



ARTICLE: **Direct Solution to the Minimal Generalized Pose**

Author:
Pedro MIRALDO
miraldo@isr.uc.pt

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

CONTENTS

I	Introduction	2
I-A	General Imaging Model	3
I-B	Previous Approaches vs Our Approach	3
I-C	Background and Notation	4
II	Proposed Approach	4
II-A	Decomposition of the <i>Homography</i> Matrix	4
II-B	Formalization of the problem	5
II-C	The Case of a Central Camera	6
III	Solving for the Intersection of Quadrics	6
IV	Algorithm Outline	7
V	Experimental Results	7
V-A	Experiments with Synthetic Data	7
V-A1	Numerical Errors	8
V-A2	Number of Solutions	9
V-A3	Computational Effort	9
V-B	Experimental Results with Real Data	10
V-C	Analysis of the Experimental Results	11
VI	Conclusions	11
	References	12
	Biographies	13
	Pedro Miraldo	13
	Helder Araujo	13

Direct Solution to the Minimal Generalized Pose

Pedro Miraldo and Helder Araujo

Abstract—Pose estimation is a relevant problem for imaging systems whose applications range from augmented reality to robotics. In this paper we propose a novel solution for the minimal pose problem, within the framework of generalized camera models and using a planar homography. Within this framework and considering only the geometric elements of the generalized camera models, an imaging system can be modeled by a set of mappings associating image pixels to 3D straight lines. This mapping is defined in a 3D world coordinate system. Pose estimation performs the computation of the rigid transformation between the original 3D world coordinate system and the one in which the camera was calibrated. Using synthetic data, we compare the proposed minimal-based method with the state-of-the-art methods in terms of numerical errors, number of solutions and processing time. From the experiments, we conclude that the proposed method performs better, especially because there is a smaller variation in numerical errors, while results are similar in terms of number of solutions and computation time. To further evaluate the proposed approach we tested our method with real data. One of the relevant contributions of this paper is theoretical. When compared to the state-of-the-art approaches, we propose a completely new parametrization of the problem that can be solved in four simple steps. In addition, our approach does not require any predefined transformation of the dataset, which yields a simpler solution for the problem.

Index Terms—Minimal problems in computer vision, absolute pose, generalized camera models, homography matrix.

I. INTRODUCTION

Camera calibration within the framework of the generalized camera models consists of estimating the correspondences between image pixels and the corresponding projecting 3D straight lines. Image space does not usually change and therefore we can define a 3D coordinate system for the image coordinates. But a camera is a mobile device and so we cannot define a fixed global coordinate system to represent the lines mapped into the image points. Therefore, we define a 3D reference coordinate system associated with the camera to represent the 3D lines mapped into the image pixels [1]. As a consequence, to estimate the coordinates of 3D entities represented in a different coordinate system we need to estimate a rigid transformation that maps the camera coordinate system into the other reference coordinate system, which we call the absolute pose problem. This problem can be divided into two subproblems: minimal and non-minimal cases.

Absolute pose is important for many applications that need to use the 3D entities imaged by a camera. The analysis of solutions using minimal data is important because it provides insights and a better understanding of the problem. Non-minimal solutions, meanwhile, are used to estimate the pose

for more than the minimal point correspondences, when we need a robust solution. The advantages of minimal solutions are in terms of computation speed and their applications in the hypothesis-and-test estimation methods such as RANSAC [2], [3].

Pose estimation has been subject of a significant amount of research for the case of any camera that can be modeled by a common perspective imaging system, where all the rays intersect at a single point (Fig. 1(b)). These works [4], [5], [6], [7], [8] are examples for non-minimal pose problems, and in the case of minimal pose problems these works [9], [10], [11] (only three 3D points in the world and their images are used) are also examples. Guo and Bujnak *et al.*, in [12], [13], developed a minimal solution for the computation of the pose of an uncalibrated perspective camera. In this case, instead of three 3D points, their method requires four 3D points and their images.

If we consider imaging devices where image formation cannot be modeled by a central projection (see Fig. 1(a)), e.g. when the imaging rays are subject to reflection and/or refraction [14], [15], [16], [17], [18], [19] or linear cameras [20], [21], [22], [23], new methods and algorithms have to be considered. In this article we address minimal pose estimation for general imaging devices that uses a single image of three points in the scene. We do not address the relative pose problem where two or more images of the scene are used. In general, estimates based on minimal data provide multiple solutions. Here we describe a new method for pose estimation using the minimal number of three known coordinates of 3D points. As far as we know, this problem has been studied by Nistér & Stewénus [24], [25], by Chen & Chang [26], [27] and by Ramalingam *et al.* at [28]. A more detailed analysis of these methods is presented in the next section.

We use a planar 3D homography transformation to define an algebraic relationship that relates any three points in the world (represented in the world coordinate system) and any three 3D lines that pass through those points (the lines are represented in the 3D camera coordinate system). Note that, since we are using only three 3D points, the use of a planar homography to represent the rigid transformation does not restrict the use of the method to planar patterns. Any three linearly independent 3D points define a plane in the world. However, as we will see (and without additional constraints), the use of three points and three lines leads to an infinite number of possible solutions. To have a finite number of solutions, the constraint that the homography matrix must define a rigid transformation is used.

In this article, existing and proposed approaches are also compared. The proposed method performs better than the previous approaches in terms of numerical error and is similar to the best state-of-the-art method in terms of the number of solutions and computation effort.

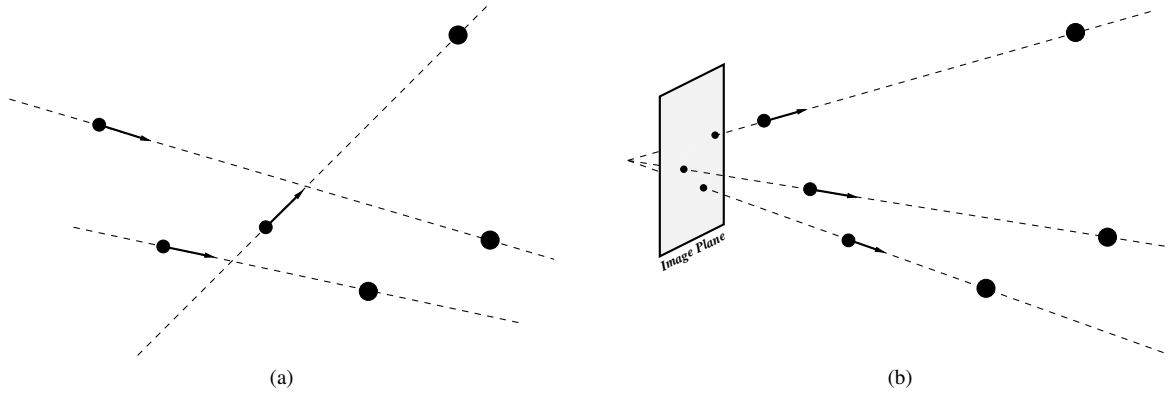


Fig. 1. Representation of the minimal pose estimation problem, considering that we have a calibrated generalized camera. Three 3D points are imaged through three rays. Since the generic camera model is calibrated, the 3D ray coordinates are known in the camera coordinate system. Pose estimation consists therefore on the estimation of the rotation and translation that define the transformation from the 3D world coordinate system into the camera coordinate system, such that the incidence relationship between points and lines is verified.

We do not consider the non-minimal problem here. However, solutions for these cases were also developed by Chen and Chang [26], [27] and by Schweighofer and Pinz [29].

This paper is organized as follows: in the rest of this section we describe the framework for the generalized camera model, give a short description of the previous approaches, and indicate the mathematical notation used in the article. The proposed approach is presented in Sec. II. A method to estimate the intersection points of three quadrics is described in Sec. III. The structure of the algorithm is shown in Sec. IV and in Sec. V-A we compare the proposed approach with the state-of-the-art methods in terms of numerical accuracy, number of possible solutions and processing. Some conclusions are set out in Sec. VI.

The contributions of the paper are stated in Sec. II and III. In Sec. II, we propose the homography-based solution for our problem (to the best of our knowledge, it is the first time that homography has been used for this kind of problem). In Sec. III, we adjust a general minimal solver (based on polynomial eigenvalues) to compute the intersection of the quadrics, as required by our method (solution presented in Sec. II).

A. General Imaging Model

For the estimation of the 3D pose, the calibration of the imaging device is assumed to be known. We use a model that can represent any type of imaging device. The model used is the general imaging model proposed by Grossberg and Nayar [30]. This model assumes that each image pixel is mapped into a ray in 3D world. The mapping parameters form an array of parameters called *raxels*. The set of all *raxels* that represent all image pixels constitutes the complete camera model. Note that there are no constraints on the mapping parameters, which means that the imaging model can represent any type of imaging system, central or non-central.

Since we assume that the camera has been previously calibrated and represented by the *raxel* imaging model for all image pixels, the corresponding 3D line coordinates are known in the camera coordinate system. Several calibration

procedures for the general imaging model were described in the literature, for instance [30], [31], [32].

B. Previous Approaches vs Our Approach

The algorithm proposed by Chen and Chang at [26], [27] was the first method to estimate the minimal three-point pose problem for general imaging devices. The algorithm is based on the estimation of the roots of an eighth degree polynomial equation. To get this polynomial equation, the data-set is transformed such that lines and points verify a set of constraints. An algorithm is used in the new coordinate system to relate the triangle defined by distances between the three known world points and the three projection lines.

This algorithm estimates the coordinates of the points in the camera coordinate system. Thus, since the pose is described by a rotation and a translation an additional estimation of the rigid transformation is also required, based on the matching between the coordinates of the points in the camera and world coordinate systems, for instance [33], [34].

The approach of Ramalingam *et al.* [28] is similar to that proposed by Chen and Chang, which is based on the fact that the distances between the world points must be preserved. They claim that the three distances' constraints generate an eighth degree polynomial equation. However, unlike the algorithm proposed by Chen and Chang, they do not give a closed-form solution for the polynomial parameters. Since the basis of this algorithm is similar to the method of Chen and Chang, we will restrict our analysis to the latter algorithm.

Nistér and Stewénus [24], [25] proposed an algorithm that is also based on the estimation of the roots of a single variable eighth degree polynomial equation. This algorithm also requires predefined transformations to the 3D points and 3D lines, in the world and camera coordinate systems. The coefficients of the polynomial equation are obtained from the intersection between a ruled quartic surface and a circle.

Finally, we note that both algorithms are based on the derivation of an eighth degree polynomial equation, as a function of a single variable. Despite the differences between

the approaches, these algorithms require a set of rigid transformations of the dataset to be performed to get the coefficients of the eighth degree polynomial equation. These transformations are more complex – see, for example, the derivation of [35], [25] – and can lead to critical configurations – see, for instance [27, Sec. 3.2].

In this article, we propose a different approach. We do not aim to obtain a polynomial equation as a function of a single variable. Instead, we express the solutions as the intersection between three quadrics, where the number of solutions will be up to eight, as with the previous approaches. There are, however, methods to express the intersection of three quadrics as a problem of finding the roots of an eighth degree polynomial equation. In Guo [12, Theorem 1] the author shows a procedure to generate, in closed form, the coefficient parameters of an eighth degree polynomial equation, from a general problem of finding the points where quadrics intersect. The main results of our approach, in terms of problem parametrization, are that it does not require any transformation of the dataset and it is based on a very simple and intuitive four-step algorithm.

Note, too, that no comparison between previous methods has yet been made.

C. Background and Notation

By default, bold capital letters represent matrices (eg. $\mathbf{A} \in \mathbb{R}^{n \times m}$, n rows and m columns), bold small letters represent vectors (eg. $\mathbf{a} \in \mathbb{R}^n$, n elements) and small letters represent one-dimensional elements (eg. a). By default, a vector is a column.

Let us consider $\mathbf{U} \in \mathbb{R}^{n \times m}$ and $\mathbf{V} \in \mathbb{R}^{k \times l}$ and the equation

$$\mathbf{UXV}^T = \mathbf{C} \quad (1)$$

where $\mathbf{X} \in \mathbb{R}^{m \times l}$ is the equation unknown. Using the Kronecker product, this equation can be rewritten as

$$(\mathbf{V} \otimes \mathbf{U}) \text{vec}(\mathbf{X}) = \text{vec}(\mathbf{C}) \quad (2)$$

where \otimes represents the Kronecker product with $(\mathbf{V} \otimes \mathbf{U}) \in \mathbb{R}^{kn \times lm}$ and $\text{vec}(\cdot)$ is a vector formed by stacking the columns of the relevant matrix.

The hat operator, $\hat{\mathbf{A}} \in \mathbb{R}^{3 \times 3}$, denotes the skew-symmetric matrix that linearizes the exterior product such that $\mathbf{a} \times \mathbf{b} = \hat{\mathbf{A}}\mathbf{b}$.

In this article we use Plücker coordinates to represent lines $\mathbb{L} \doteq (\mathbf{d}, \mathbf{m})$, where the elements of $\mathbf{l} \in \mathbb{R}^6$ are constrained to $\langle \mathbf{d}, \mathbf{m} \rangle = 0$ (known as the Klien quadric). For more information about Plücker coordinates see [36]. $\mathbf{d} \in \mathbb{R}^3$ and $\mathbf{m} \in \mathbb{R}^3$ are respectively the direction and moment of the line and for any point $\mathbf{x} \in \mathbb{R}^3$ incident with the line, we can write $\mathbf{m} = \mathbf{x} \times \mathbf{d}$.

The dimensions of the column-space and dimension of the null-space are represented by $\text{rank}(\cdot)$ and $\text{nullity}(\cdot)$, respectively.

II. PROPOSED APPROACH

Pose estimation requires the estimation of a rotation matrix $\mathbf{R} \in \mathcal{SO}(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$ that define the rigid transformation between the world and camera coordinate systems. Since we consider that the imaging device is calibrated, 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. To distinguish between features represented in the world coordinate system and entities in the camera coordinate system, we use the superscripts (\mathcal{W}) and (\mathcal{C}) , respectively.

The rigid transformation between a point in world coordinates and the same point in camera coordinates is given by

$$\mathbf{p}^{(\mathcal{C})} = \mathbf{R}\mathbf{p}^{(\mathcal{W})} + \mathbf{t}. \quad (3)$$

Let us consider a plane $\pi^{(\mathcal{W})} \in \mathbb{R}^4$, defined by any three 3D points. Using this assumption and from the definition of the homography map [37], [1], we can rewrite (3) as

$$\mathbf{p}^{(\mathcal{C})} = \underbrace{\left(\mathbf{R} + \frac{1}{\pi_0} \mathbf{t} \bar{\boldsymbol{\pi}}^T \right)}_{\mathbf{H}} \mathbf{p}^{(\mathcal{W})}, \quad (4)$$

valid for any point that belongs to that plane. The matrix \mathbf{H} is called the homography matrix, π_0 and $\bar{\boldsymbol{\pi}}$ are respectively the distance from the plane to the origin and the unit normal vector to the plane.

The homography matrix can be defined up to a scale factor. However, this matrix only defines a rigid transformation for points belonging to the plane $\pi^{(\mathcal{W})}$ if it meets the following condition:

Proposition 1. *A matrix \mathbf{H} belongs to the space of the homography matrices \mathcal{H} if it verifies following condition*

$$\mathbf{H} \in \mathcal{H} \doteq \{ \mathbf{H} \in \mathbb{R}^{3 \times 3} : \sigma_2(\mathbf{H}) = 1 \} \quad (5)$$

where $\sigma_2(\mathbf{H})$ is the second biggest singular value of \mathbf{H} .

The *proof* of this proposition is a simple extension of the proof found in [37, Lemma 5.18].

A. Decomposition of the Homography Matrix

For the decomposition of the homography matrix we use the method described in Section 5.3.3 of [37]. In this section we briefly describe this method and analyze the decomposition used in our approach, in which the coordinates of $\pi^{(\mathcal{W})}$ are known.

Let us consider the eigen decomposition of $\mathbf{H}^T \mathbf{H}$ as

$$\mathbf{H}^T \mathbf{H} = \mathbf{V} \boldsymbol{\Sigma}^2 \mathbf{V}^T, \quad (7)$$

\mathbf{v}_i denotes the i th column of \mathbf{V} and σ_i^2 the i th eigenvalue of \mathbf{H} (note that from Proposition 1, we know that $\sigma_2^2 = 1$). Using these assumptions, we define

$$\mathbf{u}_1 = \frac{\sqrt{1 - \sigma_3^2} \mathbf{v}_1 + \sqrt{\sigma_1^2 - 1} \mathbf{v}_3}{\sqrt{\sigma_1^2 - \sigma_3^2}} \quad (8)$$

$$\mathbf{u}_2 = \frac{\sqrt{1 - \sigma_3^2} \mathbf{v}_1 - \sqrt{\sigma_1^2 - 1} \mathbf{v}_3}{\sqrt{\sigma_1^2 - \sigma_3^2}} \quad (9)$$

Solution 1: $\mathbf{R}^{(1)} = \mathbf{W}^{(1)} (\mathbf{U}^{(1)})^T$ $\bar{\boldsymbol{\pi}}^{(1)} = \mathbf{v}_2 \times \mathbf{u}_1$ $\frac{1}{d} \mathbf{t}^{(1)} = (\mathbf{H} - \mathbf{R}^{(1)}) \mathbf{R}^{(1)}$	Solution 3: $\mathbf{R}^{(3)} = \mathbf{R}^{(1)}$ $\bar{\boldsymbol{\pi}}^{(3)} = -\bar{\boldsymbol{\pi}}^{(1)}$ $\frac{1}{d} \mathbf{t}^{(3)} = -\frac{1}{d} \mathbf{t}^{(1)}$	Solution 2: $\mathbf{R}^{(2)} = \mathbf{W}^{(2)} (\mathbf{U}^{(2)})^T$ $\bar{\boldsymbol{\pi}}^{(2)} = \mathbf{v}_2 \times \mathbf{u}_2$ $\frac{1}{d} \mathbf{t}^{(2)} = (\mathbf{H} - \mathbf{R}^{(2)}) \mathbf{R}^{(2)}$	Solution 4: $\mathbf{R}^{(4)} = \mathbf{R}^{(2)}$ $\bar{\boldsymbol{\pi}}^{(4)} = -\bar{\boldsymbol{\pi}}^{(2)}$ $\frac{1}{d} \mathbf{t}^{(4)} = -\frac{1}{d} \mathbf{t}^{(2)}$
--	---	--	---

(6)

and matrices

$$\mathbf{U}^{(1)} = \begin{pmatrix} \mathbf{v}_2 & \mathbf{u}_1 & \mathbf{v}_2 \times \mathbf{u}_1 \end{pmatrix} \quad (10)$$

$$\mathbf{U}^{(2)} = \begin{pmatrix} \mathbf{v}_2 & \mathbf{u}_2 & \mathbf{v}_2 \times \mathbf{u}_2 \end{pmatrix} \quad (11)$$

$$\mathbf{W}^{(1)} = \begin{pmatrix} \mathbf{H}\mathbf{v}_2 & \mathbf{H}\mathbf{u}_1 & \mathbf{H}\mathbf{v}_2 \times \mathbf{H}\mathbf{u}_1 \end{pmatrix} \quad (12)$$

$$\mathbf{W}^{(2)} = \begin{pmatrix} \mathbf{H}\mathbf{v}_2 & \mathbf{H}\mathbf{u}_2 & \mathbf{H}\mathbf{v}_2 \times \mathbf{H}\mathbf{u}_2 \end{pmatrix}. \quad (13)$$

To conclude, the four possible solutions for the decomposition of the homography matrix are shown in (6).

Usually, the homography matrix is used to estimate the relative pose between two perspective images when a planar pattern is viewed by both cameras. In that case, the decomposition of \mathbf{H} into a rotation matrix and translation vector can yield up to four solutions [37]. However, in the problem addressed in this paper, we assume that the 3D coordinates of the points are known in one of the coordinate systems, which means that the plane coordinates are known in that coordinate system. For this problem, it is possible to define the following theorem.

Theorem 1. *Given a homography matrix, $\mathbf{H} \in \mathcal{H}$, and a plane $\pi^{(W)}$ of known coordinates, the decomposition of \mathbf{H} into rotation and translation has a unique solution.*

The proof of this theorem is given in Supplementary Material [38].

B. Formalization of the problem

As mentioned in Sec. I-A the camera is deemed to have been previously calibrated. Therefore, for each pixel the corresponding 3D line (in the camera coordinate system) is known. A 3D line can be represented in Plücker coordinates, Section I-C.

Using this representation and from [36], a point that is incident on a line satisfies the following relationship

$$\mathbf{d}^{(C)} \times \mathbf{p}^{(C)} = \hat{\mathbf{D}}^{(C)} \mathbf{p}^{(C)} = \mathbf{m}^{(C)}. \quad (14)$$

We wish to determine the relationship between the points in the world coordinate system and the lines in the camera coordinate system. Thus, for the plane $\pi^{(W)}$, defined by the three world points, we can write

$$\hat{\mathbf{D}}^{(C)} \mathbf{H} \mathbf{p}^{(W)} = \mathbf{m}^{(C)}. \quad (15)$$

The aim of this approach is to estimate the pose based on the homography matrix. Thus, let us consider the linearization of the unknown matrix \mathbf{H} in (15), using the Kronecker product, Sec. I-C,

$$(\mathbf{p}^{(W)T} \otimes \hat{\mathbf{D}}^{(C)}) \text{vec}(\mathbf{H}) = \mathbf{m}^{(C)}. \quad (16)$$

The minimal problem, in the case of the camera pose, corresponds to the determination of the mapping between three world points (represented in the world coordinate system) and their corresponding 3D lines (represented in the camera

coordinate system). Therefore, using the representation of (16), we can define the following algebraic relation

$$\underbrace{\begin{pmatrix} \mathbf{p}_1^{(W)T} \otimes \hat{\mathbf{D}}_1^{(C)} \\ \mathbf{p}_2^{(W)T} \otimes \hat{\mathbf{D}}_2^{(C)} \\ \mathbf{p}_3^{(W)T} \otimes \hat{\mathbf{D}}_3^{(C)} \end{pmatrix}}_{\mathbf{M}} \text{vec}(\mathbf{H}) = \underbrace{\begin{pmatrix} \mathbf{m}_1^{(C)} \\ \mathbf{m}_2^{(C)} \\ \mathbf{m}_3^{(C)} \end{pmatrix}}_{\mathbf{w}}. \quad (17)$$

The column-space of the matrix \mathbf{M} is important for the estimation of the unknown parameters. Let us therefore consider the following theorem.

Theorem 2. *Consider a set of three points defined in the world coordinate system and their corresponding lines in the camera coordinate system. If the three points define a plane that does not pass through the origin, the dimension of the column-space of \mathbf{M} at (17) will be rank $(\mathbf{M}) = 6$.*

The proof of this theorem is given in Supplementary Material [38]. Even if the plane contains the origin, a simple change of coordinate system ensures that the conditions of the theorem hold.

The goal of the method is to compute the parameter of \mathbf{H} . As a result, let us consider the following derivation

$$\underbrace{\begin{pmatrix} \mathbf{M} & -\mathbf{w} \end{pmatrix}}_{\mathbf{N}} \boldsymbol{\xi} = \mathbf{0}, \quad \text{where } \boldsymbol{\xi} = \begin{pmatrix} \text{vec}(\mathbf{H}) \\ 1 \end{pmatrix}. \quad (18)$$

From Theorem 2, it can be seen that nullity $(\mathbf{N}) = 4$. Let us represent the null-space of \mathbf{N} as

$$\text{null}(\mathbf{N}) \doteq \left\{ a\mathbf{e}^{(1)} + b\mathbf{e}^{(2)} + c\mathbf{e}^{(3)} + d\mathbf{e}^{(4)} : a, b, c, d \in \mathbb{R} \right\}, \quad (19)$$

where the $\mathbf{e}^{(i)}$ are the vectors defining its basis. Note that $\boldsymbol{\xi}$ has four degrees of freedom. However, from (18), we see that the tenth element of vector $\boldsymbol{\xi}$ must be $\xi_{10} = 1$. Without loss of generality, let us consider $e_{10}^{(i)} = 1$, for all i . Using these definitions, one can see that $\xi_{10} = 1$ is satisfied for $\boldsymbol{\xi} \in \mathcal{R}$, such that

$$\mathcal{R} \doteq \left\{ a\mathbf{f}^{(1)} + b\mathbf{f}^{(2)} + c\mathbf{f}^{(3)} + \mathbf{f}^{(4)} : a, b, c \in \mathbb{R} \right\}, \quad (20)$$

and

$$\mathbf{f}^{(1)} = \mathbf{e}^{(1)} - \mathbf{e}^{(4)} \quad (21)$$

$$\mathbf{f}^{(2)} = \mathbf{e}^{(2)} - \mathbf{e}^{(4)} \quad (22)$$

$$\mathbf{f}^{(3)} = \mathbf{e}^{(3)} - \mathbf{e}^{(4)} \quad (23)$$

$$\mathbf{f}^{(4)} = \mathbf{e}^{(4)}. \quad (24)$$

Note that in this representation the space has only three degrees of freedom.

Let us consider the estimation of the homography matrix represented as $\bar{\mathbf{H}}$. Using the result of (20), we can see that the homography matrix can be represented as

$$\bar{\mathbf{H}} = a\mathbf{F}^{(1)} + b\mathbf{F}^{(2)} + c\mathbf{F}^{(3)} + \mathbf{F}^{(4)}, \quad (25)$$

where $\mathbf{F}^{(i)} \in \mathbb{R}^{3 \times 3}$ are the un-stacking of the vectors $\mathbf{f}^{(i)}$ and a , b and c are the unknowns.

The estimated homography matrix must represent a rigid-body transformation for points at $\pi^{(W)}$, which means that the distance between any two points that belong to that plane must be preserved. Formally, since $\mathbf{p}_i^{(C)} = \mathbf{H}\mathbf{p}_i^{(W)}$,

$$\|\mathbf{p}_i^{(W)} - \mathbf{p}_j^{(W)}\| = \|\mathbf{p}_i^{(C)} - \mathbf{p}_j^{(C)}\| = \|\mathbf{H}\mathbf{p}_i^{(W)} - \mathbf{H}\mathbf{p}_j^{(W)}\|. \quad (26)$$

Using this result and considering $\mathbf{q}_{i,j} = \mathbf{p}_i^{(W)} - \mathbf{p}_j^{(W)}$ for the three world points, one can define three constraints that the estimated homography matrix must satisfy

$$\begin{cases} \mathbf{q}_{1,2}^T \mathbf{q}_{1,2} &= \mathbf{q}_{1,2}^T \bar{\mathbf{H}}^T \bar{\mathbf{H}} \mathbf{q}_{1,2} \\ \mathbf{q}_{1,3}^T \mathbf{q}_{1,3} &= \mathbf{q}_{1,3}^T \bar{\mathbf{H}}^T \bar{\mathbf{H}} \mathbf{q}_{1,3} \\ \mathbf{q}_{2,3}^T \mathbf{q}_{2,3} &= \mathbf{q}_{2,3}^T \bar{\mathbf{H}}^T \bar{\mathbf{H}} \mathbf{q}_{2,3} \end{cases} \quad (27)$$

Using (25), we can rewrite these three constraints as a function of the unknowns as

$$f_{1,2}(a, b, c) = f_{1,3}(a, b, c) = f_{2,3}(a, b, c) = 0 \quad (28)$$

where each $f_{i,j}(a, b, c)$ is such that

$$f_{i,j}(a, b, c) = k_1^{(i,j)} a^2 + k_2^{(i,j)} b^2 + k_3^{(i,j)} c^2 + k_4^{(i,j)} ab + k_5^{(i,j)} ac + k_6^{(i,j)} bc + k_7^{(i,j)} a + k_8^{(i,j)} b + k_9^{(i,j)} c + k_{10}^{(i,j)}. \quad (29)$$

The parameters of the coefficients, $k_n^{(i,j)}$ for $n = 1, \dots, 10$, are shown in (30). Note that each of these functions represents a quadric. As a result, the solutions for vector (a, b, c) that verify the system of (28), can be seen as corresponding to the points where the three quadrics intersect. A method to estimate the intersection of quadrics is described in Sec. III.

A necessary condition for the decomposition of the homography matrix described in Sec. II-A is that the estimated matrices satisfy $\bar{\mathbf{H}} \in \mathcal{H}$. This is the same as proving that $\bar{\mathbf{H}}$ satisfies the conditions of a rigid transformation for any point that belongs to the plane $\pi^{(W)}$. Therefore, this requirement must be met to ensure that the estimates $\bar{\mathbf{R}} \in \mathcal{SO}(3)$ and $\bar{\mathbf{t}} \in \mathbb{R}^3$.

From the following Proposition, all the solutions for $\bar{\mathbf{H}}$ that are obtained from the intersection of the three quadrics, (27) and (28), satisfy the condition $\bar{\mathbf{H}} \in \mathcal{H}$.

Proposition 2. A matrix $\bar{\mathbf{H}} \in \mathbb{R}^{3 \times 3}$ that satisfies (27) and (28) defines a rigid transformation for any point that belongs to $\pi^{(W)}$ (that is defined by the three points $\mathbf{p}_i^{(W)}$) and, as a result, also satisfies the Proposition 1.

The proof of this Proposition is presented in Supplementary Material [38]. This Proposition proves that a homography defines a rigid transformation of a plane if it preserves the distances between any three points of the plane.

Note that from the Theorem 1, we ensure that each solution estimated for $\bar{\mathbf{H}} \in \mathcal{H}$ will generate a single solution for $\bar{\mathbf{R}} \in \mathcal{SO}(3)$ and $\bar{\mathbf{t}} \in \mathbb{R}^3$.

C. The Case of a Central Camera

Let us consider the case of a central/perspective camera. Without loss of generality, we take the origin of the camera

coordinate system to be the center of projection, i.e., the point where all the rays intersect. Since all the rays pass through the origin, the moments of the 3D lines will be zero and we can write $\mathbf{w} = \mathbf{0}$. In this case, from the algebraic relationship of (17), the solutions for the unknown vector $\text{vec}(\mathbf{H})$ will belong to the null-space of \mathbf{M} . From Theorem 2, we have nullity(\mathbf{M}) = 3, which means that we can define $\text{vec}(\mathbf{H}) \in \text{null}(\mathbf{M})$, such that

$$\text{null}(\mathbf{M}) \doteq \left\{ a\mathbf{f}^{(1)} + b\mathbf{f}^{(2)} + c\mathbf{f}^{(3)} : a, b, c \in \mathbb{R} \right\} \quad (31)$$

where $\mathbf{f}^{(i)}$ are the vectors corresponding to the basis for the null-space. As with the general case, a space of solutions for $\bar{\mathbf{H}}$ can be obtained (in this case from the null-space of \mathbf{M}) using (25), by setting $\mathbf{f}^{(4)} = \mathbf{0}$.

Since $\bar{\mathbf{H}}$ must represent a rigid transformation for points that belong to the plane $\pi^{(W)}$, the constraints defined in (27) must be enforced. As with the general case, each of these equations will correspond a quadric surface $f_{i,j}(a, b, c)$, where

$$f_{i,j}(a, b, c) = k_1^{(i,j)} a^2 + k_2^{(i,j)} b^2 + k_3^{(i,j)} c^2 + k_4^{(i,j)} ab + k_5^{(i,j)} ac + k_6^{(i,j)} bc + k_{10}^{(i,j)} \quad (32)$$

and the parameters $k_m^{(i,j)}$ are given by (30), by setting $\mathbf{F}_4 = \mathbf{0}$.

Note that in this case Proposition 2 is also satisfied and, as a result, $\bar{\mathbf{H}} \in \mathcal{H}$. This means that, for each estimate $\bar{\mathbf{H}} \in \mathcal{H}$, we have a single $\bar{\mathbf{R}} \in \mathcal{SO}(3)$ and $\bar{\mathbf{t}} \in \mathbb{R}^3$.

Note that in the case of a minimal pose problem for a central camera, we have up to four solutions [9]. Let us consider the solutions for $\text{vec}(\mathbf{H})$ as \mathbf{u}_i , for $i = 1, \dots, n$ where $n \leq 8$ is the number of possible solutions. It is straightforward to verify that the solutions can be grouped into pairs $\{\mathbf{u}_l, \mathbf{u}_m\}$, where $\mathbf{u}_m = -\mathbf{u}_l$ (if we have a solution \mathbf{u}_l that satisfies $\mathbf{M}\mathbf{u}_l = \mathbf{0}$, then we will have $\mathbf{u}_m = -\mathbf{u}_l$ that satisfies $\mathbf{M}\mathbf{u}_m = \mathbf{0}$). As a result, if we consider only solutions that correspond to points in front of the camera (which is usually used in these cases [9]), one of the solutions from each pair can be eliminated and then we will have up to four solutions, too.

III. SOLVING FOR THE INTERSECTION OF QUADRICS

The most important step in previous algorithms was the estimation of the roots of an eighth degree polynomial equation. There is no analytical solution for the estimation of these roots [26], [27]. However, there are several numerical methods that can be used to estimate these solutions. Nistér & Stewénus and Chen & Chang proposed the use of the companion matrix [39], [24], [25]. According to Chen and Chang, this procedure is suitable to solve the proposed problem when considering criteria of speed and accuracy. With this method, the solutions for the roots are given by the eigenvalues of the companion matrix.

For the algorithm proposed in this article, the solutions for the pose problem are obtained by estimating the intersection points of three quadrics. As mentioned above, Guo at [12] describes a method to derive the intersection of three generic quadrics by solving for the roots of an eighth degree polynomial equation – the coefficients of this polynomial equation can be given in closed form. But other solvers can be used to

$$\begin{aligned}
k_1^{(i,j)} &= \mathbf{q}_{i,j}^T \mathbf{F}^{(1)T} \mathbf{F}^{(1)} \mathbf{q}_{i,j}, & k_6^{(i,j)} &= \mathbf{q}_{i,j}^T \left(\mathbf{F}^{(2)T} \mathbf{F}^{(3)} + \mathbf{F}^{(3)T} \mathbf{F}^{(2)} \right) \mathbf{q}_{i,j}, \\
k_2^{(i,j)} &= \mathbf{q}_{i,j}^T \mathbf{F}^{(2)T} \mathbf{F}^{(2)} \mathbf{q}_{i,j}, & k_7^{(i,j)} &= \mathbf{q}_{i,j}^T \left(\mathbf{F}^{(1)T} \mathbf{F}^{(4)} + \mathbf{F}^{(4)T} \mathbf{F}^{(1)} \right) \mathbf{q}_{i,j}, \\
k_3^{(i,j)} &= \mathbf{q}_{i,j}^T \mathbf{F}^{(3)T} \mathbf{F}^{(3)} \mathbf{q}_{i,j}, & k_8^{(i,j)} &= \mathbf{q}_{i,j}^T \left(\mathbf{F}^{(2)T} \mathbf{F}^{(4)} + \mathbf{F}^{(4)T} \mathbf{F}^{(2)} \right) \mathbf{q}_{i,j}, \\
k_4^{(i,j)} &= \mathbf{q}_{i,j}^T \left(\mathbf{F}^{(1)T} \mathbf{F}^{(2)} + \mathbf{F}^{(2)T} \mathbf{F}^{(1)} \right) \mathbf{q}_{i,j}, & k_9^{(i,j)} &= \mathbf{q}_{i,j}^T \left(\mathbf{F}^{(3)T} \mathbf{F}^{(4)} + \mathbf{F}^{(4)T} \mathbf{F}^{(3)} \right) \mathbf{q}_{i,j}, \\
k_5^{(i,j)} &= \mathbf{q}_{i,j}^T \left(\mathbf{F}^{(1)T} \mathbf{F}^{(3)} + \mathbf{F}^{(3)T} \mathbf{F}^{(1)} \right) \mathbf{q}_{i,j}, & k_{10}^{(i,j)} &= \mathbf{q}_{i,j}^T \left(\mathbf{F}^{(4)T} \mathbf{F}^{(4)} - \mathbf{I} \right) \mathbf{q}_{i,j}.
\end{aligned} \tag{30}$$

solve the intersection of the quadrics. The usual methods to estimate these intersection points are based on the use of the Grobner basis [40], [41] and hidden variable techniques [42]. More recently, a polynomial eigenvalue method [39] for solving minimal problems in computer vision has been presented [43]. As claimed by Kukulova *et al.* [43], these approaches are accurate, simple, fast and easy to implement. As a result, in this paper, we derive a polynomial eigenvalue solution to estimate the intersection points of the quadrics. We now describe the proposed derivation.

Let us consider the system of three equations represented in (28). If we choose a as the eigenvalue, the polynomial eigenvalue has the form

$$(\mathbf{C}_2 a^2 + \mathbf{C}_1 a + \mathbf{C}_0) \mathbf{v} = \mathbf{0}. \tag{33}$$

Polynomial eigenvalue solvers require that the matrices \mathbf{C}_i have square shapes. Note that since the original problem has only three equations and the size of the vector \mathbf{v} is six, we cannot get square matrices \mathbf{C}_i . We have to generate new equations as a polynomial combination of the original ones (*resultant-based methods*), to ensure that the coefficient matrices \mathbf{C}_i have a square shape.

After the generation of the new polynomial equations, we get matrices $\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2 \in \mathbb{R}^{15 \times 15}$ from (35) and (34), and $\mathbf{v} \in \mathbb{R}^{15}$ is such that

$$\mathbf{v} = (b^4, b^3 c, b c^3, b^2 c^2, c^4, b^3, b^2 c, b c^2, c^3, b^2, b c, c^2, b, c, 1). \tag{36}$$

Since matrix \mathbf{C}_0 will be non-singular, the polynomial eigenvalue problem of (33) can be rewritten as a standard eigen decomposition problem $\mathbf{C} \boldsymbol{\nu} = \bar{a} \boldsymbol{\nu}$, where $\bar{a} = \frac{1}{a}$ and matrix $\mathbf{C} \in \mathbb{R}^{30 \times 30}$ and vector $\boldsymbol{\nu} \in \mathbb{R}^{30}$ are

$$\mathbf{C} = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{C}_0^{-1} \mathbf{C}_2 & -\mathbf{C}_0^{-1} \mathbf{C}_1 \end{pmatrix}, \quad \boldsymbol{\nu} = \begin{pmatrix} \mathbf{v} \\ \bar{a} \mathbf{v} \end{pmatrix}. \tag{37}$$

To conclude, the solution for the unknowns (a, b, c) that satisfies (28) can be given by the eigen decomposition of matrix \mathbf{C} where: a is given by the inverse of the eigenvalue of \mathbf{C} and the corresponding b and c are given by the thirteenth and fourteenth element of the corresponding eigenvector.

Note that given the size of matrix \mathbf{C} there will be thirty eigenvalues. However, from (34), we can see that there will be sixteen columns of matrices $-\mathbf{C}_0^{-1} \mathbf{C}_2$ and $-\mathbf{C}_0^{-1} \mathbf{C}_1$ equal to zero. These columns will generate zero eigenvalues, which cannot be used since the value of a will be the inverse of each eigenvalue. Thus, we can eliminate these columns and the corresponding rows, resulting in a 14×14 matrix. Additional zero and complex eigenvalues are also generated and have to be eliminated, which will yield up to eight real solutions.

IV. ALGORITHM OUTLINE

We will now give an overview of the proposed method. The algorithm can be decomposed into the following four steps:

- 1) compute the null-space of matrix \mathbf{N} at (18), which is obtained from the set of correspondences between world points and 3D lines from the general imaging model;
- 2) using the basis of the estimated null-space, compute the coefficients of the quadric surfaces using (30); using these coefficients get matrices $\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2$ from (35) and (34), and use these matrices to compute matrix \mathbf{C} , as shown in (37);
- 3) compute the eigen decomposition of matrix \mathbf{C} , and, for each eigenvalue, set a equal to the inverse of the eigenvalue and b and c equal respectively to the thirteenth and fourteenth elements of the corresponding eigenvector;
- 4) for each set of (a, b, c) , compute the estimates for the homography matrix using (25); and decompose each estimate using the method described in Sec. II-A.

We again note that the third step – which corresponds to the solver for the quadrics intersection proposed at Sec. III – can be replaced by the solution described by Guo in [12, Theorem 1]. The method proposed in Sec. III is much more intuitive and easy to implement than the method described in Guo's article, which derives an eighth degree polynomial equation for the problem. However, as we describe in Sec. V-A3, finding the roots of an eighth degree polynomial equations can solve the intersection of the quadrics faster.

The algorithm proposed in this paper is implemented in MATLAB and will be available on the author's page.

V. EXPERIMENTAL RESULTS

In this section we evaluate the proposed method against the state-of-the-art approaches, using synthetic data. In addition, we validate our method by testing it using real data. We conclude by analyzing the experimental results.

A. Experiments with Synthetic Data

The important parameters to evaluate the solutions of minimal problems are numerical accuracy, computation time, and the number of possible solutions given by the algorithms. As noted by Nistér and Stewénus [24], [25], all minimal solutions will behave in much the same way with respect to noise, which means that tests of this type are less relevant. We compare the algorithm proposed in this article with the methods proposed by Nistér and Stewénus at [24], [25] and by Chen and Chang at [26], [27]. Afterwards, we describe the generation of the dataset used to test the minimal three-point pose problem with the three algorithms.

$$\mathbf{C}_1^{(6:15)\dagger} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_4^{(1,2)} & k_5^{(1,2)} & k_7^{(1,2)} \\ 0 & 0 & 0 & 0 & k_4^{(1,2)} & k_5^{(1,2)} & 0 & k_7^{(1,2)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_4^{(1,2)} & k_5^{(1,2)} & 0 & k_7^{(1,2)} & 0 \\ 0 & k_4^{(1,2)} & k_5^{(1,2)} & 0 & 0 & k_7^{(1,2)} & 0 & 0 & 0 & 0 \\ k_4^{(1,2)} & k_5^{(1,2)} & 0 & 0 & k_7^{(1,2)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_4^{(1,3)} & k_5^{(1,3)} & k_7^{(1,3)} \\ 0 & 0 & 0 & 0 & k_4^{(1,3)} & k_5^{(1,3)} & 0 & k_7^{(1,3)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_4^{(1,3)} & k_5^{(1,3)} & 0 & k_7^{(1,3)} & 0 \\ 0 & k_4^{(1,3)} & k_5^{(1,3)} & 0 & 0 & k_7^{(1,3)} & 0 & 0 & 0 & 0 \\ 0 & 0 & k_4^{(1,3)} & k_5^{(1,3)} & 0 & 0 & k_7^{(1,3)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_4^{(2,3)} & k_5^{(2,3)} & k_7^{(2,3)} \\ 0 & 0 & 0 & 0 & k_4^{(2,3)} & k_5^{(2,3)} & 0 & k_7^{(2,3)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_4^{(2,3)} & k_5^{(2,3)} & 0 & k_7^{(2,3)} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_4^{(2,3)} & k_5^{(2,3)} & k_7^{(2,3)} \\ k_4^{(2,3)} & k_5^{(2,3)} & 0 & 0 & k_7^{(2,3)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_4^{(2,3)} & k_5^{(2,3)} & 0 & 0 & k_7^{(2,3)} & 0 & 0 & 0 \end{pmatrix}, \mathbf{C}_2^{(10:15)\dagger} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & k_1^{(1,2)} \\ 0 & 0 & 0 & k_1^{(1,2)} & 0 & 0 \\ 0 & 0 & 0 & 0 & k_1^{(1,2)} & 0 \\ 0 & k_1^{(1,2)} & 0 & 0 & 0 & 0 \\ k_1^{(1,2)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_1^{(1,3)} \\ 0 & 0 & 0 & k_1^{(1,3)} & 0 & 0 \\ 0 & 0 & 0 & 0 & k_1^{(1,3)} & 0 \\ 0 & 0 & 0 & 0 & k_1^{(1,3)} & 0 \\ 0 & k_1^{(1,3)} & 0 & 0 & 0 & 0 \\ 0 & 0 & k_1^{(1,3)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_1^{(2,3)} \\ 0 & 0 & 0 & k_1^{(2,3)} & 0 & 0 \\ 0 & 0 & 0 & 0 & k_1^{(2,3)} & 0 \\ k_1^{(2,3)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_1^{(2,3)} & 0 & 0 & 0 \end{pmatrix}, \quad (34)$$

$$\mathbf{C}_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_2^{(1,2)} & k_6^{(1,2)} & k_3^{(1,2)} & k_8^{(1,2)} & k_9^{(1,2)} & k_{10}^{(1,2)} \\ 0 & 0 & 0 & 0 & 0 & k_2^{(1,2)} & k_6^{(1,2)} & k_3^{(1,2)} & 0 & k_8^{(1,2)} & k_9^{(1,2)} & 0 & k_{10}^{(1,2)} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & k_2^{(1,2)} & k_6^{(1,2)} & k_3^{(1,2)} & 0 & k_8^{(1,2)} & k_9^{(1,2)} & 0 & k_{10}^{(1,2)} \\ 0 & k_2^{(1,2)} & k_6^{(1,2)} & 0 & k_3^{(1,2)} & 0 & 0 & k_8^{(1,2)} & k_9^{(1,2)} & 0 & 0 & k_{10}^{(1,2)} & 0 & 0 \\ k_2^{(1,2)} & k_6^{(1,2)} & 0 & k_3^{(1,2)} & 0 & k_8^{(1,2)} & k_9^{(1,2)} & 0 & 0 & k_{10}^{(1,2)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_2^{(1,3)} & k_6^{(1,3)} & k_3^{(1,3)} & k_8^{(1,3)} & k_9^{(1,3)} & k_{10}^{(1,3)} \\ 0 & 0 & 0 & 0 & 0 & k_2^{(1,3)} & k_6^{(1,3)} & k_3^{(1,3)} & 0 & k_8^{(1,3)} & k_9^{(1,3)} & 0 & k_{10}^{(1,3)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & k_2^{(1,3)} & k_6^{(1,3)} & k_3^{(1,3)} & 0 & k_8^{(1,3)} & k_9^{(1,3)} & 0 & k_{10}^{(1,3)} & 0 \\ 0 & k_2^{(1,3)} & k_6^{(1,3)} & 0 & k_3^{(1,3)} & 0 & 0 & k_8^{(1,3)} & k_9^{(1,3)} & 0 & 0 & k_{10}^{(1,3)} & 0 & 0 & 0 \\ 0 & 0 & k_6^{(1,3)} & k_2^{(1,3)} & k_3^{(1,3)} & 0 & 0 & k_8^{(1,3)} & k_9^{(1,3)} & 0 & 0 & k_{10}^{(1,3)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_2^{(2,3)} & k_6^{(2,3)} & k_3^{(2,3)} & k_8^{(2,3)} & k_9^{(2,3)} & k_{10}^{(2,3)} \\ 0 & 0 & 0 & 0 & 0 & k_2^{(2,3)} & k_6^{(2,3)} & k_3^{(2,3)} & 0 & k_8^{(2,3)} & k_9^{(2,3)} & 0 & k_{10}^{(2,3)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & k_2^{(2,3)} & k_6^{(2,3)} & k_3^{(2,3)} & 0 & k_8^{(2,3)} & k_9^{(2,3)} & 0 & k_{10}^{(2,3)} & 0 \\ k_2^{(2,3)} & k_6^{(2,3)} & 0 & k_3^{(2,3)} & 0 & k_8^{(2,3)} & k_9^{(2,3)} & 0 & 0 & k_{10}^{(2,3)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_6^{(2,3)} & k_2^{(2,3)} & k_3^{(2,3)} & 0 & 0 & k_8^{(2,3)} & k_9^{(2,3)} & 0 & 0 & k_{10}^{(2,3)} & 0 & 0 & 0 \end{pmatrix}. \quad (35)$$

[†] – The superscripts at $\mathbf{C}_1^{(6:15)}$ and $\mathbf{C}_2^{(10:15)}$ represent the columns of respective matrices. The disregarded columns will have only zero elements.

TABLE I

IN THIS TABLE, WE SHOW THE MEDIAN AND MODE FOR EACH GRAPHIC OF THE FIG. 2, THAT REPRESENT THE NUMERICAL ERRORS FOR ROTATION, TRANSLATION AND DISTANCES BETWEEN POINTS.

Numerical Errors:	Our Method		Nistér and Stewénus		Chen and Chang	
	median	mode	median	mode	median	mode
Rotation	1, 4.10 ⁻¹³	1.9.10 ⁻¹⁴	9, 5.10 ⁻¹²	8, 3.10 ⁻¹³	1, 2.10 ⁻¹²	8.3.10 ⁻¹³
Translation	2, 3.10 ⁻¹¹	3, 7.10 ⁻¹²	1, 5.10 ⁻⁹	7, 5.10 ⁻¹¹	1, 7.10 ⁻¹⁰	3, 5.10 ⁻¹¹
Distance between points	9, 4.10 ⁻¹²	8, 3.10 ⁻¹³	3, 8.10 ⁻¹⁰	7, 5.10 ⁻¹¹	3, 6.10 ⁻¹¹	1, 7.10 ⁻¹¹

We used synthetic datasets for these experiments. For that purpose consider a cube with a side length of 500 units. The data were generated by randomly mapping 3D lines and points $\mathbf{l}_i^{(C)} \leftrightarrow \mathbf{p}_i^{(C)}$, for $i = 1, 2, 3$. A random rigid transformation was randomly generated ($\mathbf{R} \in \mathcal{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$). This rigid transformation was applied to the set of points, such that $\mathbf{p}_i^{(C)} \mapsto \mathbf{p}_i^{(W)}$. For the data generated and using the mappings $\mathbf{l}_i^{(C)} \leftrightarrow \mathbf{p}_i^{(W)}$, for $i = 1, 2, 3$, the pose is estimated using the corresponding algorithms.

This process is repeated for 10^5 trials with a new pose and data being randomly generated for each trial.

1) *Numerical Errors:* The characterization of the numerical error is one of the main parameters used to evaluate minimal

three-point pose algorithms. In Fig. 2, we show the results for the three algorithms that are compared in this article.

We consider three different metrics for the numerical errors:

- 1) rotation error;
- 2) translation error;
- 3) average of the distance error between the estimated and ground-truth world points in the camera coordinate system.

Next, we define the errors in detail. For each of the 10^5 trials we computed the estimated poses using the algorithms mentioned. From the set of possible solutions, we eliminated those that do not fit the problem, using the distance from the estimated solutions to the ground-truth.

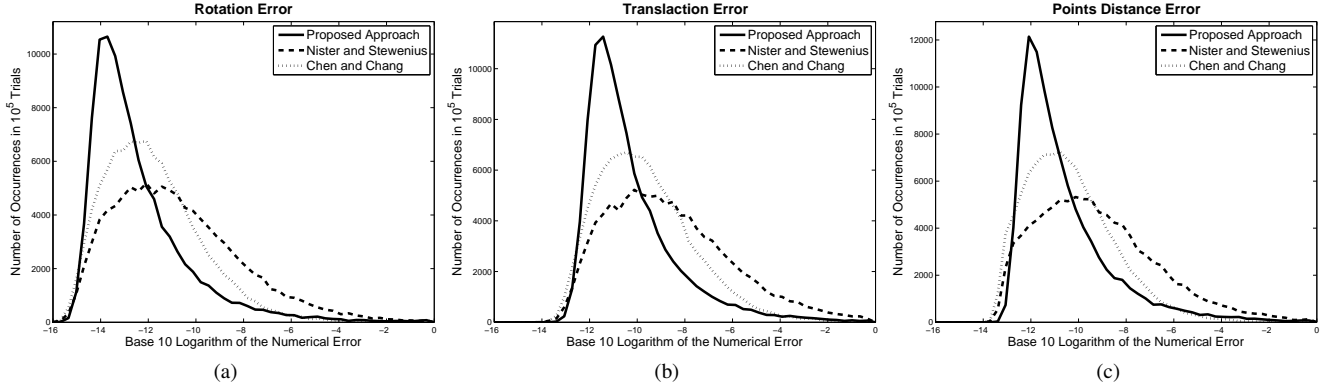


Fig. 2. In this figure, we show the distribution of the numerical error in the computation of the rotation parameters (a) (distance between the estimated and *ground-truth* angles); in the computation of the translation vector (b) (distance between the estimated and *ground-truth* translation vector; and in terms of the distance between the *ground-truth* and estimated points in the camera coordinate system (c).

For each of the trial we computed

$$d_{\text{rotation}} = \|(\bar{\alpha}, \bar{\beta}, \bar{\gamma}) - (\alpha, \beta, \gamma)\| \quad (38)$$

where $(\bar{\alpha}, \bar{\beta}, \bar{\gamma})$ and (α, β, γ) are the rotation angles of the estimated solutions and ground-truth solution, respectively. The error is the usual Euclidean error defined between two vectors (L_2 metric). The frequency distribution of the error values is shown in Fig. 2(a).

The numerical error for translation was computed using

$$d_{\text{translation}} = \|\bar{\mathbf{t}} - \mathbf{t}\|, \quad (39)$$

where $\bar{\mathbf{t}}$ and \mathbf{t} are the estimated and ground-truth translation vectors, respectively. The distribution of the numerical errors of translation is shown in Fig. 2(b).

We also computed an error measure related to the estimation of the coordinates of the world points in the camera coordinate system. Since the ground-truth $\mathbf{p}^{(c)}$ is known, we computed the error as

$$d_{\text{points}} = \frac{1}{3} \sum_{i=1}^3 \|\bar{\mathbf{p}}_i^{(c)} - \mathbf{p}_i^{(c)}\|, \text{ where } \bar{\mathbf{p}}_i^{(c)} = \bar{\mathbf{R}}\mathbf{p}_i^{(c)}\bar{\mathbf{t}}, \quad (40)$$

where $\bar{\mathbf{R}}$ and $\bar{\mathbf{t}}$ are the estimates for rotation and translation respectively. The frequency distribution of the numerical errors is shown in Fig. 2(c).

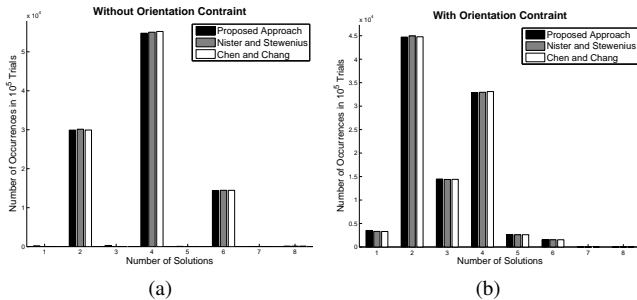


Fig. 3. In this figure we show the distribution of the number of solutions for the three algorithms. Figures (a) and (b) display the number of solutions before and after the application of the orientation constraint respectively.

We also computed the median and mode for each curve. The results are shown in Tab. I.

2) *Number of Solutions*: Note that in general, minimal problems in computer vision do not generate single solutions. As a result, comparing the different methods in terms of the number of solutions is important. In this section we analyze the distribution of the number of solutions for the 10^5 trials described above.

Wrong solutions are eliminated by assuming the constraint that “the camera looks ahead”. Assuming that the line is represented by a point $\mathbf{x}^{(c)}$ and its direction $\mathbf{d}^{(c)}$, this constraint can be formalized by the inequality $\mathbf{d}_i^{(c)T}(\bar{\mathbf{p}}_i^{(c)} - \mathbf{x}_i^{(c)}) > 0$, for $i = 1, 2, 3$.

We computed the number of solutions for all the 10^5 trials and the results are shown in Figs. 3(a) and 3(b), before and after imposing the orientation constraint, respectively.

We also computed the averages of the number of solutions generated by the three algorithms considered, with and without the imposition of the orientation constraint. The results are presented in Tab. II.

3) *Computational Effort*: We implemented the algorithms in MATLAB using an *Intel core i7-3930k* with a *3.2GHz* processor. All the computation times given in the paper were estimated as the median of the respective computation times, for all the 10^5 trials. All the proposed approaches require iterative steps to solve the problem. When compared to the iterative steps, the computation time spent on simple analytical operations is negligible. As a result, for the evaluation we only considered the steps that require the most significant

TABLE II
IN THIS TABLE WE SHOW THE AVERAGES OF THE NUMBER OF SOLUTIONS GENERATED BY EACH ALGORITHM, WITHOUT AND WITH THE APPLICATION OF THE ORIENTATION CONSTRAINT.

Mean:	Orientation Constraint	
	Without	With
Our Method	3,686	2,909
Nister and Stewenius	3.691	2.912
Chen and Chang	3.695	2.915

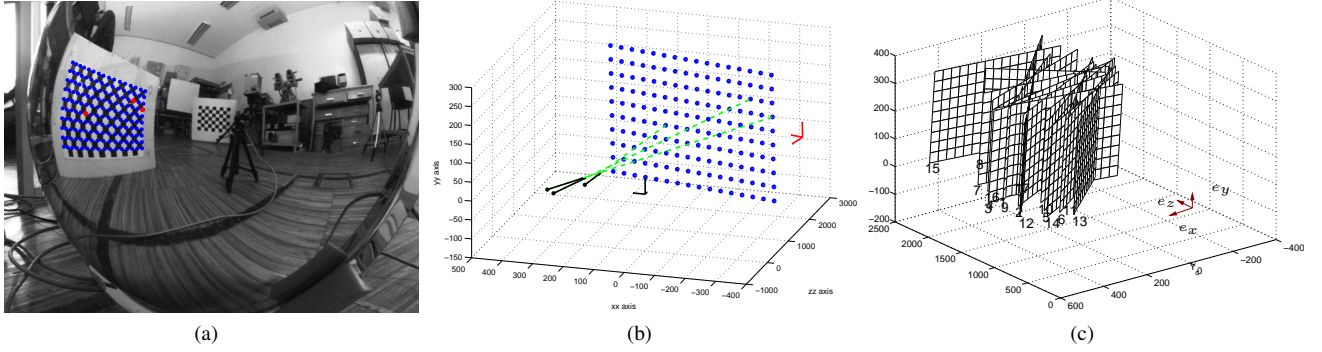


Fig. 4. For results with real data, we use a catadioptric system with quadric mirrors. Seventeen images of 3D planes were acquired (c). An example of data used for pose estimation is displayed in Figs. (a) and (b). Using only three points in the image and their corresponding points in the chess-board coordinate system (red coordinate system in (a)), rotation and translation (corresponding to the rigid transformation between the chess-board and camera coordinate system) are estimated. In (b) the camera coordinate system is shown in black and the estimated chess-board coordinate system is shown in red. In Fig. (c) the metric units are in millimeters.

computational effort, i.e., the iterative steps.

The main computation step for all the algorithms was the computation of the minimal solver, which may correspond to finding the roots of an eighth degree polynomial equation (other simple solvers can be used for the method proposed in this paper). As suggested by both Nistér & Stewénus and Chen & Chang, we used the companion matrix method to solve the roots of the eighth degree polynomial equations. This method amounts to performing an eigen decomposition of an 8×8 matrix, which takes $43\mu s$.

For our parametrization, we had to consider the estimation of the null-space of the matrix \mathbf{N} , at (18). Iterative singular value decomposition methods would require that we take into account $40\mu s$. However, there are solutions that can be used to derive general analytical solutions to compute the singular value decomposition of matrix \mathbf{N} . For instance, we can symbolically produce the reduced row echelon form of matrix \mathbf{N} and use this derivation to compute the null-space. We also derived an analytical solution using the geometric properties of the cross and dot products and considering the rows of matrices \mathbf{M} and \mathbf{N} from (17) and (18). As a result, it is possible to compute the null-space in closed-form, using very simple analytical operations.

Both our approach and Chen & Chang's method require additional steps to recover the rotation and translation that define the pose. Chen & Chang give the coordinates of the world points in the camera coordinate system and the method we propose estimates the homography matrix that defines pose. In both cases, the computation of an eigen decomposition of a 3×3 matrix is required for each valid solution. Note that this decomposition can be computed in closed form. We now describe how we can achieve this decomposition.

Let us consider that we want to compute the eigen decomposition of the matrix $\mathbf{H}^T \mathbf{H} \in \mathbb{R}^{3 \times 3}$ from the problem described in (7). The eigenvalues σ_i^2 , which will be the diagonal elements of the matrix Σ^2 in (7), are computed such that

$$\det(\mathbf{H}^T \mathbf{H} - \sigma^2 \mathbf{I}) = 0, \quad (41)$$

where $\det(\cdot)$ denotes the determinant and \mathbf{I} denotes the 3×3 identity matrix. (41) defines a third degree polynomial

equation with unknown σ^2 . As a result, the solutions for σ_i^2 for $i = 1, 2, 3$ are given by the roots of this equation, which can be computed in closed-form simply by applying Ferrari's method. To conclude, and since we already have σ_i^2 , the solutions for the columns $\mathbf{v}_i \in \mathbb{R}^3$, for $i = 1, 2, 3$, that form \mathbf{V} of (7) are given by the equation

$$\mathbf{H}^T \mathbf{H} \mathbf{v}_i = \sigma_i^2 \mathbf{v}_i. \quad (42)$$

B. Experimental Results with Real Data

To obtain data from real images, we considered a non-central catadioptric imaging device comprising a perspective camera and a quadric mirror. To obtain the data, we used a chess board in seventeen different positions and orientations, Fig. 4(c). The images of the chess board were acquired by the non-central catadioptric camera. To obtain the ground-truth, the 3D positions of the points of the chess board were estimated in world coordinates using a set-up with perspective cameras. An example displaying the points in the world and their corresponding image coordinates is shown in Figs. 4(a) and 4(b). In both figures blue dots represent image points and their corresponding world points in ground-truth coordinates.

Sixteen of the seventeen chess board planes (and their images) were used to calibrate the camera (using [32]). The resulting calibration provides the coordinates of the 3D projecting rays for each image pixel. A reference coordinate system was associated with the chess board plane that was not used to calibrate the camera. The coordinate system was defined so that Theorem 2 is satisfied (the plane does not contain the origin of the coordinate system). Three points of the chess board plane and their images were then used to estimate the pose of the plane. The 3D projection line was estimated in the camera coordinate system, for each of the three points in the image. Figs. 4(a) and 4(b) show the image points plus an example of the 3D lines (in green).

The mappings between the three points in the world (chess board coordinate system) and their projection lines (represented in the camera coordinate system) were used to estimate pose. In addition, we selected the best solution (in terms of the errors in the 3D reconstructed points) from the set of possible

TABLE III

IN THIS TABLE, WE SHOW THE RESULTS OF THE ABSOLUTE ERRORS FOR ALL SIX POSE PARAMETERS IN ALL OF THE SEVENTEEN POSE ESTIMATES. ROTATION ANGLES ARE IN DEGREES. NOTE THE THE METRIC IN THE TRANSLATION ARE DIFFERENT FROM THE GROUND-TRUTH TO THE ABSOLUTE ERRORS. WHILE IN THE FIRST CASE IS IN METERS, IN THE SECOND CASE IS IN MILLIMETERS.

Plane	Ground Truth						Absolute Error					
	α_x	α_y	α_z	t_x [m]	t_y [m]	t_z [m]	$\hat{\alpha}_x$	$\hat{\alpha}_y$	$\hat{\alpha}_z$	\hat{t}_x [mm]	\hat{t}_y [mm]	\hat{t}_z [mm]
1	0.86	-18.68	169.21	-0.37	0.06	2.10	0.0053	0.2831	0.1082	1.8810	0.4149	0.0305
2	0.99	-39.83	168.92	-0.43	0.09	1.91	0.2139	0.6406	0.2383	1.1964	0.3307	3.0765
3	0.89	-8.89	169.30	-0.41	0.06	2.08	0.3361	1.2368	0.4696	5.2738	1.6796	1.1239
4	0.94	8.04	169.46	-0.32	0.06	2.11	0.0072	0.0781	0.1519	0.6116	0.8114	0.2942
5	1.57	-51.95	168.14	-0.38	0.12	1.76	0.4777	0.2570	0.6365	0.7986	1.4233	1.0428
6	1.60	-50.09	168.08	-0.32	0.13	1.72	1.4092	0.3863	1.8659	1.1793	5.3712	2.4334
7	0.97	4.18	169.50	-0.34	0.02	2.30	0.0027	1.1902	0.8675	6.2205	4.8714	5.1137
8	1.26	37.49	170.08	-0.12	-0.01	2.48	0.0507	0.0752	0.0321	1.1442	0.6326	1.3746
9	1.34	-54.10	168.32	-0.51	0.08	1.97	0.0292	0.0882	0.1513	0.0811	0.3610	0.0094
10	1.28	-41.00	168.50	-0.33	0.10	1.90	0.1598	0.1031	0.2384	0.7491	1.0101	1.0732
11	1.79	-62.64	167.90	-0.30	0.12	1.75	0.3740	0.0378	0.5223	0.5351	0.5899	0.4138
12	1.95	-68.73	167.74	-0.53	0.13	1.70	0.5593	1.5524	0.2773	9.6657	5.4284	11.1939
13	1.95	-58.60	167.68	-0.29	0.15	1.63	4.4823	1.6889	5.1984	7.1865	13.5883	10.0518
14	1.35	-42.33	168.38	-0.35	0.13	1.73	0.3031	0.9430	0.0728	2.0597	0.7150	9.8622
15	0.98	5.81	169.55	-0.45	-0.04	2.63	0.0538	0.1246	0.1570	0.6553	0.7758	1.3818
16	0.96	-31.92	168.95	-0.47	0.04	2.16	0.9328	2.9167	2.2671	2.4659	10.1745	24.6971
17	2.12	-63.70	167.52	-0.42	0.06	2.09	3.7783	1.3850	4.6183	3.6914	7.4837	0.1563

pose estimates. Using the pose estimate thus obtained, the errors for all six pose parameters were computed.

This procedure was repeated for all 17 planes using a leave-one-out procedure.

We computed the pose parameter's errors for all the seventeen different configurations. The absolute errors in the estimated parameters are shown in Tab. III.

C. Analysis of the Experimental Results

Let us first look at the experiments with synthetic data, Sec. V-A. In terms of the number of solutions (Fig. 3) with and without the orientation constraint, the results are similar for the three methods and so this comparison does not offer additional information on the relative merits of the algorithms. This can be further checked by considering the averages of the number of solutions generated by the three methods that are presented in Tab. II. Thus, in this sub-section we discuss the experimental results in terms of numerical accuracy and processing time.

In terms of numerical accuracy (Fig. 2) it can be seen that the method proposed in this article performed better than the state-of-the-art algorithm, in all the tests. The algorithm proposed by Chen & Chang performs the second best followed by the method proposed by Nistér & Stewénus. This analysis can be confirmed by taking into account the data in Tab. I. This shows that the numerical errors for the method proposed in this paper are better than state-of-the-art methods, for all the evaluation parameters. Moreover, note that the mode of the curves for the results obtained with the previous approaches give worse results than the median of the method proposed in this article, for all the numerical accuracy experiments.

In terms of computational time, Sec. V-A3, we saw that all three algorithms can be implemented with similar computational effort (solving for the roots of an eighth degree polynomial equation). Nevertheless, both our method and

Chen & Chang's method may or may not have additional iterative parts. These iterative parts may be considered if fast implementation is required. However, if computation time is important, analytical derivations must be considered.

Now let us look at the experiments with real data, Sec. V-B. As noted earlier and as mentioned by Nistér and Stewénus [24], [25], all minimal solutions will behave in the same way with respect to noise. But to further evaluate the proposed approach we tested our method using real images. As we can see from Tab. III, and despite the fact that we are using minimal data, the results are good. Note that the absolute errors for the angles are in degrees and that the translation errors are in millimeters (while the ground truth is in meters).

VI. CONCLUSIONS

In this article we have presented a novel homography-based algorithm for the minimal three-point pose problem for generalized cameras, where the constraint that the projection rays must intersect at a single point is not considered. This method can be used to estimate pose for imaging models where reflection or refraction occurs. That is the case, for example, of catadioptric systems made up of arbitrary mirrors, or of a perspective camera looking at a water environment.

Previous approaches used algebra techniques that lead to a single variable eighth degree polynomial and the pose estimates are given by roots of the polynomial. In the case of this paper, the important step is the estimation of the intersection points of three quadrics (which can also be represented by the roots of an eighth degree polynomial equation). The algorithm proposed estimates the pose by computing four vectors obtained from a null-space and, using this null-space, we can define the three quadric equations. The intersection of quadric surfaces occurs in several minimal problems in computer vision. We have described a polynomial eigenvalue representation of the intersection of quadrics, where the solutions are given by an eigen decomposition. Unlike

previous methods, our proposed approach does not require any transformation of the dataset either, and this is why we call it the direct solution to the problem.

The algorithm presented in this paper was not intended to be used where non-minimal data is available; it is intended to be a simple method to allow the analysis of the solutions when one only has access to minimal data. This method is suitable for RANSAC-based approaches for pose estimation. Also, using this algorithm, the solutions with minimal data can be easily obtained and then analyzed for the purpose of comparing non-central cameras.

We compared the proposed algorithm with the state-of-the-art methods and concluded that our algorithm performs better than them in terms of numerical accuracy, with similar computational effort and a similar number of solutions. The validation of the method by testing with real experiments showed good results.

ACKNOWLEDGEMENTS

This work was supported by the Portuguese Foundation for Science and Technology, scholarship SFRH/BD/49054/2008 and by project A “Surgery and Diagnosis Assisted by Computer Using Images” (SCT_2011_02_027_4824) funded by the QREN programme “Mais Centro” with financing from FEDER. Helder Araujo would also like to thank the support of Project FCT/PTDC/EIA-EIA/122454/2010, funded by the Portuguese Science Foundation (FCT) by means of national funds (PIDDAC) and co-funded by the European Fund for Regional Development (FEDER) through COMPETE Operational Programme Competitive Factors (POFC).

REFERENCES

- [1] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [2] M. A. Fischler and R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” *Comm. of the ACM*, 1981.
- [3] D. Nistér, “Preemptive RANSAC for Live Structure and Motion Estimation,” *IEEE Proc. Int’l Conf. Computer Vision*, 2003.
- [4] R. M. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V. G. Vaidya, and M. B. Kim, “Pose Estimation from Corresponding Point Data,” *IEEE Trans. System, Man and Cybernetics*, 1989.
- [5] C.-P. Lu, G. D. Hager, and E. Mjølness, “Fast and Globally Convergent Pose Estimation from Video Images,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2000.
- [6] F. Moreno Noguer, V. Lepetit, and P. Fua, “Accurate Non-Interactive $O(n)$ Solution to the PnP Problem,” *IEEE Proc. Int’l Conf. Computer Vision*, 2007.
- [7] J. Hesch and S. Roumeliotis, “A Direct Least-Squares (DLS) Method for PnP,” *IEEE Proc. Int’l Conf. Computer Vision*, 2011.
- [8] O. Tahri, H. Araujo, Y. Mezouar, and F. Chaumette, “Efficient Iterative Pose Estimation using an Invariant to Rotations,” *IEEE Trans. Cybernetics*, 2014.
- [9] R. Haralick, C.-N. Lee, K. Ottenberg, and M. Nolle, “Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem,” *Int’l J. Computer Vision*, 1994.
- [10] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, “Complete Solution Classification for the Perspective-Three-Point Problem,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2003.
- [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] Y. Guo, “A Novel Solution to the P4P Problem for an Uncalibrated Camera,” *In J. Math. Imaging Vis*, 2012.
- [13] M. Bujnak, Z. Kukelova, and T. Pajdla, “A general solution to the P4P problem for camera with unknown focal length,” *IEEE Proc. Computer Vision and Patter Recognition*, 2008.
- [14] N. Goncalves, “Noncentral Catadioptric Systems with Quadric Mirrors, Geometry and Calibration,” Ph.D. dissertation, University of Coimbra, October 2008.
- [15] A. Agrawal, Y. Taguchi, and S. Ramalingam, “Analytical Forward Projection for Axial Non-Central Dioptric & Catadioptric Cameras,” *Proc. European Conf. Computer Vision*, 2010.
- [16] R. Swaminathan, M. D. Grossberg, and S. K. Nayar, “Non-Single Viewpoint Catadioptric Cameras: Geometry and Analysis,” *Int’l J. Computer Vision*, 2006.
- [17] B. Micusik and T. Pajdla, “Autocalibration & 3D Reconstruction with Non-Central Catadioptric Cameras,” *IEEE Proc. Computer Vision and Patter Recognition*, 2004.
- [18] T. Treibitz, Y. Y. Schechner, and H. Singh, “Flat Refractive Geometry,” *IEEE Proc. Computer Vision and Patter Recognition*, 2008.
- [19] K. N. Kutulakos and E. Steger, “A Theory of Refractive and Specular 3D Shape by Light-Path Triangulation,” *Int’l J. Computer Vision*, 2008.
- [20] J. Ponce, “What is a Camera?” *IEEE Proc. Computer Vision and Patter Recognition*, 2009.
- [21] R. Gupta and R. I. Hartley, “Linear Pushbroom Cameras,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1997.
- [22] J. Yu and L. McMillan, “General Linear Cameras,” *Proc. European Conf. Computer Vision*, 2004.
- [23] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall, “Mosaicing New Views: The Crossed-Slits Projection,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2003.
- [24] D. Nistér, “A Minimal Solution to the Generalized 3-Point Pose Problem,” *IEEE Proc. Computer Vision and Patter Recognition*, 2004.
- [25] D. Nistér and H. Stewénus, “A Minimal Solution to the Generalized 3-Point Pose Problem,” *In J. Math. Imaging Vis*, 2007.
- [26] C.-S. Chen and W.-Y. Chang, “Pose Estimation for Generalized Imaging Device via Solving Non-Perspective n Point Problem,” *IEEE Proc. Int’l Conf. Robotics and Automation*, 2002.
- [27] —, “On Pose Recovery for Generalized Visual Sensors,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2004.
- [28] S. Ramalingam, S. K. Lodha, and P. Sturm, “A Generic Structure-from-Motion Framework,” *Computer Vision and Image Understanding*, 2006.
- [29] G. Schweighofer and A. Pinz, “Globally Optimal $O(n)$ Solution to the PnP Problem for General Camera Models,” *Proc. British Machine Vision Conf.*, 2008.
- [30] G. Grossberg and S. Nayar, “A General Imaging Model and a Method for Finding its Parameters,” *IEEE Proc. Int’l Conf. Computer Vision*, 2001.
- [31] P. Sturm and S. Ramalingam, “A Generic Concept for Camera Calibration,” *Proc. European Conf. Computer Vision*, 2004.
- [32] P. Miraldo, H. Araujo, and Queiró Joao, “Point-based Calibration Using a Parametric Representation of the General Imaging Model,” *IEEE Proc. Int’l Conf. Computer Vision*, 2011.
- [33] 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.
- [34] S. Umeyama, “Least-Square Estimation of Transformation Between Two Point Patterns,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1991.
- [35] D. Nistér, “An Efficient Solution to the Five-Point Relative Pose Problem,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2004.
- [36] H. Pottmann and J. Wallner, *Computational Line Geometry*. Springer-Verlag, 2001.
- [37] Y. Ma, S. Soatto, J. Kořecká, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometry Models*. Springer Science+Business, 2004.
- [38] P. Miraldo and H. Araujo, “Direct solution to the minimal generalized pose,” *Supplement Material*, 2013.
- [39] D. A. Cox, J. Little, and D. O’Shea, *Using Algebraic Geometry*. Springer Science+Business, 2004.
- [40] H. Stewénus, “Gröbner Basis Methods for Minimal Problems in Computer Vision,” PhD Thesis, Lund University, 2005.
- [41] Z. Kukelova, M. Bujnak, and T. Pajdla, “Automatic Generator of Minimal Problem Solvers,” *Proc. European Conf. Computer Vision*, 2008.
- [42] R. Hartley and H. Li, “An Efficient Hidden Variable Approach to Minimal-Case Camera Motion Estimation,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2012.

- [43] Z. Kukelova, M. Bujnak, and T. Pajdla, "Polynomial Eigenvalue Solutions to Minimal Problems in Computer Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2012.



Pedro Miraldo received the master's degree in electrical and computer engineering from the University of Coimbra in 2008 and the doctor's degree in electrical engineering from University of Coimbra in 2013. Currently he is a researcher at the Institute for Systems and Robotics-Coimbra. His work focusses on general camera models and camera localization.



Helder Araujo is a Professor in the Department of Electrical and Computer Engineering of the University of Coimbra, Portugal and also an Invited Professor at Blaise Pascal University, Clermont-Ferrand, France. He is a researcher at the Institute for Systems and Robotics-Coimbra. His research interests include computer and robot vision, robot navigation and localization, and sensor modeling for robot navigation and localization.