Visual odometry using a homography formulation with decoupled rotation and translation estimation using minimal solutions

Banglei Guan, Pascal Vasseur, Cédric Demonceaux, Friedrich Fraundorfer

Visual odometry using a homography formulation with decoupled rotation and translation estimation using minimal solutions

Banglei Guan[1], Pascal Vasseur[2], Cédric Demonceaux[3] and Friedrich Fraundorfer[4]

***Abstract*— In this paper we present minimal solutions for two-view relative motion estimation based on a homography formulation. By assuming a known vertical direction (e.g. from an IMU) and assuming a dominant ground plane we demonstrate that rotation and translation estimation can be decoupled. This result allows us to reduce the number of point matches needed to compute a motion hypothesis. We then derive different algorithms based on this decoupling that allow an efficient estimation. We also demonstrate how these algorithms can be used efficiently to compute an optimal inlier set using exhaustive search or histogram voting instead of a traditional RANSAC step. Our methods are evaluated on synthetic data and on the KITTI data set, demonstrating that our methods are well suited for visual odometry in road driving scenarios.**

I. INTRODUCTION

Visual odometry and visual SLAM [14] play an immensely important role for mobile robotics. Many different approaches for visual odometry have been proposed already, and for a wide variety of applications visual odometry has been used successfully. However, reliability, long-term stability and accuracy of visual odometry algorithms are still a topic of research as can be seen by the many contributions to the KITTI visual odometry benchmark [4]. Most approaches for visual odometry follow the scheme where first feature correspondences between subsequent views are established, then they are screened for outliers and then egomotion estimation is done on inliers only [14]. The reliability and robustness of such a scheme is heavily dependent on the outlier screening step. In addition the outlier screening process has to be fast and efficient. The use of RANSAC [2] is widely accepted for this step.

However, the complexity of the RANSAC process being exponentially related to the minimal number of points necessary for the solution estimation, reducing this number is very interesting. For instance, a standard solution for two-views egomotion estimation is to use the essential matrix with 5 matching points [11] in a RANSAC process to increase the robustness. Nevertheless, the number of points needed for estimating the parameters is really crucial for the RANSAC algorithm. Indeed, the runtime of the RANSAC increases exponentially according to the number of points we need. Thus, before estimating the parameters, we have to be sure that we use the minimal number of points for that. One such idea is to take motion constraints into account, e.g. a planar motion (2pt algorithm) or the Ackermann steering motion for self-driving cars (1pt algorithm [15]). Another idea is to utilize an additional sensor like an inertial measurement unit to improve this step. Traditional sensor fusion methods [16] perform a late fusion of the individual vision and IMU measurements. However, it is possible to utilize the IMU measurements much earlier to aid the visual odometry algorithm for outlier screening. This idea has already been utilized in [3], [8], [10], [13] in which partial IMU measurements have been used to design more efficient motion estimation algorithms for outlier screening.

In this paper we follow this idea by proposing a low complexity algorithm for unconstrained two-view motion estimation that can be used for efficient outlier screening and initial motion estimation. Our method assumes a known gravity vector (measured by an IMU) and is based on a homography relation between two views. In [13] a 2pt algorithm has been proposed exactly for this case. In this work we will improve on [13] and show that actually an algorithm can be found that needs fewer than 2 data points for a motion hypothesis. To achieve this, the first step is to separate the rotation and translation estimation. This is possible if the scene contains features that are far away. Such features are only influenced by rotation and only the x-coordinate of a single feature point is sufficient to find the remaining rotational degree of freedom (DOF), so we call this the 0.5pt method. After this the remaining 3 DOFs for the translation t_x, t_y, t_z are computed. We present a linear solution that needs 1.5pt correspondences. However, more important is our proposal of using a discrete sampling for determining one of the remaining parameters and then use a 1pt algorithm for the remaining 2 parameters. This makes it possible to completely determine a motion hypothesis from a single point correspondence. Thus, we obtain an extremely fast algorithm even within a RANSAC loop. The actual motion hypotheses can be computed exhaustively for each point correspondence and the best solution can be found by a voting scheme.

The proposed methods are evaluated experimentally on synthetic and real data sets. We test the algorithms under different image noise and IMU measurement noise. We demonstrate the proposed algorithms on KITTI [4] data set and evaluate the accuracy compared to the ground truth.

[1]Banglei Guan is with College of Aerospace Science and Engineering, National University of Defense Technology, China. banglei.guan@hotmail.com

[2]Pascal Vasseur is with LITIS, Université de Rouen, France. pascal.vasseur@univ-rouen.fr

[3]Cédric Demonceaux is with Le2i, VIBOT ERL CNRS 6000, Université Bourgogne Franche-Comté, France. cedric.demonceaux@u-bourgogne.fr

[4]Friedrich Fraundorfer is with the Institute of Computer Graphics and Vision, TU Graz, Austria and with the Remote Sensing Technology Institute, German Aerospace Center, Germany. fraundorfer@icg.tugraz.at

These experiments also demonstrate that the assumptions taken hold very well in practice and the results on the KITTI data set show that the proposed methods are useful within the self-driving car context.

II. Related work

With known intrinsic parameters, a minimum of 5 point correspondences is sufficient to estimate the essential matrix [11], and a minimum of 4 point correspondences is required to estimate the homography if all the 3D points lie on a plane [5]. Then the essential matrix or the homography can be decomposed into the motion of the camera between two views, *i.e.* a relative rotation and translation direction.

A reduction of the number of needed point correspondences between views is important in terms of computational efficiency and of robustness and reliability. Such a reduction is possible if some additional information is available or assumptions about the scene and camera motion are taken. If for instance the motion is constrained to be on a plane, which is typical for ground based robots or self-driving cars, 2 point correspondences are only needed for computing the 3-DOFs motion [12]. If further the motion is constrained by Ackermann steering typical for cars only 1 point correspondence is necessary [15]. In contrast if additional information e.g. from an IMU is available and the complete rotation between the two views is provided by the IMU, the remaining translation can be recovered up to scale using only 2 points [7].

Using this concept a variety of algorithms have recently been proposed for egomotion estimation when knowing a common direction [3], [6], [10]. The common direction between the two views can be given by an IMU (measuring the gravity direction) or by vanishing points extraction in the images. All these works propose different algorithms for solving the essential matrix with 3 point correspondences. For this they start with a simplified essential matrix (due to the known common direction) and then derive a polynomial equation system for the solution.

To further reduce the number of point correspondences, the homography relation between two views can be used instead of the epipolar constraint expressed by the essential matrix. Under the assumption that the scene contains a large enough plane that is normal or parallel to the gravity vector measured by an IMU (a typical case for indoor or road driving scenarios) the egomotion can be computed from 2 point correspondences [13]. This idea however can be extended even further which is what we propose in this work.

We start with the formulation of [13] where the cameras are aligned to the gravity vector and the remaining DOFs are one rotation parameter and three translation parameters. We solve for rotation and translation separately. This uses the fact that for far scene points, the parallax-shift (induced by translation) between two views is hardly noticeable. The motion of these far points is close enough to a pure rotation case such that the rotation between two views can be estimated firstly (and independent from translation) using these far points. Every single point correspondence can produce a hypothesis for the remaining rotation parameter which can be used in a 1pt RANSAC algorithm for rotation estimation or for histogram voting by computing the hypothesis from all the point matches. This step also allows to separate the correspondences into two sets, a far set and a near set. The further processing for the translation estimation can then be continued on the smaller near set only, as the effect of translation is not noticeable in the far set. Such a configuration is typical for road driving imagery. For estimating the remaining translation parameters we propose a linear 1.5pt algorithm. However, practically this solution does not give a direct computational advantage over the 2pt algorithm of Saurer et al. [13]. Instead we propose to use a combination of discrete sampling and parameter estimation. The idea is that 1 parameter of the remaining 3 is sampled in discrete steps. For each sampled value it is possible to estimate the remaining parameters from a single point correspondence. This can be then done efficiently using a 1pt RANSAC step or an exhaustive search to find the globally optimal value. The great benefit about this approach is that instead of performing 2pt RANSAC a sequence of 1pt RANSAC steps with a constant overhead for bounded discrete sampling is used. This exhaustive search gives us an efficient way to find the globally optimal solution.

III. Basics and notations

With known intrinsic camera parameters, a general homography relation between two different views is represented as follows [5]:

$$\lambda \mathbf{x}_j = \mathbf{H}\mathbf{x}_i, \tag{1}$$

where $\mathbf{x}_i = [x_i, y_i, 1]^T$ and $\mathbf{x}_j = [x_j, y_j, 1]^T$ are the normalized homogeneous image coordinates of the points in views i and j, λ is a scale factor. The homography matrix $\mathbf{H}$ is given by:

$$\mathbf{H} = \mathbf{R} - \frac{1}{d}\mathbf{t}\mathbf{N}^T, \tag{2}$$

where $\mathbf{R} = \mathbf{R}_y\mathbf{R}_x\mathbf{R}_z$ and $\mathbf{t} = [t_x,\ t_y,\ t_z]$ are respectively the rotation and the translation from views i to j. $\mathbf{R}_y$, $\mathbf{R}_x$ and $\mathbf{R}_z$ are the rotation matrices along y-, x- and z-axis, respectively. With the knowledge of the vertical direction the rotation matrix $\mathbf{R}$ can be simplified such that $\mathbf{R} = \mathbf{R}_y$ by pre-rotating the feature points with $\mathbf{R}_x\mathbf{R}_z$, which can be measured from the IMU (or alternatively from vanishing points [1]). After this rotation, the cameras are in a configuration such that the camera plane is vertical to the ground plane. d is the distance between the view i frame and the 3D plane. $\mathbf{N} = [n_1, n_2, n_3]^T$ is the unit normal vector of the 3D plane with respect to the view i frame.

For this gravity aligned camera configuration the plane normal $\mathbf{N}$ of the ground plane is $[0, 1, 0]^T$. Consequently, Equation 2 that only considers points on the ground plane can be written as:

$$\mathbf{H} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} - \frac{\mathbf{t}}{d}\begin{bmatrix}0\\1\\0\end{bmatrix}^T \tag{3}$$

The homography relation is defined up to scale, so there are 4 DOFs remaining being the rotation around the y-axis and the translation parameters t_x, t_y, t_z.

IV. 0.5PT ROTATION ESTIMATION METHOD

The rotation angle can be computed from a single point correspondence. In fact, it can be computed from only the x-coordinate of a single feature point. Rotation estimation can be done independently from the translation estimation if the scene contains far points. These points, that can be considered at infinity, are not affected any more by a translation. Consequently, the translation component is zero for these points and the Equation 3 simplifies to a rotation matrix:

$$\mathbf{H} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{4}$$

In order to further eliminate the unknown scale factor λ, multiplying both sides of Equation 1 by the skew-symmetric matrix $[\mathbf{x}_j]_\times$, yields the equation:

$$[\mathbf{x}_j]_\times \mathbf{H} \mathbf{x}_i = \mathbf{0}. \tag{5}$$

Substituting Equation 4 into the above equation and expand it:

$$\begin{bmatrix} 0 & -1 & y_j \\ 1 & 0 & -x_j \\ -y_j & x_j & 0 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \mathbf{0} \tag{6}$$

By rewriting the equation, we obtain:

$$-x_i y_j \sin(\theta) + y_j \cos(\theta) - y_i = 0 \tag{7}$$

$$(x_i x_j + 1)\sin(\theta) + (x_i - x_j)\cos(\theta) = 0 \tag{8}$$

$$-y_j \sin(\theta) - x_i y_j \cos(\theta) + x_j y_i = 0 \tag{9}$$

These equations are now derived for the case of points that lie on a plane normal to the y-axis of the cameras and which are infinitely far away. In this case, all the points have to lie on the horizon, which means that the y-coordinate of such a point in normalized coordinates is 0 (normalized coordinates are image coordinates in pixel after multiplication with the inverse calibration matrix). This equates Equation 7 and Equation 9 to zero. Only Equation 8 remains to be used. Considering the trigonometric constraint $\sin^2(\theta) + \cos^2(\theta) = 1$, the rotation parameter $\sin(\theta)$ can be obtained:

$$\sin(\theta) = \pm \frac{x_i - x_j}{\sqrt{{x_i}^2 {x_j}^2 + {x_i}^2 + {x_j}^2 + 1}} \tag{10}$$

Due to the sign ambiguity of $\sin(\theta)$, we obtain two possible solutions for the rotation angle. For every point correspondence a rotation hypothesis can be calculated. A 1pt RANSAC loop can be utilized to find a consistent hypothesis with only a few samples. Alternatively the globally optimal solution can be computed by performing an exhaustive search or histogram voting. The exhaustive search is linear in the number of point correspondences and a hypothesis can be computed for every point correspondence. The hypothesis with the maximum number of inliers is the globally optimal solution. To avoid computing the inliers and outliers for every hypothesis a histogram voting method can be used. For this all the hypothesis are collected in a histogram with discrete bins (e.g. a bin size of 0.1 degree) and the bin with the maximum count is selected as the best solution. Alternatively the mean of a window around the peak can be computed for a more accurate result.

The inliers of the pure rotation formulation belong to scene points that are very far away and don't influence the translation. For further translation estimation these point correspondences can be removed to reduce the number of data points to process. Translation estimation only needs to consider the outlier set of the rotation estimation.

V. TRANSLATION ESTIMATION METHOD

After estimation of the rotation parameter as described in the previous section, the feature points in views j can be rotated by the rotation matrix around the yaw axis:

$$\tilde{\mathbf{x}}_j = \mathbf{R}_y^T \mathbf{x}_j, \tag{11}$$

This aligns both views such that they only differ in translation $\tilde{\mathbf{t}} = [\tilde{t}_x,\ \tilde{t}_y,\ \tilde{t}_z]$ for $\mathbf{x}_i \leftrightarrow \tilde{\mathbf{x}}_j$. Equation 3 therefore is written as:

$$\tilde{\mathbf{H}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{d} \begin{bmatrix} \tilde{t}_x \\ \tilde{t}_y \\ \tilde{t}_z \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}^T \tag{12}$$

In the following subsections we describe 4 different methods of how to estimate the translation parameters.

A. 1.5pt linear solution

This subsection describes a linear method to compute the remaining translation parameters. Three equations from two point correspondences are used to set up a linear equation system to solve for the translation.

The camera-plane distance d is unknown but the translation can be known only up to scale. Therefore d can be absorbed by $\tilde{\mathbf{t}}$. We then obtain:

$$\tilde{\mathbf{H}} = \begin{bmatrix} 1 & -\tilde{t}_x & 0 \\ 0 & 1 - \tilde{t}_y & 0 \\ 0 & -\tilde{t}_z & 1 \end{bmatrix} \tag{13}$$

Substituting Equation 13 into the Equation 5, the homography constraints between $\mathbf{x}_i$ and $\tilde{\mathbf{x}}_j$ can be expressed:

$$\begin{bmatrix} 0 & -1 & \tilde{y}_j \\ 1 & 0 & -\tilde{x}_j \\ -\tilde{y}_j & \tilde{x}_j & 0 \end{bmatrix} \begin{bmatrix} 1 & -\tilde{t}_x & 0 \\ 0 & 1 - \tilde{t}_y & 0 \\ 0 & -\tilde{t}_z & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \mathbf{0} \tag{14}$$

By rewriting the equation, we obtain:

$$\begin{cases} y_i \tilde{t}_y - y_i \tilde{y}_j \tilde{t}_z = y_i - \tilde{y}_j \\ -y_i \tilde{t}_x + \tilde{x}_j y_i \tilde{t}_z = \tilde{x}_j - x_i \\ y_i \tilde{y}_j \tilde{t}_x - \tilde{x}_j y_i \tilde{t}_y = x_i \tilde{y}_j - \tilde{x}_j y_i \end{cases} \tag{15}$$

Even though Equation 15 has three rows, it only imposes two independent constraints on $\tilde{\mathbf{t}}$, because the skew-symmetric matrix $[\tilde{\mathbf{x}}_j]_\times$ has only rank 2. To solve for the

3 unknowns of $\tilde{\mathbf{t}} = [\tilde{t}_x,\ \tilde{t}_y,\ \tilde{t}_z]$, one more equation is required which has to be taken from a second point correspondence. In principle, an arbitrary equation can be chosen from Equation 15, for example, the second and third rows of the first point $\mathbf{x}_i \leftrightarrow \tilde{\mathbf{x}}_j$, and the second row of the second point $\mathbf{x}'_i \leftrightarrow \tilde{\mathbf{x}}'_j$ are stacked into 3 equations in 3 unknowns:

$$\begin{cases} -y_i\tilde{t}_x + \tilde{x}_j y_i \tilde{t}_z = \tilde{x}_j - x_i \\ y_i\tilde{y}_j\tilde{t}_x - \tilde{x}_j y_i \tilde{t}_y = x_i\tilde{y}_j - \tilde{x}_j y_i \\ -y'_i\tilde{t}_x + \tilde{x}'_j y'_i \tilde{t}_z = \tilde{x}'_j - x'_i \end{cases} \tag{16}$$

The linear solution for $\tilde{\mathbf{t}} = [\tilde{t}_x,\ \tilde{t}_y,\ \tilde{t}_z]$ can be obtained by:

$$\begin{cases} \tilde{t}_x = \dfrac{x_i\tilde{x}'_j y'_i + \tilde{x}_j y_i \tilde{x}'_j - \tilde{x}_j\tilde{x}'_j y'_i - \tilde{x}_j y_i x'_i}{y_i\tilde{x}'_j y'_i - \tilde{x}_j y_i y'_i} \\ \tilde{t}_y = \dfrac{y_i\tilde{x}'_j y'_i + y_i\tilde{y}_j\tilde{x}'_j + x_i\tilde{y}_j y'_i - y_i\tilde{y}_j x'_i - \tilde{x}_j y_i y'_i - \tilde{y}_j\tilde{x}'_j y'_i}{y_i\tilde{x}'_j y'_i - \tilde{x}_j y_i y'_i} \\ \tilde{t}_z = \dfrac{x_i y'_i + y_i\tilde{x}'_j - \tilde{x}_j y'_i - y_i x'_i}{y_i\tilde{x}'_j y'_i - \tilde{x}_j y_i y'_i} \end{cases} \tag{17}$$

The translation $\mathbf{t}$ from views i to j can be obtained:

$$\mathbf{t} = \mathbf{R}_y^T\tilde{\mathbf{t}}. \tag{18}$$

For finding the best fitting translation $\tilde{\mathbf{t}}$ a RANSAC step should be used. Here it is however possible to evaluate the full point set. From the estimated rotation and translation parameters an essential matrix can be constructed ($\mathbf{E} = [\mathbf{t}]_\times\mathbf{R}_y$) and the inliers can be tested against the epipolar geometry. This test is not limited to points on the ground plane and the final inlier set contains all scene points. For a most accurate result a non-linear optimization of the Sampson distance on all the inliers is advised. The techniques of constructing the epipolar geometry and performing the non-linear optimization are also applicable to other translation estimation methods.

B. 1pt method by discrete sampling of the relative height change

Translation estimation as explained in the previous section needs 1.5pt correspondences. However, if one of the remaining parameters is known only a single point correspondence is needed for computing the remaining two parameters. This leads to 1pt algorithm for the translation. It is possible to perform a discrete sampling of a suitable parameter within a suitable bounded range. This allows to perform an exhaustive search to find the global optimal solution which produces the highest number of inliers. The time complexity of this exhaustive search is linear in the number of point correspondences if the number of discrete samples is significantly smaller than the number of point correspondences.

In order to use the sampling method, Equation 12 can be written as:

$$\tilde{\mathbf{H}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{\tilde{t}_y}{d}\begin{bmatrix} \tilde{t}_x/\tilde{t}_y \\ 1 \\ \tilde{t}_z/\tilde{t}_y \end{bmatrix}\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}^T \tag{19}$$

In this variant the relative height change over ground $a = \tilde{t}_y/d$ is sampled in discrete steps which leads to an equation system with only 2 unknowns, $b = \tilde{t}_x/\tilde{t}_y$, $c = \tilde{t}_z/\tilde{t}_y$. Only 1pt is needed to compute a solution b and c for a given a. In the same way, we can choose the second and third row from Equation 15 to compute b and c:

$$\begin{cases} b = \dfrac{a\tilde{x}_j y_i + x_i\tilde{y}_j - \tilde{x}_j y_i}{a y_i\tilde{y}_j} \\ c = \dfrac{a b y_i + \tilde{x}_j - x_i}{a\tilde{x}_j y_i} \end{cases} \tag{20}$$

Based on the value of b and c, we can recover $\tilde{\mathbf{t}} = [b, 1, c]^T$ up to scale. Then the estimated translation $\mathbf{t}$ between views i and j is also recovered up to scale by Equation 18.

C. 1pt method by discrete sampling for x-z translation direction

The sampling method in the previous section worked by discretizing the relative height change between two views. As this can be up to scale there is no obvious value for the step sizes and bounds. However, if one is to sample the direction vector of the translation in the x-z plane this represents the discretization of an angle between $0...360$ degrees. In this case a meaningful step size can easily be defined.

For this variant Equation 12 can be written as:

$$\tilde{\mathbf{H}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{\sqrt{\tilde{t}_x^2 + \tilde{t}_z^2}}{d}\begin{bmatrix} \cos(\delta) \\ \tilde{t}_y/\sqrt{\tilde{t}_x^2 + \tilde{t}_z^2} \\ \sin(\delta) \end{bmatrix}\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}^T \tag{21}$$

The translation direction can be represented as an angle δ and e.g. can be sampled in steps of 1 degree from $0°$ to $360°$, which leads to an equation system with only 2 unknowns, $a = \sqrt{\tilde{t}_x^2 + \tilde{t}_z^2}/d$, $b = \tilde{t}_y/\sqrt{\tilde{t}_x^2 + \tilde{t}_z^2}$. Only 1pt is needed to compute a solution a and b for a given angle δ.

In the same way, we can choose the second and third row from Equation 15 to compute a and b:

$$\begin{cases} a = \dfrac{\tilde{x}_j - x_i}{\tilde{x}_j y_i \sin(\delta) - y_i\cos(\delta)} \\ b = \dfrac{a y_i\tilde{y}_i\cos(\delta) + \tilde{x}_j y_i - x_i\tilde{y}_j}{a\tilde{x}_j y_i} \end{cases} \tag{22}$$

Based on the vector $[\cos(\delta), b, \sin(\delta)]^T$, we can recover $\tilde{\mathbf{t}}$ up to scale. Then we obtain $\mathbf{t}$ by Equation 18.

D. 1pt method by discrete sampling of the in-plane scale change

The method described in this section is another variant of choosing a meaningful parameter for discrete sampling. For easy explanation of this idea one should imagine a camera setup with downward looking cameras with a camera plane parallel to the ground plane. The previously aligned camera setup with the camera plane normal to the ground plane can easily be transformed into such a setup by rotating the feature points about $90°$ around the x-axis of camera by multiplication with a rotation matrix $\mathbf{R}_d$.

Moving a camera looking downwards at a plane (e.g. the street) up and down results in an in-plane scale change of

the image, i.e. the points will move inwards to or outwards from the center. The scale change directly corresponds to the effects of a translation in z-direction. This makes the scale change a good parameter for discrete sampling, as the in-plane scale change can be expressed in pixel distances.

In this approach discrete values for the scale change are sampled and the remaining translation direction in the x-y plane can be computed for every point correspondence from a single point. Points lying on the same plane, have exactly the same translation shift for all the feature matches. Now, we derive the formula in detail.

We assume that $(x_i, y_i, 1) \leftrightarrow (x_j, y_j, 1)$ are the normalized homogeneous image coordinates of the points in downward looking views i and j. The heights of the downward looking views i and j are h_i and h_j, respectively. The ground points can be represented in the camera coordinate system of views i and j: $\mathbf{X}_i = h_i * [x_i, y_i, 1]^T$, $\mathbf{X}_j = h_j * [x_j, y_j, 1]^T$. The translation between views i and j can be computed directly:

$$\tilde{\mathbf{t}}_d = h_j \begin{bmatrix} x_j \\ y_j \\ 1 \end{bmatrix} - h_i \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} (x_j h_j - x_i h_i) \\ (y_j h_j - y_i h_i) \\ h_j - h_i \end{bmatrix} \quad (23)$$

We set the in-plane scale $\kappa = f * (h_i/h_j)$, f is the focal length. We substitute κ into the above equation. We can obtain the translation vector directly as the difference of the image coordinates:

$$\tilde{\mathbf{t}}_d = \begin{bmatrix} f x_j - \kappa x_i \\ f y_j - \kappa y_i \\ f - \kappa \end{bmatrix} \quad (24)$$

By sampling the in-plane scale κ, we can compute the translation vector $\tilde{\mathbf{t}}_d$ using one point, and choose the solution which has the maximum number of inliers. This sampling interval is defined in pixels and allows setting a meaningful step size (e.g. 1 pixel).

The final translation $\mathbf{t}$ between two views can be obtained:

$$\mathbf{t} = \mathbf{R}_y^T \mathbf{R}_d^T \tilde{\mathbf{t}}_d \quad (25)$$

VI. Experiments

We validate the performance of the proposed methods using both synthetic and real scene data. The experiments with synthetic scenes will demonstrate the behavior of our derivations in the presence of image noise and IMU noise. The experiments using the KITTI data set [4] will demonstrate the suitability of the methods for use in road driving scenarios. This experiments will also demonstrate that the assumptions taken will hold for real scenarios.

A. Experiments with synthetic data

To evaluate the algorithms on synthetic data we choose the following setup. The distance of the ground to the first camera center is set to 1. The baseline between two cameras is set to be 0.2 and the direction is either along the x-axis of the first camera (sideways) or along the z-axis of the first camera (forward). Additionally, the second camera is rotated around every axis, three rotation angles varies from $-90°$ to $90°$. The Roll angle (around z-axis) and Pitch angle (around x-axis) are known. The generated scene points can be set to lie on the ground plane or be distributed freely in space.

We evaluate the accuracy of the presented algorithms on synthetic data under different image noise and IMU noise. The focal length is set to 1000 pixels. The solutions for relative rotation and translation are obtained by RANSAC or histogram voting. We assess the rotation and translation error by the root mean square error of the errors. We report the results on the data points within the first two intervals of a 5-quantile partitioning[1] (Quintile) of 1000 trials. The proposed methods are also compared against the 2pt method [13].

In all of the experiments, we compare the relative rotation and translation between views i and j separately. The error measure compares the angle difference between the true rotation and estimated rotation. Since the estimated translation between views i and j is only known up to scale, we compare the angle difference between the true translation and estimated translation. The errors are computed as follows:

- Rotation error: $\xi_{\mathbf{R}} = \arccos((Tr(\mathbf{R}_{gt}\mathbf{R}^T) - 1)/2)$
- Translation error: $\xi_{\mathbf{t}} = \arccos((\mathbf{t}_{gt}^T \mathbf{t})/(\|\mathbf{t}_{gt}\| \, \|\mathbf{t}\|))$

$\mathbf{R}_{gt}$, $\mathbf{t}_{gt}$ denote the ground-truth transformation and $\mathbf{R}$, $\mathbf{t}$ are the corresponding estimated transformations.

1) Pure planar scene setting ("PLANAR"): In this setting the generated scene points are constrained to lie on the ground plane. The scene points consist of two parts: near points (0 to 5 meters) and far points (5 to 500 meters). Both parts have 200 randomly generated points.

Figure 1(a) and (b) show the results of the 0.5pt method and histogram voting for rotation estimation for gradually increased image noise levels with perfect IMU data. It is interesting to see that our method performs better for forward motion than for sideways motion. The histogram voting has an higher error, because of the binning which has more effect than the image noise. Figure 1 (b) does not show a clear trend with increased image noise levels. It seems that the influence of the sideways motion is stronger than the influence of the image noise. Figure 1(c)-(f) show the influence of increasing noise on the IMU data while assuming image noise with 0.5 pixel standard deviation.

Figure 2 shows the results of the 1.5pt linear solution method, 1pt method by sampling for x-z translation direction and 2pt method, for gradually increased image noise levels with perfect IMU data or increasing noise on the IMU data while assuming image noise with 0.5 pixel standard deviation. Note that we use the histogram voting method to compute the rotation first, in order to compare the accuracy of different translation estimation methods. The 1.5pt linear solution method and the 1pt method by sampling for x-z translation direction are robust to the increased image noise and IMU data noise.

2) Mixed scene setting ("MIXED"): In this experiment not all the scene points were constrained to lie on the ground plane. Far points (from 5 to 500 meters distance) do not lie

[1] k-quantiles divide an ordered dataset into k regular intervals

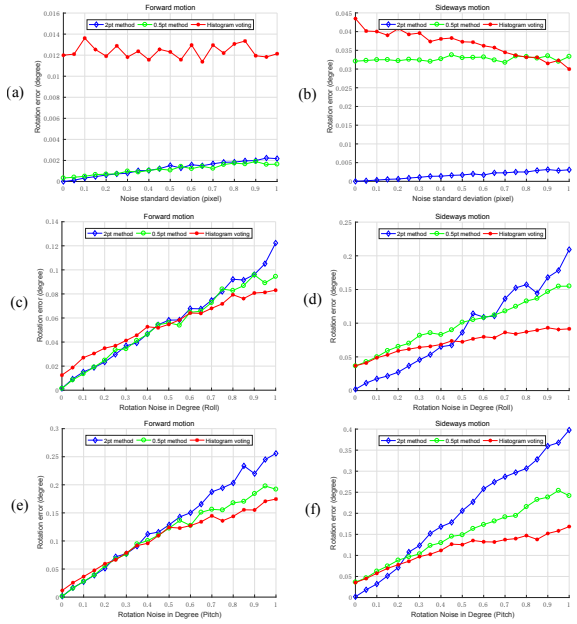


Fig. 1. Rotation error with "PLANAR" setting: Evaluation of the 0.5pt method, histogram voting and 2pt method. Left: forward motion, right: sideways motion. (a) and (b) are with varying image noise. (c), (d), (e) and (f) are under different IMU noise and constant image noise 0.5 pixel standard deviation. (c) and (d) are with Roll angle noise. (e) and (f) are with Pitch angle noise.

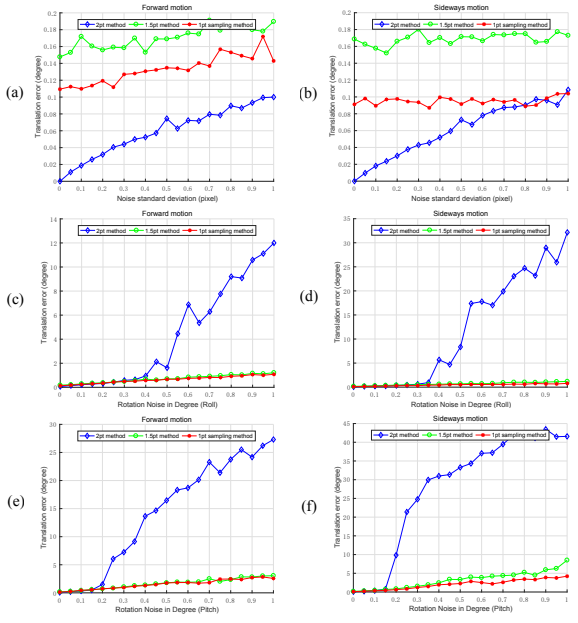


Fig. 2. Translation error with "PLANAR" setting: Evaluation of the 1.5pt linear solution method, 1pt method by sampling for x-z translation direction and 2pt method. Left: forward motion, right: sideways motion. (a) and (b) are with varying image noise. (c), (d), (e) and (f) are under different IMU noise and constant image noise 0.5 pixel standard deviation. (c) and (d) are with Roll angle noise. (e) and (f) are with Pitch angle noise.

on the ground only. They are generated a heights that vary from 0 to 10 meters. The near points however (from 0 to 5 meters distance) are constrained to lie on the ground plane. Both sets, near points and far points consist of 200 randomly generated points.

Figure 3 shows the results of the 0.5pt method and histogram voting for rotation estimation for gradually increased image noise levels with perfect IMU data, or increasing noise on the IMU data while assuming image noise with 0.5 pixel standard deviation.

Figure 4 shows the results of the 1.5pt linear solution method, 1pt method by sampling for x-z translation direction and 2pt method, for gradually increased image noise levels with perfect IMU data or increasing noise on the IMU data while assuming image noise with 0.5 pixel standard deviation. In Figure 4 (f) a sensitivity of the 1.5pt linear solution method to noise in Pitch angle can be seen in sideways motion.

The experiments on synthetic data validate the derivation of the minimal solution solvers and quantify the stability of the solvers with respect to image noise and IMU noise. The synthetic data did not contain outliers. The use of the methods within a RANSAC loop for outlier detection is part of the experiments using real data.

B. Experiments on real data

Experiments on real data were performed on the KITTI data set [4]. For the evaluation we utilized all the available 11 sequences which have ground truth data (labeled from 0 to 10 on the KITTI webpage) and together consist of around 23000 images. The KITTI data set provides a challenging environment for our experiments, however, such a road driving scenario does fit our method very well. In all the images a large scene plane is visible (the road) and features at far distances are present as well. For our experiments we performed SIFT feature matching [9] between consecutive frames. The ground truth data of the sequences is used to pre-rotate the feature-points by $\mathbf{R}_x\mathbf{R}_z$, basically simulating IMU measurements. Then the remaining relative rotation and translation are estimated with our methods. We perform 3 sets of experiments with the KITTI data set. In a first experiment we test the effectiveness of our proposed quick test for rotation inliers. In the second experiment we compute rotation and translation using all our proposed methods and compare it to the ground truth. We also compare the results to the 5pt method [11] and 2pt method [13]. In a third experiment we test the quality of the inlier detection by using the different methods.

1) Rotation estimation inlier selection using y-coordinate test: The 0.5pt method for rotation estimation is working under the assumption that a scene point is far away. Using RANSAC or histogram voting the inliers of this assumption can be found. However, even before computing rotation hypothesis with the 0.5pt method the point correspondences can already be checked if they stem from far points, to remove all the near points. For a far point in this setting the y coordinate of a point feature does not change. So any point correspondence where the y coordinate changes is a near feature that can be discarded for the rotation estimation. This allows to significantly remove a big part of feature points for the rotation estimation to make it more efficient. Table I shows how many of the feature points

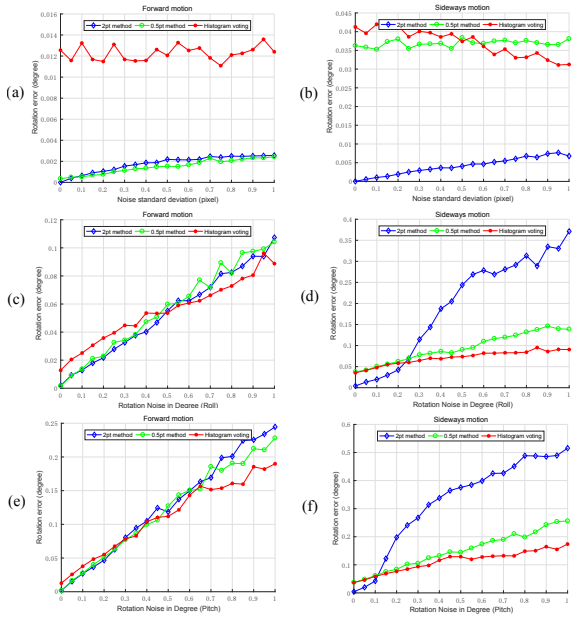


Fig. 3. Rotation error with "MIXED" setting: Evaluation of the 0.5pt method, histogram voting and 2pt method. Left: forward motion, right: sideways motion. (a) and (b) are with varying image noise. (c), (d), (e) and (f) are under different IMU noise and constant image noise 0.5 pixel standard deviation. (c) and (d) are with Roll angle noise. (e) and (f) are with Pitch angle noise.

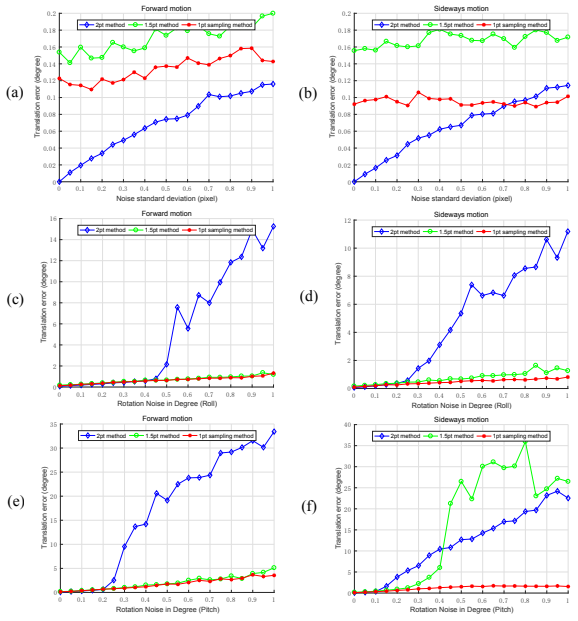


Fig. 4. Translation error with "MIXED" setting: Evaluation of the 1.5pt linear solution method, 1pt method by sampling for x-z translation direction and 2pt method. Left: forward motion, right: sideways motion. (a) and (b) are with varying image noise. (c), (d), (e) and (f) are under different IMU noise and constant image noise 0.5 pixel standard deviation. (c) and (d) are with Roll angle noise. (e) and (f) are with Pitch angle noise.

TABLE I

EFFECT OF THE Y-COORDINATE TEST FOR OUTLIER REMOVAL.

Sequences	$NumSIFT$	$NumRemove$	$Ratio$
00 (4541 images)	634	463	74.15%
01 (1101 images)	398	213	52.26%
02 (4661 images)	648	508	78.74%
03 (801 images)	886	572	65.50%
04 (271 images)	561	421	74.83%
05 (2761 images)	672	450	69.82%
06 (1101 images)	539	383	71.32%
07 (1101 images)	750	446	63.97%
08 (4071 images)	658	460	71.65%
09 (1591 images)	581	434	75.27%
10 (1201 images)	652	454	73.35%

TABLE II

ROTATION ERROR FOR KITTI SEQUENCES [DEGREES].

Seq.	0.5pt method	Histogram voting	5pt	2pt
00	0.060	0.051	0.13	0.24
01	0.073	0.063	0.12	0.23
02	0.068	0.057	0.12	0.26
03	0.068	0.057	0.11	0.21
04	0.074	0.030	0.11	0.17
05	0.049	0.032	0.11	0.21
06	0.073	0.050	0.11	0.21
07	0.052	0.034	0.11	0.19
08	0.051	0.037	0.12	0.19
09	0.079	0.094	0.12	0.24
10	0.059	0.048	0.12	0.22

can be removed already based on this simple criteria. For this test a feature was classified as a far feature if the y coordinate does not change more than 1 pixel. $NumSIFT$ is the average number of point correspondences within a sequence, $NumRemove$ is the average number of outliers removed, and $Ratio = NumRemove/NumSIFT$ is the average of the percentage of the removed outliers. It can be seen that at least more than 52% feature matches can be removed due to this criteria.

2) Comparison of rotation and translation estimation to ground truth: In this experiment we compare the rotation and translation estimates of our methods to ground truth and also to the results of the 5pt method [11] and the 2pt method [13]. Table II lists the results of the rotation estimation and Table III lists the results for the translation estimation. In this experiment the relative rotations and translations between two consecutive images are compared to the ground truth relative poses. The tables show the median error for each individual sequence. For rotation estimation the RANSAC variant and the histogram voting scheme was tested. For the RANSAC variant a fixed number of 100 iterations with an inlier threshold of 2 pixels has been used. For the histogram voting a rotation hypothesis is computed for every point correspondence exhaustively and entered into a histogram. The rotation value at the peak in the histogram is then selected.

Both 0.5pt method and histogram voting method provide better results than the 5pt method and 2pt method. The histogram voting method is slightly more accurate than the 0.5pt method. In subsequent experiments, we use the histogram voting method to estimate the rotation first, then estimate the translation using the different methods.

The translation error for all sequences is shown in Table III. All the four methods for translation estimation, the 1.5pt linear solution method (1.5pt_lin), the 1pt method by discrete sampling of the relative height change (1pt_h), the

1pt method by discrete sampling for x-z translation direction (1pt_d) and the 1pt method by discrete sampling of the in-plane scale change (1pt_s) are compared to ground truth. The 1.5pt method is used within a RANSAC loop with a fix number of 100 iteration and an inliers threshold of 2 pixel. For the 1pt methods an exhaustive search is performed and the solution with the highest number of inliers is used.

The table shows that all of our methods provide better results than the 2pt method. The 1pt methods for translation estimation and the 5pt method are more accurate than the linear solution using 1.5pt. The 1pt_d method offers the best overall performance among all the translation estimation methods.

TABLE III

TRANSLATION ERROR FOR KITTI SEQUENCES [DEGREES].

Seq.	1.5pt_lin	1pt_h	1pt_d	1pt_s	5pt	2pt
00	4.23	1.90	1.64	1.58	1.93	8.03
01	7.34	2.01	1.18	1.20	1.41	10.74
02	3.68	1.83	1.53	1.54	1.53	6.47
03	4.69	2.13	1.88	2.58	2.12	8.61
04	2.64	0.95	0.88	0.92	1.19	5.45
05	3.92	1.57	1.37	1.34	1.67	7.90
06	4.02	1.27	1.20	1.12	1.37	5.57
07	4.89	2.20	1.82	1.89	2.37	10.09
08	4.23	2.17	1.86	1.84	2.06	7.41
09	4.20	2.04	1.53	1.53	1.54	7.20
10	3.90	1.78	1.61	1.58	1.73	7.39

3) Inlier recovery rate: The main usage for our proposed algorithms should be to efficiently find a correct inlier set which can then be used for accurate motion estimation using e.g. non-linear optimization (maybe also using our motion estimates as initial value). We therefore perform an experiment that tests how many of the real inliers (calculated from the ground truth) can be found by our methods. This inlier recovery rate is shown in Table IV as an average over all sequences (an inlier threshold of 2 pixels). All of our four methods can be used to find a correct inlier set, and provide a more complete inlier set than the 2pt method. The inlier recovery rate of 1pt_d method is slightly better than the 5pt method. Inlier detection using the 1pt_d method is shown in Figure 5.

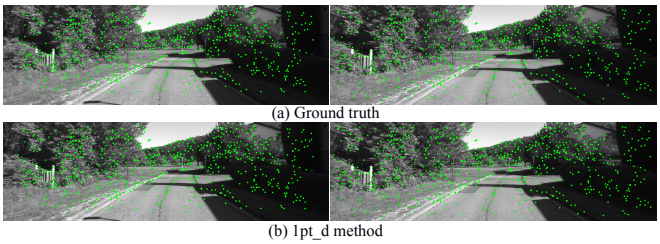
TABLE IV

INLIER RECOVERY RATE FOR ALL KITTI SEQUENCES.

Seq.	1.5pt_lin	1pt_h	1pt_d	1pt_s	5pt	2pt
all	88.47%	96.97%	98.29%	96.51%	98.27%	84.37%

VII. CONCLUSION

The presented algorithms allow to compute motion estimation and inlier sets by exhaustive search or by histogram voting. This is an interesting alternative to the traditional RANSAC method. RANSAC finds an inlier set with high probability but there is no guarantee that it is really a good one. Also, our experiments demonstrate that the assumptions taken in these algorithms are commonly met in road driving scenes (e.g. the KITTI data set), which could be a very interesting application area for it.



(a) Ground truth

(b) 1pt_d method

Fig. 5. Inlier detection example, left: previous frame; right: current frame. (a). Ground truth inliers: 885 matches; (b). Inliers detected by the 1pt_d method: 884 matches.

ACKNOWLEDGMENT

This work has been partially funded by CopTer Project of Grands Réseaux de Recherche Haut-Normands.

REFERENCES

[1] J.C. Bazin, C. Demonceaux, P. Vasseur, and I.S. Kweon. Motion estimation by decoupling rotation and translation in catadioptric vision. *Computer Vision and Image Understanding*, 114(2):254 – 273, 2010. Special issue on Omnidirectional Vision, Camera Networks and Non-conventional Cameras.

[2] M. A. Fischler and R. C. Bolles. RANSAC random sampling concensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of ACM*, 26:381–395, 1981.

[3] Friedrich Fraundorfer, Petri Tanskanen, and Marc Pollefeys. A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. In *Proc. 11th European Conference on Computer Vision*, pages 1–14, 2010.

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[5] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge, 2000.

[6] Mahzad Kalantari, Amir Hashemi, Franck Jung, and Jean-Pierre Guédon. A new solution to the relative orientation problem using only 3 points and the vertical direction. *Journal of Mathematical Imaging and Vision*, 39(3):259–268, 2011.

[7] Laurent Kneip, Margarita Chli, and Roland Siegwart. Robust real-time visual odometry with a single camera and an imu. In *British Machine Vision Conference (BMVC), Dundee, August 2011*, pages 1–11, 2011.

[8] Gim Hee Lee, Marc Pollefeys, and Friedrich Fraundorfer. Relative pose estimation for a multi-camera system with known vertical direction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[9] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[10] O. Naroditsky, X. S. Zhou, J. Gallier, S. I. Roumeliotis, and K. Daniilidis. Two efficient solutions for visual odometry using directional correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):818–824, April 2012.

[11] D. Nistér. An efficient solution to the five-point relative pose problem. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin*, pages II: 195–202, 2003.

[12] Diego Ortin and JMM Montiel. Indoor robot motion based on monocular images. *Robotica*, 19(3):331–342, 2001.

[13] O. Saurer, P. Vasseur, R. Boutteau, C. Demonceaux, M. Pollefeys, and F. Fraundorfer. Homography based egomotion estimation with a common direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):327–341, Feb 2017.

[14] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial] part1: The first 30 years and fundamentals. *Robotics Automation Magazine, IEEE*, 18(4):80 –92, dec. 2011.

[15] Davide Scaramuzza, Friedrich Fraundorfer, and Roland Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In *2009 IEEE International Conference on Robotics and Automation*, pages 1–7, 2009.

[16] S Weiss and R Siegwart. Real-time metric state estimation for modular vision-inertial systems. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.