· U C ·

UNIVERSIDADE DE COIMBRA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES
INSTITUTO DE SISTEMAS E ROBÓTICA
3030-290 COIMBRA, PORTUGAL

ARTICLE: **A Simple and Robust Solution to the Minimal General Pose Estimation**

Author:
Pedro MIRALDO
miraldo@isr.uc.pt

Co-Author:
Helder ARAUJO
helder@isr.uc.pt

CONTENTS

Preprint: submitted to the IEEE Proc. Int'l Conf. Robotics & Automation (ICRA 2014)
September, 2013.

A Simple and Robust Solution to the Minimal General Pose Estimation*

Pedro Miraldo and Helder Araujo

Abstract— **In this article we address the problem of minimal pose under the framework of the generalized camera models. Previous approaches were based on geometric properties, such as the preservation of distance between points. In this paper we propose a novel formulation to the problem using an algebraic-based approach. We represent the pose by a 3×3 matrix. Using both the algebraic relationship between three incident 3D points and straight lines and the underlying constraints of the pose matrix, pose can be computed. In terms of experimental results, the main contribution of the proposed method is the robustness to critical configurations. In addition, a full comparison and analysis between state-of-the-art methods is made (so far, there is no published comparison between state-of-the-art methods published).**

I. INTRODUCTION

In this article we address the problem of minimal absolute pose for general camera models, Fig. 1. This problem consists on the estimation of the rotation and translation parameters that define the rigid transformation between three 3D points (in the world coordinate system), and three 3D lines (in the camera coordinate system). To handle any type of camera (central and non-central), we consider the general camera models [6] which basically consist on the individual association of image pixels and non-constrained 3D lines.

The analysis of solutions using minimal data is important because of their computation speed and their applications in hypothesis-and-test estimation methods such as RANSAC [5], [16]. They also allow insights and a better understanding of the problem.

The basic approach to solve this problem consists in computing the coordinates of the 3D points in the camera coordinate system, using the known distance between the three points. This formulation was studied by both Chen & Chang [2], [3] and Ramalingam et al. at [21]. In both articles, to solve the problem, the authors claim that it is possible to derive an eight degree polynomial equation. Chen & Chang proposed a set of transformations of the data set so that the computation of the coefficients of the eight degree polynomial is easier. For these two approaches and since the pose is given by the transformation between the world and camera coordinate systems, to compute the rotation and translation from two 3D points sets, additional steps are required. To compute this rigid transformation the conventional methods such as [1], [25], require the *singular value decomposition* (SVD) of a 3×3 matrix for each valid solution. Note that there are closed-form solutions for this decomposition. However, as it can be seen from the experimental results, the numerical accuracy deteriorates when the closed-form solutions are used. Thus, it is usual to use iterative methods to compute the SVD, which increases the computational effort. See for instance the analysis made by Kneip et al. [11], for the case of minimal pose problem for central cameras.

To deal with these drawbacks, Nistér and Stewénius at [17], [19] express the rotation and translation parameters as a problem of intersection between a ruled quartic surface and a circle. Differently from Chen & Chang's method, Nistér & Stewénius's algorithm outputs the rotation and translation parameters that represent pose. However, their formulation of this problem is complex.

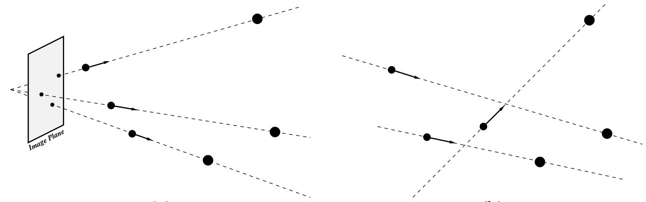
In this article we propose a new parameterization of the minimal pose problem. As Nistér and Stewénius's method, our approach also yields the pose parameters directly. In the remaining of this section, we summarise the main contributions of the paper.

Let us consider general calibration methods, such as [6], [24], [15]. For any specific camera model it is expected that, due to noise, the 3D straight lines generated by the general camera models do not fit the underlying camera geometry. For instance, when we calibrate a perspective camera with noisy data and using general methods, the probability of the lines intersect at a single point is very small. However, depending on the quality of the calibration data and calibration process, all the 3D lines must pass close to a common point. The same analysis can be made for other configurations such as *Pushbroom* cameras [8], [10] Fig. 3(c) (these types of cameras are used in a wide variety of applications that goes from CT X-rays to satellite imaging) and *X-Slit* cameras [26] Fig. 3(e) (used in photography to create new images). For these cases, general solutions for the minimal absolute pose problem must be used. Thus, general solutions for pose must perform in these situations as well as they perform in general configurations (when 3D straight lines can be random). The main contributions of the paper, are:

- When compared to previous methods, our formulation is significantly more robust to critical configurations.
- A simple solution that can be easily implemented (specially when compared with the Nistér & Stewénius algorithm);
- Outputs the rotation and translation parameters that define the pose. When compared to Chen & Chang and Ramalingam et al., it does not require additional steps for the computation of the rotation and translation from

*This work was not supported by the Portuguese Foundation for Science and Technology with grant SFRH/BD/49054/2008

Both authors are with the Institute for System and Robotics and Departments of Electrical and Computer Engineering, University of Coimbra, 3030-290 Coimbra, Portugal. Correspondence: {miraldo,helder}@isr.uc.pt.



(a) (b)

Fig. 1. Fig. (a) and Fig. (b) represent the minimal absolute pose problem for central and non-central camera models repsectively.

two 3D points sets;

- Full comparison and analysis between the proposed algorithm and state-of-the-art methods. So far, there is no published work comparing state-of-the-art approaches, Moreover, in the author page, we released the code for the proposed algorithm as well as the code for the state-of-the-art-methods.

II. Proposed Solution

Pose Estimation requires the estimation of a rotation matrix $\mathbf{R}_{\text{out}} \in \mathscr{SO}(3)$ and a translation vector $\mathbf{t}_{\text{out}} \in \mathbb{R}^3$ that define the rigid transformation between the world and camera coordinate systems. Since we consider that the imaging device is calibrated according to [6], pose is specified by the rigid transformation that satisfies the relationship of incidence between points in the world coordinate system and 3D straight lines represented in the camera coordinate system, Fig. 1.

The rigid transformation between a point in world coordinates $\mathbf{p}^{(\mathscr{W})}$ and the same point in camera coordinates $\mathbf{p}^{(\mathscr{C})}$ is given by

$$\mathbf{p}^{(\mathscr{C})} = \mathbf{R}\mathbf{p}^{(\mathscr{W})} + \mathbf{t}. \tag{1}$$

$\mathbf{R}$ represents the rotation from the world to the camera coordinate system. The translation $\mathbf{t}$ is defined in the camera coordinate system. The minimal pose uses only three world points. Let us consider the plane $\Pi^{(\mathscr{W})}$ defined by the three world points

$$\Pi^{(\mathscr{W})} \doteq \left[\zeta^{(\mathscr{W})}, \pi^{(\mathscr{W})}\right] \doteq \mathbf{p}_1^{(\mathscr{W})} \cup \mathbf{p}_2^{(\mathscr{W})} \cup \mathbf{p}_3^{(\mathscr{W})}, \tag{2}$$

where $\zeta^{(\mathscr{W})}$ and $\pi^{(\mathscr{W})}$ are the distance from the plane to the origin and the unit normal vector to the plane.

Without loss of generality and since we are using only three world points, we can use the *planar homography* [10] to represent the transformation from points in the world to the camera coordinate system. Thus, we can rewrite (1) as

$$\mathbf{p}^{(\mathscr{C})} = \underbrace{\left(\mathbf{R} + \frac{1}{\zeta^{(\mathscr{W})}}\mathbf{t}\,\pi^{(\mathscr{W})\,T}\right)}_{\mathbf{H}}\mathbf{p}^{(\mathscr{W})} \tag{3}$$

where $\mathbf{H} \in \mathbb{R}^{3\times 3}$ is called the *homography* matrix [10], [14].

Moreover and again without loss of generality, we can apply a pre-defined transformation to the data points and plane coordinates $\widetilde{\mathbf{R}}$ and $\widetilde{\mathbf{t}}$. We consider a transformation such that

$$\widetilde{\mathbf{p}}^{(\mathscr{W})} = \widetilde{\mathbf{R}}\mathbf{p}^{(\mathscr{W})} + \widetilde{\mathbf{t}} \tag{4}$$

$$\widetilde{\Pi}^{(\mathscr{W})} \doteq \left[\widetilde{\zeta}^{(\mathscr{W})}, \widetilde{\pi}^{(\mathscr{W})}\right] = \left[\zeta^{(\mathscr{W})} - \widetilde{\mathbf{t}}^T\widetilde{\mathbf{R}}\pi^{(\mathscr{W})}, \widetilde{\mathbf{R}}\pi^{(\mathscr{W})}\right], \tag{5}$$

which makes $\widetilde{\pi}^{(\mathscr{W})}$ parallel to the z-axis (additional information regarding the computation of the to get the transformation parameters will be available in the author's page). The choice for $\widetilde{\zeta}^{(\mathscr{W})}$ is not irrelevant. In the next section we describe the constraints for the selection of this parameter –see the text after the Theorem 1.

Using this representation we can simplify (3) such that

$$\mathbf{p}^{(\mathscr{C})} = \underbrace{\left(\mathbf{R} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \frac{1}{\widetilde{\zeta}^{(\mathscr{W})}}\mathbf{t} \end{bmatrix}\right)}_{\mathbf{H}}\widetilde{\mathbf{p}}^{(\mathscr{W})}, \tag{6}$$

which means that matrix $\mathbf{H}$ – that defines the aimed transformation, is such that

$$\mathbf{H} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \left(\mathbf{r}_3 + \frac{1}{\widetilde{\zeta}^{(\mathscr{W})}}\mathbf{t}\right) \end{bmatrix} \tag{7}$$

where $\mathbf{r}_i$ is the i^{th} column of the rotation matrix $\mathbf{R}$. From the fact that $\mathbf{R}$ must belong to the special orthogonal group, the following three constraints can be easily derived

$$\mathbf{r}_1^T\mathbf{r}_1 = 1, \text{ which implies } \mathbf{h}_1^T\mathbf{h}_1 = 1 \tag{8}$$

$$\mathbf{r}_2^T\mathbf{r}_2 = 1, \text{ which implies } \mathbf{h}_2^T\mathbf{h}_2 = 1 \tag{9}$$

$$\mathbf{r}_1^T\mathbf{r}_2 = 0, \text{ which implies } \mathbf{h}_1^T\mathbf{h}_2 = 0. \tag{10}$$

From now on, we will consider that the points in the world are represented in this coordinate system.

A. Formalization

Let us consider the general case where the 3D lines can or cannot intersect at a single point in the world. Let us consider the 3D straight lines represented in *Plücker* coordinates $\mathbf{l}^{(\mathscr{C})}\mathbb{R} \doteq \left(\mathbf{d}^{(\mathscr{C})}, \mathbf{m}^{(\mathscr{C})}\right)$, where $\mathbf{d}^{(\mathscr{C})}$ and $\mathbf{m}^{(\mathscr{C})}$ represent the direction and moment of the line respectively. Using this representation and from [20], a point that is incident on a line verifies the following relationship

$$\mathbf{d}^{(\mathscr{C})} \times \mathbf{p}^{(\mathscr{C})} = \widehat{\mathbf{d}}^{(\mathscr{C})}\mathbf{p}^{(\mathscr{C})} = \mathbf{m}^{(\mathscr{C})}. \tag{11}$$

$\widehat{\mathbf{a}} \in \mathbb{R}^{3\times 3}$ represents the skew–symmetric matrix that linearizes the exterior product such that $\mathbf{a} \times \mathbf{b} = \widehat{\mathbf{a}}\mathbf{b}$.

We wish to determine the relationship between the points in the world coordinate system and the lines in the camera coordinate system. Thus and considering (6), we can rewrite (11) such that

$$\widehat{\mathbf{d}}^{(\mathscr{C})}\mathbf{H}\widetilde{\mathbf{p}}^{(\mathscr{W})} = \mathbf{m}^{(\mathscr{C})}. \tag{12}$$

The unknown is the matrix $\mathbf{H}$. Thus, let us consider the linearization of the unknown matrix $\mathbf{H}$ in (12), using the *Kronecker* product,

$$\left(\widetilde{\mathbf{p}}^{(\mathscr{W})\,T} \otimes \widehat{\mathbf{d}}^{(\mathscr{C})}\right)\operatorname{vec}(\mathbf{H}) = \mathbf{m}^{(\mathscr{C})}. \tag{13}$$

This minimal problem, corresponds to the determination of the mapping between three world points and their corresponding 3D straight lines. Therefore and using the representation of (13), we can define the following algebraic relation

$$\underbrace{\begin{bmatrix} \widetilde{\mathbf{p}}_1^{(\mathscr{W})\,T} \otimes \hat{\mathbf{d}}_1^{(\mathscr{C})} \\ \widetilde{\mathbf{p}}_2^{(\mathscr{W})\,T} \otimes \hat{\mathbf{d}}_2^{(\mathscr{C})} \\ \widetilde{\mathbf{p}}_3^{(\mathscr{W})\,T} \otimes \hat{\mathbf{d}}_3^{(\mathscr{C})} \end{bmatrix}}_{\mathbf{M}} \operatorname{vec}(\mathbf{H}) = \underbrace{\begin{bmatrix} \mathbf{m}_1^{(\mathscr{C})} \\ \mathbf{m}_2^{(\mathscr{C})} \\ \mathbf{m}_3^{(\mathscr{C})} \end{bmatrix}}_{\mathbf{w}} \tag{14}$$

where $\mathbf{M} \in \mathbb{R}^{9\times 9}$, $\mathbf{w} \in \mathbb{R}^9$. The space of the solutions for the unknown matrix $\mathbf{H}$ will depends on the dimension of the column space of the matrix $\mathbf{M}$. To handle that issue, the following theorem was derived:

Theorem 1: Consider a set of three points defined in the world coordinate system $\widetilde{\mathbf{p}}_i^{(\mathscr{W})}$ and their correspondent lines in the camera coordinate system $\left(\mathbf{d}_i^{(\mathscr{C})}, \mathbf{m}_i^{(\mathscr{C})}\right)$ for $i = 1, 2, 3$. If the three points define a plane that does not contain the origin, the dimension of the *column-space* of $\mathbf{M}$ in (14) will be $\operatorname{rank}(\mathbf{M}) = 6$.

The *proof* of this theorem will be available in the author's page.

Note that we can choose the transformation parameters $\widetilde{\mathbf{R}}$ and $\widetilde{\mathbf{t}}$, such that the plane does not pass through the origin. For that, we have to ensure that the $\widetilde{\zeta}^{(\mathscr{W})} \neq 0$.

To solve this problem, we define a matrix $\mathbf{N} \in \mathbb{R}^{9\times 10}$ such that the solution vector $\operatorname{vec}(\mathbf{H})$ is represented in homogeneous coordinates $\xi \in \mathbb{R}^{10}$, and such that

$$\underbrace{\begin{bmatrix} \mathbf{M} & -\mathbf{w} \end{bmatrix}}_{\mathbf{N}} \xi = \mathbf{0}, \quad \text{where} \quad \xi = \begin{bmatrix} \operatorname{vec}(\mathbf{H}) \\ 1 \end{bmatrix}. \tag{15}$$

From linear algebra the dimension of the column-space of the matrix $\mathbf{N}$ is $\operatorname{rank}(\mathbf{N}) = \operatorname{rank}(\mathbf{M})$, which means that from Theorem 1, $\operatorname{rank}(\mathbf{N}) = 6$.

It is well known from linear algebra – see [23], that $\operatorname{rank}(\mathbf{N}) + \operatorname{nullity}(\mathbf{N}) = 10$. Thus, we conclude that the dimension of the null-space of $\mathbf{N}$ is $\operatorname{nullity}(\mathbf{N}) = 4$. Moreover, from (15), we conclude that $\xi \subset \operatorname{null}(\mathbf{N})$. Since the dimension of the null-space of $\mathbf{N}$ is four, we define

$$\operatorname{null}(\mathbf{N}) \doteq \{\alpha_1 \mathbf{e}_1 + \alpha_2 \mathbf{e}_2 + \alpha_3 \mathbf{e}_3 + \alpha_4 \mathbf{e}_4 : \alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{R}\} \tag{16}$$

where the $\mathbf{e}_i \in \mathbb{R}^{10}$ are the vectors defining the basis of the null-space. However, from (15), we see that the tenth element of vector ξ must be $\xi_{10} = 1$. Note that the basis $\mathbf{e}_i$ are defined up to a scale factor. As a result, and without loss of generality, we can consider that the tenth element of all $\mathbf{e}_i$ is equal to one. Using this result, and to ensure that $\xi_{10} = 1$, we define the following constraint

$$\alpha_4 = 1 - \alpha_1 - \alpha_2 - \alpha_3. \tag{17}$$

Redefining the basis as $\widetilde{\mathbf{e}}_1 = \mathbf{e}_1 - \mathbf{e}_4$, $\widetilde{\mathbf{e}}_2 = \mathbf{e}_2 - \mathbf{e}_4$, $\widetilde{\mathbf{e}}_3 = \mathbf{e}_3 - \mathbf{e}_4$ and $\mathbf{e}_4 = \mathbf{e}_4$, we define a three dimensional affine space $\mathscr{Q}$

$$\mathscr{Q} \doteq \{\alpha_1 \widetilde{\mathbf{e}}_1 + \alpha_2 \widetilde{\mathbf{e}}_2 + \alpha_3 \widetilde{\mathbf{e}}_3 + \widetilde{\mathbf{e}}_4 : \alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}\} \tag{18}$$

such that $\xi \in \mathscr{Q}$. To compute the basis $\widetilde{\mathbf{e}}_i$, we can use SVD to estimate $\mathbf{e}_i$ and derive $\widetilde{\mathbf{e}}_i$ as suggested. On the other hand, we can also use an analytical solution as the one derived in the Appendix .

Defining the matrices $\widetilde{\mathbf{E}}_i \in \mathbb{R}^{3\times 3}$ as the un–stacking matrices of the vectors $\widetilde{\mathbf{e}}_i$, we can define $\mathbf{H}$ as a function of the unknowns α_1, α_2 and α_3 such that

$$\mathbf{H} = \alpha_1 \widetilde{\mathbf{E}}_1 + \alpha_2 \widetilde{\mathbf{E}}_2 + \alpha_3 \widetilde{\mathbf{E}}_3 + \widetilde{\mathbf{E}}_4. \tag{19}$$

Let us consider the vectors $\mathbf{f}_i^{(j)}$ as the i^{th} column of the matrix $\widetilde{\mathbf{E}}_j$. Using these vectors, we define two affine spaces for the two first columns of the aimed matrix $\mathbf{H}$ as

$$\mathbf{h}_1 = \alpha_1 \mathbf{f}_1^{(1)} + \alpha_2 \mathbf{f}_1^{(2)} + \alpha_3 \mathbf{f}_1^{(3)} + \mathbf{f}_1^{(4)} \tag{20}$$

$$\mathbf{h}_2 = \alpha_1 \mathbf{f}_2^{(1)} + \alpha_2 \mathbf{f}_2^{(2)} + \alpha_3 \mathbf{f}_2^{(3)} + \mathbf{f}_2^{(4)}. \tag{21}$$

Since the matrix $\mathbf{H}$ must be as (7), the constraints defined by (8), (9) and (10) can apply to both $\mathbf{h}_1$ and $\mathbf{h}_2$. Thus, three constraints of the type $g_{i,j}(\alpha_1, \alpha_2, \alpha_3) = 0$ can be derived and the solutions for the unknowns α_i are given by

$$g_{1,1}(\alpha_1, \alpha_2, \alpha_3) = g_{2,2}(\alpha_1, \alpha_2, \alpha_3) = g_{1,2}(\alpha_1, \alpha_2, \alpha_3) = 0, \tag{22}$$

where each function $g_{i,j}(\alpha_1, \alpha_2, \alpha_3)$ is such that

$$g_{i,j}(\alpha_1, \alpha_2, \alpha_3) = \alpha_1^2 \kappa_1^{(i,j)} + \alpha_2^2 \kappa_2^{(i,j)} + \alpha_3^2 \kappa_3^{(i,j)} + \alpha_1 \alpha_2 \kappa_4^{(i,j)} + \\ + \alpha_1 \alpha_3 \kappa_5^{(i,j)} + \alpha_2 \alpha_3 \kappa_6^{(i,j)} + \alpha_1 \kappa_7^{(i,j)} + \alpha_2 \kappa_8^{(i,j)} + \alpha_3 \kappa_9^{(i,j)} + \kappa_{10}^{(i,j)}. \tag{23}$$

According to Nister et al. at [18], the minimal absolute pose problem requires a solution of a single variable polynomial equation with degree no less than eight. From *Bézout*'s theorem [4], we can conclude that the problem of (22) can have up to eight solutions for $(\alpha_1, \alpha_2, \alpha_3)$, which correspond to points where three quadrics intersect. Moreover, as shown by Guo at [7], it is possible to derive an eight degree polynomial equation for the problem of (22) in closed-form, which means that finding the points where where three quadrics intersect has the same computational complexity as state-of-the-art approaches (finding the roots of an eight degree polynomial equation).

It is also possible to use conventional methods such as *Gröebner basis* [22], [12], *hidden variable technique* [9] or *polynomial eigenvalue* [13]. Note that the aim of this paper is not to develop a minimal solver but a new parameterization of the problem.

B. *Decomposition of Matrix* H

Note that as a result of the pre-defined transformation of the data set, each solution for $\mathbf{H}$ will verify (7). Therefor and since the rotation matrix must belong to the special orthogonal group, the rotation matrix to be estimated will be

$$\mathbf{R} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & (\mathbf{h}_1 \times \mathbf{h}_2) \end{bmatrix} \tag{24}$$

where the column vectors $\mathbf{h}_i$ correspond to the i^{th} columns of the estimated matrix $\mathbf{H}$. Moreover, from (7) and since we already know the rotation matrix, we can get the coordinates of the translation vector as

$$\mathbf{t} = \widetilde{\zeta}^{(\mathscr{W})} (\mathbf{h}_3 - \mathbf{h}_1 \times \mathbf{h}_2). \tag{25}$$

Note that as a result of the application of a pre–defined transformation to the data set, the rotation and translation defined by the decomposition of $\mathbf{H}$ matrix will not represent the pose of the camera. We have to take into account the transformation parameters defined by $\widetilde{\mathbf{R}}$ and $\widetilde{\mathbf{t}}$. Consequently, we define the pose parameters by the rotation matrix $\mathbf{R}_{\text{out}}$ and translation vector $\mathbf{t}_{\text{out}}$, such that

$$\mathbf{R}_{\text{out}} = \mathbf{R}\widetilde{\mathbf{R}} \text{ and } \mathbf{t}_{\text{out}} = \mathbf{R}\widetilde{\mathbf{t}} + \mathbf{t}. \tag{26}$$

III. Experimental Results

To evaluate the proposed method against the state-of-the-art algorithms, we considered synthetic data-sets. We analyze the numerical errors, and the number of solutions. In addition, and to assess the robustness of each method, we consider three critical configurations. We consider two linear cameras and an orthogonal camera – the orthogonal camera is a degenerate case, and compute the pose for configurations close to these cases.

Since there are no comparisons between the previous algorithms, we perform the evaluation using the following approaches:

- `Our` – denotes the method described in this paper when $\text{null}(\mathbf{N})$ is computed using iterative methods;
- `Our - CfN` – denotes the method described in this paper when $\text{null}(\mathbf{N})$ is computed using the analytical solution presented in the Appendix ;
- `Nister and Stewenius` – denotes the method described in [17], [19];
- `Chen and Chang` – denotes the method described in [2], [3] where the SVD – required for [1], [25], is computed using iterative methods;
- `Chen and Chang - CfSVD` – the same as `Chen and Chang` with the SVD computed using analytical solutions;

For that purpose we consider a cube with 200 units of side length. We consider lines defined by a point and a direction. Three points belonging to line $\mathbf{x}_i^{(\mathscr{C})}$ were computed inside the cube. Random directions $\mathbf{d}_i^{(\mathscr{C})}$ were computed – where $\left|\mathbf{d}_i^{(\mathscr{C})}\right| = 1$. Three depths λ_i, ranging between 20 and 500 were randomly generated and the coordinates of 3D points in the camera coordinate system are computed using $\mathbf{p}_i^{(\mathscr{C})} = \lambda_i \mathbf{d}_i^{(\mathscr{C})} + \mathbf{x}_i^{(\mathscr{C})}$. A rigid transformation was randomly generated ($\mathbf{R} \in \mathscr{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$). The rigid transformation generated was applied to the set of points so that $\mathbf{p}_i^{(\mathscr{C})} \mapsto \mathbf{p}_i^{(\mathscr{W})}$. For the data set $\left\{\mathbf{l}_i^{(\mathscr{C})}, \mathbf{p}_i^{(\mathscr{W})}\right\}$ pose is estimated using the corresponding algorithms. Note that we can easily get the *Plücker* coordinates from a point and a direction [20].

This procedure is repeated for 10^6 trials where, for each trial, a new pose and data are randomly generated.

Solutions of the minimal absolute pose problem must yield similar results in both the number of solutions and the for the values of the parameters. As a result and as was emphasized and discussed by Nistér and Stewénius [19], tests with noisy data are not relevant for comparison. For experiments with real data, the same analysis can be made.

A. Numerical Errors and Number of Solutions

For each of the 10^6 trials we evaluate the error in the rotation parameters by computing the square root of the sum of the error on the rotation angles. The numerical error for the translation is computed using the norm of the vector $\mathbf{t}_{\text{out}} - \mathbf{t}_{\text{gt}}$. The distribution of the numerical errors on the values of the rotation and translation are shown in Figs. 2(a).

To eliminate abnormal solutions, it is usual to consider the constraint that points should be in forward direction. In the framework of generalized camera models, we consider this constraint as $\mathbf{d}_i^{(\mathscr{C})\,T}\left(\mathbf{p}_i^{(\mathscr{C})} - \mathbf{x}_i^{(\mathscr{C})}\right) > 0$ for $i = 1,2,3$, where: $\mathbf{p}_i^{(\mathscr{C})}$ is the i^{th} estimated 3D point in the camera coordinate system; and $\mathbf{x}_i^{(\mathscr{C})}$ is the i^{th} point in the camera coordinate system that defines the forward direction. In the experiments, we call this constraint the "Orientation Constraint". The result of the application of the "Orientation Constraint" is shown in Fig. 2(b).

B. Critical Configurations

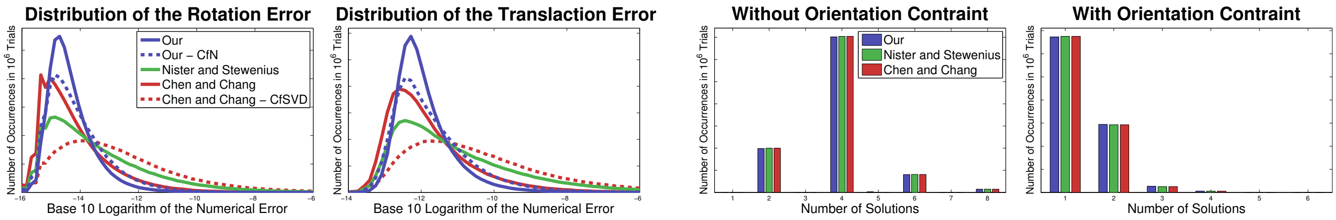
We consider following critical cases: *orthographic* configuration (affine camera) [10] where the pose is degenerate – there are an infinite number of solutions; the classical *linear pushbroom* configuration; and the X-Slits cameras. For that purpose, we consider the same data-set generation previously defined. However, instead of considering general direction and 3D points: $\mathbf{d}_i^{(\mathscr{C})}$ and $\mathbf{x}_i^{(\mathscr{C})}$, we constrain this random data to the proposed configurations. We compute three vectors $\mathbf{v}_i^{(\mathscr{C})}$ with random directions and the norm randomly destributed over a normal distrubution with standard deviation defined by a variable called `Distance from Critical Case`. The 3D points in the camera coordinate system are then given by $\mathbf{p}_i^{(\mathscr{C})} = \lambda_i \breve{\mathbf{d}}_i^{(\mathscr{C})} + \mathbf{x}_i^{(\mathscr{C})}$, where the directions of the lines are $\breve{\mathbf{d}}_i^{(\mathscr{C})} = \mathbf{d}_i^{(\mathscr{C})} + \mathbf{v}_i^{(\mathscr{C})}$.

As the variable `Distance from Critical Case` goes from one to zero, the trial approximates the critical configuration. From each value for this variable, we compute the median of 10^3 trials. If some algorithm fails, we discard that trial and randomly generate a new one. This procedure is repeated until we get 10^3 trials. The results are shown in Fig. 3. In addition to the errors in the rotation and translation parameters, we show the number of failures for each one of the algorithms (occurred until 10^3 valid trials were obtained).

IV. Discussion

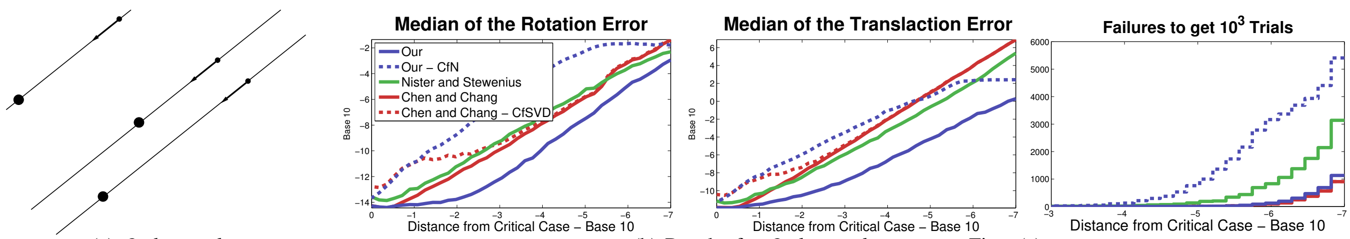
The comparison in terms of number of solutions (Fig. 2(b)) does not add relevant information concerning the relative merits of the algorithms.

In terms of numerical accuracy (Fig. 2(a)), it can be seen that four of the five suggested algorithms perform similarly. We note that our method performs slightly better when taking into account the variation of the distribution of the errors, specially for the case where the matrix $\mathbf{N}$ is computed using SVD. The algorithm `Chen and Chang - CfSVD` performs the worst, specially due to a larger variation on the distribution of the numerical errors.

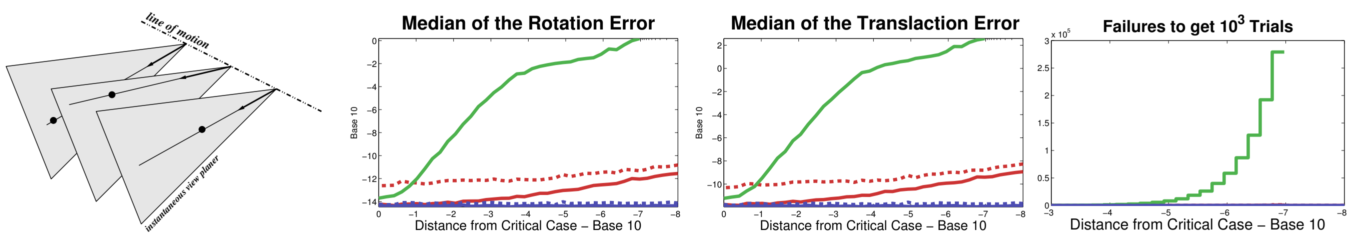


(a) Evaluation for the numerical accuracy. (b) Results for the number os solutions.

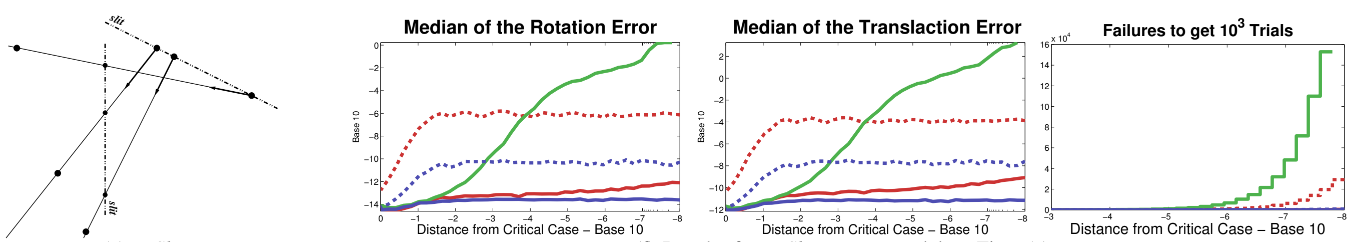
Fig. 2. In this figure we show the numerical distribution of the errors – Fig. (a), and the distribution of the number of solutions – Fig. (b). The proposed method is compared against the state-of-the-art algorithms proposed by Nistér and Stewénius at [17], [19] and by Chen and Chang at [2], [3]. More information abount the evaluated algorithms is given in the text.



(a) *Orthogonal* (b) Results for *Orthogonal* cameras – Figs. (a).



(c) *Pushbroom* (d) Rotation & translation errors for *Pushbroom* cameras – Figs. (c).



(e) *X-Slit* (f) Results for *X-Slit* camera models – Figs. (e).

Fig. 3. In this figure we evaluate the stability and robustness of the proposed algorithms. We consider the algorithms that we used in previous tests. For that, we consider three critical cases: *orthogonal* configuration (a) and (b); *pushbroom* cameras (c) and (d); and *X-Slit* cameras (e) and (f). The stability and robustness are analyzed by computing the errors in the rotation and translation parameters that define the pose, as a function of a `Distance from Critical Case` – see the Sec. III-B. Note that the errors are shown in a log-base 10 representation. The errors are computed as the median of 10^3 valid trials. Close to critical cases it is possible that some algorithms fail to compute a solution. As a result, and for each algorithm, we show the number of failures that we get until 10^3 valid trials are obtained.

A. Computational Effort

We have implemented all the algorithms in MATLAB using a *Intel core i7–3930k* with a 3.2*GHz* processor. All the computation times that are shown in the paper were computed as the median of the respective computation times, for all the 10^6 trials. For the evaluation, we will consider only the steps that require the most significant computational effort, which correspond to the iterative steps.

The main computation step for all the algorithms consists in the computation of the minimal solver which, for all the algorithms, corresponds to finding the roots of an eight degree polynomial equation. As suggested by both Nistér & Stewénius and Chen & Chang, the *companion matrix* is suitable in terms of both speed and accuracy. This method corresponds to performing an *eigen decomposition* of an 8×8 matrix, which takes $43\mu s$.

For our parameterization, we considered two possibilities for the estimation of the null-space of the matrix $\mathbf{N}\in\mathbb{R}^{9\times 10}$, (15): using an analytical solution (`Our - CfN`); and using the iterative SVD (`Our`). If we use the latter, we have to take into account $40\mu s$.

Note that the Chen & Chang formulation estimates the coordinates of the 3D points in the camera coordinate system and since the pose is given by the rotational and translational parameters (that define the transformation between the world and camera coordinate systems) it is therefore necessary, for each valid solution, the computation of a SVD. If we use iterative methods (`Chen & Chang`), each valid solution takes $19\mu s$.

To conclude, in Table I we present figures representing the main computational load required by all the five algorithms evaluated. From this table, we conclude that the method that we denote as `Our - CfN` method requires the same computational effort as the fastest method (`Nister and Stewenius`) , `Our` is faster than the `Chen & Chang` algorithm in the case where the number of possible solutions is bigger then two (which, according to Fig. 2(b), happens in most of the cases).

B. Critical Camera Models

Let us consider general calibration methods, such as [6], [24], [15], to calibrate *X-Slit* cameras – Fig. 3(e). For noisy data, the 3D lines generated from the calibrated *general camera model* will not pass through the two slits that define the camera geometry. However, and again depending on the quality of the calibration, they must pass close to these slits. The same analysis can be made for *Pushbroom* cameras – Fig. 3(c). For data with noise, the 3D lines will not pass through the line of motion nor will belong the the instantaneous view plane. As a result, the solutions for the minimal pose have to be computed using a general method, such as the one presented in this article.

As we can see from the results, we conclude the method denoted as `Our` behaves better than the sate–of–the–art methods in all the tests. `Our - CfN` only performs worst when compared with the `Chen & Chang` method and for the `X-Slit` critical configuration. However, we note that `Chen & Chang` method is significantly slower than `Our - CfN` method – it requires the computation of an iterative SVD for each valid solution. We note that `Nister & Stewenius` method performs up to 10^{13} worst than the algorithm `Our`.

In additional to the previous previous configurations, we consider the *Orthographic* critical configuration. Pose estimation for these cameras is degenerate. As we can see from the results, the median of the numerical errors for the algorithm `Our` are up to 10^6 times better, relative to all the other algorithms. Note that when method `Our` has a median of the errors close to one, the other algorithms have values close to 10^6, which is a significant difference. However and for the case where the null-space of $\mathbf{N}$ is computed using the analytical solution (`Our - CfN`), the results are not as god as `Our` algorithm. Moreover and in most of the cases, the approach performs slightly worst than state-of-the-art methods. We also note that other analytical solutions for the null-space can be derived, which can improve these results.

To conclude, we remark that the parameterization proposed in this article is significantly more robust than the state-of-the-art formalizations.

V. Conclusions

In this article we proposed a novel parameterization for the minimal absolute pose problem within the framework of generalized camera models. In terms of formalization, our parameterization is very simple and gives the rotation and translation parameters directly. We note that this parameterization can be easily changed to allow a closed-form solution for central cameras – that also gives the rotation and translation parameters directly as the recent method proposed by Kneip et al. [11].

The main contribution of the proposed parameterization is its robustness in the cases of imaging devices with critical configurations. Note that when considering both the general camera model and the associated calibration procedure, it is very important to analyze the robustness of the proposed solvers in the cases of non-central imaging systems– specially when considering noisy data. From the results shown in the paper, we see that our method performs significantly better than state-of-the-art methods, specially when considering the algorithms with the smallest computational effort.

The Matlab code will be avaliable in the author's web page.

Appendix

In this appendix we derive an analytical solution for the basis of $\text{null}(\mathbf{N})$. This solution should be used when the computation time is essential. From the definition of the *Kronecker* product we can rewrite the matrix $\mathbf{M}$, (14), as

$$\mathbf{M} = \begin{bmatrix} \tilde{p}_{1,1}^{(\mathscr{W})}\hat{\mathbf{d}}_1^{(\mathscr{C})} & \tilde{p}_{1,2}^{(\mathscr{W})}\hat{\mathbf{d}}_1^{(\mathscr{C})} & \tilde{p}_{1,3}^{(\mathscr{W})}\hat{\mathbf{d}}_1^{(\mathscr{C})} \\ \tilde{p}_{2,1}^{(\mathscr{W})}\hat{\mathbf{d}}_2^{(\mathscr{C})} & \tilde{p}_{2,2}^{(\mathscr{W})}\hat{\mathbf{d}}_2^{(\mathscr{C})} & \tilde{p}_{2,3}^{(\mathscr{W})}\hat{\mathbf{d}}_2^{(\mathscr{C})} \\ \tilde{p}_{3,1}^{(\mathscr{W})}\hat{\mathbf{d}}_3^{(\mathscr{C})} & \tilde{p}_{3,2}^{(\mathscr{W})}\hat{\mathbf{d}}_3^{(\mathscr{C})} & \tilde{p}_{3,3}^{(\mathscr{W})}\hat{\mathbf{d}}_3^{(\mathscr{C})} \end{bmatrix} \quad (27)$$

where $\tilde{p}_{i,j}^{(\mathscr{W})}$ is the j^{th} element of the vector $\tilde{\mathbf{p}}_i^{(\mathscr{W})}$.

Let us define the vectors $\mathbf{q}_i^{(\mathscr{W})}$ – orthogonal to both $\tilde{\mathbf{p}}_j^{(\mathscr{W})}$ and $\tilde{\mathbf{p}}_k^{(\mathscr{W})}$,

$$\mathbf{q}_1^{(\mathscr{W})} = \tilde{\mathbf{p}}_2^{(\mathscr{W})} \times \tilde{\mathbf{p}}_3^{(\mathscr{W})},\ \mathbf{q}_2^{(\mathscr{W})} = \tilde{\mathbf{p}}_1^{(\mathscr{W})} \times \tilde{\mathbf{p}}_3^{(\mathscr{W})} \text{ and } \mathbf{q}_3^{(\mathscr{W})} = \tilde{\mathbf{p}}_1^{(\mathscr{W})} \times \tilde{\mathbf{p}}_2^{(\mathscr{W})}, \quad (28)$$

Using these vectors, the fact that $\mathbf{d}_i^{(\mathscr{C})} \times \mathbf{d}_i^{(\mathscr{C})} = \mathbf{0}$ and since the null-space of $\mathbf{M}$ has dimension equal to three – Theorem 1, we define the three basis vectors for $\text{null}(\mathbf{M})$ as

$$\mathbf{e}_1 = \begin{bmatrix} q_{1,1}^{(\mathscr{W})}\mathbf{d}_1^{(\mathscr{C})}, & q_{1,2}^{(\mathscr{W})}\mathbf{d}_1^{(\mathscr{C})}, & q_{1,3}^{(\mathscr{W})}\mathbf{d}_1^{(\mathscr{C})} \end{bmatrix} \quad (29)$$

$$\mathbf{e}_2 = \begin{bmatrix} q_{2,1}^{(\mathscr{W})}\mathbf{d}_2^{(\mathscr{C})}, & q_{2,2}^{(\mathscr{W})}\mathbf{d}_2^{(\mathscr{C})}, & q_{2,3}^{(\mathscr{W})}\mathbf{d}_2^{(\mathscr{C})} \end{bmatrix} \quad (30)$$

$$\mathbf{e}_3 = \begin{bmatrix} q_{3,1}^{(\mathscr{W})}\mathbf{d}_3^{(\mathscr{C})}, & q_{3,2}^{(\mathscr{W})}\mathbf{d}_3^{(\mathscr{C})}, & q_{3,3}^{(\mathscr{W})}\mathbf{d}_3^{(\mathscr{C})} \end{bmatrix} \quad (31)$$

where $q_{i,j}^{(\mathscr{W})}$ is the j^{th} element of the vector $\mathbf{q}_i^{(\mathscr{W})}$. It can be seen that these bases are linearly independent.

Let us now consider matrix $\mathbf{N}$ as described in (15). From the basis for the null-space of $\mathbf{M}$ – $\mathbf{e}_i$, (29-31), and from definition of matrix $\mathbf{N}$, we define $\tilde{\mathbf{e}}_i = [\mathbf{e}_i, 0]$ for $i = 1,2,3$. It can be seen that $\tilde{\mathbf{e}}_i$, for $i = 1,2,3$, are three linearly independent basis for the null-space of $\mathbf{N}$. However and since the dimension of the null-space must be four, there is one basis left.

TABLE I

MAIN COMPUTATIONAL EFFORT REQUIRED FOR THE COMPUTATION OF THE PROPOSED ALGORITHMS. K – REPRESENTS THE NUMBER OF VALID SOLUTIONS GIVEN BY THE ALGORITHMS.

Methods:	Our	Our - CfN	Nister & Stewenius	Chen & Chang	Chen & Chang - CfSVD
Times	$43+40\mu s$	$43\mu s$	$43\mu s$	$43+K19\mu s$	$43\mu s$

From geometric properties, we know that $\mathbf{m}_i^{(\mathscr{C})} = \mathbf{x}_i^{(\mathscr{C})} \times \mathbf{d}_i^{(\mathscr{C})}$, for any point $\mathbf{x}_i^{(\mathscr{C})}$ that belongs to the line. As a result and considering vector $\mathbf{q}_i^{(\mathscr{W})}$ defined in (28), we derived the following three vectors

$$\widetilde{\mathbf{e}}_4^{(1)} = \begin{bmatrix} \breve{q}_{1,1}^{(\mathscr{W})}\mathbf{x}_1^{(\mathscr{C})}, & \breve{q}_{1,2}^{(\mathscr{W})}\mathbf{x}_1^{(\mathscr{C})}, & \breve{q}_{1,3}^{(\mathscr{W})}\mathbf{x}_1^{(\mathscr{C})}, & 1 \end{bmatrix} \quad (32)$$

$$\widetilde{\mathbf{e}}_4^{(2)} = \begin{bmatrix} \breve{q}_{2,1}^{(\mathscr{W})}\mathbf{x}_2^{(\mathscr{C})}, & \breve{q}_{2,2}^{(\mathscr{W})}\mathbf{x}_2^{(\mathscr{C})}, & \breve{q}_{2,3}^{(\mathscr{W})}\mathbf{x}_2^{(\mathscr{C})}, & 1 \end{bmatrix} \quad (33)$$

$$\widetilde{\mathbf{e}}_4^{(3)} = \begin{bmatrix} \breve{q}_{3,1}^{(\mathscr{W})}\mathbf{x}_3^{(\mathscr{C})}, & \breve{q}_{3,2}^{(\mathscr{W})}\mathbf{x}_3^{(\mathscr{C})}, & \breve{q}_{3,3}^{(\mathscr{W})}\mathbf{x}_3^{(\mathscr{C})}, & 1 \end{bmatrix} \quad (34)$$

where

$$\breve{q}_{i,j}^{(\mathscr{W})} = \frac{q_{i,j}^{(\mathscr{W})}}{q_{i,1}^{(\mathscr{W})}\widetilde{p}_{i,1}^{(\mathscr{W})} + q_{i,2}^{(\mathscr{W})}\widetilde{p}_{i,2}^{(\mathscr{W})} + q_{i,3}^{(\mathscr{W})}\widetilde{p}_{i,3}^{(\mathscr{W})}}. \quad (35)$$

Using these vectors, we have

$$\mathbf{N}\widetilde{\mathbf{e}}_4^{(1)} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{m}_2^{(\mathscr{C})} \\ -\mathbf{m}_3^{(\mathscr{C})} \end{bmatrix}, \mathbf{N}\widetilde{\mathbf{e}}_4^{(2)} = \begin{bmatrix} -\mathbf{m}_1^{(\mathscr{C})} \\ \mathbf{0} \\ -\mathbf{m}_3^{(\mathscr{C})} \end{bmatrix}, \mathbf{N}\widetilde{\mathbf{e}}_4^{(3)} = \begin{bmatrix} -\mathbf{m}_1^{(\mathscr{C})} \\ -\mathbf{m}_2^{(\mathscr{C})} \\ \mathbf{0} \end{bmatrix}. \quad (36)$$

As a result and taking into account (15), we define the last basis for the null-space of $\mathbf{N}$ as

$$\widetilde{\mathbf{e}}_4 = \widetilde{\mathbf{e}}_4^{(1)} + \widetilde{\mathbf{e}}_4^{(2)} + \widetilde{\mathbf{e}}_4^{(3)} + [0,\ldots,0,-2]^T. \quad (37)$$

The null-space for the matrix $\mathbf{N}$ is thus given by

$$\operatorname{null}(\mathbf{N}) \doteq \{\alpha_1\widetilde{\mathbf{e}}_1 + \alpha_2\widetilde{\mathbf{e}}_2 + \alpha_3\widetilde{\mathbf{e}}_3 + \alpha_4\widetilde{\mathbf{e}}_4 \;:\; \alpha_i \in \mathbb{R}, \forall i\}. \quad (38)$$

Note that the tenth element of $\widetilde{\mathbf{e}}_1$, $\widetilde{\mathbf{e}}_2$ $\widetilde{\mathbf{e}}_3$ and $\widetilde{\mathbf{e}}_4$ are 0, 0, 0 and 1 respectively. From Sec. II-A, we want $\xi \in \operatorname{null}(\mathbf{N})$ such that $\xi_{10} = 1$. Using the solution for the null-space described in this appendix, (38), we get this constraint by ensuring that $\alpha_4 = 1$.

REFERENCES

[1] K. S. Arun, T. S. Huang, and S. D. Blostein. Least–Square Fitting of Two $3-D$ Point Sets. IEEE Trans. Pattern Analysis and Machine Intelligence, 1987.

[2] C.-S. Chen and W.-Y. Chang. Pose Estimation for Generalized Imaging Device via Solving Non–Prespective n Point Problem. IEEE Proc. Int'l Conf. Robotics and Automation, 2002.

[3] C.-S. Chen and W.-Y. Chang. On Pose Recovery for Generalized Visual Sensers. IEEE Trans. Pattern Analysis and Machine Intelligence, 2004.

[4] D. A. Cox, J. Little, and D. O'Shea. Using Algebraic Geometry. Springer Science+Business, 2004.

[5] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applicatlons to Image Analysis and Automated Cartography. Comm. of the ACM, 1981.

[6] G. Grossberg and S. Nayar. A General Imaging Model and a Method for Finding its Parameters. IEEE Proc. Int'l Conf. Computer Vision, 2001.

[7] Y. Guo. A Novel Solution to the P4P Problem for an Uncalibrated Camera. J. Math. Imaging Vis, 2012.

[8] R. Gupta and R. I. Hartley. Linear Pushbroom Cameras. IEEE Trans. Pattern Analysis and Machine Intelligence, 1997.

[9] R. Hartley and H. Li. An Efficient Hidden Variable Approach to Minimal-Case Camera Motion Estimation. IEEE Trans. Pattern Analysis and Machine Intelligence, 2012.

[10] R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, 2000.

[11] L. Kneip, D. Scaramuzza, and R. Siegwart. A Novel Parametrization of the Perspective–Three–Point Problem for a Direct Computation of Absolute Camera Position and Orientation. IEEE Proc. Computer Vision and Patter Recognition, 2011.

[12] Z. Kukelova, M. Bujnak, and T. Pajdla. Automatic Generator of Minimal Problem Solvers. Proc. European Conf. Computer Vision, 2008.

[13] Z. Kulelova, M. Bujnak, and T. Pajdla. Polynomial Eigenvalue Solutions to Minimal Problems in Computer Vision. IEEE Trans. Pattern Analysis and Machine Intelligence, 2012.

[14] Y. Ma, S. Soatto, J. Košecká, and S. S. Sastry. An Invitation to 3–D Vision: From Images to Geometry Models. Springer Science+Business, 2004.

[15] P. Miraldo, H. Araujo, and J. Queiró. Point–based Calibration Using a Parametric Representation of the General Imagin Model. IEEE Proc. Int'l Conf. Computer Vision, 2011.

[16] D. Nistér. Preemptive RANSAC for Live Structure and Motion Estimation. IEEE Proc. Int'l Conf. Computer Vision, 2003.

[17] D. Nistér. A Minimal Solution to the Generalized 3–Point Pose Problem. IEEE Proc. Computer Vision and Patter Recognition, 2004.

[18] D. Nistér, R. Hartley, and H. Stewénius. Using Galois Theory to Prove Structure from Motion Algorithms are Optimal. IEEE Proc. Computer Vision and Patter Recognition, 2007.

[19] D. Nistér and H. Stewénius. A Minimal Solution to the Generalized 3–Point Pose Problem. J. Math. Imaging Vis, 2007.

[20] H. Pottmann and J. Wallner. Computational Line Geometry. Springer–Verlag, 2001.

[21] S. Ramalingam, S. K. Lodha, and P. Sturm. A Generic Structure–from–Motion Framework. Computer Vision and Image Understanding, 2006.

[22] H. Stewénius. Gröbner Basis Methods for Minimal Problems in Computer Vision. PhD Thesis, Lund University, 2005.

[23] G. Strang. Linear Algebra and its Applications. Academic Press, 1980.

[24] P. Sturm and S. Ramalingam. A Generic Concept for Camera Calibration. Proc. European Conf. Computer Vision, 2004.

[25] S. Umeyama. Least–Square Estimation of Transformation Between Two Point Patterns. IEEE Trans. Pattern Analysis and Machine Intelligence, 1991.

[26] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing New Views: The Crossed–Slits Projection. IEEE Trans. Pattern Analysis and Machine Intelligence, 2003.