Enforcing the General Planar Motion Model: Bundle Adjustment for Planar Scenes

Marcus Valtonen Örnhag[1(✉)] and Mårten Wadenbäck[2]

[1] Centre for Mathematical Sciences, Lund University, Lund, Sweden
marcus.valtonen_ornhag@math.lth.se
[2] Department of Mathematical Sciences,
Chalmers University of Technology and the University of Gothenburg,
Gothenburg, Sweden
marten.wadenback@chalmers.se

Abstract. In this paper we consider the case of planar motion, where a mobile platform equipped with two cameras moves freely on a planar surface. The cameras are assumed to be directed towards the floor, as well as being connected by a rigid body motion, which constrains the relative motion of the cameras and introduces new geometric constraints. In the existing literature, there are several algorithms available to obtain planar motion compatible homographies. These methods, however, do not minimise a physically meaningful quantity, which may lead to issues when tracking the mobile platform globally. As a remedy, we propose a bundle adjustment algorithm tailored for the specific problem geometry. Due to the new constrained model, general bundle adjustment frameworks, compatible with the standard six degree of freedom model, are not directly applicable, and we propose an efficient method to reduce the computational complexity, by utilising the sparse structure of the problem. We explore the impact of different polynomial solvers on synthetic data, and highlight various trade-offs between speed and accuracy. Furthermore, on real data, the proposed method shows an improvement compared to generic methods not enforcing the general planar motion model.

Keywords: Planar motion · Bundle adjustment · SLAM · Visual Odometry

1 Introduction

The prototypical problem in geometric computer vision is the so called Structure from Motion (SfM) problem [12,24]; the objective of which is to recover the scene geometry and camera poses from a collection of images of a scene. The SfM problem has, in some form or other, been studied since the very earliest days of photography, and many fundamental aspects of SfM were well understood already by the end of the 19th century [23]. Solving SfM problems of meaningful size and with actual image data, however, has been made possible only through the computerisation efforts that were commenced in the late 1970s, and which have since led to increasingly automatic methods for SfM. Modern SfM systems, e.g. *Bundler* [22] and other systems under the wider

M. De Marsico et al. (Eds.): ICPRAM 2019, LNCS 11996, pp. 119–135, 2020.
https://doi.org/10.1007/978-3-030-40014-9_6

BigSFM banner[1] [1,8], have managed to produce impressive city-scale reconstructions from large unordered and unlabelled sets of images.

A major paradigm in SfM, which has proven hugely successful, is Bundle Adjustment (BA) [26], which treats SfM as a large optimisation problem. With a parameterisation describing the scene geometry and the cameras, BA employs numerical optimisation techniques to find parameter values which best explain the observed images. Here, 'best' is determined by evaluating a cost function which is often—but not always—chosen as the sum of squared geometric reprojection errors. The BA formulation of the SfM problem puts it in a unified framework which still has extensive model flexibility, e.g. with regards to (a) assumptions on the camera calibration, (b) different cost functions, and (c) different parameterisations of the cameras and the scene geometry—including implicit and explicit constraints to enforce a particular motion model.

While camera based Simultaneous Localisation and Mapping (SLAM) and Visual Odometry (VO) can be thought of as special classes of SfM, the computational effort to approach SLAM and VO via BA has traditionally been inhibiting, and for this reason, BA has mostly been used in offline batch processing systems such as the BigSFM systems mentioned earlier. During the last two decades, however, SLAM and VO systems have started incorporating regular BA steps to improve the consistency of the reconstruction and the precision of the camera pose estimation. Performance improvements across the spectrum—the algorithms, their implementation, the hardware—are paving the way for application specific BA to make its entrance in the area of real-time systems.

Especially in the case of visual SLAM, there are a number of factors which can be exploited to alleviate the computational burden compared to a more generic SfM system. The images are acquired in an ordered sequence, and this can significantly speed up the search for correspondences by avoiding the expensive 'all-vs-all' matching. Additionally, a suitable motion model may often be incorporated in a SLAM system, which can be used e.g. (a) to further speed up the search for correspondences by predicting feature locations in subsequent images [5,6], (b) to facilitate faster and more accurate local motion estimation via nonholonomic constraints [20,21,39] or other constraints which reduce the set of parameters [29,33], or (c) to enforce globally a planar motion assumption on the camera motion [10,18,32].

In this paper, we present a BA approach to visual SLAM for the case of a stereo rig, where the cameras do not necessarily have an overlapping field of view, and where each of the two cameras move in parallel to a common ground plane. The present paper is an extension of the system described earlier in [30], to which a more extensive experimental evaluation has been added. In particular, we have investigated how initialisation using planar motion compatible homographies based on minimal [33] or non-minimal [29] polynomial solvers affect the final reconstruction.

2 Related Work

Planar Motion is a frequently occurring constrained camera motion, which arises naturally when cameras are attached to a ground vehicle operating on a planar ground

[1] http://www.cs.cornell.edu/projects/bigsfm/.

surface. As mentioned in the introduction, deliberately enforcing planar motion can help to improve the quality of the reconstruction.

An early SfM approach to plane constrained visual navigation was proposed by Wiles and Brady [34,35]. They suggested a hierarchical framework of camera parameterisations, and explored in detail the remaining structural ambiguity for each of these. The lasting contribution of this work lies chiefly in its classification and description of the different modes of motion. The least ambiguous level in the case of planar motion—which they called α-structure—contains only an arbitrary global scaling ambiguity and an arbitrary planar Euclidean transformation parallel to the ground plane, and is precisely the level aimed at in the present paper.

If the optical axis of the camera is either orthogonal or parallel to the ground plane, the parameterisation can be much simplified compared to the general case described by Wiles and Brady. This situation can of course also be achieved if the camera tilt is known with sufficient precision to allow a transformation to, e.g., an overhead view. An approach for this case by Ortín and Montiel parameterises the essential matrix explicitly in the motion parameters, and then estimates the parameters using either a linear three-point method or a non-linear two-point method [18]. Scaramuzza used essentially the same parameterisation of the essential matrix, but combined it with an additional nonholonomic constraint based on the assumption that the local motion is a circular motion [20,21]. Because of this additional constraint, the local motion can be computed from only one point correspondence, and this allows for an exceptionally efficient outlier removal scheme based on histogram voting.

Since the essential matrix is a homogeneous entity, it does not capture the length of the translation, and the maintaining of a consistent global scale then requires some additional information. One possibility for this, explored by Chen and Liu, is to add a second camera [4]. This allows the length of the local translation to be computed in terms of the distance between the two cameras, and since this remains constant, it provides a way to prevent scale drift.

If the camera is oriented such that it views a reasonable part of the ground plane, an alternative to using the essential matrix is to instead use homographies for the local motion estimation. This has the advantage that the length of the translation between frames can be expressed in terms of the height above the ground plane, which thus defines the global scale. The homography based approach by Liang and Pears is based on an eigendecomposition of the homography matrix, and it is shown that the rotation about the vertical axis can be determined from the eigenvalues, regardless of the camera tilt [14]. Hajjdiab and Laganière parameterised the homography matrix under the assumption of only one tilt angle, and then transformed the images into a synthetic overhead view to compute the residual rigid body motion in the plane [10].

A more recent homography based method by Wadenbäck and Heyden, which also exploits a decoupling of the camera tilt and the camera motion, uses an alternating iterative estimation scheme to compute the two tilt angles and the three motion parameters [31,32]. Zienkiewicz and Davison solved the same 5-DoF problem through a joint non-linear optimisation over all five parameters to achieve a dense matching of successive views, with the implementation running on a GPU to reach very high frame rates [39].

Valtonen Örnhag and Heyden extended the general 5-DoF situation to handle a binocular setup, where the two cameras are connected by a fixed (but unknown) rigid body motion in 3D, and where the fields of view do not necessarily overlap [27,28].

Bundle Adjustment is used to optimise a set of structure and motion parameters, and is typically performed over several camera views. Triggs et al. give an excellent overview [26]. Since the number of parameters optimised over is in most cases very large, naïve implementations will not work, and care must be taken to exploit the problem structure (e.g. the sparsity pattern of the Jacobian).

Generic software packages for bundle adjustment, which use sparsity of the Jacobian matrix together with Schur complementation to speed up the computations, include *SBA* (Sparse Bundle Adjustment) by Lourakis and Argyros, *sSBA* (Sparse Sparse Bundle Adjustment) by Konolige, and *SSBA* (Simple Sparse Bundle Adjustment) by Zach [13,16,37].

Additional performance gains may sometimes be obtained through parallelisation. GPU accelerated BA systems using parallelised versions of the Levenberg–Marquardt algorithm [11] and the conjugate gradients method [36] have been presented e.g. by Hänsch et al. and by Wu et al.. More recently, distributed approaches by e.g. Eriksson et al. and by Zhang et al. have employed splitting methods to make very large SfM problems tractable [7,38].

The present paper extends the sparse bundle adjustment system for the binocular planar motion case by Valtonen Örnhag and Wadenbäck. The aim of our approach is to exploit the particular structure in the Jacobian which arises due to the planar motion assumption for the two cameras. We demonstrate how this particular situation can be attacked via the use of nested Schur complementations when solving the normal equations. In comparison to the earlier paper [30], we have significantly extended the experimental evaluation of the system. Additionally, we have investigated the effect of enforcing the planar motion assumption earlier on a local level, by using homographies estimated such that they are compatible with this assumption [29,33].

3 Theory

3.1 Problem Geometry

The geometrical situation we consider in this paper is that of two cameras which have been rigidly mounted onto a mobile platform. Due to this setup, which is illustrated in Fig. 1, the cameras are connected by a rigid body motion which remains constant over time but which is initially not known. Each camera is assumed to be mounted in such a way that it can view a portion of the ground plane, but it is *not* a requirement that the cameras have any portion of their fields of view in common. The world coordinate system is chosen such that the ground plane is positioned at $z = 0$, whereas the cameras move in the planes $z = a$ and $z = b$, respectively. We may also, without loss of generality, assume that the centre of rotation of the mobile platform coincides with the centre of the first camera.

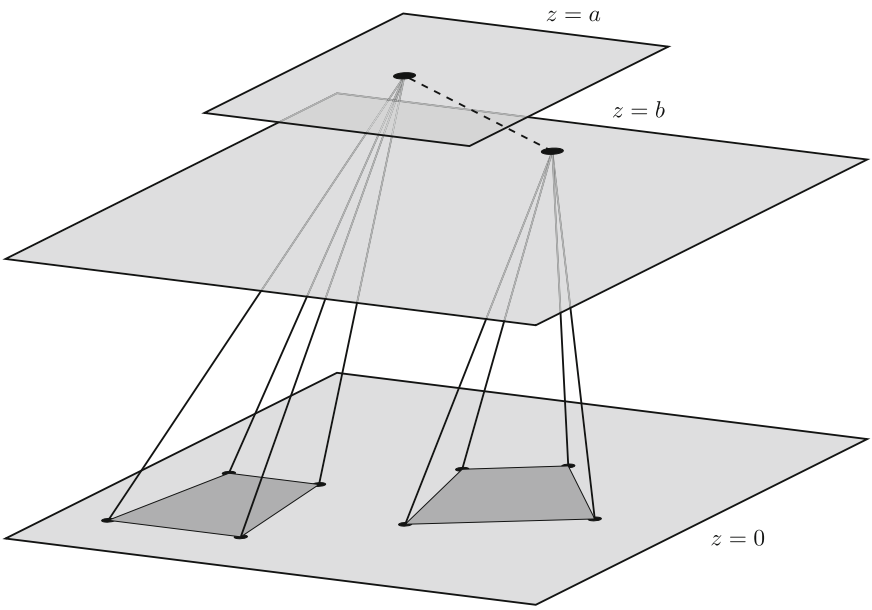


Fig. 1. Illustration of the problem geometry considered in this paper. Two cameras are assumed to be rigidly mounted on a mobile platform, and may be positioned at different heights above, the ground floor, hence move in the planes $z = a$ and $z = b$. Due to the rigidity assumption, the relative orientation between them are constant, and so is the overhead tilt. Figure reproduced from [30].

3.2 Camera Parameterisation

We shall adopt the camera parameterisation for internally calibrated monocular planar motion that was introduced in [31]. With this parameterisation, the camera matrix associated with the image taken at position j will be

$$\boldsymbol{P}^{(j)} = \boldsymbol{R}_{\psi\theta}\boldsymbol{R}_{\varphi}^{(j)}[\boldsymbol{I} \mid -\boldsymbol{t}^{(j)}], \tag{1}$$

where $\boldsymbol{R}_{\psi\theta}$ is a rotation θ about the y-axis followed by a rotation of ψ about the x-axis. The motion of the mobile platform contains for each frame a rotation $\varphi^{(j)}$ about the z-axis, encoded as $\boldsymbol{R}_{\varphi}^{(j)}$, and a vector $\boldsymbol{t}^{(j)}$ for the translational part. The second camera, which is related to the first camera through a constant rigid body motion, uses the parameterisation

$$\boldsymbol{P}'^{(j)} = \boldsymbol{R}_{\psi'\theta'}\boldsymbol{R}_{\eta}\boldsymbol{T}_{\boldsymbol{\tau}}(b)\boldsymbol{R}_{\varphi}^{(j)}[\boldsymbol{I} \mid -\boldsymbol{t}^{(j)}], \tag{2}$$

introduced in [27]. Here, ψ' and θ' are the tilt angles (defined in the same way as for the first camera), $\boldsymbol{\tau}$ is the relative translation between the camera centres and η is the constant rotation about the z-axis relative to the first camera. We do not assume any prior knowledge of these constant parameters. Define the translation matrix $\boldsymbol{T}_{\boldsymbol{\tau}}(b)$ as

$\boldsymbol{T}_{\boldsymbol{\tau}}(b) = \boldsymbol{I} - \boldsymbol{\tau}\boldsymbol{n}^\intercal/b$, where $\boldsymbol{\tau} = (\tau_x, \tau_y, 0)^\intercal$, $\boldsymbol{n}$ is a floor normal and b is the height above the ground floor. The global scale ambiguity allows us to set $a = 1$ without any loss of generality.

4 Prerequisites

4.1 Geometric Reprojection Error

The particular BA problem considered in this paper concerns the minimisation of the *geometric reprojection error* in the two views over the entire motion sequence. In order to write down this cost function explicitly we need to introduce some additional notation.

For this purpose, let the two cameras at a particular position j be given by the expressions in (1) and (2), respectively. We use the homogeneous representation $\boldsymbol{X}_i = (X_i, Y_i, 0, 1)^\intercal$ to parameterise the estimate of the i:th 3D point, corresponding to the measured image point with inhomogeneous representations $\boldsymbol{x}_i^{(j)}$ in the first camera and $\boldsymbol{x}_i'^{(j)}$ in the second. Let $\hat{\hat{\boldsymbol{x}}}_i^{(j)}$ and $\hat{\hat{\boldsymbol{x}}}_i'^{(j)}$ be the inhomogeneous representations for the projections into the two views, i.e.

$$\begin{bmatrix} \hat{\hat{\boldsymbol{x}}}_i^{(j)} \\ 1 \end{bmatrix} \sim \boldsymbol{P}^{(j)} \boldsymbol{X}_i \qquad \text{and} \qquad \begin{bmatrix} \hat{\hat{\boldsymbol{x}}}_i'^{(j)} \\ 1 \end{bmatrix} \sim \boldsymbol{P}'^{(j)} \boldsymbol{X}_i. \tag{3}$$

Given N stereo camera locations and M scene points, the geometric reprojection error that we seek to minimise can now be written concisely as

$$E(\boldsymbol{\beta}) = \sum_{i=1}^{N} \sum_{j=1}^{M} \|\boldsymbol{r}_{ij}\|_2^2 + \|\boldsymbol{r}'_{ij}\|_2^2, \tag{4}$$

where $\boldsymbol{\beta}$ is the parameter vector consisting of the camera parameters and the scene point parameters, and where $\boldsymbol{r}_{ij}$ and $\boldsymbol{r}'_{ij}$ are the residuals

$$\boldsymbol{r}_{ij} = \boldsymbol{x}_i^{(j)} - \hat{\hat{\boldsymbol{x}}}_i^{(j)} \qquad \text{and} \qquad \boldsymbol{r}'_{ij} = \boldsymbol{x}_i'^{(j)} - \hat{\hat{\boldsymbol{x}}}_i'^{(j)}. \tag{5}$$

4.2 The Levenberg–Marquardt Algorithm

We will in this approach use the Levenberg–Marquardt algorithm (LM) when minimising (4). There are of course other alternatives to the LM algorithm, e.g. the dog-leg solver [15] and preconditioned CG [3]; however, LM is one of the most commonly used algorithms for BA, and is used in major modern systems such as SBA [16] and sSBA [13]. Note that these systems do not account for the particular problem geometry that we consider in this paper, which forces some extrinsic parameters to be shared among all camera matrices.

We will not go into details of the LM algorithm here—please refer to more extensive treatments in e.g. [26] and [16] for a more complete discussion—but for future reference we simply recall that it works by iteratively solving the augmented normal equations

$$(\boldsymbol{J}^\intercal \boldsymbol{J} + \mu \boldsymbol{I})\, \boldsymbol{\delta} = \boldsymbol{J}^\intercal \boldsymbol{\varepsilon} \tag{6}$$

until some convergence criteria have been met. Here $\boldsymbol{J}$ is the Jacobian associated with the cost function (4), $\boldsymbol{\varepsilon}$ is the residual vector, and $\mu \geq 0$ is the iteratively adjusted *damping parameter* of the LM algorithm.

4.3 Obtaining an Initial Solution for the Camera Parameters

Homographies can be estimated in a number of different ways; however, the classical approach is to compute point correspondences from matching robust feature points in subsequent images. Popular feature extraction algorithms include SIFT [17] and SURF [2], but many more are available and implemented in various computer vision software. When the putative point correspondences have been matched a popular choice is to use RANSAC (or similar frameworks) to robustly estimate a homography. Such an approach is suitable in order to discard mismatched feature points. A well-known method is the Direct Linear Transform (DLT); however, it requires four point correspondences, and does not generate a homography compatible with the general planar motion model. A good rule of thumb is to use a minimal amount of point correspondences, since the probability of finding a set of points containing only inliers decreases with each additional point that is used. However, as e.g. Pham et al. point out, for very severely noisy data it may in some cases still be preferable to use a non-minimal set [19].

In [33] a minimal solver compatible with the general planar motion model was studied. It was shown that a homography compatible with the general planar motion model must fulfil 11 quartic constraints, and that, a minimal solver only requires 2.5 point correspondences. In a recent paper, a variety of different non-minimal polynomial solvers are considered, partly because of execution time, but also because of sensitivity to noise [29]. These non-minimal solvers enforce a subset of the necessary and sufficient conditions for compatibility with the general planar motion model, thus enforcing a weaker form of it. By accurately making a trade-off between fitting the model constraints (i.e. using more model constraints) and tuning to data (i.e. using more point correspondences), one can increase the performance for noisy data. It is important to note that the assumption of constant tilt parameters cannot be enforced by only considering a single homography, and, therefore, pre-optimisation in an early step of the complete SfM pipeline is not guaranteed to yield better performance.

Once the homographies are obtained, one may enforce the constant tilt constraint by employing the method proposed by Wadenbäck and Heyden [32], to obtain a good initial solution for the monocular case. The method starts by computing the overhead tilt $\boldsymbol{R}_{\psi\theta}$ from an arbitrary number of homographies, followed by estimating the translation and orientation about the floor normal.

The method by Valtonen Örnhag and Heyden [27] extended the method to include the stereo case, and starts off by treating the two stereo trajectories individually, and estimates the tilt parameters by employing the monocular method described in the previous paragraph. Once the monocular parameters are known for the individual tracks, the relative pose can be extracted by minimising an algebraic error in the relative translation between the cameras, followed by estimating the relative orientation about the floor normal.

4.4 Obtaining an Initial Solution for the Scene Points

Linear triangulation of scene points does not guarantee that all points lie in a plane, and the resulting initial solution would not be compatible with the general planar motion model. In order to obtain a physically meaningful solution we make use of the fact that there is a homography relating the measured points and the ground plane positioned at $z = 0$.

Given a camera $\boldsymbol{P}$, an image point $\boldsymbol{x}$ and the corresponding scene point $\boldsymbol{X} \sim (X, Y, 0, 1)^\intercal$, they are related by $\boldsymbol{x} \sim \boldsymbol{P}\boldsymbol{X} = \boldsymbol{H}\tilde{\boldsymbol{X}}$, where $\boldsymbol{H}$ is the sought homography. By denoting the i:th column of $\boldsymbol{P}$ by $\boldsymbol{P}_i$, it may be expressed as $\boldsymbol{H} = [\boldsymbol{P}_1\ \boldsymbol{P}_2\ \boldsymbol{P}_4]$, where $\tilde{\boldsymbol{X}} \sim (X, Y, 1)^\intercal$ contains the unknown scene point coordinates. It follows that the corresponding scene point can be extracted from $\tilde{\boldsymbol{X}} \sim \boldsymbol{H}^{-1}\boldsymbol{x}$.

In the presence of noise, using more than one camera results in different scene points, which all will be projected onto the plane $z = 0$. In order to triangulate the points we compute the centre mass; such an approach is computationally inexpensive, however, it is not robust to outliers, which have to be excluded in order to get a reliable result.

5 Planar Motion Bundle Adjustment

5.1 Block Structure of the Jacobian

Denote the unknown and constant parameters for the first camera path by $\boldsymbol{\gamma} = (\psi, \theta)$ and the second camera path by $\boldsymbol{\gamma}' = (\psi', \theta', \tau_x, \tau_y, b, \eta)$. Furthermore, let the non-constant parameters for position j be denoted by $\boldsymbol{\xi}_j = (\varphi^{(j)}, t_x^{(j)}, t_y^{(j)})$. Given N stereo camera positions and M scene points, the following, highly structured Jacobian $\boldsymbol{J}$, is obtained

$$\boldsymbol{J} = \begin{bmatrix}
\boldsymbol{\Gamma}_{11} & & & \boldsymbol{A}_{11} & & & & \boldsymbol{B}_{11} & & & \\
\vdots & & & & \ddots & & & \vdots & & & \\
\boldsymbol{\Gamma}_{1N} & & & & & \boldsymbol{A}_{1N} & & \boldsymbol{B}_{1N} & & & \\
\vdots & & & \vdots & \vdots & \vdots & & & \ddots & & \\
\boldsymbol{\Gamma}_{M1} & & & \boldsymbol{A}_{M1} & & & & & & \boldsymbol{B}_{M1} & \\
\vdots & & & & \ddots & & & & & \vdots & \\
\boldsymbol{\Gamma}_{MN} & & & & & \boldsymbol{A}_{MN} & & & & \boldsymbol{B}_{MN} & \\
 & \boldsymbol{\Gamma}'_{11} & & \boldsymbol{A}'_{11} & & & & \boldsymbol{B}'_{11} & & & \\
 & \vdots & & & \ddots & & & \vdots & & & \\
 & \boldsymbol{\Gamma}'_{1N} & & & & \boldsymbol{A}'_{1N} & & \boldsymbol{B}'_{1N} & & & \\
 & \vdots & & \vdots & \vdots & \vdots & & & \ddots & & \\
 & \boldsymbol{\Gamma}'_{M1} & & \boldsymbol{A}'_{M1} & & & & & & \boldsymbol{B}'_{M1} & \\
 & \vdots & & & \ddots & & & & & \vdots & \\
 & \boldsymbol{\Gamma}'_{MN} & & & & \boldsymbol{A}'_{MN} & & & & \boldsymbol{B}'_{MN} &
\end{bmatrix}, \tag{7}$$

where we use the following notation for the derivative blocks

$$\begin{aligned}
\boldsymbol{A}_{ij} &= \frac{\partial \boldsymbol{r}_{ij}}{\partial \boldsymbol{\xi}_j}, \quad \boldsymbol{B}_{ij} = \frac{\partial \boldsymbol{r}_{ij}}{\partial \tilde{\boldsymbol{X}}_i}, \qquad \boldsymbol{\Gamma}_{ij} = \frac{\partial \boldsymbol{r}_{ij}}{\partial \boldsymbol{\gamma}}, \\
\boldsymbol{A}'_{ij} &= \frac{\partial \boldsymbol{r}'_{ij}}{\partial \boldsymbol{\xi}_j}, \quad \boldsymbol{B}'_{ij} = \frac{\partial \boldsymbol{r}'_{ij}}{\partial \tilde{\boldsymbol{X}}_i}, \qquad \boldsymbol{\Gamma}'_{ij} = \frac{\partial \boldsymbol{r}'_{ij}}{\partial \boldsymbol{\gamma}'},
\end{aligned} \tag{8}$$

where $\tilde{\boldsymbol{X}}_i = (X_i, Y_i)$ are the unknown scene coordinates. This can be written in a more compact manner as

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{\Gamma} & \boldsymbol{0} & \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{\Gamma}' & \boldsymbol{A}' & \boldsymbol{B}' \end{bmatrix}. \tag{9}$$

5.2 Utilising the Sparse Structure

In SfM, the number of scene points is often significantly larger than the number of cameras, which makes Schur complementation tractable, and can significantly decrease the execution time. Standard Schur complementation is, however, not directly applicable due to the constant parameters giving rise to the blocks $\boldsymbol{\Gamma}$ and $\boldsymbol{\Gamma}'$. We will, however, show in this section, that it is indeed possible to use *nested Schur complements*, i.e. to recursively apply Schur complements to different parts, and that, in fact, several of the intermediate computations can be stored, thus drastically decreasing the computational time. First, note that the approximate Hessian $\boldsymbol{J}^\intercal\boldsymbol{J}$, in compact form, can be written

$$\boldsymbol{J}^\intercal\boldsymbol{J} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{E} \\ \boldsymbol{E}^\intercal & \boldsymbol{D} \end{bmatrix}. \tag{10}$$

Here the contribution from the constant parameters are stored in $\boldsymbol{C}$, the contribution from the nonconstant parameters and the scene points are stored in $\boldsymbol{D}$, and the mixed contributions are stored in $\boldsymbol{E}$. Furthermore, the matrix $\boldsymbol{D}$ can be written as

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{U} & \boldsymbol{W} \\ \boldsymbol{W}^\intercal & \boldsymbol{V} \end{bmatrix}, \tag{11}$$

with block diagonal matrices $\boldsymbol{U} = \operatorname{diag}(\boldsymbol{U}_1, \ldots, \boldsymbol{U}_N)$ and $\boldsymbol{V} = \operatorname{diag}(\boldsymbol{V}_1, \ldots, \boldsymbol{V}_M)$, where

$$\begin{aligned} \boldsymbol{U}_j &= \sum_{i=1}^{M} \boldsymbol{A}_{ij}^\intercal \boldsymbol{A}_{ij} + \boldsymbol{A}_{ij}'^\intercal \boldsymbol{A}_{ij}', \\ \boldsymbol{V}_i &= \sum_{j=1}^{N} \boldsymbol{B}_{ij}^\intercal \boldsymbol{B}_{ij} + \boldsymbol{B}_{ij}'^\intercal \boldsymbol{B}_{ij}', \\ \boldsymbol{W}_{ij} &= \boldsymbol{A}_{ij}^\intercal \boldsymbol{B}_{ij} + \boldsymbol{A}_{ij}'^\intercal \boldsymbol{B}_{ij}'. \end{aligned} \tag{12}$$

First, note that the system $(\boldsymbol{D}+\mu\boldsymbol{I})\boldsymbol{\delta} = \boldsymbol{\varepsilon}$, where $\boldsymbol{D}$ is defined as in (11), is not affected by the constant parameters. Such a system reduces to that of the unconstrained case, which can be solved using standard SfM frameworks, such as SBA, or other packages utilising Schur complementation.

We will now show how to efficiently treat the decomposition of (10) as nested Schur complements, by reducing the problem to a series of subproblems of the form used in SBA and other computer vision software packages. In order to do so, consider the augmented normal equations (6) in block form

$$\begin{bmatrix} \boldsymbol{C}^* & \boldsymbol{E} \\ \boldsymbol{E}^\intercal & \boldsymbol{D}^* \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_c \\ \boldsymbol{\delta}_d \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varepsilon}_c \\ \boldsymbol{\varepsilon}_d \end{bmatrix}, \tag{13}$$

where $\boldsymbol{C}^* = \boldsymbol{C} + \mu\boldsymbol{I}$ and $\boldsymbol{D}^* = \boldsymbol{D} + \mu\boldsymbol{I}$ denote the augmented matrices, with the added contribution from the damping factor μ, as in (6). Now, utilising Schur complementation yields

$$\begin{bmatrix} \boldsymbol{C}^* - \boldsymbol{E}\boldsymbol{D}^{*-1}\boldsymbol{E}^\intercal & \boldsymbol{0} \\ \boldsymbol{E}^\intercal & \boldsymbol{D}^* \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_c \\ \boldsymbol{\delta}_d \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varepsilon}_c - \boldsymbol{E}\boldsymbol{D}^{*-1}\boldsymbol{\varepsilon}_d \\ \boldsymbol{\varepsilon}_d \end{bmatrix}. \tag{14}$$

Let us take a step back and reflect over the consequences of the above equation. First, note that $\boldsymbol{D}^{*-1}$ is present in (14) twice, and is infeasible to compute explicitly. This can be avoided by introducing the auxiliary variable $\boldsymbol{\delta}_{\text{aux}}$, defined as

$$\boldsymbol{D}^*\boldsymbol{\delta}_{\text{aux}} = \boldsymbol{\varepsilon}_d. \tag{15}$$

Again, such as system is not affected by the constraints of the constant parameters, and can be solved with standard computer vision software. Furthermore, we may introduce $\boldsymbol{\Delta}_{\text{aux}}$ and solve the system $\boldsymbol{D}^*\boldsymbol{\Delta}_{\text{aux}} = \boldsymbol{E}^\intercal$ in a similar manner by iterating over the columns of $\boldsymbol{E}^\intercal$. Since the number of constant parameters are low, such an approach is highly feasible, but the performance can be further boosted by storing the Schur complement and the intermediate matrices not depending on the right-hand side, from the previous computations of obtaining $\boldsymbol{\delta}_{\text{aux}}$ from (15).

When the auxiliary variables have been obtained, we proceed to compute $\boldsymbol{\delta}_c$ from

$$\left(\boldsymbol{C}^* - \boldsymbol{E}\boldsymbol{\Delta}_{\text{aux}}\right)\boldsymbol{\delta}_c = \boldsymbol{\varepsilon}_c - \boldsymbol{E}\boldsymbol{\delta}_{\text{aux}}, \tag{16}$$

and, lastly, $\boldsymbol{\delta}_d$ by back-substitution

$$\boldsymbol{D}^*\boldsymbol{\delta}_d = \boldsymbol{\varepsilon}_d - \boldsymbol{E}^\intercal\boldsymbol{\delta}_c. \tag{17}$$

Again, by storing the computation of the Schur complement and intermediate matrices, these can be reused to solve (17) efficiently.

6 Experiments

6.1 Initial Solution

The inter-image homographies were estimated using the MSAC algorithm [25] from point correspondences by extracting SURF keypoints and applying a KNN algorithm to establish the matches. In the first experiment, we use the standard DLT solver, the minimal 2.5 pt solver [33] and the four different polynomial solvers studied in [29].

In all experiments we use all available homographies, and extract the monocular parameters using the method proposed in [32]. Similarly, the binocular parameters were extracted using [27]. When all motion parameter have been estimated the camera path is reconstructed by aligning the first camera position to the origin, and use the estimated camera poses to triangulate the scene points as in Sect. 4.4.

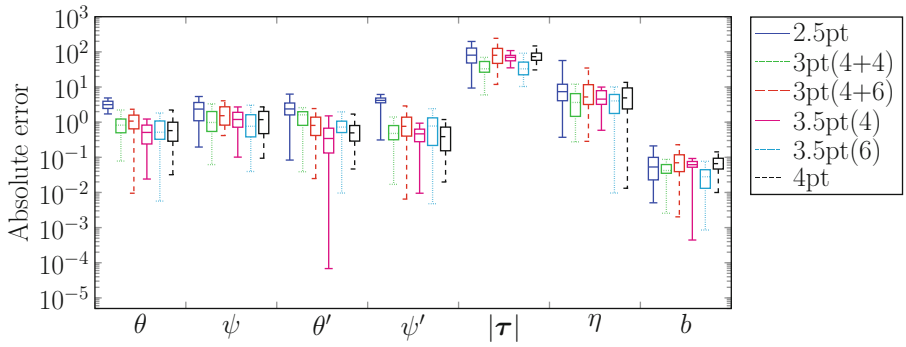


Fig. 2. Errors before applying BA. The angles are measured in degrees, and the translation in pixels.

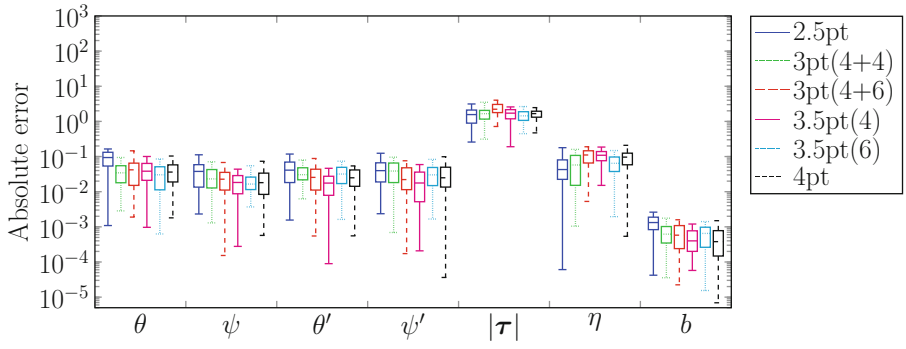


Fig. 3. Errors after applying BA. The angles are measured in degrees, and the translation in pixels.

6.2 Impact of Pre-processing Steps

In this section we work with synthetic data in order to have access to accurate ground truth data. We generate an image sequence from a high-resolution image, depicting a floor, which is the typical use case for the algorithm. This is done by constructing a path compatible with the general planar motion model, and project that part of the floor through the camera and extract the corresponding image. The resulting image is 400×400 pixels, and all cameras are set to a field of view of $90°$, with parameters $\psi = -2°$, $\theta = -4°$, $\psi' = 6°$, $\theta' = 4°$, $\boldsymbol{\tau} = \begin{pmatrix} 0 & 400 \end{pmatrix}$, $\eta = 20°$ and $b = 1$. In total, the image sequence consists of 20 images. Lastly, to simulate image noise, we add Gaussian noise with a standard deviation of five pixels, where the pixel depth allows 256 different intensities per channel.

In order to study the difference in accuracy for the constant parameters, we proceed by obtaining homographies as described in Sect. 6.1, using the minimal 2.5 point solver [33], four non-minimal solvers [29] and the DLT equations (4 point). The accuracy, over 50 iterations, is reported before BA, in Fig. 2, and after BA, in Fig. 3. In gen-

eral, the overall performance of the solvers are almost equal; however, some tendencies are present. The minimal solver performs worse than the other before BA, but this deviation is smaller after BA, although present. One possible explanation is that the general planar motion model is enforced too early in the pipeline—in fact, since it is enforced between two consecutive image pairs only, it does not guarantee that the overhead tilt is constant throughout the entire sequence, and thus, in the presence of noise, the error propagates differently, compared to the other methods that partially (non-minimal) or completely (DLT) tune to the data.

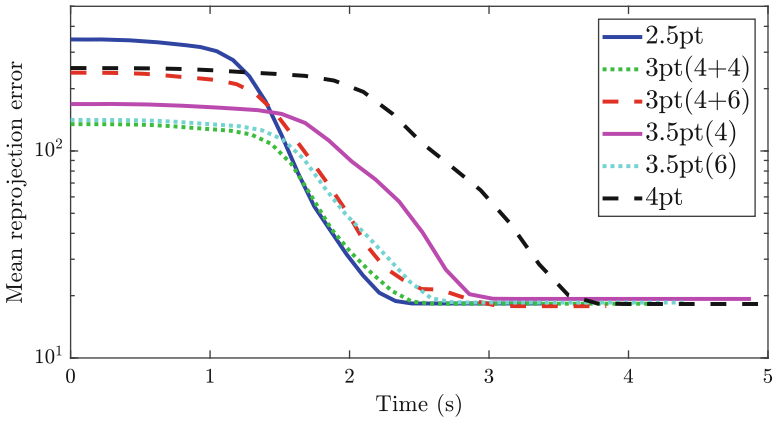


Fig. 4. Mean reprojection error vs execution time (s) over 50 iterations.

Overall, the performance is acceptable after BA, regardless of how the homographies are obtained. Hence, the differentiating factors come down to convergence rates. For the same problem instances as in the previous section we also save the convergence history in terms of the mean reprojection error and the execution time in seconds. The results are shown in Fig. 4. It is clear that the execution time for reaching convergence increase with the number of point correspondences required by the polynomial solvers. This suggests that one can make a trade-off between speed and accuracy when designing a planar motion compatible BA framework by choosing different solvers, in order to suit ones specific needs. Note, however, that the implementation used in this paper is a native Matlab implementation, and that the absolute timings can be greatly improved by careful implementation; however, the relative execution time between the solvers will be similar.

6.3 Bundle Adjustment Comparison

In this section we compare the qualitative difference between enforcing the general planar motion model versus the general unconstrained six degree of freedom model on a real dataset. Currently, there is not a good or well-established dataset compatible with

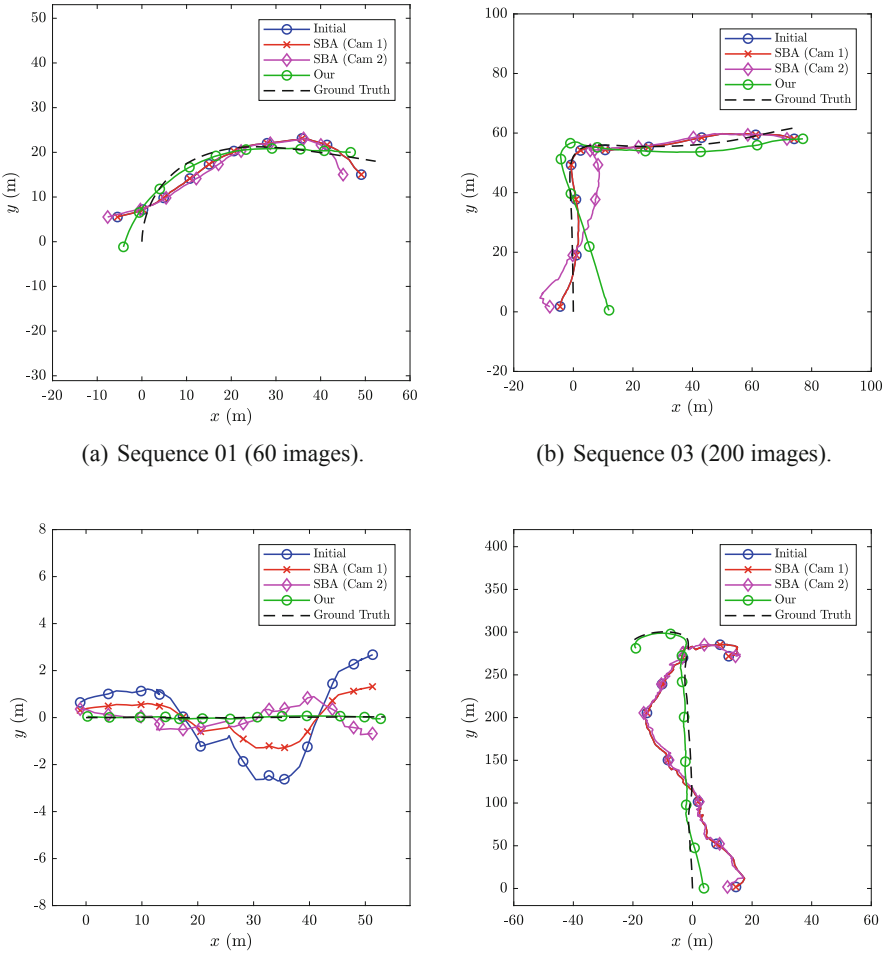
the general planar motion model, and as a substitute, we use the KITTI Visual Odometry/SLAM benchmark [9]. Since many sequences or subsequences depict urban environments with paved roads, the general planar motion model can roughly be applied. In case of clear violation of the general planar motion model, we proceed to use only subsequences where the model is applicable. As we are only interested by the road in front of the vehicle, and not the sky and other objects by the roadside, we proceed to crop a part of the image prior to estimating the homography. An example of this is shown in Fig. 5.

Fig. 5. Images from the KITTI Visual Odometry/SLAM benchmark, Sequence 01 (left) and 03 (right). Since the algorithm is homography-based the images are cropped *a priori* in order to contain a significant portion of planar or near planar surface. Such an assumption is not valid on all sequences of the dataset, however, certain cases, such as the highway of Sequence 01 (left) is a good candidate. There are several examples where occlusions occur, such as the car in Sequence 03 (right). These situations typically occur at crossroads and turns. Image credit: KITTI dataset [9].

We use SBA [16] to enforce the general 6-DoF model from the initial trajectory obtained using the traditional 4-point DLT solver, and from the same trajectory our proposed BA algorithm is used. The same thresholds for absolute and relative errors, termination control and damping factors are used for both methods. Furthermore, we do not match features between the stereo views, in order to demonstrate that enforcing the model is enough to increase the overall performance. The results are shown in Fig. 6.

In most cases it is favourable to impose the proposed method compared to the general 6-DoF method, using SBA. Furthermore, note that irregularities that are present in the initial trajectory is often transferred to the solutions obtained by SBA, thus producing physically improbable solutions. These irregularities are rarely seen using the proposed method, which results in smooth realistic trajectories under general conditions, regardless of whether the initial solution contains irregularities or not.

In fact, it is interesting to see what happens in cases where the general planar motion model is violated. Such an instance occurs in Fig. 6(b) depicting Sequence 03, and is due to the car approaching a crossroads, where a passing vehicle enters the field of view. The observed car, and the surroundings, are highly non-planar; one would, perhaps, expect such a clear violation to result in completely unreliable output, however, the only inconsistency in comparison to the ground truth, is that the resulting turn is too sharp, and the remaining path is consistent with the ground truth. This is not true for the general 6-DoF model, where several obvious inconsistencies are present.



(a) Sequence 01 (60 images).

(b) Sequence 03 (200 images).

(c) Sequence 04 (40 images).

(d) Sequence 06 (330 images).

Fig. 6. Estimated trajectories of subsequences of Sequence 01, 03, 04 and 06. In order to align the estimated paths with the ground truth, Procrustes analysis has been carried out. N.B. the different aspect ratio in (c), which is intentionally added in order to clearly visualise the difference. Figure reproduced from [30].

7 Conclusion

In this paper a novel bundle adjustment method has been devised, which enforces the general planar motion model. We provide an efficient implementation scheme that exploits the sparse structure of the Jacobian, and, additionally, avoids recomputing unnecessary quantities, making it highly attractive for real-time computations.

The performance of different polynomial solvers are studied, in terms of both accuracy and speed, taking the entire bundle adjustment framework into account. We dis-

cuss how enforcing different polynomial constraints, through planar motion compatible homography solvers, in an early part of the bundle adjustment framework affect the end results. Furthermore, we discuss which trade-offs between speed and accuracy that can be made to suit ones specific priorities.

The proposed method has been tested on real data and was compared to state-of-the-art methods for sparse bundle adjustment, for which it performs well, and gives physically accurate solutions, despite some model assumptions not being fulfilled.

Acknowledgements. This work has been funded by the Swedish Research Council through grant no. 2015-05639 'Visual SLAM based on Planar Homographies'.

References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building Rome in a day. Commun. ACM **54**(10), 105–112 (2011)
2. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006). https://doi.org/10.1007/11744023_32
3. Byröd, M., Åström, K.: Conjugate gradient bundle adjustment. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6312, pp. 114–127. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15552-9_9
4. Chen, T., Liu, Y.H.: A robust approach for structure from planar motion by stereo image sequences. Mach. Vis. Appl. (MVA) **17**(3), 197–209 (2006)
5. Davison, A.J.: Real-time simultaneous localisation and mapping with a single camera. In: International Conference on Computer Vision (ICCV), Nice, France, pp. 1403–1410, October 2003
6. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: real-time single camera SLAM. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **29**(6), 1052–1067 (2007)
7. Eriksson, A., Bastian, J., Chin, T., Isaksson, M.: A consensus-based framework for distributed bundle adjustment. In: Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 1754–1762, June 2016
8. Frahm, J.-M., et al.: Building Rome on a cloudless day. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 368–381. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_27
9. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, June 2012
10. Hajjdiab, H., Laganière, R.: Vision-based multi-robot simultaneous localization and mapping. In: Canadian Conference on Computer and Robot Vision (CRV), London, ON, Canada, pp. 155–162, May 2004
11. Hänsch, R., Drude, I., Hellwich, O.: Modern methods of bundle adjustment on the GPU. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS Congress), Prague, Czech Republic, pp. 43–50, July 2016
12. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
13. Konolige, K.: Sparse sparse bundle adjustment. In: British Machine Vision Conference (BMVC), Aberystwyth, Wales, pp. 102.1–11, August 2010

14. Liang, B., Pears, N.: Visual navigation using planar homographies. In: International Conference on Robotics and Automation (ICRA), Washington, DC, USA, pp. 205–210, May 2002
15. Lourakis, M.I.A., Argyros, A.A.: Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment? In: International Conference on Computer Vision (ICCV), Beijing, China, PRC, pp. 1526–1531, October 2005
16. Lourakis, M.I.A., Argyros, A.A.: SBA: a software package for generic sparse bundle adjustment. ACM Trans. Math. Softw. (TOMS) **36**(1), 2:1–2:30 (2009)
17. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. (IJCV) **60**(2), 91–110 (2004)
18. Ortín, D., Montiel, J.M.M.: Indoor robot motion based on monocular images. Robotica **19**(3), 331–342 (2001)
19. Pham, T.T., Chin, T.J., Yu, J., Suter, D.: The random cluster model for robust geometric fitting. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **36**(8), 1658–1671 (2014)
20. Scaramuzza, D.: 1-point-RANSAC structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. Int. J. Comput. Vis. (IJCV) **95**(1), 74–85 (2011)
21. Scaramuzza, D.: Performance evaluation of 1-point-RANSAC visual odometry. J. Field Robot. (JFR) **28**(5), 792–811 (2011)
22. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. Int. J. Comput. Vis. (IJCV) **80**(2), 189–210 (2008)
23. Sturm, P.: A historical survey of geometric computer vision. In: Real, P., Diaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W. (eds.) CAIP 2011. LNCS, vol. 6854, pp. 1–8. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23672-3_1
24. Szeliski, R.: Computer Vision: Applications and Algorithms. Springer, London (2011). https://doi.org/10.1007/978-1-84882-935-0
25. Torr, P.H.S., Zisserman, A.: MLESAC: a new robust estimator with application to estimating image geometry. Comput. Vis. Image Underst. (CVIU) **78**(1), 138–156 (2000)
26. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment — a modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) IWVA 1999. LNCS, vol. 1883, pp. 298–372. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44480-7_21
27. Valtonen Örnhag, M., Heyden, A.: Generalization of parameter recovery in binocular vision for a planar scene. In: International Conference on Pattern Recognition and Artificial Intelligence, Montréal, Canada, pp. 37–42, May 2018
28. Valtonen Örnhag, M., Heyden, A.: Relative pose estimation in binocular vision for a planar scene using inter-image homographies. In: International Conference on Pattern Recognition Applications and Methods (ICPRAM), Funchal, Madeira, Portugal, pp. 568–575, January 2018
29. Valtonen Örnhag, M.: Fast non-minimal solvers for planar motion compatible homographies. In: Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods (ICPRAM), pp. 40–51. SCITEPRESS, Prague, February 2019
30. Valtonen Örnhag, M., Wadenbäck, M.: Planar motion bundle adjustment. In: Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods (ICPRAM), pp. 24–31. SCITEPRESS, Prague, February 2019
31. Wadenbäck, M., Heyden, A.: Planar motion and hand-eye calibration using inter-image homographies from a planar scene. In: International Conference on Computer Vision Theory and Applications (VISAPP), Barcelona, Spain, pp. 164–168, February 2013
32. Wadenbäck, M., Heyden, A.: Ego-motion recovery and robust tilt estimation for planar motion using several homographies. In: International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal, pp. 635–639, January 2014

33. Wadenbäck, M., Åström, K., Heyden, A.: Recovering planar motion from homographies obtained using a 2.5-point solver for a polynomial system. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 2966–2970. IEEE-Institute of Electrical and Electronics Engineers Inc., September 2016. https://doi.org/10.1109/ICIP.2016.7532903
34. Wiles, C., Brady, M.: Closing the loop on multiple motions. In: Proceedings of the Fifth IEEE International Conference on Computer Vision (ICCV), pp. 308–313. IEEE Computer Society, Cambridge, June 1995
35. Wiles, C., Brady, M.: Ground plane motion camera models. In: Buxton, B., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 238–247. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61123-1_143
36. Wu, C., Agarwal, S., Curless, B., Seitz, S.M.: Multicore bundle adjustment. In: Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, pp. 3057–3064, June 2011
37. Zach, C.: Robust bundle adjustment revisited. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 772–787. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_50
38. Zhang, R., Zhu, S., Fang, T., Quan, L.: Distributed very large scale bundle adjustment by global camera consensus. In: International Conference on Computer Vision (ICCV), Venice, Italy, pp. 29–38, October 2017
39. Zienkiewicz, J., Davison, A.J.: Extrinsics autocalibration for dense planar visual odometry. J. Field Robot. (JFR) **32**(5), 803–825 (2015)