Planar Motion and Visual Odometry: Pose Estimation from Homographies

Wadenbäck, Mårten

2014

PLANAR MOTION AND VISUAL ODOMETRY

POSE ESTIMATION FROM HOMOGRAPHIES

MÅRTEN WADENBÄCK

LUND UNIVERSITY

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118
SE-221 00 Lund
Sweden

http://www.maths.lth.se/

Licentiate Theses in Mathematical Sciences 2014:2
ISSN 1404-028X

ISBN 978-91-7623-102-9 (printed)
ISBN 978-91-7623-103-6 (electronic)
LUTFMA-2038-2014

Printed in Sweden by MediaTryck, Lund 2014

Contents

Introduction

One of the long-standing aims in robotics research is the development of algorithms for autonomous navigation. A popular class of such algorithms are the ones concerned with so called *Simultaneous Localisation and Mapping* (SLAM), in which a mobile platform, equipped with an array of suitable sensors (laser scanners, cameras, odometers, sonar, …), explores and maps the surrounding environment while keeping track of its own location with respect to the map. If the navigation relies mainly on integrating local motion estimates from cameras, as is the case in this work, a more specific term that is used is *Visual Odometry* (VO).

In this thesis, we will consider how to estimate the local robot motion based only on information from a single camera. The problem of how to represent efficiently the map is not addressed, and we thus only deal with one part of the SLAM problem. The thesis contains the papers

A Planar Motion and Hand-Eye Calibration Using Inter-Image Homographies from a Planar Scene (Wadenbäck and Heyden, 2013),

B Ego-Motion Recovery and Robust Tilt Estimation for Planar Motion Using Several Homographies (Wadenbäck and Heyden, 2014b),

C Trajectory Estimation Using Relative Distances Extracted from Inter-Image Homographies (Wadenbäck and Heyden, 2014a).

Paper A introduces a method for estimating the pose and motion of a camera which undergoes planar motion. This method is based on an explicit parametrisation of the inter-image homographies and an iterative scheme for determining the pose and motion of the camera. Experiments on both real and synthetic data are used to evaluate the method.

Paper B extends the method in Paper A by using more than one homography in the estimation of the camera pose, and thereby improves the accuracy and greatly reduces the number of breakdown cases. The evaluation in this paper is done only on synthetic data.

The motion estimation approach in Paper A and Paper B relies on the pose to be estimated first, and if this estimate is inaccurate, the motion estimation suffers. Because of this, we wanted to derive a motion estimation method which is independent of the pose estimation. Paper C tries to address this issue by devising a method for estimating the travelled distance between two camera positions. We show how the travelled distance may be expressed in terms of the condition number of the inter-image homography. Some sensitivity analysis is conducted on synthetic data for this method.

1 Simultaneous Localisation and Mapping

Autonomous navigation for robots is an important concept which has attracted increasing interest over the years. The applications of mobile robots are numerous, and include (to name just a few) flexible assembly lines, robotic vacuum cleaners, logistics applications, search and rescue operations, and planetary exploration. A common framework that has proven successful for enabling autonomous navigation is *Simultaneous Localisation and Mapping* (SLAM), in which the robot makes use of various sensors to map the surrounding environment and at the same time position itself within this map. The map created in the process should mark notable objects and landmarks in a way which allows for reliable re-identification. The type of map that can be created is highly dependent on the kinds of sensors employed and on the environment being mapped, and can range from sparsely placed points to dense and detailed textured 3D models.

Much of the early work on SLAM was focused on sensors such as laser range finders and wheel encoders (odometers), but with improvements in digital cameras and computational power, more and more SLAM systems rely (at least partly) on cameras for navigation. Among the well-known implementations of camera-based SLAM is the pioneering work by Harris and Pike (1988), in which filtering and estimation techniques were used to estimate the camera position over a short image sequence. More recent successful approaches include the vSLAM system (Karlsson et al., 2005) and the MonoSLAM system (Davison et al., 2007), which both represent

probabilistic viewpoints based on *Extended Kalman Filters* (EKF).

If the robot motion is somehow constrained, this may be taken into account in order to decrease the uncertainty in the estimated position. For the application in this thesis, the camera motion is constrained to a plane parallel with a planar floor. In that respect, our work resembles other SLAM systems such as Liang and Pears (2002) and Hajjdiab and Laganière (2004), which also navigate using images of the floor.

2 Parameter Estimation

An ever-occurring problem in mathematics and its applications is the problem of estimating a set of parameters detailing some mathematical model. The field of estimation is immense, and we will only scratch the surface of estimation in this short overview.

In our formulation, we assume that $y \in \mathbb{R}$ depends on $\boldsymbol{x} \in \mathbb{R}^n$ as

$$y = f(\boldsymbol{x}; \boldsymbol{\beta}), \tag{1}$$

where f comes from some predetermined[1] class of functions specified by the unknown (but constant) parameter vector $\boldsymbol{\beta} \in \mathbb{R}^p$. The problem is to find, given some data $\{(\boldsymbol{x}_j, y_j)\}_{j=1}^N$, a parameter vector $\boldsymbol{\beta}$ which agrees well with the model (1) and the provided data.

Occasionally it is possible to find a parameter vector $\boldsymbol{\beta}$ for which $y_j = f(\boldsymbol{x}_j; \boldsymbol{\beta})$ for all $j = 1, \ldots, N$, but in most practical cases one has to tolerate some discrepancies $e_j = y_j - f(\boldsymbol{x}_j; \boldsymbol{\beta})$. In such cases, it is often useful to try to minimise (with regards to $\boldsymbol{\beta}$) some kind of *cost function*[2], which produces a scalar measure of the size of the errors e_j.

[1] In general, the class which f comes from is not predetermined, and the selection of it is one of the most important steps in the whole modelling process, but in this discussion we shall assume that this has already been done for us. It should be mentioned in passing that this class should be chosen flexible enough to be able to capture the behaviour we try to model, yet simple enough to enable the estimation of $\boldsymbol{\beta}$.

[2] Another common method, which we will not consider here, is the so called *Maximum Likelihood* (ML) estimation method. Instead of minimising a cost function, one tries to find parameters which maximise the likelihood of the obtained observations.

There are many cost functions to choose from, but one of the most popular ones is the sum of squared errors,

$$E_{LS}(\boldsymbol{\beta}) = \frac{1}{2}\sum_{j=1}^{N}(y_j - f(\boldsymbol{x}_j;\boldsymbol{\beta}))^2 = \frac{1}{2}\sum_{j=1}^{N} e_j^2, \tag{2}$$

which gives rise to a so called least squares (LS) problem.

One of the reasons for the popularity of the least squares approach is that if the function f is differentiable (considered as a function of $\boldsymbol{\beta}$), then E_{LS} also becomes differentiable. This makes it tractable for numerical optimisation methods such as the Gauß-Newton algorithm and the Levenberg-Marquardt method. For the details of these algorithms, see Hartley and Zisserman (2004, App. 6). Note, however, that (2) will be non-convex for most choices of f, and for non-convex cost functions there is no guarantee that the optimisation ends up at the global optimum.

2.1 Robust Estimation using RANSAC

With the cost function (2), samples with large errors are assigned a very large penalty, and the optimisation thus prefers to decrease a large error at the expense of increasing many of the smaller errors. For various reasons (incorrect data association, bad equipment, human error, …) it may happen that some (or even most) of the measurements $(\boldsymbol{x}_j, y_j)$ fit the model extremely poorly, no matter what parameters are chosen. As just noted, this means that those bad samples give rise to large error terms, which in turn means that they have an unduly large influence on the estimated $\boldsymbol{\beta}$. A better estimate of $\boldsymbol{\beta}$ might be obtained if one ignored those bad samples.

One way to address this issue was presented by Fischler and Bolles (1981). They proposed a framework termed *RANdom SAmple Consensus* (RANSAC), in which the data is partitioned into so called *inliers*, which agree with the model, and *outliers*, which do not agree with the model.

Their idea was to repeatedly fit the model to a small random subset of the data and count the number of apparent inliers for this choice of parameters. After repeating this a suitable number of times, one takes the partition with the most number of inliers found so far and fits the model

Input: Model $y = f(\boldsymbol{x}; \boldsymbol{\beta})$, data $\{(\boldsymbol{x}_j, y_j)\}_{j=1}^{N}$, threshold δ
Output: $\boldsymbol{\beta}$ fitted to the largest set of inliers
1: **for** $j = 1, \ldots, K$ **do**
2: Select a small (minimal) random subset of the data
3: Estimate $\boldsymbol{\beta}$ from the selected subset
4: Count the number of samples for which $|y - f(\boldsymbol{x}; \boldsymbol{\beta})| < \delta$
5: **end for**
6: Estimate $\boldsymbol{\beta}$ from the largest found set of inliers

Algorithm 1: The RANSAC framework is useful when estimating model parameters from noisy or corrupted data.

to the set of inliers for this partition. Intuitively, if a subset of the data is chosen which only contains true inliers, the other true inliers should also appear to be inliers. If, on the other hand, some of the selected samples are not true inliers, then only a small number of samples will by chance appear to be inliers. Algorithm 1 shows the general procedure.

To use RANSAC, it is necessary to somehow determine the threshold δ deciding if a sample is an inlier, as well as the number of iterations K to run. To select δ requires some knowledge of how large errors one should expect, and the number of iterations K depends on the size of the selected subset as well as the fraction of true inliers in the data.

3 The Pinhole Camera Model

In this section we shall briefly describe the classic pinhole perspective camera model. A much more detailed discussion and derivation may be found in Hartley and Zisserman (2004, Ch. 6). For notes on how to compensate for lens distortions, which we will skip here and henceforth simply assume to have been done, see Hartley and Zisserman (2004, Sec. 7.4).

Intuition about the geometrical situation may be drawn from the idealised physical model of image formation shown in Figure 1. Introduce an orthonormal coordinate system in which the focal point (called the *camera centre*) of the camera is at the origin, and in which the image sensor

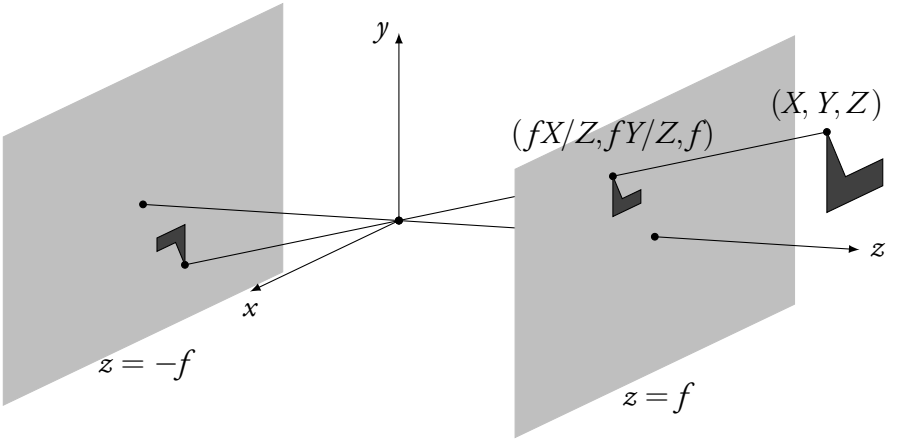

Figure 1: An idealised model of the image formation process. Light is emitted from the object and passes through the focal point, giving rise to an inverted image on the image plane $z = -f$.

lies in the plane $z = -f$. Suppose an object in front of the camera emits light, which passes through the focal point an falls onto the sensor, creating an inverted (horizontally as well as vertically flipped) image of the object. To mathematically undo the inverting is equivalent to moving the image sensor to the front of the camera, at $z = f$ (which we shall call the *image plane*). The line which is perpendicular to the image plane and passes through the camera centre (here the z-axis) is termed the *optical axis*, and its intersection with the image plane is called the *principal point*.

By considering similar triangles, it is seen that the scene point (X, Y, Z) is projected onto the image plane at $(fX/Z, fY/Z, f)$. Here it is clear that we may omit the third coordinate, and thus the camera induces a mapping from scene points (X, Y, Z) to image points, which may be written as

$$(X, Y, Z) \longmapsto (fX/Z, fY/Z). \tag{3}$$

By using *homogeneous coordinates*, where every scene point (X, Y, Z) is represented by the four-dimensional ray (WX, WY, WZ, W), and every image point (x, y) is represented by the three-dimensional ray (wx, wy, w), the

camera mapping (3) may conveniently be expressed using matrices as

$$\begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{4}$$

In some cases, it is more natural to work with pixel coordinates instead of the abstract image coordinates obtained by the mapping (3). Changing to pixel coordinates means scaling the x-coordinate with a factor d_x and the y-coordinate with a factor d_y (typically $d_x \approx d_y$ since pixels often are almost square), and moving the origin to one of the corners (usually the upper left corner). Defining $f_x = d_x f$ and $f_y = d_y f$, the mapping to pixels is given by

$$\begin{cases} x = f_x X/Z + c_x \\ y = f_y Y/Z + c_y \end{cases}, \tag{5}$$

where (c_x, c_y) are the pixel coordinates of the principal point. The camera mapping to pixel coordinates becomes[3]

$$\begin{bmatrix} xZ \\ yZ \\ Z \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{6}$$

We often want to work with a global coordinate system which is not necessarily aligned with the camera coordinate frame. Supposing the camera has coordinates $\boldsymbol{t} = (t_x, t_y, t_z)$ in this global coordinate frame and is rotated by the rotation matrix $\boldsymbol{R}$, the mapping from global coordinates to pixels will be given by

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \boldsymbol{K}\boldsymbol{R} \left[\boldsymbol{I} \mid -\boldsymbol{t} \right] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{7}$$

[3]Sometimes it is also necessary to introduce a *skew parameter*, accounting for non-rectangular pixels. We do not model this here.

Here the *camera calibration matrix* $\boldsymbol{K}$ contains the *intrinsic parameters* of the camera (the *principal point* (c_x, c_y) and the *focal lengths* f_x and f_y),

$$\boldsymbol{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

The matrix

$$\boldsymbol{P} = \boldsymbol{K}\boldsymbol{R}\left[\boldsymbol{I} \mid -\boldsymbol{t}\right] \tag{9}$$

is often called the *camera projection matrix*.

4 Homographies

The concept of a *homography* is a central theme throughout this thesis. For our purposes,[4] a homography may be defined as a bijective linear transformation on homogeneous coordinates (in the plane). This means that every homography may be represented by a non-singular 3×3-matrix $\boldsymbol{H}$, which is determined up to scale. To have a unique representation, one may require that $\|\boldsymbol{H}\| = 1$ (for some matrix norm).

One important property of the pinhole camera model is that if the scene points all lie in a plane[5] (as they do in the application described in this thesis), the coordinate transformation from one image to another will be given by a homography. With exception for certain degenerate cases, the homography is uniquely determined if one knows how it transforms four points.

The papers included in this thesis all describe algorithms which use homographies as input. It is briefly mentioned in the papers that these homographies can be obtained from the images, but the procedure is not explained. Here we will try to convey the idea behind the basic method of homography estimation.

[4] Homographies may be defined for arbitrary *projective spaces*. Projective spaces is a fascinating subject which unfortunately is outside the scope of this thesis, but the interested reader should definitely look up Busemann and Kelly (1953).

[5] This plane may not contain the camera centre.

4.1 Homography Estimation

In this section we will assume that we have a number of point correspondences $\boldsymbol{x}_j \leftrightarrow \hat{\boldsymbol{x}}_j$, where $\boldsymbol{x}_j = (x_j, y_j, 1)$ and $\hat{\boldsymbol{x}}_j = (\hat{x}_j, \hat{y}_j, 1)$ are homogeneous coordinates measured in each of the images. Such point correspondences may be found either by manually marking corresponding points in each image, or by automatically associating feature points found by methods such as SIFT (Lowe, 2004) or other similar approaches.

If the correspondences are found automatically, there are potentially many false associations, and it will then be necessary to use a robust framework such as RANSAC (see Section 2.1). It is common to use the *Direct Linear Transformation* (DLT) together with RANSAC, and then at a final stage use the obtained homography as an initial solution to minimise the *geometric error* (Section 4.1.2) for the obtained set of inliers. The reason for doing it in two steps is that the minimisation of the geometric error is a very complicated problem which cannot be solved without a good initial solution.

4.1.1 Direct Linear Transformation (DLT)

The Direct Linear Transformation works by setting up and solving a linear system of equations for the elements of $\boldsymbol{H}$. The aim is to find a 3×3-matrix $\boldsymbol{H}$ such that

$$\hat{w}_j \hat{\boldsymbol{x}}_j = \boldsymbol{H}\boldsymbol{x}_j, \quad j = 1, \ldots, N, \tag{10}$$

for some arbitrary scalars $\hat{w}_j$, or equivalently,

$$\hat{\boldsymbol{x}}_j \times \boldsymbol{H}\boldsymbol{x}_j = \boldsymbol{0}, \quad j = 1, \ldots, N. \tag{11}$$

If we let $\boldsymbol{h}_k^T$ denote row k in $\boldsymbol{H}$, then (11) may be written as

$$\boldsymbol{0} = \hat{\boldsymbol{x}}_j \times \boldsymbol{H}\boldsymbol{x}_j = \begin{bmatrix} \hat{y}_j \boldsymbol{h}_3^T \boldsymbol{x}_j - \boldsymbol{h}_2^T \boldsymbol{x}_j \\ \boldsymbol{h}_1^T \boldsymbol{x}_j - \hat{x} \boldsymbol{h}_3^T \boldsymbol{x}_j \\ \hat{x} \boldsymbol{h}_2^T \boldsymbol{x}_j - \hat{y} \boldsymbol{h}_1^T \boldsymbol{x}_j \end{bmatrix} = \begin{bmatrix} \hat{y}_j \boldsymbol{x}_j^T \boldsymbol{h}_3 - \boldsymbol{x}_j^T \boldsymbol{h}_2 \\ \boldsymbol{x}_j^T \boldsymbol{h}_1 - \hat{x} \boldsymbol{x}_j^T \boldsymbol{h}_3 \\ \hat{x} \boldsymbol{x}_j^T \boldsymbol{h}_2 - \hat{y} \boldsymbol{x}_j^T \boldsymbol{h}_1 \end{bmatrix}, \tag{12}$$

which may be turned into a system of linear equations with three equations and nine unknowns,

$$\underbrace{\begin{bmatrix} \mathbf{0} & -\boldsymbol{x}_j^T & \hat{y}_j \boldsymbol{x}_j^T \\ \boldsymbol{x}_j^T & \mathbf{0} & -\hat{x}\boldsymbol{x}_j^T \\ -\hat{y}\boldsymbol{x}_j^T & \hat{x}\boldsymbol{x}_j^T & \mathbf{0} \end{bmatrix}}_{=\boldsymbol{M}_j} \begin{bmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \\ \boldsymbol{h}_3 \end{bmatrix} = \mathbf{0}. \tag{13}$$

By stacking the $\boldsymbol{M}_j$ into a $3N \times 9$-matrix[6] $\boldsymbol{M}$, the homography $\boldsymbol{H}$ may be found as the null space of $\boldsymbol{M}$. The null space of $\boldsymbol{M}$ may be found numerically using the *Singular Value Decomposition* (SVD).

4.1.2 Geometric Error

An objection which may rightly be raised against the DLT estimate described in Section 4.1.1 is that it does not directly relate to actual distances as measured in the images, but rather to some abstract algebraic measure of fitness of the model.

One way to refine the homography estimate obtained via DLT is to determine a homography $\boldsymbol{H}$ as well as a number of *geometric corrections* to the points, and minimise these geometric corrections. More precisely, we want to find corrections $(\Delta x_j, \Delta y_j)$ and $(\Delta \hat{x}_j, \Delta \hat{y}_j)$ along with a homography $\boldsymbol{H}$ which maps $(x_j + \Delta x_j, y_j + \Delta y_j)$ *exactly* to $(\hat{x}_j + \Delta \hat{x}_j, \hat{y}_j + \Delta \hat{y}_j)$, minimising

$$E = \sum_{j=1}^{N} \Delta x_j^2 + \Delta y_j^2 + \Delta \hat{x}_j^2 + \Delta \hat{y}_j^2. \tag{14}$$

The minimisation of (14) is not a convex problem, and the number of variables is typically quite large. Approaching this problem without a reasonable initial solution, such as one found using the DLT approach, is not practicable.

[6] The rows in $\boldsymbol{M}_j$ are linearly dependent, and it is actually sufficient to only use two rows from each $\boldsymbol{M}_j$. This reduces the size of $\boldsymbol{M}$ to $2N \times 9$, and gives a performance gain.

5 Closing and Future Work

The SLAM problem consists of a mapping component as well as a component for estimating location and motion, and it has been demonstrated that both parts are important for achieving accurate navigation over longer distances. The papers in this thesis focus on the motion estimation part, and the approach taken is to estimate motion parameters from homographies, which in turn have to be estimated from images of the floor. While this approach works well locally, it exhibits notable error accumulation over longer distances.

In the future, one natural way to improve performance over longer distances would be to also consider the map building. This could be formulated as a large estimation problem where parameters describing the map (such as the landmark positions) are estimated along with the motion parameters. This offers many interesting challenges, such as how best to represent the map and how to handle the growing parameter space.

Another interesting direction for future investigations would be to incorporate other sensors into the motion estimation. The research area concerned with combining sensor data in order to improve estimation accuracy is called *Sensor Fusion*, and this has become a very active area of research as of late.

Bibliography

H. Busemann and P. J. Kelly. *Projective Geometry and Projective Metrics*, volume 3 of *Pure and Applied Mathematics*. Academic Press, Mineola, NY, USA, 1953. ISBN 0486445828.

A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

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*, 24(6):381–395, 1981.

H. Hajjdiab and R. Laganière. Vision-Based Multi-Robot Simultaneous Localization and Mapping. In *CRV '04: Proceedings of the 1st Canadian Conference on Computer and Robot Vision*, pages 155–162, Washington, DC, USA, 2004. IEEE Computer Society.

C. G. Harris and J. M. Pike. 3D Positional Integration from Image Sequences. *Image and Vision Computing*, 6(2):87–90, 1988.

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, England, United Kingdom, Second edition, 2004. ISBN 0521540518.

N. Karlsson, E. D. Bernardo, J. P. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich. The vSLAM Algorithm for Robust Localization and Mapping. In *ICRA '05: Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 24–29, Barcelona, Spain, 2005. IEEE.

B. Liang and N. Pears. Visual Navigation using Planar Homographies. In *ICRA '02: Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pages 205–210, Washington, DC, USA, 2002.

D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004. ISSN 0920-5691.

M. Wadenbäck and A. Heyden. Planar Motion and Hand-Eye Calibration Using Inter-Image Homographies from a Planar Scene. In *Proceedings of VISIGRAPP 2013*, pages 164–168, Barcelona, Spain, February 2013. SCITEPRESS.

M. Wadenbäck and A. Heyden. Trajectory Estimation Using Relative Distances Extracted from Inter-Image Homographies. In *CRV '14: Proceedings of the 11th Canadian Conference on Computer and Robot Vision*, pages 232–237, Montréal, QC, Canada, May 2014a. IEEE Computer Society.

M. Wadenbäck and A. Heyden. Ego-Motion Recovery and Robust Tilt Estimation for Planar Motion Using Several Homographies. In *Proceedings of VISIGRAPP 2014*, pages 635–639, Lisbon, Portugal, January 2014b. SCITEPRESS.

Planar Motion and Hand-Eye Calibration Using Inter-Image Homographies from a Planar Scene

Mårten Wadenbäck and Anders Heyden
Centre for Mathematical Sciences, Lund University

Abstract: In this paper we consider a mobile platform performing partial hand-eye calibration and Simultaneous Localisation and Mapping (SLAM) using images of the floor along with the assumptions of planar motion and constant internal camera parameters. The method used is based on a direct parametrisation of the camera motion, combined with an iterative scheme for determining the motion parameters from inter-image homographies. Experiments are carried out on both real and synthetic data. For the real data, the estimates obtained are compared to measurements by an industrial robot, which serve as ground truth. The results demonstrate that our method produces consistent estimates of the camera position and orientation. We also make some remarks about patterns of motion for which the method fails.

1 Introduction

The development of algorithms for Simultaneous Localisation and Mapping (SLAM) has been a major focus in robotics research the past few decades. Such algorithms aim at enabling a mobile platform to explore and map its surroundings, while at the same time maintaining accurate knowledge of its position. Many types of sensors may be used to this end, and are often combined to supplement each other.

For the mapping part of SLAM, a reconstruction (broadly interpreted) must be created from the scene. For some work on SLAM using visual sensors, see for example Davison (2003), Karlsson et al. (2005) and Koch et al. (2010). Scene reconstruction from images is a well studied problem in computer vision, and is still a very active research area. Since the introduction of the fundamental matrix in Faugeras (1992) and Hartley (1992), epipolar geometry has been the foundation of many successful approaches to visual reconstruction.

However, a planar or near-planar scene is well known to be a degenerate or ill-conditioned case for reconstruction based on the fundamental matrix and similar approaches. Since planar scenes and objects are very common in man-made or indoor environments, a navigation system intended to operate in such environments must take care to avoid degeneracy. Planar homographies, on the other hand, are particularly well suited to planar scenes, but are unable to describe general 3D structure. This insight has been utilised for visual navigation in Liang and Pears (2002) and Hajjdiab and Laganière (2004), among others.

In this paper we shall consider a single camera, with square pixels and zero skew, moving at a constant height above the floor. We will further assume that the internal parameters of the camera are constant, and that the camera orientation is fixed except for a rotation about the normal to the floor plane. Using inter-image homographies not only avoids the degeneracy issue mentioned above, but in addition allows us to use an explicit parametrisation of this particular kind of camera motion.

In the case where the inter-image homographies describe a Euclidean (or, in general, an affine) transformation, the motion parameters are easily recovered using the QR decomposition. This happens when the image plane is parallel to the floor. In the presence of a tilt, however, it is not as straightforward to extract the motion information from the homographies. The main contribution of this paper is a method to compute both the tilt and the motion information from a single homography.

Another reason for estimating the tilt is that only rectified images may be stitched consistently into a mosaic. A visual navigation system based on a sparse feature based map of the floor plane also needs rectified images to construct the map. It is in general not trivial to mount a camera with very high precision, so avoiding the need for this would be useful.

Determining the tilt can also be seen as a partial hand-eye calibration. The original formulation of the hand-eye calibration problem was to recover the relative orientation between a robot arm and a camera mounted on the arm. Tsai and Lenz showed that with known 3D feature points, known motion of the robot arm, and known transformations $\boldsymbol{A}$ and $\boldsymbol{B}$, the unknown relative orientation $\boldsymbol{X}$ can be determined from the equation

$\boldsymbol{AX} = \boldsymbol{XB}$ (Tsai and Lenz, 1989). The problem was later reformulated using quaternions to parametrise rotations and 3×4 camera matrices instead of classical transformation matrices (Horaud and Dornaika, 1995).

2 Camera Parametrisation

We assume that the camera is mounted rigidly onto a mobile platform, and directed towards the floor. This means that the position and orientation of the camera can be parametrised by a translation vector $\boldsymbol{t} = (t_x, t_y, t_z)$ and a rotation in the floor plane of an angle φ. The tilt is described by the constant angles ψ and θ. Both the translation and the three angles will be estimated. The camera is assumed to move in the plane $z = 0$, and the ground plane is taken to be at $z = 1$. This is not a restriction, since it only reflects our choice of the world coordinate system and global scale fixation (corresponding to the unknown focal length).

We will consider two consecutive images, A and B, with associated camera matrices

$$\begin{aligned} \boldsymbol{P}_A &= \boldsymbol{R}_{\psi\theta}[\boldsymbol{I} \mid \boldsymbol{0}], \\ \boldsymbol{P}_B &= \boldsymbol{R}_{\psi\theta}\boldsymbol{R}_{\varphi}[\boldsymbol{I} \mid -\boldsymbol{t}]. \end{aligned} \tag{1}$$

Here $\boldsymbol{R}_{\psi\theta}$ is a rotation of θ around the y-axis followed by a rotation of ψ around the x-axis, and $\boldsymbol{R}_{\varphi}$ is a rotation of φ around the z-axis (the floor normal).

Using (1), one can easily verify that the homography $\boldsymbol{H}$ from A to B is

$$\boldsymbol{H} = \lambda \boldsymbol{R}_{\psi\theta}\boldsymbol{R}_{\varphi}\boldsymbol{T}\boldsymbol{R}_{\psi\theta}^{T}, \tag{2}$$

for any non-zero $\lambda \in \mathbb{R}$ and with

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

3 Tilt Estimation

The presence of a tilt gives rise to perspective effects. These distort the geometry perceived by the camera, and prevent easy extraction of motion

information. If the tilt angles ψ and θ can be determined, one can rectify the image and then use the QR decomposition to retrieve the translation $\boldsymbol{t}$ and the free rotation φ.

To estimate ψ and θ, we derive equations that contain these angles but which do not contain $\boldsymbol{t}$ and φ. These equations will then be solved using an iterative scheme.

3.1 Eliminating φ

Separating the tilt angles ψ and θ from the motion parameters $\boldsymbol{t}$ and φ in (2), we get

$$\boldsymbol{R}_{\psi\theta}^T \boldsymbol{H} \boldsymbol{R}_{\psi\theta} = \lambda \boldsymbol{R}_\varphi \boldsymbol{T}. \tag{4}$$

Here, one notes that $\boldsymbol{R}_\varphi$ can be eliminated by multiplying with the transpose from the left on both sides. This results in the relation

$$\boldsymbol{R}_{\psi\theta}^T \boldsymbol{M} \boldsymbol{R}_{\psi\theta} = \lambda^2 \boldsymbol{T}^T \boldsymbol{T}, \tag{5}$$

with (symmetric)

$$\boldsymbol{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{12} & m_{22} & m_{23} \\ m_{13} & m_{23} & m_{33} \end{bmatrix} = \boldsymbol{H}^T \boldsymbol{H}. \tag{6}$$

Since both sides of (5) are symmetric matrices, one obtains six unique equations. Let $\mathcal{L} = \boldsymbol{R}_{\psi\theta}^T \boldsymbol{M} \boldsymbol{R}_{\psi\theta}$ and $\mathcal{R} = \lambda^2 \boldsymbol{T}^T \boldsymbol{T}$ be the left and right hand sides of (5), respectively. Evaluating $\mathcal{R}$, one obtains

$$\mathcal{R} = \lambda^2 \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ -t_x & -t_y & 1 + t_x^2 + t_y^2 \end{bmatrix}. \tag{7}$$

3.2 Iterative Scheme

As described in Section 2, $\boldsymbol{R}_{\psi\theta} = \boldsymbol{R}_\psi \boldsymbol{R}_\theta$ is a rotation of θ around the y-axis followed by a rotation of ψ around the x-axis. Direct multiplication of the rotation matrices allows us to evaluate $\mathcal{L}$ (though this margin is too narrow

Input: An inter-image homography $\boldsymbol{H}$
Output: An approximation $\boldsymbol{R}$ of $\boldsymbol{R}_{\psi\theta}$
1: $\widehat{\boldsymbol{M}} \leftarrow \boldsymbol{H}^T\boldsymbol{H}$
2: $\theta_0 \leftarrow 0$
3: **for** $j = 1, \ldots, N$ **do**
4: $\quad \widehat{\boldsymbol{M}} \leftarrow \boldsymbol{R}_{\theta_{j-1}}^T \widehat{\boldsymbol{M}} \boldsymbol{R}_{\theta_{j-1}}$
5: $\quad$ Solve for ψ_j
6: $\quad \widehat{\boldsymbol{M}} \leftarrow \boldsymbol{R}_{\psi_j}^T \widehat{\boldsymbol{M}} \boldsymbol{R}_{\psi_j}$
7: $\quad$ Solve for θ_j
8: **end for**
9: $\boldsymbol{R} \leftarrow \boldsymbol{R}_{\theta_0}\boldsymbol{R}_{\psi_1}\boldsymbol{R}_{\theta_1}\boldsymbol{R}_{\psi_2}\boldsymbol{R}_{\theta_2}\cdots\boldsymbol{R}_{\psi_N}\boldsymbol{R}_{\theta_N}$

Algorithm 2: Iteratively approximate $\boldsymbol{R}_{\psi\theta}$. The steps on line 5 and line 7 are detailed in Sections 3.2.1 and 3.2.2. Since the current approximation is absorbed into $\widehat{\boldsymbol{M}}$, we may assume that the fixed angle is zero when solving for the free one.

to contain the result), and one finds that $\mathcal{L}$ is a fourth degree expression in $c_\psi = \cos\psi$, $s_\psi = \sin\psi$, $c_\theta = \cos\theta$ and $s_\theta = \sin\theta$.

Noting that $\mathcal{R}_{11}$, $\mathcal{R}_{12}$ and $\mathcal{R}_{22}$ are independent of $\boldsymbol{t}$, the equations for ψ and θ become

$$\begin{cases} \mathcal{L}_{11} - \mathcal{L}_{22} = 0 \\ \mathcal{L}_{12} = 0 \end{cases}. \tag{8}$$

But instead of trying to solve (8) for both ψ and θ at the same time, we will iteratively alternate between solving for one angle, with the other held fixed. This reduces the problem of solving a fourth degree trigonometric equation, so that we instead iterate and solve a second degree equation in each iteration.

Before explaining in detail how these equations are solved, we first outline in Algorithm 2 the iterative scheme which produces an approximation

to $\boldsymbol{R}_{\psi\theta}$. Since

$$\boldsymbol{R}_{\psi\theta} = \boldsymbol{R}_{\psi}\boldsymbol{R}_{\theta} = \begin{bmatrix} c_\theta & 0 & s_\theta \\ s_\psi s_\theta & c_\psi & -s_\psi c_\theta \\ -c_\psi s_\theta & s_\psi & c_\psi c_\theta \end{bmatrix} \tag{9}$$

it is trivial to find ψ and θ from this approximation.

3.2.1 Solving for ψ

Since c_ψ and s_ψ cannot both be zero, (8) is equivalent to

$$\begin{cases} \mathcal{L}_{11} - \mathcal{L}_{22} = 0 \\ c_\psi \mathcal{L}_{12} = 0 \\ s_\psi \mathcal{L}_{12} = 0 \end{cases}. \tag{10}$$

By letting $\widehat{\boldsymbol{M}} = \boldsymbol{R}_\theta^T \boldsymbol{M} \boldsymbol{R}_\theta$ this can be written in matrix form as

$$\begin{bmatrix} \widehat{m}_{11} - \widehat{m}_{22} & -2\widehat{m}_{23} & \widehat{m}_{11} - \widehat{m}_{33} \\ \widehat{m}_{12} & \widehat{m}_{13} & 0 \\ 0 & \widehat{m}_{12} & \widehat{m}_{13} \end{bmatrix} \begin{bmatrix} c_\psi^2 \\ c_\psi s_\psi \\ s_\psi^2 \end{bmatrix} = 0. \tag{11}$$

This means that $(c_\psi^2, c_\psi s_\psi, s_\psi^2)$ lies in the null space of the coefficient matrix in (11). Unless $\widehat{m}_{12} = \widehat{m}_{13} = 0$, the rank of the coefficient matrix in (11) is clearly at least two. For this reason, we should expect a one dimensional null space. Due to measurement noise this will not be the case, so instead we use the singular vector $\boldsymbol{v} = (v_1, v_2, v_3)$ corresponding to the smallest singular value as our null vector.

Provided the singular vector $\boldsymbol{v}$ one obtains ψ as

$$\psi = \frac{1}{2} \arcsin \frac{2v_2}{v_1 + v_3}. \tag{12}$$

3.2.2 Solving for θ

Now θ can be found in much a similar way as ψ. Physical considerations imply that, at least for moderately sized angles, $\boldsymbol{R}_\psi \boldsymbol{R}_\theta$ has approximately the

same effect on the camera as $\boldsymbol{R}_\theta \boldsymbol{R}_\psi$. Examination of the matrices confirms this for small angles.

Therefore, if $\widehat{\boldsymbol{M}} = \boldsymbol{R}_\psi^T \boldsymbol{M} \boldsymbol{R}_\psi$, then

$$\begin{cases} \mathcal{L}_{11} - \mathcal{L}_{22} = 0 \\ c_\theta \mathcal{L}_{12} = 0 \\ s_\theta \mathcal{L}_{12} = 0 \end{cases}, \tag{13}$$

can be written in matrix form as

$$\begin{bmatrix} \widehat{m}_{11} - \widehat{m}_{22} & -2\widehat{m}_{13} & \widehat{m}_{33} - \widehat{m}_{22} \\ \widehat{m}_{12} & -\widehat{m}_{23} & 0 \\ 0 & \widehat{m}_{12} & -\widehat{m}_{23} \end{bmatrix} \begin{bmatrix} c_\theta^2 \\ c_\theta s_\theta \\ s_\theta^2 \end{bmatrix} = 0. \tag{14}$$

We find, in the same way as in Section 3.2.1 that the null vector $\boldsymbol{v}$ can be used to find

$$\theta = \frac{1}{2} \arcsin \frac{2v_2}{v_1 + v_3}. \tag{15}$$

4 Experiments

In order to test how well the tilt estimation works in practice, fifty homographies of the form (2) were generated with random values for ψ, θ, φ and $\boldsymbol{t}$. The true angles and their corresponding estimates can be seen in Figure 1.

4.1 Path Reconstruction

A simple path estimation has also been tried on both synthetic and real data using the QR decomposition to determine translation and planar rotation, after estimating the tilt as described in Section 3. In the simulation, noise of a magnitude corresponding to a few pixels have been added to the points used to estimate the homographies. Results for this experiment are shown in Figure 2 and Figure 3.

We have also carried out experiments with real data. A camera mounted onto an industrial robot has been used to take images, from which

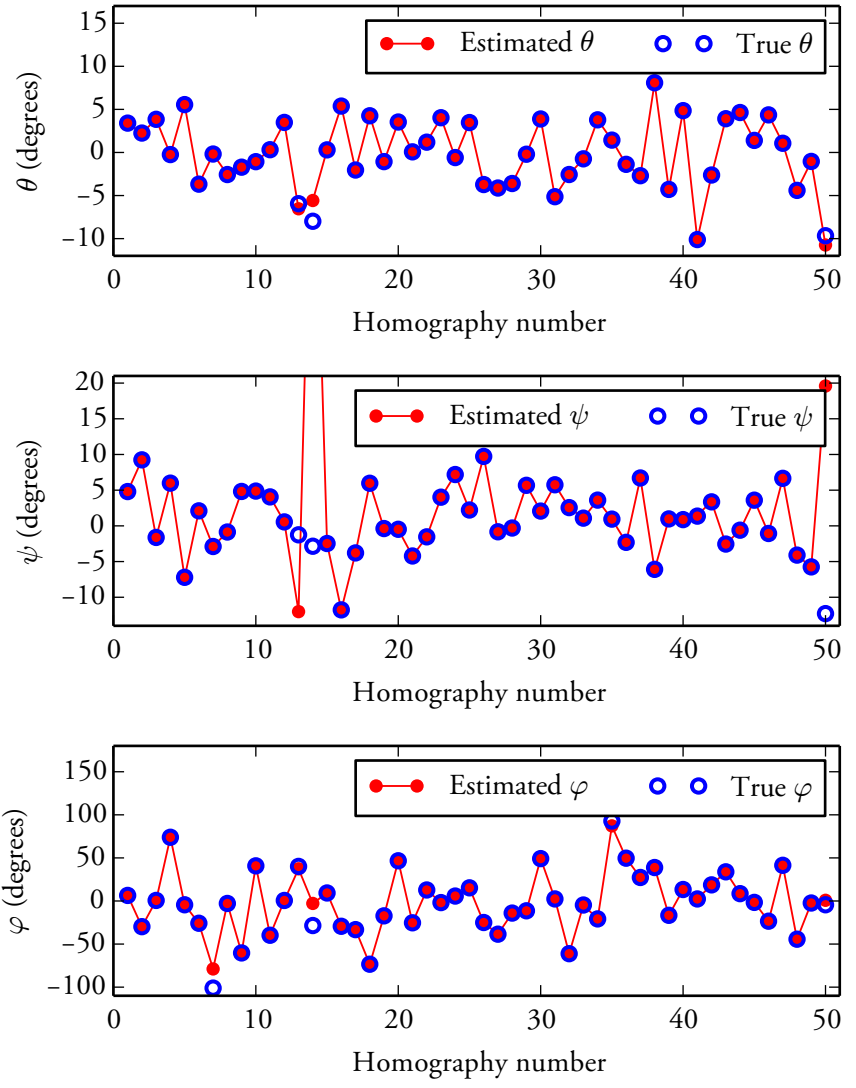

Figure 1: True and estimated values for ψ, θ and φ for fifty randomly generated homographies. As can be seen, the estimation works well in most instances.

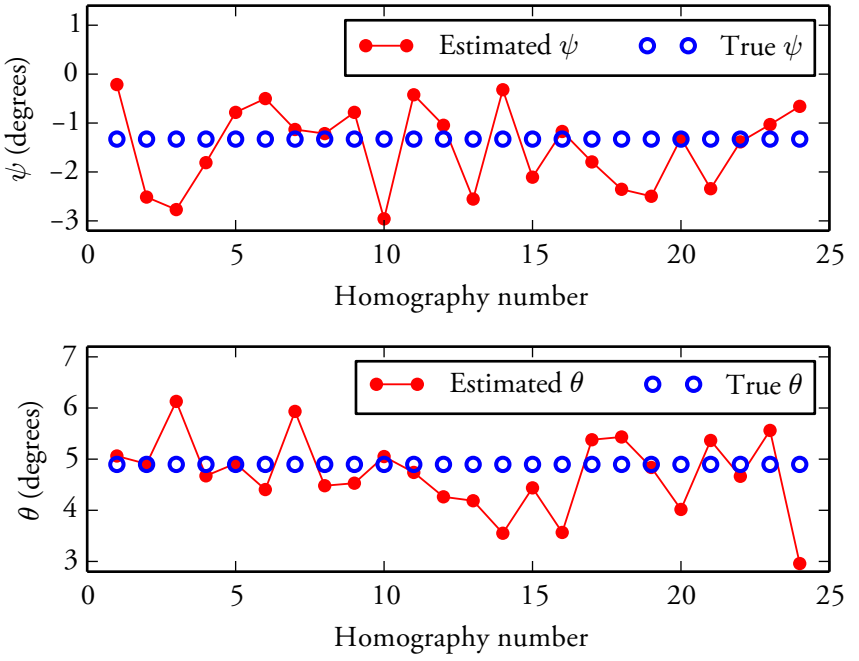

Figure 2: We see that the estimated values of ψ and θ are, on average, close to the true values. Since ψ and θ are constant, temporal filtering could be used to get better estimates over time.

homographies were computed. The resulting reconstruction can be seen in Figure 4. For comparison, we have additionally estimated the non constant angle φ using a method based on conjugate rotations, see Liang and Pears (2002) for details. This method computes φ from the eigenvalues of the homography without estimating the tilt. Figure 5 shows this estimate compared to our estimate and the true value (as measured by the robot). Both methods perform well, however some statistical measures shown in Table 1 suggest that tilt estimation followed by QR decomposition has a slightly favourable performance.

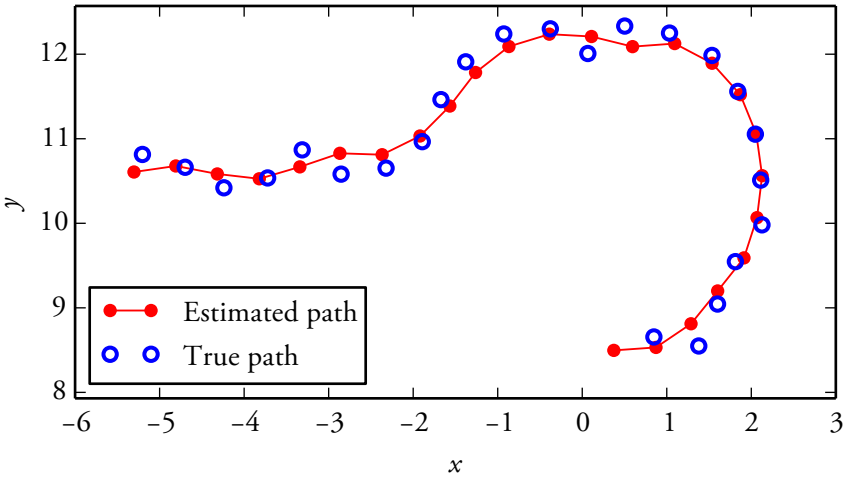

Figure 3: The simulated and the estimated paths. Procrustes analysis has been carried out to align the path curves for easy comparison.

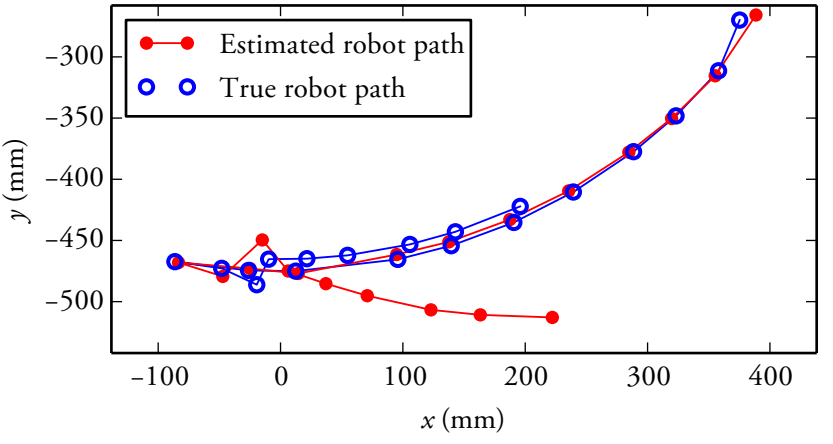

Figure 4: True and estimated paths for the robot experiment. At the lower part of the plot some erroneous estimates are made, which results in the estimated path being deflected away.

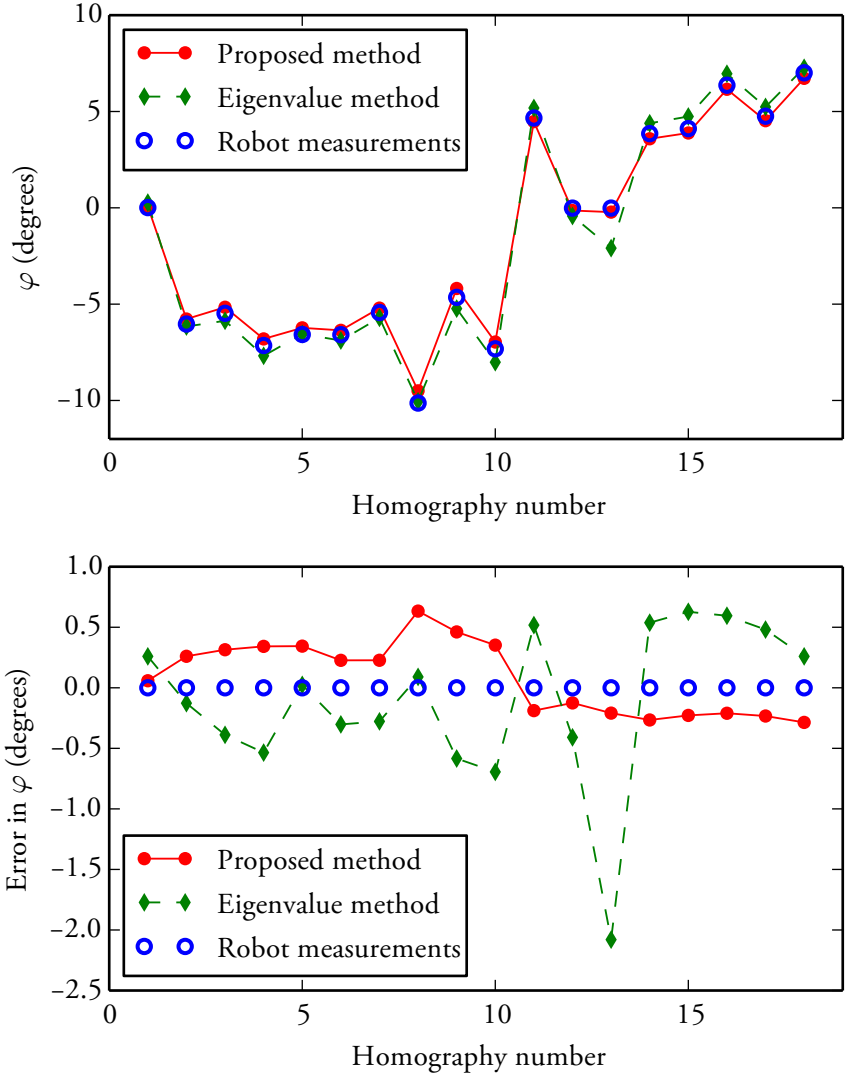

Figure 5: The upper plot shows the difference in orientation between consecutive images, and the lower plot shows the angular error. The plots show that our estimates of φ and the eigenvalue-based estimates of φ are both close to the truth (robot measurements).

Table 1: Mean, median and variance of the magnitude of the angular error. For the eigenvalue-based method, the thirteenth measurement is considered an outlier and has been omitted. Despite this, the proposed method is clearly seen to give more accurate estimates.

	Mean	Median	Variance
Proposed (QR)	0.2759	0.2467	0.0161
Eigenvalue	0.3947	0.4092	0.0403

4.2 Ill-Conditioned Motion

Empirical evidence suggests that the instances where the tilt estimation fails are the ones where the translation $\boldsymbol{t}$ is close to either a pure x-translation or a pure y-translation. Randomly generating homographies with this pattern of motion provides further evidence for this. It can further be seen that a pure x-translation gives rise to a poor estimate of ψ, while a pure y-translation results in a poor estimate of θ. Results for this experiment are presented in Figure 6 and Figure 7. Theoretical understanding of this will be necessary if the instability is to be addressed.

5 Conclusion

Tilt estimation is a prerequisite for constructing consistent floor maps using images from a tilted camera. In this paper we have presented an iterative scheme for determining the tilt from a single homography. Experiments with a simple path reconstruction have been conducted, which show that if the tilt is rectified then the correct Euclidean motion can be found using the QR decomposition. Experiments using synthetic data show that the estimated tilt angles are close to the true tilt angles in most instances, however some especially troublesome motions have been found.

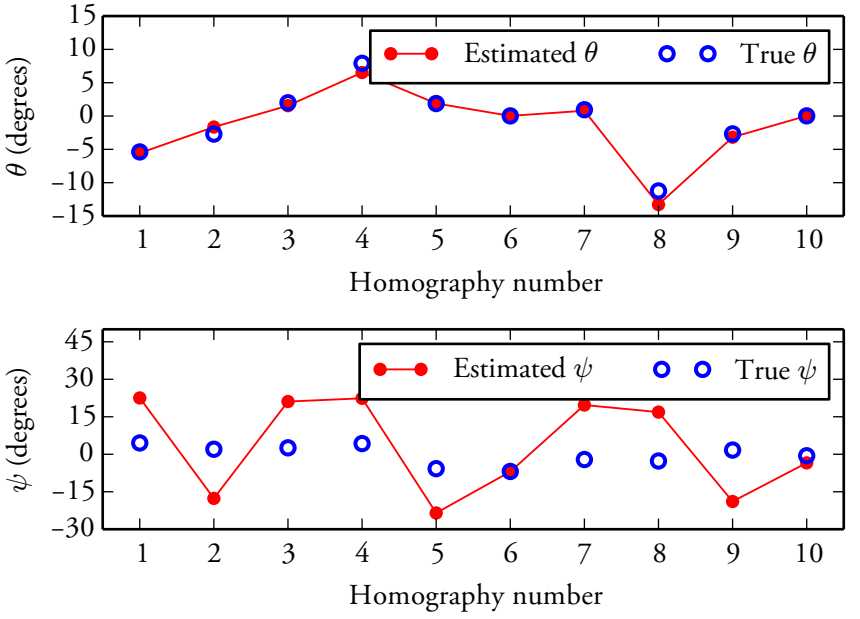

Figure 6: When $\boldsymbol{t}$ is a pure x-translation, ψ seems to be unreliably estimated.

Acknowledgements

This work has been funded by the Swedish Foundation for Strategic Research through the SSF project ENGROSS (web page at `www.engross.lth.se`).

Bibliography

A. J. Davison. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, volume 2 of *ICCV '03*, pages 1403–1410, Nice, France, 2003. IEEE Computer Society. ISBN 0-7695-1950-4.

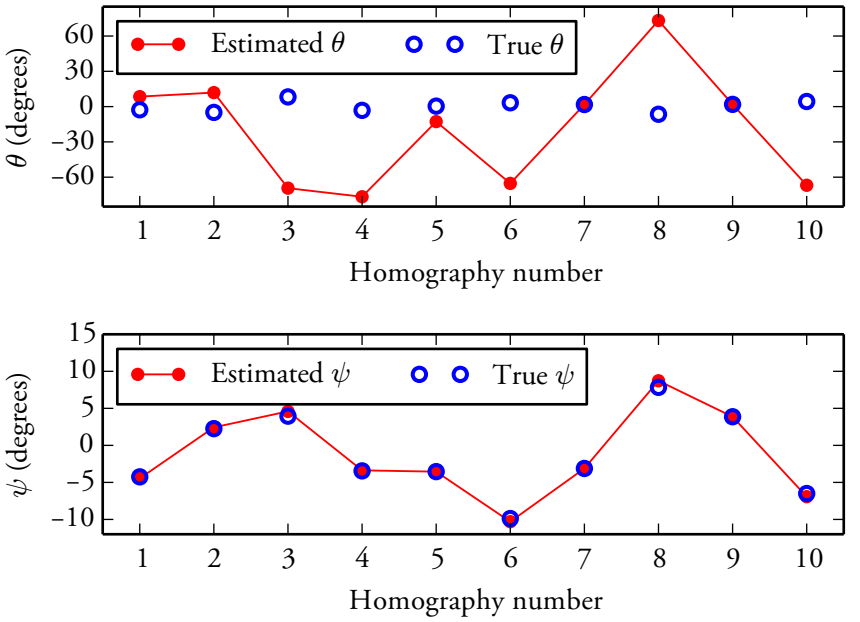

Figure 7: When $\boldsymbol{t}$ is a pure y-translation, θ seems to be unreliably estimated.

O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proceedings of the Second European Conference on Computer Vision*, volume 588 of *ECCV '92*, pages 563–578, Santa Margherita Ligure, Italy, 1992. Springer-Verlag.

H. Hajjdiab and R. Laganière. Vision-Based Multi-Robot Simultaneous Localization and Mapping. In *CRV '04: Proceedings of the 1st Canadian Conference on Computer and Robot Vision*, pages 155–162, Washington, DC, USA, 2004. IEEE Computer Society.

R. I. Hartley. Estimation of Relative Camera Positions for Uncalibrated Cameras. In *Proceedings of the Second European Conference on Computer*

Vision, volume 588, pages 579–587, Santa Margherita Ligure, Italy, 1992. Springer-Verlag.

R. Horaud and F. Dornaika. Hand-Eye Calibration. *International Journal of Robotics Research*, 14(3):195–210, 1995.

N. Karlsson, E. D. Bernardo, J. P. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich. The vSLAM Algorithm for Robust Localization and Mapping. In *ICRA '05: Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 24–29, Barcelona, Spain, 2005. IEEE.

O. Koch, M. R. Walter, A. S. Huang, and S. J. Teller. Ground Robot Navigation using Uncalibrated Cameras. In *ICRA '10: IEEE International Conference on Robotics and Automation*, pages 2423–2430, Anchorage, Alaska, USA, 2010. IEEE.

B. Liang and N. Pears. Visual Navigation using Planar Homographies. In *ICRA '02: Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pages 205–210, Washington, DC, USA, 2002.

R. Tsai and R. Lenz. A New Technique for Fully Autonomous and Efficient 3D Robotics Hand/Eye Calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358, June 1989.

Ego-Motion Recovery and Robust Tilt Estimation for Planar Motion Using Several Homographies

Mårten Wadenbäck and Anders Heyden
Centre for Mathematical Sciences, Lund University

Abstract: In this paper we suggest an improvement to a recent algorithm for estimating the pose and ego-motion of a camera which is constrained to planar motion at a constant height above the floor, with a constant tilt. Such motion is common in robotics applications where a camera is mounted onto a mobile platform and directed towards the floor. Due to the planar nature of the scene, images taken with such a camera will be related by a planar homography, which may be used to extract the ego-motion and camera pose. Earlier algorithms for this particular kind of motion were not concerned with determining the tilt of the camera, focusing instead on recovering only the motion. Estimating the tilt is a necessary step in order to create a rectified map for a SLAM system. Our contribution extends the aforementioned recent method, and we demonstrate that our enhanced algorithm gives more accurate estimates of the motion parameters.

1 Introduction

One of the long-standing aims in robotics research is the development of algorithms for autonomous navigation. A popular class of such algorithms are the ones concerned with so called *Simultaneous Localisation and Mapping* (SLAM), in which a mobile platform, equipped with an array of suitable sensors (laser scanners, cameras, odometers, sonar, …), explores and maps the surrounding environment while keeping track of its own location with respect to the map. The map created in the process should mark notable objects and landmarks in a way which allows for reliable re-identification. The type of map that can be created is highly dependent on the kinds of sensors employed and on the environment being mapped, and can range from sparsely placed points to dense and detailed textured 3D models.

Using cameras to build the map is becoming increasingly attractive, as they are cheap compared to many of the other sensors, and since the traditional obstacle of high computational cost becomes less inhibiting with time as computational power increases. Another advantage of using cameras is that it allows for utilisation of the increasingly sophisticated methods and great experience that the computer vision community has produced during the past few decades. Indeed, scene reconstruction from images is a classical and continually studied problem in computer vision, and various methods have been proposed for both general cases and specialised applications.

Many of the successful general reconstruction techniques are based on epipolar geometry, and in particular the *fundamental matrix*, which was introduced independently in Faugeras (1992) and Hartley (1992). Such methods make the implicit assumption that the data are not positioned in one of the so called *critical configurations*, and in many practical cases such degeneracies are indeed very unlikely to occur. However, one of the less unlikely critical configurations occurs when the data points are coplanar — indeed, the application to navigation that we describe in this paper *requires* the data points to lie in a plane. Since planar structures are very common in man-made environments, this is an area in which specialised algorithms which can avoid degeneracy can have great advantages.

While invariant local features, for instance SIFT (Lowe, 2004) and other similar features, are standard in *Structure from Motion* (SfM), their use in camera based SLAM has been less prevalent. One of the main reasons for this is probably, as observed in Davison et al. (2007), that though such features allow for accurate and robust re-identification, their computational cost has traditionally been obstructive for real time applications. Although this is essentially still a valid point, particularly on embedded systems or with high resolution images, computational power continues to improve. In our view, feature based approaches are inevitably becoming feasible for real-time operation.

2 Related Work

A robot mapping application not only requires an incremental reconstruction, as data becomes available sequentially, but in contrast to Structure from Motion approaches such as the popular Bundler system described in Snavely et al. (2008), the order in which views are added is more or less predetermined. Though the views are added to the reconstruction in a fixed order, some SLAM approaches allow the robot path itself to be planned so that the images can be taken from locations which make the reconstruction better (Haner and Heyden, 2011), but we will in this paper consider the path to already be decided. Some very early work which respects the restriction on the order of views is Harris and Pike (1988), in which a Kalman filter was used to estimate camera position based on inter-image point correspondences throughout a short image sequence. Probabilistic viewpoints based on extended Kalman filters (EKF) remain popular in later systems such as the vSLAM system (Karlsson et al., 2005) and the MonoSLAM system (Davison et al., 2007).

The systems mentioned above allow general 3D camera motion, but this is not always necessary or even desired. A camera that has been mounted onto a mobile platform will typically perform two-dimensional motion since it remains at a fixed height above the ground, and with this knowledge one can eliminate some of the uncertainty which 3D motion allows. Our work continues in the spirit of Liang and Pears (2002) and Hajjdiab and Laganière (2004) and others, in that we intend to navigate using images of the floor. Since the scene is planar, the images will be related by planar homographies.

Liang and Pears find the robot rotation angle φ by noting that the eigenvalues of the inter-image homography are (up to scale) 1 and $e^{\pm i\varphi}$, and they derive an expression for the translation from the eigenvectors. One drawback of this method is that it does not determine the tilt. Determining the tilt allows a rectified map to be created, and is therefore highly desirable.

A more recent method described in Wadenbäck and Heyden (2013) starts with estimating the tilt $\boldsymbol{R}_{\psi\theta}$, and then performs a QR decomposition

(a) Original image. (b) Rectified image.

Figure 1: A typical image taken by a camera under the conditions described in this paper is shown in Figure B.1(a). A rectified version, as if seen straight from above, can be seen in Figure B.1(b). In order to rectify such images, it is necessary to be able to estimate the camera tilt.

of $\boldsymbol{R}_{\psi\theta}^{T}\boldsymbol{H}\boldsymbol{R}_{\psi\theta}$ to determine φ and the translation (t_x, t_y).

We show in this paper how to extend this estimation algorithm to use more than one homography for estimating the tilt. This improves robustness to noise and erroneous measurements.

3 Problem Geometry

We shall consider the navigation of a mobile platform equipped with a single camera that has been mounted rigidly onto the platform and directed towards the floor. This setup means that the camera will move at constant height in a plane parallel to the floor, and have a constant angle to the plane normal (tilt). Figure 1 shows a typical image from one of our datasets, taken under the conditions described here. Figure 2 shows an illustration of the geometrical situation. We will further assume zero skew and square pixels, and that the camera parameters remain constant during the motion (no zooming or refocusing). It will be convenient to work with a global coordinate system in which the camera moves in the plane $z = 0$ and the ground plane is represented by the plane $z = 1$.

As already noted, two images will be related by a planar homography.

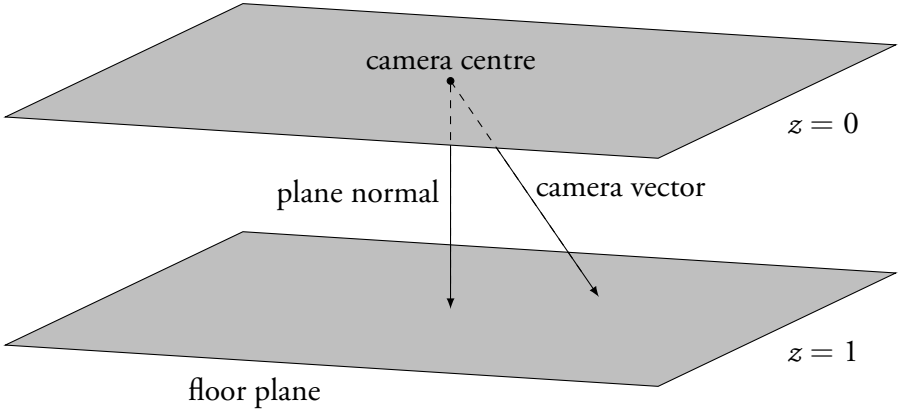

Figure 2: The camera moves freely in the plane $z = 0$, and can rotate about the normal of the plane, but the angle to the plane normal (tilt) is held constant.

We model the camera motion by a translation $\boldsymbol{t} = (t_x, t_y, 0)$ and a rotation $\boldsymbol{R}_\varphi$ an angle φ about the normal of the floor plane (the z-axis). Using homogeneous coordinates in the plane, the motion of the camera is represented by the transformation $\boldsymbol{R}_\varphi \boldsymbol{T}$, with

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

If the camera is tilted, the camera coordinate system and the world coordinate system are related by a rotation $\boldsymbol{R}_{\psi\theta} = \boldsymbol{R}_\psi \boldsymbol{R}_\theta$. This means that the inter-image homographies will be of the form

$$\boldsymbol{H} = \lambda \boldsymbol{R}_{\psi\theta} \boldsymbol{R}_\varphi \boldsymbol{T} \boldsymbol{R}_{\psi\theta}^T, \tag{2}$$

where $\lambda \neq 0$ is an unknown scale parameter.

Estimating these homographies from the images can be done using point correspondences and a robust method such as RANSAC (Fischler and Bolles, 1981). This is not the focus of our work, and we will henceforth assume that well-estimated homographies are available, without concerning ourselves with how they were obtained.

4 Parameter Recovery

Suppose we have a number of homographies of the form in (2), that is,

$$\boldsymbol{H}_j = \lambda_j \boldsymbol{R}_{\psi\theta} \boldsymbol{R}_{\varphi_j} \boldsymbol{T}_j \boldsymbol{R}_{\psi\theta}^T, \quad j = 1, \dots N, \tag{3}$$

and want to recover the motion parameters. As observed in Wadenbäck and Heyden (2013), the products

$$\boldsymbol{M}_j = \begin{bmatrix} m_{11}^j & m_{12}^j & m_{13}^j \\ m_{12}^j & m_{22}^j & m_{23}^j \\ m_{13}^j & m_{23}^j & m_{33}^j \end{bmatrix} = \boldsymbol{H}_j^T \boldsymbol{H}_j \tag{4}$$

are all independent of φ.

An iterative scheme is also presented which alternates between solving for ψ and θ, keeping the other one fixed. Their paper demonstrates that this can be accomplished by finding the null space of the matrix

$$\boldsymbol{\Psi}_j = \begin{bmatrix} \widehat{m}_{11}^j - \widehat{m}_{22}^j & -2\widehat{m}_{23}^j & \widehat{m}_{11}^j - \widehat{m}_{33}^j \\ \widehat{m}_{12}^j & \widehat{m}_{13}^j & 0 \\ 0 & \widehat{m}_{12}^j & \widehat{m}_{13}^j \end{bmatrix} \tag{5}$$

in the ψ case (where $\widehat{\boldsymbol{M}}_j = \boldsymbol{R}_\theta^T \boldsymbol{M}_j \boldsymbol{R}_\theta$), and of the matrix

$$\boldsymbol{\Theta}_j = \begin{bmatrix} \widehat{m}_{11}^j - \widehat{m}_{22}^j & -2\widehat{m}_{13}^j & \widehat{m}_{33}^j - \widehat{m}_{22}^j \\ \widehat{m}_{12}^j & -\widehat{m}_{23}^j & 0 \\ 0 & \widehat{m}_{12}^j & -\widehat{m}_{23}^j \end{bmatrix} \tag{6}$$

in the θ case (with $\widehat{\boldsymbol{M}}_j = \boldsymbol{R}_\psi^T \boldsymbol{M}_j \boldsymbol{R}_\psi$). It can clearly be seen that these matrices have at least rank two, except in the case where the bottom two rows are identically zero, so a one dimensional null space is expected. Due to measurement errors the null space will in practice be trivial, and a one dimensional approximation is computed as the singular vector $\boldsymbol{v} = (v_1, v_2, v_3)$ corresponding to the smallest singular value. In the ψ case, any vector $\boldsymbol{v}$ in the null space should be a scalar multiple of $(c_\psi^2, c_\psi s_\psi, s_\psi^2)$, which gives

$$\psi = \frac{1}{2} \arcsin \frac{2v_2}{v_1 + v_3}, \tag{7}$$

while in the same way, the the solution in the θ case is a scalar multiple of $(c_\theta^2, c_\theta s_\theta, s_\theta^2)$, and

$$\theta = \frac{1}{2} \arcsin \frac{2v_2}{v_1 + v_3}. \tag{8}$$

This paper presents the insight that if the tilt $\boldsymbol{R}_{\psi\theta}$ remains constant, then the matrices $\boldsymbol{\Psi}_j$ all should have the same null space. Instead of considering each $\boldsymbol{\Psi}_j$ separately, we can therefore solve

$$\boldsymbol{\Psi}\boldsymbol{v} = \begin{bmatrix} \boldsymbol{\Psi}_1 \\ \vdots \\ \boldsymbol{\Psi}_N \end{bmatrix} \begin{bmatrix} c_\psi^2 \\ c_\psi s_\psi \\ s_\psi^2 \end{bmatrix} = \boldsymbol{0}. \tag{9}$$

In the same way, we may combine the equations for θ into

$$\boldsymbol{\Theta}\boldsymbol{v} = \begin{bmatrix} \boldsymbol{\Theta}_1 \\ \vdots \\ \boldsymbol{\Theta}_N \end{bmatrix} \begin{bmatrix} c_\theta^2 \\ c_\theta s_\theta \\ s_\theta^2 \end{bmatrix} = \boldsymbol{0}. \tag{10}$$

The angles are computed from the solution $\boldsymbol{v}$ in the same way as above using (7) and (8).

5 Experiments

For the purpose of comparing the original algorithm outlined in Wadenbäck and Heyden (2013) with our enhanced version, we have randomly generated a large number of homographies of the form in (3). Gaußian noise with a standard deviation of 0.5° was added to each of the angles, intended to simulate measurement noise. Figure 3 shows the estimation results obtained using only one homography at a time, and Figure 4 shows the results using our proposed method with five homographies used at each step. The same number of iterations were used for the two methods. Note that the scale on the axes is the same in both figures, for the benefit of easier comparison. It is readily seen that the proposed method drastically decreases the number of cases where the algorithm fails to converge.

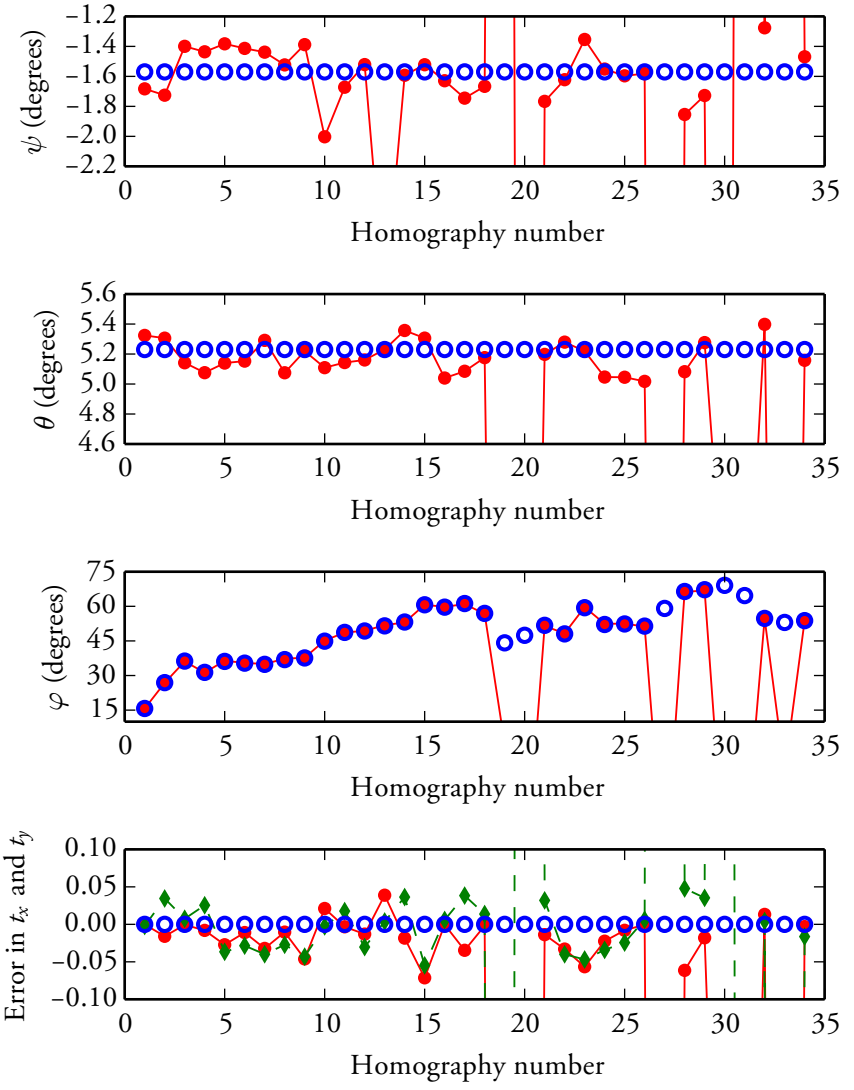

Figure 3: Estimates from one homography at a time using the unmodified method. In the top three plots, the red bullets are the estimated angles. In the bottom plot, the red bullets and green diamonds show the error in t_x and t_y, respectively. The blue circles represent ground truth.

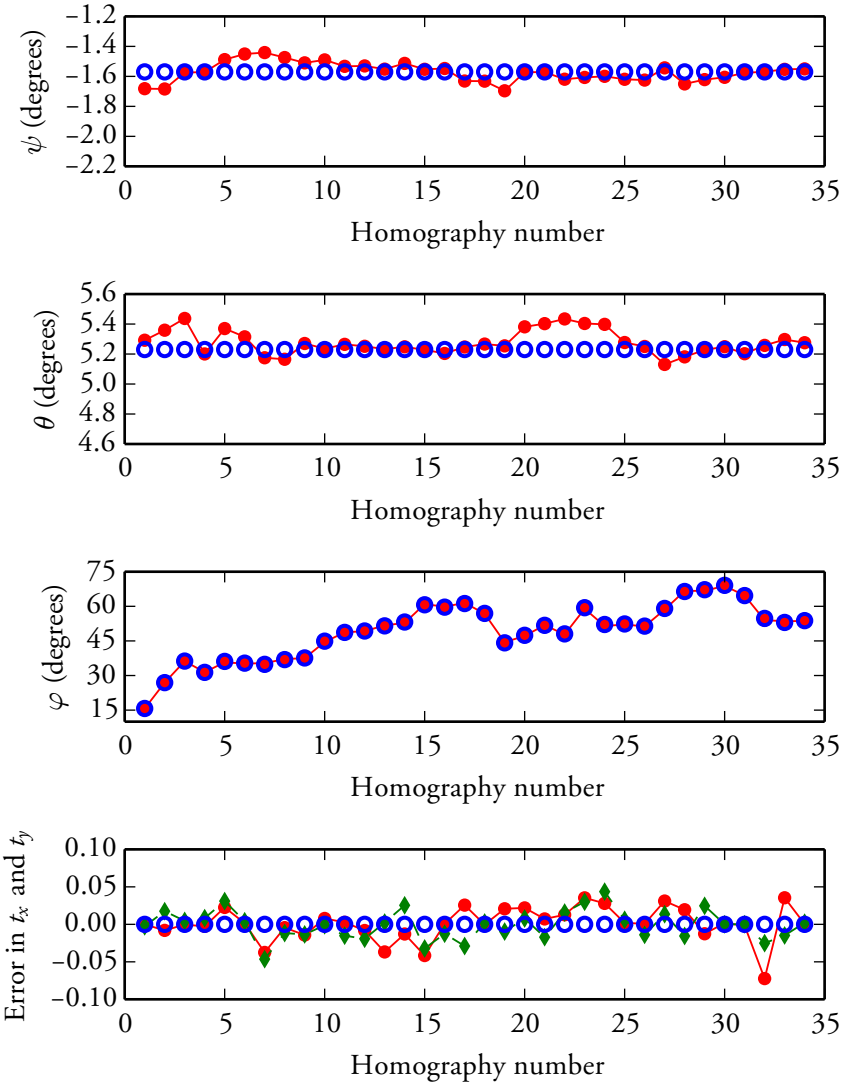

Figure 4: Estimates from five homographies at a time using our proposed method. In the top three plots, the red bullets are the estimated angles. In the bottom plot, the red bullets and green diamonds show the error in t_x and t_y, respectively. The blue circles represent ground truth.

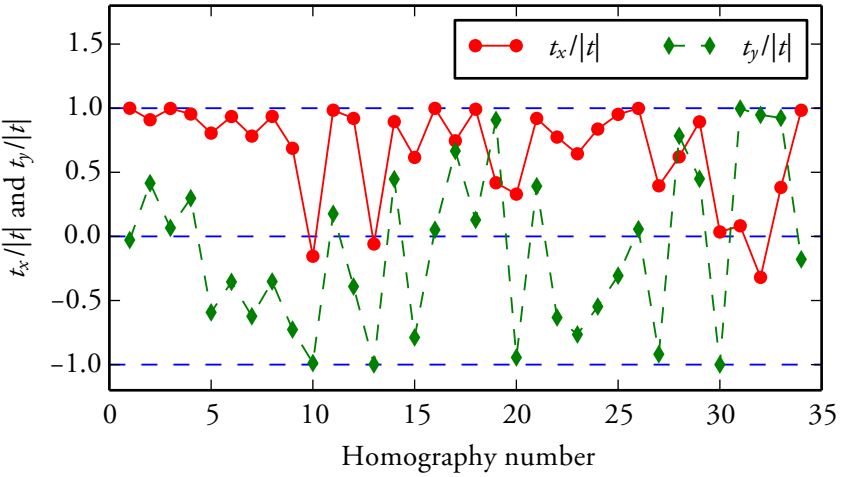

Figure 5: The x- and y components of the translation that was used to generate the homographies, normalised by the length of the translation in that step. Some of the translations used are apparently close to pure x-translations or pure y-translations, which were reported to be problematic for the original algorithm.

It should be pointed out that while the results from the unmodified method can be much improved using filtering techniques, the same is true for our enhanced method.

The unmodified algorithm was reported to have difficulties when the translation was close to a pure x-translation or a pure y-translation. In the case of an x-translation, θ would be poorly estimated, and conversely for a y-translation. Figure 5 shows the x- and y components of the translation used to generate the homographies, normalised by the length of the translation in that step. Certainly, some of the translations are close to pure x-translations or y-translations, and some of them do indeed coincide with bad estimates in Figure 3. The proposed method, on the other hand, handles these translations without significant difficulties, as Figure 4 confirms.

6 Conclusion

In this paper we have extended the estimation method in Wadenbäck and Heyden (2013) to use more than one homography to estimate the tilt. This enhancement produces a robuster and more accurate estimate, which demonstrably allows the other motion parameters to be recovered with higher precision. The problems with ill-conditioned motion patterns that were reported in for the original algorithm have also been remedied by using more than one homography at a time.

Acknowledgements

This work has been funded by the Swedish Foundation for Strategic Research through the SSF project ENGROSS (web page at `www.engross.lth.se`).

Bibliography

A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proceedings of the Second European Conference on Computer Vision*, volume 588 of *ECCV '92*, pages 563–578, Santa Margherita Ligure, Italy, 1992. Springer-Verlag.

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*, 24(6):381–395, 1981.

H. Hajjdiab and R. Laganière. Vision-Based Multi-Robot Simultaneous Localization and Mapping. In *CRV '04: Proceedings of the 1st Canadian Conference on Computer and Robot Vision*, pages 155–162, Washington, DC, USA, 2004. IEEE Computer Society.

S. Haner and A. Heyden. Optimal View Path Planning for Visual SLAM. In *Proceedings of the 17th Scandinavian Conference on Image Analysis (SCIA)*, volume 6688 of *Lecture Notes in Computer Science*, pages 370–380. Springer Berlin Heidelberg, 2011.

C. G. Harris and J. M. Pike. 3D Positional Integration from Image Sequences. *Image and Vision Computing*, 6(2):87–90, 1988.

R. I. Hartley. Estimation of Relative Camera Positions for Uncalibrated Cameras. In *Proceedings of the Second European Conference on Computer Vision*, volume 588, pages 579–587, Santa Margherita Ligure, Italy, 1992. Springer-Verlag.

N. Karlsson, E. D. Bernardo, J. P. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich. The vSLAM Algorithm for Robust Localization and Mapping. In *ICRA '05: Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 24–29, Barcelona, Spain, 2005. IEEE.

B. Liang and N. Pears. Visual Navigation using Planar Homographies. In *ICRA '02: Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pages 205–210, Washington, DC, USA, 2002.

D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004. ISSN 0920-5691.

N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the World from Internet Photo Collections. *International Journal of Computer Vision*, 80(2): 189–210, 2008.

M. Wadenbäck and A. Heyden. Planar Motion and Hand-Eye Calibration Using Inter-Image Homographies from a Planar Scene. In *Proceedings of VISIGRAPP 2013*, pages 164–168, Barcelona, Spain, February 2013. SCITEPRESS.

Trajectory Estimation Using Relative Distances Extracted from Inter-Image Homographies

Mårten Wadenbäck and Anders Heyden
Centre for Mathematical Sciences, Lund University

Abstract: The main idea of this paper is to use distances between camera positions to recover the trajectory of a mobile robot. We consider a mobile platform equipped with a single fixed camera using images of the floor and their associated inter-image homographies to find these distances. We show that under the assumptions that the camera is rigidly mounted with a constant tilt and travelling at a constant height above the floor, the distance between two camera positions may be expressed in terms of the condition number of the inter-image homography. Experiments are conducted on synthetic data to verify that the derived distance formula gives distances close to the true ones and is not too sensitive to noise. We also describe how the robot trajectory may be represented as a graph with edge lengths determined by the distances computed using the formula above, and present one possible method to construct this graph given some of these distances. The experiments show promising results.

1 Introduction

Autonomous navigation is a central theme in mobile robotics applications, and the development of methods for *Simultaneous Localisation and Mapping* (SLAM) has long been a major area of research. The scenario for such algorithms is that a mobile robot makes use of a number of suitable sensors (laser range finders, cameras, odometers, sonar, …) to autonomously map and explore its surroundings. The type of map that can be created is highly dependent on the kinds of sensors employed and on the environment being mapped, but typically the aim is to mark notable objects and landmarks in a way that allows for reliable re-identification. This paper will not consider the mapping part, but will only focus on recovering the robot trajectory.

In recent years there has been an increased interest in methods for

SLAM which primarily rely on cameras for navigation, to a large extent thanks to the rapid increase in computational power during this period. The computationally intense image processing involved in making use of the images has indeed been a major inhibiting factor for their use in real-time systems, and still makes it difficult to use high resolution images in such applications. However, since the price of a decent camera is much lower than that of some other sensors which have traditionally been used for SLAM, cameras naturally find a larger user base, and development goes on despite the computational obstacles.

A common scenario in mobile robotics is that of a camera that has been rigidly mounted onto some kind of mobile platform. If the robot is expected to operate in environments where there are people, it may be a good idea to direct the camera towards the floor, since this means that movable or deformable objects will not occlude or be mistaken for the stationary scene. Under such circumstances, since the scene is planar, images taken at different locations will be related by a planar homography. If the camera motion is planar, which is the case if the suspension of the platform is negligible, this allows for an explicit parametrisation of the homography, as explained in Section 2.2.

Our hope is to use information about the distances between the camera centres to improve the trajectory recovery of a SLAM system incorporating a number of sensors besides vision. Incorporating measurements from other sensors could of course, if done with care and thought, improve the accuracy above the level achieved in this paper. We intend to do so at a later stage, but in this paper, we will investigate what can be achieved using only information about the distances.

The proposed method relies on reliably and accurately estimated homographies between the camera poses. Practical methods for finding a homography between two partially overlapping images is given a thorough and in-depth treatment in Hartley and Zisserman (2004), and the details of this is outside the scope of this work.

The organisation of the paper is as follows. Section 1.1 gives a brief overview of related literature, and Section 2 describes the camera set-up and the geometry of the problem. In Section 3 we devise a method for comput-

ing the distances. A method using the inter-camera distances to represent the trajectory as a graph is outlined in Section 4. Experiments investigating the the accuracy of distance formula are described in Section 5.1, and preliminary results for a path estimation problem are shown in Section 5.2. Section 6 concludes the findings and experiments in this paper.

1.1 Related Work

Many successful approaches to the SLAM problem are based on probabilistic viewpoints, where the extended Kalman filters (EKF) remain popular in recent systems such as the vSLAM system by Karlsson et al. (2005) and the MonoSLAM system by Davison et al. (2007). The creation of a map is part of the SLAM problem, and it is often deemed essential to have a good map in order to allow for accurate trajectory recovery. The present paper shows that the trajectory can in fact be accurately estimated without considering the mapping problem.

The problem of determining the locations of a number of points given all inter-point distances was studied in the thirties by Young and Householder (1938), and they gave a method for finding the locations. Their paper studies the problem in n-dimensional space, but a method for finding a lower-dimensional approximation is mentioned. Their method needs all inter-point distances to be known, which may not always be the case.

This problem of finding the locations from distances has a number of inherent ambiguities, since rotations, translations, and reflections do not influence the distances. However, the point locations need not be uniquely determined, even after factoring out these ambiguities. These ambiguities are of interest in the study of rigidity of graphs, for example the works by Asimow and Roth (1978, 1979). In addition to the general situation, there has been work done on various special cases, such as for bipartite graphs (Bolker and Roth, 1980).

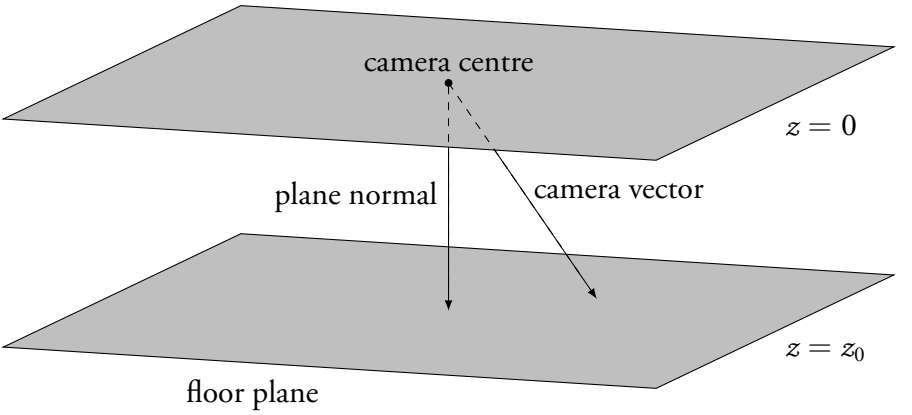

Figure 1: The camera moves freely in the plane $z = 0$ at a constant height above the floor. Rotations about the normal of the plane are allowed, but the angle to the plane normal (tilt) is held constant.

2 Problem Geometry

2.1 Camera Parametrisation

In this paper we consider a standard pinhole perspective camera with square pixels and zero skew, which has been rigidly mounted onto a mobile robot and directed towards the floor. We will further assume that no zooming or refocusing occurs during the motion. The geometrical situation under consideration is illustrated in Figure 1. This situation or similar ones are not uncommon in mobile robotics applications, and have been considered in Liang and Pears (2002), Hajjdiab and Laganière (2004), Taddei et al. (2012) and Wadenbäck and Heyden (2013), among others. Since the camera will remain at a fixed height above the floor, it will be convenient to work with a global coordinate system in which the camera moves in the plane $z = 0$ and the ground plane is represented by the plane $z = z_0$.

In order to describe the direction of the camera, we use three rotations $\boldsymbol{R}_\psi$, $\boldsymbol{R}_\theta$ and $\boldsymbol{R}_\varphi$. The rotation $\boldsymbol{R}_\varphi$ is a rotation the angle φ about the z-axis, while $\boldsymbol{R}_\psi$ and $\boldsymbol{R}_\theta$ correspondingly describe rotations about the x- and y-

axes, respectively. The *camera tilt* is $\boldsymbol{R}_{\psi\theta} = \boldsymbol{R}_{\psi}\boldsymbol{R}_{\theta}$ and will be identical for all images, while φ and the camera centre $\boldsymbol{t} = (t_x, t_y, 0)$ may vary between images.

Under these conditions, the camera projection matrix associated with an image taken at position $\boldsymbol{t}$ will be

$$\boldsymbol{P} = \boldsymbol{R}_{\psi\theta}\boldsymbol{R}_{\varphi}[\boldsymbol{I} \mid -\boldsymbol{t}]. \tag{1}$$

For simplicity of presentation, we will in the remainder of this paper assume that $z_0 = 1$. The choice of z_0 only determines the global scale factor, so this is not a restriction, and it is straightforward to consider other choices of z_0.

2.2 The Inter-Image Homography

We will now consider two images taken at different locations under the geometrical premises described in Section 2.1. The global coordinate system may be chosen in such a way that one of the cameras is in the origin and aligned with the coordinate system, in which case the camera projection matrices associated with the two images become

$$\begin{aligned} \boldsymbol{P}_1 &= \boldsymbol{R}_{\psi\theta}[\boldsymbol{I} \mid \boldsymbol{0}], \\ \boldsymbol{P}_2 &= \boldsymbol{R}_{\psi\theta}\boldsymbol{R}_{\varphi}[\boldsymbol{I} \mid -\boldsymbol{t}]. \end{aligned} \tag{2}$$

A point in the floor plane, $X = \begin{bmatrix} x & y & 1 & 1 \end{bmatrix}^T$, will be projected into the first image as

$$\boldsymbol{x}_1 = \boldsymbol{P}_1 X = \boldsymbol{R}_{\psi\theta} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3}$$

and into the second image as

$$\boldsymbol{x}_2 = \boldsymbol{P}_2 X = \boldsymbol{R}_{\psi\theta}\boldsymbol{R}_{\varphi} \begin{bmatrix} x - t_x \\ y - t_y \\ 1 \end{bmatrix} = \boldsymbol{R}_{\psi\theta}\boldsymbol{R}_{\varphi}\boldsymbol{T} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{4}$$

where

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

From (3) and (4) it follows that the homography between the two images is (a scalar multiple of)

$$\boldsymbol{H} = \boldsymbol{R}_{\psi\theta}\boldsymbol{R}_{\varphi}\boldsymbol{T}\boldsymbol{R}_{\psi\theta}^{T}. \tag{6}$$

For two partially overlapping images of a planar scene, the homography $\boldsymbol{H}$ can be robustly and accurately estimated (for the details, see for example Hartley and Zisserman (2004)), but the problem of finding the parameters on the right hand side seems to be a decidedly less well studied problem. While this paper uses the existence of the decomposition (6), we make no effort to explicitly determine the individual parameters.

3 Finding the Travelled Distance

Since $\boldsymbol{R}_{\psi\theta}$ and $\boldsymbol{R}_{\varphi}$ are both orthonormal matrices, $\boldsymbol{H}$ and $\boldsymbol{T}$ must have the same condition number, which we shall denote by κ. This observation suggests that there might be a formula relating κ to the travelled distance between the two images.

For the purpose of finding this relation, we compute κ via the eigenvalues of $\boldsymbol{T}^T\boldsymbol{T}$. The characteristic polynomial of $\boldsymbol{T}^T\boldsymbol{T}$ is

$$(\sigma - 1)\left(\sigma^2 - (2 + t_x^2 + t_y^2)\sigma + 1\right), \tag{7}$$

and we see immediately that one singular value of $\boldsymbol{T}$ equals one, and that the product of the other two also equals one. We thus have the three singular values $\sigma_1 \geq 1$, $\sigma_2 = 1$ and $\sigma_3 = \frac{1}{\sigma_1} \leq 1$, so that

$$\kappa = \frac{\sigma_1}{\sigma_3} = \sigma_1^2. \tag{8}$$

If we denote the distance travelled between the images by $d = \sqrt{t_x^2 + t_y^2}$, then κ becomes

$$\kappa = \left(1 + \frac{d^2}{2} + \frac{d}{2}\sqrt{4 + d^2}\right)^2, \tag{9}$$

and solving (9) for d yields the simple expression

$$d = \frac{\sqrt{\kappa} - 1}{\sqrt[4]{\kappa}}. \tag{10}$$

This relation between the travelled distance and the condition number of the homography is investigated experimentally in Section 5.1 in terms of accuracy and sensitivity to noise. We now turn to the problem of determining the trajectory if a number of distances are given.

4 Finding the Trajectory

Let us assume that we have a series of images taken at locations $\boldsymbol{p}_j = (x_j, y_j)$ for $j = 1, \ldots, N$, and that there is sufficient overlap so that it is possible to compute the homography to at least the three previous points for $j = 4, \ldots, N$. This means that it is possible to compute some of the distances $d_{j,k} = \|\boldsymbol{p}_j - \boldsymbol{p}_k\|$. See Figure 2.

Since our reconstruction can only be determined up to a an unknown rotation, an unknown reflection and an unknown translation, we may without loss of generality assume that $\boldsymbol{p}_1 = (0, 0)$, $y_2 = 0$ and that the first non-zero y-coordinate is positive. This fixates the coordinate system with respect to those ambiguities.

If we assume that the distances $d_{j,k}$ are measured with independent and identically distributed zero-mean Gaußian errors, the ℓ_2-optimal locations of the $\boldsymbol{p}_j$ are found by solving the minimisation problem (assuming y_3 is

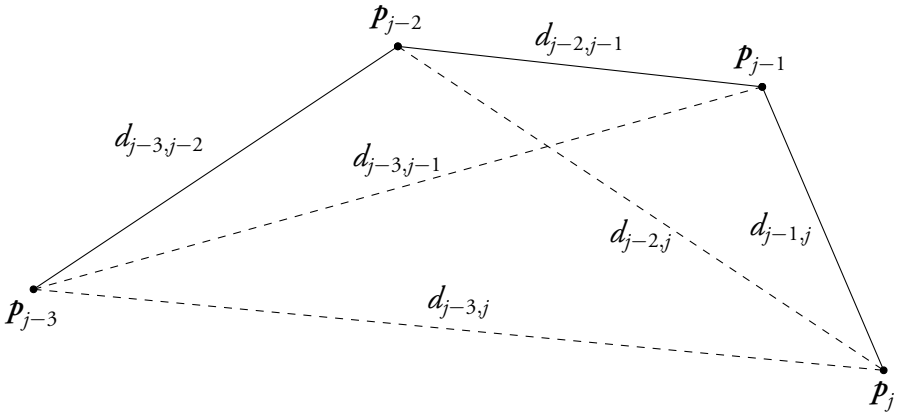

Figure 2: The robot trajectory is represented by the solid lines, and the dashed lines represent the other distances which have been measured.

the first non-zero y-coordinate)

$$\begin{aligned} \text{minimise} \quad & \sum_{(j,k)\in S} \left(\|\boldsymbol{p}_j - \boldsymbol{p}_k\| - d_{j,k} \right)^2 \\ \text{subject to} \quad & \boldsymbol{p}_1 = (0,0) \\ & y_2 = 0 \\ & y_3 > 0 \end{aligned} \tag{11}$$

where the set S which the sum is taken over is the set of (j,k) for which the distances $d_{j,k}$ have been measured. The optimisation problem (11) is not convex, and there is a significant risk that we will not find the global minimum, but only a local one. For this reason it is important to create a good initial guess before optimising (11) using some local method, for instance Gauß-Newton or Conjugate Gradient method (CG).

4.1 Constructing an Initial Guess

In order to construct an initial guess for $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$ we initialise $\boldsymbol{p}_1$, $\boldsymbol{p}_2$ and $\boldsymbol{p}_3$, and then successively find candidates for $\boldsymbol{p}_4, \ldots, \boldsymbol{p}_N$. Let $C_{j,k}$ be the circle at $\boldsymbol{p}_j$ with radius $d_{j,k}$. A point $\boldsymbol{p}_j$ is found from the three previous

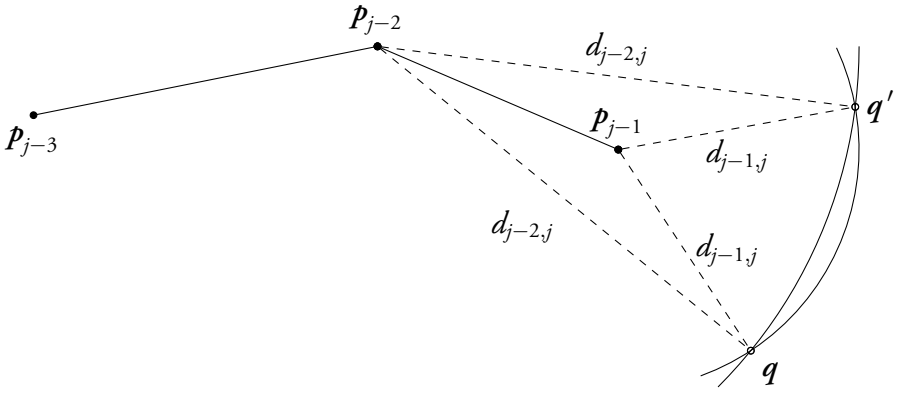

Figure 3: The points q and q' are both candidates for an initial guess of p_j since both have the same distances to p_{j-2} and p_{j-1}. Which candidate to choose can be determined by comparing $\|p_{j-3} - q\|$ and $\|p_{j-3} - q'\|$ to $d_{j-3,j}$ and taking the one for which the distance matches best.

points as one of the intersections of $C_{j-1,j}$ and $C_{j-2,j}$ if they intersect, or as the mean of their respective closest points otherwise. Which of the intersections to choose is determined by considering which best matches the distance $d_{j-3,j}$. Figure 3 illustrates the intersecting case, and the overall algorithm can be seen in Figure 3.

5 Experiments

5.1 Accuracy of Distances

In this experiment we use synthetic data to investigate how well (10) corresponds to the true distance. For this purpose, we simulate a number of cameras (resolution 2000 by 2000 pixels) of the the form (1) with identical tilt, looking at a number of synthetically generated keypoints. For each pair of cameras, the ground truth distance is computed, and a subset of the keypoints is chosen. After this subset is projected into the two views, and noise is added, a homography is estimated for the pair. From this homography, the distance is computed using (10).

Input: Distances $d_{j,k}$ for all j, k such that $|j-k| < 4$
Output: An initial guess for $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$

$\boldsymbol{p}_1 \leftarrow (0,0)$
$\boldsymbol{p}_2 \leftarrow (d_{1,2}, 0)$
Let $C_{j,k}$ denote the circle at $\boldsymbol{p}_j$ with radius $d_{j,k}$
Set $\boldsymbol{p}_3$ to the intersection of $C_{2,3}$ and $C_{1,3}$ with $y_3 \geq 0$
for $j = 4, \ldots, N$ **do**
    Find intersections $\boldsymbol{q}$ and $\boldsymbol{q}'$ of $C_{j-1,j}$ and $C_{j-2,j}$
    $d_q \leftarrow \|\boldsymbol{q} - \boldsymbol{p}_{j-3}\|, \quad d_{q'} \leftarrow \|\boldsymbol{q}' - \boldsymbol{p}_{j-3}\|$
    if $|d_q - d_{j-3,j}| > |d_{q'} - d_{j-3,j}|$ **then**
        $\boldsymbol{p}_j \leftarrow \boldsymbol{q}'$
    else
        $\boldsymbol{p}_j \leftarrow \boldsymbol{q}$
    end if
end for

Algorithm 3: Construction of an initial guess to the minimisation problem (11). The idea is to use the distances to the two most recent points to get two candidates for the next point, and then selecting the candidate which best matches the distance to the third most recent point.

Using the synthetic data described above, we have investigated the quotient of the estimated distance and the ground truth distance. Figure 4 shows how the standard deviation of this quotient depends on the noise level, and Figure 5 shows the particular distribution at a noise level of four pixels in both the horizontal and vertical directions. All the noise levels we have tried show distributions very similar to the one in Figure 5.

Even though it is clear from these results that the distance estimated using (10) indeed is close to the true distance, it is interesting to note that it on average tends to be slightly shorter. Table 1 appears to suggest that the estimate will typically be about one percent shorter than the true distance. Furthermore, this one percent difference appears to be more or less independent of the noise level, and is present even when no noise is added. Further investigation will be required if this discrepancy is to be explained.

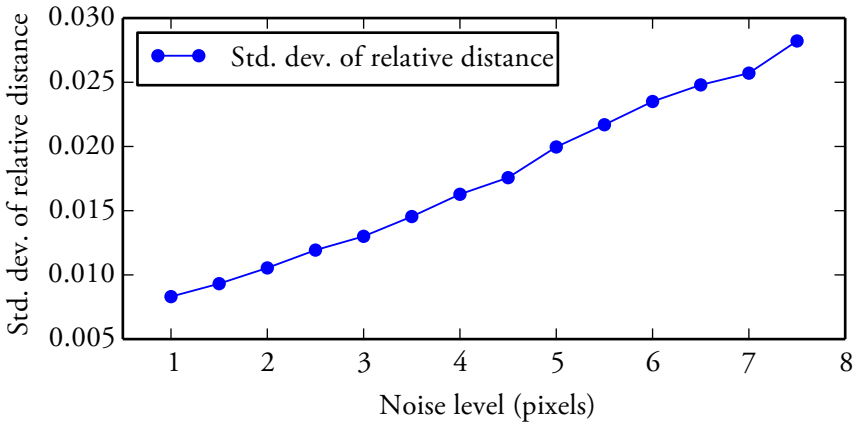

Figure 4: This plot shows the standard deviation of the quotient of the estimated distance and the ground truth distance at some different noise levels. The noise level along the x-axis should be interpreted as the standard deviation of a Gaußian noise added to each coordinate before computing the homography.

5.2 Trajectory Estimation

This experiment evaluates the scheme for recovering the trajectory that was outlined in Section 4. For this purpose, a sequence of cameras of the form (1) were generated, and the homographies between each pair of cameras were computed in the same way as in the previous section, and from each homography the inter-camera distance was obtained using the distance formula (10).

An initial guess for the trajectory was then constructed as described in Section 4.1, and this was used to initialise the optimisation in (11). Figure 6 shows the resulting trajectories for both the case when all distances were used and the case when only the distances to the six most recent camera positions was used. (The reason for not using the distances to all other

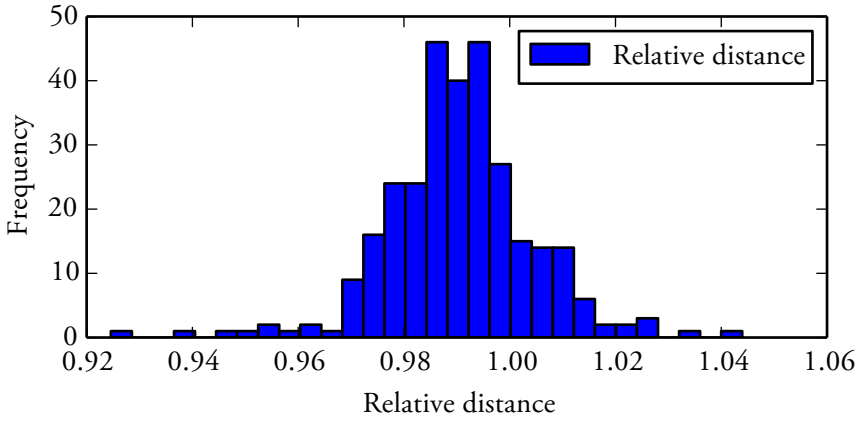

Figure 5: This histogram shows the distribution of the quotient of the estimated distance and the ground truth distance at a noise level of four pixels.

Table 1: Mean, median and standard deviation of the quotient of the estimated distance and the ground truth distance at different noise levels. The estimated distance appears to be on average one percent shorter than the true distance.

Noise level	Mean	Median	Std. dev.
0 pixels	0.9877	0.9894	0.0058
1 pixel	0.9907	0.9922	0.0083
2 pixels	0.9899	0.9908	0.0105
3 pixels	0.9904	0.9913	0.0130
4 pixels	0.9907	0.9900	0.0163
5 pixels	0.9864	0.9889	0.0200
6 pixels	0.9904	0.9891	0.0235
7 pixels	0.9922	0.9885	0.0256

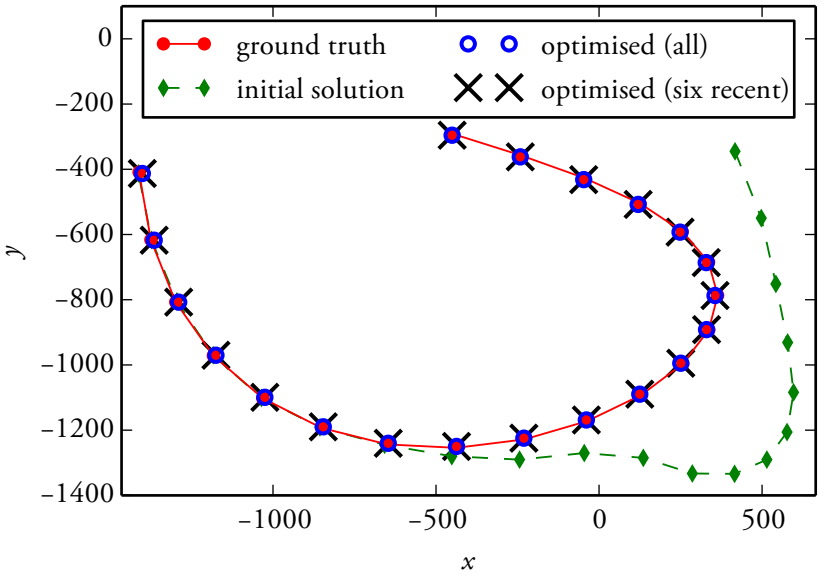

Figure 6: In this experiment we have simulated a path and estimated the trajectory. A noise level of three pixels was used in this particular example. The initial guess (green diamonds) was obtained using the algorithm outlined in Figure 3. The optimisation problem (11) was then solved using the initial guess as starting point. We see that the estimates end up close to the ground truth (red bullets) both when all distances are used and when only the six most recent are used.

cameras is to simulate the fact that for real images only a few consecutive images will overlap sufficiently to compute a homography.)

In Figure 6, the trajectory estimated using only some of the most recent distances is indistinguishable from the trajectory estimated using all distances, but using all distances will generally give more accurate results, especially for longer trajectories or higher noise levels. As mentioned however, in practice only distances to a few recent positions will be available, but this still produces good results.

6 Conclusion

This paper has shown how the distance between two cameras of the form (1) may be expressed in terms of the condition number of the inter-image homography. It has been demonstrated experimentally that this formula gives practically useful values for the distance.

In addition, we have shown how these distances may be used to recover the trajectory of a mobile platform, by representing the trajectory as a graph with edge lengths given by these distances. Preliminary experiments on synthetic data show that the method can recover the trajectory accurately.

As mentioned in Section 1.1, depending on which inter-camera distances are known and depending on their values, there may be further ambiguities apart from rotations, translations, and reflections. For instance, if the trajectory contains long straight parts, then it is expected that these may cause "flips". This is because the solution is not unique. If other sensor data is available, this could perhaps be detected and remedied. Since it is not inconceivable for robots to travel along almost straight lines at times, this would be a natural direction to explore in the future.

Acknowledgements

This work has been funded by the Swedish Foundation for Strategic Research through the SSF project ENGROSS (web page at `www.engross.lth.se`).

Bibliography

L. Asimow and B. Roth. The Rigidity of Graphs. *Transactions of the American Mathematical Society*, 245:279–289, 1978.

L. Asimow and B. Roth. The rigidity of graphs, II. *Journal of Mathematical Analysis and Applications*, 68(1):171–190, 1979.

E. D. Bolker and B. Roth. When is a bipartite graph a rigid framework? *Pacific Journal of Mathematics*, 90(1):27–44, 1980.

A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

H. Hajjdiab and R. Laganière. Vision-Based Multi-Robot Simultaneous Localization and Mapping. In *CRV '04: Proceedings of the 1st Canadian Conference on Computer and Robot Vision*, pages 155–162, Washington, DC, USA, 2004. IEEE Computer Society.

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, England, United Kingdom, Second edition, 2004. ISBN 0521540518.

N. Karlsson, E. D. Bernardo, J. P. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich. The vSLAM Algorithm for Robust Localization and Mapping. In *ICRA '05: Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 24–29, Barcelona, Spain, 2005. IEEE.

B. Liang and N. Pears. Visual Navigation using Planar Homographies. In *ICRA '02: Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pages 205–210, Washington, DC, USA, 2002.

P. Taddei, F. Espuny, and V. Caglioti. Planar Motion Estimation and Linear Ground Plane Rectification using an Uncalibrated Generic Camera.

International Journal of Computer Vision, 96(2):162–174, 2012. ISSN 09205691.

M. Wadenbäck and A. Heyden. Planar Motion and Hand-Eye Calibration Using Inter-Image Homographies from a Planar Scene. In *Proceedings of VISIGRAPP 2013*, pages 164–168, Barcelona, Spain, February 2013. SCITEPRESS.

G. Young and A. S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3(1):19–22, 1938.