2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

Object-Aware Bundle Adjustment For Correcting Monocular Scale Drift

Duncan P. Frost, Olaf Kähler and David W. Murray

Abstract— **Without knowledge of the absolute baseline between images, the scale of a map from single-camera simultaneous localization and mapping system is subject to calamitous drift over time. We describe a monocular approach that in addition to point measurements also considers object detections to resolve this scale ambiguity and drift. By placing a prior on the size of the objects, the scale estimation can be seamlessly integrated into a bundle adjustment. When object observations are available, the local scale of the map is then determined jointly with the camera pose in local adjustments. Unlike many previous visual odometry methods, our approach does not impose restrictions such as approximately constant camera height or planar roadways, and is therefore applicable to a much wider range of applications. We evaluate our approach on the KITTI dataset and show that it reduces scale drift over long-range outdoor sequences with a total length of 40 km. Qualitative evaluation is also performed on video footage from a hand-held camera.**

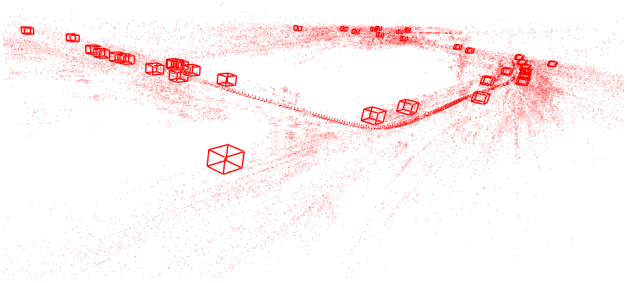
I. Introduction

Simultaneous localization and mapping involves the estimation of a sensor's spatial environment while determining its pose within it. Pioneered in [1], [2], SLAM remains most developed in mobile robotics, where LIDAR and large baseline stereo rigs measure absolute depth and allow mapping of trajectories of kilometres in length [3], [4], [5]. Deploying SLAM for augmented reality on mobiles and wearables imposes much tighter size and power budgets, making the pursuit of single-camera SLAM an imperative [6], [7], [8], [9]. However, if distance and speed are not measured, single-camera reconstructions are subject to a global scale ambiguity. Even if the ambiguity is fixed, say by specifying the scale of the camera's first movements, this merely ensures the map is appropriately scaled in that initial area: when the camera moves further away the process is subject to drift and the map's scale and camera's speed become ever more error-prone [10], [11].

All SLAM systems accumulate error in their estimates, which can be redistributed and reduced on loop-closure [12], [10], [13], [14]. However, scale drift in monocular methods is often so severe that even searching for closure is infeasible, leading to complete mapping failure. Some studies (*e.g.* [15]) also report difficulty rectifying scale drift some way into the process of map building, indicating that the chosen methods of accounting for and distributing error are irreversible.

Contrast this with the case of a cyclopean human observer [16]. With one eye open, we draw on a wealth of prior scene knowledge to stabilize scale as we move. This paper leverages analogous computational capacities to detect objects from classes of known size and to use their size in bundle adjustment to counter scale drift. We acknowledge both computational and psychophysical observations by applying drift correction frequently in local adjustments rather than occasionally in a more global correction.

(a)



(b)

Fig. 1. Example tracking frame (a) and map (b) generated by our method on a sequence from the KITTI dataset. Localized objects are shown as cubes.

The key contribution of this paper is the introduction of an elegant and computationally inexpensive modification to bundle adjustment that relies on minimal prior information to correct scale drift in long-range monocular SLAM sequences. We will first relate our contribution to existing methods in Section II. In Sections III and IV we will then describe the method and its implementation framework. We evaluate our proposed system in Section V by deploying it on extended sequences drawn from the KITTI dataset and a hand-held video sequence. Conclusions and possible future work are presented in Section VI.

II. Related Work

One method of mitigating scale uncertainly is to employ additional sensors — the IMU is a favourite [17], [18], [19] but stereo systems, structured light and LIDAR also fall into this category. Keeping power-, space- and weight-constrained applications in mind, we prefer to avoid additional sensors and to utilize more general visual cues available in monocular sequences. A popular approach for

*This work was supported by the UK's Engineering and Physical Science Research Council [grant number EP/J014990].

*The authors are with the Active Vision Lab, Department of Engineering Sciences, University of Oxford, Oxford, UK {duncan,olaf,dwm}@robots.ox.ac.uk

Fig. 2. Parametrizations of points, objects and their respective projections. Object detections and their labels.

this monocular case is to impose a geometrical constraint, often assuming the camera is at a fixed height above a ground plane [20], [21], [22], [23], [24], [25]. These methods excel when this assumption holds, but are incapable of operating otherwise, e.g. with aerial images taken from a drone flying at variable altitudes. Some of these methods, e.g. [25], use object detections as we do. However, these detections are only used to infer the ground plane, hence still requiring a fixed camera height and causing additional complications in e.g. indoor environments, where detected objects may also reside on tables and shelves in addition to the floor.

We next discuss methods addressing scale drift in the more general, unconstrained case. Strasdat *et al.* [11] describe a pose-graph optimization that corrects scale drift at loop-closure. Once a loop is closed, the depth of map points in keyframes at the start and end of the loop is compared, and the scale correction is then propagated around the loop. An important drawback of such methods is that scale drift cannot be compensated for if the trajectory contains no closed loops.

Botterill *et al.* [26] assume objects in the scene are seen more than once at well-separated parts of the camera trajectory. Landmarks are paired together as objects whose size distributions (the distance between their constituent landmarks) are learnt online. When a previously-learnt object is observed again, its expected size is used to measure and correct scale drift. However, as object sizes are learnt online they are themselves subject to drift in their estimates, and the lack of prior knowledge prevents solving for global scale.

Castle *et al.* [15] detected previously learnt planar objects while running monoSLAM [27], [7]. Recovered homographies are decomposed to allow incorporation of the object geometry into monoSLAM's EKF. The method resolves the depth/speed scaling ambiguity, but it requires that *specific* objects be learnt and detected rather than object classes. In later work using bundle adjustment rather than an EKF, they used objects merely as augmentations to the map rather than allowing their geometry to influence processing [28]. Civera *et al.* [29] also used an EKF, and augmented their map with full three-dimensional objects; but they too considered particular objects rather than object classes.

Two further works that make use of objects to improve map accuracy are [30] and [31]. Bao *et al.* [30] jointly estimate camera, point and rectangular object positions in a bundle adjustment, however the focus of their work is on enhancing 3D reconstruction and object recognition within *offline* structure from motion. Gálvez-López *et al.* [31] recently proposed a method that adds objects modelled from point clouds to the map in which known distances between object-points are used for adding additional geometrical constraints to a bundle adjustment and enforcing scale in the map. A database of 500 objects is used, although the method is unable to extend to general object classes.

Dame *et al.* [32] use a monocular method to produce a dense map that is later refined using 3D shape priors embedded in a low-dimensional latent space. As the scale of shapes is known, the scale of the map may be set. This method has only been tested on small indoor examples rather than long range data and it is unclear whether it can actually correct scale-drift rather than simply set the global scale.

Our work borrows something from each of the above. We consider objects of a generic class rather than specific object instances, we favour minimal object representations for speed, and we consider on-line operation where scale is corrected without loop closure. Rather than using objects simply to localize a ground-plane, we exploit them in a local optimization which requires no constraints on the environment. We choose bundle adjustment over a pose-graph-based optimization for accuracy's sake. Finally, rather than estimating object size online, we use prior knowledge of size, thereby avoiding size distribution drift and allowing objects to set a globally correct scale.

III. Method

Our aim is to incorporate objects as additional measurements into bundle adjustment with the goal of guiding scale estimation. To achieve this, we first discuss and formalize the relevant observation and world models in Section III-A. We then briefly explain how we detect and measure objects in the images in Section III-B. The core aspects of how we incorporate our detections into the bundle adjustment, how this bundle adjustment can then be used to correct and guide the scale estimation, and how this all can be used in an online framework are presented in Sections III-C, III-D, and III-E, respectively. Finally we consider outliers, false detections and other model violations in Section III-F.

A. Observation and World Model

Scale in itself is reflected by a single degree of freedom in a 3D world model. When optimizing the 3D model, the scale can therefore be resolved by a single additional constraint. Accordingly we parametrize objects as 3D points that have a single size or physical extent parameter in addition to

their coordinates in space. For each object observation we measure a bounding box in the image, which will provide the constraints for scene size. In contrast, more complicated object models e.g. in [33], [34], can only be interpreted if the pose of the object is known and a particular 3D object model is given. More sophisticated methods with 2D silhouette segmentation and dense 3D object models such as [32] could also be devised, however, we argue that such a level of complexity is not needed to provide the basic scale information we are concerned with in this paper.

As shown in Fig. 2 (left), an object k is represented in the world frame w by $\mathbf{Q}_{kw} = [\mathbf{X}^\top, R]^\top_{kw}$, where $\mathbf{X} = [X, Y, Z, 1]^\top$ is the homogeneous location of its centre and R its extent. The camera is assumed to have a known intrinsic matrix $\mathtt{K}$ comprised of focal lengths f_u and f_v and principal point $[u_0, v_0]$. Its i-th pose is given by the Euclidean transformation $\mathtt{T}_i$, and in the frame of camera i the object is

$$\mathbf{Q}_{ki} = \begin{bmatrix}\mathbf{X} \\ R\end{bmatrix}_{ki} = \begin{bmatrix}\mathtt{T}_i & \mathbf{0}^{4\times1} \\ \mathbf{0}^{1\times4} & 1\end{bmatrix} \mathbf{Q}_{kw} \ . \tag{1}$$

The predicted projection of the object is given by

$$\hat{\mathbf{q}}_{ki}(\mathtt{T}_i, \mathbf{Q}_{kw}) = [u, v, w, h]^\top_{ki} \tag{2}$$

where $[u, v]$ are the inhomogeneous image coordinates found from the homogeneous projection $\mathbf{x}_{ki} \sim \mathtt{K}[\mathtt{I}|\mathbf{0}]\mathbf{X}_{ki}$, and where the width and height are $[w, h]_{ki} = 2R_{ki}Z^{-1}_{ki}[f_u, f_v]$. A map point $\mathbf{P}_{kw}$ is treated as a volume-less object, so that the representation in the image is

$$\hat{\mathbf{p}}_{ki}(\mathtt{T}_i, \mathbf{P}_{kw}) = [u, v, 0, 0]^\top_{ki} \ . \tag{3}$$

B. Object Measurements and Data Association

The projection of an object is invertible as its extent parameter resolves depth-ambiguity. An object can therefore be localized from a single measurement in a keyframe. However, to provide information about the camera's speed and hence the scale, data association must be solved between object detections in successive frames.

We obtain detections in keyframes and corresponding data association labels using the tracking-by-detection algorithm of [35], [36]. This system is tailored to the detection of vehicles in street settings and hence well suited for our experimental evaluation using the established KITTI data set [21]; however it is by no means essential for our work and could easily be replaced with a different system in other application domains. Any method that is able to detect and associate objects in the input sequence is able to fit into our method, and we do not concern ourselves with details of the detection algorithm. Example detections and their object identifiers are shown in Fig. 2 (right).

C. Object Bundle Adjustment

Consider a set of landmarks $\mathcal{P} = \{\mathbf{P}_{0w}, \mathbf{P}_{1w}..., \mathbf{P}_{Jw}\}$ potentially observed in a set of video frames with poses $\mathcal{T} = \{\mathtt{T}_0, \mathtt{T}_1, ..., \mathtt{T}_M\}$. Bundle adjustment seeks to find landmark positions and camera poses that minimize the summed weighted 2-norms of the errors $\mathbf{r}_{ij} = (\mathbf{p}_{ij} - \hat{\mathbf{p}}(\mathtt{T}_i, \mathbf{P}_{jw}))$ between between measurements and their landmark projections. Assuming errors are normally distributed with zero mean and covariance $\mathtt{W}_{ij}$, bundle adjustment is the maximum likelihood estimator of $\mathcal{P}$ and $\mathcal{T}$. When an object is detected in a keyframe with pose $\mathtt{T}_i$ it will produce an image measurement $\mathbf{q}_{ik}$. If object measurements are similarly distributed with covariance $\mathtt{V}_{ik}$ a new bundle adjustment may be introduced that seeks to find the most likely set of keyframes, points, and objects $\mathcal{Q} = \{\mathbf{Q}_{0w}, \mathbf{Q}_{1w}..., \mathbf{Q}_{Kw}\}$ as $\{\mathcal{T}, \mathcal{P}, \mathcal{Q}\}$ from

$$\underset{\{\mathcal{T},\mathcal{P},\mathcal{Q}\}}{\arg\min} \left(\sum_{i\in\mathcal{T}} \sum_{j\in\mathcal{P}} \mathbf{r}^\top_{ij} \mathtt{W}^{-1}_{ij} \mathbf{r}_{ij} + \sum_{i\in\mathcal{T}} \sum_{k\in\mathcal{Q}} \mathbf{r}^\top_{ik} \mathtt{V}^{-1}_{ik} \mathbf{r}_{ik} \right) \ , \tag{4}$$

where $\mathbf{r}_{ik} = (\mathbf{q}_{ik} - \hat{\mathbf{q}}(\mathtt{T}_i, \mathbf{Q}_{kw}))$. In our implementation, we take point measurements to be corrupted with zero-mean Gaussian noise with a standard deviation of 1 pixel.

Estimating noise for object detections is less straightforward. If, as it is in our case, the object detector provides a hypothesis score S_d for each detection, we can assume that the lower this confidence, the more noisy the overall detection. Accordingly we can set $\mathtt{V}_{ik} = \text{diag}(S_d^{-1}, S_d^{-1}, S_d^{-1}, S_d^{-1})$, otherwise all detections are weighted equally.

D. Scale Correction

To correct scale, a prior distribution is placed on the physical extent R of an object. Its size is now a hyper-parameter of the system, and only the object's position in 3D space is optimized. By minimizing a global bundle adjustment in this way, the scale of the map inherently becomes consistent with the prior distribution over the object's size.

In the 3D world model, objects and landmarks are now equivalently parametrized, yielding a cost function that is much easier to optimize. Furthermore, points can be treated as zero-sized objects, and thus $\mathcal{P}$ becomes a subset of $\mathcal{Q}$ with $\mathtt{V}_{ik} = \text{diag}(1, 1, 0, 0)$ for points. Eq. (4) then becomes

$$\{\mathcal{Q}, \mathcal{T}\} = \underset{\{\mathcal{Q},\mathcal{T}\}}{\arg\min} \sum_{i\in\mathcal{T}} \sum_{j\in\mathcal{Q}} \mathbf{r}^\top_{ik} \mathtt{V}^{-1}_{ik} \mathbf{r}_{ik} \ . \tag{5}$$

This simplification permits us to use standard sparse bundle adjustment methods directly (*e.g.* [37], [38]). Aside from the additional operations to deal with the extra parameters in measurement-residuals, the computational complexity is essentially identical to bundle adjustment using points alone.

E. Tracking and Local Bundle Adjustment

Applying a global bundle adjustment for every frame in the video sequence quickly becomes computationally expensive and, as the number of frames grows, infeasible for real time operation. Our online system therefore only runs bundle adjustment on a sparse selection of keyframes. This local set of n keyframes is surrounding the current estimated camera position, and in our case we set $n = 10$. This keeps the computational complexity of the online system constant and manageable in a real time online system.

Consider the current camera pose $\mathtt{T}_{\text{cam}}$, the set of its n nearest keyframes $\mathcal{T}_{\text{local}}$, the objects and points $\mathcal{Q}_{\text{local}}$ visible in the these frames and finally the set of all keyframes $\mathcal{T}_{\text{fixed}}$ which have measurements of $\mathcal{Q}_{\text{local}}$ in them. We find

$$\{\mathcal{Q}_{\text{local}}, \mathcal{T}_{\text{local}}, \mathtt{T}_{\text{cam}}\} = \underset{\{\mathcal{Q}_{\text{local}}, \mathcal{T}_{\text{local}}, \mathtt{T}_{\text{cam}}\}}{\arg\min} \sum_i \sum_k \mathbf{r}^\top_{ik} \mathtt{V}_{ik} \mathbf{r}_{ik} \tag{6}$$

where $i \in \{\mathtt{T}_{\text{cam}}, \mathcal{T}_{\text{fixed}}, \mathcal{T}_{\text{local}}, \}$ and $k \in \mathcal{Q}_{\text{local}}$. This performs the local bundle adjustment and defines the local scale in the keyframes $\mathcal{T}_{\text{local}}$ via the object detections in these frames.

As the bundle adjustment is run in only a local area around the current camera position, it is unable to propagate corrections from the local window to the rest of the map. If scale drift occurs due to a lack of object detections for a long period of operation, only the most recent portion of the map consisting of the n keyframes in our local window will be corrected when objects are seen again.

F. Outlier Rejection

Object measurements are subject to a number of sources of error which must be managed for the method to function properly. Two sources of error are the detector, which might provide false positives, or the tracker, which might incorrectly associate otherwise correct detections. Next, detections are seldom absolutely accurate in terms of their size and location in the image, which will affect the accuracy of the position of the 3D world model estimated by our local bundle adjustment. Lastly and most importantly moving objects can lead to grossly inaccurate scale estimations if they are allowed to be used in the map.

The first strategy we propose for dealing with these is to require a minimum number of detections before an object is allowed to be used in the bundle adjustment. We assume that false positives and moving objects will only have a limited number of detections and thus will have insufficient detections to be allowed into the bundle adjustment. This also ensures good object localization as the more detections an object has, the more accurate its localization becomes. Objects that move in the same direction as the camera might exceed this threshold number of detections, and so we use a robust error function on residuals to reject these measurements as outliers. Eq. (5) becomes

$$\{\mathcal{Q}, \mathcal{T}\} = \underset{\{\mathcal{Q},\mathcal{T}\}}{\arg\min} \sum_{i \in \mathcal{T}} \sum_{j \in \mathcal{Q}} \text{Obj}\left(|\mathbf{r}_{ik}|/\mathtt{V}_{ik}, \sigma_T\right) \ , \quad (7)$$

where $\text{Obj}(\cdot, \sigma_T)$ is the Tukey biweight objective function [39]. For point measurements, σ_T is set as an estimated standard deviation of the distribution of point errors. For the centre error component in object residuals, we assume a 50 pixel error, and thus set σ_T to twice this at 100 pixels.

If objects are not seen for a long period of time, it is possible that scale will drift. In this case, estimated objects will be significantly larger or smaller than the surrounding map, resulting in high error in the boundary component of object residuals $\mathbf{r}_{ik}$. To allow the bundle adjustment to correct scale in this case, rather than rejecting measurements as outliers, no robust estimator is used for boundary components.

IV. Implementation Outline

PTAM [8] forms a basis for our implementation, with minor modifications to the tracking thread and significant ones to mapping thread. Fig. 1 shows examples of the tracking output and the map output of the method.

After initialization, the tracker runs continuously every frame, using a simple motion prediction to assist matching of FAST features [40] to landmark projections and thence to iteratively optimize the pose using a re-weighted least squares algorithm. Like PTAM, our method matches features using 8×8 pixel templates that are first coarsely matched with patches from other features in a search radius and then iteratively refined for sub-pixel precision.

To determine whether a keyframe is required, in addition to measuring the distance from other keyframes we monitor the entropy ratio between the current estimate and the last keyframe dropped [41]. If either of these crosses a threshold, the current frame is used as a keyframe. As a subset of map points are used for tracking, the mapmaker first searches for measurements of any other map points and adds them to the keyframe. A set of epipolar matches is found with the closest neighbouring keyframe using PTAM's patch matching algorithm, which is used to triangulate new map points. Any object measurements (in the form of bounding boxes from object detections) are then added to the keyframe. If an object has not been seen before, it can be localised immediately from a single measurement in the keyframe.

If there are enough object measurements in the surrounding keyframes, a local point and object bundle adjustment is applied. If no objects are found, or the currently visible ones do not have enough measurements, a local point-only bundle adjustment is performed. With a window of 10 adjustable keyframes the bundle adjustment takes around 150ms to converge and so tracking is not disrupted. The new keyframe is then added to a queue in the mapmaker, which adds it to the map in a separate thread. Finally the motion model is updated with the current position of the camera.

Unlike PTAM's mapmaker, ours does not perform any global or local optimization, but is simply responsible for adding keyframes. For thread safety, it does not run while the tracker is performing its local bundle adjustment.

V. Experiments and Results

To evaluate the performance of our method we tested it on a set of 11 long-range outdoor sequences from the KITTI street scene dataset [42]. We emphasize that many methods previously tested on this dataset impose the geometric constraint that the camera is at fixed height above a ground plane, as outlined in Section II. Such methods are obviously well-suited to this specific dataset, and will outperform our proposed method. However, our objective is to demonstrate an alternative method that is applicable to general 3D motion, and still has comparable performance to existing monocular methods. A number of the sequences in the dataset also contain loops, although we explicitly do *not* perform closure to simulate open loop trajectories.

In Section III-D we mentioned that our system has a hyper-parameter, namely the real world dimensions of the detected object classes. Due to the nature of the KITTI dataset, we have chosen cars as the only object class, although our system could deal with more classes. In all of our experiments we set the single resulting hyper-parameter, the

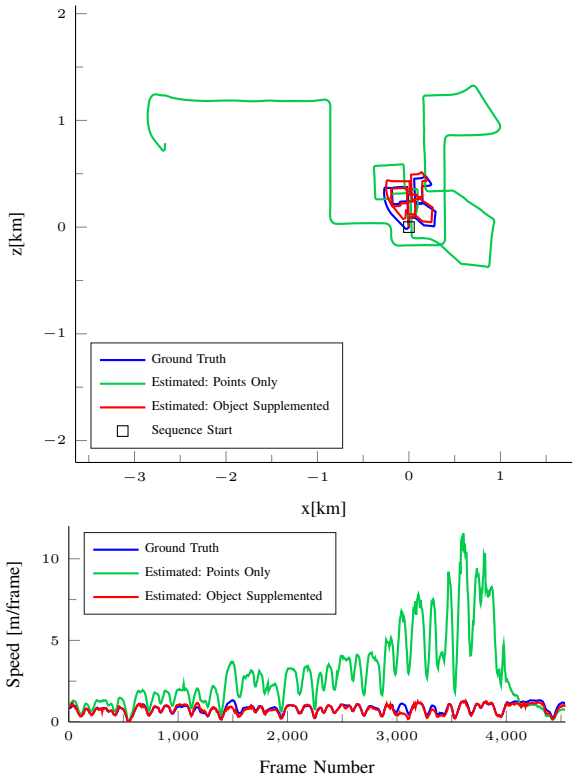


Fig. 3. Comparison between (a) the trajectories obtained using monocular SLAM with a bundle adjustment using points alone and (b) those obtained using a bundle adjustment supplemented with objects for KITTI sequence 0. Ground truth data is included for comparison.

typical size of a car, as 2.4 m, which is a mean from several manufacturers' specifications and which we will validate experimentally in Section V-D.

First we will investigate the importance of correcting for scale drift in Section V-A. Here we compare our proposed object and scale aware system to a virtually identical one that ignores objects and has no information about scale. In Section V-B, we then take a look at the performance of our proposed system in adverse circumstances, when objects in the scene are scarce. We compare our results obtained with KITTI's online assessment tool to the current top-performing monocular odometry methods in Section V-C. As our method does not rely on a fixed camera-height we present qualitative results from a video sequence captured with a hand-held camera at varying heights in Section V-E.

A. Comparison with Bundle Adjustment with Points Alone

Fig. 3 shows example trajectories and camera speeds obtained from a system that uses points alone, and from one which is supplemented with objects. Although both experience rotational and translational drift, the object-supplemented method is able to stay well within the actual speed of the camera. The point-method accumulates considerable amounts of error due to the scale drift it experiences.

Table I shows the summary of our results for the first 11 sequences in the KITTI dataset which come with ground truth data. To assess the extent of scale drift, we use the measure introduced by Strasdat *et al.* [11] between the estimated keyframes $\mathtt{T}_i^{\text{est}}$ and the true keyframes $\mathtt{T}_i^{\text{true}}$ using

TABLE I

RMSE WITH (THIS WORK) AND WITHOUT OBJECTS FOR THE FIRST 11 SEQUENCES IN THE KITTI DATASET.

Seq #	No of frames	E, this work	E, no objects	Seq #	No of frames	E, this work	E, no objects
0	4541	**73.4**	1181.0	6	1101	**73.1**	244.9
1	1101	**545.8**	712.0	7	1101	**47.1**	110.2
2	4661	**55.5**	815.7	8	4071	**72.2**	1907.6
3	801	**30.6**	81.1	9	1591	**31.2**	139.6
4	271	10.7	**7.4**	10	1201	**53.5**	115.3
5	2761	**50.8**	798.8				

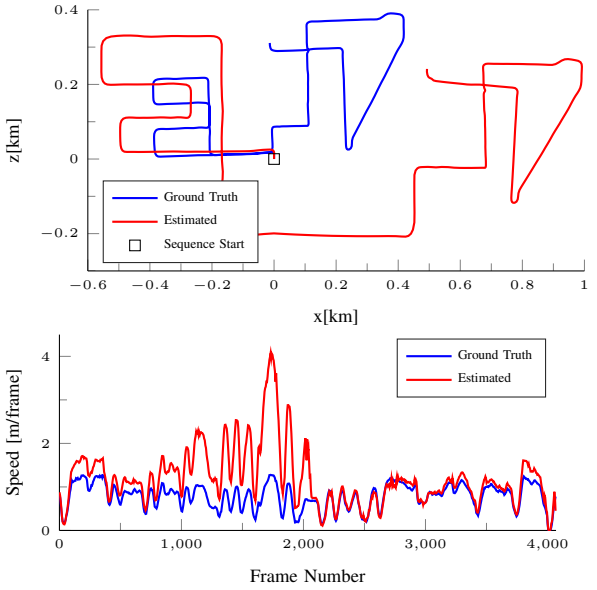


Fig. 4. Estimated trajectory (a) and translational speed (b) obtained by initially ignoring objects in the first 2000 frames of the KITTI sequence 8. At frame 2000, once objects are included in the bundle adjustment, scale is promptly corrected.

the minimum of the sum of square differences. Root mean square error is then calculated as $E_{\text{RMSE}} = \sqrt{M/N}$, where N is the number of keyframes in the trajectory, $M = \arg\min_s \sum_{i \in \mathcal{T}} (\mathbf{t}_i^{\text{true}} - s\mathbf{t}_i^{\text{est}})^2$, s is a global scale factor, and $\mathbf{t}_i$ is the translational component of keyframe $\mathtt{T}_i$. The results of our method without objects are included for comparison.

B. Correction of Long Sequences with Scale Drift

Even if scale has drifted significantly because of a lack of object measurements, object-supplemented bundle adjustment is still able to correct the scale estimate once objects eventually are seen again. This is demonstrated in Fig. 4 which shows the estimated trajectory obtained by initially ignoring all object measurements for the first 2000 frames. Almost immediately after objects are used again, the scale is corrected to a value close to ground truth.

C. Quantitative Evaluation

Our results for the last 10 sequences of the KITTI dataset have been evaluated using the online assessment tool at *http://www.cvlibs.net/datasets/kitti/eval_odometry.php* [42] for comparison with other results. The average performance over all monocular SLAM systems is captured by some 9% translation error and 0.020 deg/m rotational error, while

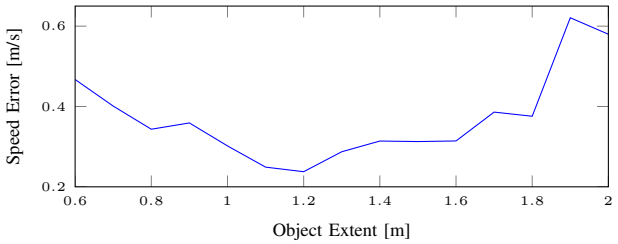


Fig. 5. Per-frame average RMS error in speed vs. object extent for KITTI training sequences that contain cars.

the best performance [23], [25] achieves 2.4% translation error and 0.006 deg/m rotational error. Unfortunately for comparison purposes, all reference methods employ a planar constraint in one form or other. Our method yields 20% translation error and 0.014 deg/m rotational error on average, results which at least approach, and for rotation exceed, these other methods.

We have considered a number of possibilities for reducing error. First, we find that the large inter-frame motion in the KITTI sequences is over-stretching the feature and camera tracking stage of PTAM, which was developed specifically for small AR workspaces. A simple KLT-based [43] odometry system is able to find numerous matches between every pair of frames on sequences where PTAM lost a number of map points. Sequence 1 in Table. I is one such example.

Secondly, a number of the test trajectories do not contain many static objects of our interest class, making scale-drift correction impossible on these sequences. A prime example of this is sequence 14, which takes place in a park where the only scenery are a number of rows of shrubs. We discuss the need for more variety in object classes later.

A third potential source of error is that of setting the object extent parameter at a value slightly larger or slightly smaller than reality. This possibility is explored next.

D. Selection of Object Extent

As mentioned earlier, the object extent R for detected cars was set as an average from the model ranges of several popular manufacturers. However, an empirical search over this parameter is also possible. To select an optimal extent parameter we enumerate a range of plausible values and evaluate the performance of our system on all training sequences of the KITTI dataset that contain cars. We measure the average per-frame RMS error in estimated speed, as illustrated in Fig. 5, and choose the object radius/extent R as the one with lowest error. This coincides with our choice of the object diameter $2.4\,\mathrm{m}$.

E. Varying Camera Height

A major benefit of our method over other monocular methods relying on a ground plane to estimate scale is the ability to deal with a varying camera height. To test this, we recorded an outdoor sequence using a hand-held camera at various heights. As ground truth was not available, we chose to run our method on a small loop and evaluate its performance based on distance between the start and end of the loop. The footage was obtained using a Samsung Galaxy S6 using a fixed focal-length video mode that was calibrated beforehand. Due to the short trajectory length, scale-drift was artificially amplified by corrupting image measurements with zero-mean Gaussian noise with a standard deviation of 1.5 pixels. Fig. 6 shows the resultant trajectories estimated using a point-only and an object-supplemented bundle adjustment overlaid onto satellite imagery of the actual area where the sequence was filmed. Cars with a diameter of $2.4\,\mathrm{m}$ were once again used as the class and extent for the object-supplemented method. Comparison of the trajectories with satellite imagery shows that object-supplemented bundle adjustment provides a marked decrease in scale drift over its point-only counterpart.

Fig. 6. Resultant trajectories estimated using point-only bundle adjustment (yellow), and object-supplemented bundle adjustment (red). Satellite imagery of the mapped area is shown for comparison. Satellite imagery: Google, DigitalGlobe.

VI. Conclusion

We have presented a new method of incorporating prior scale information from object classes into monocular SLAM systems without significant increases in computational complexity. We have shown that the system is able to keep scale drift low throughout a trajectory and correct scale drift after long periods without object detections. While other methods rely on external hardware or assumptions about the height of the camera above the ground plane, our method only assumes that there is some known object in the map and with a relatively uniform size. This is ideal for general purpose applications, when other assumptions no longer hold.

A number of avenues might be pursued to improve performance. On its own the method is unable to correct scale in parts of the map outside of the local bundle adjustment window. In Section V-B for example, the bundle adjustment isn't able to correct parts of the trajectory where no object observations are available. We propose combining the local bundle adjustment with a pose-graph optimisation as described in Strasdat *et al.*'s double window method [44]. Scale information from the local bundle adjustment could be propagated along the pose-graph using relative similarity transformations [11] between poses.

One challenge faced on the KITTI dataset was a lack of sequences with static objects of our particular class. Due to

the simplicity of our method, adding object classes is trivial only requiring a viable object detector and tracker. Care has to be taken to correctly balance possibly conflicting scale information from different object classes.

Online estimation of object sizes is also possible. Using more certain areas of the map to estimate the size distribution, this certainty could be propagated to more uncertain areas containing the same object class. Although this would likely result in maps that are not metrically accurate, hand selection of parameters would no longer be required.

Currently object data association between frames is performed independently prior to running the method. As a possible extension detection and association can be performed online, utilizing geometric information for better results.

REFERENCES

[1] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[2] J. J. Leonard, H. F. Durrant-Whyte, and I. J. Cox, "Dynamic map building for an autonomous mobile robot," *International Journal of Robotics Research*, vol. 11, no. 8, pp. 286–298, 1992.

[3] W. Maddern, G. Pascoe, and P. Newman, "Leveraging Experience for Large-Scale LIDAR Localisation in Changing Cities," in *International Conference on Robotics and Automation*, Seattle, WA, USA, May 2015.

[4] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "RSLAM: a system for large-scale mapping in constant time using stereo," *International Journal of Computer Vision*, vol. 94, no. 2, pp. 198–214, 2011.

[5] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *International Conference on Robotics and Automation*, 2010, pp. 4372–4378.

[6] A. J. Davison, W. W. Mayol, and D. W. Murray, "Real-time localisation and mapping with wearable active vision," in *Int Symp on Mixed and Augmented Reality*, 2003, pp. 18–27.

[7] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 1052–1067, 2007.

[8] G. Klein and D. W. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc 6th IEEE/ACM Int Symp on Mixed and Augmented Reality*, 2007, pp. 225–234.

[9] G. Graber, T. Pock, and H. Bischof, "Online 3d reconstruction using convex optimization," in *International Conference on Computer Vision Workshops*, 2011, pp. 708–711.

[10] A. J. Davison and D. W. Murray, "Sequential localisation and map-building using active vision," *Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 865–880, July 2002.

[11] H. Strasdat, J. M. Montiel, and A. J. Davison, "Drift aware large scale monocular SLAM," in *Robotics: Science and Systems*, 2010.

[12] J.-S.Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc IEEE Int Symp on Computational Intelligence in Robotics and Automation*, 1999, pp. 318–325.

[13] S. Se, D. G. Lowe, and J. J. Little, "Local and global localization for mobile robots using visual landmarks," *Intelligent Robots and Systems*, pp. 414–420, 2002.

[14] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[15] R. O. Castle, G. Klein, and D. W. Murray, "Combining monoSLAM with object recognition for scene augmentation using a wearable camera," *Image and Vision Computing*, vol. 28, no. 12, pp. 1548–1556, 2010.

[16] B. Julesz, *Foundations of Cyclopean Perception*. University of Chicago Press, 1971.

[17] C. Hide, T. Botterill, and M. Andreotti, "Vision-aided IMU for handheld pedestrian navigation," in *Proc of the Inst of Navigation Conference, GNSS 2010*, 2010.

[18] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of IMU and vision for absolute scale estimation in monocular SLAM," *Journal of Intelligent and Robotic Systems*, vol. 61, no. 1-4, pp. 287–299, 2011.

[19] M. Achtelik, S. Weiss, and R. Siegwart, "Onboard IMU and monocular vision based control for MAVs in unknown in-and outdoor environments," in *Int Conf on Robotics and Automation*, 2011, pp. 3056–3063.

[20] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3D reconstruction in real-time," in *Intelligent Vehicles Symposium*, 2011, pp. 963–968.

[21] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme," in *Intelligent Vehicles Symposium*, 2010, pp. 486–492.

[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 *Int Conf on Computer Vision*. IEEE, 2009, pp. 1413–1419.

[23] S. Song, M. Chandraker, and C. C. Guest, "Parallel, real-time monocular visual odometry," in *International Conference on Robotics and Automation*. IEEE, 2013, pp. 4698–4705.

[24] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium*, 2011, pp. 963–968.

[25] S. Song and M. Chandraker, "Robust scale estimation in real-time monocular sfm for autonomous driving," in *Computer Vision and Pattern Recognition*, 2014, pp. 1566–1573.

[26] T. Botterill, S. Mills, and R. D. Green, "Correcting scale drift by object recognition in single-camera SLAM," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1767–1780, 2013.

[27] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc 9th IEEE Int Conf on Computer Vision*, 2003, pp. II: 1403–1410.

[28] R. O. Castle and D. W. Murray, "Keyframe-based recognition and localization during video-rate parallel tracking and mapping," *Image and Vision Computing*, vol. 29, no. 8, pp. 524–532, 2011.

[29] J. Civera, D. Gálvez-López, L. Riazuelo, , J. D. Tardós, and J. M. M. Montiel, "Towards semantic SLAM using a monocular camera," in *Intelligent Robots and Systems*, 2011, pp. 1277–1284.

[30] S. Y. Bao and S. Savarese, "Semantic structure from motion," in *Proc 24th IEEE Conf on Computer Vision and Pattern Recognition*, 2011, pp. 2025–2032.

[31] D. Gálvez-López, M. Salas, and J. M. M. Monteil, "Real-time monocular object slam," `arXiv:1504.02398 [cs.CV]`, 2015.

[32] A. Dame, V. A. Prisacariu, C. Y. Ren, and I. Reid, "Dense reconstruction using 3d object shape priors," in *Computer Vision and Pattern Recognition*, 2013, pp. 1288–1295.

[33] S. Y. Bao, M. Bagra, and S. Savarese, "Semantic structure from motion with object and point interactions," in *Proc Workshops, 13th IEEE Conf on Computer Vision*, 2011, pp. 982–989.

[34] S. Y. Bao, M. Bagra, Y.-W. Chao, and S. Savarese, "Semantic structure from motion with points, regions, and objects," in *Computer Vision and Pattern Recognition*, 2012, pp. 2703–2710.

[35] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3D traffic scene understanding from movable platforms," *Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 1012–1025, 2014.

[36] H. Zhang, A. Geiger, and R. Urtasun, "Understanding high-level semantics by modeling traffic patterns," in *Proc 14th IEEE Int Conf on Computer Vision*, 2013, pp. 3056–3063.

[37] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[38] C. Engels, H. Stewénius, and D. Nistér, "Bundle adjustment rules," in *Photogrammetric Computer Vision*, 2006, pp. 266–271.

[39] Z. Zhang, "Parameter estimation techniques: a tutorial with applications to conic fitting," *Image and Vision Computing*, vol. 15, no. 1, pp. 59–76, 1996.

[40] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc 9th European Conf on Computer Vision*, 2006.

[41] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *Intelligent Robots and Systems*, 2013, pp. 2100–2106.

[42] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[43] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International journal of computer vision*, vol. 56, no. 3, pp. 221–255, 2004.

[44] H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige, "Double window optimisation for constant time visual SLAM," in *Int Conf on Computer Vision*, 2011, pp. 2352–2359.