• U C •

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

ARTICLE: Pose Estimation for Non-Central Cameras Using Planes

Author:
Pedro MIRALDO
miraldo@isr.uc.pt

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

CONTENTS

Preprint: submitted to the IEEE Int'l Conf. Autonomous Robot Systems & Competitions – ROBÓTICA'2014
February, 2014.

Pose Estimation for Non-Central Cameras Using Planes*

Pedro Miraldo and Helder Araujo
Institute of Systems and Robotics, Department of Electrical and Computer Engineering
University of Coimbra, 3030-290 COIMBRA, PORTUGAL
Email: {miraldo, helder}@isr.uc.pt

***Abstract*—In this paper we study pose estimation for non-central cameras, using planes. The method proposed uses non-minimal data. Using the homography matrix to represent the transformation between the world and camera coordinate systems, we describe a non-iterative algorithm for pose estimation. In addition, we propose a parameter optimization to refine the pose estimate. We evaluate the proposed solutions against the state-of-the-art method in terms of both robustness to noise and computation time. From the experiments, we conclude that the proposed method is more accurate against noise. We also conclude that the numerical results obtained with this method improve with increasing number of data points. In terms of processing speed both versions of the algorithm presented are faster than the state-of-the-art algorithm.**

I. Introduction

Considering only geometric entities, an imaging system is a mapping between the 3D world and a 2D image [1]. This mapping can be represented by an individual association between world 3D lines and pixels in the image plane [2], [3]. Camera calibration consists in the estimation of the correspondences between image pixels and the corresponding projecting 3D straight lines.

Usually image space does not change and, as a result, we can define a 3D coordinate system for the image coordinates. On the other hand, a camera is a mobile device and as a consequence we can not 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 mapping the camera coordinate system into the world coordinate system.

For central camera models (which can be modeled by a perspective projection Figure 1(a)), several approaches to absolute pose estimation, such as [4], [5], have been proposed for both minimal configurations (suitable for hypothesize-and-test architectures like RANSAC [6]) and non–minimal cases [7], [8], [9], [10]. Some work has also been developed extending pose estimation methods so that points and lines can be used, such as e.g. [11], [12].

Most of the algorithms for the estimation of pose are based on arbitrary 3D target point configurations. In many problems such as mobile robotics and augmented reality, it is pratical to use planar patterns to compute the absolute pose. For the case of central cameras several approaches using 3D coplanar points were developed, such as [13], [14].

In the last few years, cameras whose projection rays do not intersect at a single effective view point (imaging devices that can not be modeled by a central projection, see Figure 1(b)) started to be used, due essentially to the large fields of view that can be obtained. Examples of such configurations occur when the imaging rays are subject to reflection and/or refraction. For these imaging devices new methods and algorithms have to be considered for the estimation of the absolute pose.

For non-central camera models, there are algorithms for the minimal case, [15], [16]. For the non-minimal case, Schweighofer and Pinz at [17] proposed an iterative solution for global pose estimation. Despite the fact that the method proposed by Schweighofer and Pinz can be applied to both planar and non-planar cases, the method is iterative.

In this article, we address the problem non-minimal absolute pose estimation for general non-central cameras, when considering the case where the world points belong to a plane. We present a non-iterative algorithm to estimate the pose. In addition, we also propose a refinement of the estimation of the pose parameters by means of an optimization using the *Quasi-Newton* algorithm.

A. Proposed Approach

For the estimation of the 3D pose, the calibration of the imaging device is assumed to be known. We use the *Generalized Camera Model* [2], which can represent any type of imaging device (central or non-central). This model assumes that an image pixel is mapped into an arbitrary ray in 3D world. Since we assume that the camera has been previously calibrated, for all image pixels we know the corresponding 3D straight line coordinates in the camera coordinate system.

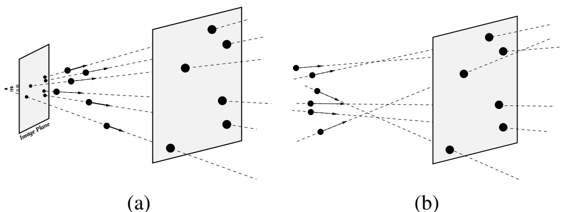
Pose is given by the estimates of the rotation and translation parameters that define the transformation between the camera and the world coordinate systems. In this article we use the homography map to represent this transformation. Using the homography matrix and based on the relationship of incidence between points and lines in 3D space, we define an algebraic relationship for the pose. However, the homography matrix is a function of both transformation and 3D plane parameters [18]. As a result, we divided the estimation of the homography into two steps: we define a space of solutions with three degrees of freedom for the homography matrix, based on the algebraic relationship between the 3D points and 3D lines; using the information of the 3D plane, three constraints that the space of solutions must satisfy are defined. The homography matrix is estimated using these constraints.

B. Notation

In general, bold capital letters (*e.g.* $\mathbf{A} \in \mathbb{R}^{n\times m}$, n rows and m columns), bold small letters (*e.g.* $\mathbf{a} \in \mathbb{R}^{n}$, n elements) and small letters (*e.g.* a) represent matrices, vectors and one dimensional elements respectively. The matrix represented as $\widehat{\mathbf{a}}$ linearizes the exterior product such that $\mathbf{a} \times \mathbf{b} = \widehat{\mathbf{a}}\mathbf{b}$.

Let us consider: known matrices $\mathbf{U} \in \mathbb{R}^{n\times m}$, $\mathbf{V} \in \mathbb{R}^{k\times l}$ and $\mathbf{C}$; and an unknown matrix $\mathbf{X} \in \mathbb{R}^{n\times l}$. Using *Kronecker*

*This work was supported by the Portuguese Foundation for Science and Technology (Ref: PTDC/EIA-EIA/122454/2010) and the Portuguese program Mais-Centro (Ref: CENTRO-07-ST24-FEDER-002027).



(a) (b)

Fig. 1. Depiction of the pose estimation problem, using planar patterns. Figure (a) shows pose estimation using central cameras. Figure (b) shows the pose estimation configuration in the case of a general non-central camera.

product we can define the following relation

$$\mathbf{U}\mathbf{X}\mathbf{V}^T = \mathbf{C} \Rightarrow \left(\mathbf{V} \otimes \mathbf{U}\right) \text{vec}\left(\mathbf{X}\right) = \text{vec}\left(\mathbf{C}\right) \tag{1}$$

where $\otimes$ represent the *Kronecker* product with $(\mathbf{V} \otimes \mathbf{U}) \in \mathbb{R}^{nk \times nl}$ and vec $(.)$ is a vector formed by the stacking of the columns of the respective matrix.

II. Relationship of Incidence using the Homography

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, Figure 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 $\mathbf{p}^{(\mathcal{W})}$ and the same point in camera coordinates $\mathbf{p}^{(\mathcal{C})}$ is given by

$$\mathbf{p}^{(\mathcal{C})} = \mathbf{R}\mathbf{p}^{(\mathcal{W})} + \mathbf{t}. \tag{2}$$

Since we use the assumption that all the points belong to a plane $\mathbf{\Pi}^{(\mathcal{W})}$, from the homography map [1], [18], we can rewrite Equation (2) as

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

where $\mathbf{\Pi}^{(\mathcal{W})} \doteq (\zeta, \boldsymbol{\pi}) \in \mathbb{R}^4$, $\mathbf{H} \in \mathbb{R}^{3\times 3}$ is called the homography matrix, ζ and $\boldsymbol{\pi}$ are the distance from the plane to the origin and the unit normal vector to the plane $\mathbf{\Pi}^{(\mathcal{W})}$ respectively.

For pose estimation, we assume that the non-central camera is calibrated. Therefore for each pixel the corresponding 3D line is known (in the camera coordinates system). Let us consider that lines are defined using *Plucker* coordinates $\mathbf{l}^{(\mathcal{C})}\mathbb{R} \doteq \left(\mathbf{d}^{(\mathcal{C})}, \mathbf{m}^{(\mathcal{C})}\right)$, where $\mathbf{d}^{(\mathcal{C})}$ and $\mathbf{m}^{(\mathcal{C})}$ are the direction and moment of the line respectively, constrained to $\left\langle \mathbf{d}^{(\mathcal{C})}, \mathbf{m}^{(\mathcal{C})} \right\rangle = 0$. From the 3D incident relation between a line and a point [19], we have

$$\mathbf{d}^{(\mathcal{C})} \times \mathbf{p}^{(\mathcal{C})} = \mathbf{m}^{(\mathcal{C})} \Rightarrow \widehat{\mathbf{d}}^{(\mathcal{C})}\mathbf{p}^{(\mathcal{C})} - \mathbf{m}^{(\mathcal{C})} = \mathbf{0}. \tag{4}$$

Since our goal is to estimate the pose using coplanar points $\mathbf{p}^{(\mathcal{W})} \in \mathbf{\Pi}^{(\mathcal{W})}$, we can use the homography map to transform points from camera coordinates into the world coordinates (Equation (3)). Thus, from Equation (4) and since $\mathbf{m}^{(\mathcal{C})} \times \mathbf{m}^{(\mathcal{C})} = \mathbf{0}$, we derive the following relation

$$\widehat{\mathbf{d}}^{(\mathcal{C})}\mathbf{H}\mathbf{p}^{(\mathcal{W})} - \mathbf{m}^{(\mathcal{C})} = \widehat{\mathbf{m}}^{(\mathcal{C})}\widehat{\mathbf{d}}^{(\mathcal{C})}\mathbf{H}\mathbf{p}^{(\mathcal{W})} = \mathbf{0}. \tag{5}$$

Using the *Kronecker* product, we isolate the unknown matrix $\mathbf{H}$, such that

$$\left(\mathbf{p}^{(\mathcal{W})\,T} \otimes \widehat{\mathbf{m}}^{(\mathcal{C})}\widehat{\mathbf{d}}^{(\mathcal{C})}\right) \text{vec}\left(\mathbf{H}\right) = \mathbf{0}. \tag{6}$$

From the properties of the *Kronecker* product [20], the dimension of the *column-space* of $\mathbf{p}^{(\mathcal{W})\,T} \otimes \widehat{\mathbf{m}}^{(\mathcal{C})}\widehat{\mathbf{d}}^{(\mathcal{C})}$ is equal to the product of the dimension of the *column-space* of $\mathbf{p}^{(\mathcal{W})\,T}$ and $\widehat{\mathbf{m}}^{(\mathcal{C})}\widehat{\mathbf{d}}^{(\mathcal{C})}$. Since $\left\langle \mathbf{d}^{(\mathcal{C})}, \mathbf{m}^{(\mathcal{C})} \right\rangle = 0$, it can be shown that the dimension of the *column-space* of both $\mathbf{p}^{(\mathcal{W})\,T}$ and $\widehat{\mathbf{m}}^{(\mathcal{C})}\widehat{\mathbf{d}}^{(\mathcal{C})}$ are equal to one. As a result, the dimension of the *column-space* of $\mathbf{p}^{(\mathcal{W})\,T} \otimes \widehat{\mathbf{m}}^{(\mathcal{C})}\widehat{\mathbf{d}}^{(\mathcal{C})}$ is one. Since the dimension of the *column–space* is equal to the number of linearly independent columns/rows, we conclude that Equation (6) only has one linearly independent rows.

III. Proposed Algorithm

In the previous section we describe an algebraic relationship between the coordinates of points represented in the world coordinate system and the coordinates of lines represented in the camera coordinate system, for an unknown homography matrix. However, we note that it does not contain all the known information. Since the coordinates of the points are known, the coordinates of the plane $\mathbf{\Pi}^{(\mathcal{W})} \doteq (\zeta, \boldsymbol{\pi})$ are also known which, according to the definition of the homography matrix (Equation (3)), must be taken into account on the estimation of the homography map.

However, and if the data are not corrupted with noise, these constraints need not to be taken into account in the estimation of the homography matrix. If, on the other hand, data are affected by noise the estimated homography map will be an approximation. If the constraints associated to the plane coordinates $\mathbf{\Pi}^{(\mathcal{W})}$ are not imposed, the error on the estimation of the parameters will affect the elements of $(\zeta, \boldsymbol{\pi})$, which will decrease the accuracy of the method. In the rest of this section we derive an approach which takes into account the plane parameters in the computation of the homography matrix.

Without loss of generality, we consider a rigid transformation $\left\{\widetilde{\mathbf{R}}, \widetilde{\mathbf{t}}\right\}$ of the coordinates of the world points, such that the coordinates of the 3D points and of the plane are

$$\begin{aligned} \widetilde{\mathbf{p}}_i^{(\mathcal{W})} &= \widetilde{\mathbf{R}}\mathbf{p}_i^{(\mathcal{W})} + \widetilde{\mathbf{t}}, \ \ \forall i & (7) \\ \widetilde{\mathbf{\Pi}}^{(\mathcal{W})} \doteq \left(\widetilde{\zeta}, \widetilde{\boldsymbol{\pi}}\right) &= \left(-\widetilde{\mathbf{t}}^T\widetilde{\mathbf{R}}\boldsymbol{\pi} + \zeta, \widetilde{\mathbf{R}}\boldsymbol{\pi}\right) & (8) \end{aligned}$$

such that $\widetilde{\boldsymbol{\pi}}$ is proportional to the z–axis and $\widetilde{\zeta} = 1$.

A. Estimation of the Homography Matrix

Using the representation of the world points described in the previous section and the algebraic constraints defined in Equation (6), for a set of points in the world and respective 3D lines, we aim to estimate a homography matrix $\mathbf{H}^{(1)}$, such that

$$\underbrace{\begin{bmatrix} \widetilde{\mathbf{p}}_1^{(\mathcal{W})\,T} \otimes \widehat{\mathbf{m}}_1^{(\mathcal{C})}\widehat{\mathbf{d}}_1^{(\mathcal{C})} \\ \vdots \\ \widetilde{\mathbf{p}}_N^{(\mathcal{W})\,T} \otimes \widehat{\mathbf{m}}_N^{(\mathcal{C})}\widehat{\mathbf{d}}_N^{(\mathcal{C})} \end{bmatrix}}_{\mathbf{M}} \text{vec}\left(\mathbf{H}^{(1)}\right) = \mathbf{0}. \tag{9}$$

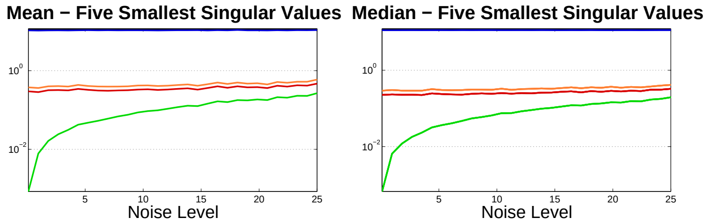


Fig. 2. In this figure we show the mean and median of the variation of the five smallest *singular values* of the matrix $\mathbf{M}$ as a function of the noise (we use the variable "Noise Level" mentioned in the section describing the experiments).

In theory, and without noise, for $N \geq 8$, we will have one *singular value* of $\mathbf{M}$ equal to zero. This means that the space of solutions for the homography matrix $\mathbf{H}^{(1)}$ is one-dimensional. In that case the solution is given by the right *singular vector* that corresponds to the zero *singular value*. Since matrix $\mathbf{H}^{(1)}$ must have the second smallest *singular value* equal to one, this condition can be used to determine the correct solution from the one-dimensional space of solutions.

However, with noisy data, and in general, no *singular value* is equal to zero. The variation of the five smallest *singular values* as a function of the noise level is shown in the Figure 2. As we can see from this figure, when the noise standard deviation increases, the three smallest *singular values* take similar values.

As a result, we select three right *singular vectors* $\left\{\boldsymbol{e}^{(1)}, \boldsymbol{e}^{(2)}, \boldsymbol{e}^{(3)}\right\}$ that correspond to the three smallest *singular values* of the matrix $\mathbf{M}^{(1)}$. Using this set of right *singular vectors* we define the space of solutions for vec $\left(\mathbf{H}^{(1)}\right)$ as

$$\text{vec}\left(\mathbf{H}^{(1)}\right) \doteq \left\{\alpha^{(1)}\boldsymbol{e}^{(1)} + \alpha^{(2)}\boldsymbol{e}^{(2)} + \alpha^{(3)}\boldsymbol{e}^{(3)} : \alpha^{(i)} \in \mathbb{R}, \forall i\right\}. \tag{10}$$

Unstacking the vectors $\boldsymbol{e}^{(i)}$ to matrices $\boldsymbol{F}^{(i)}$, we define the matrix $\mathbf{H}^{(1)}$ as a function of the unknowns $\alpha^{(i)}$, such that

$$\mathbf{H}^{(1)} = \alpha^{(1)}\boldsymbol{F}^{(1)} + \alpha^{(2)}\boldsymbol{F}^{(2)} + \alpha^{(3)}\boldsymbol{F}^{(3)}. \tag{11}$$

However and from the fact that $\widetilde{\zeta} = 1$, $\widetilde{\boldsymbol{\pi}}$ must be parallel to the z–axis and from Equation (3), the homography matrix must verify

$$\mathbf{p}^{(\mathcal{C})} = \underbrace{\left(\mathbf{R}^{(1)} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{t}^{(1)} \end{bmatrix}\right)}_{\mathbf{H}^{(1)}} \widetilde{\mathbf{p}}^{(\mathcal{W})} \tag{12}$$

where $\mathbf{R}^{(1)}$ and $\mathbf{t}^{(1)}$ are respectively the unknown rotation and translation that define pose. Since $\mathbf{R}^{(1)} \in \mathcal{SO}(3)$, $\mathbf{h}_1^{(1)} = \mathbf{r}_1^{(1)}$ and $\mathbf{h}_2^{(1)} = \mathbf{r}_2^{(1)}$ ($\mathbf{h}_i^{(1)}$ and $\mathbf{r}_i^{(1)}$ are the i^{th} columns of matrices $\mathbf{H}^{(1)}$ and $\mathbf{R}^{(1)}$ respectively), it is possible to define the following constraints that apply to the first and second column of the estimated homography matrix

$$\mathbf{h}_1^{(1)\,T}\,\mathbf{h}_1^{(1)} = 1, \quad \mathbf{h}_2^{(1)\,T}\,\mathbf{h}_2^{(1)} = 1 \text{ and } \mathbf{h}_1^{(1)\,T}\,\mathbf{h}_2^{(1)} = 0. \tag{13}$$

From the space of solutions for the homography matrix defined at Equation (11), we can define the columns $\mathbf{h}_1^{(1)}$ and $\mathbf{h}_2^{(1)}$ as

$$\mathbf{h}_1^{(1)} = \alpha^{(1)}\boldsymbol{f}_1^{(1)} + \alpha^{(2)}\boldsymbol{f}_1^{(2)} + \alpha^{(3)}\boldsymbol{f}_1^{(3)} \tag{14}$$

$$\mathbf{h}_2^{(1)} = \alpha^{(1)}\boldsymbol{f}_2^{(1)} + \alpha^{(2)}\boldsymbol{f}_2^{(2)} + \alpha^{(3)}\boldsymbol{f}_2^{(3)} \tag{15}$$

where $\boldsymbol{f}_j^{(i)}$ is the j^{th} column of the matrix $\boldsymbol{F}^{(i)}$. Without loss of generality, we can define $\widetilde{\mathbf{h}}_i^{(1)} = \mathbf{h}_i^{(1)}/\alpha^{(1)}$, which means

$$\widetilde{\mathbf{h}}_1^{(1)} = \boldsymbol{f}_1^{(1)} + b\boldsymbol{f}_1^{(2)} + c\boldsymbol{f}_1^{(3)} \text{ and } \widetilde{\mathbf{h}}_2^{(1)} = \boldsymbol{f}_2^{(1)} + b\boldsymbol{f}_2^{(2)} + c\boldsymbol{f}_2^{(3)} \tag{16}$$

and $b = \alpha^{(2)}/\alpha^{(1)}$ and $c = \alpha^{(3)}/\alpha^{(1)}$. Using this formulation we rewrite the constraints of Equation (13) as

$$\widetilde{\mathbf{h}}_1^{(1)\,T}\,\widetilde{\mathbf{h}}_2^{(1)} = 0 \text{ and } \widetilde{\mathbf{h}}_1^{(1)\,T}\,\widetilde{\mathbf{h}}_1^{(1)} - \widetilde{\mathbf{h}}_2^{(1)\,T}\,\widetilde{\mathbf{h}}_2^{(1)} = 0. \tag{17}$$

Replacing the columns of the homography matrix in these constraints using Equation (16), we define two constraints that apply to the space of the unknowns b and c. These constraints can be expressed by two functions $g_i(b,c) = 0$, for $i = 1, 2$, of the form

$$g_i(b,c) = \kappa_1^{(i)}b^2 + \kappa_2^{(i)}bc + \kappa_3^{(i)}c^2 + \kappa_4^{(i)}b + \kappa_5^{(i)}c + \kappa_6^{(i)}. \tag{18}$$

Thus, the solution for the proposed problem is the set of unknowns b and c such that

$$g_i(b,c) = g_i(b,c) = 0. \tag{19}$$

The formulation of the Equation (19) represents the estimation of the intersection points between two quadratic lines. From the *Bézout's* theorem [21], the theoretical maximum number of solution for this problem is four. In the remaining of this section we describe a method to solve this problem.

Let us consider the constraint $g_1(b,c) = 0$. Solving this equation for the unknown b we get two solutions

$$b = \frac{\mathrm{p}_1[c]}{2\kappa_1^{(1)}} \pm \frac{\mathrm{v}[c]^{1/2}}{2\kappa_1^{(1)}} \tag{20}$$

where $\mathrm{p}_1[c]$ and $\mathrm{v}[c]$ are two polynomial equations with unknown c and degrees one and two respectively.

Substituting the unknown b on $g_2(b,c) = 0$ using Equation (20), we get the constraint

$$\mathrm{p}_2[c] \pm \mathrm{p}_3[c]\mathrm{v}[c]^{1/2} = 0 \Rightarrow \mathrm{p}_2[c] = \mp\mathrm{p}_3[c]\mathrm{v}[c]^{1/2} \tag{21}$$

where the degree of the polynomial equations $\mathrm{p}_2[c]$ and $\mathrm{p}_3[c]$ are respectively two and one. Squaring both sides of Equation (21) we get

$$\mathrm{p}_2[c]^2 = \mathrm{p}_3[c]^2\mathrm{v}[c] \Rightarrow \mathrm{p}_4[c] = \mathrm{p}_2[c]^2 - \mathrm{p}_3[c]^2\mathrm{v}[c] = 0 \tag{22}$$

where the polynomial equation $\mathrm{p}_4[c]$ has degree four.

Thus, to find c that solves the problem defined by Equation (19) we just need to find the roots of the four degree polynomial equation $\mathrm{p}_4[c]$, which can be solved in closed-form (*e.g.* using the Ferrari's technique for solving the general quartic roots). For each real solution of c we get the unknown b selecting the correct solution on Equation (20).

To conclude the algorithm, we recover the solution for $\alpha^{(i)}$ using

$$\alpha^{(1)} = \pm\left|\widetilde{\mathbf{h}}_1^{(1)}\right|, \quad \alpha^{(2)} = b\alpha^{(1)} \text{ and } \alpha^{(3)} = c\alpha^{(1)}. \tag{23}$$

Note that if we have a solution array $(\alpha^{(1)}, \alpha^{(2)}, \alpha^{(3)})$, from Equations (10) and (9), the solutions array $-(\alpha^{(1)}, \alpha^{(2)}, \alpha^{(3)})$ will also verify the same contraints, and that is why we attribute both signs to $\alpha^{(1)}$.

B. Ambiguities

From the previous section, we see that we can have multiple solutions for the set of unknowns $(\alpha^{(1)}, \alpha^{(2)}, \alpha^{(3)})$.

For the computation of the solution described at Section III-A, it is only required that $N \geq 6$. However for $N \geq 8$ the dimension of the *null–space* of $\mathbf{M}^{(1)}$ will be equal or smaller than one and, as a result, for the different set of possible arrays $(\alpha^{(1)}, \alpha^{(2)}, \alpha^{(3)})$ (and as a result vec $\left(\mathbf{H}^{(1)}\right)$) we will get non–zero solutions for the algebraic relation of Equation (9). Only when the data is noiseless, we can get a single zero solution. As a result and from the set of possible solutions, we can choose the one that minimizes the Equation (9).

Note that the solutions are obtained in pairs $(\alpha^{(1)}, \alpha^{(2)}, \alpha^{(3)})$ and $-(\alpha^{(1)}, \alpha^{(2)}, \alpha^{(3)})$. Thus, two solutions will be selected from the previous paragraph. However these two solutions will generate two homography matrices. Moreover, these two solutions will be differ only with respect to the sign, $\pm\widetilde{\mathbf{H}}^{(1)}$. From Equation (4), the estimated solutions for the homography matrix must verify the following condition

$$\pm\, \widehat{\mathbf{d}}_i^{(\mathcal{C})} \widetilde{\mathbf{H}}^{(1)} \mathbf{p}_i^{(\mathcal{W})} = \mathbf{m}_i^{(\mathcal{C})}, \ \ \forall i. \tag{24}$$

As a result, we choose the sign of the estimated *homography* matrix that minimizes this equation, for all the mappings between 3D points and lines.

C. Recovery of the Pose Parameters

To recover the pose parameters, $(\mathbf{R}, \mathbf{t})$, we first have to decompose the matrix $\mathbf{H}^{(1)}$ into $\mathbf{R}^{(1)}$ and $\mathbf{t}^{(1)}$. Since $\mathbf{h}_1^{(1)} = \mathbf{r}_1^{(1)}$ and $\mathbf{h}_2^{(1)} = \mathbf{r}_2^{(1)}$ and from Equation (13), using $\mathbf{H}^{(1)}$ we can define

$$\mathbf{R}^{(1)} = \left(\begin{array}{ccc} \mathbf{h}_1^{(1)} & \mathbf{h}_2^{(1)} & \mathbf{h}_1^{(1)} \times \mathbf{h}_2^{(1)} \end{array}\right), \tag{25}$$

$$\mathbf{t}^{(1)} = \mathbf{h}_3^{(1)} - \mathbf{h}_1^{(1)} \times \mathbf{h}_2^{(1)}. \tag{26}$$

Note that the constraints defined in Equation (14) are verified, which means that $\mathbf{R}^{(1)} \in \mathcal{SO}(3)$.

To conclude, the estimation of the absolute pose $\mathbf{R}$ and $\mathbf{t}$, taking into account the rigid transformation defined by $\widetilde{\mathbf{R}}$ and $\widetilde{\mathbf{t}}$, are given by

$$\mathbf{R} = \mathbf{R}^{(1)}\widetilde{\mathbf{R}} \ \text{ and } \ \mathbf{t} = \mathbf{R}^{(1)}\widetilde{\mathbf{t}} + \mathbf{t}^{(1)}. \tag{27}$$

IV. Refinement of the Parameters

In addition to the non-iterative algorithm described in Section III-A, we propose an iterative refinement of the rotation and translation parameters that define the pose. Using the geometric distance between a 3D line and a world point,

$$\mathrm{d}\left(\mathbf{l}^{(\mathcal{C})}, \mathbf{p}^{(\mathcal{C})}\right) \doteq \frac{\left|\widehat{\mathbf{d}}^{(\mathcal{C})}\mathbf{p}^{(\mathcal{C})} - \mathbf{m}^{(\mathcal{C})}\right|}{\left|\mathbf{d}^{(\mathcal{C})}\right|}, \tag{28}$$

and since we are considering coplanar points such that $\widetilde{\boldsymbol{\pi}}$ is parallel to the z–axis and $\widetilde{\zeta} = 1$ and using Equation (3), we define the geometric distance between a world point and a 3D line as

$$\mathrm{d}\left(\mathbf{l}_i^{(\mathcal{C})}, \widetilde{\mathbf{p}}_i^{(\mathcal{W})}\right) = \frac{\left|\widehat{\mathbf{d}}_i^{(\mathcal{C})}\left(\mathbf{R}^{(1)} + \left[\begin{array}{ccc}\mathbf{0} & \mathbf{0} & \mathbf{t}^{(1)}\end{array}\right]\right)\widetilde{\mathbf{p}}^{(\mathcal{W})} - \mathbf{m}^{(\mathcal{C})}\right|}{\left|\mathbf{d}^{(\mathcal{C})}\right|}. \tag{29}$$

We aim to minimize the sum of the squares of the geometric distance defined in the previous equation

$$\underset{\mathbf{R}^{(1)}, \mathbf{t}^{(1)}}{\operatorname{argmin}} \sum_i \mathrm{d}\left(\mathbf{l}_i^{(\mathcal{C})}, \widetilde{\mathbf{p}}_i^{(\mathcal{W})}\right)^2 \tag{30}$$

for all the mappings between world points and 3D lines. We consider the rotation parametrization using *quaternions* [18].

To find the solution for Equation (30), we consider the non-iterative solution at Section III-A and use the *Quasi-Newton* optimization technique (iteration method) [1] to refine the parameters of both rotation and translation.

V. Experiments

We evaluate the proposed algorithm by comparing it to the method proposed by Schweighofer and Pinz at [17], using both synthetic and real data.

A. Synthetic Experiments

In this section we evaluate both the non-iterative algorithm and the non-iterative algorithm plus the refinement of the pose parameters, using synthetic data sets, against the state-of-the-art method. For that purpose consider a cube with 800 units of side length. The data was generated by randomly mapping 3D lines and points $\left\{\mathbf{l}_i^{(\mathcal{C})} \leftrightarrow \mathbf{p}^{(\mathcal{C})}\right\}$, for $i = 1, \ldots, N$. A random rigid transformation was randomly generated ($\mathbf{R}$ and $\mathbf{t}$, where the translation parameter is defined in the same cube with 800 units of side length) and applied to the set of points such that $\mathbf{p}^{(\mathcal{C})} \mapsto \mathbf{p}^{(\mathcal{W})}$. The dataset for the pose problem is $\left\{\mathbf{l}_i^{(\mathcal{C})} \leftrightarrow \mathbf{p}_i^{(\mathcal{W})}\right\}$.

Let us consider that the estimated pose is given by $\left\{\widehat{\mathbf{R}}, \widehat{\mathbf{t}}\right\}$. We consider both rotation and translation metrics for the computation of the error such that:

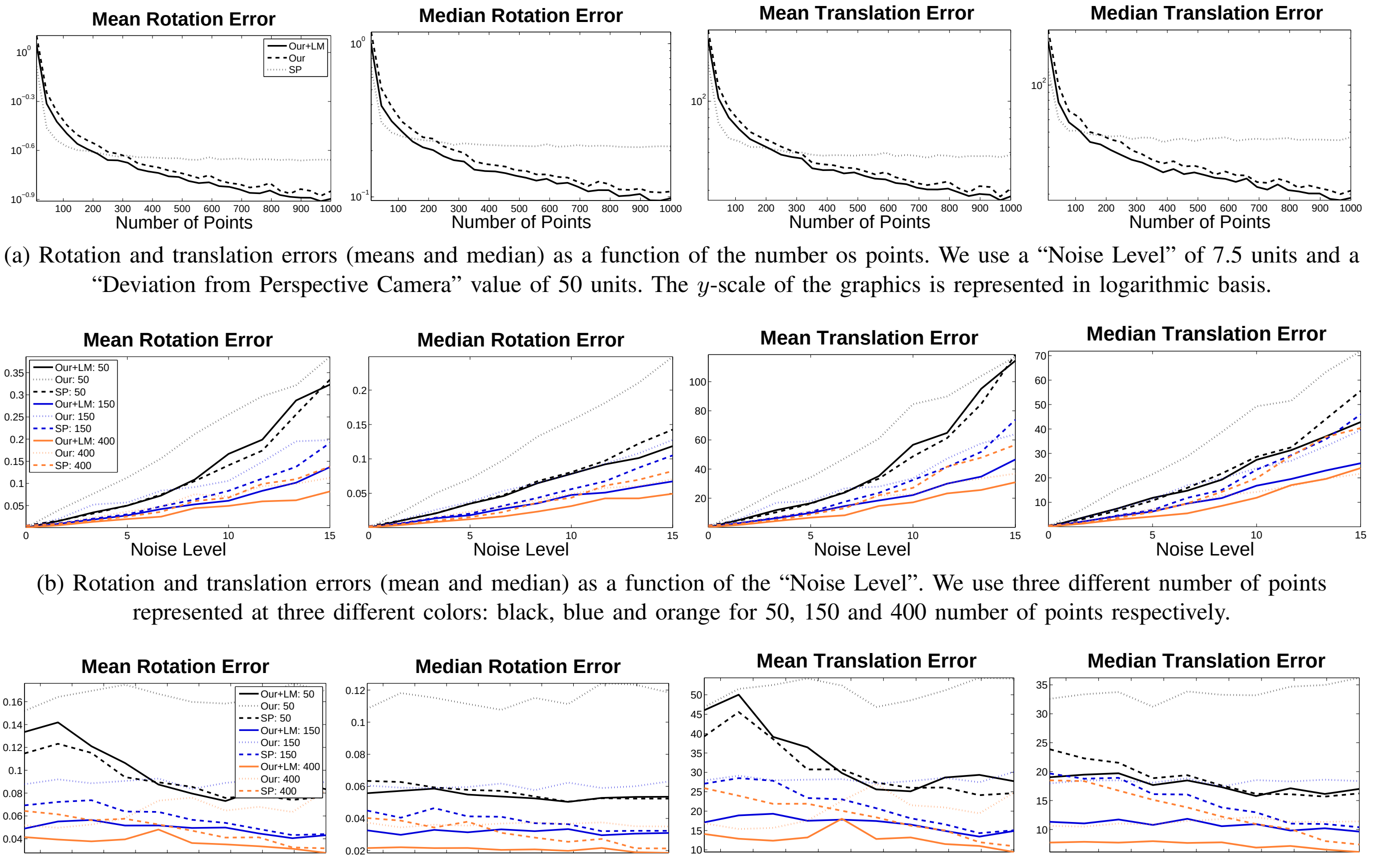
1) Rotation error: $d_{\text{rotation}} = \left|\mathbf{R} - \widehat{\mathbf{R}}\right|_{\text{frob}}$;
2) Translation error: $d_{\text{translation}} = \left|\mathbf{t} - \widehat{\mathbf{t}}\right|$.

To generate the 3D lines $\mathbf{l}_i^{(\mathcal{C})}$ we consider the following procedure. For each $\mathbf{p}_i^{(\mathcal{C})}$, an additional world point $\mathbf{q}_i^{(\mathcal{C})}$ is computed and thus, the line (in *Plücker* coordinates) $\mathbf{l}_i^{(\mathcal{C})}$ is computed as shown in [19]. A variable labeled as "Deviation From Perspective Camera" is also defined: the value of this variable represents the length of the sides of the cube to which the set of points $\left\{\mathbf{q}_i^{(\mathcal{C})}\right\}$ must belong. Note that when this value tends to zero, the camera model tends to central and that is the reason why the variable is named "Deviation From Perspective Camera".

In addition, we also define a variable to represent noise. Instead of considering the set $\left\{\mathbf{p}_i^{(\mathcal{C})}, \mathbf{q}_i^{(\mathcal{C})}\right\}$ to compute the line, we consider $\left\{\mathbf{p}_i^{(\mathcal{C})} + \mathbf{r}_i, \mathbf{q}_i^{(\mathcal{C})}\right\}$, where vector $\mathbf{r}_i$ has random direction and whose the norm is distributed according to a normal distribution whose standard deviation is variable. This variable is named "Noise Level" in the experiments.

We compare the state-of-the-art algorithm proposed by Schweighofer and Pinz at [17] with both the non-iterative method and its variation with the iterative parameter refinement method. The comparison is performed taking into account the accuracy and the processing time.

The accuracy was evaluated as a function of the number of points used to compute the pose, Figure 3(a); the "Noise Level", Figure 3(b); and the "Deviation From Perspective Camera", Figure 3(c).



(a) Rotation and translation errors (means and median) as a function of the number os points. We use a "Noise Level" of 7.5 units and a "Deviation from Perspective Camera" value of 50 units. The y-scale of the graphics is represented in logarithmic basis.

(b) Rotation and translation errors (mean and median) as a function of the "Noise Level". We use three different number of points represented at three different colors: black, blue and orange for 50, 150 and 400 number of points respectively.

(c) Rotation and translation errors (mean and median) as a function of the distribution of the 3D lines. We use a "Noise Level" of 7.5 units and three different values for the number of points (as the Figure (b)).

Fig. 3. In this figure we show the accuracy results of the application of the proposed non-iterative without and with parameters refinement algorithms compared to the state-of-the-art methods of Schweighofer and Pinz (we identify our non-iterative solution as "Our", our non-iterative solution plus a parameter refinement by "Our + LM" and the Schweighofer and Pinz algorithm as "SP"). We evaluate both algorithms in terms of: number of points used, Figure (a); in terms of the standard deviation of the noise, Figure (b); and in terms of the distribution of the 3D lines, Figure (c). In all the cases, the accuracy of the pose was measured for both rotation and translation (mean and median).

To conclude the experiments with synthetic data we show a comparison between the processing times for the proposed algorithm and for the Schweighofer and Pinz algorithm, in Figure 4. We note that our algorithm was fully implemented in MATLAB while the algorithm of Schweighofer and Pinz uses the SEDUMI optimization toolbox [22], which is implemented in C/C++.

B. Experiments with real data

In addition to the experiments with synthetic data, we evaluate our algorithm the state-of-the-art algorithms using real data.

We consider calibrated non-central spherical catadioptric camera as shown in Figure 5. Using a chess-board plane with 160 points, we get a set of images, moving the plane to different positions and with different orientations. The metrics for the errors used in this section were the same as those used in the previous section.

The results were: for our non-linear method, we obtained a mean error of 0.022 for the rotation matrix and a mean of 477.37[mm] for error on the translation vector; for our non-linear method plus with iterative refinement, we obtained a mean error of 0.0079 for estimate of the rotation matrix and 15.59[mm] for error on the translation vector; for the algorithm proposed by Schweighofer and Pinz, we obtained a mean of 0.0084 for the rotation error and 16.51[mm] for the translation error.

VI. CONCLUSIONS

A. Analysis of the Experiments

From the experiments with synthetic data we notice that the results obtained with non-iterative algorithms are, in general, worse than both non-iterative plus parameters refinement (proposed also in this article) and the iterative solution proposed by Schweighofer and Pinz. As we can see from Fig. 3(a), the results for the non-iterative case are inferior to both iterative cases when the number of points is smaller than 300.

In general, as it can be seen from Figure 3(b), the non-iterative solution yileds worse results when compared with both iterative methods. In some cases, it can be seen that the non-iterative method proposed in this paper yields better results than the iterative state-of-the-art method proposed by Schweighofer and Pinz. However and from the same figure, we can see that the non-iterative approach plus the iterative

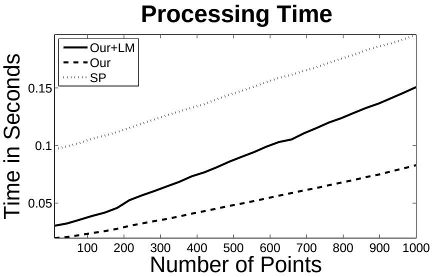


Fig. 4. In this figure we display the processing times corresponding to the method proposed by Schweighofer and Pinz and to our algorithm, as a function of the number of points. These results correspond to the experiment described in Figure 3(a). We note that while our method is fully implemented in MATLAB, the optimization of the Schweighofer and Pinz algorithm is implemented in C/C++ (we label our non-iterative solution as "Our", our non-iterative solution plus a parameter refinement by "Our + QN" and the Schweighofer and Pinz algorithm as "SP").

refinement, suggested in Sec. IV, gives better results than the non-iterative method proposed by Schweighofer and Pinz. We have to take also into account that, as shown in Fig. 4, both our methods are significantly faster than Schweighofer and Pinz approach, which by itself is an advantage.

The method non-iterative plus a parameter refinement gives better results then the non-iterative and Schweighofer and Pinz for almost all cases in both Figs 3(a) and 3(b).

To conclude the analysis of the errors, we note that from Fig. 3(c), the application of the method proposed by Schweighofer and Pinz deteriorates when the "Deviation from Perspective Camera" decreases. On the other hand, and for the non-iterative method proposed in this paper, such effect is not noticeable.

In terms of computation time (Fig. 4), the non-iterative method is clearly faster. The processing time for the non-linear plus parameter refinement tends to grow more rapidly than the processing times for the Schweighofer and Pinz algorithm.

B. Closure

In this paper we addressed the planar pose problem for non-central camera models. To the best of our knowledge, this is the first planar-based algorithm for general non-central cameras. We propose two methods: a fast non-iterative solution; and this solution plus a parameter refinement.

From the experimental results, we can conclude that both our approaches are significantly faster than the state-of-the-art method, specially the non-iterative solution. The non-iterative solution gives, in general, worse results than the state-of-the-art approach. However, the non-iterative solution plus a parameter refinement gives, in general, better results than the state-of-the-art method. In addition, we also observed that, contrarily to the state-of-the-art approach, the results given by our methods do not degrade when the camera model approximates the central camera, specially with the non-iterative approach.

References

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

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

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

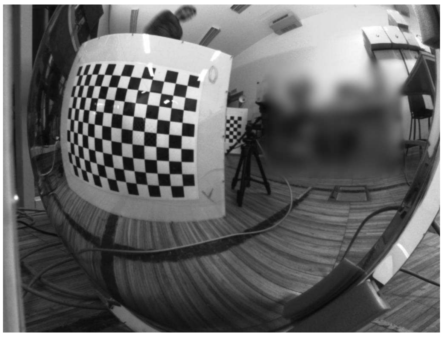


Fig. 5. In this figure we show an image of the non-central spherical catadioptric camera, used to acquire the real data.

[4] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle, "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem," *Int'l J. Computer Vision*, 1994.

[5] 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," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2011.

[6] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Commun. Assoc. Comp. Mach.*, 1981.

[7] H. Araujo, R. L. Carceroni, and C. M. Brown, "A Fully Projective Formulation to Improve the Accuracy of Lowe's Pose–Estimation Algorithm," *Computer Vision and Image Understanding*, 1998.

[8] C.-P. Lu, G. D. Hager, and E. Mjolsness, "Fast and Globally Convergent Pose Estimation from Video Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2000.

[9] F. Moreno Noguer, V. Lepetit, and P. Fua, "Accurate Non–Iteractive $O\left(n\right)$ Solution to the PnP Problem," *Proc. IEEE Int'l Conf. Computer Vision*, 2007.

[10] J. Hesch and S. Roumeliotis, "A Direct Least-Squares (DLS) Method for PnP," *Proc. IEEE Int'l Conf. Computer Vision*, 2011.

[11] A. Ansar and K. Daniilidis, "Linear Pose Estimation from Points or Lines," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2004.

[12] S. Ramalingam, S. Bouaziz, and P. Sturm, "Pose Estimation using Both Points and Lines for Geo–Localization," *Proc. IEEE Int'l Conf. Robotics and Automation*, 2011.

[13] D. Oberkampf, D. F. Dementhon, and L. S. Davis, "Iterative Pose Estimation Using Coplanar Feature Points," *Computer Vision and Image Understanding*, 1996.

[14] G. Schweighofer and A. Pinz, "Robust Pose Estimation from a Planar Target," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2006.

[15] D. Nistér, "A Minimal Solution to the Generalized 3–Point Pose Problem," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2004.

[16] C. Chu Song and C. Wen Yan, "On Pose Recovery for Generalized Visual Sensers," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2004.

[17] G. Schweighofer and A. Pinz, "Globally Optimal $O\left(n\right)$ Solution to the PnP Problem for General Camera Models," *Proc. British Machine Vision Conf.*, 2008.

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

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

[20] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd ed.)*. Johns Hopkins University Press, 1996.

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

[22] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB Toolbox for Optimization Over Symmetric Cones," 1999.