are highly correlated entities.

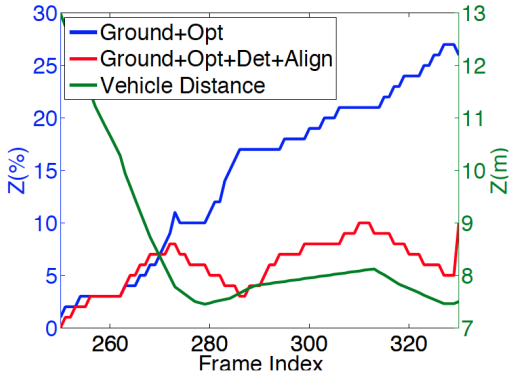

Figure 5. Benefit of SFM cues for 3D object localization. The green curve plotted against the right axis shows distance of an object as it approaches the camera. On the left axis, the blue curve shows object depth error when only object bounding box cues are used for localization, while the red curve incorporates SFM cues. SFM cues have a significant impact on localization accuracy in the near field.

In `Ground+Opt`, besides using the adaptive ground plane, we also estimate 3D bounding boxes that best fit the tracked 2D bounding boxes. Priors are also enabled for 3D trajectory smoothness and size constancy. Note that while [17] uses the same ground plane for all objects, in our case, each object is allowed to optimize its ground plane, which enhances accuracy by accounting for local variations. We observe that the injection of further 3D cues causes the errors to decrease, especially for near objects.

Next, we add object detection cues in the joint optimization framework to incorporate raw detection scores. The results are shown in the row labeled `Ground+Opt+Det`. It is clear that the error decreases further, since the system can now search for 2D detection bounding boxes that have high scores and are more consistent with 3D geometry.

In `Ground+Opt+Det+PnP`, we incorporate SFM cues in the joint optimization framework, but with a PnP based pose estimation. A full optical flow must be used now instead of an epipolar-guided one, since feature tracks are precursors to PnP. The remaining validation mechanisms are the same as the description in Section 5. However, due to the challenging

Seq. No.		0004				0047				0056	Total
Obj. ID		1	2	3	6	0	4	9	12	0	
No. Frames		91	251	284	169	170	96	94	637	293	
Z (%)	[3]	14.4	17.6	12.9	12.3	16.2	18.1	13.8	11.6	13.9	13.8
	[17]	**4.1**	6.8	5.3	7.3	9.6	**11.4**	7.1	10.5	**5.5**	7.9
	Ours	6.0	**5.6**	**4.9**	**5.9**	**5.9**	12.5	**7.0**	**8.2**	6.0	**6.8**

Table 2. Our 3D object localization can improve the performance of existing scene understanding frameworks such as [3]. This is due to our joint optimization that makes judicious use of an adaptive ground plane, 3D points, object bounding boxes and detection scores. Note that [17] uses a global optimization, while we use a windowed one and yet perform better in most instances.

size and texture of the objects under consideration, PnP is limited by outliers from unstable tracking. A PnP based object pose estimation based on sparse SIFT feature matches causes breakdowns due to highly inaccurate poses stemming from too few matches. We note that the improvement from adding PnP based SFM cues is quite limited.

Finally, in `Ground+Opt+Det+Align`, we use dense tracking based on epipolar-guided optical flow, along with the intensity alignment based pose estimation to replace the PnP. It is seen that errors decrease for the ground truth bounding boxes in Table 1, but even more so for the actual detection bounding boxes. This clearly demonstrates that SFM cues can help 3D object localization to account for unstable detection and tracking inputs. Intuitively, SFM cues are expected to be more helpful for close objects, for which better quality 3D points can be estimated, while detection cues are more reliable in the far field. Our results are consistent with this intuition.

To further illustrate the relative benefits obtained from SFM cues, Figure 5 shows a sample output of the system from a few frames in the KITTI dataset. The green curve corresponds to distance of an object as it approaches the camera (right axis). The left axis shows the error in depth estimate for the methods `Ground+Opt` (blue curve) and `Ground+Opt+Det+Align` (red curve). The latter includes SFM cues, while the former does not. It can be seen that SFM cues are inactive when the object is further, but keep the error rate low when the object is near. On the other hand, ignoring SFM cues and relying only on object bounding boxes impacts performance in the near field.

Finally, we show the improvement in 3D localization that our use of SFM and detection cues affords for other scene understanding frameworks. We use the tracking output provided by [3] on a few KITTI sequences, along with its raw detection output based on [5]. The localization error is compared to the method of [17] that only uses an adaptive ground plane and detection bounding boxes, as well as our method that additionally incorporates 3D points and detection scores. Note that [17] performs a global optimization over all the frames, while we use only a windowed optimization. It is evident from Table 2 that our use of SFM and detection cues can also benefit other scene understanding frameworks.

An example output from our system is shown in Figure 6. Note the accuracy relative to ground truth from laser scanner.

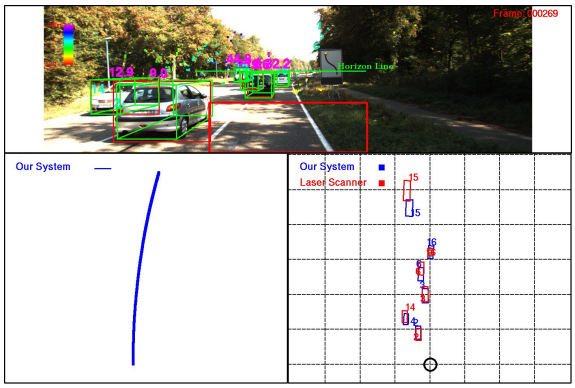

Figure 6. Output of our localization system. The bottom left panel shows the monocular SFM camera trajectory. The top panel shows input 2D bounding boxes in red, horizon from estimated ground plane and the estimated 3D bounding boxes in green with distances in magenta. The bottom right panel shows the top view of the ground truth object localization from laser scanner in red, compared to our 3D object localization in blue.

7. Discussion and Future Work

We have presented a novel framework for 3D object localization, designed for autonomous driving applications. It recognizes and exploits the complementary strengths of SFM cues (3D points and ground plane) and object cues (bounding boxes and detection scores), to achieve good localization accuracy in both near and far fields. Our system is fast and can be considered an extension of traditional bundle adjustment with object cues. The generality of our framework means it can be used to readily improve the performance of most 3D scene understanding systems that rely on object tracking. Our system uses object detection as input and a challenge for future work is to obtain this input in real-time.

Our work does have a few limitations. We assume objects are rigid bodies for computing SFM cues, which is not true for some categories such as pedestrians. Unlike some recent works [23], we do not explicitly model occlusions. Since fast operation is essential in our application, a possible solution is to use detection and tracking frameworks that are more robust to occlusions. Our future work also explores the use of our 3D object localization in autonomous driving applications that involve comprehensive scene understanding.

References

[1] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 41–48, June 2009. 3

[2] T. Brox, B. Rosenhahn, J. Gall, and D. Cremers. Combined region and motion-based 3D tracking of rigid and articulated objects. *PAMI*, 32(3):402–415, March 2010. 2

[3] W. Choi and S. Savarese. Multi-target tracking in world coordinate with single, minimally calibrated camera. In *ECCV*, pages 553–567, 2010. 2, 3, 6, 8

[4] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. Robust multiperson tracking from a mobile platform. *PAMI*, 31(10):1831–1846, 2009. 2, 3

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010. 4, 8

[6] S. Fidler, S. Dickinson, and R. Urtasun. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In *NIPS*, 2012. 3

[7] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3D traffic scene understanding from movable platforms. *PAMI*, 36(5):1012–1025, 2014. 3, 5, 6, 7

[8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, 2012. 6

[9] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for RGB-D cameras. In *ICRA*, pages 3748–3754, 2013. 2, 3, 6

[10] A. Kundu, K. M. Krishna, and C. V. Jawahar. Realtime multibody visual SLAM with a smoothly moving monocular camera. In *ICCV*, pages 2080–2087, 2011. 2

[11] T. Li, V. Kallem, D. Singaraju, and R. Vidal. Projective factorization of multiple rigid-body motions. In *CVPR*, pages 1–6, June 2007. 2

[12] K. Ozden, K. Schindler, and L. Van Gool. Simultaneous segmentation and 3D reconstruction of monocular image sequences. In *ICCV*, pages 1–8, 2007. 2

[13] K. E. Ozden, K. Schindler, and L. V. Gool. Multibody structure-from-motion in practice. *PAMI*, 32(6):1134–1141, 2010. 3

[14] B. Pepik, P. Gehler, M. Stark, and B. Schiele. 3D2PM–3D deformable part models. In *ECCV*, pages 356–370, 2012. 3

[15] K. Schindler, J. U, and H. Wang. Perspective n-view multibody structure-and-motion through model selection. In *ECCV*, volume 3951, pages 606–619, 2006. 2

[16] N. Slesareva, A. Bruhn, and J. Weickert. Optic flow goes stereo: A variational method for estimating discontinuity-preserving dense disparity maps. In *Pattern Recognition (DAGM)*, pages 33–40, 2005. 2, 3, 6

[17] S. Song and M. Chandraker. Robust scale estimation in real-time monocular SFM for autonomous driving. In *CVPR*, pages 1566–1573, 2014. 3, 7, 8

[18] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by GPU-accelerated large displacement optical flow. In *ECCV*, pages 438–451, 2010. 6

[19] A. Wedel, T. Pock, J. Braun, U. Franke, and D. Cremers. Duality tv-l1 flow with fundamental matrix prior. In *IVCNZ*, pages 1–6, Nov 2008. 3

[20] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele. Monocular visual scene understanding: Understanding multi-object traffic scenes. *PAMI*, 35(4):882–897, 2013. 2, 3

[21] K. Yamaguchi, D. McAllester, and R. Urtasun. Robust monocular epipolar flow estimation. In *CVPR*, pages 1862–1869, June 2013. 3

[22] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. In *DAGM on Pattern Recognition*, pages 214–223, 2007. 3, 6

[23] M. Zeeshan Zia, M. Stark, and K. Schindler. Are cars just 3d boxes? - jointly estimating the 3d shape of multiple objects. In *CVPR*, 2014. 3, 8