Extrinsics Autocalibration for Dense Planar Visual Odometry

Jacek Zienkiewicz
Department of Computing
Imperial College London
London, UK
`j.zienkiewicz12@imperial.ac.uk`

Andrew Davison
Department of Computing
Imperial College London
London, UK
`a.davison@imperial.ac.uk`

Abstract

A single downward-looking camera can be used as a high precision visual odometry sensor in a wide range of real-world mobile robotics applications. In particular, a simple and computationally efficient dense alignment approach can take full advantage of the local planarity of floor surfaces to make use of the whole texture available rather than sparse feature points. In this paper we detailed present analysis and highly practical solutions for auto-calibration of such a camera's extrinsic orientation and position relative to a mobile robot's coordinate frame. We show that two degrees of freedom, the out-of-plane camera angles, can be auto-calibrated in any conditions; and that bringing in a small amount of information from wheel odometry or another independent motion source allows rapid, full and accurate 6 DoF calibration. Of particular practical interest is the result that this can be achieved to almost the same level even without wheel odometry and only widely-applicable assumptions about nonholonomic robot motion and the forward/backward direction of its movement. We show accurate, rapid and robust performance of our auto-calibration techniques for varied camera positions over a range of low-textured real surfaces both indoors and outdoors.

1 Introduction

Computer vision is increasingly widely accepted as suitable as the main outward-looking sensing modality for mobile robot navigation. One or more specialised cameras can potentially play the key role in all of the capabilities required for autonomous navigation [Carrera et al., 2011], including motion estimation, place recognition, obstacle detection and target or object detection and recognition. However, there are still relatively few significant deployments of vision-based navigation in real-world applications; particularly in the domain of low-cost robotics aiming at mass-market service tasks. This is because robot vision systems are commonly perceived as complex, heavyweight and lacking robustness in changing conditions. A particular concern is the need for difficult and precise calibration because the accuracy of vision systems depends strongly on this and it must usually be performed off-line on a per-robot, per-camera basis. Clearly auto-calibration solutions can go a long way to addressing these concerns, and offer the possibility of low-cost robots which can calibrate themselves in the field and even be able to re-calibrate continuously as their own condition and that of their surroundings change.

In this paper we study an approach to Visual Odometry (VO), or incremental camera-based motion estimation, which is based on observation of the planar surface on which a robot is driving, and solve the necessary extrinsics auto-calibration problem with a one-shot, infrastructure-free method. VO can jump over the well-known problems with dead reckoning on the basis of wheel odometry, which are acute in the case of light-weight, low-cost robots moving on unpredictable surfaces such as carpet or dirt. Most VO systems are heavyweight, complicated and often fragile approaches which reconstruct and track the world in full 3D, usually in the form of a point cloud. Here, we instead take direct advantage of domain knowledge that wheeled mobile robots move on globally or locally planar surfaces. This simplifies both the process of VO and the auto-calibration of the camera used. We mount a standard single camera on a robot, point it down at the floor, and estimate motion by tracking the natural floor texture which moves past the camera using whole image dense alignment. Almost every surface, even those which are apparently quite blank and do not lead to the extraction of standard point features, has trackable texture when used in whole image alignment. Knowledge of the extrinsic pose of the camera relative to the robot frame allows this alignment to take place directly with respect to the three degrees of freedom of planar robot motion, and precisely allows *all* of the surface texture available to contribute to the motion estimate.

Auto-calibrating a planar VO system requires all six degrees of freedom of the $\mathbf{SE}(3)$ transformation between the robot's drive frame and the camera frame to be estimated. In planar VO, we have strong information to help this calibration procedure, because we can assume that the downward-looking camera directly observes the plane on which the robot is locally driving. This means that of the six unknown extrinsic parameters, two can be estimated reliably by capturing a very short sequence of video as the robot moves, and without needing any external reference. These are the roll and pitch angles of the camera, since only a correct estimate of these will lead to inter-frame homography warps of the planar texture which are consistent with movement over the same plane. We experimentally demonstrate the ease and accuracy of this first part of the auto-calibration procedure.

Calibrating the remaining four parameters — the camera's yaw angle, and its translational position relative to the robot frame — cannot be achieved without some kind of external reference, but we show clearly that there are several highly simple and practical forms this can take which lead to rapid and precise full calibration, by formulating the problem as a pose-graph optimisation with the help of wheel odometry or a nonholonomic constraint on the vehicle motion. First, if we have a sequence of camera trajectory where the robot makes a small number of movements and turns with accompanying synchronised wheel odometry on a good surface, we present a graph optimisation algorithm which produces an unbiased estimate of the full calibration. Note that the purely relative information from wheel odometry is sufficient here, and we do not need to perform full SLAM or have an absolute external motion reference. We perform auto-calibration using only wheel odometry as an additional reference, and take the uncertainty in that wheel odometry into account, only requiring that the wheel odometry can be assumed to have zero-mean errors incrementally on a region with good surface grip to achieve unbiased camera extrinsics estimation.

We then examine even weaker cases for a robot which does not have wheel odometry, and show that very generally applicable assumptions about the nonholonomic motion of most wheel robots are sufficient for accurate auto-calibration of all but two of the remaining degrees of freedom of the robot-camera transformation. If we then further augment this with simple labels about which parts of the calibration trajectory involve forward or backward robot motion, which should be available from the robot's control signal even if it has no odometry, we can calibrate everything apart from the height of the camera above the plane, which requires just one manual measurement or known object to resolve.

A wide range of real-world surfaces permit successful and precise planar visual odometry and auto-calibration, and we will demonstrate results on surfaces such as carpet, vinyl floor tiles, concrete, grass and wooden boards. We demonstrate experimentally that unbiased, highly robust VO is obtained from our system over the full range of dynamics of our experimental robot's motion. We analyse the performance of our auto-calibration techniques against ground truth where possible, and check them for consistency by repeatedly achieving the same auto-calibration results over different motions and surface types.

Our real-time implementation of both VO and auto-calibration makes efficient use of parallel processing, currently provided via a laptop PC GPU. We consider this a great strength rather than a weakness moving forward. If one considers the embedded computing platforms we can anticipate on low-cost mobile devices perhaps five years hence, it is strongly likely that massively parallel GPU-like, FPGA-like or specialised DSP units will offer the dominant low-power processing resource and parallelisable algorithms will increasingly come to the fore.

This paper is a substantially extended version of conference paper [Zienkiewicz et al., 2013], with focus on auto-calibration of planar visual odometry and includes new experimentation validation and theoretical contributions on auto-calibration with weaker background information (assumptions of nonholonomic motion and simple motion priors). We also apply our methods to a dataset from an entirely different setting to the indoor robot used in most of our experiments: video captured from a car's downward-looking parking camera while travelling at normal road speeds. Furthermore we analyse how violation of planar assumptions affects the performance of our methods.

The paper is organised as follows. We present related work in Section 2, then proceed to mathematical preliminaries in Section 3, and the main exposition of our approaches to visual odometry and auto-calibration in Sections 4 and 5. We present experimental results and discussion in Section 6 and give conclusions in Section 7.

2 Related work

2.1 Visual odometry

The term *visual odometry* [Nistér et al., 2004] identifies the important class of problems where accurate but purely incremental motion estimation can usefully be provided by a camera system. This is in contrast to more general visual SLAM systems (e.g. [Davison, 2003]) of the time, aiming at drift-free localisation but with more restrictions on local accuracy and scale of operation.

There have been a number of notable VO systems which track general 6DoF motion; the most successful relying on stereo vision (e.g. the incremental components of [Konolige et al., 2007, Mei et al., 2009]). VO becomes easier if domain assumptions can be brought strongly into play, and other authors have considered the special case of planar robot motion over a ground surface. As in our work, [Campbell et al., 2005] demonstrated how a single camera can track floor texture in different situations, but using optical flow computation and with a partial forward-looking view to estimate orientation. [Kitt et al., 2011] argued convincingly for taking advantage of prior knowledge that a camera is viewing a planar floor in their system using feature correspondences. [Nourani-Vatani et al., 2009] used a downward, fronto-parallel looking camera and correlation of patches. There also exists a wide range of homography-based methods that estimate camera ego-motion using the assumption of a planar scene, e.g. [Pirchheim and Reitmayr, 2011, Saurer et al., 2012, Adams et al., 2002]. These methods utilize homography decomposition to compute the camera motion given a homography matrix [Faugeras and Lustman, 1988], and the homographies are usually determined by detecting corresponding visual features in two images to the same plane.

None of these authors went all the way in making the best possible use of the planar scene assumption. Used properly it allows *all the pixels* in a video sequence to contribute to motion estimation via iterative dense alignment. [Lovegrove et al., 2011] showed the power of this approach in the specific application of on-road vehicle motion estimation from a rear parking camera by observing the planar road surface. We adopt the core approach of that work, but now bring it to the domain of a small indoor robot which drives and rotates rapidly over a variety of real-world surfaces with different texture characteristics. Importantly, unlike the manual extrinsic calibration used by [Lovegrove et al., 2011] and the other authors above we show that a planar VO system can be rapidly auto-calibrated from a short sequence without the need for any special markers or targets.

2.2 Auto-calibration

Our method for auto-calibration of the robot-camera configuration merits comparison with other recent work on auto-calibrating the extrinsics of outward-looking sensors on mobile robots, and more generally with methods for auto-calibrating the extrinsic transformations between multiple sensors [Underwood et al., 2010]. When calibrating the transformation between a camera and an inertial measurement unit (IMU), very often approaches based on the Kalman Filter are proposed (e.g. [Kelly and Sukhatme, 2010], [Mirzaei and Roumeliotis, 2008]). Calibration of the camera extrinsics in most systems requires special reference targets [Martinelli et al., 2006, Antonelli et al., 2010], or is achieved by calibrating with reference to some carefully engineered ground truth positioning system. Recently [Knorr et al., 2013] presented a system that exploits a planar assumption for multi-camera extrinsics calibration. Their system is based on a Kalman Filter and does not require any external reference as the ground plane serves as a natural reference object. Similarly to our method, the calibration procedure relies on plane-induced homographies between successive frames. Also [Miksch et al., 2010] used the assumption that the road surface visible in the images is approximately flat to calculate the orientation of the camera with respect to the vehicle frame of reference. However, their method relies on feature extraction and mapping, puts constraints on the vehicle motion forcing it to move in straight line, and does not extend naturally to multiple frames (a final parameter is determined by a recursive filter which averages various estimates over time).

The part of our auto-calibration that is based on graph optimisation has more in common with the work by [Censi et al., 2008] and especially with [Kümmerle et al., 2011a] who performed a calibration against a full SLAM system. In their recent work [Brookshire and Teller, 2011] demonstrated that external incremental pose measurements are sufficient to recover sensor calibration provided that degenerate trajectories are avoided.

To our knowledge our work is the first time that nonholonomic constraints have been used for sensor extrinsics calibration. In the robot vision and tracking literature there are only a few examples of applying non-holonomic motion constraints to sensing directly. In their feature-based VO system, [Scaramuzza et al., 2009b] used a one-point RANSAC outlier rejection scheme based on nonholonomic constraints to reduce the computational burden of data association therefore speed up egomotion estimation. Also [Scaramuzza et al., 2009a] used nonholonomic constraints, and the known offset between camera and vehicle frame of reference, to estimate the absolute scale in SfM. [Fossati and Fua, 2008] did not directly apply nonholonomic constraints, but instead showed that incorporating the local direction of travel while estimating motion can greatly improve visual tracking, since the pose of an object has a direct influence on its direction of travel.

3 Mathematical preliminaries: frames of reference and transformations

There are two main frames of references considered in the paper, one associated with the camera and another with the robot. $\mathtt{T}^{vc}$ describes the transformation from the camera frame of reference to the robot frame of reference, whereas $\mathtt{T}^{cv} = (\mathtt{T}^{vc})^{-1}$ is its inverse. We define $\mathtt{T}^{vc} = (\mathtt{R}^{vc} \mid \mathbf{t}^{vc})$, where $R^{vc} \in \mathbf{SO}(3)$. We will refer to the position of the camera in the robot frame of reference by $\mathbf{t}^{vc} = (x_{vc}, y_{vc}, z_{vc})^\top$, and its orientation $\mathtt{R}^{vc}$ using roll, pitch and yaw angles (denoted by α_{vc}, β_{vc}, and γ_{vc} respectively).

Let $\mathtt{T}^{c_{lr}}$ be the transformation matrix describing the relative motion of the camera between time-steps when the camera captures two images; a 'live' image $\mathcal{I}^l$ and an overlapping earlier 'reference' image $\mathcal{I}^r$. Even though $\mathtt{T}^{c_{lr}}$ is a 3D motion in the camera frame of reference, the observation that the camera is fixed on the robot and therefore is moving parallel to the ground plane allows us to parameterise 3D camera motion using relative 2D robot's motion, $\mathtt{T}^{v_{lr}}$ in the following form:

$$\mathtt{T}^{c_{lr}} = \mathtt{T}^{cv}\mathtt{T}^{v_{lr}}\mathtt{T}^{vc} \; . \tag{1}$$

4 Visual Odometry

We will now describe the tracking part of our method, which uses a whole image alignment approach evolved from the iterative technique introduced by [Lucas and Kanade, 1981] to estimate the incremental motion of a camera. We assume that the camera is bolted to the robot in an arbitrary but fixed location, such that its field of view is filled with the planar floor surface. At this stage we assume that the camera extrinsics as well as intrinsics are known. Determining camera intrinsics is a straightforward, one-off procedure for a certain sensor and lens and is out of the scope of this work. Our method for auto-calibrating camera extrinsics will be presented in Section 5.

Two images (live $\mathcal{I}^l$ and reference $\mathcal{I}^r$) of the same plane are related by a plane-induced homography $\mathtt{H}^{lr}$. This homography transforms pixel coordinates in the reference image $\mathbf{p}_r$ into the live image $\mathbf{p}_l$ and depends on the camera motion $\mathtt{T}^{c_{lr}} = (\mathtt{R}^{c_{lr}} \mid \mathbf{t}^{c_{lr}})$ and the parameters $(\mathbf{n}^\top, d)$ of the plane defined in the camera reference frame:

$$\mathtt{H}^{lr} = \mathtt{K}\mathtt{T}^{c_{lr}}(I \mid -\mathbf{n}_{d_c})^\top \mathtt{K}^{-1} \; . \tag{2}$$

Here $\mathbf{n}_{d_c} = \frac{\mathbf{n}}{d}$, where $\mathbf{n}$ is the unit vector normal to the plane, and d is the perpendicular distance from the plane to the centre of camera frame r. $\mathtt{K}$ is a known camera intrinsic calibration matrix.

We can use Eq. 1 to express the homography in terms of the vehicle planar motion:

$$\mathtt{H}^{lr} = \mathtt{K}\mathtt{T}^{cv}\mathtt{T}^{v_{lr}}\mathtt{T}^{vc}(I| - \mathbf{n}_{d_c})^\top \mathtt{K}^{-1} \; . \tag{3}$$

Since the camera pose $\mathtt{T}^{vc}$ as well as $\mathbf{n}_{d_c}$ remain constant (in fact $\mathbf{n}_{d_c}$ is uniquely determined by $\mathtt{T}^{vc}$), and the robot moves on a plane, we only need to determine 3 degrees of freedom of the robot planar motion

$\mathbf{x} = (x, y, \theta)^T$. Although $\mathtt{T}^{v_{lr}}$ is in $\mathbf{SE}(2)$ we can refer to it in 3D by raising it into $\mathbf{SE}(3)$ and enforcing that the robot moves on the xy plane:

$$\mathtt{T}^{v_{lr}}(\mathbf{x}) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & x \\ \sin\theta & \cos\theta & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \tag{4}$$

Now the goal is to find a motion $\mathbf{x}$ that best registers $\mathcal{I}^l$ and $\mathcal{I}^r$ using homography $\mathtt{H}^{lr}$ parameterised by $\mathbf{x}$. To achieve this we transform all pixels in the live image via the current homography estimate into the reference image, and define an energy function F which measures the discrepancy across the whole image in the form of the sum of squared differences between all pairs of superposed pixels:

$$F(\mathtt{H}^{lr}) = \frac{1}{2} \| \mathcal{I}^l(\pi(\mathtt{H}^{lr}\mathbf{p}_r)) - \mathcal{I}^r(\pi(\mathbf{p}_r)) \|_2^2 . \tag{5}$$

Here $\mathbf{p}_r$ are homogeneous image coordinates and π is a function that performs homogeneous projection. Expressing Eq. 5 in terms of the planar motion $\mathbf{x}$ leads to the following energy function to minimise:

$$F(\mathbf{x}) = \frac{1}{2} \| f(\mathbf{x}) \|_2^2 , \tag{6}$$

with:

$$f(\mathbf{x}) = \mathcal{I}^l(\pi(\mathtt{H}^{lr}(\mathbf{x})\mathbf{p}_r)) - \mathcal{I}^r(\pi(\mathbf{p}_r)) . \tag{7}$$

An example plot of the cost function with respect to the translational degrees of freedom of the robot can be seen in Figure 1.

In order to minimise $F(\mathbf{x})$ we perform an iterative optimisation based on the Efficient Second-order Minimisation (ESM) method [Malis, 2004, Mei et al., 2008], which improves on Lucas and Kanade's original technique [Lucas and Kanade, 1981]. For details of our ESM implementation we refer to [Lovegrove et al., 2011]. In brief, we parameterise $\mathtt{H}^{lr}$ with $\mathtt{T}(\mathbf{x})$, an update matrix that represents small changes to the current estimate of the solution $\hat{\mathtt{T}}^{v_{lr}}$, in the following form:

$$\mathtt{H}^{lr}(\mathbf{x}) = \mathtt{K}\mathtt{T}^{cv}\hat{\mathtt{T}}^{v_{lr}}\mathtt{T}(\mathbf{x})\mathtt{T}^{vc}(I| - \mathbf{n}_{d_c})^\top \mathtt{K}^{-1} . \tag{8}$$

$\mathtt{T}(\mathbf{x})$ is parameterised by $\mathbf{x}$ belonging to the Lie Algebra $\mathfrak{se}(2)$ and we iteratively apply the following update

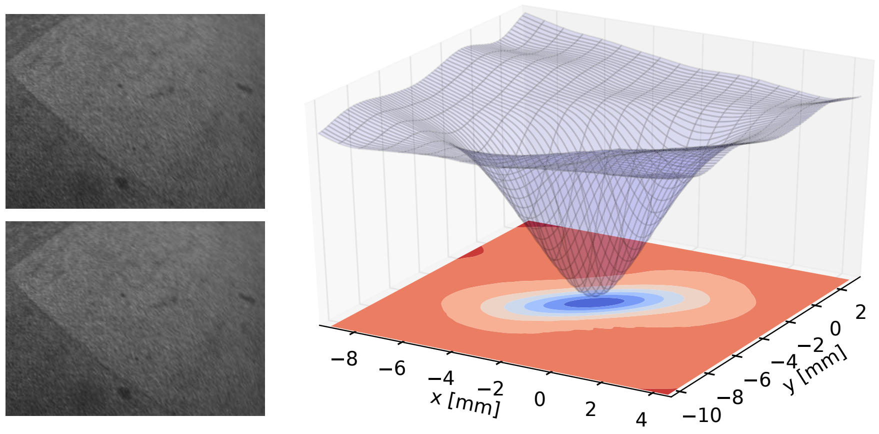

Figure 1: Typical dense tracking cost function. The function depends on the three degrees of freedom of planar robot motion, but only a translational slice is shown here.

rule:

$$\hat{\mathtt{T}}^{v_{lr}} \leftarrow \hat{\mathtt{T}}^{v_{lr}}\mathtt{T}(\hat{\mathbf{x}}) \ , \tag{9}$$

where $\hat{\mathbf{x}}$ is an update parameter estimated during an iteration. We initialise the solution $\hat{\mathtt{T}}^{v_{lr}}$ to the value found on the previous VO timestep, taking advantage of the general smoothness of robot motion relative to our camera capture rate to improve convergence. At each iteration we find an updated estimate $\hat{\mathbf{x}}$ of the state via the expression:

$$\hat{\mathbf{x}} = -(\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top f(\mathbf{0}) \ , \tag{10}$$

where:

$$\mathtt{J} = \frac{1}{2}\left(\boldsymbol{\nabla} f(\mathbf{0}) + \boldsymbol{\nabla} f(\hat{\mathbf{x}})\right) \ . \tag{11}$$

This approach requires us to evaluate the partial derivatives of the cost function $f(\mathbf{x})$ at $\mathbf{0}$ and at $\hat{\mathbf{x}}$. This may appear to be difficult but thanks to our parameterisation we can compute it effectively using the chain and product rules and the gradients of the live and reference images, as shown in [Lovegrove et al., 2011]:

$$\frac{\partial f(\mathbf{x})}{\partial x_i} = \left.\frac{\partial \mathcal{I}^l(\mathbf{a})}{\partial \mathbf{a}}\right|_{\mathbf{a}=\pi(\mathtt{H}^{lr}(\mathbf{x})\mathbf{p}_r)} \left.\frac{\partial \pi(\mathbf{b})}{\partial \mathbf{b}}\right|_{\mathbf{b}=\mathtt{H}^{lr}(\mathbf{x})\mathbf{p}_r} \frac{\partial \mathtt{H}^{lr}(\mathbf{x})}{\partial x_i}\mathbf{p}_r \tag{12}$$

$$\frac{\partial \mathtt{H}^{lr}(\mathbf{x})}{\partial x_i} = \mathtt{K}\mathtt{T}^{cv}\hat{\mathtt{T}}^{v_{lr}}\frac{\partial \mathtt{T}(\mathbf{x})}{\partial x_i}\mathtt{T}^{vc}(I| - \mathbf{n}_{d_c})^\top \mathtt{K}^{-1} \ . \tag{13}$$

Here we exploit the fact that for small $\hat{\mathbf{x}}$, $\frac{\partial \mathtt{H}^{lr}(\mathbf{0})}{\partial x_i}$ and $\frac{\partial \mathtt{H}^{lr}(\hat{\mathbf{x}})}{\partial x_i}$ are equivalent. $\left.\frac{\partial \mathtt{T}(\mathbf{x})}{\partial x_i}\right|_{\mathbf{x}=\mathbf{0}} = \mathtt{G}_i^{\mathbf{SE}(2)}$ is the ith generator of the $\mathbf{SE}(2)$ group.

Instead of using the $l2$-norm as indicated in Eq. 5, in practice we use a Huber norm to robustify the cost function in an iteratively reweighted least squares framework. This helps us to reject partially nonplanar structures in the field of view as well as other effects that are not directly modelled in our approach, e.g. specularities and shadows. We also make use of a coarse-to-fine multiresolution approach to speed up computations and extend the basin of convergence.

5 Auto-calibration

The goal of auto-calibration is to estimate the pose of the camera relative to the robot which carries it. Its translational position is determined by the position vector (x_{vc}, y_{vc}, z_{vc}) and its orientation in the form of roll, pitch and yaw angles. Our auto-calibration method has two distinct steps. First, the roll and pitch angles of the camera are estimated purely from a short image sequence without the need for targets or markers, or constraints on robot trajectory. Correct estimates of roll and pitch angles make it possible to track planar camera motion and we then acquire a camera trajectory featuring a short sequence of manoeuvres which is required for the second step of calibration. There, the remaining 4 DoF, the yaw angle and translations of the camera, are estimated by considering the estimated planar camera motion in the context of the robot's whole trajectory. There are two methods available here. In the most general case these 4 DoF can be calibrated by bringing in relative motion estimates from wheel odometry as an external motion.

Since synchronised and unbiased odometry may not be available on many platforms, we also present an alternative for the weaker but common case where a wheeled robot's motion can be assumed to be non-holonomic, showing that further 2 DoF (the x_{vc} coordinate of the camera position and the yaw angle) can be determined without the need for any source reference. A final extension of our calibration framework allows us to obtain a reasonable estimate of the y_{vc} camera coordinate without any precise measurements from wheel odometry, using only motion direction priors; i.e. simple information on whether the robot was moving forward or backward at each point on its trajectory. The last degree of freedom, the height of the camera above the plane which determines the overall scale of motion estimation, in this case needs to be estimated with one manual measurement (directly of the camera height, or perhaps by recognising a single known object or marking in the visual scene).

5.1 Vision-based calibration

In Section 4 we demonstrated how to track camera motion expressed in terms of a planar robot motion, provided that the full 6 DoF transformation between the robot and camera frames $\mathtt{T}^{cv}$ is given. Here we will present a method for calibrating the first 2 DoF of the full transformation, sufficient to allow planar camera motion tracking.

We first assume that the camera is aligned with the robot frame of reference, e.g. that $x_{vc} = 0$, $y_{vc} = 0$, $z_{vc} = 1$, $\gamma_{vc} = 0$, and therefore we have:

$$\mathtt{T}^{vc} = \left(\mathtt{R}^{vc} \mid (0,0,1)^{\top}\right) \quad . \tag{14}$$

Let us recall Eq. 1:

$$\mathtt{T}^{c_{lr}} = \mathtt{T}^{cv}\mathtt{T}^{v_{lr}}\mathtt{T}^{vc} \ .$$

Now, since the frames of references are aligned, we can think of $\mathtt{T}^{v_{lr}}$ as a camera planar motion, and there are only two unknowns in $\mathtt{T}^{vc}$, roll and pitch angles, $\mathbf{y} = (\alpha_{vc}, \beta_{vc})^{\top}$, that define the camera orientation matrix $\mathtt{R}^{vc}$.

By estimating camera orientation $\mathtt{R}^{vc}$ with respect to the ground plane we can simultaneously estimate the normal of the plane. Unlike [Silveira et al., 2008] we are not using inverse depth to estimate the normal, but instead we exploit the fact that the unit plane normal vector $\mathbf{n}$ is uniquely defined by the camera orientation $\mathtt{R}^{cv}$ and depends only on the roll and pitch angles.

$$\mathbf{n} = \mathtt{R}^{cv}\,(0,0,1)^{\top} \quad . \tag{15}$$

We can now readily parameterise the homography in Eq. 3 by two sets of parameters: the planar motion of the camera from frame to frame $\mathbf{x} = (x, y, \theta)^{\top}$, and the orientation of the camera with respect to the plane using roll and pitch angels $\mathbf{y} = (\alpha_{vc}, \beta_{vc})^{\top}$:

$$\mathtt{H}^{lr}(\mathbf{x},\mathbf{y}) = \mathtt{K}\mathtt{T}^{cv}(\mathbf{y})\mathtt{T}^{v_{lr}}(\mathbf{x})\mathtt{T}^{vc}(\mathbf{y})(I| - \mathbf{n}(\mathbf{y}))^{\top}\mathtt{K}^{-1} \ . \tag{16}$$

Using Eq. 14 and 15 we can simplify and rewrite the homography in Eq. 16 into the following form:

$$\mathtt{H}^{lr}(\mathbf{x},\mathbf{y}) = \mathtt{K}\mathtt{R}^{cv}(\mathbf{y})\mathtt{M}\mathtt{T}^{v_{lr}}(\mathbf{x})\mathtt{N}\mathtt{R}^{vc}(\mathbf{y})\mathtt{K}^{-1} \ , \tag{17}$$

with two constant matrices:

$$\mathtt{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \quad \text{and} \quad \mathtt{N} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} . \tag{18}$$

Similarly to Eq. 8 we choose to parameterise the homography by a transformation matrix $\mathtt{T}(\mathbf{x})$ that represents a small change to the estimate of planar motion $\hat{\mathtt{T}}^{v_{lr}}$, as well as by an update matrix $\mathtt{R}(\mathbf{y})$ representing a small update to the current estimate of camera orientation $\hat{\mathtt{R}}^{cv}$:

$$\mathtt{H}^{lr}(\mathbf{x},\mathbf{y}) = \mathtt{K}\hat{\mathtt{R}}^{cv}\mathtt{R}(\mathbf{y})\mathtt{M}\hat{\mathtt{T}}^{v_{lr}}\mathtt{T}(\mathbf{x})\mathtt{N}\mathtt{R}(-\mathbf{y})\hat{\mathtt{R}}^{vc}\mathtt{K}^{-1} \ . \tag{19}$$

For $\hat{\mathtt{T}}^{v_{lr}}$ we use the same update rule as in Eq. 9, whereas for the rotation matrix $\hat{\mathtt{R}}^{cv}$ the following update rule is applied:

$$\hat{\mathtt{R}}^{cv} \leftarrow \hat{\mathtt{R}}^{cv}\mathtt{R}(\hat{\mathbf{y}}) \ , \tag{20}$$

where $\hat{\mathbf{y}}$ belongs to the Lie Algebra $\mathfrak{so}(3)$ and represents a small update to the estimate.

By calculating the partial derivatives of homography $\mathtt{H}^{lr}$ with respect to its parameters we obtain the essential blocks necessary to apply the machinery of ESM introduced in Section 4. The partial derivative of $\mathtt{H}^{lr}$ around $\mathbf{x} = \mathbf{0}, \mathbf{y} = \mathbf{0}$ with respect to $\mathbf{x}$ is equal to:

$$\left.\frac{\partial \mathtt{H}^{lr}(\mathbf{x},\mathbf{y})}{\partial x_i}\right|_{\mathbf{x},\mathbf{y}=\mathbf{0}} = \mathtt{K}\hat{\mathtt{R}}^{cv}\mathtt{M}\hat{\mathtt{T}}^{v_{lr}}\frac{\partial \mathtt{T}(\mathbf{x})}{\partial x_i}\mathtt{N}\hat{\mathtt{R}}^{vc}\mathtt{K}^{-1} \ , \tag{21}$$

while the derivative with respect to $\mathbf{y}$ is expressed in the following form:

$$\left.\frac{\partial \mathtt{H}^{lr}(\mathbf{x},\mathbf{y})}{\partial y_i}\right|_{\mathbf{x},\mathbf{y}=\mathbf{0}} = \mathtt{K}\hat{\mathtt{R}}^{cv}\frac{\partial (\mathtt{R}(\mathbf{y})\mathtt{M}\hat{\mathtt{T}}^{v_{lr}}\mathtt{N}\mathtt{R}(-\mathbf{y}))}{\partial y_i}\hat{\mathtt{R}}^{vc}\mathtt{K}^{-1} \ . \tag{22}$$

Finally, we can use the fact that:

$$\left.\frac{\partial \mathtt{R}(\mathbf{y})}{\partial y_i}\right|_{\mathbf{y}=\mathbf{0}} = \mathtt{G}_i^{\mathbf{SO}(3)} \qquad \text{and} \qquad \left.\frac{\partial \mathtt{R}(-\mathbf{y})}{\partial y_i}\right|_{\mathbf{y}=\mathbf{0}} = -\mathtt{G}_i^{\mathbf{SO}(3)} \ , \tag{23}$$

where $\mathtt{G}_i^{\mathbf{SO}(3)}$ is the ith group generator for $\mathbf{SO}(3)$, and apply the product rule to show that:

$$\left.\frac{\partial(\mathtt{R}(\mathbf{y})\mathtt{M}\hat{\mathtt{T}}^{v_{lr}}\mathtt{N}\mathtt{R}(-\mathbf{y}))}{\partial y_i}\right|_{\mathbf{y}=\mathbf{0}} = \mathtt{G}_i^{\mathbf{SO}(3)}\mathtt{M}\hat{\mathtt{T}}^{v_{lr}}\mathtt{N} - \mathtt{M}\hat{\mathtt{T}}^{v_{lr}}\mathtt{N}\mathtt{G}_i^{\mathbf{SO}(3)} \ . \tag{24}$$

Having arrived at these expressions we can now efficiently use ESM to find the planar frame to frame camera motion as well as the normal of the plane parallel to which the camera is moving.

Although it is possible to estimate the plane normal from just two images, to better constrain and improve the robustness of the estimation process we combine multiple frames into a local dense map and jointly estimate the motion between consecutive keyframes as well as the parameters of the plane normal. When the overlap between the last keyframe and the current frame falls below a defined threshold we add this frame to the map. In practice for estimating roll and pitch angles we usually used maps containing between 5 and 50 frames.

5.2 Graph-based calibration

At this step of calibration, we assume that the roll and pitch angles were calibrated using the method described in the previous section, and therefore that we can accurately track camera motion with respect to the ground plane. Now the remaining 4 DoF, the camera position, (x_{vc}, y_{vc}, z_{vc}) as well as the yaw angle, need to be estimated in order to relate the camera motion to the robot motion. In the following we will refer to VO as the (up-to-scale) planar motion obtained from camera tracking using already calibrated roll and pitch angles only, and use WO to mean measurements coming from wheel odometry. Therefore all the motions considered in this section are assumed to be planar.

There are two distinct graph-based calibration methods presented: one relies on external motion estimation source, whereas the second method exploits general motion assumptions.

5.2.1 Calibration against wheel odometry

In this type of auto-calibration we are interested in finding a set of parameters $(x_{vc}, y_{vc}, z_{vc}, \gamma_{vc})$ that best explain the incremental measurements from VO and the corresponding WO measurements. Our formulation is similar to that proposed in [Brookshire and Teller, 2011] and is based only on incremental measurements, but we model the problem directly as a factor-graph (as proposed in [Kümmerle et al., 2011a]). This formulation allows us to benefit from all the techniques available for graph optimisation and solve the problem very efficiently.

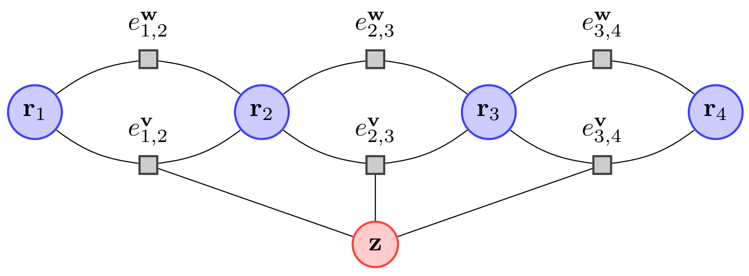

Figure 2: Factor graph formulation of our calibration against wheel odometry. Variable nodes $\mathbf{r}_i$ represent robot poses along the trajectory, and one additional node $\mathbf{z} = (x_{vc}, y_{vc}, z_{vc}, \gamma_{vc})$ represents the camera calibration. Factor nodes $\mathbf{w}_{i,i+1}$ (rectangles) with error functions $e^{\mathbf{w}}_i$ represent the measurement obtained from wheel odometry, whereas $\mathbf{v}_{i,i+1}$ with error function $e^{\mathbf{v}}_i$ are obtained from camera tracking.

An example factor graph is shown in Figure 2. In the graph the nodes $\mathbf{r}_i = (x_{r_i}, y_{r_i}, \theta_{r_i})$ represent robot poses along the trajectory, and one additional node $\mathbf{z} = (x_{vc}, y_{vc}, z_{vc}, \gamma_{vc})$ represents the camera configuration. Every time an odometry measurement is available we add a new node to the graph. We assume that camera and wheel odometry are well synchronised, and that jitter and synchronisation errors are negligible. Two consecutive nodes $\mathbf{r}_i$ and $\mathbf{r}_{i+1}$ are connected by two factors, one representing the measurement obtained by wheel odometry $\mathbf{w}_{i,i+1} = (x_{w_{i,i+1}}, y_{w_{i,i+1}}, \theta_{w_{i,i+1}})$ and the second the visual odometry measurement $\mathbf{v}_{i,i+1} = (x_{v_{i,i+1}}, y_{v_{i,i+1}}, \theta_{v_{i,i+1}})$. Visual odometry factors $\mathbf{v}_{i,i+1}$ are also connected to the camera configuration $\mathbf{z}$. There are no loops in this graph and the graph is fixed by the first node. In the graph we are primarily interested in determining camera calibration $\mathbf{z}$, though as the robot poses $\mathbf{r}_i$ are also variable, as a byproduct of the camera calibration we obtain an estimate of the robot trajectory.

The error function in the wheel odometry factors $e^{\mathbf{w}}_i$ measures how well the parameters $\mathbf{r}_i, \mathbf{r}_{i+1}$ satisfy the constraint arising from the wheel odometry measurement $\mathbf{w}_{i,i+1}$:

$$e^{\mathbf{w}}_i(\mathbf{r}_i, \mathbf{r}_{i+1}, \mathbf{w}_{i,i+1}) = (\mathbf{r}_{i+1} \ominus \mathbf{r}_i) \ominus \mathbf{w}_{i,i+1} \;, \tag{25}$$

where $\ominus$ is the inverse of the usual motion composition operator $\oplus$.

The error function in the visual odometry factors $e^{\mathbf{v}}_i$ measures how well the parameters $\mathbf{r}_i$, $\mathbf{r}_{i+1}$ and $\mathbf{z}$ satisfy the constraints arising from the visual odometry measurement $\mathbf{v}_{i,i+1}$:

$$e_i^{\mathbf{v}}(\mathbf{r}_i, \mathbf{r}_{i+1}, \mathbf{z}, \mathbf{v}_{i,i+1}) = (\mathbf{r}_{i+1} \ominus \mathbf{r}_i) \oplus f(\mathbf{z}, \mathbf{v}_{i,i+1}) \ , \tag{26}$$

where f is the function that transforms the motion from the camera frame of reference to the robot frame of reference using $(x_{vc}, y_{vc}, z_{vc}, \gamma_{vc})$.

The goal of the graph optimisation is to find a set of parameters $(\mathbf{r}^\star, \mathbf{z}^\star)$ that maximise the likelihood of the observed data, i.e. that minimise the total error in the graph:

$$\sum_i e_i^{\mathbf{v}\top} \hat{\Omega}_i^{\mathbf{v}} e_i^{\mathbf{v}} + \sum_i e_i^{\mathbf{w}\top} \Omega_i^{\mathbf{w}} e_i^{\mathbf{w}} \ , \tag{27}$$

where $\Omega_i^{\mathbf{w}}$ is the information matrix of the measurement $\mathbf{w}_{i,i+1}$ and $\hat{\Omega}_i^{\mathbf{v}}$ is the projection of the information matrix $\Omega_i^{\mathbf{v}}$ using the current estimate of $\mathbf{z}$. Least squares estimation on this hyper graph is performed using g^2o [Kümmerle et al., 2011b].

5.2.2 Calibration using practical assumptions

Nonholonomic calibration Calibration against wheel odometry works well in practice and is the most universal method. However, many mobile robots belong to a class of nonholonomic systems and therefore are subject to motion constraints, which can also be used for auto-calibration even when odometry is now available. We show that for a sensor that is placed on a nonholonomic mobile platform and that can measure its incremental motion in its own frame of reference, we are able to the estimate x_{vc} component of the sensor's position on the robot, as well as the yaw angle γ_{vc}, just using nonholonomic constraints without the need of wheel odometry or any other reference system.

The key observation here is that even though a planar trajectory $\mathtt{T}^c(t)$ measured by camera (or another sensor) in its frame of reference does not need to satisfy nonholonomic constraints, the resulting estimated robot trajectory,

$$\mathtt{T}^v(t) = \mathtt{T}^{vc}\mathtt{T}^c(t)\mathtt{T}^{cv} \ , \tag{28}$$

should satisfy these motion constraints. Therefore we are looking for calibration parameters that produce a nonholonomic vehicle trajectory from a camera trajectory. Note that here we consider the whole trajectories. A camera trajectory is created from a sequence of consecutive incremental planar motions, and therefore we used t to denote time in Eq. 28. The variables in the transformation $\mathtt{T}^{vc}$ from camera frame of reference to the robot frame of reference are now $(x_{vc}, y_{vc}, \gamma_{vc})$. In fact it turns out that for correct values of x_{vc} and γ_{vc} we can satisfy nonholonomic constraints independent of y_{vc} (see Appendix for more details), and therefore we cannot determine y_{vc} by means of this method. Additionally, using this algorithm we cannot estimate the absolute scale, unless the sensor can measure its motion metrically. Therefore we can only use this algorithm for estimating two degrees of freedom.

Our method is applicable to various types of nonholonomic wheeled robots (e.g. unicycle, differential drive or car-like robots), whose kinematic models are derived from the *rolling without slipping* assumption. Rolling without slipping means that the translational and rotational velocities of a rolling wheel are not independent. If the configuration of a robot at a time step t is represented by $\mathtt{T}^v(t) = (x_v, y_v, \theta_v)$, and its velocities are denoted by $\dot{x}_v$ and $\dot{y}_v$, this constraint can be expressed by the following formula:

$$\dot{x}_v \sin\theta_v - \dot{y}_v \cos\theta_v = 0 \ . \tag{29}$$

In other words, the nonholonomic constraint limits robots velocity in its current configuration and forces the vehicle to move tangentially to its main axis.

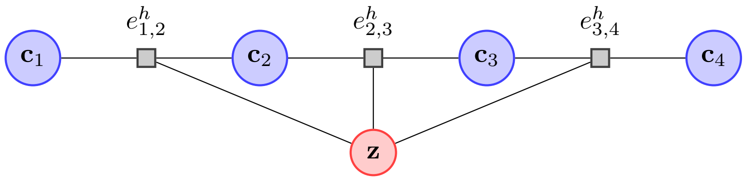

Figure 3: Factor graph formulation for nonholonomic calibration. Nodes $\mathbf{c}_i$ correspond to the camera poses and are fixed, whereas a single variable node $\mathbf{z} = (x_{vc}, \gamma_{vc})$ represents camera extrinsic parameters. Factor nodes $h_{i,i+1}$ (rectangles) with corresponding error functions, $e^h_{i,i+1}$, represent violations of nonholonomic constraints.

A graph-based formulation of this calibration strategy is depicted in Figure 3. Fixed nodes $\mathbf{c}_i$ represent camera poses along the trajectory, and there is only a single variable node in the graph, $\mathbf{z}(x_{vc}, \gamma_{vc})$, for camera extrinsics calibration parameters. The nodes are connected by factors $h_{i,i+1}$ that do not store any measurements but penalise deviation from nonholonomic motion $e^h_{i,i+1}$ using Eq. 29:

$$e^h_{i,i+1} = \dot{x}_{r_{i,i+1}} \sin\theta_{r_i} - \dot{y}_{r_{i,i+1}} \cos\theta_{r_i} \ . \tag{30}$$

As already mentioned, the nonholonomic error does not depend on y_{vc}. The robot poses $(x_{r_i}, y_{r_i}, \theta_{r_i})$ required for calculating the nonholonomic error are obtained from the camera poses $\mathbf{c}_i$ and current calibration parameters $\mathbf{z}$ using Eq. 28. To evaluate the error, it is also necessary to calculate the velocities from the poses. We estimate a velocity at every node by calculating the constant velocities that bring the robot from the current to the next pose. Although this is only an approximation, it works well in practice.

Again, least squares estimation on this graph is performed using g^2o.

Calibration using known direction of motion A limitation of nonholonomic calibration is that it does not provide enough constraints to estimate y_{vc}. However, one does not necessarily need to use a well-synchronised wheel odometry to overcome this shortcoming. Instead we show that using weaker constraints, namely information about whether the robot was moving forward or backward (i.e. whether its linear velocity v was positive or negative), is sufficient to constrain y_{vc}. This data can come e.g. from the commands the robot was supposed to execute, or from a very rudimentary odometry system. Obviously, a trajectory used for calibration with the help of this method should include a balanced number of sections of backward and forward motion. Also, correct values of x_{vc} and γ_{vc} are necessary at this stage.

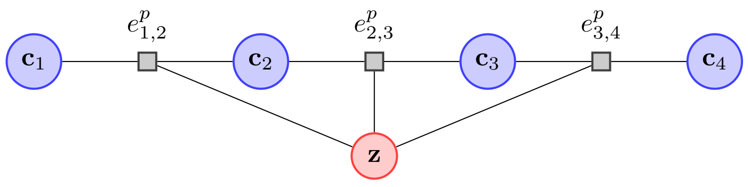

Figure 4: Factor graph calibration using motion direction priors. Factor nodes $p_{i,i+1}$ (rectangles) store information (sign) about whether the robot was moving forward (+) or backward (−).

The graph shown in Figure 4 is not significantly different from the nonholonomic graph and it mainly consists of fixed camera poses, $\mathbf{c}_i$, calculated from the incremental measurements. There is one additional node for camera extrinsic configuration $\mathbf{z}$. The factors in the graph, $p_{i,i+1}$ (denoted as rectangles), now store *prior* information (sign) about whether the robot was moving forward (+) or backward (−). To calculate the error in a factor we first need to calculate the robot linear velocity $v_{i,i+1}$ between two consecutive robot poses using the current estimate of the calibration parameters. The error depends on y_{vc} only (the parameter we want to estimate), though proper values of x_{vc} and γ_{vc} are required for the calculation. We use the following (heuristic) formula for error calculation in the graph:

$$e^p_{i,i+1} = \left\{ \begin{array}{ll} 0 & \text{if } \operatorname{sgn}(v_{i,i+1}) = prior_{i,i+1} \\ |v_{i,i+1}| & \text{otherwise} \end{array} \right\} . \qquad (31)$$

The graph optimisation framework is quite flexible and allows the possible inclusion of additional factors, e.g. representing smoothness of motion or constraints on the robot's dynamics that represents the limits of its acceleration and velocity.

Even though we formulated the calibration as a graph optimisation problem with a heuristic error function, it can be useful to think about it as classification task, where we are looking for a value y_{vc} that minimises the number of mismatches between the priors (forward/backward motion) and the motion direction calculated from visual odometry.

6 Experiments

6.1 Experimental configuration and ground truth reference

We have tested our approach on two different platforms in a series of experiments. Most of the evaluations were performed using a Pioneer 3 DX platform, a differential drive wheeled mobile robot, carrying an NVidia GPU-equipped laptop for real-time vision processing and with an adjustable rigid camera mount (Figure 5), and camera relatively close to the ground. We have used a Point Grey Plea2 Camera capturing greyscale images at 640×480 resolution and 30Hz frame-rate. To compensate for motion blur we reduce the shutter time and use automatic in-built camera settings for gain. The camera has a lens with approximately 80° field of view and is calibrated off-line for intrinsic parameters including significant radial distortion. Note that we undistort our images on the GPU at frame-rate before applying the method in this paper which assumes perspective projection. An additional evaluation was performed using data obtained from a rear parking camera, located at a height of about 1 metre, on a passenger vehicle travelling through an urban setting at speeds of up to 45km/h; this data set was collected by Renault and previously used in published work [Lovegrove et al., 2011]. The road vehicle data were also captured at 640×480 resolution and 30Hz frame-rate, but the camera field of view was approximately 45°.

Our method operates very comfortably in real-time on a modern PC with GPU, and this gives very good prospects for efficient embedded use on a parallel processing board or FPGA in the near future. The mean frame-rate of full resolution processing required for visual odometry is about 270–300 FPS (10 times real-time) on a desktop GTX480 GPU and about 60–65 FPS on a mid-range GT650M. The run-time for calibration depends on the number of frames used for calculating the camera orientation, and the length of trajectory used for estimating the remaining 4 degrees of freedom. In our test cases we were able to run continuous auto-calibration using 10 frames and hundreds to thousands poses at a rate of a few frames per second.

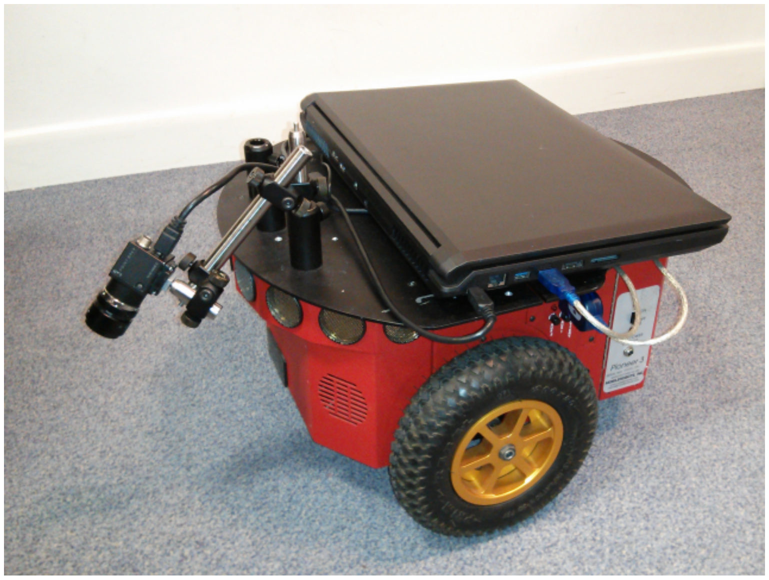

Figure 5: Robot Pioneer 3 DX used for most of the experiments.

Obtaining ground truth for our experiments was in general quite difficult. It is cumbersome to measure precisely a full 6 DoF camera pose as the camera frame of reference is hidden inside a camera housing. We also wanted to evaluate our method in different field conditions, with different camera poses, on various surfaces and lighting conditions, where setting up a ground truth system for tracking camera motion was not feasible.

The ground truth for extrinsics auto-calibration was determined using a standard camera calibration procedure that makes use of a chessboard pattern (similar to [Zhang, 1999]). We placed the chessboard pattern on the floor and first acquired a few images of it from different perspectives. Next, we mounted and fixed the camera on the robot and captured an image of the chessboard pattern again. The standard procedure estimates camera intrinsics as well as the 6 DoF of the camera with respect to the chessboard pattern. However, since the position of the chessboard pattern with respect to the robot frame of reference cannot be reliably measured, only 3 DoF can be used as our ground truth, namely the roll and pitch angles as well as the height above the chessboard pattern. These are equivalent to the roll and pitch angles and height of the camera in the robot frame of reference.

We also undertook to obtain ground truth for robot motion in order to evaluate the quality of visual odometry

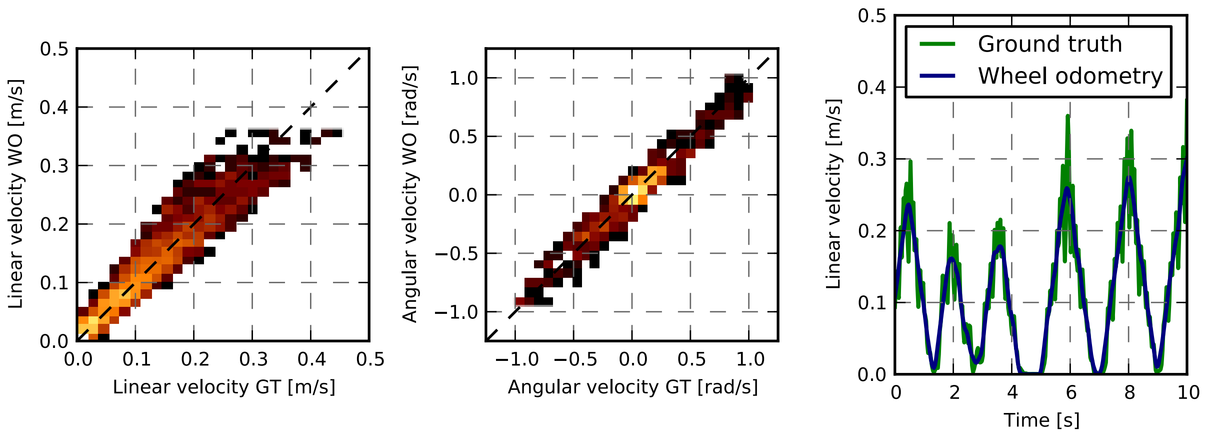

Figure 6: Comparison of wheel odometry against external camera-based ground truth for short motions of our indoor robot. We carry out a motion of several minutes, and for each new camera frame (at 30Hz) we obtain the incremental velocity and angular velocity measurements from both wheel odometry and the ground truth reference and plot points on two histograms. Colours in these 2D 'heat map' histograms indicate the frequency of a measurements. We see that the data are tightly clustered around the $x = y$ axis in both cases, indicating that the robot's wheel odometry is accurate and unbiased. In fact the robot has better local motion estimation accuracy as shown by the increased smoothness of the wheel odometry line in the short time series plot on the right.

once calibrated. For our indoor robotic platform we used the wheel odometry of the Pioneer robot either as a reference for most experiments.While wheel odometry of course is not usually considered suitable as a ground truth reference, this robot is heavy and precise and turns out to have locally very accurate wheel odometry on surfaces with good grip such as short carpet. To verify this and justify its use as the reference for the rest of our evaluation, we conducted an experiment where wheel odometry was compared against a further ground truth reference consisting of an external overhead calibrated camera system observing a target marker placed on top of the robot. The results are shown in Figure 6. What we see is that the Pioneer's wheel odometry is indeed accurate, with no observable bias in incremental motion estimation; in addition the robot has better local motion estimation accuracy as seen in the smoothness of the wheel odometry plot. We decided therefore to base our VO experiments on a comparison with this high quality wheel odometry rather than the external camera system, with the large advantage that this easily enables extended experimentation and long trajectories where providing an external ground truth reference might be more challenging and prone to error than the visual odometry itself.

In the experiments based on the road car dataset from Renault, we made use of the high quality ground truth trajectory data available from a PHINS system capable of estimating with high-accuracy the vehicle's position and orientation. This system consists of a precise Inertial Measurement Unit made of 3 fibre optic gyroscopes and 3 pendulum type accelerometers, a bi-frequency GPS receiver and the vehicle wheel odometry.

6.2 Visual odometry evaluation

Having validated our ground truth reference, we now present qualitative results of an evaluation of the visual odometry results using the Pioneer 3 DX robot. Probably the simplest and most intuitive demonstration of the performance of our auto-calibrating visual odometry is a plot that shows superimposed trajectories generated both by wheel and visual odometry, as depicted in Figure 7. We see a good overall performance over significant distances apart from bigger errors caused by low-lighting conditions and self-shadowing, issues we will examine later. This can be compared with a trajectory generated by an uncalibrated visual odometry, when the camera extrinsics were off just by a few millimetres and degrees.

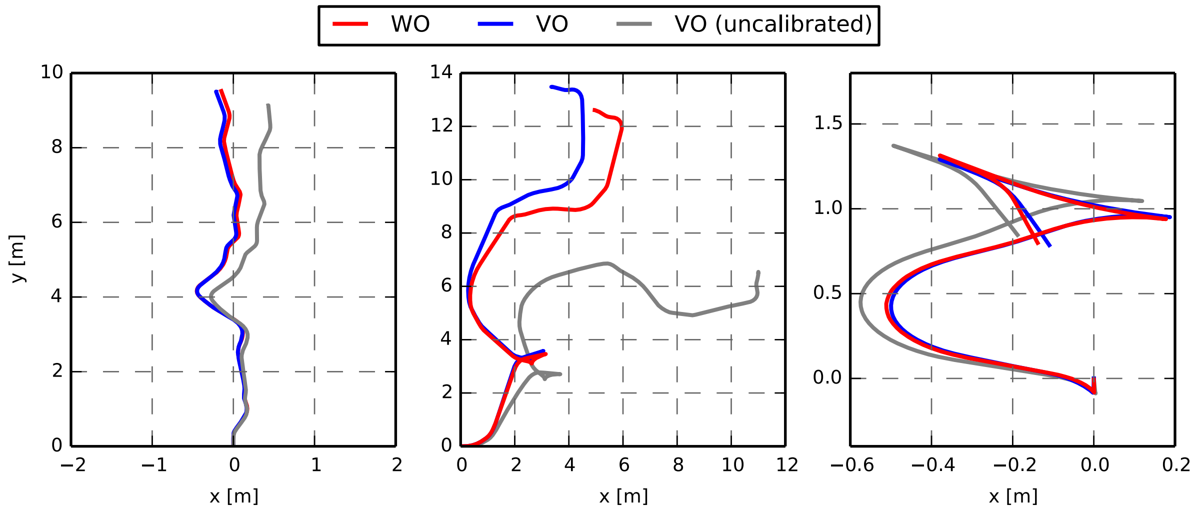

Figure 7: Example of different trajectories estimated by WO and calibrated and uncalibrated VO. The uncalibrated trajectories were generated using only a manual calibration, and the camera extrinsics were off by only ±10 mm and ±2.5°. The increased error in the middle figure is due to significant self-shadowing and low lighting conditions.

Additionally, Figure 8 shows examples of planar mosaics that can be created as a side product of our method. A high quality mosaic indicates both correct estimation of the planar camera motion and of the out-of-plane angles. Note that these mosaics were created using incremental motion only; there is no loop-closure or global optimisation.

As both odometry systems will eventually drift over time, observing trajectory plots is not a strong basis for quantitive evaluation of VO quality. Instead, for quantitative evaluation we analyse the statistics of incremental motion estimation accuracy. Figure 9 summarises the behaviour of VO against our validated WO 'ground truth' in form of 2D histograms. These histograms were created from more than 350 metres

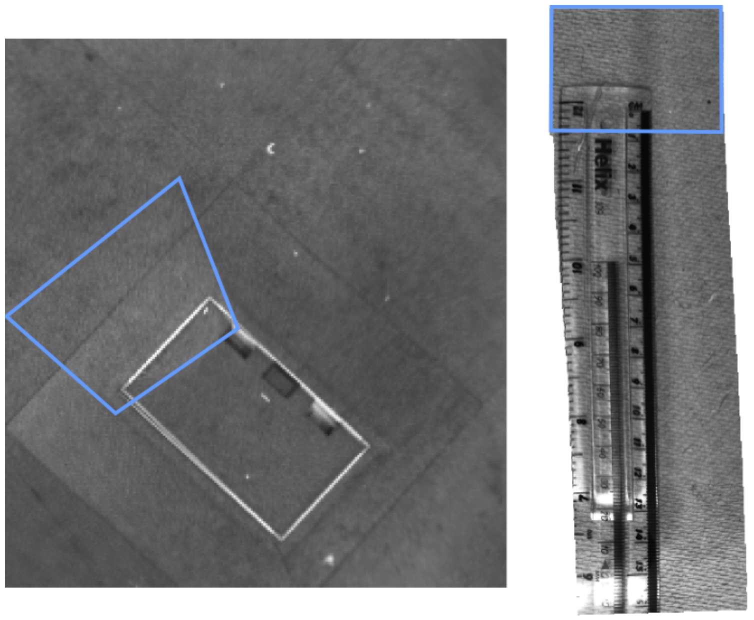

Figure 8: Consistent planar mosaics created by simple stitching of frames using the homographies estimated by our visual odometry system. The field of view of a single camera frame is superimposed for comparison.

of motion (about 45000 frames, 25 minutes) that featured a wide range of motions and turns up to the maximum dynamics of the robot (peak linear velocity 0.8 ms^{-1}, average 0.2 ms^{-1}; peak angular velocity 1.5 rads^{-1}, average 0.24 rads^{-1} respectively) on various surfaces and using 10 different camera configurations with camera heights from 7 cm to 25 cm. At the beginning of every individual test we performed a short auto-calibration using our method. In the histograms each point represents the estimated velocity (linear and angular on the left and right respectively) plotted against ground truth, with a heat map showing the frequency of each pair of values. The tight packing of these distributions around the $x = y$ axis indicates the unbiased and low uncertainty estimation of robot motion and the good performance of the algorithm most of the time. In the velocity range of up to around 0.6 ms^{-1} we see extremely robust performance. Beyond this speed we see a very small fraction of gross errors that result mostly from a combination of disadvantageous effects like, reduced inter-frame overlap, motion blur, shadows cast by the robot or low-light conditions.

6.3 Calibration on different surfaces

In order to evaluate the performance and stability of our auto-calibration algorithm three different camera configurations (Figure 10) we tested the procedure on quite different surface types (Figure 11). In some cases the assumptions about a planar scene and the planarity of the robot motion were violated. Each individual run was on average 15–20 metres long, but we subdivided every trajectory into multiple, overlapping, approx-

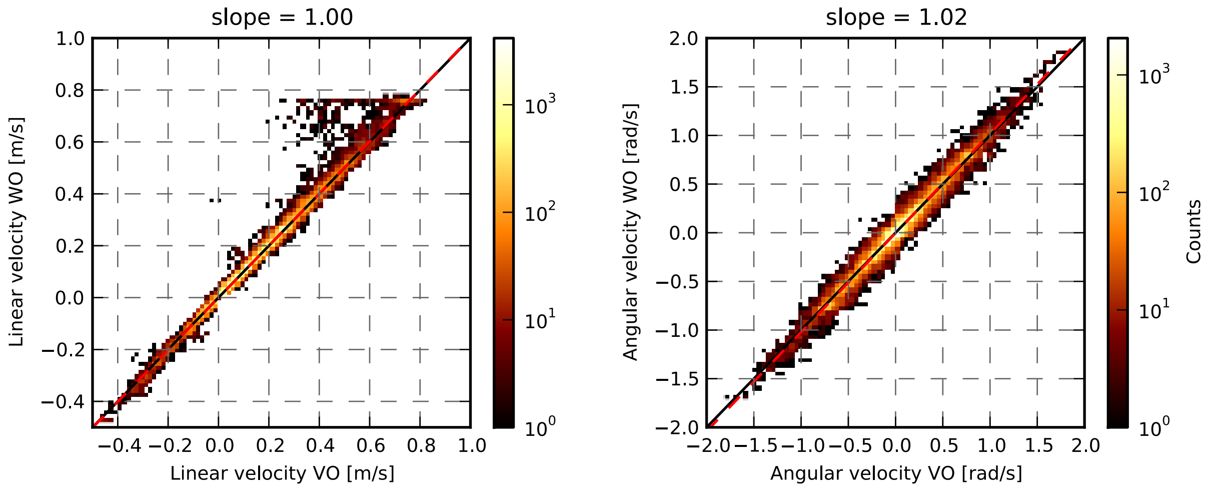

Figure 9: 2D heat-map histograms showing the statistical performance of auto-calibrated visual odometry against validated WO ground truth. Histograms were created using about 45000 data points, note the log scale used for the heat map; gross errors are in fact extremely rare and the distribution is tightly packed about the $x = y$ line indicating unbiased and precise performance: the red dashed line is a fit to our data which is almost perfectly aligned with the desired black line. Without a systematic error we expect the slope of the fitted line to be equal to 1, and indeed we obtain a slope of 1.00 for the linear velocity and 1.02 for the angular velocity.

imately 4 metre long trajectories starting with an every new frame. This allowed us to perform a statistical analysis and provide distributions of the parameters, as from every individual run we had thousands of data points. For every sub-trajectory we estimated the roll and pitch angle of camera using the first 10 frames, whereas the remaining degrees of freedom were calculated using the rest of available robot trajectory (every time we started with the same initial conditions, $x_0 = 200$ mm, $y_0 = 0$ mm, $z_0 = 200$ mm, $\alpha_0 = 0°$, $\beta_0 = 0°$, $\gamma_0 = 0°$).

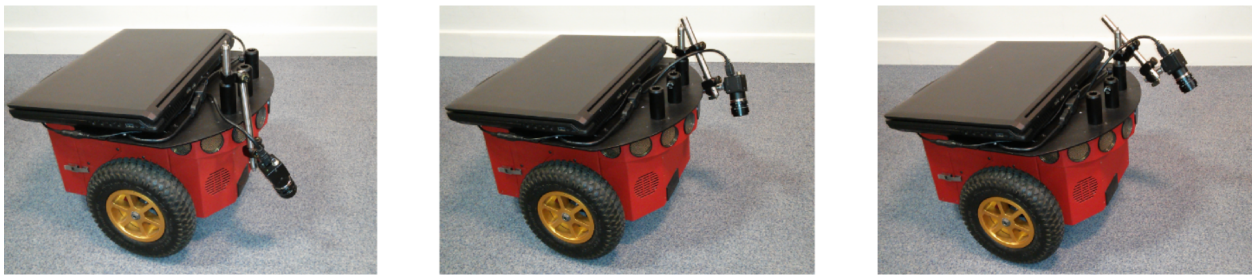

Figure 10: We performed verification of the auto-calibration for 3 different camera poses.

Tables 1, 2 and 3 show the results from our auto-calibration method together with the ground truth measurements for three different camera configurations. For each individual configuration the estimates converge to the same results independent on the surface and in multiple trials and we can conclude that the algorithm

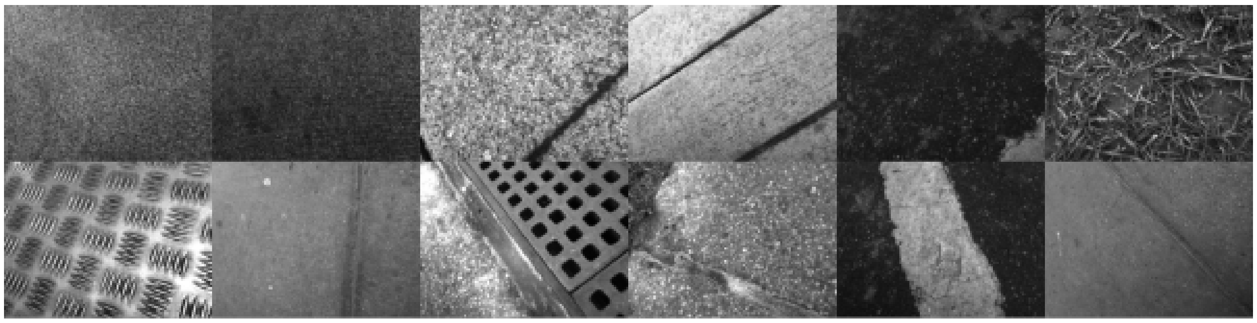

Figure 11: We tested our system on various different surfaces, both indoors and outdoors.

is stable. Only on a very uneven surface (sequence ‘outdoor grass’) do the estimates diverge from the rest of the measurements. The estimates of the camera height, as well as the orientation, seem to be most stable, as indicated by a relative low variance, whereas the estimates of the x_{vc} and y_{vc} coordinates are more noisy.

Table 1: Calibration on different surfaces, camera configuration 1. About 1000-2000 subtrajectories were used on each surface separately for calculation of mean and standard deviation. Ground truth values were obtained using chessboard pattern method described in Section 6.1.

	x [mm]	y [mm]	z [mm]	roll [°]	pitch [°]	yaw [°]
ground truth			**178.7**	**12.4**	**17.6**	
carpet 1	244.1 ±4.3	-18.5 ±5.2	179.1 ±0.8	12.3 ±0.3	17.7 ±0.4	-9.2 ±0.5
carpet 2	241.8 ±2.1	-16.9 ±2.5	178.9 ±0.6	12.3 ±0.3	17.5 ±0.3	-9.2 ±0.3
outdoor 1	245.7 ±5.9	-18.9 ±4.5	178.5 ±0.9	12.1 ±0.9	17.5 ±1.3	-9.5 ±0.2
outdoor 2	250.2 ±3.4	-12.9 ±3.4	180.1 ±0.4	12.2 ±1.2	17.1 ±1.7	-9.7 ±0.2
outdoor 3	244.3 ±2.1	-19.2 ±2.2	178.1 ±0.5	12.0 ±1.4	17.4 ±1.9	-9.3 ±0.2
outdoor 4 (grass)	254.2 ±9.5	-3.6 ±17.8	174.1 ±6.9	11.9 ±2.0	17.3 ±2.8	-10.0 ±0.7

Table 2: Calibration on different surfaces, camera configuration 2. About 1000-2000 subtrajectories were used on each surface separately for calculation of mean and standard deviation. Ground truth values were obtained using chessboard pattern method described in Section 6.1.

	x [mm]	y [mm]	z [mm]	roll [°]	pitch [°]	yaw [°]
ground truth			**216.4**	**29.8**	**-4.6**	
carpet 1	191.7 ±2.1	-107.8 ±3.6	218.1 ±0.7	30.1 ±1.0	-4.4 ±0.6	75.8 ±0.3
outdoor 1	195.5 ±1.2	-104.6 ±1.8	218.3 ±0.2	29.7 ±0.8	-4.7 ±0.7	75.9 ±0.1
outdoor 2	198.7 ±2.7	-106.0 ±1.6	218.5 ±0.9	29.5 ±1.3	-4.7 ±1.5	75.6 ±0.4

The most stable results in terms of camera orientation are obtained on a carpet surface, when the surface on which the robot is driving is mostly flat. On the other hand outdoors, where the robot was driving over rough and slightly uneven surfaces, the variance of the estimates increases; this is caused by a locally nonplanar

Table 3: Calibration on different surfaces, camera configuration 3. About 1000-2000 subtrajectories were used on each surface separately for calculation of mean and standard deviation. Ground truth values were obtained using chessboard pattern method described in Section 6.1.

	x [mm]	y [mm]	z [mm]	roll [°]	pitch [°]	yaw [°]
ground truth			**146.9**	**-26.6**	**4.7**	
carpet 1	222.6 ±4.6	80.6 ±4.5	148.0 ±0.7	-26.8 ±0.4	4.5 ±0.3	-18.6 ±0.3
outdoor 1	231.0 ±2.6	78.5 ±4.2	148.6 ±1.4	-26.4 ±1.1	4.3 ±1.1	-18.5 ±0.2
outdoor 2	226.6 ±1.4	79.7 ±1.6	149.3 ±0.4	-26.4 ±1.8	4.5 ±1.3	-18.2 ±0.3

motion and nonplanar structures in the field of view. As we already showed in Figure 7 even relatively small miscalibration can eventually lead to significant errors in visual odometry. Therefore it is recommended to perform the auto-calibration on mostly flat surfaces and with moderate robot velocities, and to increase the number of frames used for calculating camera orientation to improve the stability.

Whereas the process of estimating camera orientation with respect to the plane of motion is independent of the robot's trajectory, estimation of the remaining degrees of freedom (camera metric position and the yaw angle) is governed by other rules. Firstly, it depends on proper frame-to-frame tracking and therefore proper values of camera orientation. If the first step of our auto-calibration is unsuccessful, it cannot be compensated by the graph optimisation and it is impossible to recover unbiased values of the 4 DoF. Secondly, certain trajectories are degenerate and prevent unambiguous calibration. As in [Brookshire and Teller, 2011], calibration cannot be recovered when the robot is driven in a straight line only, or when the camera experiences only concentric, circular motion. In practice these cases are easily avoided by varying the robot's velocities and the trajectory geometry.

Figure 12 illustrates the behaviour of estimates as time evolves for the same configuration on two different surfaces. Here every time point represents another trajectory segment. We observe in general stable behaviour, and see that the roll and pitch angles are independent of the trajectory. Sudden jumps in the values of x_{vc}, y_{vc}, z_{vc} and yaw angle correspond to different robot motions occurring during trajectory segments, which impose slightly different constraints on the calibration graph. Even though the 4 metre long trajectory-segments used in this evaluation were enough to avoid degenerate cases, we generally observed that longer trajectories are desirable for more stable estimates of x_{vc}, y_{vc}, z_{vc} and the yaw angle γ_{vc}.

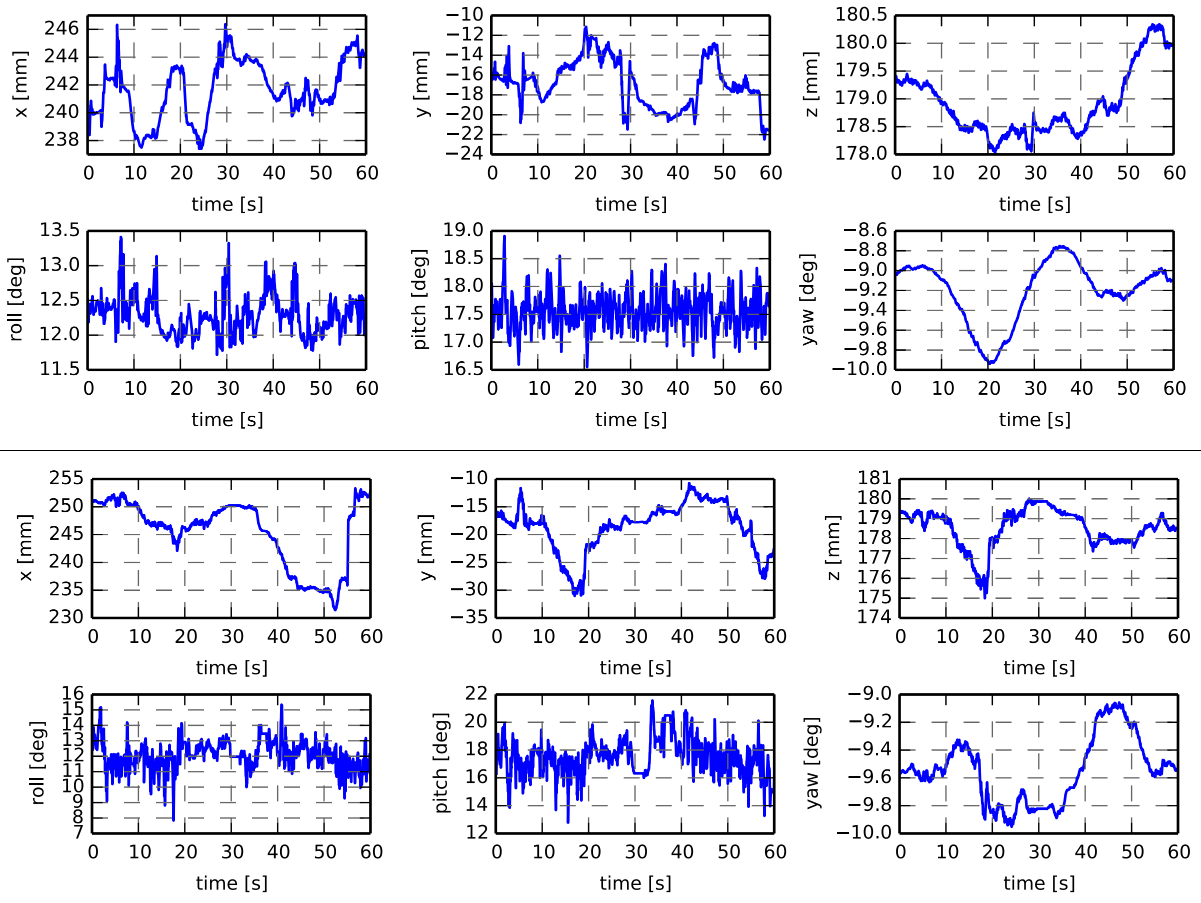

Figure 12: The development over time of our estimates of extrinsics parameters of camera configuration 1 on the 'carpet 1' (top) and 'outdoor 1' (bottom) surface when auto-calibration is run online. Every time point represents another trajectory segment. The values of the roll and pitch angles are stable and independent of the robot trajectory. On the other hand, the remaining degrees of freedom change as different trajectory segments are used.

6.4 Odometry from a road vehicle's rear parking camera

Next we evaluated our auto-calibration system in a completely different settings, using a video stream that was captured from a standard rear parking camera as a vehicle was moving in an urban environment. The data comes from Renault [Lovegrove et al., 2011]. The camera was placed at about 1 metre above the ground, a position significantly higher than in the experiment with the Pioneer robot, and was viewing the road surface directly behind the vehicle. Some of the frames captured are depicted in Figure 13. Most of the time the camera observed a flat road surface, but in some parts of the images nonplanar structures are visible. These were mostly pavement (sidewalk) or other passing vehicles. Out test vehicle also experienced some nonplanar motion, when it was turning at high speed and driving over speed bumps. Consequently, this data set allows us to examine the effects of violating planar assumptions both with respect to the planarity of the motion as well as the planar-environment assumption. For the experiment we have full, high-accuracy, ground truth for visual odometry obtained from a PHINS system. Ground truth for the calibration was obtained in a similar way as for the indoor robot.

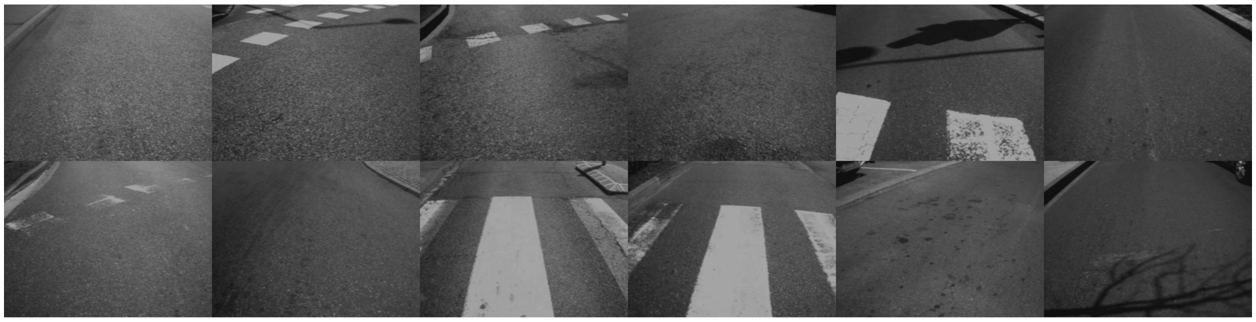

Figure 13: Examples of frames obtained from a parking camera. Most of the time the camera observes a flat road surface, with small fraction of images containing nonplanar structure, e.g. pavement or passing cars.

At the beginning of the evaluation we ran our standard calibration procedure: first camera orientation was estimated using a short sequence of frames and next the remaining 4 DoF were calculated using a short sequence of the trajectory (about 200 metre long). After the camera extrinsics were available, we executed our planar visual odometry on the whole 2.5 km-long trajectory. The results are summarised in the form of a 2D histogram in Figure 14. Additionally, in Figure 15, we present a short sequence of velocities measured by visual odometry and the PHINS ground truth system in the form of a time series'.

The estimation of the linear velocity is accurate and unbiased, as indicated by tight packing of the points along the $x = y$ axis in the histogram. However, there is a clearly observably systematic error in the measured angular velocity. This is confirmed in the time series included in Figure 15 where we can recognise that the visual odometry over-estimates the angular velocity. This can indicate that when the vehicle is turning, the camera is locally experiencing a nonplanar motion and the estimated orientation is not valid. To confirm this observation we run an additional evaluation, where the orientation of the camera with respect to the plane of motion was continuously estimated using the five most recent frames, and we calculate the planar motion with respect to the locally estimated plane. As depicted in Figs. 16 and 17 this procedure helps to reduce the systematic error in angular velocity, but it does not eliminate it completely. As a bonus, we obtain less noisy estimation of the linear velocity, although it comes at the extra cost required for performing continuous calibration.

Figure 18 provides an insight in what is happening when the vehicle is performing a turn, and helps to explain why we observe a systematic error in angular velocity. There we have plotted angular velocity measured

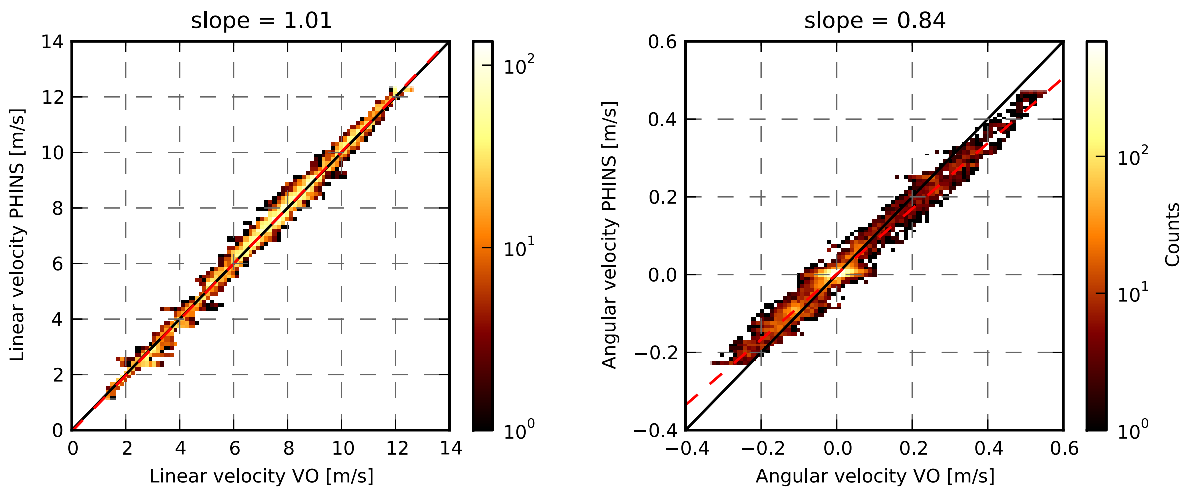

Figure 14: 2D histograms of the velocities measured by auto-calibrated visual odometry and the ground truth system. The frequency of a measurement is represented by colour in the heat map. We observe unbiased estimation of the linear velocity (left). The dashed red line represents a straight line fit to the data, and in this case it is perfectly aligned with the $x = y$ line. There is a systematic error measured in the angular velocity (right) as the fitted line diverges from the $x = y$ line. This is also confirmed by the slope of 0.84 for the line fitted to the measurements. Histograms were created using about 10000 data points.

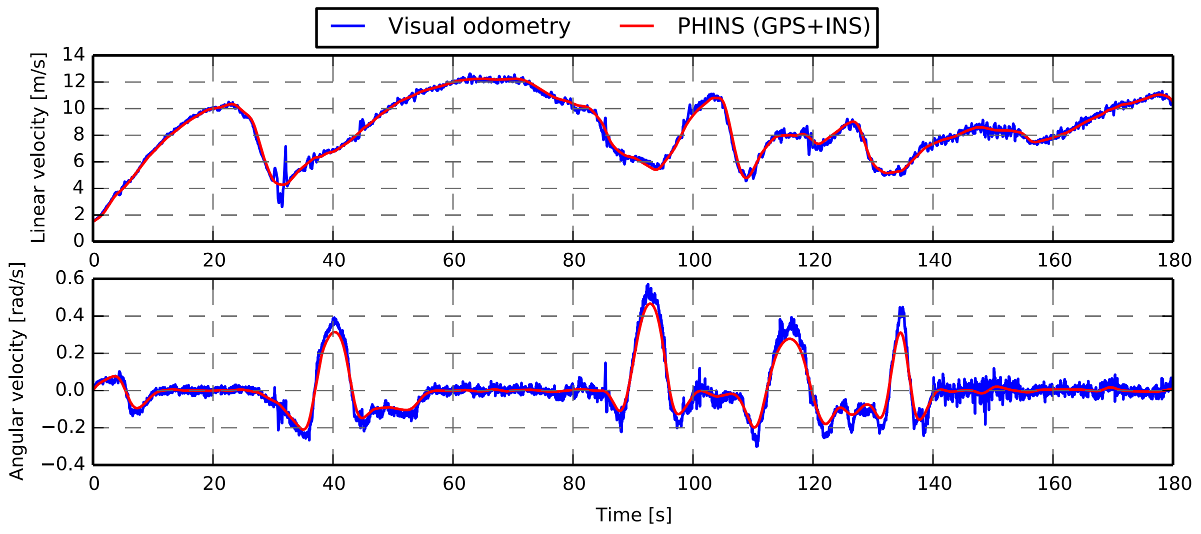

Figure 15: Comparison of the velocities measured by visual odometry and the PHINS ground truth system for the first 3 minutes of the sequence. On one hand we see accurate estimation of the linear velocity with only a small error around 30–35 seconds of the sequence, when the vehicle is driving over a speed bump. On the other hand, when the vehicle is performing turning manoeuvres, the visual odometry systematically overestimates the angular velocity of the vehicle.

by the ground truth system against the camera roll angle obtained during the continuous auto-calibration. The roll angle varies in a range of about $\pm 2.5°$ and is proportional to the current vehicle angular velocity. We suspect that as the vehicle is turning at a relatively high speed, vehicle suspension causes the car to lean to the side and consequently change the camera roll angle to violate the planar motion assumption.

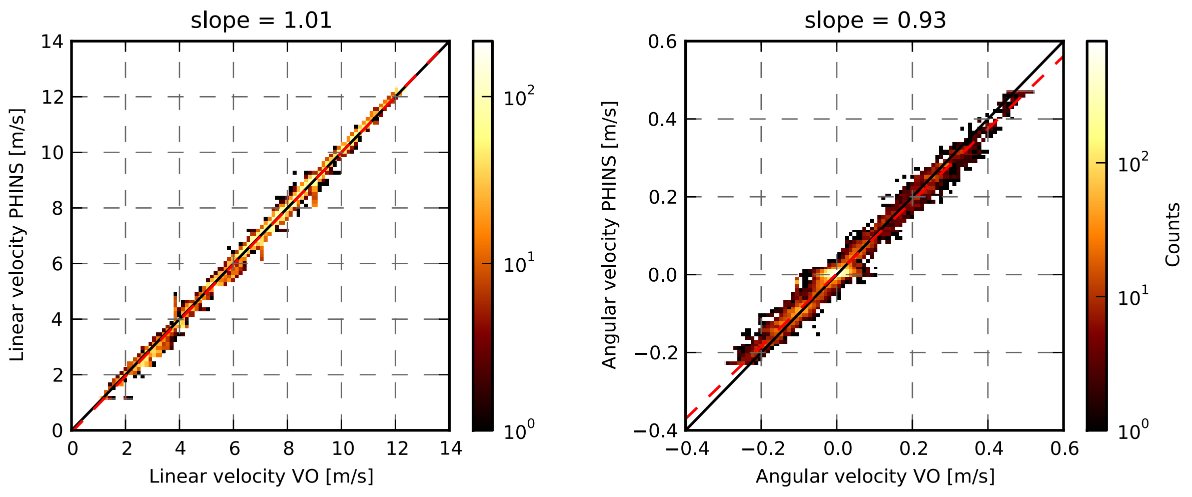

Figure 16: Estimating a local plane and tracking motion with respect to it seems to improve the performance of the visual odometry, accounting for the roll of the road vehicle during high speed turns. We still see unbiased and even more accurate estimation of the linear velocity. The systematic error in the angular velocity is reduced but not eliminated, as the slope of the line fitted is now 0.93 compared to 0.84 when global settings were used (note that a slope of 1 would indicate an unbiased behaviour). Histograms were created using about 10000 data points.

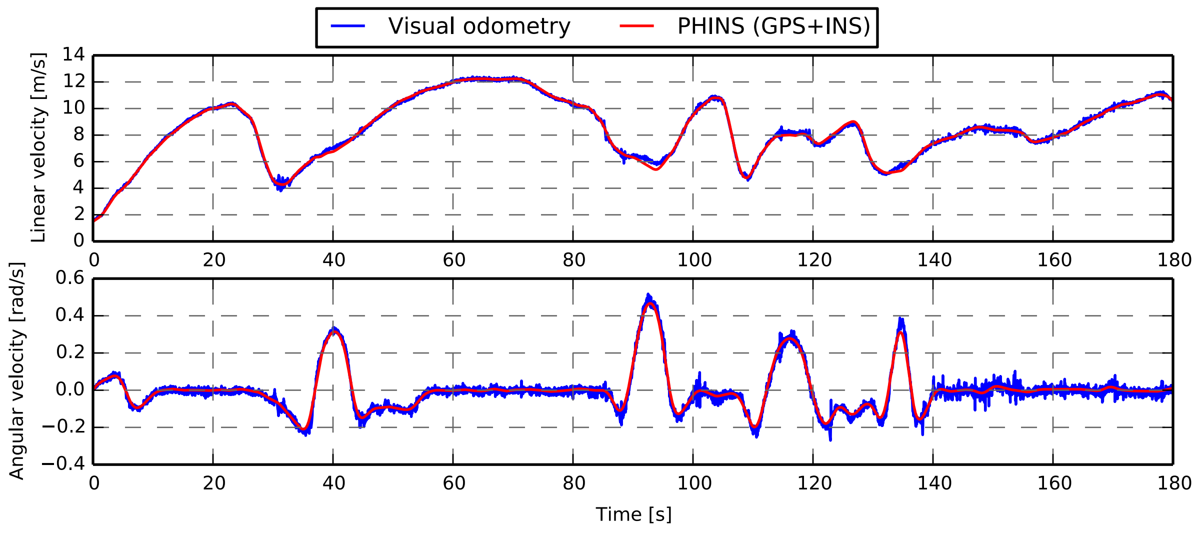

Figure 17: Improvement in the tracking obtained by continuous calculation of a local plane of motion is also visible in the time series. The estimation of the linear velocity is smoother and less noisy, whereas the angular velocity is no longer permanently overestimated.

One could argue that we should be able to compensate for this systematic behaviour without the need of a continuous auto-calibration, but by applying a simple proportional correction factor defined by the line shown in Figure 18; however, we did not evaluate this approach further.

We also analysed the behaviour of the pitch angle during the test drive and were expecting to see a similar

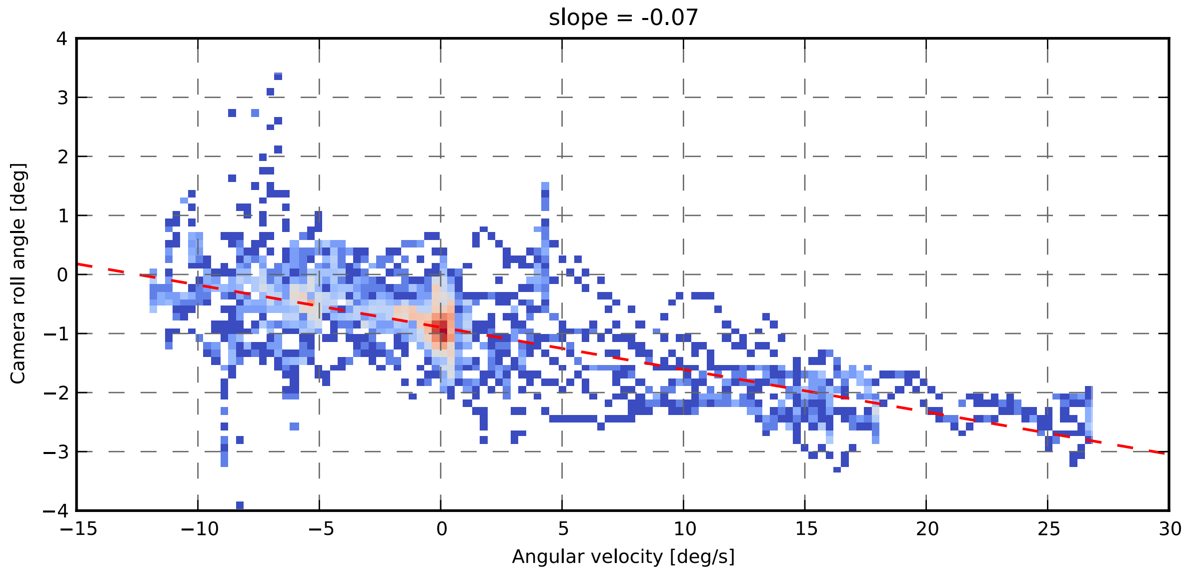

Figure 18: Angular velocity against a measured camera orientation with respect to the local plane of motion. The roll angle varies in the range of about ±2.5° and is correlated with the angular velocity. The red dashed line represents the fit and the slope of the line −0.07 can be understood as a measure of the dependence between these two parameters.

pattern during acceleration and braking of the vehicle. However, this was not confirmed by the data. Apparently, the vehicle was not accelerating at high enough rates to cause a significant front/back tilt of the camera.

A change in the camera pitch angle is observable in another situation, namely when the vehicle was driving over speed bumps. This occurs for example at around 30–35 seconds of the time sequence (see Figure 15) and is manifested by an increased error in both linear and angular velocity. Continuously estimating the camera orientation improves the performance of the tracking, which is best visible in Figure 19. There we present a close-up of the two situations when the vehicle is driving over speed bumps. The pitch angle varies significantly in a range of about ±10°, and especially in the first sequence a clear pattern is visible: the first oscillation is due to the front of the vehicle passing the bump, and the second oscillation is due to the rear wheels driving over the bump. Obviously, the orientation of the camera is changing from frame to frame, so it is important to remember that estimated roll and pitch angles are only an approximation for the five most recent frames.

When the car is driving over speed bumps we can observe yet another effect, a violation of the planar environment assumption, which we will analyse now. As already mentioned and shown in Figure 13 in many frames nonplanar structures were visible from the camera. In most situations these elements were a

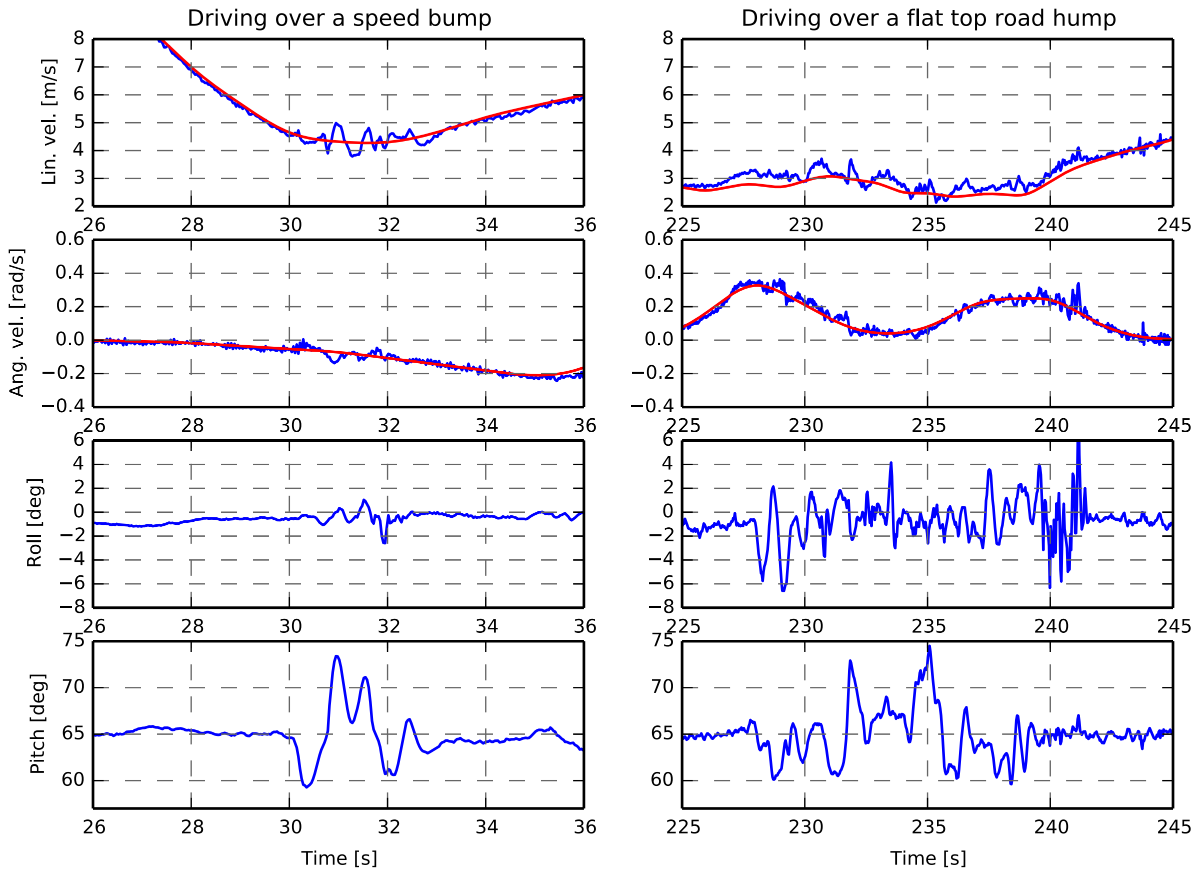

Figure 19: Driving over speed bumps causes a violation of the planar motion assumption. However we can detect a change in the pitch angle and by estimating a local plane of motion, and at least partially compensate for errors.

pavement or other car passing by, and occupied only a small fraction of the image, usually close to the image borders. Since we are using a robust norm, our system can handle these situations to some degree and we did not observed any particular systematic error due to small fractions of nonplanar object in the scene. Figure 20 shows a frame where a large fraction of the image is occupied by a passing car and also illustrates which pixels were automatically marked as outliers during the iterative reweighted least-squares estimation of planar frame-to-frame motion (Section 4). Obviously, in this situation we also benefit from a large image overlap between consecutive frames and the mostly uniform texture of the nonplanar object.

Structures visible in the camera field of view, which do not belong to the main plane of motion, do not affect significantly the auto-calibration. To illustrate that, we ran auto-calibration twice on a short sequence where some part of a pavement was continuously visible. An example frame of this sequence is depicted in the top left corner in Figure 13, and the pavement occupies about 10% of the image. In the first run we were using the whole image, and the second time we manually masked and ignored the nonplanar part of the images.

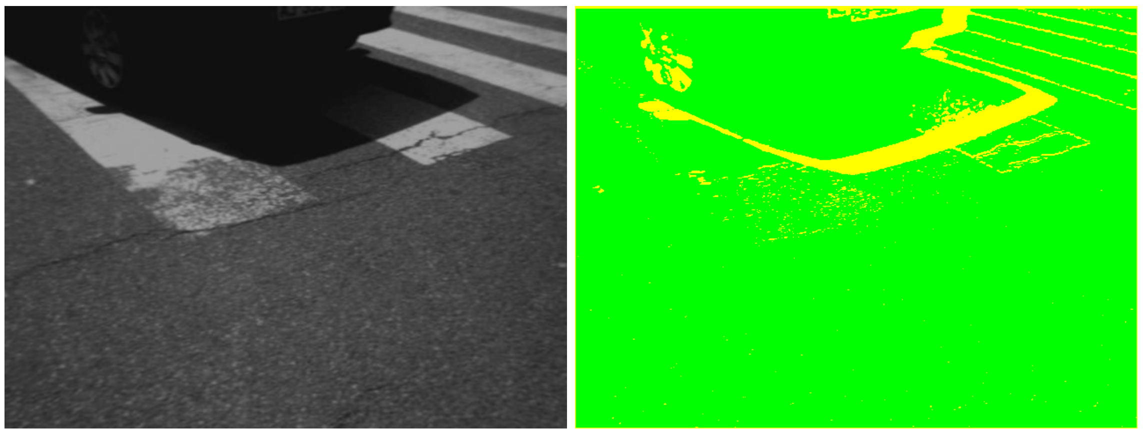

Figure 20: Left: An example of a frame where a significant fraction of the image is occupied by a nonplanar structure. Right: During the iterative reweighted least-squares estimation of the frame to frame motion (Section 4), some pixels are automatically marked as outliers (yellow) by the robust norm used in the cost function.

Figure 21 shows that there are no obvious differences in the roll and pitch angle estimated during these two runs. A pavement (sidewalk) has quite similar appearance to the road surface and the camera moves parallel to it as well. We could probably expect more significant degradation of the performance auto-calibration with more non-road structures, e.g. walls, trees, in the camera field of view. As already pointed out, it is recommended to perform calibration on under conditions that satisfy the assumptions.

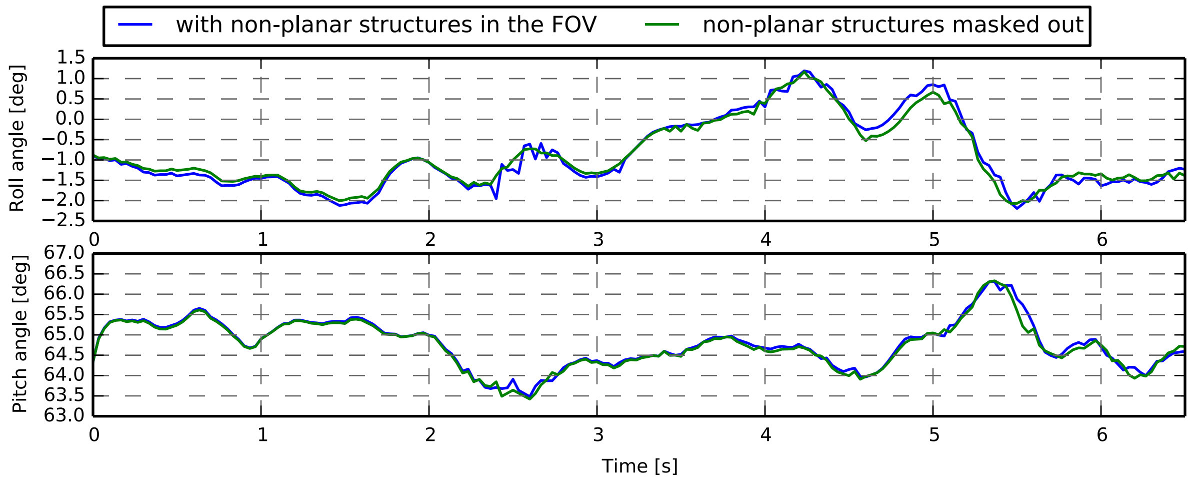

Figure 21: Calibration with nonplanar structures in the field of view. We do not observe a significant difference between the values obtained on a sequence where a nonplanar structure was visible all the time, and the same sequence but with the nonplanar structures manually masked out.

The only situation when our auto-calibrating visual odometry failed during this test case happened where the effects of significant nonplanar motion and nonplanar structures were combined with self-shadowing. We also tested our system on video sequences available in the KITTI Vision Benchmark Suite [Geiger et al., 2012] and the Malaga Stereo and Laser Urban Data Set [Blanco et al., 2014], but given that we have chosen a different computer vision operating point our approach is not designed for this data, as the camera is looking forward and not downward resulting in mostly nonplanar structures in the field of view, with little texture available on the road surface. In addition the frame rate of this data, 10 Hz for KITTI, and 20 Hz for Malaga, is too low compared to the camera motion for a dense method to work reliably.

For the completeness of our evaluation in Table 4 we include results of the same test as in Section 6.3. Here we divided the whole 2.5 km long trajectory in a sequence of shorter, approximately 200 metre long sections and run the auto-calibration on each individual sequence. As already pointed out, the estimated values of the roll and pitch angles vary along the trajectory, but in general the values match the ground truth parameters.

Table 4: Mean and standard deviation of estimated roll and pitch angles, as well as the height of the camera.

	GT	estimate
roll [°]	0.0	-0.87 $\pm$0.68
pitch [°]	66.0	65.02 $\pm$1.35
height [mm]	1015	1031.99 $\pm$8.51

6.5 Calibration using nonholonomic constraints and direction of motion

In the final experiment we tested the ability of the method that exploits the nonholonomic constraints of a robot and motion priors to accurately estimate the calibration of the camera. This calibration has two stages: first, a nonholonomic graph is built and estimates of x_{vc} and yaw angle, γ_{vc} are produced. Here, no source of reference is used. Next, a second graph is constructed, where we use the already estimated x_{vc} and γ_{vc} together with information about the direction of motion to determine the remaining degree of freedom, the y_{vc} coordinate of the camera pose in the robot frame of reference. For both graphs the same sequence of camera motion can be used. We assumed that the height of the camera is known (we used the values obtained using the standard procedure as it makes comparisons easier), and that the roll and pitch angles were known and the frame-to-frame tracking was correct.

For three different camera configurations (the same as in Section 6.3, on the surface ‘carpet 1’) we compare the estimates obtained from the standard calibration procedure that uses wheel odometry with the nonholonomic calibration. As with the experiments on different surfaces (Section 6.3), we divided the longer trajectory into a set of shorter, overlapping trajectories. This way we obtained hundreds of test trajectories that allow us to perform a statistical analysis of the performance of the proposed method. For all trajectories the initial conditions were the same, $x_0 = 200$, and $\gamma_0 = 0$. The results are presented in Table 5[1].

Table 5: Results of the calibration using the standard procedure that relies on wheel odometry and nonholonomic calibration. Note a very good agreement between the parameters obtained using the two independent strategies for three different camera configurations.

	x [mm]	yaw [°]
config. 1: standard	241.7 ±5.5	-9.2 ±0.6
config. 1: nonholonomic	241.6 ±5.4	-9.2 ±0.5
config. 2: standard	197.9 ±1.2	75.4 ±0.1
config. 2: nonholonomic	198.6 ±2.7	75.4 ±0.1
config. 3: standard	223.0 ±5.8	-18.4 ±0.4
config. 3: nonholonomic	224.9 ±5.4	-18.4 ±0.4

We obtained a remarkable agreement between the values calculated using the two different calibration strategies. Figure 22 shows how the estimates evolve over time for one of the tested camera configurations. We see that the time series behave quite similarly and the difference in the estimated x_{vc} values can be ascribed to the fact that in the standard calibration procedure the height is estimated continuously, whereas for the nonholonomic calibration a constant value of camera height was used.

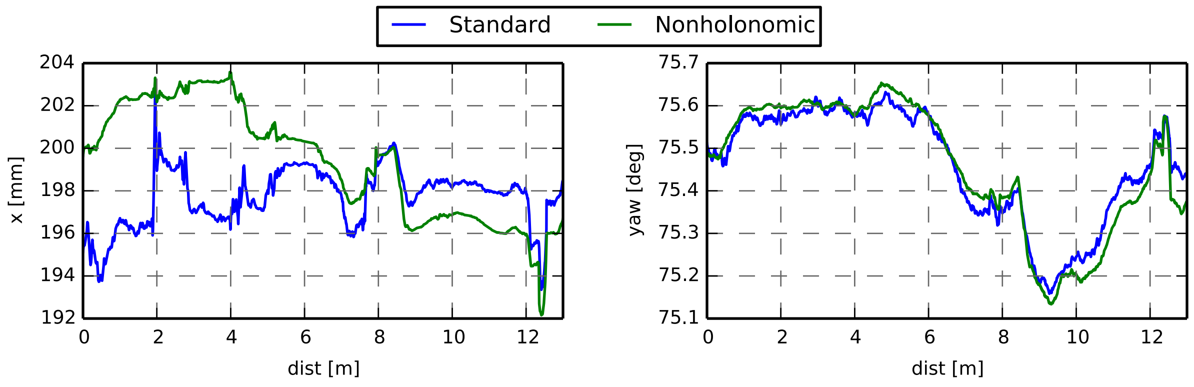

Figure 22: Comparison of the calibration using nonholonomic constraints and the standard procedure for the second tested configuration in Table 5.

Having shown that x_{vc} and γ_{vc} can be reliably estimated using nonholonomic calibration, we are ready to demonstrate that the remaining degree of freedom, the y_{vc} coordinate of the camera pose in the robot frame of reference, can be obtained using our motion priors formulation, i.e. when for every VO measurement

[1]The values obtained for standard calibration differ slightly from the values presented in Section 6.3 (Tables 1, 2, 3) as here it was performing only a continuous graph calibration, whereas previously we run a completed 6 DoF calibration for each trajectory segment.

we only know whether the robot was moving forward or backward. Obviously, this formulation puts more requirements on a trajectory that is suitable for calibration. In fact we found that most of the trajectories the robot executed during our test were not suitable for this kind of calibration, as the robot was mainly moving forward. A suitable trajectory features a rich combination of forward and backward motions with a range of angular and linear velocities.

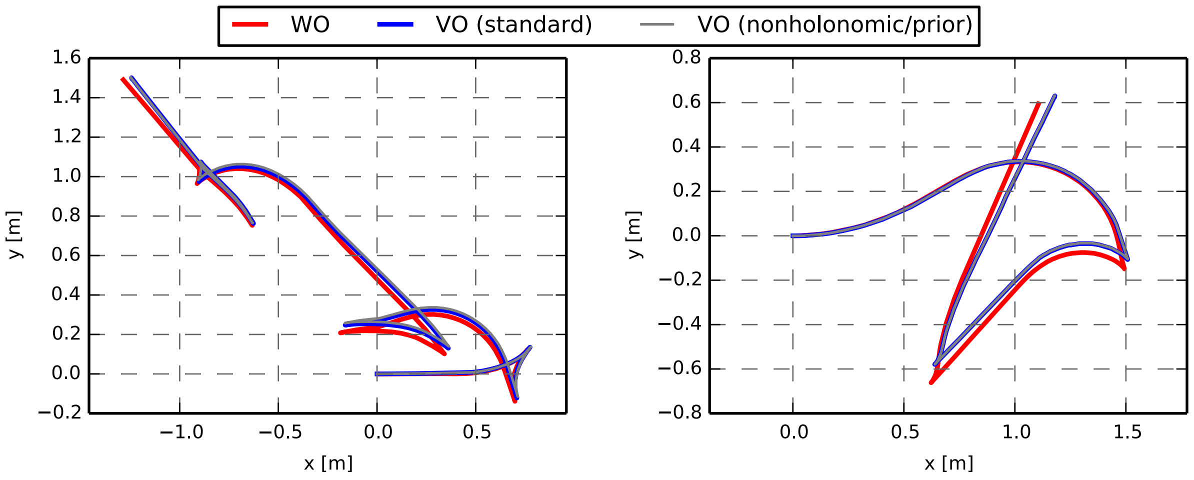

Figure 23: Robot trajectories obtained from WO ground truth compared with VO calibrated using our standard method and nonholonomic/direction of motion calibration. We see very good results independent of the calibration method.

To perform this evaluation we executed the following steps. Assuming that the roll and pitch angles are already estimated, and the scale is also known, we first performed nonholonomic calibration to determine x_{vc} and the yaw angle. Next, we used the wheel odometry to calculate the directions of robot motion between consecutive camera frames and constructed a graph as explained in Section 5.2.2. Figure 23 compares trajectories obtained using standard graph based auto-calibration as well as the proposed nonholonomic/motion direction calibration; a comparison of values obtained is also shown in Table 6. We can see that for the two independent calibration strategies, both the calibration values and the trajectories match very closely.

Table 6: Calibration results obtained using nonholonomic constraints and direction of motion compared with the values calculated for the same trajectories using standard calibration procedure that relies on synchronised wheel odometry. Trajectories are depicted in Figure 23. We see a very good agreement between these two methods.

	x [mm]	y [mm]	yaw [°]
config. 1: standard	241.6	-9.1	27.1
config. 1: nonholonomic + prior	241.8	-3.7	27.0
config. 2: standard	243.3	-24.6	-9.2
config. 2: nonholonomic + prior	243.2	-25.1	-9.1

7 Conclusions

We have demonstrated that a dense planar alignment approach offers a robust and accurate solution for visual odometry, and that it can be precisely auto-calibrated in highly practical settings. In particular, the approach based on nonholonomic constraints permits usable and infrastructure-free auto-calibration, which opens the door to a full re-calibration in the field. Continuous and lightweight auto-calibration is a realistic prospect in many applications with the methods we have presented. We have seen for instance that in the road car dataset there is significant roll induced when turning, breaking our assumptions but with the possibility to be corrected by continuous auto-calibration which may also provide very useful extra information on for instance whether the suspension settings of a car are correctly adjusted.

The methods are well suited to low-cost mobile robots of the near future because they benefit from the ease of efficient parallel implementation. Dense alignment methods are highly scalable, and it will be interesting to see the trade-offs in VO as camera frame-rate and resolution are varied [Handa et al., 2012].

Although our main assumption that the ground is flat and parallel to the camera motion limits the applicability of our method to certain applications, we have extensively shown how this greatly simplifies VO and robustifies it by allowing all image information in a video sequence to be used. In auto-calibration, our assumption allows the camera roll and pitch to be calculated by estimating the orientation of the ground. This permits the separation of calibration into two steps, vision-based and the graph-based, making the graph-based calibration (also the nonholonomic) more general and not restricted to cameras, but other different types of sensors that can measure incremental, planar motions such as laser scanners. However, we should note that miscalibration of the roll and pitch leads to an incorrect estimate of planar motion, which cannot be compensated for in the subsequent stages of auto-calibration.

In future work, we are interested to investigate the role that continuous-time trajectory representations (e.g. [Lovegrove et al., 2013, Furgale et al., 2012]) might have in nonholonomic calibration, with the ease that they offer in formulating and evaluation constraints on a robots motion and velocity, or to take advantage of information from other unsynchronised sensors.

Acknowledgements

We are grateful to Dyson Ltd. for funding this research and collaboration on techniques and ideas. We also thank our colleagues at Imperial College London for daily discussions and assistance, in particular Ankur

Handa and Robert Lukierski.

Appendix

Nonholonomic calibration

When calculating the nonholonomic error as defined in Eq. 30 one has to use the values measured in the camera frame of reference $(\dot{x}_c, \dot{y}_c, \dot{\theta}_c)$ and θ_c, together with the current estimates of the calibration parameters $(x_{vc}, y_{vc}, \gamma_{vc})$ to obtain the velocities in the robot frame of reference $\dot{x}_r$ and $\dot{y}_r$ as well as the robot orientation θ_r. One can show that these values depends on each other in the following way:

$$\begin{aligned}
\dot{x}_r &= \dot{x}_c \cos\gamma_{vc} - \dot{y}_c \sin\gamma_{vc} + \dot{\theta}_c \left(y_{vc}\cos\theta_c + x_{vc}\sin\theta_c\right) \\
\dot{y}_r &= \dot{x}_c \sin\gamma_{vc} + \dot{y}_c \cos\gamma_{vc} - \dot{\theta}_c \left(x_{vc}\cos\theta_c - y_{vc}\sin\theta_c\right) \\
\theta_r &= \theta_c
\end{aligned} \tag{32}$$

The nonholonomic error assumes rolling without slipping and has the form:

$$e^h = \dot{x}_r \sin\theta_r - \dot{y}_r \cos\theta_r \tag{33}$$

Substituting Eq. 32 into Eq. 33 leads to:

$$\begin{aligned}
e^h &= \dot{x}_c \sin\theta_c \cos\gamma_{vc} - \dot{y}_c \sin\theta_c \sin\gamma_{vc} + \dot{\theta}_c \sin\theta_c \left(y_{vc}\cos\theta_c + x_{vc}\sin\theta_c\right) \\
&\quad - \dot{x}_c \cos\theta_c \sin\gamma_{vc} - \dot{y}_c \cos\theta_c \cos\gamma_{vc} + \dot{\theta}_c \cos\theta_c \left(x_{vc}\cos\theta_c - y_{vc}\sin\theta_c\right)
\end{aligned} \tag{34}$$

Simplifying the equation and using the trigonometric identity $(\sin^2\theta_c + \cos^2\theta_c = 1)$, we can reduce the error to the following formula:

$$\begin{aligned}
e^h &= \dot{x}_c \sin\theta_c \cos\gamma_{vc} - \dot{y}_c \sin\theta_c \sin\gamma_{vc} - \dot{x}_c \cos\theta_c \sin\gamma_{vc} - \dot{y}_c \cos\theta_c \cos\gamma_{vc} + \dot{\theta}_c x_{vc} \\
&= \dot{y}_c \cos(\gamma_{vc} - \theta_c) + \dot{x}_c \sin(\gamma_{vc} - \theta_c) + \dot{\theta}_c x_{vc}
\end{aligned} \tag{35}$$

The error e^h depends on the velocities measured in the camera frame of reference $(\dot{x}_c, \dot{y}_c, \dot{\theta}_c)$, the corresponding camera orientation θ_c, as well as the x_{vc} and γ_{vc}, however it does not depend on the y_{vc}-coordinate of the camera pose on the robot, and therefore we cannot estimate this quantity using nonholonomic calibration method.

References

[Adams et al., 2002] Adams, H., Singh, S., and Strelow, D. (2002). An empirical comparison of methods for image-based motion estimation. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*.

[Antonelli et al., 2010] Antonelli, G., Caccavale, F., Grossi, F., and Marino, A. (2010). Simultaneous calibration of odometry and camera for a differential drive mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Blanco et al., 2014] Blanco, J.-L., Moreno, F.-A., and González-Jiménez, J. (2014). The Málaga Urban Dataset: High-rate Stereo and Lidars in a Realistic Urban Scenario. *International Journal of Robotics Research (IJRR)*, 33(2).

[Brookshire and Teller, 2011] Brookshire, J. and Teller, S. (2011). Automatic Calibration of Multiple Coplanar Sensors. In *Proceedings of Robotics: Science and Systems (RSS)*.

[Campbell et al., 2005] Campbell, J., Sukthankar, R., Nourbakhsh, I., and Pahwa, A. (2005). A Robust Visual Odometry and Precipice Detection System Using Consumergrade Monocular Vision. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Carrera et al., 2011] Carrera, G., Angeli, A., and Davison, A. J. (2011). Lightweight SLAM and Navigation with a Multi-Camera Rig. In *Proceedings of the European Conference on Mobile Robotics (ECMR)*.

[Censi et al., 2008] Censi, A., Marchionni, L., and Oriolo, G. (2008). Simultaneous Maximum-Likelihood Calibration of Odometry and Sensor Parameters. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Davison, 2003] Davison, A. J. (2003). Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Faugeras and Lustman, 1988] Faugeras, O. D. and Lustman, F. (1988). Motion and Structure From Motion in a Piecewise Planar Environment. *International Journal of Pattern Recognition in Artificial Intelligence*, 2(3):485–508.

[Fossati and Fua, 2008] Fossati, A. and Fua, P. (2008). Linking pose and motion. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[Furgale et al., 2012] Furgale, P., Barfoot, T. D., and Sibley, G. (2012). Continuous-time batch estimation using temporal basis functions. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Geiger et al., 2012] Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Handa et al., 2012] Handa, A., Newcombe, R. A., Angeli, A., and Davison, A. J. (2012). Real-Time Camera Tracking: When is High Frame-Rate Best? In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[Kelly and Sukhatme, 2010] Kelly, J. and Sukhatme, G. S. (2010). Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *International Journal of Robotics Research (IJRR)*, 30(1):56–79.

[Kitt et al., 2011] Kitt, B., Rehder, J., Chambers, A., Schonbein, M., Lategahn, H., and Singh, S. (2011). Monocular Visual Odometry using a Planar Road Model to Solve Scale Ambiguity. In *Proceedings of the European Conference on Mobile Robotics (ECMR)*.

[Knorr et al., 2013] Knorr, M., Niehsen, W., Member, S., and Stiller, C. (2013). Online extrinsic multi-camera calibration using ground plane induced homographies. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*.

[Konolige et al., 2007] Konolige, K., Agrawal, M., and Solà, J. (2007). Large Scale Visual Odometry for Rough Terrain. In *Proceedings of the International Symposium on Robotics Research (ISRR)*.

[Kümmerle et al., 2011a] Kümmerle, R., Grisetti, G., and Burgard, W. (2011a). Simultaneous Calibration, Localization, and Mapping. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*.

[Kümmerle et al., 2011b] Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011b). g^2o: A General Framework for Graph Optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Lovegrove et al., 2013] Lovegrove, S., Patron-Perez, A., and Sibley, G. (2013). Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras. In *Proceedings of the British Machine Vision Conference (BMVC)*.

[Lovegrove et al., 2011] Lovegrove, S. J., Davison, A. J., and Ibanez-Guzmán, J. (2011). Accurate Visual Odometry from a Rear Parking Camera. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*.

[Lucas and Kanade, 1981] Lucas, B. D. and Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

[Malis, 2004] Malis, E. (2004). Improving vision-based control using efficient second-order minimization techniques. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Martinelli et al., 2006] Martinelli, A., Scaramuzza, D., and Siegwart, R. (2006). Automatic self-calibration of a vision system during robot motion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Mei et al., 2008] Mei, C., Benhimane, S., Malis, E., and Rives, P. (2008). Efficient Homography-Based Tracking and 3-D Reconstruction for Single-Viewpoint Sensors. *IEEE Transactions on Robotics (T-RO)*, 24(6):1352–1364.

[Mei et al., 2009] Mei, C., Sibley, G., Cummins, M., Newman, P., and Reid, I. (2009). A Constant Time Efficient Stereo SLAM System. In *Proceedings of the British Machine Vision Conference (BMVC)*.

[Miksch et al., 2010] Miksch, M., Yang, B., and Zimmerman, K. (2010). Automatic Extrinsic Camera Self-Calibration Based on Homography and Epipolar Geometry. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*.

[Mirzaei and Roumeliotis, 2008] Mirzaei, F. and Roumeliotis, S. (2008). A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation. *IEEE Transactions on Robotics (T-RO)*, 24(5):1143–1156.

[Nistér et al., 2004] Nistér, D., Naroditsky, O., and Bergen, J. (2004). Visual Odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Nourani-Vatani et al., 2009] Nourani-Vatani, N., Roberts, J., and Srinivasan, M. V. (2009). Practical visual odometry for car-like vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Pirchheim and Reitmayr, 2011] Pirchheim, C. and Reitmayr, G. (2011). Homography-based planar mapping and tracking for mobile phones. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*.

[Saurer et al., 2012] Saurer, O., Fraundorfer, F., and Pollefeys, M. (2012). Homography based visual odometry with known vertical direction and weak Manhattan world assumption. In *IEEE/IROS Workshop on Visual Control of Mobile Robots*.

[Scaramuzza et al., 2009a] Scaramuzza, D., Fraundorfer, F., Pollefeys, M., and Siegwart, R. (2009a). Absolute Scale in Structure from Motion from a Single Vehicle Mounted Camera by Exploiting Nonholonomic Constraints. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Scaramuzza et al., 2009b] Scaramuzza, D., Fraundorfer, F., and Siegwart, R. (2009b). Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Silveira et al., 2008] Silveira, G., Malis, E., and Rives, P. (2008). An Efficient Direct Approach to Visual SLAM. *IEEE Transactions on Robotics (T-RO)*, 24(5):969–979.

[Underwood et al., 2010] Underwood, J. P., Hill, A., Peynot, T., and Scheding, S. J. (2010). Error modeling and calibration of exteroceptive sensors for accurate mapping applications. *Journal of Field Robotics (JFR)*, 27(1):2–20.

[Zhang, 1999] Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Zienkiewicz et al., 2013] Zienkiewicz, J., Lukierski, R., and Davison, A. J. (2013). Dense, Autocalibrating Visual Odometry from a Downward-Looking Camera. In *Proceedings of the British Machine Vision Conference (BMVC)*.