Bias Compensation in Visual Odometry

Gijs Dubbelman, Peter Hansen, Brett Browning

Abstract— **Empirical evidence shows that error growth in visual odometry is *biased*. A projective *bias model* is developed and its parameters are estimated *offline* from trajectories encompassing loops. The model is used *online* to compensate for bias and thereby significantly reduces error growth. We validate our approach with more than 25 km of stereo data collected in two very different urban environments from a moving vehicle. Our results demonstrate significant reduction in error, typically on the order of 50%, suggesting that our technique has significant applicability to deployed robot systems in GPS denied environments.**

I. Introduction

Estimating the pose of a moving camera solely from its recorded images, i.e. *visual odometry* (VO), is important to many robotic tasks [1]–[8]. Here we focus on applications where the full 6 *degrees of freedom* (dof.) robot pose needs to be tracked accurately over several kilometers. Empirical results show that the error distribution in the tracked absolute pose has small variance with a significant *non-zero* mean, i.e. the errors are *biased* (note that this differs from the statistical definition of estimator bias).

It is due to recent advances in robotics, computer vision and related fields of modern science that bias is left as the main source of error growth. In earlier work the effects of bias were unobservable as VO drift was dominated by sub-optimal optimization [9]–[11], i.e. minimizing residuals in the 3D domain instead of the image domain, and due to *outlier* image point correspondences. Significant increases in computational power currently allow for using methods which minimize *reprojection errors* [12] and which are unaffected by outliers in realistic circumstances [3], [13]–[18]. These methods exhibit significant less error growth but their errors are still biased for trajectories that are several kilometers long.

Bundle adjustment (BA) [19]–[23] is used as the final optimization step of many *state-of-the-art* visual odometers. BA minimizes a non-linear model which captures the physical and statistical relationship between scene points and their projections onto the camera plane. Such models often account for lens aberrations like radial distortion and decentering distortion. It is straightforward to verify (e.g. by using Monte-Carlo simulations) that when such models truly form the image data, then *state-of-the-art* VO methods exhibit zero-mean error distributions. We show that results obtained from real image data are biased. This indicates that the true physical and statistical phenomena underlying real camera observations are not fully captured by current models.

Two main lines of research can be followed to reduce bias. First, one can research and develop more accurate physical and statistical models for camera observations. Second, one can use current models, accept their imperfections and compensate for their bias. Our contribution follows this second line of research. The motivations for this choice are given in Sec. II which also provides the used camera model and used calibration technique. Our projective model for bias is described in Sec. III along with the method to estimate its parameters. It involves projectivities, defined by motion dependent polynomials, being applied to the relative pose estimates. In Sec. IV it is shown on over 25 km of binocular data that our model can be used online to compensate for bias and significantly reduce VO drift. Our conclusions are provided in Sec. V.

II. Camera models and calibration

Improving camera models has a long history in modern science [24]–[29]. Here we use the current *de facto* standard for *projective fixed-focal-length* stereo cameras which accounts for radial distortion and decentering distortion.

The stereo camera consists of a *left camera*, chosen to be the origin, and a *right camera*. The known 3D world point $\mathbf{w}$ is transformed into the left camera's coordinate frame with $\mathrm{A}^{-1}\mathbf{w} = (x, y, z, 1)^\top$ where $\mathrm{A} \in \mathrm{SE}(3)$ is the stereo camera's absolute pose. For the right camera the transformation from world coordinates to camera coordinates includes the baseline offset $\mathrm{B} \in \mathrm{SE}(3)$, i.e. $(\mathrm{AB})^{-1}\mathbf{w}$. To allow for efficient notation we define $\mathrm{M} = [\mathrm{R}, \mathbf{t}]$ to be the 4×4 homogenous matrix $\mathrm{M} \in \mathrm{SE}(3)$ expressing the rotation R and translation $\mathbf{t}$.

The 3D point is projected onto the normalized camera plane with $(x/z, y/z)^\top = (u_x, u_y)^\top = \mathbf{u}_\mathrm{n}$. Radial distortion and decentering distortion in the left camera plane are modeled with

$$\mathbf{u}_\mathrm{d} = \mathbf{u}_\mathrm{n}(1 + r_2^l\|\mathbf{u}_\mathrm{n}\|^2 + r_4^l\|\mathbf{u}_\mathrm{n}\|^4 + r_6^l\|\mathbf{u}_\mathrm{n}\|^6) + \mathbf{d} \quad (1)$$

where

$$\mathbf{d} = \begin{bmatrix} 2t_1^l u_x u_y + t_2^l(\|\mathbf{u}_\mathrm{n}\|^2 + 2u_x^2) \\ t_1^l(\|\mathbf{u}_\mathrm{n}\|^2 + 2u_y^2) + 2t_2^l u_x u_y \end{bmatrix} \quad (2)$$

is the decentering distortion component. The image point $\mathbf{u}$ in the left camera plane is obtained with

$$\mathbf{u} = \mathrm{K}\begin{bmatrix} \mathbf{u}_\mathrm{d} \\ 1 \end{bmatrix} \quad \text{where} \quad \mathrm{K} = \begin{bmatrix} f^l & 0 & c_x^l \\ 0 & f^l & c_y^l \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

This report was made possible by the support of an NPRP grant from the Qatar National Research Fund. The statements made herein are solely the responsibility of the authors. Dubbelman and Browning are with Carnegie Mellon's Robotics Institute (Pittsburgh, USA), while Hansen is with Carnegie Mellon's Computer Science department at the Qatar campus (Doha, Qatar): gijsdubb@cs.cmu.edu, phansen@qatar.cmu.edu, brettb@cs.cmu.edu

is the left camera's projection matrix. The vector of all intrinsic parameters defining the left camera is $\theta_l = (f^l, c_x^l, c_y^l, r_2^l, r_4^l, r_6^l, t_1^l, t_2^l)^\top$ The same model is employed for the right camera using the parameters $\theta_r = (f^r, c_x^r, c_y^r, r_2^r, r_4^r, r_6^r, t_1^r, t_2^r)^\top$.

We use a Levenberg-Marquardt optimizer with converging outlier suppression to estimate the intrinsic camera parameters θ_l and θ_r, the stereo baseline B, the extrinsic camera parameters $\mathrm{A}_1, .., \mathrm{A}_n$ for each calibration image and the 3D position of the calibration points $\mathbf{w}_1, ..., \mathbf{w}_m$. These 3D points are the corners of a checker-board calibration target and extraction of the projections of these corners from the calibration images is fully automated. This allows us to calibrate on several thousand calibration images which in turn allows for estimating the 3D positions of the calibration points [30]. This accounts for imperfections in the manufacturing (up to sub-millimeter accuracy) of the calibration target.

To the best of our knowledge it is currently not known which possible deviation from the used camera model, e.g. depth-dependent distortion [31], non-central projection [32], imperfections in the camera plane and lenses [33], instability of optical and baseline components, is most significant for a VO system using a projective fixed-focal-length stereo camera. Our calibration results suggest that such possible deviations are in the range of less than $1/10$ of a pixel (i.e. less than $1\mu m$). For such small deviations it is challenging to distinguish a genuinely better model from an overparametrized model which is being overfitted to the image data [24]. Calibrating more sophisticated camera models will therefore likely require more advanced (and significantly more costly) optical measurement devices than the calibration targets used in robotics and in computer vision. While such devices do exist and are used in optical design and verification, it is not realistic to assume that such devices will become widely available to robotic and computer vision researchers in the near future. Although improving camera models is relevant, it is not the line of research we choose here.

In this research we accept that the used camera model is an approximation and pursue a general model to describe the bias this approximation causes in the tracked camera pose. The goal is to estimate the parameters defining this model from calibration trajectories and to apply them to trajectories not used during calibration. We stress that this is conceptually different from auto-calibration or online calibration-refinement [12], [34], [35].

The camera models used for auto-calibration are typically less complex than the model used here and are often linear. The utility of auto-calibration is also not to provide highly accurate camera parameters. It is used when one has no or very inaccurate knowledge of the camera parameters and allows for lifting a projective reconstruction to an Euclidean reconstruction (up to scale). Only *full*-BA (i.e. BA wrt. motion, structure and camera intrinsics) can potentially improve upon offline camera calibration. However, it is well known that full BA requires a strong network structure (e.g. many overlapping fields of view) and a good distribution of camera positions, camera orientations and landmarks. These requirements are typically not sufficiently met for VO applications. The utility of full BA is also mainly as a final refinement step for *multiple-camera varying-focal-length* photogrammetric reconstruction [36]. While using BA can improve the accuracy of *single-camera fixed-focal-length* VO, especially when loop-closing is possible, it has never been proven or shown that also optimizing with respect to camera intrinsics will lead to truly more accurate camera parameters than those obtained from extensive offline calibration (at least not for a properly manufactured stereo camera which retains its calibration under realistic circumstances).

III. A PROJECTIVE MODEL FOR MOTION BIAS

The goal is to specify a function that takes each biased relative pose estimate $\mathrm{M}' \in \mathrm{SE}(3)$ to an unbiased relative pose estimate $\mathrm{M} \in \mathrm{SE}(3)$. It can potentially be any high-order function which takes points on $\mathrm{SE}(3)$ to points on $\mathrm{SE}(3)$. However, to come to an applicable and stable approach, it is best to use more constrained functional models. In this contribution we explore projective models (but other models might potentially be just as useable).

In theory, a monocular camera with projection matrix K moving from its initial pose I to $\mathrm{M} \in \mathrm{SE}(3)$ while observing the landmarks $\mathbf{w}_{1,..,n}$ will result in the observations $\mathbf{u}_{1,..,n}$ before the motion and the observations $\mathbf{v}_{1,..,n}$ after the motion. If K is known, then M can be estimated unbiased up to scale. Now assume that only an erroneous estimate for K is available and that this erroneous estimate K' differs from the true K by a projectivity H, i.e. $\mathrm{K}' = \mathrm{KH}$. If we would not restrict the estimate for M to be an element of $\mathrm{SE}(3)$, then the optimizer would converge to $\mathrm{M}' = \mathrm{H}^{-1}\mathrm{MH}$. This is because the projections of the landmark estimates $\mathrm{H}^{-1}\mathbf{w}_{1,..,n}$ given M' and K' are exactly equal to $\mathbf{u}_{1,..,n}$ and $\mathbf{v}_{1,..,n}$ [12]. In this projective setting, the function that takes an erroneous projective estimate M' back to its correct Euclidean estimate M would involve nothing more than $\mathrm{M} = \mathrm{HM}'\mathrm{H}^{-1}$. Typically, the optimizers used in VO constrain the estimate M' to be in $\mathrm{SE}(3)$. This can be seen as a map $\mathrm{SE} : \mathbb{P}^3 \to \mathrm{SE}(3)$ that takes the projectivity $\mathrm{H}^{-1}\mathrm{MH}$ back to an $\mathrm{M}' \in \mathrm{SE}(3)$. When assuming that the difference between the used camera model and the true unknown camera model can be captured (up to first order) by a projectivity $\mathrm{H} \approx \mathrm{I}$ (i.e. a projectivity close to the identity matrix), then the process that takes the true M to a biased estimate M' can in theory be modeled with $\mathrm{M}' = \mathrm{SE}(\mathrm{H}^{-1}\mathrm{MH})$ up to some random perturbation. We are interested in the inverse of this process.

Inspired by projective geometry, our model first maps $\mathrm{M}' = [\mathrm{R}, \mathbf{t}]$ to a projectivity $\mathrm{M}'_{\mathbb{P}}$ and then takes this projectivity back to $\mathrm{SE}(3)$ to produce the desired unbiased estimate M, i.e.

$$\mathrm{SE}(3) \to \mathbb{P}^3 \to \mathrm{SE}(3). \tag{4}$$

The first part, i.e. $\mathrm{SE}(3) \rightarrow \mathbb{P}^3$, is modeled as being rotation dependent with

$$\mathrm{M}'_{\mathbb{P}} = \mathcal{H}(\mathrm{M}', \Theta)\ \mathrm{M}'\ \mathcal{H}(\mathrm{M}', \Theta)^{-1} \tag{5}$$

where the projectivity $\mathcal{H}(\mathrm{M}, \Theta)$ is restricted to have the form

$$\mathcal{H}(\mathrm{M}, \Theta) = \begin{bmatrix} s_x & 0 & a_x & 0 \\ 0 & s_y & a_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

(i.e. non-uniform scaling and shear mapping) and where its values are specified by first order polynomials

$$\begin{array}{rcl} s_x & = & c_{s_x} + c_{s_x}^{r_x} r_x + c_{s_x}^{r_y} r_y + c_{s_x}^{r_z} r_z \\ s_y & = & c_{s_y} + c_{s_y}^{r_x} r_x + c_{s_y}^{r_y} r_y + c_{s_y}^{r_z} r_z \\ a_x & = & c_{a_x} + c_{a_x}^{r_x} r_x + c_{a_x}^{r_y} r_y + c_{a_x}^{r_z} r_z \\ a_y & = & c_{a_y} + c_{a_y}^{r_x} r_x + c_{a_y}^{r_y} r_y + c_{a_y}^{r_z} r_z \end{array} \tag{7}$$

with $(r_x, r_y, r_z)^\top = \mathrm{Log}(\mathrm{R})$, see Appendix A. The vector $\Theta = (c_{s_x}, c_{s_x}^{r_x}, c_{s_x}^{r_y}, c_{s_x}^{r_z}, ..., c_{c_y}, c_{c_y}^{r_x}, c_{c_y}^{r_y}, c_{c_y}^{r_z})$ is the vector of all polynomial coefficients (note that s_x and s_y cannot be equal to zero).

The matrix $\mathrm{M}'_{\mathbb{P}} = [\mathrm{A}, \mathbf{b}]$ will in general be a projectivity and can be taken back to $\mathrm{SE}(3)$ with the map

$$\mathrm{M} = \mathrm{SE}(\mathrm{M}_{\mathbb{P}}) = [\mathrm{U}\mathrm{V}^\top, \mathbf{b}] \tag{8}$$

where $\mathrm{U}\Sigma\mathrm{V}^\top = \mathrm{A}$, i.e. the singular value decomposition (SVD) of A.

Our functional model that allows for mapping biased relative pose estimates $\mathrm{M}' \in \mathrm{SE}(3)$ back to unbiased relative pose estimates $\mathrm{M} \in \mathrm{SE}(3)$ is then provided by

$$\mathrm{M} = \mathrm{SE}(\mathcal{H}(\mathrm{M}', \Theta)\ \mathrm{M}'\ \mathcal{H}(\mathrm{M}', \Theta)^{-1}). \tag{9}$$

The challenge is to find appropriate values for the coefficients in Θ, i.e. finding a specific applicable function from within our model.

We point out that our model in Eq. 9 is just one out of many potentially applicable models. While our model allows for efficient and significant bias compensation, we do not dismiss the possibility that better models can be developed.

A. Estimating the projective coefficients

The main objective is to estimate appropriate values for the projective coefficients Θ in Eq. 9 such that bias is minimized. Bias is too small to be measured accurately and reliably from single relative pose estimates. When sequentially integrating many relative pose estimates, bias will accumulate and cause a significant and measurable drift in the estimated absolute pose. For trajectories comprising a loop this drift can be quantized through the *error at loop-closure*, see Fig.1 for an illustration.

In the error-free case the sequential pose estimate, i.e. $\mathrm{A}_n = \mathrm{A}_1\mathrm{M}_{1:2}\mathrm{M}_{2:3}...\mathrm{M}_{n-1:n}$, together with the *loop-closing pose*, i.e. $\mathrm{M}_{n:1}$ which goes from the last frame directly back to the first frame, brings the camera back in its original starting position A_1. In a realistic setting $\mathrm{A}_n\mathrm{M}_{n:1}$ will not be equal to A_1 due to accumulated errors.

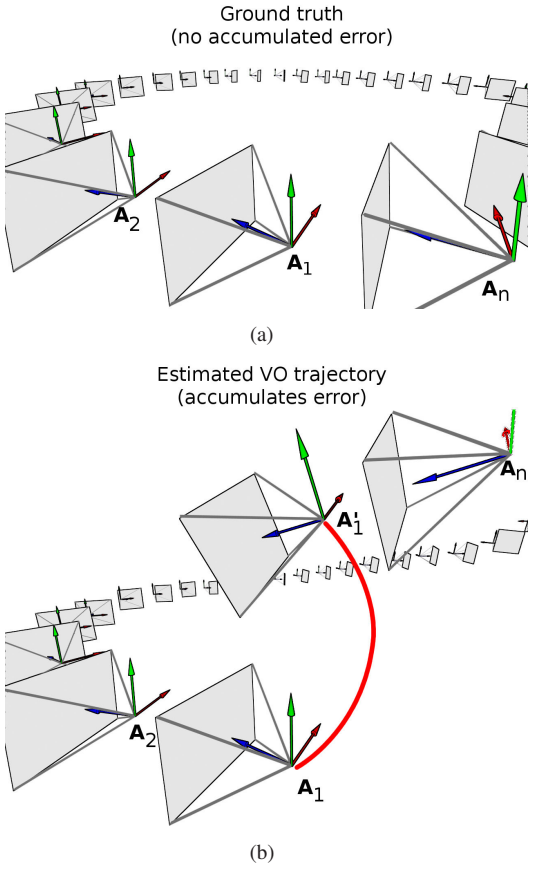


Fig. 1. Illustration of the *error at loop-closure* performance metric. When sequentially estimating the camera pose from the first frame located at A_1 all the way to the last frame located at A_n and from this last frame back to the first frame, i.e. estimating the loop-closing pose, then in the ideal error-free case we arrive exactly back at A_1. This is illustrated in (a). Estimating the loop-closing pose is important as the true pose of the last frame A_n itself will never be exactly the same as the true pose of the first frame A_1. Our performance metric takes this into account. In the error-prone case we do not arrive exactly back at A_1 but end up at A'_1. This is illustrated in (b). Note that A_1 and A'_1 both describe the pose of exactly the same frame, i.e. the first frame. In the error-free case A_1 and A'_1 would be exactly the same. In reality due to accumulated errors they are not. The 6 dof. pose displacement between A_1 and A'_1, depicted by the red line in (b), is therefore representative for the accumulated error in the trajectory. Conceptually, the shorter this red line is (in $\mathbb{R}^3$ and in $\mathrm{SO}(3)$), the more accurate the trajectory is.

The 6 dof. pose difference between A_1 and $\mathrm{A}_n\mathrm{M}_{n:1}$ is $\mathrm{E} = \mathrm{A}_1^{-1}\mathrm{A}_n\mathrm{M}_{n:1}$ and is representative for the accumulated error of sequentially estimated trajectories.

As in [37] a geometrically meaningful metric is expressed on the *error at loop-closure* $\mathrm{E} = [\mathrm{R}, \mathbf{t}]$ with

$$\|\mathrm{E}\|_\alpha = \sqrt{\|\mathrm{Log}(\mathrm{R})\|^2 + \alpha\|\mathbf{t}\|^2} \tag{10}$$

where Log is the logarithmic map of $\mathrm{SO}(3)$ as provided in Appendix A. The purpose of the scalar α is to provide a trade-off between positional and orientational errors, we come back to this in Sec. III-B. The metric in Eq. 10 is the basis for the cost function of our optimizer.

Let us denote the ordered set of all relative pose estimates, including the *loop-closing pose*, of a trajectory with $\mathbf{M} = (\mathrm{M}_{1:2}, \mathrm{M}_{2:3}, ..., \mathrm{M}_{n-1:n}, \mathrm{M}_{n:1})$ and let $\mathbf{M}_i^j$ denote the ith

relative pose of the jth trajectory. The error at loop-closure E for the jth trajectory given the projective coefficients Θ can then be computed with

$$\mathcal{E}(\mathbf{M}^j, \Theta) = \overrightarrow{\prod_{\mathrm{M}_i^j \in \mathbf{M}^j}} \mathrm{SE}(\mathcal{H}(\mathrm{M}_i^j, \Theta)\mathrm{M}_i^j \mathcal{H}(\mathrm{M}_i^j, \Theta)^{-1}) \quad (11)$$

where $\overrightarrow{\prod}$ denotes a matrix product that expands to the right.

The function $\mathcal{E}(\mathbf{M}^j, \Theta)$ together with the metric in Eq. 10 allow for the following non-linear minimization task

$$\underset{\Theta}{\operatorname{argmin}} \sum_{j=1}^{k} \beta^j \|\mathcal{E}(\mathbf{M}^j, \Theta)\|_{\alpha^j}^2 \quad (12)$$

or in words: we seek those projective coefficients Θ which reduce bias, as measured by our metric on errors at loop-closure, over k calibration trajectories. The per-trajectory scalars α^j and β^j are used to precondition the errors as discussed in Sec. III-B.

This non-linear objective function is minimized wrt. Θ using Levenberg-Marqaurdt. This can be done relatively efficiently as computing its Jacobian with a finite difference scheme only involves 4×4 matrix multiplications and SVD's of 3×3 matrices. Furthermore, this objective function is only minimized wrt. Θ and not wrt. to all poses, all landmarks and all intrinsics as is done in full BA. It is therefore significantly more efficient than full BA, e.g. in Matlab it only takes around 20 seconds to calibrate Θ on 10 trajectories of 1000 poses each.

B. Preventing over fitting

Preventing over fitting is the key factor towards successfully applying our approach in practice. In this setting over fitting refers to obtaining projective coefficients which reduce bias for the calibration trajectories but do not reduce bias (or even increase bias) for other trajectories. We discuss how some general well known strategies can be used to prevent over fitting in our setting.

As always one should calibrate using sufficient data, i.e. sufficient trajectories which are representative for the robot's environment. The optimizer must also be encouraged to find projective coefficients which reduce bias for all calibration trajectories (and not only for certain calibration trajectories at the expense of other calibration trajectories). A well known technique to do so is pre-conditioning the errors before optimization. In our setting this refers to setting the scalar α^j in Eq. 12 for each trajectory such that its orientational errors are weighted equally as its positional errors. We also re-weight the total error of each trajectory by setting appropriate (normalizing) values for each β^j in Eq. 12 such that each trajectory contributes equally to the summed squared error. Finally, and most importantly, one should not use too many and irrelevant coefficients for the polynomials. Again we can use a well known technique which starts with the constant coefficients of Θ, estimates their values and sees if they are significant wrt. their uncertainties. It then ignores that coefficient which is most insignificant and repeats this process until all remaining constant coefficients are either significant or ignored. The process then repeats itself for the linear coefficients and so on.

This process, while effective, can probably be improved upon by e.g. using *train-* and *cross-validation* trajectories while estimating the projective coefficients.

IV. Evaluation

The goal of this section is firstly to validate our working hypothesis, i.e. VO errors are biased, and secondly to show that our bias compensation method reduces VO drift. We will do so on the basis of more than 25 km of binocular data.

Our first data set, Doha-A, comprises of 12 loops (6 clockwise and 6 counter clockwise) each around 1 km long. A single loop from this data set is depicted in Fig. 4.a. The second data set, Doha-B, comprises a single loop of 2.5 km long and is depicted in Fig. 5. The third data set, Pittsburgh-A, consists of 6 loops (3 clockwise-to-counter-clockwise and 3 counter-clockwise-to-clockwise) each 1 km long. Two such loops are shown in Fig. 4.b,c. The fourth data set, Pittsburgh-B, consist of two loops (1 clockwise and 1 counter clockwise) each 3 km long, these are depicted in Fig. 6. See Fig. 2 for example images from our data sets.

The Doha data sets are recorded with a Point Grey Bumblebee2 (640x480 colored pixels, 3.8 mm focal length, 65° horizontal field of view, 10 fps) stereo camera. The Pittsburgh data sets are recorded with another Point Grey Bumblebee2 having the same specifications (but slightly different optical properties) running at 20 fps. Our visual-odometry technique is similar to that of [3], [11] except that we use a ML optimizer (which minimizes reprojection residuals and therefore is theoretically unbiased) instead of the HEIV algorithm (which minimizes errors in 3D and therefore is inherently biased [11]). More information on this VO approach can be found in [38]. Furthermore, sliding-window BA is performed in batches of 10 automatically selected key-frames. We enforce that the last key-frame is always exactly the same stereo pair as the first key-frame so that we can use our *error at loop-closure* performance metric. The stereo camera used for the Doha data sets is calibrated using approximately 2000 images and the stereo camera used for the Pittsburgh data sets is calibrated using approximately 3000 images. The used camera model and calibration methods were described in Sec. II.

When testing our bias compensation technique we exclusively divide data sets into *train-loops* and *test-loops* (i.e. a trajectory cannot be used as a train-loop and as a test-loop). We estimate the coefficients Θ on train-loops and then verify if the obtained coefficients also reduce bias on test-loops. Only train-loops explicitly need to encompass a loop. The reason that all test-loops also encompass a loop is only because this allows us to use the *error at loop-closure* to accurately measure error reductions of our bias compensation method. In the error-free case the last estimated absolute pose $\mathrm{A}_1' = \mathrm{M}_{1:2}\mathrm{M}_{2:3}...\mathrm{M}_{n-1:n}\mathrm{M}_{n:1}$ should be exactly equal to the starting pose A_1, see Fig. 1. In a realistic error-prone case the pose difference $\mathrm{E} = \mathrm{A}_1^{-1}\mathrm{A}_1'$ averaged over multiple trajectories is representative for the accuracy of a

VO method. A norm can be expressed on the pose difference $\mathrm{E} = [\mathrm{R}, \mathbf{t}]$ with $\|\mathrm{Log}(\mathrm{R})\|$ for the orientational error, see Appendix A, and the usual $\|\mathbf{t}\|$ for the positional error. We also report *errors wrt. distance traveled* (EDT).

We stress that orientational errors are more important than positional errors because orientational errors dominate long-term error growth. The orientational error nor the out-of-ground-plane positional error can be conveyed accurately by the often used 2D GPS based comparisons nor by the often used aerial-overlay plots. We only use GPS to determine the starting position and heading when creating our aerial-overlay plots. These plots are provided for esthetic reasons and the true evaluation of our research is reported strictly in quantitative terms.

Note that the goal of our method is explicitly not to *close the loop* as in [39]. Its goal is to reduce error growth for general trajectories which are not necessarily loop-like. We only exploit loops to train the coefficients Θ and to accurately measure error reductions.

A. Bias in visual odometry

In our first experiment we confirm that errors in visual odometry are biased. Fig.3 depicts the loop closure errors, i.e. E, in motion space for the 12 loops in the Doha-A data sets and for the 6 loops in the Pittsburgh-A data set.

The first thing to notice is that these errors form clusters whose mean does not coincide with the ground truth (which is the center of the depicted coordinate system). Also the deviation of each cluster wrt. its mean is smaller, or comparable to, the distance of its mean to the ground truth. It is clear that the errors of the VO system are biased. It can also be observed that clockwise and counter clock wise loops of the Doha-A data set have different orientational bias. For the Pittsburgh-A data sets, for which each data set has clockwise and counter clockwise parts, the orientational bias is similar.

This plot also indicates that significant increases in accuracy can be expected when fully preventing, or compensating for, bias. This would only leave the random error, i.e. the deviation wrt. the mean of each cluster, as source of VO drift.

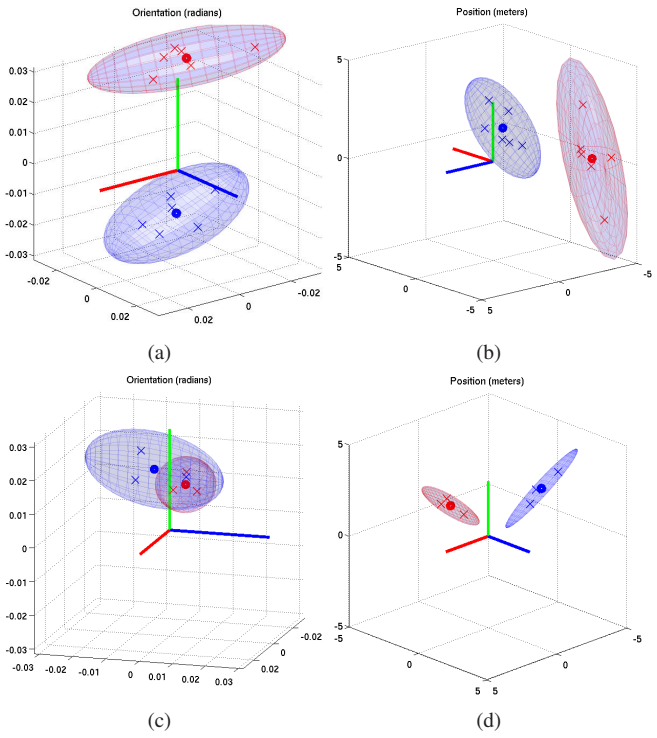
B. Compensating bias in visual odometry

We now evaluate our bias compensation technique. We start with training the bias coefficients on 4 randomly selected loops (2 clockwise and 2 counter clockwise) of the Doha-A data set and testing them on the other 8 loops. The results averaged over all test-loops are summarized in Table 1 from which it can observed that our bias compensation approach was able to reduce error growth by 50%.

The Doha-B data set encompasses a single loop. Due to the automatic key framing method used within our VO system, a single run of the VO system does not use all images of the data set. We can therefore bootstrap the data set, i.e. starting the VO system with a different initial step-size of the automatic key framing method, to obtain more statistics from this single recording. We bootstrapped 5 times and the images shared per bootstrap is around 30%. In Table

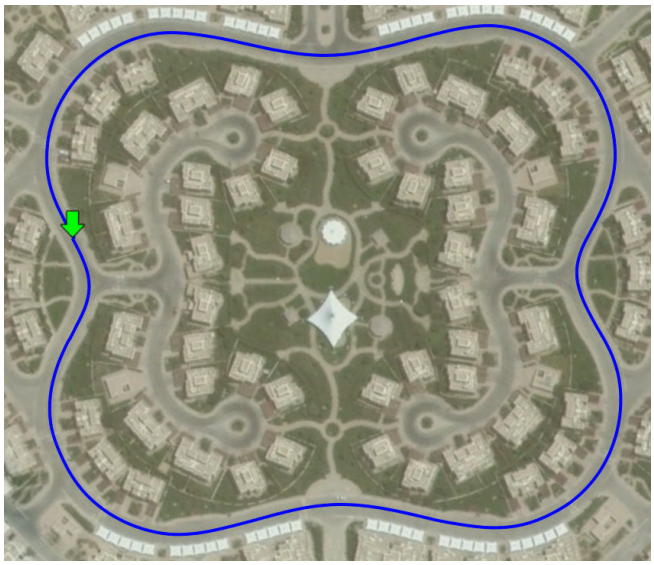
(a) (b) (c) (d)

Fig. 2. Example images from our data sets, (a) Doha-A (36 thousand frames in total), (b) Doha-B (7 thousand frames in total), (c) Pittsburgh-A (31 thousand frames in total) and (d) Pittsburgh-B (32 thousand frames in total).



(a) (b) (c) (d)

Fig. 3. Bias in motion space $\mathrm{SE}(3)$. The orientational errors at loop-closure are depicted by crosses in the left column and the positional errors by crosses in the right column. The top two plots are from the 12 loops of the Doha-A data set and the bottom two of the 6 loops of the Pittsburgh-A data set. Errors for clockwise(-to-counter-clockwise) loops are shown in red and for counter-clockwise(-to-clockwise) loops in blue. The mean of each clockwise/counter-clockwise cluster is depicted by a solid dot and their covariance ellipsoids are rendered at $3\times$ their standard deviations. The ground truth coincides with the origin of the coordinate axes. It can clearly be observed that error distributions do not have zero-mean, they are biased.

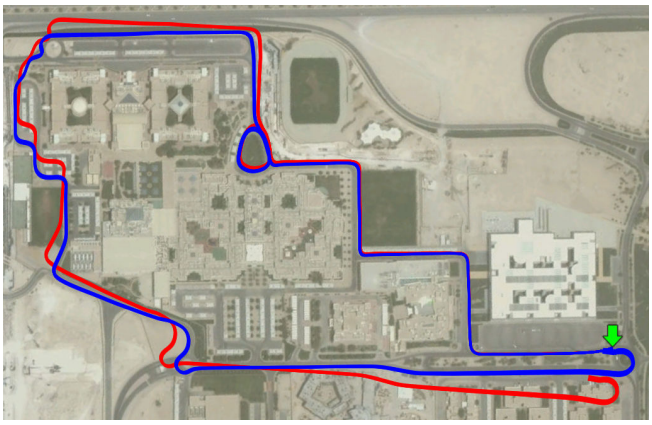


(a)

(b)

(c)

Fig. 4. Example calibration trajectories (after bias compensation), (a) 1 out of 12 trajectories of the Doha-A data set, (b) 1 out of the 3 counter-clockwise-to-clockwise trajectories of the Pittsburgh-A data set, (c) 1 out of 3 clockwise-to-counter-clockwise trajectories of the Pittsburgh-A data set. The loop-closing positions are marked by green arrows.



(a)

Fig. 5. Doha-B trajectories. The VO trajectories are shown in red and the trajectories after bias compensation in blue. The loop-closing position is marker by the green arrow.

2 we show the error reduction when training on 2 randomly selected bootstrapped runs of the Doha-B data set and testing on the other 3 bootstrapped runs. An error reduction of 25% is obtained.

Most interestingly is to see if the bias coefficients can be trained on the 12 loops of the Doha-A data set and can be applied to the Doha-B data set. These data sets do not share any images and are recorded in different environments with the same camera. These results of this experiment are summarized in Table 3. It can be observed that errors are reduced by at least 50%. The trajectories with and without bias compensation are also visualized in Fig. 5. This experiment shows that it is possible to apply the bias coefficients to more complex loops than those from which they were estimated. It also illustrates that the bias coefficients are not over-fitted when trained on the Doha-A data set.

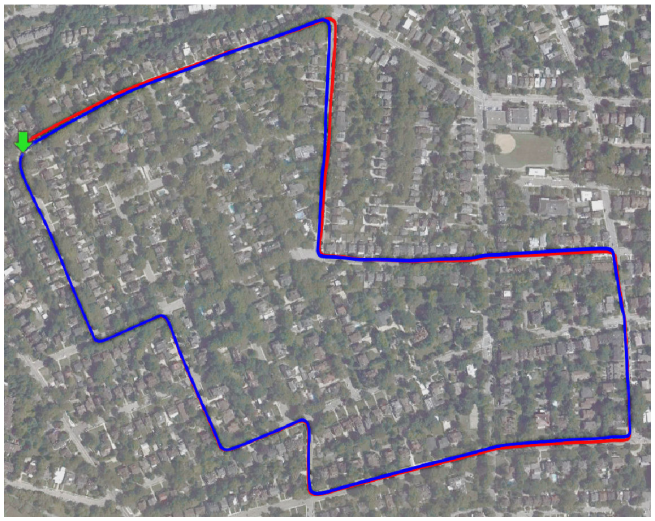
For the two loops in the Pittsburgh-B data set a similar bootstrap approach was used resulting in three tracks for each of the two loops. Due to using a higher recording frame-rate these bootstrapped runs only share 10% of their images. Again we start by training the bias coefficients on two bootstrapped runs of the Pittsburgh-B data set and testing them on the other 4 bootstrapped runs. The results in Table 4 show that an error reduction of 50% is accomplished.

Finally, Table 5 reports the results when training on the 6 loops of the Pittsburgh-A data set and testing on the 6 bootstrapped runs of the Pittsburgh-B data sets. In this case the train set and test set do not share any images and the test set is significantly more complex than the train set. Despite this challenging task our approach accomplishes 59% error reduction for orientation and 70% error reduction for position. The trajectories with and without bias compensation are also visualized in Fig. 6.

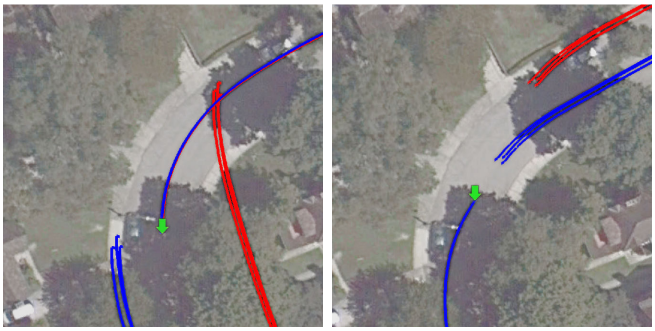
V. Conclusions

We presented a novel methodology to model and compensate for bias in visual odometry. Our model involves

(a)



(b)



(c) (d)

Fig. 6. Pittsburgh-B trajectories, (a) overview of counter clockwise loop, (b) overview of clockwise loop, (c) close-up at loop closing position for the counter clockwise loop, (d) close-up at loop closing position for the clockwise loop. The VO trajectories are shown in red and the trajectories after bias compensation in blue. The loop-closing positions are marked by the green arrows.

TABLE I

TRAIN ON DOHA-A (4 LOOPS) TEST ON DOHA-A (8 LOOPS)

	VO	Bias Comp. VO	Error Reduction	EDT
Ori.	1.59°	0.74°	**47%**	$7e^{-4}$ °/m
Pos.	3.76 m	1.84 m	**49%**	2 mm/m

TABLE II

TRAIN ON DOHA-B (2 LOOPS) TEST ON DOHA-B (3 LOOPS)

	VO	Bias Comp. VO	Error Reduction	EDT
Ori.	5.22°	1.40 °	**27%**	$6e^{-4}$ °/m
Pos.	54.69 m	12.59 m	**23%**	5 mm/m

TABLE III

TRAIN ON DOHA-A (12 LOOPS) TEST ON DOHA-B (5 LOOPS)

	VO	Bias Comp. VO	Error Reduction	EDT
Ori.	5.23°	2.31 °	**44%**	$9e^{-4}$ °/m
Pos.	54.66 m	8.70 m	**16%**	3 mm/m

TABLE IV

TRAIN ON PITTSB.-B (2 LOOPS) TEST ON PITTSB.-B (4 LOOPS)

	VO	Bias Comp. VO	Error Reduction	EDT
Ori.	2.50°	1.10 °	**44%**	$4e^{-4}$ °/m
Pos.	20.11 m	10.77 m	**54%**	4 mm/m

TABLE V

TRAIN ON PITTSB.-A (6 LOOPS) TEST ON PITTSB.-B (6 LOOPS)

	VO	Bias Comp. VO	Error Reduction	EDT
Ori.	2.50°	1.48 °	**59%**	$5e^{-4}$ °/m
Pos.	20.11 m	14.67 m	**73%**	5 mm/m

motion dependent polynomials which define a projectivity. The projectivity is applied to the estimated motion and the result is taken from projective space back to the space of Euclidean motions. An offline calibration procedure is provided to estimate the coefficients of the motion dependent polynomials. Results obtained on 25 km of binocular data show that our model can be used online to compensate for bias and that it reduces drift significantly.

REFERENCES

[1] A. I. Comport, E. Malis, and P. Rives, "Accurate Quadrifocal Tracking for Robust 3D Visual Odometry," in *IEEE International Conference on Robotics and Automation*, Apr. 2007, pp. 40–45.

[2] M. K. Dellaert, K. Ni, and Frank, "Flow separation for fast and robust stereo odometry," in *IEEE international conference on Robotics and Automation*, 2009.

[3] G. Dubbelman, W. van der Mark, and F. C. A. Groen, "Accurate and Robust Ego-Motion Estimation using Expectation Maximization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 2008, pp. 3914–3920.

[4] A. Levin and R. Szeliski, "Visual odometry and map correlation," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, June 2004, pp. 611–618.

[5] M. Maimone, Y. Cheng, and L. Matthies, "Two years of Visual Odometry on the Mars Exploration Rovers: Field Reports," *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, Mar. 2007.

[6] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, Jan. 2006. [Online]. Available: http://doi.wiley.com/10.1002/rob.20103

[7] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimoneb, "Rover Navigation using Stereo Ego-motion," *Robotics and Autonomous Systems*, vol. 43, no. 4, pp. 215–229, June 2003.
[8] Z. Zhu, T. Oskiper, S. Samarasekera, R. Kumar, and H. S. Sawhney, "Ten-fold Improvement in Visual Odometry Using Landmark Matching," in *IEEE International Conference on Computer Vision*, Oct. 2007, pp. 1–8.
[9] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*. Dover Publications, ISBN: 0486443086, 1996.
[10] B. C. Matei and P. Meer, "Optimal rigid motion estimation and performance evaluation with bootstrap," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 1, 1999, p. 345.
[11] G. Dubbelman and F. Groen, "Bias Reduction for Stereo Based Motion Estimation with Applications to Large Scale Visual Odometry," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, MIAMI, USA, June 2009, pp. 1–8.
[12] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.
[13] D. G. Lowe, "Distinctive Image Features from Scale-invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[14] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.
[15] R. Raguram, J. Frahm, and M. Pollefeys, "A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus," in *European Conference on Computer Vision*, 2008, pp. 500–513.
[16] P. L. Rosin, "Robust Pose Estimation," *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 29, no. 2, pp. 297–303, Apr. 1999.
[17] R. Subbarao, Y. Genc, and P. Meer, "Nonlinear Mean Shift for Robust Pose Estimation," in *IEEE Workshop on Applications of Computer Vision*, Feb. 2007, p. 6.
[18] O. Chum, J. Matas, and J. Kittler, "Locally Optimized RANSAC," *Lecture Notes in Computer Science*, vol. 2781, pp. 236–243, 2003.
[19] C. Engels, H. Stewénius, and D. Nistér, "Bundle Adjustment Rules," *Photogrammetric Computer Vision*, 2006.
[20] K. Konolige and M. Agrawal, "FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1066–1077, Oct. 2008.
[21] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua, "View-based Maps," *International Journal of Robotics Research*, vol. 29, no. 8, 2010.
[22] N. Sunderhauf, K. Konolige, S. Lacroix, and P. Protzel, "Visual Odometry using Sparse Bundle Adjustment on an Autonomous Outdoor Vehicle," *Tagungsband Autonome Mobile Systeme*, pp. 157–163, 2005.
[23] B. Triggs, P. F. Mclauchlan, R. I. Hartley, and A. W. Fitzibbon, "Bundle adjustment - a modern synthesis," *In Vision Algorithms: Theory and Practice, LNCS*, vol. pages, pp. 298–375, 1999.
[24] T. Clarke and F. T.G., "The development of camera calibration methods and models," *Photogrammetric Record*, vol. 16, no. 91, pp. 51–66, 1998.
[25] D. Brown, "Close-Range Camera Calibration," *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866, 1971.
[26] J. Weng, P. Cohen, and M. Herniou, "Camera Calibration with Distortion Models and Accuracy Evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965–979, 1992.
[27] J. Heikkilä and O. Sivén, "A Four-Step Camera Calibration Procedure with Implicit Image Correction," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 1106–1112.
[28] C. Strecha, W. von Hansen, L. van Gool, P. Fua, and U. Thoennessen, "On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
[29] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
[30] K. H. Strobl and G. Hirzinger, "More Accurate Pinhole Camera Calibration with Imperfect Planar Target," in *IEEE International Conference on Computer Vision, 1st IEEE Workshop on Challenges and Opportunities in Robot Perception*, 2011, pp. 1068–1075.
[31] T. Hanning, *High Precision Camera Calibration*, 1st ed. Vieweg+Teubner Verlag — Springer, 2011.
[32] J. Kannala, J. Heikkilä, and S. Brandt, "Geometric Camera Calibration," 2008.
[33] A. Gruen and T. S. Huang, *Calibration and Orientation of Cameras in Computer Vision*, 1st ed. Springer Series in Information Sciences, 2001.
[34] J. Civera, D. R. Bueno, A. J. Davison, and J. M. M. Montiel, "Camera self-calibration for sequential Bayesian structure from motion," in *IEEE International Conference on Robotics and Automation*. Ieee, May 2009, pp. 403–408. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5152719
[35] J. Courchay, A. Dalalyan, R. Keriven, and P. Sturm, "On Camera Calibration with Linear Programming and Loop Constraint Linearization," *International Journal of Computer Vision*, vol. 97, no. 1, pp. 71–90, 2012.
[36] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the World from Internet Photo Collections," *International Journal of Computer Vision*, vol. 80, no. 2, pp. 189–210, Nov. 2008.
[37] F. C. Park, "Distance Metrics on the Rigid-Body Motions with Applications to Mechanism Design," *Transactions of the ASME*, vol. 117, pp. 48–54, Mar. 1995.
[38] G. Dubbelman, "Intrinsic Statistical Techniques for Robust Pose Estimation," PhD, University of Amsterdam, 2011.
[39] G. Dubbelman, I. Esteban, and K. Schutte, "Efficient Trajectory Bending with Applications to Loop Closure," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 1–7.
[40] J. M. Selig, *Geometrical Methods in Robotics*, D. Gries and F. B. Schneider, Eds. Springer, 1996.

APPENDIX A

The exponential map for a tangent vector $\mathfrak{r}$ is

$$\mathrm{R} = \mathrm{Exp}(\mathfrak{r}) = \mathrm{I} + \sin(\|\mathfrak{r}\|)[\frac{\mathfrak{r}}{\|\mathfrak{r}\|}]_\times + (1-\cos(\|\mathfrak{r}\|))[\frac{\mathfrak{r}}{\|\mathfrak{r}\|}]_\times^2, \quad (13)$$

where

$$\left[(s_x, s_y, s_z)^\top\right]_\times = \begin{bmatrix} 0 & -s_z & s_y \\ s_z & 0 & -s_x \\ -s_y & s_x & 0 \end{bmatrix}. \quad (14)$$

It coincides with the Rodrigues rotation formula in that it produces a rotation matrix R which encodes a rotation around an axis $\frac{\mathfrak{r}}{\|\mathfrak{r}\|}$ with angle $\|\mathfrak{r}\|$.

To express the logarithmic map we need the notation

$$[\mathrm{S}]_\uparrow = \begin{bmatrix} 0 & -s_z & s_y \\ s_z & 0 & -s_x \\ -s_y & s_x & 0 \end{bmatrix}_\uparrow = (s_x, s_y, s_z)^\top \quad (15)$$

to map a skew symmetric matrix S into a vector. With this notation the logarithmic map is

$$\mathfrak{r} = \mathrm{Log}(\mathrm{R}) = \begin{cases} \left[\theta \frac{(\mathrm{R}-\mathrm{R}^\top)}{2\sin(\theta)}\right]_\uparrow & , \quad \theta \neq 0 \\ (0,0,0)^\top & , \quad \theta = 0 \end{cases} \quad (16)$$

with

$$\theta = \arccos\left(\frac{\mathrm{trace}(\mathrm{R})-1}{2}\right) \quad (17)$$

[40]. For a rotation matrix R expressing a rotation around normalized axis $\mathbf{r}$ with angle α it effectively produces the tangent vector $\mathfrak{r} = \alpha\mathbf{r}$.