This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2469274, IEEE Transactions on Pattern Analysis and Machine Intelligence

High Accuracy Monocular SFM and Scale Correction for Autonomous Driving

Shiyu Song Manmohan Chandraker Clark C. Guest

Abstract—We present a real-time monocular visual odometry system that achieves high accuracy in real-world autonomous driving applications. First, we demonstrate robust monocular SFM that exploits multithreading to handle driving scenes with large motions and rapidly changing imagery. To correct for scale drift, we use known height of the camera from the ground plane. Our second contribution is a novel data-driven mechanism for cue combination that allows highly accurate ground plane estimation by adapting observation covariances of multiple cues, such as sparse feature matching and dense inter-frame stereo, based on their relative confidences inferred from visual data on a per-frame basis. Finally, we demonstrate extensive benchmark performance and comparisons on the challenging KITTI dataset, achieving accuracy comparable to stereo and exceeding prior monocular systems. Our SFM system is optimized to output pose within 50 ms in the worst case, while average case operation is over 30 fps. Our framework also significantly boosts the accuracy of applications like object localization that rely on the ground plane.

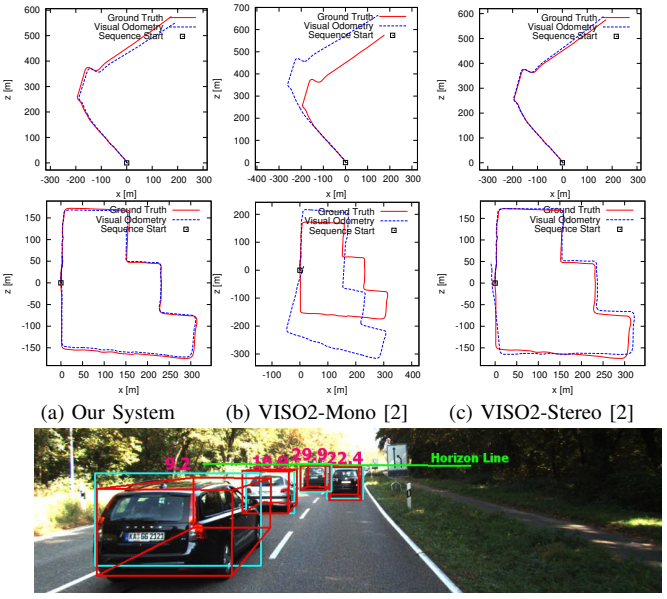
Index Terms—Monocular structure-from-motion, Scale drift, Ground plane estimation, Object localization

1 Introduction

Structure from Motion (SFM) for real-world autonomous outdoor driving is a problem that had gained immense traction in recent years. This paper presents a real-time, monocular vision-based system that relies on several innovations in multithreaded SFM for autonomous driving. It achieves outstanding accuracy in sequences spanning several kilometers of real-world environments. On the challenging KITTI dataset [1], we achieve a rotation accuracy of 0.0057 degrees per meter, even outperforming several state-of-the-art stereo systems. Our translation error is a low 2.53%, which is also competitive with stereo and outperforms previous state-of-the-art monocular systems.

While stereo SFM systems routinely achieve high accuracy and real-time performance, the challenge remains daunting for monocular ones. Yet, monocular systems are attractive for the automobile industry since they are cheaper and calibration effort is lower. Costs of consumer cameras have steadily declined in recent years, but cameras for practical SFM in automobiles are expensive since they are produced in lesser volume, must support high frame rates and be robust to extreme temperatures, weather and jitters.

The challenges of monocular visual odometry for autonomous driving are both fundamental and practical. For instance, it has been observed empirically and theoretically that forward motion with epipoles within the image is a "high error" situation for visual SFM [3]. Vehicle speeds in outdoor environments can be high, so even with high frame rate cameras, large motions may occur between consecutive frames. This places severe demands on an autonomous driving visual odometry system, necessitating extensive validation and refinement mechanisms that conventional systems do not require. Our system makes judicious use of a novel multithreaded design to ensure that location estimates become available only after extensive validation



(a) Our System (b) VISO2-Mono [2] (c) VISO2-Stereo [2]

Fig. 1: (Top row) (a) Our monocular SFM yields camera trajectories close to the ground truth over several kilometers of real-world driving. (b) Our monocular system significantly outperforms prior works that also use the ground plane for scale correction. (c) Our performance is comparable to stereo-based visual SFM. [Bottom row: Object localization] Accuracy of applications like 3D object localization that rely on the ground plane is also enhanced. The green line is the horizon from the estimated ground plane.

- *Shiyu Song and Clark C. Guest are with the Department of Electrical and Computer Engineering, University of California, San Diego, CA, 92093 USA. E-mail: shs012@ucsd.edu and cguest@ucsd.edu.*
- *Manmohan Chandraker is with NEC Labs America, Cupertino, CA 95014. E-mail: manu@nec-labs.com.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2469274, IEEE Transactions on Pattern Analysis and Machine Intelligence

with long-range constraints and thorough bundle adjustments, but without delay.

The timing requirements for visual odometry in autonomous driving are equally stringent. Thus, our system is optimized for worst-case timing scenarios, rather than the average-case optimization for most traditional systems. For instance, traditional systems may produce a spike in timings when keyframes are added, or loop closure is performed [4]. In particular, our multithreaded system produces pose outputs in at most 50 ms, regardless of whether a keyframe is added or scale correction performed. The average frame rate of our system is much higher, at above 30 fps.

Monocular vision-based frameworks are attractive due to lower cost and calibration requirements. However, the lack of a fixed stereo baseline leads to inevitable scale drift, which is a primary bottleneck that has prevented monocular visual SFM from attaining accuracy comparable to stereo. To counter scale drift, we use prior knowledge in the form of known fixed height of the camera from the ground plane. Thus, a robust and accurate estimation of the ground plane is crucial to achieve good performance. However, in real-world autonomous driving, the ground corresponds to a rapidly moving, low-textured road surface, which makes its estimation from image data challenging.

We overcome this challenge with two innovations in Sec. 5 and 6. First, we incorporate cues from multiple methods of ground plane estimation and second, we combine them in a framework that accounts for their per-frame relative confidences, using models learned from training data. While prior works have used sparse feature matching for ground plane estimation [2], [5], [6], it is demonstrably inadequate in practice and must be augmented by other cues such as the plane-guided dense stereo of Sec. 5.

Accordingly, in Sec. 5, we propose incorporating cues besides sparse 3D points, from dense stereo between successive frames and 2D detection bounding boxes (for the object localization application). The dense stereo cue vastly improves SFM, while the detection cue aids object localization. To combine cues, Sec. 6 presents a novel data-driven framework. During training, we learn models that relate the observation covariance for each cue to error behaviors of its underlying variables. For instance, the underlying variable for dense stereo may be the SAD cost. At test time, fusion of the covariances predicted by these models allows the contribution of each cue to adapt on a per-frame basis, reflecting belief in its relative accuracy. The significant improvement in ground plane estimation using our framework is demonstrated on the KITTI dataset in Sec. 7. In turn, this leads to excellent performance in applications like monocular SFM and 3D object localization.

This paper is an extension of our prior works [6], [7]. On the KITTI visual odometry training set, we achieve translation errors of 6.42%, 3.37% and 2.03% in [6], [7] and this work, respectively. For KITTI test benchmark, for which ground truth is not public, translation errors improve from 3.21% in [7] to 2.53% in this work. To achieve these improvements, additional system features are included in the monocular SFM architecture in Sec. 3.2 and 3.3 to allow uniform handling of fast and near-stationary motions in driving sequences. More intuitions behind our multithreaded design are presented in Sec. 3. We show improved results in Sec. 7.1 and 7.2, and also illustrate the failure cases of our system. Further, we provide new comparisons to previous state-of-the-art and alternate implementations in Sec. 7.4 and 7.5, as well as more explanatory figures.

2 Related Work

Stereo-based SFM systems now routinely achieve real-time performance in both indoor [4] and outdoor environments [8]. Parallel implementations for visual stereo SFM that harness the power of GPUs have been demonstrated to achieve frame rates exceeding 30 fps in indoor environments [4]. Several approaches have also been proposed that use or combine information from alternate acquisition modalities such as omnidirectional [9], ultrasound [10] or depth sensors [11].

In constrast to prior real-time SFM systems, our system architecture is intricately designed to meet the challenge of accurate and efficient monocular autonomous driving. In Section 2.1, we discuss how our design is different, better suited to the application and easily extensible.

2.1 Monocular Architectures

Early work on real-time, large-scale visual odometry includes the system of Nistér et al. that proposes both stereo and monocular systems [8]. In recent years, a few purely vision-based monocular systems have achieved good localization accuracy [12], [13], [14], [15]. For example, PTAM is an elegant two-thread architecture separating the tracking and mapping aspects [12]. It is designed for small workspace environments, focuses on 3D reconstruction and relies extensively on repeatedly observing a small set of 3D points ("loopy browsing motions"). In our testing, PTAM usually breaks down after 30 – 50 frames in datasets captured by fast forward moving vehicles, such as KITTI. In contrast, our system is designed to scale well to large outdoor environments or driving situations where scene points rapidly disappear from the field of view.

Another category is the V-SLAM system of Davison et al. based on extended Kalman Filter (EKF) [16], [17], which is improved by Handa et al. using an active matching technique based on a probabilistic framework [18]. The EKFMonoSLAM system developed by Civera et al. proposes to integrate a 1-point RANSAC within the Kalman filter that uses the available prior probabilistic information from the EKF in the RANSAC model hypothesis stage [19]. However, it is known that approaches based on bundle adjustment have better scalability [20]. A counterpoint to our system is the VISO2-M system of Geiger et al. in [2]. It relies on matching and computing relative pose between every consecutive pair of frames through a fundamental matrix estimation and uses continuous scale correction against a locally planar ground. However, it is known that two-view estimation leads to high translational errors in the case of narrow baseline forward motion [3].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2469274, IEEE Transactions on Pattern Analysis and Machine Intelligence

In Sec. 7, we compare our system with EKFMonoSLAM and VISO2-M, as well as a stereo SFM system VISO2-S [2]. The results show that our system performs comparably to stereo and outperforms prior monocular systems.

2.2 Scale Drift Correction

Successful large-scale monocular systems for autonomous navigation are uncommon, primarily due to scale drift. Strasdat et al. [21] recently proposed a monocular system that handles scale drift with loop closure. While desirable for map building, delayed scale correction from loop closure is not an option for autonomous driving. Prior knowledge of the environment is often used to counter scale drift, such as nonholonomic constraints for wheeled robots [22], or the geometry of circular pipes [23].

We use fixed height of the camera above the ground plane to handle scale drift. Several prior systems have also handled scale drift using this constraint [2], [5], [6]. However, they usually rely on triangulation or homography decomposition from feature matches that are noisy for low-textured road surfaces, or do not provide unified frameworks for including multiple cues. In contrast, we achieve superior results by combining cues from both sparse features and dense stereo, in a data-driven framework whose observation covariances are weighted by instantaneous visual data.

In contrast to most of the above systems, we present strong monocular SFM results on real-world driving benchmarks over several kilometers [1] and report accurate localization performance relative to ground truth.

2.3 Object Localization

To localize moving objects, Ozden et al. [24] and Kundu et al. [25] use simultaneous motion segmentation and SFM. A different approach is that of multi-target tracking frameworks that combine object detection with stereo [26] or monocular SFM [27], [28]. Detection can handle farther 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 [29]. Note that [26], [28] also jointly optimize the ground plane and the object position, but we incorporate more cues in ground plane estimation, and introduce an adaptive cue combination framework. We note that the utility of our accurate ground plane estimation is demonstrable for any object tracking framework, including [26], [27], [28].

3 System Architecture

Similar to prior works, a set of 3D points is initialized by relative pose estimation [30], triangulation and bundle adjustment. In normal operation, referred here as steady state, our system maintains a stable set of triangulated 3D points, which are used for estimating the camera pose at the next time instant. Unlike prior works like [12], [15] that focus on small-scale environments, outdoor applications like autonomous navigation must handle 3D scene points that rapidly move out of view within a few frames. Thus, the stable set of points used for pose computation must be continually updated, which requires a novel multithreaded architecture. The system architecture at every frame in steady state operation is illustrated in Figure 2.

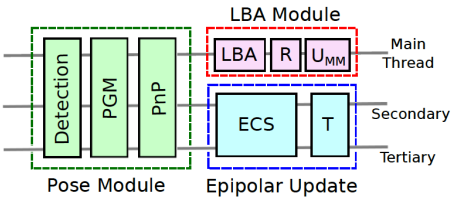


Fig. 2: System architecture for every steady state frame. The acronyms above represent PGM: Pose-guided matching, LBA: local bundle adjustment, R: re-finding, U: Update motion model, ECS: Epipolar search, T: triangulation. The modules are depicted in their multithreading arrangement, in correct synchronization order but not to scale.

3.1 Pose Module

At steady state, the system has access to a stable set of at least 100 3D points. Around 2000 FAST corners with Shi-Tomasi filtering [31] are extracted from a typical outdoor image. Similar to prior works [8], [12], we compute camera poses using pose prediction, 3D-2D matching and RANSAC-based PnP pose estimation, for which we use EPnP [32] with a model size of four points.

3.2 Epipolar Update Module

As depicted in Figure 2, our epipolar module runs at every frame. This is in contrast to its on-demand nature in prior works. The epipolar search module is parallelized across two threads and follows pose estimation at each frame. The mechanism for epipolar search is illustrated in Figure 3. Let the most recent prior keyframe be frame 0. After pose computation at frame n, for every feature f_0 in the keyframe at location (x_0, y_0), we consider a window of side $2r_e$ centered at $(x_0 + \Delta x, y_0 + \Delta y)$ in frame n, with r_e proportional to camera velocity. The introduction of the displacement $(\Delta x, \Delta y)$ is an improvement over [6], allowing the search center (x_0, y_0) to move to a more desirable position along the epipolar line. It is computed based on the distance of (x_0, y_0) from the center of the horizon, which is computed using the ground plane estimated in Sec. 5. Adapting r_e and $(\Delta x, \Delta y)$ to the velocity helps in fast highway sequences, where disparity ranges can vary significantly between far and near fields.

We consider the intersection region of this square with a rectilinear band p pixels wide, centered around the epipolar line corresponding to f_0 in frame n. The closest match is found within this intersection region. This epipolar matching procedure is also repeated by computing the closest match to f_n in frame $n-1$, call it f_{n-1}. A match is accepted only if f_{n-1} also matches f_0 - we call this circular matching and it is useful to eliminate spurious matches. Note that the matches between frames 0 and $n-1$ have already been computed at frame $n-1$. Since pose estimates are highly

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2469274, IEEE Transactions on Pattern Analysis and Machine Intelligence

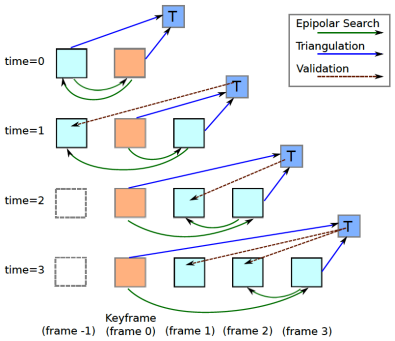


Fig. 3: Mechanism of epipolar constrained search, triangulation and validation by reprojection to existing poses. For current frame n, only 3D points that are validated against all frames 1 to $n-1$ are retained. Only persistent 3D points that survive for greater than L frames may be collected by the next keyframe.

accurate due to continuous refinement by bundle adjustment, epipolar lines are deemed accurate and we choose a stringent value of $p = 3$ to impose the epipolar constraint.

The features that are circularly matched in frame n are triangulated with respect to the most recent keyframe (frame 0). These 3D points are used as candidates ready for adding to the 3D point cloud when the system demands them at a keyframe. All candidate 3D points are continually verified by back-projecting to all the frames $1, \cdots, n-1$, and are retained only if a match is found within a tight window of side $2r_b$ pixels (we set $r_b = 3$). Working together with the local bundle adjustment in Section 3.3, this acts as a replacement for a more accurate, but expensive, multiview triangulation and is satisfactory since epipolar search produces a large number of 3D points, but only the most reliable ones may be used for pose estimation.

3.3 Local Bundle Adjustment Module

To refine camera poses and 3D points incorporating information from multiple frames, we implement a sliding window local bundle adjustment over the L most recent frames. To maintain desired frame rates and accuracy in our system, a value of $L = 10$ suffices. An improvement over [6] is the ability to handle small motions. When camera motion is small, the system prevents addition of new keyframes and forces addition of the previous keyframe in the local bundle. This guarantees that the baseline between the previous keyframe and the current frame does not become too small, which improves the stability of bundle adjustment and yields accurate pose estimates even in near-stationary situations. The vehicle speed measurement is from the SFM itself. After bundle adjustment, we give the system a chance to re-find lost 3D points using the optimized pose. An image window of radius 3 pixels is used for feature refinding.

3.4 Keyframe and Recovery

The system cannot maintain steady state indefinitely, since 3D points are gradually lost due to tracking failures or when they move out of the field of view. The latter is an important consideration in forward moving systems for autonomous driving (as opposed to browsing systems such as PTAM), so the role of keyframes is very important in keeping the system alive. The purpose of a keyframe is threefold:

- Collect 3D points with long tracks from the epipolar thread, refine them with local bundle adjustment and add to the set of stable points in the main thread.
- Trigger a bundle adjustment (we call it "keyframe bundle") that includes the recent K keyframes, to refine 3D points and keyframe poses.
- Provide the frame where new 3D points have matches.

For our application, bundle adjustment over $K = 5$ previous keyframes suffices. There are two reasons a more expensive optimization over a larger set of keyframes (or even the whole map) is not necessary to refine 3D points with long-range constraints. First, the imagery in autonomous driving applications is fast moving and does not involve repetitions, so introducing more keyframes into the bundle yields marginal benefits. Second, our goal is instantaneous pose output rather than map-building, so even keyframes are not afforded the luxury of delayed output. This is in contrast to parallel systems such as [4], where keyframes may produce a noticeable spike in per-frame timings.

On rare occasions, the system might encounter a frame where pose-guided matching fails to track features and generate enough 3D - 2D matches for PnP to work robustly (due to imaging artifacts or a sudden large motion). In such a situation, we reinitialize the system and recover the scale with 1-point RANSAC. Usually, we encounter 0-2 recovery instances per sequence in KITTI. More details on the keyframe and recovery architectures are in [6].

3.5 Discussion

We note that besides the obvious speed advantages, moving epipolar search to a new thread also greatly contributes to the accuracy and robustness of the system. A system that relies on 2D-3D correspondences might update its stable point set by performing an epipolar search only in the frame preceding a keyframe. However, the support for the 3D points introduced by this mechanism is limited to just the triplet used for the circular matching and triangulation, so the quality of those 3D points might be poor. By performing circular matching at every frame, we supply 3D points with tracks of length up to the distance from the preceding keyframe. Additionally, this also allows repeated validation and outlier rejection at every frame. Clearly, the extensively validated set of long tracks provided by the epipolar thread in our multithreaded system is far more likely to be free of outliers, while contributing longer-range constraints for a more stable pose estimation.

Our multithreaded architecture also has efficiency advantages. In our design, the epipolar module operates in parallel with the local bundle module. In contrast to large scale multithreaded bundle adjustment [33], small scale bundle (for example, with 10 views and a few hundred points) is not significantly faster with multithreading. The epipolar update module, thus, allows better 3D points while occupying the idle secondary and tertiary threads.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2469274, IEEE Transactions on Pattern Analysis and Machine Intelligence

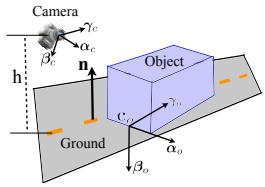


Fig. 4: Geometry of ground plane estimation. The camera height h is the distance from its optical center to ground plane. The ground plane normal is $\mathbf{n}$. Thus, the ground plane is defined by $(\mathbf{n}^\top, h)^\top$.

4 Background of Scale Correction

Scale drift correction is an integral component of monocular SFM. In practice, it is the single most important aspect that ensures accuracy. We estimate the depth and orientation of the ground plane relative to the camera for scale correction.

Multiple methods like triangulation of sparse feature matches and dense stereo between successive frames can be used to estimate the ground plane. We propose a principled approach to combine these cues to reflect our belief in the relative accuracy of each cue. Naturally, this belief should be influenced by both the input at a particular frame and observations from training data. We achieve this by learning models from extensive training data to relate the observation covariance for each cue to error behavior of its underlying variables. During testing, the error distributions at every frame adapt the data fusion observation covariances using those learned models.

4.1 Ground Plane Geometry

As shown in Fig. 4, the camera height (also called ground height) h is defined as the distance from the optical 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. For a 3D point $\mathbf{X} = (X_1, X_2, X_3)^\top$, the ground height h and the unit normal vector $\mathbf{n} = (n_1, n_2, n_3)^\top$ define the ground plane as:

$$\mathbf{n}^\top \mathbf{X} + h = 0. \tag{1}$$

4.2 Scale Correction in Monocular SFM

Scale drift correction is an integral component of monocular SFM. In practice, it is the single most important aspect that ensures accuracy. We estimate the ground plane geometry for scale correction as described in Sections 5 and 6.

Under scale drift, any estimated length l is ambiguous up to a scale factor $s = l/l^*$, where l^* is the ground truth length. The objective of scale correction is to compute s. Given the calibrated height of camera from ground h^*, computing the apparent height h yields the scale factor $s = h/h^*$. Then the camera translation $\mathbf{t}$ can be adjusted as $\mathbf{t}_{\text{new}} = \mathbf{t}/s$, thereby correcting the scale drift. In our implementation, we use either the previous frame or the previous keyframe as the origin for scale drift correction, based on the vehicle speed and the camera frame rate. In the KITTI dataset, where the frame rate is relatively slow (10 Hz), using the previous frame as the origin suffices. This happens before the system enters the local bundle adjustment step, so the corrected scale can be further optimized by the bundle adjustment.

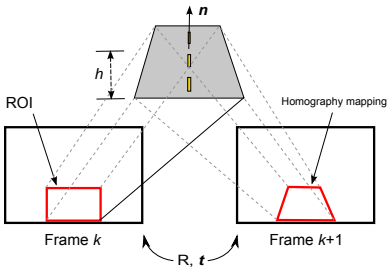


Fig. 5: Homography mapping for plane-guided dense stereo. For a hypothesized ground plane $\{\mathbf{n}, h\}$ and relative camera pose $(\mathbf{R}, \mathbf{t})$ between frames k and $k+1$, a per-pixel mapping can be computed within a region of interest (ROI) by using the homography matrix $\mathbf{G} = \mathbf{R} + h^{-1}\mathbf{t}\mathbf{n}^\top$.

5 Cues for Ground Plane Estimation

This section proposes multiple methods such as triangulation of sparse feature matches, dense stereo between successive frames and object detection bounding boxes to estimate the ground plane. In the following section, the outputs of these methods are combined in a framework that accounts for their per-frame relative effectiveness.

5.1 Plane-Guided Dense Stereo

We assume that a region of interest (ROI) in the foreground (middle fifth of the lower third of the image) corresponds to a planar ground. For a hypothesized value of $\{h, \mathbf{n}\}$ and relative camera pose $\{\mathbf{R}, \mathbf{t}\}$ between frames k and $k+1$, a per-pixel homography mapping can be computed as:

$$\mathbf{G} = \mathbf{R} + \frac{1}{h}\mathbf{t}\mathbf{n}^\top. \tag{2}$$

For KITTI's 10 Hz input frame rate, there is often little overlap of ROI between frames k and $k+2$. Conversely, baseline between frames k and $k+1$ is sufficient. For other data with 30 Hz imagery, we adapt the baseline accordingly. The homography mapping is illustrated in Figure 5. Note that $\mathbf{t}$ differs from the true translation $\mathbf{t}^*$ by an unknown scale drift factor, encoded in the h we wish to estimate. Pixels within the ROI in frame $k+1$ are mapped to frame k (subpixel accuracy is important for good performance) and the sum of absolute differences (SAD) is computed over bilinearly interpolated image intensities. A Nelder-Mead simplex routine [34] is used to estimate $\{h, \mathbf{n}\}$ as:

$$\min_{h,\mathbf{n}} (1 - \rho^{-\text{SAD}^*}), \tag{3}$$

where SAD^* denotes SAD averaged over the number of ROI pixels. We empirically choose $\rho = 1.5$ to make slices of the above cost close to bell-shaped on KITTI data, which is exploited in Sec. 6. Note that the optimization only involves h, n_1 and n_3, since $\|\mathbf{n}\| = 1$. Enforcing the norm constraint has marginal effect, since the calibration pitch is a good initialization and the cost function usually has a clear local minimum in its vicinity. The $\{h, \mathbf{n}\}$ that minimizes (3) is the estimated ground plane from the stereo cue.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2469274, IEEE Transactions on Pattern Analysis and Machine Intelligence

5.2 Triangulated 3D Points

Next, we consider matched sparse SIFT [35] descriptors between frames k and $k+1$, computed within the above region of interest (we find SIFT a better choice than ORB for the low-textured road and real-time performance is attainable for SIFT in the small ROI). To fit a plane through the triangulated 3D points, one option is to estimate $\{h, \mathbf{n}\}$ using a 3-point RANSAC for plane-fitting. However, in our experiments, better results are obtained using the method of [2], by assuming the camera pitch to be fixed from calibration. For every triangulated 3D point, the height h is computed using (1). The height difference Δh_{ij} is computed for every 3D point i with respect to every other point j. The estimated ground plane height is the height of the point i corresponding to the maximal score q, where

$$q = \max_i \left\{ \sum_{j \neq i} \exp\left(-\mu \Delta h_{ij}^2\right) \right\}, \text{ with } \mu = 50. \quad (4)$$

Note: Prior works like [5], [6] decompose the homography $\mathbf{G}$ between frames to yield the camera height [36]. However, in practice, the decomposition is very sensitive to noise, which is a severe problem since the homography is computed using noisy feature matches from the low-textured road.

5.3 Object Detection Cues

We can also use object detection bounding boxes as cues when they are available, for instance, within the object localization application. The ground plane pitch angle θ can be estimated from this cue. Recall that $n_3 = \sin\theta$, for the ground normal $\mathbf{n} = (n_1, n_2, n_3)^\top$.

Given a 2D bounding box, we can compute the 3D object height h_b through the ground plane, using (10). With a prior value $\bar{h}_b$ for object height, we obtain n_3 by solving:

$$\min_{n_3} (h_b - \bar{h}_b)^2. \quad (5)$$

The ground height h used in (10) is set to the calibration value to avoid incorporating SFM scale drift and n_1 is set to 0 since it has negligible effect on object height.

Note: Object bounding box cues provide information only on ground orientation, so their effect is negligible for SFM scale drift correction. However, for applications such as 3D localization, they provide unique long distance information, unlike dense stereo and 3D points cues that only consider an ROI close to the vehicle. An inaccurate pitch angle can lead to large errors for far objects. Thus, the 3D localization accuracy of far objects is significantly improved by incorporating this cue, as shown in Sec. **??**.

6 Adaptive Cue Combination

6.1 Data Fusion with Kalman Filter

We now propose a principled approach to combine the above cues while reflecting the per-frame relative accuracy of each. To combine estimates from various methods, a natural framework is a Kalman filter:

$$\begin{aligned} \mathbf{x}^k &= \mathbf{A}\mathbf{x}^{k-1} + \mathbf{w}^{k-1}, \quad & p(\mathbf{w}) \sim N(0, \mathbf{Q}), \\ \mathbf{z}^k &= \mathbf{H}\mathbf{x}^k + \mathbf{v}^{k-1}, & p(\mathbf{v}) \sim N(0, \mathbf{U}), \end{aligned} \quad (6)$$

where $\mathbf{x}$ and $\mathbf{z}$ are the state and observation vectors, $\mathbf{A}$ and $\mathbf{H}$ are the state and observation transition matrices, $\mathbf{w}$ and $\mathbf{v}$ are the process and observation errors. It is assumed that $\mathbf{w}$ and $\mathbf{v}$ are zero mean Gaussian distributed with the covariance $\mathbf{Q}$ and $\mathbf{U}$, respectively.

In our application, the state variable in (6) is the ground plane $\mathbf{x} = (\mathbf{n}^\top, h)^\top$. Since $\|\mathbf{n}\| = 1$, n_2 is determined by n_1 and n_3 and our observation is $\mathbf{z} = (n_1, n_3, h)^\top$. For simplicity and real-time consideration, we assume n_1, n_3 and h are independent, so $\mathbf{U} = \text{diag}(u_{n_1}, u_{n_3}, u_h)$. Thus, our state and observation transition matrix are given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}^\top, \quad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (7)$$

Suppose methods $i = 1, \cdots, m$ are used to estimate the ground plane, with corresponding observation covariances $\mathbf{U}_i = \text{diag}(u_{i,n_1}, u_{i,n_3}, u_{i,h})$. We will use the notation that $i \in \{s, p, d\}$, denoting the dense stereo, 3D points and object detection methods, respectively. Then, the fusion equations at time instant k are

$$\mathbf{U}^k = (\sum_{i=1}^{m} (\mathbf{U}_i^k)^{-1})^{-1}, \;\; \mathbf{z}^k = \mathbf{U}^k \sum_{i=1}^{m} (\mathbf{U}_i^k)^{-1} \mathbf{z}_i^k. \quad (8)$$

Naturally, the combination should be influenced by both the visual input at a particular frame and prior knowledge. Meaningful estimation of $\mathbf{U}^k$ at every frame, with the correctly proportional $\mathbf{U}_i^k$ for each cue, is essential for principled cue combination.

Usually, fixed covariances are used to combine cues, which does not account for per-frame variation in their effectiveness across a video sequence. Some methods do account for varying covariances by using auto-covariance techniques [37]. In contrast, we propose a data-driven mechanism to learn models to adapt per-frame covariances $\mathbf{U}_i^k$ for each cue, based on error distributions of certain underlying variables. These variables correspond to a physical basis for belief in accuracy of each cue (such as peakiness of SAD cost for the dense stereo cue). At test time, our learned models allow adapting each cue's observation covariance on a per-frame basis. The performance of our adaptive cue fusion is shown in Sec. 7.3.

Assuming the error behavior for a cue i is governed by an underlying variable $\mathbf{a}_i$: $p(\mathbf{v}|\mathbf{a}_i) \sim N(0, \mathbf{U}_i)$, the goal of our training procedure is to find a function that relates $\mathbf{U}_i$ and $\mathbf{a}_i$, as $\mathbf{U}_i = \mathcal{C}_i(\mathbf{a}_i)$. As we see in the following, linear functions suffice for each cue in our application.

6.2 Training

For the dense stereo and 3D points cues, we use the KITTI visual odometry dataset for training, consisting of $F = 23201$ frames. Sequences 0 to 8 of the KITTI

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2469274, IEEE Transactions on Pattern Analysis and Machine Intelligence

Algorithm 1 Data-Driven Training of Cue $i \in \{s, p, d\}$

① **for** Training frames $k = 1 : F$ **do**

② Compute the observation $\mathbf{z}_i^k$: Let f_i be the objective function for cue i. Obtain the optimal estimates $\mathbf{z}_i^k = \arg\min_{\mathbf{z}} f_i(\mathbf{z})$, as well as various samples $\dot{\mathbf{z}}_i^k$ and their function responses $f_i(\dot{\mathbf{z}}_i^k)$.

③ Compute underlying variable $\mathbf{a}_i^k$: Using the samples $\dot{\mathbf{z}}_i^k$, fit a model $\mathcal{A}_i^k$ to observations $(\dot{\mathbf{z}}_i^k, f_i(\dot{\mathbf{z}}_i^k))$. Parameters $\mathbf{a}_i^k$ of model $\mathcal{A}_i^k$ are chosen as the underlying variables to reflect belief in accuracy of cue i at frame k. (For instance, when $\mathcal{A}$ is a Gaussian, $\mathbf{a}$ can be its variance.)

④ Compute observation error v_i^k: $v_i^k = |\mathbf{z}_i^k - \mathbf{z}_i^{*k}|$, where $\mathbf{z}_i^{*k}$ is the ground truth ground plane.

⑤ **end for**

⑥ Quantize model parameters $\mathbf{a}_i^k$, for $k = 1, \cdots, F$, into L bins centered at c_i^l, for $l = 1, \cdots, L$.

⑦ For each $\mathbf{a}_i^k$, we have a corresponding error v_i^k. Let u_i^l be the variances of errors v_i^k, for k that fall within the bin l.

⑧ Fit a linear model $\mathcal{C}_i$ to observations $(\mathbf{c}_i^l, u_i^l)$.

tracking dataset are used to train the object detection cue. To determine the ground truth h and $\mathbf{n}$, we label regions of the image close to the camera that are road and fit a plane to the associated 3D points from the provided Velodyne data. No labeled road regions are used during testing.

Each method i described in Sec. 5 has an objective function f_i that can be evaluated for various positions $\dot{\mathbf{z}}$ of the ground plane variables $\mathbf{z} = \{n_1, n_3, h\}$. The functions f_i for stereo, 3D points and object cues are given by (3), (4) and (5), respectively. Then, Algorithm 1 is a description of the training, which we explain below in general terms and specifically for each cue afterwards.

Intuitively, at frame k, a model $\mathcal{A}_i^k$ to reflect the error behavior of the method i with respect to variation in ground plane parameters $\mathbf{z}$ is constructed. In our application, $i \in \{s, p, d\}$, standing for dense stereo, 3D points and detection cues, respectively. The parameters $\mathbf{a}_i^k$ of the model reflect belief in the effectiveness of cue i. Quantizing the parameters $\mathbf{a}_i^k$ from F training frames into L bins allows estimating the variance of observation error u_i^l of the samples in each bin $l = 1, \cdots, L$. The model $\mathcal{C}_i$ (a linear function) then relates these variances, u_i^l, to the underlying variables (represented by quantized parameters $\mathbf{c}_i^l$). Thus, at test time, for every frame, we can estimate the accuracy of each cue i based purely on visual data (that is, by computing $\mathbf{a}_i$) and use the model $\mathcal{C}_i$ to determine its observation variance u_i.

Now we describe the specifics for underlying variables $\mathbf{a}_i$ for each of dense stereo, 3D points and object cues. We will refer to various steps of Algorithm 1 in our description.

6.2.1 Dense Stereo

The error behavior of dense stereo between two consecutive frames is characterized by variation in SAD scores between road regions related by the homography (2), as we independently vary each variable h, n_1 and n_3. The variance of this

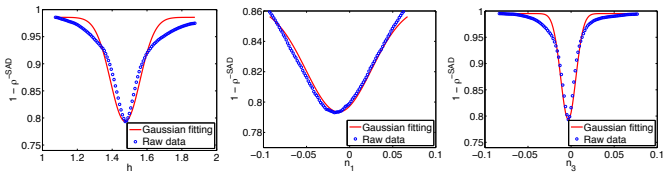


Fig. 6: Examples of 1D Gaussian fits to estimate parameters $\mathbf{a}_s^k$ for h, n_1 and n_3 of the dense stereo method respectively.

distribution of SAD scores represents the error behavior of the stereo cue and is the underlying variable $\mathbf{a}_s$.

Observation $\mathbf{z}_s$: We start at step ② in Algorithm 1. For training image k, observations $\mathbf{z}_s^k = (\widehat{n}_1^k, \widehat{n}_3^k, \widehat{h}^k)^\top$ for the ground plane are obtained using the dense stereo method, by optimizing f_s given by (3). We fix $n_1 = \widehat{n}_1^k$ and $n_3 = \widehat{n}_3^k$ and for 50 uniform samples of $\dot{h}$ in the range $[1\text{m}, 2\text{m}]$, construct homography mappings from frame k to $k+1$, according to (2) (note that $\mathbf{R}$ and $\mathbf{t}$ are already estimated by monocular SFM, up to scale). For each homography mapping, we compute the SAD score $f_s(\dot{h})$ using (3). A similar procedure applies to n_1 and n_3, with the search intervals $[-0.1, 0.1]$ for both.

Model $\mathcal{A}_s$: Following step ③ in Algorithm 1, a univariate Gaussian $\mathcal{A}_s$ is fit to the distribution of $f_s(\dot{h})$. Its variance $a_{s,h}^k$ captures the sharpness of the SAD distribution, thus, it forms the underlying variable that reflects belief in accuracy of height h estimated using dense stereo at frame k.

Note that fitting other distributions, such as Cauchy, may be also applicable. However, our intent is only to capture the sharpness of the SAD peak, for which we empirically find that a Gaussian fitting suffices. A similar procedure yields variances a_{s,n_1}^k and a_{s,n_3}^k corresponding to the orientation variables. Example fits are shown in Fig. 6.

Error v_s: Next, from step ④ in Algorithm 1, we compute $v_{s,h}^k = |\widehat{h}^k - h^{*k}|$ as the error in ground plane height relative to the ground truth h^{*k} (1.7 meters for KITTI dataset).

Model $\mathcal{C}_s$: The distributions of $a_{s,h}^k$, a_{s,n_1}^k and a_{s,n_3}^k are shown in Fig. 7 for the KITTI dataset. We quantize the parameters $a_{s,h}^k$ into $L = 100$ bins, following step ⑥ in Algorithm 1. The bin centers $c_{s,h}^l$ are positioned to match the density of $a_{s,h}^k$ (that is, we distribute F/L errors $v_{s,h}^k$ within each bin). A similar process is repeated for n_1 and n_3. We have now obtained the bin centers $\mathbf{c}_s^l$.

Next, we compute the variance $u_{s,h}^l$ of errors $v_{s,h}^k$ that fall within bin l centered at $c_{s,h}^l$ (step ⑦ in Algorithm 1). This indicates the observation error variance for the dense stereo method, corresponding to the observation variable h. We now fit a curve to the distribution of $u_{s,h}^l$ versus $c_{s,h}^l$, which provides a model to relate observation variance in h to the effectiveness of dense stereo (step ⑧ in Algorithm 1). The result is shown in Fig. 8, where each data point represents a pair of observation error covariance $u_{s,h}^l$ and parameter $c_{s,h}^l$. Empirically, we find that a straight line approximation suffices to produce a good fit. A similar process is repeated for n_1 and n_3. Thus, we have obtained linear models $\mathcal{C}_s$ (one each for h, n_1 and n_3) for the stereo method.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2469274, IEEE Transactions on Pattern Analysis and Machine Intelligence

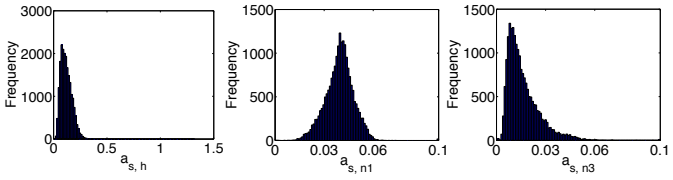


Fig. 7: Distributions of the underlying parameters $a_{s,h}$, a_{s,n_1} and a_{s,n_3} for the dense stereo cue in KITTI dataset. The parameters $\mathbf{a}_s$ roughly correspond to the peakiness of the stereo SAD cost distribution, which indicates belief in the accuracy of dense stereo.

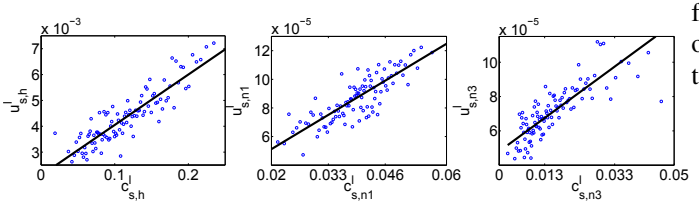


Fig. 8: Fitting a model $\mathcal{C}_s$ to relate observation variance u_s to the belief in quantized underlying parameters $\mathbf{c}_s$ of dense stereo, for h, n_1 and n_3.

6.2.2 3D Points

Similar to dense stereo, the objective of training is again to find a model $\mathcal{C}_p$ that relates the observation covariance $\mathbf{U}_p$ of the 3D points method to its underlying variables $\mathbf{a}_p$. From (4), the only estimated variable of the interest is height $\mathbf{z}_p = h$. Thus, $\mathbf{U}_p$ is given by a single variance $u_{p,h}$ and our goal is to relate it to an underlying variable $a_{p,h}$.

Observation $\mathbf{z}_p$: The optimal observation $\mathbf{z}_p^k = \hat{h}^k$ is the height of the point corresponding to the maximal score q in (4), which completes step ② in Algorithm 1.

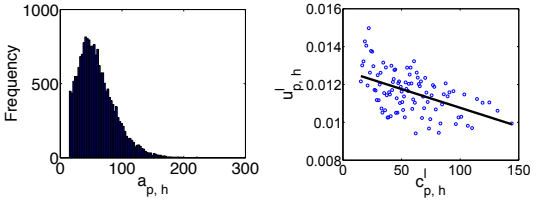
Model $\mathcal{A}_p$: We observe that the score q defined by the objective f_p in (4) is directly an indicator of belief in accuracy of the ground plane estimated using the 3D points cue. Thus, we may directly obtain the parameters $a_p^k = q^k$ (step ③ in Algorithm 1), where q^k is the optimal value of f_p at frame k, without explicitly learning a model $\mathcal{A}_p$.

Error v_p: The error $v_{p,h}^k = |\widehat{h}^k - h^{*k}|$ is computed with respect to ground truth (step ④ in Algorithm 1).

Model $\mathcal{C}_p$: The remaining procedure mirrors that for the stereo cue. The above $a_{p,h}^k$ are quantized into $L = 100$ bins centered at $c_{p,h}^l$ and the variance $u_{p,h}^l$ of the errors $v_{p,h}^k$ that fall within each bin is computed. A model $\mathcal{C}_p$ may now be fit to relate the observation variances $u_{p,h}^l$ at each bin to the corresponding quantized underlying parameter $c_{p,h}^l$. As shown in Fig. 9, a straight line fit is again reasonable.

6.2.3 Object Detection

We assume that the detector provides several candidate bounding boxes and their respective scores (for example, bounding boxes before nonmaximal suppression). A bounding box is represented by $\mathbf{b} = (x, y, w, h_b)^\top$, where x, y is its 2D position and w, h_b are its width and height. The error behavior of detection is quantified by the variation of detection scores α with respect to bounding box $\mathbf{b}$.



(a) (b)

Fig. 9: (a) Distribution of the underlying variable $a_{p,h}$ for the 3D points cue in the KITTI dataset. The parameter $a_{p,h}$ corresponds to variation in height of 3D points stemming from the ground plane, which indicates belief in accuracy of the 3D points cue. (b) Relating observation variance $u_{p,h}$ to the quantized underlying variable $c_{p,h}$.

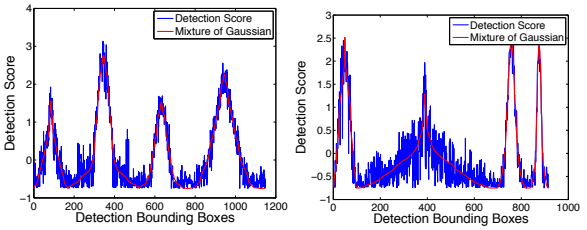


Fig. 10: Examples of mixture of Gaussians fits to detection scores. Our fitting (red) closely reflects the variation in noisy detection scores (blue). Each peak corresponds to an object.

Observation $\mathbf{z}_d$: From step ② in Algorithm 1, the ground plane pitch observation $\mathbf{z}_d = \hat{n}_3^k$ is given by solving (5).

Model $\mathcal{A}_d$: Our model $\mathcal{A}_d^k$ in Algorithm 1 is a mixture of Gaussians. At each frame, we estimate 4×4 full rank covariance matrices $\mathbf{\Sigma}_m$ centered at $\boldsymbol{\mu}_m$, as:

$$\min_{A_m, \boldsymbol{\mu_m}, \mathbf{\Sigma}_m} \sum_{n=1}^{N} \left(\sum_{m=1}^{M} A_m e^{-\frac{1}{2}\boldsymbol{\epsilon}_{mn}\mathbf{\Sigma}_m^{-1}\boldsymbol{\epsilon}_{mn}} - \alpha_n \right)^2, \quad (9)$$

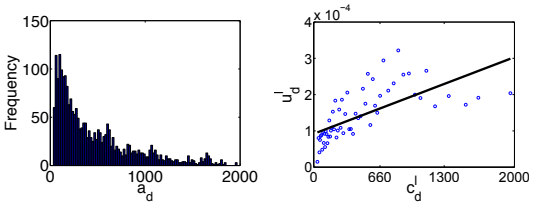
where $\boldsymbol{\epsilon}_{mn} = \mathbf{b}_n - \boldsymbol{\mu}_m$, M is number of objects and N is the number of candidate bounding boxes (the dependence on k has been suppressed for convenience). Example fitting results are shown Fig. 10. It is evident that the variation of noisy detector scores is well-captured by the model $\mathcal{A}_d^k$.

Recall that the objective f_d in (5) estimates n_3. Thus, only the entries of $\mathbf{\Sigma}_m$ corresponding to y and h_b are significant for our application. Let σ_y and σ_{h_b} be the corresponding diagonal entries of the $\mathbf{\Sigma}_m$ closest to the tracked 2D box. We combine them into a single underlying parameter, denoted $a_d^k = \frac{\sigma_y \sigma_{h_b}}{\sigma_y + \sigma_{h_b}}$, which reflects belief in the accuracy of the detection cue. This completes step ③ of Algorithm 1.

Error v_d: The error $v_{d,n_3}^k = |\widehat{n}_3^k - n_3^{*k}|$ is computed with respect to ground truth (step ④ in Algorithm 1).

Model $\mathcal{C}_d$: The remaining procedure is similar to that for the stereo and 3D points cues. The underlying parameters a_d^k are quantized and related to the corresponding variances of observation errors. The fitted linear model $\mathcal{C}_d$ that relates observation variance of the detection cue to its expected underlying parameters is shown in Fig. 11.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2469274, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) (b)

Fig. 11: (a) Distribution of the underlying variable a_{d,n_3} for the object detection cue in the KITTI dataset. The parameter a_{d,n_3} corresponds to peakiness in the distribution of object detection scores, which indicates belief in the accuracy of the object detection cue. (b) Relating observation variance u_{d,n_3} to the quantized underlying variable c_{d,n_3}.

6.3 Testing

During testing, at frame k, we fit a model $\mathcal{A}_i^k$ corresponding to each cue $i \in \{s, p, d\}$ and determine its underlying parameters $\mathbf{a}_i^k$ that convey expected accuracy. Next, we use the models $\mathcal{C}_i$ to determine the observation variances.

Dense Stereo The observation $\mathbf{z}_s^k = (\widehat{n}_1^k, \widehat{n}_3^k, \widehat{h}^k)^\top$ at frame k is obtained by minimizing f_s, given by (3). We fit 1D Gaussians to the homography-mapped SAD scores to get the values of $a_{s,h}^k$, a_{s,n_1}^k and a_{s,n_3}^k. Using the models $\mathcal{C}_s$ estimated in Fig. 8, we predict the corresponding variances u_s^k. The observation covariance for the dense stereo method is now available as $\mathbf{U}_1^k = \text{diag}(u_{s,n_1}^k, u_{s,n_3}^k, u_{s,h}^k)$.

3D Points At frame k, the observation $\mathbf{z}_p^k$ is the estimated ground height $\widehat{h}$ obtained from f_p, given by (4). The value of q^k obtained from (4) directly gives us the expected underlying parameter a_p^k. The corresponding variance $v_{p,h}^k$ is estimated from the model $\mathcal{C}_p$ of Fig. 9. The observation covariance for this cue is now available as $\mathbf{U}_p^k = u_{p,h}^k$.

Object Detection At frame k, the observation $\mathbf{z}_d^{k,m}$ is the ground pitch angle $\widehat{n}_3$ obtained by minimizing f_d, given by (5), for each object $m = 1, \cdots, M$. For each object m, we obtain the parameters $a_d^{k,m}$ after solving (9). Using the model $\mathcal{C}_d$ of Fig. 11, we predict the corresponding error variances $u_d^{k,m}$. The observation covariances for this method are now given by $\mathbf{U}_d^{k,m} = u_d^{k,m}$.

Fusion Finally, the adaptive covariance for frame k, $\mathbf{U}^k$, is computed by combining $\mathbf{U}_s^k$, $\mathbf{U}_p^k$ and the $\mathbf{U}_d^{k,m}$ from each object m. Then, our adaptive ground plane estimate $\mathbf{z}^k$ is computed by combining $\mathbf{z}_s^k$, $\mathbf{z}_p^k$ and $\mathbf{z}_d^{k,m}$, using (8).

Thus, we have described a ground plane estimation method that uses models learned from training data to adapt the relative importance of each cue – stereo, 3D points and detection bounding boxes – on a per-frame basis. A summary of the fusion framework is shown in Figure 12.

7 Experiments

We present evaluation on the KITTI dataset [1], which consists of nearly 50 km of real-world driving in 22 sequences, covering urban, residential, country and highway roads. Speeds varying from 0 to 90 kmph, a low frame rate of 10 Hz and frequent presence of other cars pose additional challenges. The evaluation metrics on KITTI are provided by [1], based on an extension of those proposed by Kümmerle et al. in [38]. Rotation and translation errors are reported as averages of all relative transformations at a fixed distance, as well as functions of various subsequence lengths and vehicle speeds. For timings, our experiments are performed on a laptop with Intel Core i7 2.40 GHz processor with 8GB DDR3 RAM and 6M cache. The main modules occupy three threads as depicted in Sec. 3, while ground plane estimation occupies two threads of its own.

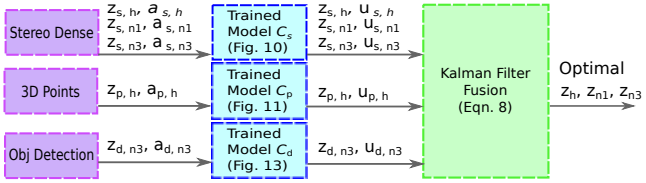
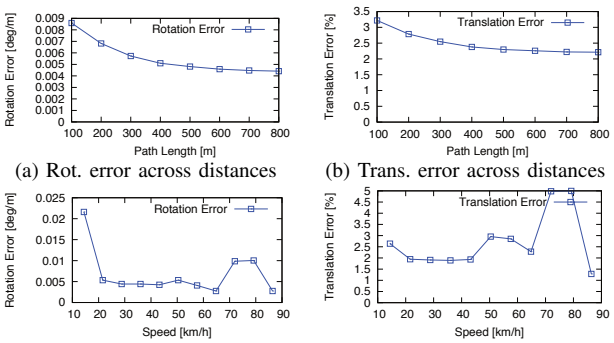


Fig. 12: Summary of adaptive cue combination. For the ground plane estimation variables $\mathbf{z}_{n_1}$, $\mathbf{z}_{n_3}$ and $\mathbf{z}_h$, the corresponding observations from individual methods $i \in \{s, p, d\}$ are given by $\mathbf{z}_{i,n_1}$, $\mathbf{z}_{i,n_3}$ and $\mathbf{z}_{i,h}$. Underlying variables $\mathbf{a}_i$ allow inference of variances u_i for adaptive fusion.



(a) Rot. error across distances (b) Trans. error across distances

(c) Rot. error across speeds (d) Trans. error across speeds

Fig. 13: SFM results on the KITTI benchmark, for rotation and translation errors over various distances and speeds.

In consideration of real-time performance, only the dense stereo and 3D points cues are used for monocular SFM. Detection bounding box cues are used for the object localization application where they are available. Note that when detection cues are available, they only improve ground orientation, without any adverse effects on the SFM. Object localization is demonstrated using object detection and tracked bounding boxes computed offline using [39].

7.1 Benchmark Monocular SFM on KITTI

The visual odometry test sequences in KITTI are numbered 11–21, for which ground truth is not public. Our system's performance for these sequences is accessible from the evaluation webpage [40], under the name **MLM-SFM**. Figure 13 shows the performance of our system, with average rotation and translation errors reported over various subsequence lengths and speeds. As on August 1 2014, our method ranks first among monocular systems and sixteenth overall (including stereo and laser-point systems).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2469274, IEEE Transactions on Pattern Analysis and Machine Intelligence

Fig. 14: Example frames from Seq 01 and 07 with repeated features and serious interference from obstacles. These are situations that our system, which relies purely on SFM, is not designed to handle.

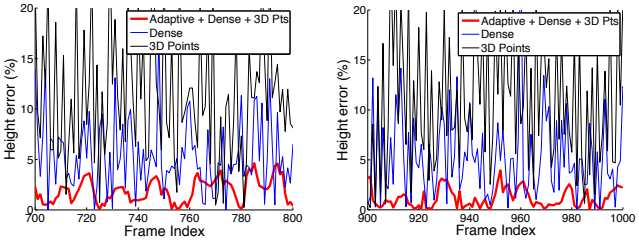
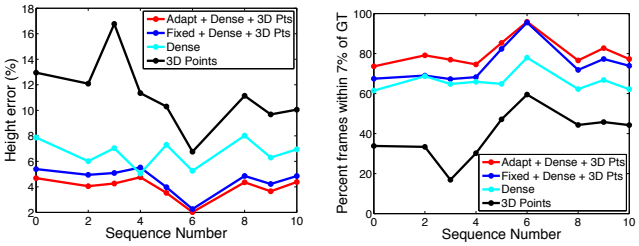


Fig. 15: Height error relative to ground truth over Seq 2 and Seq 5. The effectiveness of our data fusion is shown by less spikiness in the filter output and a far lower error.



(a) Height error (b) Success rate

Fig. 16: Error and robustness of our ground plane estimation. (a) Average error in ground plane estimation across Seq 0-10. (b) Percent number of frames where height error is less than 7%. Note that the error in our method is far lower and the robustness far higher than either method on its own.

7.2 Accuracy and Robustness of Monocular SFM

Another benefit of our ground plane estimation is enhanced robustness. As demonstration, we run 50 trials of our system on Seq $0-10$, as well as stereo and monocular systems associated with the dataset, VISO2-S and VISO2-M [2]. Errors relative to ground truth are computed using the metrics in [1]. Average errors over Seq 0–10 are shown in Table 1. Note our vast performance improvement over VISO2-M, as well as rotation and translation errors better than even the stereo system VISO2-S. All the methods encounter very high errors for sequences 1 and 7, which are not considered in this evaluation. The former is an extended highway sequence at speeds of 90 kmph with repeated textures, while the latter has a segment where a large truck occludes over 70% of the image (see Figure 14). Our monocular SFM system currently relies only on low-level features, however, our future work integrates lane and object detection which can allow handling such scenarios.

In Figure 19, we show the reconstructed trajectories from our monocular SFM with adaptive ground plane estimation, the monocular system of VISO2-M and the stereo system VISO2-S [2], for eight other KITTI sequences besides the two shown in Figure 1. All the trajectories are shown in blue, compared to ground truth shown in red. Note the high accuracy of our monocular system relative to ground truth, comparable to a stereo system and far more accurate than the prior monocular works. Also notice that our rotation error is lower than stereo, which has significant impact on long-range location error. This performance is enabled by our system architectural innovations and ground plane estimation, which combines multiple cues and adapts their relative weights to reflect per-frame uncertainties in visual data using models learned from training data.

7.3 Accuracy of Ground Plane Estimation

Ground plane estimation that combines cues in a rigorous Kalman filter and adaptively computes fusion covariances is key to achieving our robust performance. Fig. 15 shows examples of error in ground plane height relative to ground truth using 3D points and stereo cues individually, as well as the output of our combination. Note that while individual methods are very noisy, our cue combination allows a much more accurate estimation than either.

Next, we demonstrate the advantage of cue combination using the data-driven framework of Sec. 6 that uses adaptive covariances, as opposed to a traditional Kalman filter with fixed covariances. For this experiment, the fixed covariance for the Kalman filter is determined by the error variances of each variable over the entire training set (we verify by cross-validation that this is a good choice).

In Fig. 16, using only sparse feature matches causes clearly poor performance (black curve). The dense stereo performs better (cyan curve). Including the additional dense stereo cue within a Kalman filter with fixed covariances leads to an improvement (blue curve). However, using the training mechanism of Sec. 6 to adjust per-frame observation covariances in accordance with the relative confidence of each cue leads to a further reduction in error by nearly 1% (red curve). Fig. 16(b) shows that we achieve the correct scale at a rate of $75-100\%$ across all sequences, far higher than the other methods.

In particular, compare our output (red curves) to that of only 3D points (black curves). This represents the improvement by this paper over prior works like [2], [5], [6] that use only sparse feature matches from the road surface.

7.4 Effectiveness of Ground Plane Estimation

In this section, we demonstrate the effectiveness of our ground plane estimation by integrating with another publicly available monocular SFM system, VISO2-M. It relies on computing relative pose between all consecutive pairs of frames through a fundamental matrix estimation and uses continuous scale correction against a locally planar ground. This system architecture has the advantage of simplicity and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2469274, IEEE Transactions on Pattern Analysis and Machine Intelligence

Seq	Frms	VISO2-S (Stereo)				VISO2-M (Monocular)				Our Results (Monocular)			
		Rot (deg/m)	σ_R^2	Trans (%)	σ_T^2	Rot (deg/m)	σ_R^2	Trans (%)	σ_T^2	Rot (deg/m)	σ_R^2	Trans (%)	σ_T^2
0	4540	0.0109	7.1E-08	2.32	3.2E-03	0.0209	3.0E-06	11.91	8.9E-02	0.0048	1.1E-06	2.04	1.1E-01
2	4660	0.0074	3.3E-08	2.01	1.7E-03	0.0114	3.2E-07	3.33	1.6E-02	0.0035	5.7E-08	1.50	1.4E-02
3	800	0.0107	2.3E-07	2.32	1.3E-02	0.0197	6.8E-06	10.66	5.2E-01	0.0021	1.1E-07	3.37	2.9E-01
4	270	0.0081	8.8E-07	0.99	3.0E-03	0.0093	3.1E-06	7.40	9.6E-03	0.0023	4.9E-07	1.43	3.7E-01
5	2760	0.0098	3.8E-08	1.78	1.9E-03	0.0328	3.2E-06	12.67	1.1E-01	0.0038	6.6E-06	2.19	2.0E-01
6	1100	0.0072	1.6E-07	1.17	4.7E-03	0.0157	1.3E-06	4.74	1.0E-01	0.0081	1.9E-05	2.09	6.9E-01
8	4070	0.0104	6.6E-08	2.35	3.8E-03	0.0203	1.1E-06	13.94	1.0E-01	0.0044	3.8E-07	2.37	5.3E-02
9	1590	0.0094	1.6E-07	2.36	7.2E-03	0.0143	2.3E-06	4.04	8.4E-02	0.0047	5.8E-07	1.76	3.3E-02
10	1200	0.0086	4.4E-07	1.37	1.1E-02	0.0388	1.3E-05	25.20	3.2E+00	0.0085	1.0E-04	2.12	1.3E+00
Avg		0.0094		2.06		0.0203		10.18		0.0045		2.03	

TABLE 1: Comparison of rotation and translation errors for our system versus other state-of-the-art stereo and monocular systems. The values reported are statistics over 50 trials and demonstrate the robustness of our system. Note that our translation and rotation errors are lower than stereo VISO2-S, and much better than VISO2-M.

Seq	Frms	VISO2-M		VISO2-M + Our GP	
		Rot (deg/m)	Trans (%)	Rot (deg/m)	Trans (%)
0	4540	0.0209	11.9	0.0206	6.57
2	4660	0.0114	3.33	0.0114	2.73
3	800	0.0197	10.7	0.0192	5.67
4	270	0.0093	7.40	0.0087	1.49
5	2760	0.0328	12.7	0.0333	7.63
6	1100	0.0157	4.74	0.0156	4.47
8	4070	0.0203	13.9	0.0203	6.64
9	1590	0.0143	4.04	0.0145	3.04
10	1200	0.0388	25.2	0.0379	21.3
Avg		0.0204	10.43	0.0202	6.62

TABLE 2: The effectiveness of our ground plane estimation is demonstrated by replacing VISO2-M's ground plane estimation module with ours. The new method "VISO2-M + Our GP" achieves over 4% better translation error.

Seq	EKFMonoSLAM		Our SFM + no GP		Our System	
	Rot (deg/m)	Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)	Trans (%)
3	0.014	16.4	0.002	9.66	0.002	3.37
4	0.010	11.6	0.003	2.40	0.002	1.43
6	0.040	27.0	0.013	14.4	0.008	2.09
Avg	0.027	21.2	0.008	11.2	0.005	2.48

TABLE 3: The effectiveness of our monocular SFM architecture is demonstrated by comparing the raw SFM performance (without the scale correction of our ground plane estimation) with the state-of-the-art SFM system, EKFMonoSLAM [19]. Our raw translation error is 10% better than EKFMonoSLAM.

robustness. Theoretically, it can not break down, since it does not intend to build long feature tracks, but the resulting disadvantage of low accuracy has been shown in Section 7.2. In this section, we show that our ground plane estimation can significantly improve accuracy of VISO2-M, demonstrating our potential to improve other monocular SFM systems.

We replaced VISO2-M's ground plane estimation with ours (Sec. 5 and 6), keeping everything else the same. KITTI training dataset are used for testing. Again, errors relative to ground truth are computed using the metrics in [1]. Error rates using the KITTI training dataset are shown in Table 2. The method replacing VISO2-M's ground plane estimation with ours is under the name "VISO2-M + Our GP" (right column). Note the translation error improves from 10.43% to 6.62%. Comparing the errors for "VISO2-M + Our GP" with our system's from Table 1 demonstrates the effectiveness of our monocular system architecture as well. The performance of "VISO2-M + Our GP" for KITTI test dataset is accessible from the KITTI evaluation webpage [40], under the name **VISO2-M + GP**. The translation error improves from 11.94% for the original method **VISO2-M** to 7.46% for ours.

7.5 Effectiveness of Our SFM Architecture

To further demonstrate the effectiveness of our monocular SFM architecture discussed in Section 3, we compare our raw SFM performance (without the scale correction of our ground plane estimation) with another well-known SFM system, EKFMonoSLAM [19]. KITTI odometry dataset training sequences 00 - 10 and the metrics in [1] are again used. However, EKFMonoSLAM only successfully finishes three relatively short sequences 03, 04 and 06. The error rates are shown in Table 3. The middle column "Our SFM + no GP" shows the error numbers of our system without enabling the scale drift correction based on the ground plane estimation of Sec. 5 and 6. The translation error is higher compared to our full system in the third column, but it is still 10% better than EKFMonoSLAM.

Additionally, we provide experimental comparisons to demonstrate the effectiveness of the feature matching mechanism proposed in Section 3.2. We compare the SFM accuracy of our system using the proposed method against the more common used chain matching (match features in frame i to $i+1$, $i+1$ to $i+2$ and so on), with all other system components kept the same. As summarized in Table 4, for KITTI training set, our method reduces the rotation error to nearly half and the translation error by nearly 1%.

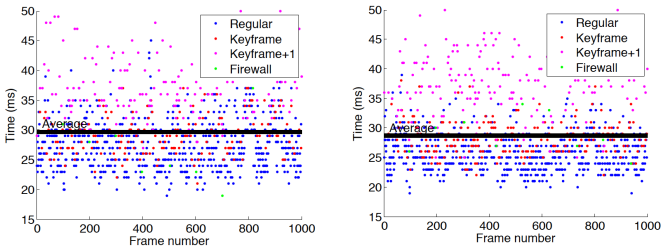
7.6 Real-time Performance

To illustrate our assertion that the system returns real-time pose at an average of 30 fps and a worst-case timing of 50 ms per frame, Figure 17 provides the timing graphs of the system on two sequences. In particular, note that the insertion of keyframes, triggering bundle adjustments or error-correcting mechanisms do not result in significant spikes in our timings, which is in contrast to several contemporary real-time systems.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2469274, IEEE Transactions on Pattern Analysis and Machine Intelligence

Seq	Frms	Chain Matching Rot (deg/m)	Chain Matching Trans (%)	Proposed Method Rot (deg/m)	Proposed Method Trans (%)
0	4540	0.0198	5.42	0.0048	2.04
2	4660	0.0045	1.99	0.0035	1.50
3	800	0.0026	3.58	0.0021	3.37
4	270	0.0022	0.63	0.0023	1.43
5	2760	0.0046	2.65	0.0038	2.19
6	1100	0.0133	3.10	0.0081	2.09
8	4070	0.0042	2.31	0.0044	2.37
9	1590	0.0064	1.48	0.0047	1.76
10	1200	0.0040	2.40	0.0085	2.12
Avg		0.0087	3.02	0.0045	2.03

TABLE 4: The effectiveness of the feature matching mechanism in Section 3.2 is demonstrated by comparing the SFM performance using the proposed method against the more commonly used chain matching method.



(a) Sequence 05 (b) Sequence 08

Fig. 17: The runtimes of our system for various types of frames. Blue denotes steady state frame, red denotes a keyframe, magenta the frame after a keyframe and green denotes a firewall insertion. The black line is the average time per frame, which correspondes to 33.7 fps for sequence 05 and 34.9 fps for sequence 08.

It can also be observed that keyframes are inserted once in about 5 and 6 frames for sequences 08 and 05, respectively. This is expected since a fast moving vehicle will demand new 3D points from the epipolar update module at frequent intervals. It does not affect the performance of our system since the keyframe bundle adjustment triggered after a keyframe finishes before the next frame's pose computation and runs in parallel to it. In fact, keyframe insertion is an opportunity to introduce long-range constraints in the optimization (so long as the epipolar update module can return long enough tracks). Thus, to ensure speed and accuracy, it is crucial for a multithreaded SFM system to not only have a well-designed keyframe architecture, but also to have its various modules like pose estimation, epipolar search and various bundle adjustments operating in optimal conjunction with each other.

7.6.1 Background

Let $\mathbf{K}$ be the camera intrinsic calibration matrix. As [26], [27], [28], 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 $\{h, \mathbf{n}\}$:

$$\mathbf{B} = (B_x, B_y, B_z)^\top = -\frac{h\mathbf{K}^{-1}\mathbf{b}}{\mathbf{n}^\top\mathbf{K}^{-1}\mathbf{b}}, \qquad (10)$$

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

Given 2D object tracks, one may estimate best-fit 3D bounding boxes. The object pitch and roll are determined by the ground plane (see Fig. 4). For a vehicle, the initial yaw angle is assumed to be its direction of motion and a prior is imposed on the ratio of its length and width. Given an initial position from (10), a 3D bounding box can be computed by minimizing the difference between its reprojection and the tracked 2D bounding box.

A detailed description of monocular object localization is beyond the scope of this paper [41]. Here, we simply note two points. First, an accurate ground plane is clearly the key to accurate monocular localization, regardless of the actual localization framework. Second, incorporating cues from detection bounding boxes into the ground plane estimation constitutes an elegant feedback mechanism between SFM and object localization.

7.6.2 Accuracy of 3D Object Localization

Now we demonstrate the benefit of the adaptive ground plane estimation of Sec. 6 for 3D object localization. KITTI does not provide a localization benchmark, so we instead use the tracking training dataset to evaluate against ground truth. We use Seq 1-8 for training and Seq 9-20 for testing. The metric we use for evaluation is percentage error in object position. For illustration, we consider only the vehicle objects and divide them into "close" and "distant", where distant objects are farther than 10m. We discard any objects that are not on the road. Candidate bounding boxes for training the object detection cue are obtained from [42].

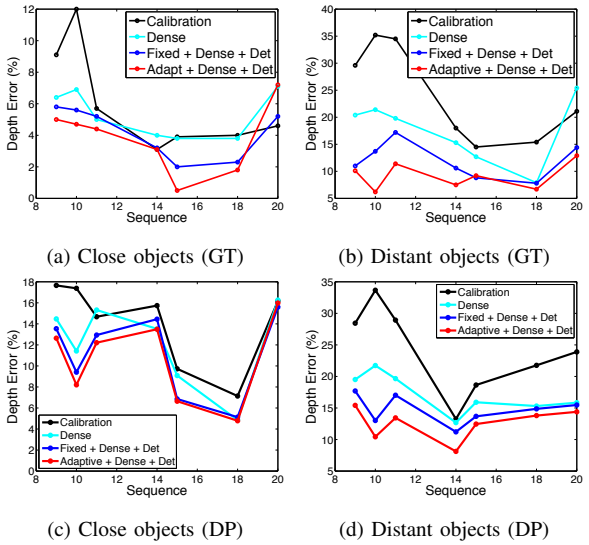
Fig. 18 compares object localization using the ground plane from our data-driven cue combination (red curve), as opposed to one estimated using fixed covariances (blue), or one that is fixed from calibration (black). The top row uses ground truth object tracks, while the bottom row uses tracks from the tracker of [39]. For each case, observe the significant improvement in localization using our cue combination. Also, from Figs. 18(b),(d), observe the significant reduction in localization error by incorporating the detection cue for ground plane estimation for distant objects. Fig. 1 shows an example of our localization output.

8 Conclusions

We have presented a novel multithreaded real-time monocular SFM that achieves outstanding accuracy in real-world autonomous driving. We demonstrate that judicious multithreading can boost both the speed and accuracy for handling challenging road conditions. The system is optimized to provide pose output in real-time at every frame, without delays for keyframe insertion or refinement.

We have demonstrated that accurate ground plane estimation allows monocular vision-based systems to achieve high accuracy and robustness. In particular, we have shown that it is beneficial to include multiple cues and proposed a data-driven mechanism to combine those cues in a framework that reflects their per-frame relative confidences. We showed that including dense stereo cues besides sparse 3D points improves monocular SFM performance through robust scale

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2469274, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) Close objects (GT) (b) Distant objects (GT)

(c) Close objects (DP) (d) Distant objects (DP)

Fig. 18: Comparison of 3D object localization errors for calibrated ground, stereo cue only, fixed covariance fusion and adaptive covariance fusion of stereo and detection cues. (Top row) Using object tracks from ground truth (Bottom row) Using object tracks from [39]. Errors reduce significantly for adaptive cue fusion, especially for distant object where detection cue is more useful.

drift correction, while further inclusion of object bounding box cues improves the accuracy of 3D object localization.

Our robust and accurate scale correction is a significant step in bridging the gap between monocular and stereo SFM. We believe this has great benefits for autonomous driving applications. Our future work will use the proposed monocular SFM and object localization towards real-time applications such as collision avoidance, scene recognition and drivable path planning.

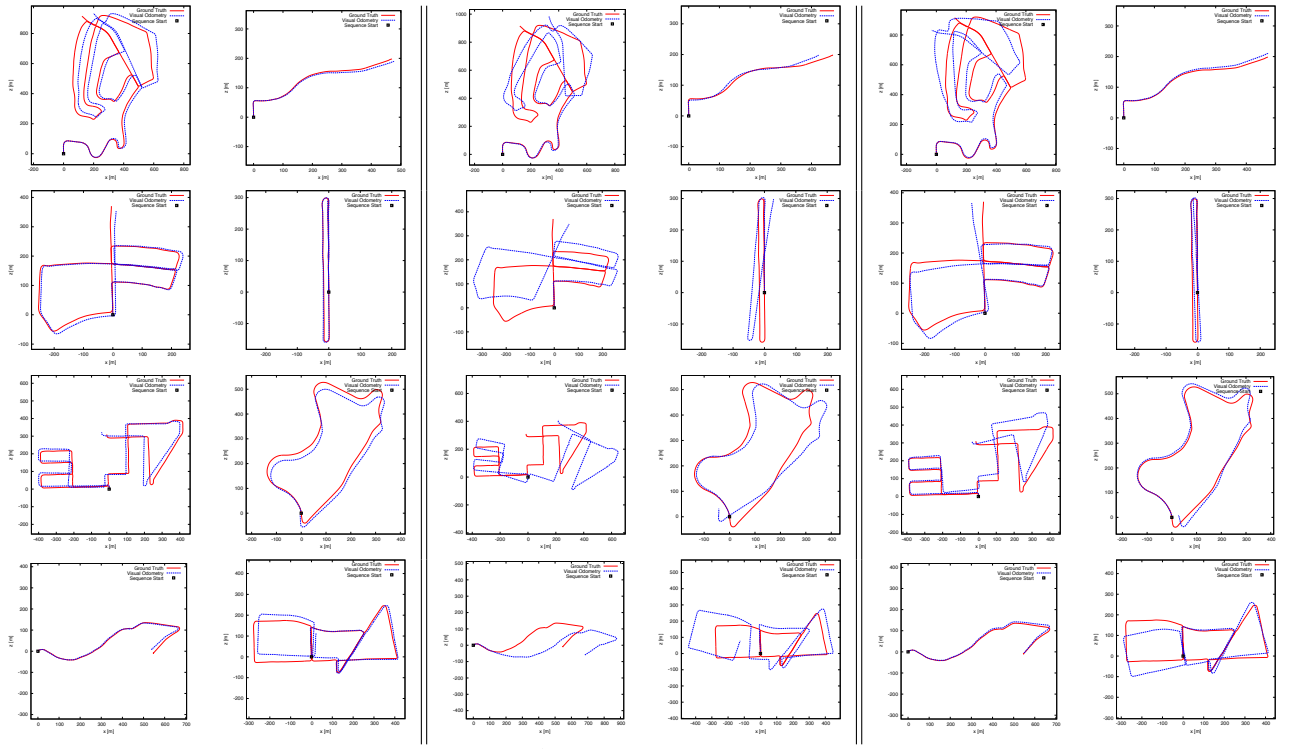
ACKNOWLEDGMENTS

This research was conducted during the first author's internship at NEC Labs America.

REFERENCES

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *CVPR*, 2012.
[2] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3D reconstruction in real-time," in *IEEE Int. Veh. Symp.*, 2011.
[3] J. Oliensis, "The least-squares error for structure from infinitesimal motion," *IJCV*, vol. 61, no. 3, pp. 259–299, 2005.
[4] B. Clipp, J. Lim, J.-M. Frahm, and M. Pollefeys, "Parallel, real-time visual SLAM," in *IROS*, 2010, pp. 3961–3968.
[5] D. Scaramuzza and R. Siegwart, "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles," *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 1015–1026, 2008.
[6] S. Song, M. Chandraker, and C. C. Guest, "Parallel, real-time monocular visual odometry," in *ICRA*, 2013, pp. 4698–4705.
[7] S. Song and M. Chandraker, "Robust scale estimation in real-time monocular SFM for autonomous driving," in *CVPR*, 2014, pp. 1566–1573.
[8] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *CVPR*, 2004, pp. 652–659.
[9] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis, "Monocular visual odometry in urban environments using an omnidirectional camera," in *IROS*, 2008.
[10] S. Ahn, J. Choi, N. L. Doh, and W. K. Chung, "A practical approach for EKF-SLAM in an indoor environment: fusing ultrasonic sensors and stereo camera," *Auto. Robots*, vol. 24, no. 3, pp. 315–335, 2008.
[11] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *ICRA*, 2012.
[12] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *ISMAR*, 2007.
[13] ——, "Improving the agility of keyframe-based SLAM," in *ECCV*, 2008.
[14] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart, "Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments," in *ICRA*, 2011, pp. 3056–3063.
[15] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular SLAM-based navigation for autonomous micro helicopters in GPS-denied environments," *J. Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011.
[16] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *ICCV*, Oct 2003, pp. 1403–1410.
[17] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *PAMI*, vol. 29, no. 6, pp. 1052–1067, 2007.
[18] A. Handa, M. Chli, H. Strasdat, and A. Davison, "Scalable active matching," in *CVPR*, June 2010, pp. 1546–1553.
[19] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-point RANSAC for Extended Kalman Filtering: Application to real-time structure from motion and visual odometry," *J. Field Robotics*, vol. 27, no. 5, pp. 609–631, Sep. 2010.
[20] H. Strasdat, J. Montiel, and A. J. Davison, "Visual SLAM: Why filter?" *IVC*, vol. 30, no. 2, pp. 65 – 77, 2012.
[21] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular SLAM," in *Robotics: Sci. and Sys.*, 2010.
[22] D. Scaramuzza, F. Fraundorfer, M. Pollefeys, and R. Siegwart, "Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints," in *ICCV*, 2009, pp. 1413–1419.
[23] P. Hansen, H. S. Alismail, P. Rander, and B. Browning, "Monocular visual odometry for robot localization in LNG pipes," in *ICRA*, 2011.
[24] K. Ozden, K. Schindler, and L. Van Gool, "Simultaneous segmentation and 3D reconstruction of monocular image sequences," in *ICCV*, 2007, pp. 1–8.
[25] A. Kundu, K. M. Krishna, and C. V. Jawahar, "Realtime multibody visual SLAM with a smoothly moving monocular camera," in *ICCV*, 2011, pp. 2080–2087.
[26] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "Robust multiperson tracking from a mobile platform," *PAMI*, vol. 31, no. 10, pp. 1831–1846, 2009.
[27] W. Choi and S. Savarese, "Multi-target tracking in world coordinate with single, minimally calibrated camera," in *ECCV*, 2010, pp. 553–567.
[28] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele, "Monocular visual scene understanding: Understanding multi-object traffic scenes," *PAMI*, vol. 35, no. 4, pp. 882–897, 2013.
[29] K. E. Ozden, K. Schindler, and L. V. Gool, "Multibody structure-from-motion in practice," *PAMI*, vol. 32, no. 6, pp. 1134–1141, 2010.
[30] D. Nistér, "An efficient solution to the five-point relative pose problem," *PAMI*, vol. 26, no. 6, pp. 756–777, 2004.
[31] J. Shi and C. Tomasi, "Good features to track," in *CVPR*, 1994, pp. 593–600.
[32] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *IJCV*, vol. 81, no. 2, pp. 155–166, 2009.
[33] C. Wu, S. Agarwal, B. Curless, and S. Seitz, "Multicore bundle adjustment," in *CVPR*, June 2011, pp. 3057–3064.
[34] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.
[35] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.
[36] O. D. Faugeras and F. Lustman, "Motion and Structure From Motion in a Piecewise Planar Environment," *Pat. Rec. AI*, vol. 2, no. 3, pp. 485–508, 1988.
[37] B. J. Odelson, M. R. Rajamani, and J. B. Rawlings, "A new autocovariance least-squares method for estimating noise covariances," *Automatica*, vol. 42, no. 2, pp. 303 – 308, 2006.
[38] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, "On measuring the accuracy of SLAM algorithms," *Auto. Robots*, vol. 27, no. 4, pp. 387–407, 2009.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2469274, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) Our System (b) VISO2-Mono [2] (c) VISO2-Stereo [2]

Fig. 19: Reconstructed trajectories from sequences in the KITTI training dataset (with ground truth). Also see trajectories in Figure 1 and the accompanying video. (a) Our monocular SFM yields camera trajectories close to the ground truth over several kilometers of real-world driving. (b) Our monocular SFM significantly outperforms prior works that also use the ground plane for scale correction. (c) Our performance is comparable to stereo SFM.

[39] H. Pirsiavash, D. Ramanan, and C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *CVPR*, 2011.
[40] A. Geiger, P. Lenz, and R. Urtasun, "The KITTI vision benchmark suite," 2012. [Online]. Available: www.cvlibs.net/datasets/kitti/eval_odometry.php
[41] S. Song and M. Chandraker, "Joint SFM and detection cues for monocular 3d localization in road scenes," in *CVPR*, June 2015.
[42] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *PAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

Shiyu Song Shiyu is a Ph.D. Candidate from University of California, San Diego. He received his M.S. degree at Dept. of Electrical and Computer Engineering at University of California, San Diego in 2010 and his B.S. degree in Dept. of Electrical Engineering at Tsinghua University in 2008. Shiyu's research interests include computer vision, structure from motion, simultaneous localization and mapping and 3D reconstruction. Shiyu did three internships at NEC Laboratories America from 2012 to 2013. Before that, he was a research assistant at the San Diego Supercomputer Center in 2011 and at UC San Diego from 2009 to 2011.

Manmohan Chandraker received a B.Tech. in Electrical Engineering at the Indian Institute of Technology, Bombay and a PhD in Computer Science at the University of California, San Diego. Following a postdoctoral scholarship at the University of California, Berkeley, he joined NEC Labs America in Cupertino, where he conducts research in computer vision. His principal research interests are 3D reconstruction, 3D scene understanding, global light transport, structure from motion and optimization methods, with applications in autonomous driving, robotics, HCI and graphics-based vision. His works have received the Marr Prize Honorable Mention for Best Paper at ICCV 2007, the 2009 CSE Dissertation Award for Best Thesis at UC San Diego, a nomination for the 2010 ACM Dissertation Award, an IEEE PAMI special issue on Best Papers of CVPR 2011 and the Best Paper Award at CVPR 2014.

Clark C. Guest Clark C. Guest received his BS and MEE degrees in electrical engineering from Rice University in 1975 and 1976, respectively. His doctorate, also in electrical engineering, was awarded by the Georgia Institute of Technology in 1983. He has been on the faculty of the Electrical and Computer Engineering Department at the University of California San Diego since 1984. His research interests include computer vision, machine learning, and intelligent optical systems. From 2005 through 2007, he participated on the Axion Racing DARPA Grand Challenge Team. He provided computer vision and navigational intelligence for the team's autonomous robotic vehicle. He is also a past president of the Optical Society of San Diego.