This ECCV 2018 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ECCV 2018
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/eccv

Stereo relative pose from line and point feature triplets

Alexander Vakhitov[1], Victor Lempitsky[1], and Yinqiang Zheng[2]

[1] Skoltech, Moscow, Nobelya Ulitsa 3, 121207, Russia
{a.vakhitov, lempitsky}@skoltech.ru
[2] NII, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
yqzheng@nii.ac.jp

Abstract. Stereo relative pose problem lies at the core of stereo visual odometry systems that are used in many applications. In this work we present two minimal solvers for the stereo relative pose. We specifically consider the case when a minimal set consists of three point or line features and each of them has three known projections on two stereo cameras. We validate the importance of this formulation for practical purposes in our experiments with motion estimation. We then present a complete classification of minimal cases with three point or line correspondences each having three projections, and present two new solvers that can handle all such cases. We demonstrate a considerable effect from the integration of the new solvers into a visual SLAM system.

Keywords: minimal solver, stereo visual odometry, generalized camera, relative pose, line features

1 Introduction

Minimal solvers in computer vision are used to generate camera motion hypotheses from required minimal sets of feature correspondences, e.g. five feature point correspondences for single camera relative pose estimation [1]. Such solvers are mostly used as a source of motion hypotheses inside a RANSAC loop [2]. They are useful in providing initialization for the optimization procedures at the core of state-of-the-art SLAM systems [3]. For many pose estimation problems, such solvers have already been developed and are extensively used, e.g. to create large-scale structure from motion reconstructions involving thousands of images [4]. It is important to develop minimal solvers taking line segment correspondences as input in addition to points. As recent works demonstrated [5, 6], the use of line segment features can considerably improve accuracy and robustness of visual SLAM and structure from motion systems.

The work is funded by the Russian MES grant RFMEFI61516X0003; a part of this work was finished when Alexander Vakhitov was visiting the National Institute of Informatics (NII), Japan, funded by the NII MOU/Non-MOU International Exchange Program.

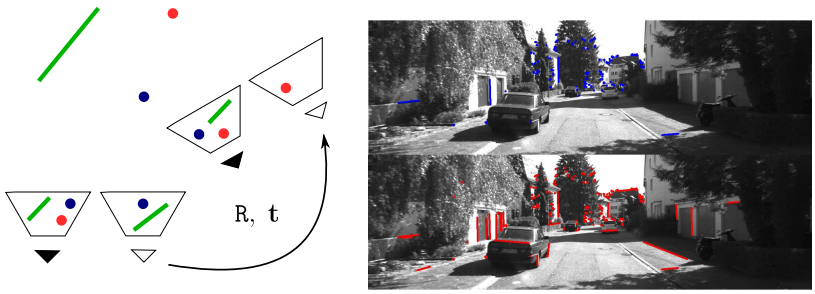

Fig. 1. Left:using three line or point features, each having exactly three projections, we seek to determine the relative pose of the two stereoviews. Right: the use of three-view matches (bottom) by the proposed solvers results in higher number of inliers compared to the use of four-view matches (top). We show the projections of the inlier correspondences on one of the images of KITTI sequence 0 chosen by the method Pradeep[8] using four-view matches (top) and by the proposed EpiSEgo solver using three-view matches (bottom).

To the best of our knowledge, there is no minimal solver for stereo camera relative pose estimation which is efficient enough for real-time use and does not rely on simplifying assumptions limiting its applicability. Thus, [7] is computationally heavy for real-time use, [8] is non-minimal and [9] employs an approximate rotation model that is valid only for small rotations. In this work, we describe two solvers that aim to close this gap, giving an efficient minimal solution to the stereo camera egomotion from three feature triplets. We assume that there are two stereo cameras with projection matrices $\mathtt{P}_{1,1} = [\mathbf{I},\ \mathbf{0}]$, $\mathtt{P}_{1,2} = [\mathbf{I},\ \mathbf{b}]$ for the first camera, $\mathtt{P}_{2,1} = [\mathbf{R},\ \mathbf{t}]$ and $\mathtt{P}_{2,2} = [\mathbf{R},\ \mathbf{b}+\mathbf{t}]$ for the second one, where the baseline $\mathbf{b}$ is known. The goal of the solvers is to find $\mathbf{R}$ and $\mathbf{t}$. In each case, we use three feature triplets, where each triplet is a set of three[3] projections of a 3D line or a 3D point computed using $\{\mathtt{P}_\alpha, \mathtt{P}_\beta, \mathtt{P}_\gamma\}$, $\alpha \neq \beta \neq \gamma$.

While the ability to use features with three rather than four known projections may seem unnecessary for a stereo system, we show that such ability actually provides considerable benefits. To illustrate this, we made a motivation experiment using the first sequence of the KITTI Odometry dataset [11]. We use ORB [12] and LBD [13] features and matched them between neighboring frames and across stereo-views. We then use the provided ground truth poses to estimate the ratio of inlier matches. We observe that for ORB matches the ratio of inlier matches across triplets of views is greater than those across quadruplets (0.121 vs 0.077). For LBD line matches, the advantage is even greater (0.019 vs 0.005). The advantage of relying on triplet matches is further corroborated

[3] In the presence of two-view correspondences only, the overlapped stereo can be regarded as a non-overlapped stereo, and some solutions have been proposed as in [10, 9, 7]. We exclude this case from consideration because the major focus of this work is on overlapped stereo systems.

in our experiments. We develop two solvers covering any combinations of the point/line correspondences among the two view pairs. The first solver delivers 16 solutions, which is equal to the degree of the corresponding algebraic variety. It is impossible to obtain a solver for the formulated equations with the smaller number of solutions. The second solver outputs 32 solutions but is computationally simpler. Both are novel: to the best of our knowledge, no prior work describes a solution to the stereo camera relative pose problem for any combination of line/point features with three projections or even only for the point features.

Experiments show that our solvers are numerically stable and computationally efficient. More interestingly, by using point and line features simultaneously, our solvers work reliably for real scenarios. The use of three-view correspondences allows increasing the inlier cardinality and ratio, which not only facilitates the RANSAC procedure, but also reduces the risk of drifting in the case of long trajectories.

To summarize, we make the following contributions. Firstly, we systematically explore the stereo ego-motion estimation problem in the case of a minimal set of three point and line features with three correspondences. Secondly, we develop new minimal solvers, which output a minimal number of solutions, and demonstrate the increase in accuracy and robustness of stereo egomotion estimation on simulated and real data.

In Section 2, we review the most closely related works on ego-motion estimation. In Section 3, we show the problem formulation and the complete categorization of the minimal point and line sets in any three views. We present the experiment results in Section 4.

2 Related work

Non-overlapping fields of view: To increase the coverage of the field of view (FoV) and to decrease the costs as much as possible, it became popular in recent years to use multiple cameras without overlapped FoVs. The generalized relative pose method proposed in [7] can be applied to estimate the relative pose of such multicamera systems, however it returns up to 64 solutions and is too computationally expensive for real-time use. To solve the problem in real-time, authors introduce certain approximations, e.g. Kneip and Li [10] proposed to use non-minimal point sets and developed an approximated iterative optimization method, whose running speed is inappropriate for realtime applications. For acceleration, Ventura et al. [9] linearized the rotation between two consecutive time frames, so the solver does not apply in the general visual odometry setting.

Overlapping fields of view: Binocular stereo systems with partially overlapping FoVs are preferable in terms of system calibration and metric reconstruction. To estimate the ego-motion of an overlapping stereo rig, Nister et al. [14] proposed to use three points or two lines matched across all four views via triangulation. Chandraker et al. [15] showed that the triangulation of four-view correspondences for ego-motion estimation is unstable, especially when the baseline

is small. They proposed instead to use three four-view line correspondences. Pradeep and Lim [8] used assorted point and line features and developed several minimal solvers for any point and line combinations, as long as these features are simultaneously visible in all four views. Clipp et al. [16] used point features in a mixed number of views, and Dunn et al. [17] used similar input data and accelerated the solving speed by using the constraints in proper ways. Discarding the correspondences without projections onto both views of one stereo camera, one can use generalized absolute pose solvers [18–23]. To summarize, no prior work addresses stereo relative pose problem for three features with three projections. Most of the studies consider the case of 4-view correspondences of only point features.

3 Stereo egomotion solvers

Case	Sect.	Example			
		1^{st} cam		2^{nd} cam	
S3P	3.5	a, b, c	a, b, c	a	b, c
S2P1L	3.5	a, b, ξ	a, b, ξ	a, ξ	b
S1P2L	3.5	a, ξ, θ	a, ξ, θ	a	ξ, θ
S3L	3.4	ξ, θ, γ	ξ, θ, γ	ξ, θ	γ
S2L-1L	3.4	ξ, θ	ξ, θ, γ	ξ, θ, γ	γ
S2P-1L	3.3	a, b	a, b, ξ	a, ξ	b, ξ
S1P1L-1P	3.3	a, ξ	a, ξ, b	a, ξ, b	b
S1P-2L	3.3	a, ξ	a, θ	a, ξ, θ	ξ, θ
S1P1L-1L	3.3	a, ξ, θ	a, ξ	a, θ	ξ, θ
S2P-1P	3.3	a, b, c	a, b	a, c	b, c
S1P-1P1L	reduces to S1P1L-1P				
S1P-2P	reduces to S2P-1P				

Table 1. The table enumerates all possible cases (excluding symmetries) and points to the section that discusses each case. Latin letters are for points and Greek are for lines (the details of the notation are discussed in the beginning of section 3.2).

We address the problem of feature-based relative pose estimation for the binocular stereo camera, assuming that each line or point has exactly three projections. The minimal set in this case consists of three features. Trifocal tensors provide a way to formulate constraints for the line and point features arising from three perspective views. Using the translation and rotation parameterizations (1), (2) which are explained below, these trifocal constraints become third-order equations. While for each line feature there are two such equations, for every point feature nine equations are obtained [24], of which only two are linearly independent. This effect complicates the solver construction.

At the same time, in our problem formulation, for each feature there always exists a stereo camera such that the feature is projected onto both of its views

(the *main camera*). This simplifies the problem and allows to use projection constraints or two-view epipolar constraints between each view of the main camera and the view of the other camera. We use these approaches below and show that we can obtain 16 or 8 solutions using the proposed solvers, compared to 64 solutions using the solver [7] for the same problem.

3.1 Problem

We assume that there are two binocular rectified and calibrated stereo cameras with the same known baseline. We are given a set of triplet feature correspondences. Each correspondence is a triplet. For point feature, a triplet is $(\mathbf{x}_{i_1,\beta_1}, \mathbf{x}_{i_2,\beta_2}, \mathbf{x}_{i_3,\beta_3})$, where $\mathbf{x}_{i,\beta}$ denotes a homogeneous vector of point projection's coordinates onto a view β of a camera i. For a line feature, a triplet is $(\mathbf{l}_{i_1,\beta_1}, \mathbf{l}_{i_2,\beta_2}, \mathbf{l}_{i_3,\beta_3})$ where $\mathbf{l}_{i,\beta}$ denotes a vector of 2D line's coefficients of a 3D line's projection onto a view β of a camera i.

W.l.o.g., we assume that the baseline has unit length ($\mathbf{b} = [1.0.0]^T$) and the projection matrices $\mathtt{P}_{i,\beta}$ for a view β of a camera i are $\mathtt{P}_{1,1} = [\mathtt{I},\ \mathbf{0}]$, $\mathtt{P}_{1,2} = [\mathtt{I},\ \mathbf{b}]$, $\mathtt{P}_{2,1} = [\mathbf{R},\ \mathbf{t}]$ and $\mathtt{P}_{2,2} = [\mathbf{R},\ \mathbf{b} + \mathbf{t}]$. Our goal is to find $\mathbf{R}, \mathbf{t}$.

3.2 Analysis of feature combinations

As long as there are exactly three projections for each feature, we use the following definition.

Definition: *If a feature is projected onto both views of some stereo camera, this camera is called the* **main camera** *for this feature.*

We use the following notation for feature/correspondence combinations. We refer to problem as $S\alpha P\beta L - \gamma P\delta L$ when the first camera is the main for α points and β lines, while the second one is the main for γ points and δ lines. To simplify the analysis, for those combinations having points we assume that the first camera is the main for at least one point feature. Some combinations are reducible to other ones by swapping the first and second cameras.

The categorization of the possible feature combinations is summarized in Tab. 1. For a homogeneous minimal set, there are two possible feature divisions between the cameras: S2L-1L and S3L for lines, or S2P-1P and S3P for points. If we have two points and one line, we can get only S2P1L, S2P-1L, S1P1L-1P cases. For one point and two lines, there are S1P2L, S1P1L-1L and S2L-1P cases. No other feature/correspondence combinations are possible.

If all the features have the same main camera (i.e. S3L, S3P, S2P1L, S1L2P), they can be triangulated in the coordinate frame of this camera, and the problem reduces to generalized absolute pose [20] for lines and points known to have 8 possible solutions. If a minimal set consists only of lines (S3L and S2L-1L), it admits a particular straightforward scheme of solution ("easy" cases).

The other situations (S2P-1L, S1P1L-1P, S1P-2L, S1P1L-1L, S2P-1P) are the "hard" cases. They share two common properties: the features have different main cameras and there is at least one point in the feature set. Minimal solvers for them are the main contributions of the paper.

In the next section, we propose two polynomial solver-based approaches for the "hard" cases. After that, we show how the other cases can be reduced to finding the roots of a single eight-degree polynomial, and then a recently proposed method [23] can be used. For the degeneracy analysis, see Supp. Mat.

3.3 "Hard" cases

In this section, we consider the situation when the features have different main cameras and there is at least one point in the minimal set. Without loss of generality, a camera is the first one if the first (and maybe the only) point feature is projected onto both views of this camera. We also assume that it is projected onto the first view of the second camera. We use the first point to express the translation $\mathbf{t}$ in terms of the point's depth and rotation matrix elements, as in [7]. In particular, from an equation describing the point's projection onto the first view of the second camera we get

$$\mathbf{t} = \alpha\mathbf{u} - \mathbf{RS}, \tag{1}$$

where $\mathbf{S}$ is the point's position triangulated in its main camera's coordinates, $\mathbf{u}$ is the homogeneous vector of the point's projection, α is the depth constant. We will denote as $\mathbf{t}_\beta = \delta_{\beta,2}\mathbf{b}$ the translation of the view β w.r.t. the stereo camera coordinate system, where $\delta_{i,j} = 1$ iff $i = j$, else $\delta_{i,j} = 0$. We use the unit quaternion-based rotation parameterization:

$$\mathbf{R} = \begin{bmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2bd+2ac \\ 2bc+2ad & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & a^2-b^2-c^2+d^2 \end{bmatrix}, \tag{2}$$

$$a^2+b^2+c^2+d^2 = 1. \tag{3}$$

We have experimented with two ways of formulation of the polynomial equations for the stereo egomotion problem explained in the following paragraphs.

Solver based on Epipolar/Pluecker constraints. We describe next a solver for the 'hard' cases which uses generalized epipolar constraints as in [7]. If the feature is a point, we analyze the epipolar constraint arising from its projection onto the view β of the first camera and onto the view γ of the second camera. The epipolar line has the equation in homogeneous coordinates $\mathbf{E}_{1,\beta\to 2,\gamma}\mathbf{x}_{1,\beta}$ using the essential matrix $\mathbf{E}_{1,\beta\to 2,\gamma}(\alpha,\mathbf{R}) = [\mathbf{e}_{\beta,\gamma}]_\times\mathbf{R}$ where $\mathbf{e}_{\beta,\gamma} = \mathbf{t} + \mathbf{t}_\gamma\mathbf{b} + \mathbf{R}\mathbf{t}_\beta$, $[\mathbf{a}]_\times$ is a matrix of a cross product with a vector $\mathbf{a}$. Then, the point's projection lies on the epipolar line, which translates to the following constraint:

$$\mathbf{x}_{2,\gamma}^T\mathbf{E}_{1,\beta\to 2,\gamma}(\alpha,\mathbf{R})\,\mathbf{x}_{1,\beta} = 0. \tag{4}$$

For the point feature, we will get two constraints of the form (4) with the unknowns $\mathbf{R}$ and α. Using 3D line's projections onto the views of its main camera j we compute a pair of 3D points lying on the line $\mathbf{X}_1, \mathbf{X}_2$. Assuming that $j = 1$, we get the following expression for the line through projections of the points $\mathbf{X}_1$ and $\mathbf{X}_2$:

$$\lambda\mathbf{l}_{i,\beta} = (\mathbf{R}\mathbf{X}_1 + \mathbf{t} + \mathbf{t}_\beta) \times (\mathbf{R}\mathbf{X}_2), \tag{5}$$

where λ is a scaling parameter. It leads to the following constraint:

$$[\mathbf{l}_{i,\beta}]_\times \mathbf{R}\big((\mathbf{X}_1 + \alpha\mathbf{u} + \mathbf{t}_\beta) \times \mathbf{X}_2 - \mathbf{X}_2 \times \mathbf{t}_\beta\big) = 0. \tag{6}$$

Likewise, we obtain the following constraint for $j = 2$:

$$[\mathbf{l}_{i,\beta}]_\times \mathbf{R}^T\big((\mathbf{X}_1 - \alpha\mathbf{u}) \times \mathbf{X}_2 - \mathbf{X}_2 \times \mathbf{t}_\beta\big). \tag{7}$$

A system of the constraints (4), (6) or (7) can be formulated as

$$\mathtt{A}\mathbf{r} + \alpha\mathtt{B}\mathbf{r} = \mathbf{0}, \tag{8}$$

where $\mathbf{r}$ is a vectorized matrix $\mathbf{R}$, and $\mathtt{A}$ and $\mathtt{B}$ are coefficient matrices.

Substituting the parameterization (2) into (8), we get four equations of degree three w.r.t. a, b, c, d, α and add to them the constraint (3). After formulating these equations over $\mathbb{Z}p$, we find using Maple [25] that the dimension of the quotient ring for the polynomial ideal is 32, see [26] for details. Each term in the equations (8) after substitution of (2) is of degree 2 w.r.t. a, b, c, d. We divide the equations by a^2, and denote $\tilde{b} = b/a$, $\tilde{c} = c/a$, $\tilde{d} = d/a$. We choose a as a divisor because it is close to one if the rotation is not big, which is the typical case for the SLAM systems. Finally, we get the constraints in the vector form:

$$\mathtt{C}(\tilde{b}, \tilde{c}, \tilde{d})[1,\ \alpha]^T = \mathbf{0}, \tag{9}$$

where $\mathtt{C}(\tilde{b}, \tilde{c}, \tilde{d})$ is a 4×2 matrix consisting of second-degree polynomials. All the 2×2 sub-matrices of $\mathtt{C}(\tilde{b}, \tilde{c}, \tilde{d})$ must have zero determinants. It gives six equations of degree four, which we multiply with all the monomials of $\tilde{b}, \tilde{c}, \tilde{d}$ of degree three and obtain 240 equations and then use them to construct an elimination template.

After the LU-decomposition of the template matrix, using the action monomial $\tilde{d}$ to construct an action matrix, we obtain the solutions by eigen-decomposition, find α from the null-space of $\mathtt{C}(\tilde{b}, \tilde{c}, \tilde{d})$, find a using the unit-norm constraint (3) and $\mathbf{t}$ using (1).

Solver based on point projection constraints. For the this solver, we apply the known preprocessing rotation $\tilde{\mathbf{R}}$ to the projections of all the features to the views of the second camera. $\tilde{\mathbf{R}}$ is chosen so that the first point's projection is in the image center: $\mathbf{u} = [0, 0, 1]^T$, see (1) for the definition of $\mathbf{u}$. The baseline vectors of the cameras become different, we denote them as $\mathbf{b}_j$, where $j = 1, 2$ is the stereo camera index, and get $\mathbf{b}_1 = \mathbf{b}$ and $\mathbf{b}_2 = \tilde{\mathbf{R}}\mathbf{b}$.

We define a function $\pi_{1,\beta}(\mathbf{R}, \alpha, \mathbf{X})$ describing the point projection process, which takes a 3D point $\mathbf{X}$ expressed in the first camera's coordinate frame and outputs the homogeneous point projection coordinates to view β of the second camera:

$$\pi_{1,\beta}(\mathbf{R}, \alpha, \mathbf{X}) = \mathbf{R}(\mathbf{X} - \mathbf{S}) + \alpha\mathbf{u} + \mathbf{t}_\beta, \tag{10}$$

which is a standard point projection equation after we substitute the translation according to (1). By noting that the rotation from the second to the first camera

is $\mathbf{R}^T$ and the translation is $-\mathbf{R}^T\mathbf{t} = -\alpha\mathbf{R}^T\mathbf{u} + \mathbf{S}$, using (1), we formulate a similar function $\pi_{2,\beta}(\mathbf{R}, \alpha, \mathbf{X})$ returning a projection of a 3D point $\mathbf{X}$ expressed in the second camera's coordinate frame to a view β of a first camera:

$$\pi_{2,\beta}(\mathbf{R}, \alpha, \mathbf{X}) = \mathbf{R}^T(\mathbf{X} - \alpha\mathbf{u}) + \mathbf{S} + \mathbf{t}_\beta. \tag{11}$$

We assume that the camera j is the main one for the feature, and that the feature also has a projection onto a view β of a camera $i \neq j$. The constraint for the point feature is obtained from $\pi_{i,\beta}(\mathbf{R}, \alpha, \mathbf{X}) = \lambda_x \mathbf{x}_{i,\beta}$ by expressing and substituting the depth parameter λ_x:

$$\pi^{(k)}_{i,\beta}(\mathbf{R}, \alpha, \mathbf{X}_p) - \mathbf{x}^{(k)}_{i,\beta}\pi^{(3)}_{i,\beta}(\mathbf{R}, \alpha, \mathbf{X}_p) = 0, \quad k = 1, 2, \tag{12}$$

where k is the coordinate index of the feature projection, and $\mathbf{X}_p$ is found by triangulation using the point's projections onto the main camera views. The constraint for the line feature is:

$$\mathbf{l}^T_{i,\beta}\pi_{i,\beta}(\mathbf{R}, \alpha, \mathbf{X}_j) = 0, \quad j = 1, 2. \tag{13}$$

Using these constraints and substituting the parameterization (2), we get a system of four equations:

$$\mathtt{D}(a, b, c, d)[1,\ \alpha]^T = \mathbf{0}, \tag{14}$$

where $\mathtt{D}$ is a matrix of second-degree polynomials.

Generating in $\mathbb{Z}p$ the systems for all the possible feature combinations together with a constraint (3) and using Maple [25] we find that the quotient ring dimension and the number of solutions is 16.

From the system (14) by subtracting equations we obtain one or two (S2L-1P) linearly independent second-degree equations free of α. As before, by computing determinants we get fourth-degree equations. The final system consists of six fourth degree equations (or five for SP-2L, because one of the determinants is identically zero), one (or two, for SP-2L) α-free second-degree equations, and a quadratic constraint (3). This system also leads to 16 solutions.

The basis of the remainder quotient ring as a vector space is not the same for different feature combinations. In particular, for the S1P1L-1P and S1P1L-1L cases there is one particular basis, and another one for the combinations S2P-1L, S2P-1P, S2L-1P (see Supp.Mat.).

We solve the obtained system by constructing an elimination template. Denote the second degree equation obtained after subtraction as $f_1 = 0$, the unit norm constraint as $f_2 = 0$, the other equations as $g_i = 0$, $i = 1..6$. We form an equation set F from f_1 multiplied with a^2, f_2 multiplied with ab, ac, b^2, bc, c^2, and f_1, f_2, g_i for $i = 1..6$. We multiply every equation from F by a, b, c, d, then by a, b, c, then by a, b, then by a, and add all the equations obtained after every multiplication operation to a set G of cardinality 975. It allows us to express all the basis monomials times the action variable a. We use LU decomposition and get the action matrix of size 16×16. It is four times smaller than in the case of the Epipolar/Pluecker constraints, so the eigendecomposition can be performed faster, but template construction and LU decomposition will be slower.

3.4 Only line features

The previous analysis is missing two situations: when all the features are lines or when they all have the same main camera. Next, we describe how both situations lead to a second-degree polynomial system in three unknowns, and therefore can be addressed by an already developed method for this type of geometric computer vision problems.

The coefficients of the 3D line's projection coincide with the direction of the normal to the plane through the camera center and the 3D line. If we observe the line from three views, we know normals of three different planes containing the line: $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$, and their triple product is equal to zero: $\mathbf{n}_1 \cdot (\mathbf{n}_2 \times \mathbf{n}_3) = 0$. We are going to use this fact to formulate the constraints as follows:

$$\mathbf{l}_{j,\gamma}^T \mathbf{R}^T (\mathbf{l}_{i,1} \times \mathbf{l}_{i,2}) = 0, \tag{15}$$

for $i = 1, j = 2$ or $i = 2, j = 1$ depending on whether the main camera for the feature is the first or the second one, $\gamma = 1, 2$ is the view number. Three such constraints formulated using (2) result in a second-degree system with 4 unknowns, the fourth equation being the unit norm constraint.

3.5 Single main camera

If all the features have the same main camera, their 3D coordinates w.r.t. this camera can be computed, and then the problem becomes a particular case of the generalized absolute pose problem (gP3P). In the case of three point features several methods are available in this case, e.g. [23]. To the best of our knowledge, there are no papers analyzing the generalized absolute pose problem for the mixed point/line minimal sets.

We propose to use the earlier introduced constraints (12,13) here as well. Without loss of generality, we assume that the first camera is the main one for all the features, so we use the constraints with $\pi_{1,\beta}$ for $\beta = 1, 2$. The depth α enters the system linearly. It can be expressed as a linear combination of terms involving other unknowns. This way we obtain a system of three equations w.r.t the unknown rotation matrix parameters.

In both cases, we transform a system with four unknowns into a system of three quadrics in three unknowns $\tilde{b} = b/a, \tilde{c} = c/a, \tilde{d} = d/a$ by the use of the constraint (3) to remove the zero-order terms and division by a^2.

The recent work [23] provides a framework to which our problem fits well, proposing a way to reduce a problem of three quadrics intersection to root-finding of a single eighth-degree polynomial by using the hidden variable method to construct a single eight-order polynomial w.r.t. b, and we customize the method by adaptively choosing the variable to hide using the condition numbers (see Supp.Mat.).

To sum up, we have considered all possible feature and correspondence combinations up to symmetries. In the cases for three line features or when all features share the same main camera, the method [23] can be applied. The remaining cases are the most difficult and we have proposed two new polynomial

solvers to cope with them. Next, we demonstrate the benefits of using the proposed solvers in synthetic experiments and on real data.

4 Experiments

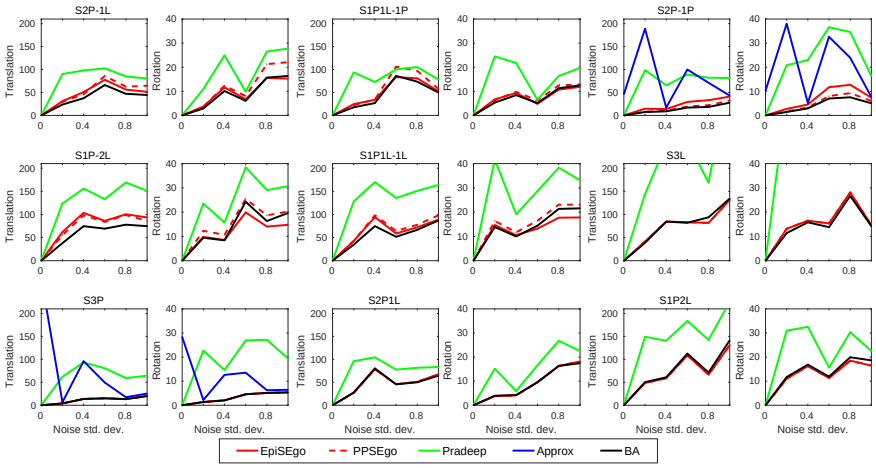

Fig. 2. The effect of additive noise variation on median relative translation and absolute rotation for each feature/correspondence combination. The accuracy of the methods degrades when noise is added. When fewer points are available, the translation error also grows. The new solvers PPSEgo and EpiSEgo are more accurate than the baselines and show almost the same accuracy as bundle adjustment (BA) started from the true solution and fitting to the noisy projections. The decrease in the error with growing noise happens simultaneously for BA and the proposed methods and can be explained by the non-linear nature of dependency between the projections and the SE3 transforms.

4.1 Simulated data

Setup: We perform a number of synthetic experiments to evaluate our method against the non-minimal assorted features solver [8] (*Pradeep*). For point-only configurations, we also compare to an approximate minimal solver for generalized relative pose from points for small rotation [9] (*Approx*). Finally, we evaluate bundle adjustment (*BA*) initialized with a true pose that uses the "gold standard" geometric feature reprojection error. We use BA with oracle initialization as a reference to demonstrate the best realistically achievable accuracy. While our methods and Approx use minimal feature sets, Pradeep needs four projections for every feature. We have re-implemented Pradeep and used the original

code of Approx. We evaluate both of the proposed solvers, namely the epipolar constraint-based (EpiSEgo) solver and the point projection constraint-based solver (PPSEgo).

We assume that the stereo camera is rectified and the baseline is $\mathbf{b} = [1; 0; 0]^T$. We also consider square images with the side of 1000 pixels and the vertical and horizontal view angle of 90°. We fix the first stereo camera at the origin and randomly place the second camera. The points, as well as the 3D endpoints of the lines, are uniformly sampled from the box $B = [-1.5, 2.5] \times [-1.5, 2.5] \times [12, 16]$. The distance between stereo cameras is sampled uniformly from the interval $[1, 10]$. The second camera is rotated with angle uniformly sampled from the interval $[0, 45]^\circ$ and around a random axis direction. If less than seven vertices of B are visible, the pose is resampled. We add Gaussian noise with $\sigma = 0.5$ pixels to the projections of the points and to line segments' endpoints. The lines have the length sampled uniformly from $[0.5, 1.5]$, the line generating process follows [27]. Each experiment consists of 1000 random simulations for each of the possible feature/correspondence combinations.

Results. We compute the median absolute rotation error (in degrees) and relative translation error (in %) for three overlapping sets of feature/correspondence combinations: 'hard' cases, 'easy' cases (i.e. three line features or features sharing the same main camera), and the point-only cases. To check numerical stability, we use zero additive noise and get the median (mean) rotation errors of 2×10^{-9} (5×10^{-7}) degrees for PPSEgo and 8×10^{-7} (2×10^{-3}) degrees for EpiSEgo, which is comparable to errors reported for similar solvers [7]. PPSEgo is thus more numerically stable.

The next experiment (Fig. 4) shows that the difference diminishes when the noise is present. Here we vary σ from 0.0 to 1.0. The errors for all methods increase with the noise level, and the accuracy of the proposed solvers is close to the reference one of the BA and better than the one of Pradeep and Approx. The translation errors tend to be higher if more line features are in the minimal set.

Next, we vary the rotation magnitude from 0 to 45 degrees (Fig.4.1). The rotation accuracy for the SEgo methods approaches BA accuracy, while translation errors are bigger. The accuracy of both rotation and translation of Approx drops rapidly due to the use of small angle rotation approximation. We then vary the translation magnitude from 1 to 33 (Fig. 4.1-right). Due to the choice of relative error to measure translation, we observe that translation accuracy increases, while the rotation errors grow and then stabilize. PPSEgo has slightly better translation and slightly worse rotation accuracy than EpiSEgo. Again, the accuracy of the new solvers approaches the reference (BA) and outperforms the baselines Pradeep and Approx.

4.2 Real experiments

Matching between frames. We use the processed and rectified grayscale stereo sequences of the KITTI dataset as input [11]. Given four views, we detect

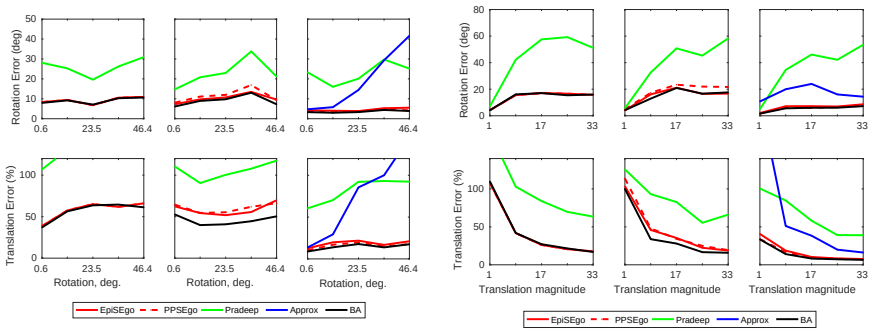

Fig. 3. The effects of rotation (left) and translation (right) magnitude variation on median relative translation and absolute rotation. The columns (left to right) correspond to: easy cases, hard cases, point-only combinations (S3P, S2P-1P). The new solvers PPSEgo and EpiSEgo have the lowest errors approaching the reference method (bundle adjustment with ground truth initialization). The new solvers PPSEgo and EpiSEgo are more accurate than the baselines (*Pradeep* and *Approx*, which is evaluated for point-only case).

and match lines and points using the EDLines + LBD [28, 13] and ORB [12] algorithms implemented in OpenCV.

We evaluate one of our solvers **EpiSEgo** against the baselines **Pradeep** [8], **Approx** [9] and **P3P** [29]. The Pradeep method takes as input four-view point and line correspondences. The P3P method emulates the classical approach of visual SLAM and takes the three-view correspondences constructed from both views of the first camera and the first view of the second camera. The Approx and our EpiSEgo methods work with three-view correspondences. While Approx uses only point features, our method employs both types of features.

To match point features between two images I_l and I_r, for each feature from I_l we find the closest one in I_r by the descriptor distance. We reject the match if its reprojection error after triangulation is less than $\tau = 5$ pixels. We match line segments in the same way, but without the reprojection validation. To find four-view correspondences, we match left and right images in both stereo pairs, and then match left images of the first and the second pairs. Denote the first stereo pair images as I_l, I_r, and the second stereo pair images as I'_l, I'_r. We find three view correspondences for each possible triplet of four images.

Then we run the classical RANSAC loop[2] with the threshold of τ pixels, $p = 0.999$ and the initial outlier ratio of 0.5. For the three-view correspondences, we triangulate a feature using its main camera, project onto the remaining view and compare the reprojection error to τ. For the four-view correspondences, we choose one stereo pair, triangulate the feature using the projections onto its views, and then test the reprojection errors onto the views of the other stereo pair.

In Fig. 4.2 we show the results of the motion estimation experiment with the consecutive frame pairs of KITTI [30] sequence 6. The use of three-view

correspondences and point and line features helps the EpiSEgo to achieve high inlier ratios and lower pose estimation errors compared to the baselines.

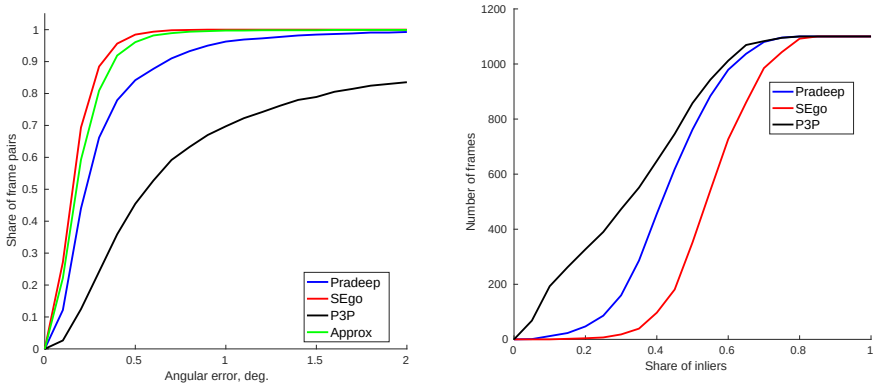

Fig. 4. Results of the experiment on a KITTI sequence 6. Left: cumulative distribution for the rotation error in degrees. Right: cumulative distribution for the ratio of inliers. The proposed EpiSEgo using line and point feature triplets has higher accuracy compared to the baselines. It has higher inlier ratio than P3P and Pradeep. The use of all possible types of feature triplets rather than quadruplets (Pradeep) is beneficial in motion estimation. We also see a benefit from new solvers compared to the classical approach (P3P). Approx is excluded from the inlier plot because it relies only on point feature triplets.

Integration into visual SLAM pipeline. While the previous experiment compares stereo egomotion methods at the task of relative pose estimation between stereo pairs, we also validate that such task can be used to improve modern stereo visual odometry pipelines. For this, we evaluate the system that integrates the proposed EpiSEgo solver into the ORB-SLAM2 [31] pipeline.

The ORB-SLAM2 pipeline uses the previous frame pose as an initial guess to estimate the next frame pose within bundle adjustment. We modify it to run the EpiSEgo solver (point-only version) inside the RANSAC loop. The pose and the inliers estimated by RANSAC are used to initialize bundle adjustment. We run this algorithm each time the standard system loses the track. We do not include line features as they are absent in the original system.

While ORB-SLAM2 works well for the original sequences, it is important to study the robustness of the pipeline to the framerate decrease (which is equivalent to faster observer motion) which can happen in a real system. To do that, we drop *every second* frame of the sequence. Note that the uniform frame drop still enables the use of velocity-based pose prediction on which ORB-SLAM2 relies, provided the frames are separated by equal time periods. At the same

Sequence	ORB-SLAM2			ORB-SLAM2+EpiSEgo		
	F%	t_{rel}	r_{rel}	F%	t_{rel}	r_{rel}
00	100%	-	-	0%	62%	$5 * 10^{-3}$
01	40%	3%	10^{-4}	40%	1.6%	10^{-4}
02	100%	-	-	0%	53%	$3.6 * 10^{-3}$
03	60%	0.8%	10^{-5}	0%	3%	10^{-4}
04	40%	0.8%	10^{-5}	0%	0.8%	10^{-5}
05	100%	-	-	0%	55%	$4.8 * 10^{-3}$
06	100%	-	-	0%	7%	10^{-4}
07	80%	1.3%	10^{-4}	0%	7%	10^{-4}
08	100%	-	-	0%	63%	$4.6 * 10^{-3}$
09	100%	-	-	80%	63%	$4.6 * 10^{-3}$
10	100%	-	-	0%	36%	$2.8 * 10^{-3}$

Table 2. The study of the robustness of the ORB-SLAM2 pipeline to the framerate decrease on the KITTI sequences 00-10. To make the task harder, we drop every second frame of each sequence. We compare the original and the modified pipeline that uses EpiSEgo solver for initialization in case of track loss. We report the percentage of runs where tracking was lost (F%), the relative pose estimation errors as proposed by the dataset authors [30]. After every second frame is dropped, there is the only sequence, which the original ORB-SLAM2 can track with probability more than 50%. At the same time, the modified version shows radical improvement, as it tracks ten out of eleven sequences with probability more than 50%.

time, it shows what can happen if motions become less predictable. Our experiments show that the ORB-SLAM2 often becomes unable to recover and loses the track, while the use of EpiSEgo solver can enable successful recovery from tracking losses. In Tab. 2, we show the results of 5 runs for the original and modified ORB-SLAM2 on 0-10 KITTI sequences. We report the percentage of failures as well as relative rotation and translation errors proposed by the dataset authors. The modified version does not lose track with probability more than 50% for all the sequences except the 9th, where a lack of tracked features in one moment is a possible problem. The original version is able to track with probability greater than 50% for only one sequence out of 11. The experiment shows that the integration of the stereo egomotion solver considerably increases the robustness of the system.

5 Summary

In this paper, we have proposed new minimal solvers that can handle the stereo relative pose problem for any combinations of point and line three-view correspondences. This case was not addressed in the previous literature. We demonstrate that the problem is practical and leads to improved performance of a well-known SLAM system.

References

1. Nister, D.: An efficient solution to the five-point relative pose problem. In: TPAMI, IEEE (2004) 756–770
2. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: Readings in computer vision. Elsevier (1987) 726–740
3. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on Robotics **32**(6) (2016) 1309–1332
4. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building Rome in a day. Communications of the ACM **54**(10) (2011) 105–112
5. Micusik, B., Wildenauer, H.: Structure from motion with line segments under relaxed endpoint constraints. International Journal of Computer Vision **124**(1) (2017) 65–79
6. Xu, C., Zhang, L., Cheng, L., Koch, R.: Pose estimation from line correspondences: A complete analysis and a series of solutions. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(6) (2017) 1209–1222
7. Stewénius, H., Nistér, D., Oskarsson, M., Åström, K.: Solutions to minimal generalized relative pose problems. In: Workshop on omnidirectional vision. Volume 1. (2005) 3
8. Pradeep, V., Lim, J.: Egomotion estimation using assorted features. International Journal of Computer Vision **98**(2) (2012) 202–216
9. Ventura, J., Arth, C., Lepetit, V.: An efficient minimal solution for multi-camera motion. In: ICCV, IEEE (2015) 747–755
10. Kneip, L., Li, H.: Efcient computation of relative pose for multi-camera systems. In: CVPR, IEEE (2014) 1–8
11. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2015)
12. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to sift or surf. In: 2011 International conference on computer vision, IEEE (2011) 2564–2571
13. Zhang, L., Koch, R.: An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency. Journal of Visual Communication and Image Representation **24**(7) (2013) 794–805
14. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry. In: CVPR, IEEE (2004) 652–659
15. Manmohan, C., Jongwoo, L., David, K.: Moving in stereo: Efficient structure and motion using lines. In: International Conference on Computer Vision, IEEE (2009) 1741–1748
16. Brian, C., Christopher, Z., Jan-Michael, F., Marc, P.: A new minimal solution to the relative pose of a calibrated stereo camera with small field of view overlap. In: International Conference on Computer Vision, IEEE (2009) 1725–1732
17. Dunn, E., Clipp, B., Frahm, J.M.: A geometric solver for calibrated stereo egomotion. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 1187–1194
18. Nister, D.: A minimal solution to the generalised 3-point pose problem. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2004)

19. Ramalingam, S., Lodha, S.K., Sturm, P.: A generic structure-from-motion framework. Computer Vision and Image Understanding **103**(3) (2006) 218–228
20. Nistér, D., Stewénius, H.: A minimal solution to the generalised 3-point pose problem. Journal of Mathematical Imaging and Vision **27**(1) (2007) 67–79
21. Merzban, M.H., Abdellatif, M., Abouelsoud, A.: A simple solution for the non perspective three point pose problem. In: 3D Imaging (IC3D), 2014 International Conference on, IEEE (2014) 1–6
22. Miraldo, P., Araujo, H.: Direct solution to the minimal generalized pose. IEEE Transactions on Cybernetics **45**(3) (2015) 404–415
23. Kukelova, Z., Heller, J., Fitzgibbon, A.: Efficient intersection of three quadrics and applications in computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 1799–1808
24. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
25. Char, B.W., Geddes, K.O., Gonnet, G.H., Leong, B.L., Monagan, M.B., Watt, S.: Maple V library reference manual. Springer Science & Business Media (2013)
26. Cox, D.A., Little, J., O'shea, D.: Using algebraic geometry. Volume 185. Springer Science & Business Media (2006)
27. Vakhitov, A., Funke, J., Moreno-Noguer, F.: Accurate and linear time pose estimation from points and lines. In: European Conference on Computer Vision, Springer (2016) 583–599
28. Akinlar, C., Topal, C.: Edlines: A real-time line segment detector with a false detection control. Pattern Recognition Letters **32**(13) (2011) 1633–1642
29. Gao, X.S., Hou, X.R., Tang, J., Cheng, H.F.: Complete solution classification for the perspective-three-point problem. IEEE transactions on pattern analysis and machine intelligence **25**(8) (2003) 930–943
30. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 3354–3361
31. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras. IEEE Transactions on Robotics **33**(5) (2017) 1255–1262