Towards a minimal solution for the relative pose between axial cameras

Francisco Vasconcelos
fpv@isr.uc.pt
João P. Barreto
jpbar@deec.uc.pt

Institute for Systems and Robotics
University of Coimbra
Coimbra, Portugal

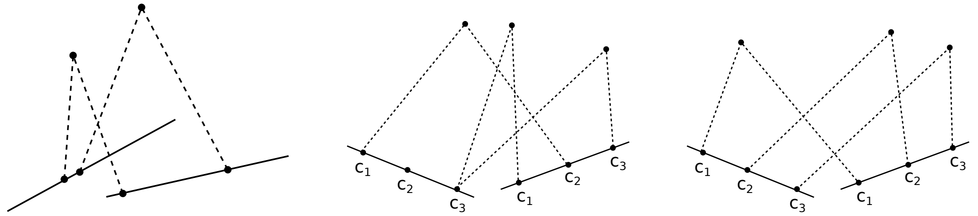
Abstract

The problem of estimating the relative pose between axial cameras from pairwise point correspondences is still open to improvement. The state-of-the-art solutions are either too specific in its scope, assuming certain types of correspondences; too broad, dealing with all types of generalized cameras and failing to address the specific issues of axial cameras; or non-minimal linear solutions. The aim of this paper is to pursue new insights on axial cameras that can lead to a suitable minimal solution for this problem. We propose a new formulation for modeling the intersection of back-projection rays of axial cameras through a 5×5 essential matrix that enables a better understanding of some particular axial configurations and leads to a new set of polynomial equations that proves to be useful in constraining the motion estimation. These equations enable to compute a solution from 10 correspondences, an improvement over the 16-point algorithm, which is the state-of-the-art solution within our aimed scope. Both synthetic and real experiments show that our algorithm achieves a better performance than the 16-point algorithm in the context of robust optimization with RANSAC.

1 Introduction

Vision systems can be classified as being central or non-central [13]. A particular imaging device is central iff all the back-projection rays intersect in a single point in 3D, that is the viewpoint of the camera. Whenever a vision system has more than one viewpoint it is said to be non-central. An axial camera is a particular case of a non-central camera where every back-projection ray intersects a line in 3D (the axis). The axial camera can be used to model vision systems and imaging situations of practical interest. Examples include any catadioptric system that combines a revolution mirror with a central camera for which the viewpoint is aligned with the mirror axis (e.g. a pinhole looking at a spherical mirror) [13]; the situation of a perspective camera looking through multiple flat refractive mediums [1]; or a multi-camera rig composed by two or more pinhole cameras with collinear optical centers [5].

This paper addresses the problem of estimating the relative pose between two axial cameras using point correspondences. Pless showed that the Plücker coordinates of two corresponding back-projection rays must satisfy a bilinear constraint that can be expressed by a 6x6 matrix [9]. This general essential matrix can be estimated from a minimum of 17 point



(a) Generic axial camera (16-point algorithm)

(b) Multi-camera rig with cross-correspondences (16-point algorithm)

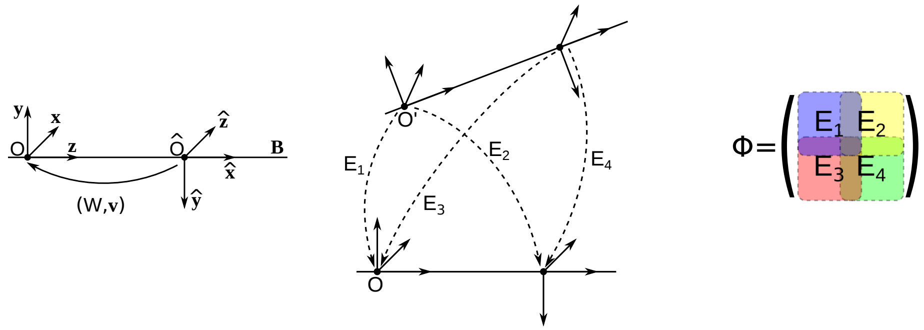
(c) Multi-camera rig with correspondences between the same camera (14-point algorithm)

Figure 1: Axial camera configurations

correspondences using a DLT like approach, and its result factorized into relative camera rotation and translation. Later in [12] Sturm observed that for the case of axial cameras the estimation of this 6x6 matrix was under-determined. He proposed a new 5x5 essential matrix that can be linearly determined from 16 point correspondences. More recently Kim *et al.* investigated the problem of motion estimation using a camera rig composed by multiple perspective cameras with aligned optical centers [5]. They confirmed Sturm's result for the case of considering cross-correspondences between different cameras in the rig (Fig. 1(b)). However, if only matches between the same cameras are allowed, then it is possible to linearly determine the relative motion using a minimum of 14 point correspondences (Fig. 1(c)).

Neither the 16-point solution described by Sturm [12], which is applicable to any axial camera, nor the 14-point algorithm proposed in [5], that is specific to non-overlapping multiple camera rigs, are minimal solutions. The relative pose problem has 6 unknowns meaning that in theory 6 point correspondences provide enough information for determining the relative rotation and translation of the axial camera. Stewenius et al. proposed in [11] a minimal solution for the relative pose between generalized cameras. However, their algorithm is complex, provides a large number of possible solutions (up to 64), and, as reported in [5], it degenerates for most axial camera configurations. This article does not provide a minimal solution for the relative pose between axial cameras, but shows how the motion can be computed using as few as 10 point correspondences. Our 10-point method is an advance with respect to the previous 16-point [5, 12] and 14-point [5] algorithms, that improves the accuracy and efficiency of motion estimation using hypothesize-and-test frameworks.

Please note that, although the 10-point algorithm generalizes to any axial camera, most of the derivations and experiments have in mind the particular case of a conventional stereo camera rig. There are prior works proposing minimal solutions for 6D stereo visual odometry but they either consider sets of features observed, respectively, in four, three, and two views [3], or use five point correspondences between two particular views plus a sixth match for solving for the scale [7]. Unlike these works, we model the stereo rig as a generic axial camera and make no assumptions about the matches. Since any pairwise correspondence can be used as input, the sampling of the solution space is more thorough, being possible to obtain correct motion estimation in circumstances for which the methods of [3, 7] are unable to provide a solution.



(a) Representation of reference frames along the camera axis

(b) The four 3×3 essential matrices encoded by Φ

(c) 5×5 essential matrix for axial cameras

Figure 2: A new parameterization for axial cameras

1.1 Notation

Scalars are represented by plain letters, e.g. a, vectors are indicated by bold symbols, e.g. $\mathbf{t}$, and matrices are denoted by letters in sans serif font, e.g. T. 3D lines are expressed in homogeneous Plucker coordinates, e.g. the 6×1 vector $\mathbf{L} = \begin{pmatrix} \mathbf{d} & \mathbf{m} \end{pmatrix}^{\mathsf{T}}$, in which $\mathbf{d}$ is the line direction and $\mathbf{m}$ is the line momentum. The operator $[\mathbf{v}]_\times$ designates the 3×3 skew symmetric matrix of a 3×1 vector $\mathbf{v}$, which verifies the following property

$$[\mathbf{v}]_\times = \mathsf{R}^{\mathsf{T}} [\mathsf{R}^{\mathsf{T}} \mathbf{v}]_\times \mathsf{R}$$

where R can be any orthonormal 3×3 matrix. We use superscripts to designate submatrices, e. g., $\mathsf{T}^{\{1:2,2:4\}}$ denotes the 2×3 matrix containing the T elements ranging from row 1 to 2 and from column 2 to 4.

2 A new parametrization for axial cameras

Sturm describes in [12] a 5x5 essential matrix that relates back-projection rays of two axial views. We provide a different parametrization of this matrix that, not only enables to understand the results described in [5], but also proves to be useful in deriving polynomial equations that will constraint the motion estimation.

2.1 Linear subspace for back-projection rays

We define two reference frames O and $\hat{\mathsf{O}}$ along the camera axis $\mathbf{B}$ as depicted in Fig. 2(a), with an arbitrary baseline b. The transformation of homogeneous coordinates from O to $\hat{\mathsf{O}}$ is given by a rotation

$$\mathsf{W} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{1}$$

and a translation

$$\mathbf{v} = \begin{pmatrix} 0 & 0 & b \end{pmatrix}^{\mathsf{T}} \tag{2}$$

Given that all back-projection rays $\mathbf{L}_i$ of an axial camera intersect the axis $\mathbf{B}$, they belong to a linear line congruent of dimension 4 [10]. This means that all rays can be represented as a linear combination of 5 base lines $\mathbf{L}_x$, $\mathbf{L}_y$, $\mathbf{L}_z$, $\hat{\mathbf{L}}_y$, $\hat{\mathbf{L}}_z$. Therefore there is a vector λ_i such that for any line $\mathbf{L}_i$ the following holds

$$\mathbf{L}_i = \underbrace{\begin{pmatrix} \mathbf{L}_x & \mathbf{L}_y & \mathbf{L}_z & \hat{\mathbf{L}}_y & \hat{\mathbf{L}}_z \end{pmatrix}}_{\mathsf{\Gamma}} \lambda_i \tag{3}$$

These 5 lines that compose the linear mapping $\mathsf{\Gamma}$ can be arbitrarily chosen, assuring that they intersect $\mathbf{B}$ and are linearly independent. For the purpose of our formulation we align these lines respectively with the axes $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, $\hat{\mathbf{y}}$, $\hat{\mathbf{z}}$ in Fig. 2(a) and therefore

$$\mathsf{\Gamma} = \begin{pmatrix} \mathsf{I}_{3\times 3} & \mathsf{W}^{\{1:3,2:3\}} \\ 0_{3\times 3} & [\mathbf{v}]_{\times}^{\{1:3,2:3\}} \end{pmatrix} \tag{4}$$

2.2 Essential matrix for axial cameras

Given a set of intersecting ray correspondences $(\mathbf{L}_i, \mathbf{L}_i')$, we can establish linear relations using the generalized camera model introduced by Pless [9]

$$\mathbf{L}_i{}^{\mathsf{T}} \begin{pmatrix} [\mathbf{t}]_{\times}\mathsf{R} & \mathsf{R} \\ \mathsf{R} & 0 \end{pmatrix} \mathbf{L}_i' = 0 \tag{5}$$

Taking into account equation 3 this expression can be rewritten as

$$\lambda_i{}^{\mathsf{T}} \underbrace{\mathsf{\Gamma}^{\mathsf{T}} \begin{pmatrix} [\mathbf{t}]_{\times}\mathsf{R} & \mathsf{R} \\ \mathsf{R} & 0 \end{pmatrix} \mathsf{\Gamma}}_{\Phi} \lambda_i' = 0 \tag{6}$$

with Φ being the 5×5 essential matrix for axial cameras. By substituting equation 4 into Φ we can define the following matrices

$$\mathsf{E}_1 = \Phi^{\{1:3,1:3\}} = [\mathbf{t}]_{\times}\mathsf{R} \tag{7}$$

$$\mathsf{E}_2 = \Phi^{\{1:3,3:5\}} = [\mathsf{R}\mathbf{v} + \mathbf{t}]_{\times}\mathsf{R}\mathsf{W} \tag{8}$$

$$\mathsf{E}_3 = \Phi^{\{3:5,1:3\}} = [\mathsf{W}^{\mathsf{T}}(\mathbf{t} - \mathbf{v})]_{\times}\mathsf{W}^{\mathsf{T}}\mathsf{R} \tag{9}$$

$$\mathsf{E}_4 = \Phi^{\{3:5,3:5\}} = [\mathsf{W}^{\mathsf{T}}(\mathsf{R}\mathbf{v} + \mathbf{t} - \mathbf{v})]_{\times}\mathsf{W}^{\mathsf{T}}\mathsf{R}\mathsf{W} \tag{10}$$

The placement of E_1, E_2, E_3, E_4 within Φ can be better visualized in Fig. 2(c). From the above expressions it can be observed that E_1, E_2, E_3, E_4 are the 3×3 essential matrices that encode the motions represented in Fig. 2(b).

From the equations 7 to 10 we can also derive the following relations

$$\mathsf{E}_1 - \mathsf{E}_2\mathsf{W}^{\mathsf{T}} - \mathsf{W}\mathsf{E}_3 + \mathsf{W}\mathsf{E}_4\mathsf{W}^{\mathsf{T}} = 0 \tag{11}$$

$$\mathsf{E}_1[\mathbf{v}]_{\times} + [\mathbf{v}]_{\times}\mathsf{E}_1 - [\mathbf{v}]_{\times}\mathsf{E}_2\mathsf{W}^{\mathsf{T}} - \mathsf{W}\mathsf{E}_3[\mathbf{v}]_{\times} = 0 \tag{12}$$

These constraints provide 8 linear equations on the parameters of Φ and therefore they can be used to reduce this matrix from its 25 parameters to a linear combination of 17 parameters. This means that it is possible to estimate Φ with a DLT like approach using 16 correspondences, which is in conformity with the linear formulations introduced in [12].

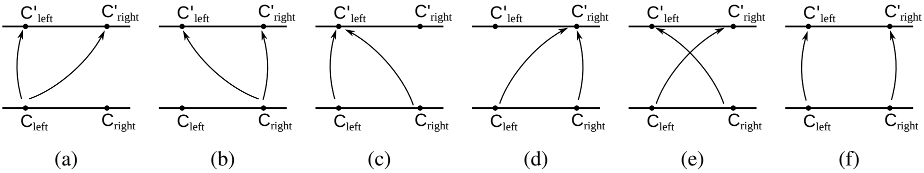


Figure 3: Different configurations when only two types of correspondences are established between two stereo pairs.

2.3 A particular axial camera: the stereo rig

The above formulation applies directly to the case of motion estimation between stereo pairs, however, there are particular configurations that require additional considerations. Furthermore, in this case it is advantageous to consider that reference frames O and $\hat{\mathsf{O}}$ from Fig. 2(a) are coincident with the principal points of the stereo pair, with the translation $\mathbf{v}$ being the baseline between the cameras. This way all line coordinates in the left and right cameras will have the form

$$\lambda_{left} = \begin{pmatrix} l_1 & l_2 & l_3 & 0 & 0 \end{pmatrix}^\mathsf{T} \tag{13}$$

$$\lambda_{right} = \begin{pmatrix} 0 & 0 & l_3 & l_4 & l_5 \end{pmatrix}^\mathsf{T} \tag{14}$$

Considering the 4 cameras in this scenario, $\{\mathsf{C}_{left}, \mathsf{C}_{right}, \mathsf{C}'_{left}, \mathsf{C}'_{right}\}$, there are 4 different types of correspondences that can be used: $(\mathsf{C}_{left}, \mathsf{C}'_{left})$, $(\mathsf{C}_{left}, \mathsf{C}'_{right})$, $(\mathsf{C}_{right}, \mathsf{C}'_{left})$, $(\mathsf{C}_{right}, \mathsf{C}'_{right})$. If at least 3 of these types of correspondences are available then we can use the formulation from section 2.2. However, when only 2 types are available Φ cannot be fully known using just linear constraints and we can further reduce the number of parameters to estimate. The cases depicted in Fig. 3(a) to 3(d) only have linear constraints on two contiguous essential matrices that share 3 parameters (see Fig. 2(c)), and therefore we can reduce Φ to either a 3×5 or a 5×3 matrix, which can be linearly estimated from just 14 correspondences. The cases depicted in Fig. 3(e) and 3(f) only have linear constraints on essential matrices that share one single element (either E_1 and E_4 or E_2 and E_3), resulting in 17 parameters. However, we must note that the following relations

$$\mathsf{F}_1 = \mathsf{W}\mathsf{E}_4\mathsf{W}^\mathsf{T} - \mathsf{E}_1 = \mathsf{R}[\mathbf{v}]_\times - [\mathbf{v}]_\times\mathsf{R} \tag{15}$$

$$\mathsf{F}_2 = \mathsf{E}_2\mathsf{W}^\mathsf{T} - \mathsf{W}\mathsf{E}_3 = \mathsf{R}[\mathbf{v}]_\times + [\mathbf{v}]_\times\mathsf{R} \tag{16}$$

imply that the diagonals of F_1 and F_2 are respectively $\begin{pmatrix} a & -a & 0 \end{pmatrix}^\mathsf{T}$ and $\begin{pmatrix} a & a & 0 \end{pmatrix}^\mathsf{T}$, with a being an unkown scalar value. In either case these equations provide 2 linear constraints that enable to reduce the number of parameters to 15, and therefore we are able to compute them from 14 correspondences.

It is noteworthy that [5] only addresses the case depicted in Fig. 3(f), and that in [14] a linear formulation with 15 parameters was proposed to address the cases 3(a) to 3(d). On the other hand our analysis covers all the cases.

3 Towards a minimal solution

Henceforth we will only address the general axial case, but analogous conclusions can be drawn for the particular cases of Fig. 3.

n. of variables	1	2	3	4	5	6	7	8	9	10
n. of monomials	4	10	20	35	56	84	120	165	220	286
n. of leading monomials	1	4	10	20	35	56	84	120	165	220

Table 1: Number of monomials in dense cubic polynomials

Using equations 7 to 10 we can write the following expression

$$\alpha \mathsf{E}_1 + \beta \mathsf{E}_2 \mathsf{W}^\mathsf{T} + \gamma \mathsf{W}\mathsf{E}_3 + \delta \mathsf{W}\mathsf{E}_4\mathsf{W}^\mathsf{T} = [\alpha \mathbf{t} + \beta(\mathsf{R}\mathbf{v}+\mathbf{t}) + \gamma(\mathbf{t}-\mathbf{v}) + \delta(\mathsf{R}\mathbf{t}+\mathbf{t}-\mathbf{v})]_\times \mathsf{R} \quad (17)$$

with α, β, γ, δ being any real values. This means that any matrix E_i that is a linear combination of E_1, $\mathsf{E}_2\mathsf{W}^\mathsf{T}$, $\mathsf{W}\mathsf{E}_3$, $\mathsf{W}^\mathsf{T}\mathsf{E}_4\mathsf{W}$ is itself an essential matrix, verifying the cubic constraints

$$2\mathsf{E}_i{\mathsf{E}_i}^\mathsf{T}\mathsf{E}_i - tr(\mathsf{E}_i{\mathsf{E}_i}^\mathsf{T})\mathsf{E}_i = 0 \quad (18)$$

$$\det \mathsf{E}_i = 0 \quad (19)$$

From this result we are able to generate a high amount of polynomial equations by choosing different values for α, β, γ, δ. Using simulated data we were able to find 78 linearly independent equations. These equations can then be used in a similar fashion to [8] in order to reduce the number of required correspondences to solve our problem. If we use $16-N$ correspondences, we can compute a $N+1$ dimensional linear subspace using the equations 6, 11 and 12. Posteriorly we introduce this subspace into instances of equations 18 and 19 to form a polynomial system in N variables that can must be solved in closed form.

The minimal solution for this problem requires only 6 correspondences, which means that a polynomial system in 10 variables would need to be solved. However, as the number of variables grows, the more difficult it is to generate a stable polynomial solver.

Solving polynomial systems can be achieved with the action matrix method described in [2]. It falls outside the scope of this paper to provide a full account of this method, however, some of its requirements should be discussed.

In order solve a polynomial system a minimum number of linearly independent polynomial equations is required. This number is usually difficult to determine, however, it is a general rule of thumb that it increases with the number of different monomials that are present in the equations. If for a given system we have less equations than the minimum required, it means that new higher order equations need to be generated by multiplying the existing ones by other polynomials, resulting in new equations that although being redundant can be useful to solve the system if they are linearly independent.

Equations 18 and 19 always produce dense cubic polynomials, which means that they contain all the monomials up to the 3rd degree. In this particular case, the system is guaranteed to be solvable if the number of linearly independent equations is greater than or equal to the number of leading monomials (3rd degree). Note however that this is not a necessary condition.

In table 1 we list the number of leading monomials for a varying number of variables in a dense cubic equation. Since we can generate a maximum of 78 linearly independent cubic equations, it is possible to build an action matrix for a maximum of 6 variables without the need for generating higher order equations. With a higher number of variables the polynomial system becomes infeasibly complex for our action matrix approach. This means that we are able to implement a 10-point algorithm using this technique, which requires 56 equations. In our implementation we generated equations from the constraints of the following set of essential matrices: $\{\mathsf{E}_1, \mathsf{E}_2\mathsf{W}^\mathsf{T}, \mathsf{W}\mathsf{E}_3, \mathsf{W}^\mathsf{T}\mathsf{E}_4\mathsf{W}, \mathsf{E}_1 + \mathsf{E}_2\mathsf{W}^\mathsf{T}, \mathsf{E}_1 + \mathsf{W}\mathsf{E}_3\}$.

4 The 10-point algorithm

In this section we summarize the steps required to estimate the relative pose between axial cameras using 10 correspondences:

1. Map correspondences $(\mathbf{L}_i, \mathbf{L}'_i)$ into 4D homogeneous coordinates (λ_i, λ'_i) taking into account the relation of equation 3.

2. Stack 10 instances of equation 6 in terms of the 25 parameters of Φ, and then use equations 11 and 12 to eliminate 8 parameters in Φ

3. Generate a 7 dimensional linear subspace of solutions using SVD. Since the linear solution is up to scale, one of the parameters can be set to 1, resulting in 6 unkowns.

4. Introduce the linear expression with 6 unknowns into the polynomial constraints resulting from the following set of essential matrices: $\{\mathsf{E}_1, \mathsf{E}_2\mathsf{W}^\mathsf{T}, \mathsf{W}\mathsf{E}_3, \mathsf{W}^\mathsf{T}\mathsf{E}_4\mathsf{W}, \mathsf{E}_1 + \mathsf{E}_2\mathsf{W}^\mathsf{T}, \mathsf{E}_1 + \mathsf{W}\mathsf{E}_3\}$.

5. Use 56 polynomial equations to compute the action matrix, and obtain 56 solutions that correspond to its eigenvectors.

6. Substitute all 56 solutions back into the polynomial system and select the solution with the smallest residue as the correct one. Since the problem is over constrained, only one solution should have a residue close to zero.

7. Project E_1, E_2, E_3, E_4 individually onto the essential matrix manifold using SVD decomposition.

8. For each of the four 3×3 essential matrices, make an independent factorization to find R and estimate $\mathbf{t}$ with the correct scale by substituting R into equation 6.

9. From the four different estimations of $(\mathsf{R}, \mathbf{t})$ choose the one with minimum residue in equation 6 (alternatively, measure the re-projection errors).

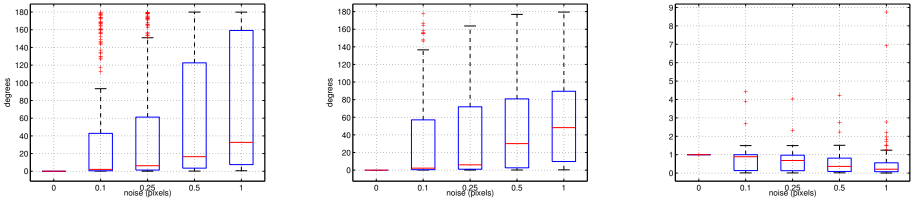
5 Experimental Validation

In this section we validate our algorithm with both synthetic and real data in the case of pose estimation between stereo cameras. We allow all types of correspondences and therefore use the 17-parameter formulation of the problem.

We compare our algorithm against the 16-point linear algorithm from Kim *et al.* [5]. In its original version this algorithm has a refinement step that alternates between successive translation and rotation estimations. We do not implement this step in our experiments because it aims at being fast and simple, sacrificing some robustness and optimality. This step would not likely be used in a real scenario, where refinement would be accomplished with bundle adjustment. Additionally, we use both algorithms within a RANSAC framework.

5.1 Synthetic data

We built a simulated environment in which two stereo camera pairs are randomly positioned with overlapping field of views, and in front of them a set of 3D points is randomly generated



(a) Rotation error (b) Translation angle error (c) Translation scale ratio

Figure 4: Error distribution of 10-point algorithm with 250 synthetic trials. The inner box markers are median values, the box limits are the 25th and 75th percentiles, the dotted line limits are the minimum and maximum values, and the outer markers are outliers.

within a bounded region. The input to the algorithms are the back-projections of the 3D points according to the pinhole model affected by image gaussian noise.

Given a motion estimation $(\mathsf{R},\mathbf{t})$ and the groundtruth values $(\mathsf{R}_{GT},\mathbf{t}_{GT})$, the error in rotation is measured by the euler angle of the residual rotation $\mathsf{R}^{\mathsf{T}}\mathsf{R}_{GT}$, the error in translation direction is measured as the angle defined by $\mathbf{t}$ and $\mathbf{t}_{GT}$, and the ratio $||\mathbf{t}||/||\mathbf{t}_{gt}||$ evaluates the quality in the estimation of the translation scale factor.

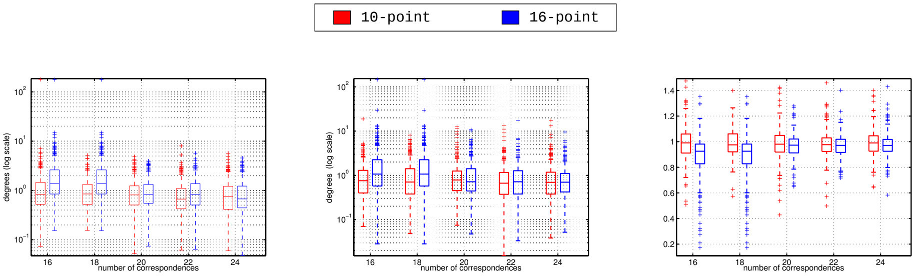
In a first simulation our algorithm is tested 250 times using 10 input pairwise correspondences for different noise magnitudes, and therefore RANSAC is not used. The error distributions for translation and rotation displayed in Fig 4 show that in a noise free scenario our algorithm outputs the exact solution. However, for a noise magnitude over 0.5 pixels the stability decreases significantly, which suggests that a robust estimation is required to improve performance. It is also noticeable that with high levels of noise there is a bias that systematically underestimates the translation scale factor, which calls for further study of the problem in the future.

In a second simulation our algorithm is compared against the 16-point linear algorithm for a different number of input correspondences while injecting noise with 1 pixel of standard deviation. Again, 250 trials were tested for each case and the error distributions for translation and rotation are displayed in Fig. 5. The better performance of our algorithm is specially evident when using a low number of correspondences. But it also must be noticed that for a higher number of correspondences, while the stability between both algorithms is similar, our algorithm is significantly faster due to the fact that RANSAC is sampling 10 points instead of 16. The RANSAC procedure eliminates the previously observed bias in the estimation of translation scale factor induced by our 10-point algorithm, furthermore, this effect is still visible on the 16-point algorithm for estimations with 16 and 18 correspondences, in which the RANSAC provides few to none advantage.

5.2 Real data

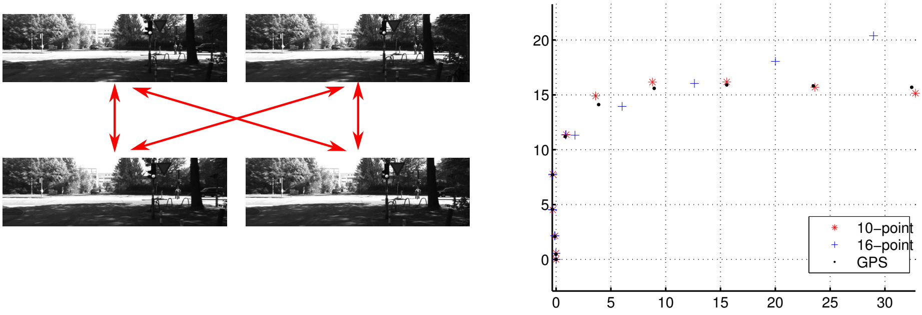
We used data from the The KITTI Vision Benchmark Suite [4], which contains fully calibrated stereo sequences and GPS measurements acquired by a vehicle on an urban environment. We selected a set of images from the sequence ”2011_09_28_drive_0001” and compare the trajectories estimated by both algorrithms.

Correspondences were obtained by matching SIFT features [6] on the four combinations of image pairs of consecutive frames (Fig. 6(a)). Both our 10-point algorithm and the 16-



(b) Rotation error (log scale) (c) Translation angle error (log scale) (d) Translation scale ratio

Figure 5: Performance comparison between 10-point algorithm and 16-point algorithm with synthetic data. The inner box markers are median values, the box limits are the 25th and 75th percentiles, the dotted line limits are the minimum and maximum values, and the outer markers are outliers.



(a) Input data (b) Trajectory

Figure 6: Performance comparison between 10-point algorithm and 16-point algorithm with real data.

point algorithm are used within a RANSAC framework. Given the high amount of sampling points for both algorithms the number of RANSAC iterations can become infeasibly high. To solve this problem a previous filtering of outliers is made using a 5-point RANSAC validation in all four combinations of matched image pairs. The set of input correspondences after filtering will contain a very high inlier ratio which significantly decreases the number of iterations required to find an accurate estimation. We use these algorithms just to make odometry estimations, i. e., the trajectory is not refined by bundle adjustment. This way the accumulation of drift error serves as an evaluation of comparative accuracy, showing that our algorithm provides more robust estimations (Fig. 6(b)).

6 Conclusions

We proposed a new parametrization for modelling the intersection of back-projection rays of axial cameras. When dealing with the particular case of stereo cameras, this new formulation presents a common framework for dealing with different scenarios that were previously treated separately. Furthermore, it allows to derive a number of non-linear constraints that had not been studied yet, and derive a 10-point algorithm that extends the range of applications covered by previous methods [5, 12]. Our long-term goal, however, is to reach a 6-point minimal algorithm, which will require a more in-depth study of the non-linear relations between the essential matrices described in this paper.

7 Acknowledgements

The authors gratefully acknowledge the Portuguese Science Foundation (FCT) that generously funded this work through project PTDC/EIA-CCO/109120/2008 and grant SFRH/BD/72323/2010.

References

[1] A. Agrawal, S. Ramalingam, Y. Taguchi, and V. Chari. A theory of multi-layer flat refractive geometry. In *CVPR*, 2012.

[2] Martin Byröd, Klas Josephson, and Kalle Åström. Fast and stable polynomial equation solving and its application to computer vision. *Int. J. Comput. Vision*, 84(3):237–256, September 2009. ISSN 0920-5691.

[3] E. Dunn, B. Clipp, and J.-M. Frahm. A geometric solver for calibrated stereo egomotion. In *ICCV*, 2011.

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, Providence, USA, June 2012.

[5] Jae-Hak Kim, Hongdong Li, and Richard Hartley. Motion estimation for nonoverlapping multicamera rigs: Linear algebraic and L_∞ geometric solutions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(6):1044–1059, June 2010.

[6] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94.

[7] Janosch Nikolic. Real-time 6d stereo visual odometry with non-overlapping fields of view. In *CVPR*, 2012.

[8] D. Nister. An efficient solution to the five-point relative pose problem. In *CVPR*, 2003.

[9] R. Pless. Using many cameras as one. In *CVPR*, 2003.

[10] H Pottmann and J Wallner. *Computational line geometry*. Springer Verlag, Berlin, 1 edition, 2001.

[11] H. Stewénius, D. Nistér, M. Oskarsson, and K. Åström. Solutions to minimal generalized relative pose problems. In *OMNIVIS*, 2005.

[12] P. Sturm. Multi-view geometry for general camera models. In *CVPR*, 2005.

[13] Peter Sturm, Srikumar Ramalingam, Jean-Philippe Tardif, Simone Gasparini, and João Barreto. Camera models and fundamental concepts used in geometric computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(1-2):1–183, 2011.

[14] Francisco Vasconcelos, João P. Barreto, and Edmond Boyer. A minimal solution for camera calibration using independent pairwise correspondences. In *ECCV*, 2012.