Journal of Intelligent & Robotic Systems: Special Issue – Robotica 2014. manuscript No.
(will be inserted by the editor)

Pose Estimation for Non-Central Cameras Using Planes

Pedro Miraldo · **Helder Araujo**

Submitted: December 28, 2014

Abstract In this paper we study pose estimation for general non-central cameras, using planar targets. 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. To improve the accuracy of the solutions, data-set normalization is used. In addition, we propose a parameter optimization to refine the pose estimate. We evaluate the proposed solutions against the state-of-the-art method (for general targets) in terms of both robustness to noise and computation time. From the experiments, we show that the proposed method plus normalization is more accurate against noise and less sensitive to variations of the imaging device. We also show that the numerical results obtained with this method improve with the increasing number of data points. In terms of processing speed, the versions of the algorithm presented are significantly faster than the state-of-the-art algorithm. To further evaluate our method, we performed an experiment of a simple augmented reality application in which we show that our method can be easily applied.

Keywords Absolute pose estimation · general camera models · planar patterns

The authors are with the Institute for Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, 3030-290 COIMBRA, Portugal.
Tel.: +351-239796200
Fax: +351-239796247
E-mail: {miraldo,helder}@isr.uc.pt

1 Introduction

The computation of absolute pose, using cameras, consists in the estimation of a rotation and a translation, that define the rigid transformation between the world and camera coordinate systems. Using known 3D features (such as points, lines or planes) and their corresponding images, the goal is to find the transformation that ensures that the incident relation between the inverse projection of the 2D entities is verified by the corresponding 3D features. For example, when matching 3D points and their respective images, the goal is to determine the rigid transformation so that the projection rays pass through the corresponding 3D points. One of the main applications of pose estimation using cameras is in robot navigation. If the system is properly calibrated, the estimation of the camera pose gives the localization of the robot in the world coordinate system.

Most of the methods proposed in the literature were developed for perspective cameras [14], for example, using a non-minimal number of known 3D points [4,26], a non-minimal solution using 3D lines [7,3], minimal solutions using points or both points and lines [12,16, 30] (suitable for hypothesize-and-test methods such as RANSAC [8]), and also solutions for point-based planar pose estimation [28,1,27,31]. The main reason for the use of these cameras is their simplicity, wide availability and well-known mathematical model. However, in the last decades, new types of imaging devices have started to be used due to several advantages related to their visual fields.

In 1996 Nalwa [23] introduced what he claimed to be the first omni-directional system, built using four cameras pointing to four planar mirrors, and which was designed to comply with the geometric properties of perspective cameras. Basically, the goal was to ensure that all the projection rays intersected at some 3D point–central camera systems. One of the main goals of omni-directional systems is the possibility of obtaining wide fields of view (e.g.over 180 degrees). This is specially important for applications in robot navigation, mainly because using these types of imaging devices, we can get more information about the environment using only one image. Some works on robot localization using non-conventional camera systems are described in [2,9]. Other applications are video surveillance or medical imaging devices where wide fields of view are fundamental.

In 1997, Nayar and Baker [24] studied the use of a single camera and a single quadric mirror to create omni-directional systems. Later [5], they determined the sufficient conditions to ensure that these systems fulfill the geometric properties of central cameras. The main difficulty is that, to obtain a central system (all 3D projection rays intersecting at a single point in the world, the single viewpoint), the camera must be perfectly aligned with the axis of symmetry of the mirror, and a specific type mirror must be used. For example, spherical mirrors can not be used. Systems with small misalignments or different types of mirrors will not verify the constraint that all the projection lines intersect at a single 3D point, the viewpoint. In those cases, we will have non-central camera systems–camera models that don't have a single viewpoint. This problem was analyzed by Swaminathan *et al.* [36]. In this case, the "locus of viewpoints" forms a caustic. They analyzed the properties of caustics and presented a calibration procedure for non-central conic catadioptric systems. Later, because of the utility of these imaging devices, several authors proposed models and calibration procedures for non-central catadioptric camera systems using general quadric mirrors. A recent type of camera that can be modeled as a non-central camera is the light-field camera. Most of the results obtained for general camera models can be applied to light-field cameras.

Considering only geometric entities, an imaging system can be modeled as a mapping between the 3D world and a 2D image [14]. In 2001, Grossberg and Nayar [11] defined the *general camera model*. The goal of this imaging model is to represent any imaging device (central or non-central) and it is modeled by the individual association between unconstrained 3D straight lines and 2D image pixels, for all image pixels. Thus, camera calibration consists in the estimation of the correspondences between image pixels and the corresponding projecting 3D straight lines [11,35,19].

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. 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. This problem is denoted as the pose of the camera. Most of the algorithms for the estimation of camera pose are based on targets with arbitrary 3D point configurations. In many problems such as mobile robotics and augmented reality, it is practical to use planar patterns to compute absolute pose.

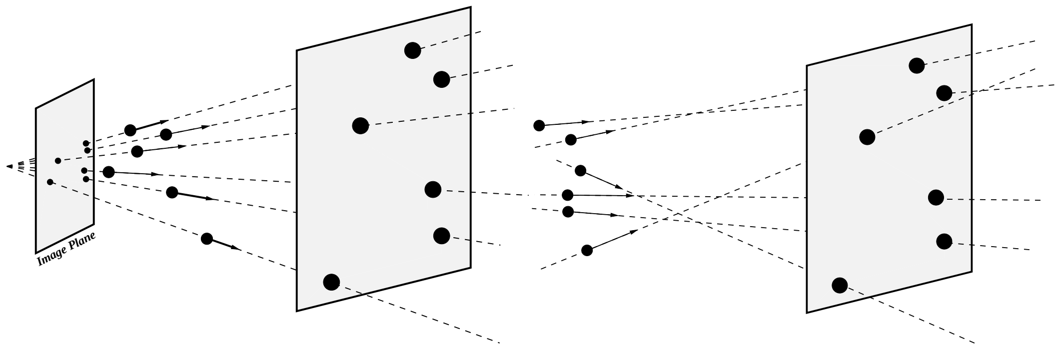
For general camera models (defined in [11]) there are algorithms to estimate pose for several conditions, namely for the minimal case [25,33], for the non-minimal case using points [33,32], and for the non-minimal case using known 3D straight lines [21,20]. In this article, we address the problem non-minimal absolute pose estimation for general non-central cameras, when considering the case where the world points belong to a plane. To the best of our knowledge, this is the first time that this problem is addressed (this paper is an extension of our paper [22]). We present a non-iterative algorithm to estimate pose. In addition, we also propose a refinement of the estimation of the pose parameters by means of an optimization using the *Levenberg-Marquardt* algorithm.

1.1 Outline of the Paper

This paper is organized as follows: in the rest of this section, we give the notation used in the paper. In Sec. 2 we briefly describe the proposed formulation. In Sec 3, we analyze the use of the homography to represent the pose and derive the constraints associated with our problem. In Sec. 4 we derive the proposed solution. To improve the accuracy of the method, in Sec. 5 we propose a data-set normalization. In Sec. 6 we propose an iterative refinement method. The experimental results are shown in Sec. 7 and the conclusions in Sec. 8.

1.2 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



(a) (b)

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

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* product we can define the following relation

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

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

2 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 proposed by Grossberg and Nayar [11], which can represent any type of imaging device (central or non-central). This model assumes that an image pixel is mapped into an arbitrary rayin 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 system. In this article we use the homography map to represent this transformation. Since we are considering a 3D point lying on a plane, we use the homography matrix to define the rigid transformation of 3D points from the world to camera coordinates. Based on the relationship of incidence between points and lines in 3D space, we define an algebraic relationship for pose. However, the homography matrix is a function of both the transformation between the world and camera coordinate system (pose of the camera) and the 3D plane parameters [17]. As a result, we divided the estimation of the homography into two steps: first we determine a space of solutions (with three degrees of freedom) for the homography matrix,and next, three constraints that the space of solutions must satisfy are defined, based on the algebraic relationship of incidence between lines and 3D points. The homography matrix is computed applying these three constraints to the three degrees of freedom space of solutions for the homography matrix.

3 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 system. 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 $\mathbf{p}^{(\mathcal{W})}$ 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 $\boldsymbol{\Pi}^{(\mathcal{W})}$, from the homography map [15, 17], we can rewrite (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 $\boldsymbol{\Pi}^{(\mathcal{W})} \doteq (\zeta, \boldsymbol{\pi}) \in \mathbb{R}^4$, $\mathbf{H} \in \mathbb{R}^{3\times 3}$ 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 $\boldsymbol{\Pi}^{(\mathcal{W})}$ 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 coordinate 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 incidence relation between a line and a point [29], we have

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

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

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

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

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

From the properties of the *Kronecker* product [10], 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 (6) only has one linearly independent row.

4 Proposed Algorithm

In the previous section we described 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 $\boldsymbol{\Pi}^{(\mathcal{W})} \doteq (\zeta, \boldsymbol{\pi})$ are also known which, according to the definition of the homography matrix (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 $\boldsymbol{\Pi}^{(\mathcal{W})}$ 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

$$\widetilde{\mathbf{p}}_i^{(\mathcal{W})} = \widetilde{\mathbf{R}}\mathbf{p}_i^{(\mathcal{W})} + \widetilde{\mathbf{t}}, \ \ \forall i \quad (7)$$

$$\widetilde{\boldsymbol{\Pi}}^{(\mathcal{W})} \doteq \left(\widetilde{\zeta}, \widetilde{\boldsymbol{\pi}}\right) = \left(-\widetilde{\mathbf{t}}^T\widetilde{\mathbf{R}}\boldsymbol{\pi} + \zeta, \widetilde{\mathbf{R}}\boldsymbol{\pi}\right) \quad (8)$$

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

4.1 Estimation of the Homography Matrix

Using the representation of the world points described in the previous section and the algebraic constraints defined in (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

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

In theory, and without noise, for $N \geqslant 8$, 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 Fig. 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

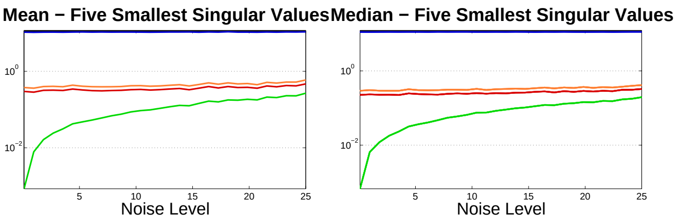


Fig. 2 Mean and median of the variation of the five smallest *singular values* of the matrix $\mathbf{M}$ as a function of the noise (we use the variable "Noise Level" mentioned in the section describing the experiments). The curves for the two largest singular values overlap (only the evaluation of singular value associated with the blue line is visible).

singular values of the matrix $\mathbf{M}^{(1)}$. Using this set of right *singular vectors* we define the space of solutions for $\text{vec}\left(\mathbf{H}^{(1)}\right)$ 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 $\tilde{\zeta} = 1$, $\tilde{\boldsymbol{\pi}}$ must be parallel to the z–axis and from (3), the homography matrix must verify

$$\mathbf{p}^{(\mathcal{C})} = \underbrace{\left(\mathbf{R}^{(1)} + \left[\mathbf{0}\ \mathbf{0}\ \mathbf{t}^{(1)}\right]\right)}_{\mathbf{H}^{(1)}} \tilde{\mathbf{p}}^{(\mathcal{W})} \tag{12}$$

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 (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 $\tilde{\mathbf{h}}_i^{(1)} = \mathbf{h}_i^{(1)}/\alpha^{(1)}$, which means

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

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

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

Replacing the columns of the homography matrix in these constraints using (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 (19) represents the estimation of the intersection points between two quadratic lines. From the *Bézout's* theorem [6], the theoretical maximum number of solutions 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 (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 (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 (19) we just need to find the roots of the fourth 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 (20).

To conclude the algorithm, we recover the solution for $\alpha^{(i)}$ (that will be used in (11)) using

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

Note that if we have a solution array $(\alpha^{(1)}, \alpha^{(2)}, \alpha^{(3)})$, from (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 $\alpha^{(1)}$.

4.2 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 in Section 4.1, it is only required that $N = 6$. However, for $N \geqslant 8$ the dimension of the *null-space* of $\mathbf{M}^{(1)}$ will be equal to one or zero. If $N \geqslant 8$ and dimension of the *null-space* is one, we will get a non-zero solution for the algebraic relation of (9). On the other hand, for $N \geqslant 9$ we can get a we can get a dimension of the null-space equals to zero. In that case, from the set of possible solutions, we can choose the one that minimizes the (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 can be considered. However these two solutions will generate two homography matrices. Moreover, these two solutions will be different only with respect to the sign, $\pm\widetilde{\mathbf{H}}^{(1)}$. From (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.

4.3 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 (12), using $\mathbf{H}^{(1)}$ we can define

$$\mathbf{R}^{(1)} = \left(\mathbf{h}_1^{(1)} \; \mathbf{h}_2^{(1)} \; \mathbf{h}_1^{(1)} \times \mathbf{h}_2^{(1)}\right), \tag{25}$$

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

Note that the constraints defined in (13) are verified, which means that $\mathbf{R}^{(1)} \in \mathcal{SO}(3)$.

Algorithm 1 Normalization of 3D points coordinates.

Let us consider a set of 3D points represented $\mathbf{p}_i^{(\mathcal{W})}$:

1. Compute $\widetilde{\mathbf{R}}$ and $\widetilde{\mathbf{t}}$ such that the z-coordinates of $\widetilde{\mathbf{p}}_i^{(\mathcal{W})} = \widetilde{\mathbf{R}}\mathbf{p}_i^{(\mathcal{W})} + \widetilde{\mathbf{t}}$ is equal to one for all i – (7).
2. Compute the positive-definite matrix $\mathbf{M} \in \mathbb{R}^{3\times 3} = \sum_{i=1}^{N} \widetilde{\mathbf{p}}_i^{(\mathcal{W})} \; \widetilde{\mathbf{p}}_i^{(\mathcal{W})\,T}$.
3. Compute the upper triangular matrix $\mathbf{K}^{(1)} \in \mathbb{R}^{3\times 3}$ such that $\mathbf{M} = N\mathbf{K}^{(1)} \; \mathbf{K}^{(1)\,T}$. Since $\mathbf{M}$ is positive-definite, $\mathbf{K}^{(1)}$ can be easily computed using the Cholesky factorization;
4. Compute normalized points $\mathbf{r}_i^{(\mathcal{W})}$ using $\mathbf{r}_i^{(\mathcal{W})} = \mathbf{K}^{(1)}\widetilde{\mathbf{p}}_i^{(\mathcal{W})}$.

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

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

5 Data-set Normalization

One of the issues in the method proposed in Sec. 4 is related to the selection of the singular values that define the space of solutions for $\mathbf{H}^{(1)}$, (10) and (11). As described in the previous section, we will select the singular vectors associated with the three smallest singular values. However, the accuracy of the solution will depend on the magnitude of these three smallest singular values.

The computation of the singular values and of the singular vectors of a matrix is affected by the condition number of the matrix [13]. In many cases, the condition number is too large,which implies that small changes on the values of the elements of the matrix will result on large changes on the singular values and singular vectors, which is an undesired effect (especially when considering data with noise). The main idea behind the data-set normalization is to decrease the condition number of matrix $\mathbf{M}$. When this condition number is small, the matrix will be well-conditioned (which means that small changes in the data will also result in small changes in the singular values and singular vectors).

Let us first consider the normalization of the 3D points that make up the data-set $\mathbf{p}_i^{(\mathcal{W})}$. Our goal is to consider nonisotropic normalization for 3D points (in the world coordinate system). To get the normalized points $\mathbf{r}_i^{(\mathcal{W})}$, we derived the algorithm 1.

The normalization of the coordinates of the 3D straight lines is not as trivial as the normalization of 3D point. Our goal is to apply an affine transformation but, in this case, to the 3D straight lines

$$\mathbf{g}_i^{(\mathcal{C})} = \begin{pmatrix} \mathbf{K}^{(2)} & \mathbf{0} \\ \mathbf{K}^{(2)} & \det\left(\mathbf{K}^{(2)}\right)\mathbf{K}^{(2)\,-T} \end{pmatrix} \mathbf{l}_i^{(\mathcal{C})} \tag{28}$$

(for more information see [18]). To get the affine parameters $\mathbf{K}^{(2)} \in \mathbb{R}^{3\times 3}$, we used the second and third points of algorithm 1 but, in this case, to the coordinates of the moments of the lines.

After the application of this normalization, we just have to perform the computation of the pose derived in Sec. 4, using $\mathbf{r}_i^{(\mathcal{W})}$ instead of $\mathbf{p}_i^{(\mathcal{W})}$ and $\mathbf{g}_i^{(\mathcal{C})}$ instead of $\mathbf{l}_i^{(\mathcal{C})}$. In addition, to ensure that the rotation and translation parameters of (27) define the real pose, we need to take into account this normalization. After this normalization and after the application of the method derived in Sec. 4, one has

$$\mathbf{K}^{(2)}\mathbf{p}^{(\mathcal{C})} = \mathbf{H}^{(1)}\mathbf{K}^{(1)}\widetilde{\mathbf{p}}^{(\mathcal{W})} \Rightarrow \mathbf{p}^{(\mathcal{C})} = \mathbf{K}^{(2)\,-1}\mathbf{H}^{(1)}\mathbf{K}^{(1)}\widetilde{\mathbf{p}}^{(\mathcal{W})}. \tag{29}$$

As a result and from (11), to reverse the proposed normalization, instead of using $\boldsymbol{F}^{(i)}$, we use $\breve{\boldsymbol{F}}^{(i)}$ such that

$$\breve{\boldsymbol{F}}^{(i)} = \mathbf{K}^{(2)\,-1}\boldsymbol{E}^{(i)}\mathbf{K}^{(1)}, \text{ for } i = 1, 2, 3. \tag{30}$$

The remaining steps of the method proposed in Sec. 4 will be the same.

6 Refinement of the Parameters

In addition to the non-iterative algorithm described in previous sections, 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,

$$\mathrm{d}\left(\mathbf{l}^{(\mathcal{C})}, \mathbf{p}^{(\mathcal{C})}\right) \doteq \frac{\left|\widehat{\mathbf{d}}^{(\mathcal{C})}\mathbf{p}^{(\mathcal{C})} - \mathbf{m}^{(\mathcal{C})}\right|}{\left|\mathbf{d}^{(\mathcal{C})}\right|} \tag{31}$$

(for more information see [18]),and since we are considering co-planar points such that $\widetilde{\boldsymbol{\pi}}$ is parallel to the z–axis and $\widetilde{\zeta} = 1$ and using (12), we define the geometric distance between a world point and a 3D line as

$$\mathrm{d}\left(\mathbf{l}_i^{(\mathcal{C})}, \widetilde{\mathbf{p}}_i^{(\mathcal{W})}\right) = \frac{\left|\widehat{\mathbf{d}}_i^{(\mathcal{C})}\left(\mathbf{R}^{(1)} + \left[\,\mathbf{0}\ \mathbf{0}\ \mathbf{t}^{(1)}\,\right]\right)\widetilde{\mathbf{p}}^{(\mathcal{W})} - \mathbf{m}^{(\mathcal{C})}\right|}{\left|\mathbf{d}^{(\mathcal{C})}\right|}. \tag{32}$$

Thus, the goal is to minimize the sum of the squares of the geometric distance defined in the previous equation

$$\underset{\mathbf{R}^{(1)},\mathbf{t}^{(1)}}{\operatorname{argmin}} \sum_i \mathrm{d}\left(\mathbf{l}_i^{(\mathcal{C})}, \widetilde{\mathbf{p}}_i^{(\mathcal{W})}\right)^2 \tag{33}$$

for all the mappings between world points and 3D lines. We consider the rotation parametrization using *quaternions* [17].

To find the solution for (33), we use the non-iterative solution proposed in Section 4 and solve the problem using *Levenberg-Marquardt* optimization technique (iteration method) [15].

7 Experimental Results

We evaluated the proposed algorithm by comparing it to the method proposed by Schweighofer and Pinz at [32], using both synthetic and real data.

7.1 Experiments with Synthetic Data

For experimental results with synthetic data, we consider the following algorithms:

– `Our`: denotes the method proposed in Sec. 4;
– `Our + N`: denotes the method proposed in Sec. 4 with the data-set normalization suggested in Sec. 5;
– `Our + LN`: denotes the method proposed in Sec. 4 with the non-linear refinement suggested in Sec. 6;
– `SP`: denotes the state-of-the-art method (for general targets) proposed by Schweighofer and Pinz at [32].

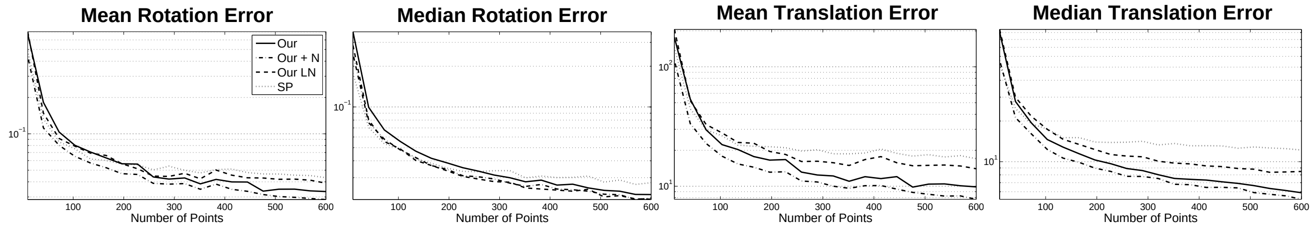
The comparison is performed taking into account the accuracy and the processing time. For that purpose we considered a cube with 800 units of side length. The data was generated by randomly mapping 3D lines and points $\left\{\mathbf{l}_i^{(\mathcal{C})} \leftrightarrow \mathbf{p}^{(\mathcal{C})}\right\}$, for $i = 1, \ldots, N$. A random rigid transformation was 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 data-set 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:

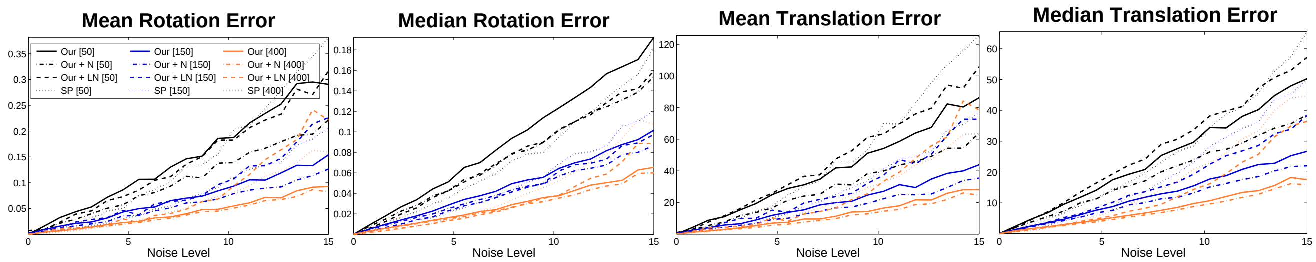
1. Rotation error: $d_{\text{rotation}} = \left|\mathbf{R} - \widehat{\mathbf{R}}\right|_{\text{frob}}$;
2. Translation error: $d_{\text{translation}} = \left|\mathbf{t} - \widehat{\mathbf{t}}\right|$.

$|.|_{\text{frob}}$ denotes the frobenius norm.

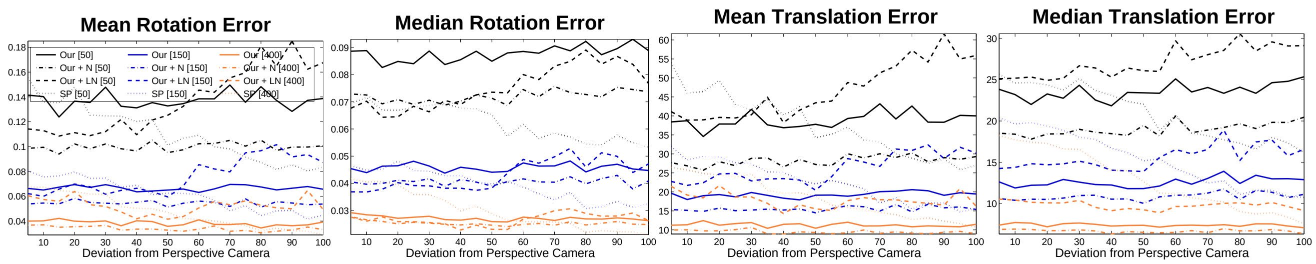
To generate the 3D lines $\mathbf{l}_i^{(\mathcal{C})}$ we used the following procedure: for each $\mathbf{p}_i^{(\mathcal{C})}$, an additional world point $\mathbf{q}_i^{(\mathcal{C})}$



(a) Rotation and translation errors (means and median) as a function of the number of 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 numbers 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 in Fig. (b)).

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

is computed and thus, the line (in *Plücker* coordinates) $\mathbf{l}_i^{(\mathcal{C})}$ is computed using

$$\mathbf{l}_i^{(\mathcal{C})} \doteq \left(\mathbf{q}_i^{(\mathcal{C})} - \mathbf{p}_i^{(\mathcal{C})}, \mathbf{p}_i^{(\mathcal{C})} \times \mathbf{q}_i^{(\mathcal{C})} \right). \quad (34)$$

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 defined 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 $\left\{ \mathbf{p}_i^{(\mathcal{C})} + \mathbf{r}_i, \mathbf{q}_i^{(\mathcal{C})} \right\}$ and the line $\mathbf{l}_i^{(\mathcal{C})}$ is, thus, given by

$$\mathbf{l}_i^{(\mathcal{C})} \doteq \left(\mathbf{q}_i^{(\mathcal{C})} - \left(\mathbf{p}_i^{(\mathcal{C})} + \mathbf{r}_i \right), \left(\mathbf{p}_i^{(\mathcal{C})} + \mathbf{r}_i \right) \times \mathbf{q}_i^{(\mathcal{C})} \right). \quad (35)$$

Vector $\mathbf{r}_i$ has random direction and its norm is distributed according to a normal distribution whose standard deviation is the value for the noise variable. This variable was named `Noise Level` in the experiments.

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

To conclude the experiments with synthetic data we show a comparison between the processing times. The results are shown in Fig. 4. The computation of `Our` and `Our + N` only differs on small direct steps which means

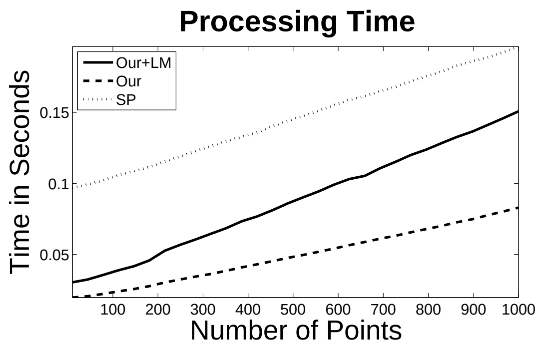


Fig. 4 Processing times corresponding to the method proposed by Schweighofer and Pinz and to our algorithm, as a function of the number of points. These results correspond to the experiment described in Fig. 3(a). We note that while our method is fully implemented in MATLAB, the optimization of the Schweighofer 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 + LN` and the Schweighofer and Pinz algorithm as `SP`).

that the difference in terms of computational time is negligible – we only show the results for `Our`. Moreover, we note that our algorithm was fully implemented in MATLAB while the algorithm of Schweighofer and Pinz uses the SEDUMI optimization toolbox [34], which is implemented in C/C++.

In addition, to further evaluate the non-linear refinement method proposed in Sec. 6, we independently evaluated its the convergence rate. To perform the evaluation data was randomly generated, as described in the previous paragraphs (noiseless). Instead of using the values estimated by the non-iterative method as initial values (Sec. 4), values differing from the ground-truth were used. These initial values were obtained by adding some pre-defined values to the ground truth. For both translation and rotation parameters ($\mathbf{t}_0$ and $\mathbf{R}_0$ respectively), we considered:

- $\mathbf{t}_0 = \mathbf{t} + \overline{\mathbf{t}}$, where $\overline{\mathbf{t}}$ is a vector with random direction (the norm will define the distance between the $\mathbf{t}$ and $\overline{\mathbf{t}}$). In the experiments, we computed this norm using a normal distribution with standard deviation equal to variable `Deviation from ground-truth translation`.
- $\mathbf{R}_0$ has yaw, pitch and roll such that $\phi_0 = \phi + \overline{\phi}$, $\theta_0 = \theta + \overline{\theta}$ and $\psi_0 = \psi + \overline{\psi}$, where ϕ, θ and ψ are the ground-truth angles, $\overline{\phi}$, $\overline{\theta}$ and $\overline{\psi}$ are angles computed using a normal distribution with standard deviation equal to variable `Deviation from ground-truth rotation`.

The results of the convergence rate as a function of both `Deviation from ground-truth translation`

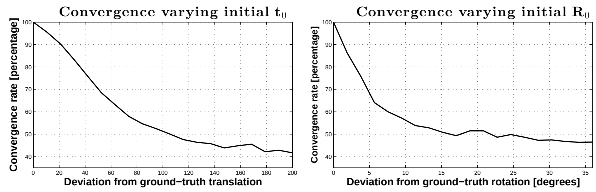
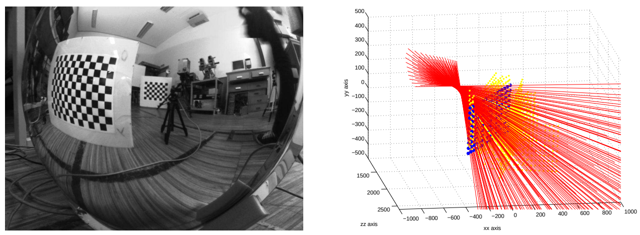


Fig. 5 Analysis of the convergence rate of the iterative refinement approach proposed in Sec. 6. To evaluate the convergence rate, we vary the initial estimate for both rotation and translation parameters. As evaluation parameter, we consider the distance from the initial estimate to the ground-truth solutions.



(a) (b)

Fig. 6 Fig. (a) shows an image of the chessboard plane using the non-central catadioptric camera. Fig. (b) shows the corresponding 3D projection lines (in red) and the corresponding 3D points (blue) in the camera coordinate system – after the application of the estimated pose.

and `Deviation from ground-truth rotation` variables are shown in Fig. 5.

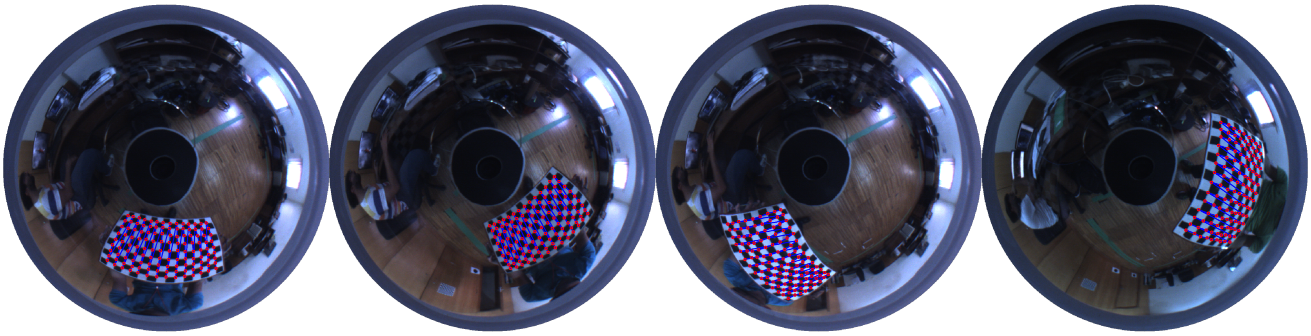
The code used to obtain these results will be available on the page of the author.

7.2 Experiments with Real Data

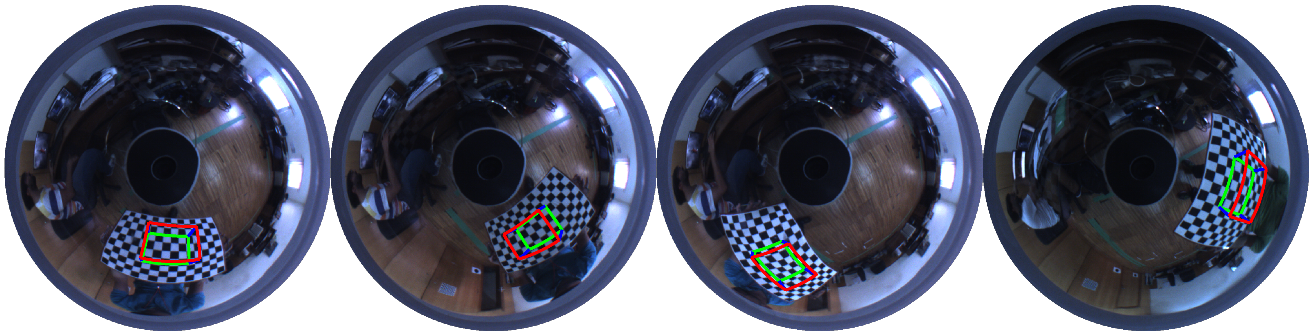
In addition to the experiments with synthetic data, we evaluated the proposed method using two experiments with real data.

7.2.1 Comparison with the State-of-the-Art Method

For these experiments, we used a calibrated non-central catadioptric camera. An example of the respective image and the associated 3D projection lines are shown in Fig. 6. Using a chessboard plane with 160 points, we get a set of images, moving the plane to different positions and with different orientations. To build the data-set we computed the 2D corners of the chessboard in the image and associated these points with the respective 3D corners in the world, for all chessboard positions. The metrics for the rotation and translation errors used in



(a) In this figure we show the computed corners (2D image points marked as pink dots) of the chessboard used for the computation of the pose. The respective 3D points are known from the problem definition.



(b) Since from pose estimation we know the position of the chessboard in the camera coordinate system, we can define a virtual object in the chessboard and project this object into the image. We define a virtual rectangular parallelepiped in the middle of the chessboard and project the coordinates of the 3D points that define the object to the image. The red and green rectangles represent the edges of the top and bottom faces of the parallelepiped and blue lines represent the side edges.

Fig. 7 Augmented reality application using a non-central catadioptric camera. Using the 2D corners of the chessboard image (a) and its size, we compute the coordinates of the chessboard in the camera coordinate system using the proposed solution for pose. Then, we generate a virtual 3D parallelepiped in the middle of the chessboard (in the camera coordinate system) and project its edges into the image. The results are shown in (b).

this section are the same as those used in the previous section.

The results were: for our non-linear method plus normalization, 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.

7.2.2 Augmented Reality Application using Real Data

To conclude, we propose a simple augmented reality application. We used a different calibrated non-central catadioptric camera and considered a sequence of four images of a known chessboard (126 corners and squares with 38[mm] of side length) in different and unknown positions and orientations. In each image, we compute the respective 2D corners of the chessboard (the 2D coordinates of the corners for each image are shown in Fig. 7(a)) and associate these points with the 3D corners of the chessboard. Then, we compute the coordinates of the chessboard in camera coordinates which is the same as to compute the pose of the camera, considering the world coordinate system attached to the chessboard. For the computation of the pose, we used the method proposed in this paper.

We generated a virtual rectangular parallelepiped in the middle of the chessboard and, for each image, we apply the rotation and translation that transform the virtual object to the camera coordinates (pose estimation). Then, we project the points that form the edges (base edges as red, top edges as green and vertical edges as blue) to the image. The results are shown in Fig. 7(b).

8 Conclusions

8.1 Discussion of the Experimental Results

As a function of noise, and from the experiments with synthetic data Fig. 3(b), we notice that the results obtained with the non-iterative method with data-set normalization gives `Our + N` significantly better results than any other algorithm, in all the experiments. This difference is greater for smaller numbers of points. In general, it can be seen that both `Our` and `Our + LN` yield better results than the iterative state-of-the-art method `SP`. However, for small levels of noise (< 7.5 units) and for small number of points one can see that `SP` gives better results than `Our` and `Our + LN`.

From Fig. 3(a), one can see that `Our + N` is also the best in all experiments. Notice that `Our` and `Our + LN` give better results for a larger number of points, mainly in terms of the translation errors (note that the error scale in these figures is on a logarithmic basis).

To conclude the analysis of the errors, we note that from Fig. 3(c), the application of the method proposed by Schweighofer and Pinz (`SP`) deteriorates when the `Deviation from Perspective Camera` decreases. These tests are very important because most of the non-central imaging devices are close to central (for example the non-central catadioptric cameras). On the other hand, for all the methods proposed in this paper such effect is not noticeable. In fact, in some cases, for `Our + LN` we can see that we get better results when the camera approximates a central configuration.

We also analyzed the convergence rate of the proposed iterative refinement approach. From Fig. 5, one can conclude that the convergence rate depends on the initial estimate, which means that a good initial estimate (in this case using the method proposed in Sec. 4) is very important for the convergence of the method.

To conclude the analysis of the experiments with synthetic data, we want to emphasize that all of the methods proposed in this paper are significantly faster than `SP` approach, which by itself is an advantage. From Fig. 4, the processing time for the non-linear plus parameter refinement `Our + LN` tends to grow more rapidly than `SP`. However, for 1000 points (very large value), the computation time for `Our + LN` is significantly lower and we also have to take into account that all of our methods were implemented with MATLAB whereas `SP` was implemented using C/C++.

In addition, we validated the proposed method using real data. We proposed a simple augmented reality application with a non-central catadioptric camera and, from Fig. 7 we proved that the proposed solution computes the pose successfully.

8.2 Closure

In this paper we addressed pose estimation for non-central cameras using planes. To the best of our knowledge, this is the first plane-based algorithm for general non-central cameras. We proposed three methods: a fast non-iterative solution; a fast non-iterative solution plus a data-set normalization; and this solution plus a parameter refinement.

From the experimental results, we can conclude that our approaches are significantly faster than the state-of-the-art method (specially the non-iterative solutions). The non-iterative solution plus data-set normalization gives significantly better results than all the other approaches. 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.

To validate the proposed solution, we implemented a simple augmented reality application which showed that our method computes pose successfully.

References

1. Aliaga, D.G.: Accurate catadioptric calibration for real-time pose estimation in room-size environments. IEEE Proc. Int'l Conf. Computer Vision (ICCV) (2001)
2. Aliakbarpour, H., Tahri, O., Araujo, H.: Visual Servoing of Mobile Robots using Non-Central Catadioptric Cameras. Robotics and Autonomous Systems (2014)
3. Ansar, A., Daniilidis, K.: Linear Pose Estimation from Points or Lines. IEEE Trans. Pattern Annalysis and Machine Intelligence (2003)
4. Araujo, H., Carceroni, R.L., Brown, C.M.: A Fully Projective Formulation to Improve the Accuracy of Lowe's Pose–Estimation Algorithm. Computer Vision and Image Understanding (1998)
5. Baker, S., Nayar, S.K.: A Theory of Single-Viewpoint Catadioptric Image Formation. International Journal of Computer Vision (1999)
6. Cox, D.A., Little, J., O'Shea, D.: Using Algebraic Geometry. Springer Science+Business (2004)
7. Dhome, M., Richetin, M., Lapreste, J.T., Rives, G.: Determination of the Attitude of 3d Objects from a Single Perspective View. IEEE Trans. Pattern Annalysis and Machine Intelligence (1989)
8. Fischler, M., Bolles, R.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Commun. Assoc. Comp. Mach. (1981)
9. Gaspar, J., Winters, N., Santos-Victor, J.: Vision-based Navigation and Environmental Representations with an Omni-directional Camera. IEEE Trans. Robotics and Automation (2000)
10. Golub, G.H., Van Loan, C.F.: Matrix Computations (3rd ed.). Johns Hopkins University Press (1996)
11. Grossberg, G., Nayar, S.: A General Imaging Model and a Method for Finding its Parameters. Proc. IEEE Int'l Conf. Computer Vision (2001)

12. Haralick, R.M., Lee, C.N., Ottenberg, K., Nölle, M.: Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem. Int'l J. Computer Vision (1994)
13. Hartley, R.: In Defense of the Eight-Point Algorithm. IEEE Trans. Pattern Annalysis and Machine Intelligence (1997)
14. Hartley, R., Zisserman, A.: Multiple View Geometry. Cambridge University Press (2000)
15. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
16. Kneip, L., Scaramuzza, D., Siegwart, R.: 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)
17. Ma, Y., Soatto, S., Košecká, J., Sastry, S.S.: An Invitation to 3D Vision: From Images to Geometry Models. Springer Science+Business (2004)
18. Miraldo, P.: General Camera Models: Calibration and Pose. Ph.D. thesis, Department of Electrical and Computer Engineering–University of Coimbra (2013)
19. Miraldo, P., Araujo, H.: Calibration of Smooth Camera Models. IEEE Trans. Pattern Annalysis and Machine Intelligence (2013)
20. Miraldo, P., Araujo, H.: Planar Pose Estimation for General Cameras using Known 3D Lines. IEEE/RSJ Pose Estimation for Non-Central Cameras Using Planes (2014)
21. Miraldo, P., Araujo, H.: Pose Estimation for General Cameras using Lines. IEEE Trans. Cybernetics (Systems, Man, and Cybernetics, Part B), to be published (2014)
22. Miraldo, P., Araujo, H.: Pose Estimation for Non-Central Cameras Using Planes. IEEE Int'l Conf. Autonomous Robot Systems & Competitions – ROBÓTICA 2014 (2014)
23. Nalwa, V.S.: A True Omni-Directional Viewer. Technichal report, Bell Laboratories (1996)
24. Nayar, S.K., Baker, S.: Catadioptric Image Formation. Proceedings of the 1997 DARPA Image Understanding Workshop (1997)
25. Nistér, D.: A Minimal Solution to the Generalized 3–Point Pose Problem. Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition (2004)
26. Moreno Noguer, F., Lepetit, V., Fua, P.: Accurate Non–Iteractive $O(n)$ Solution to the PnP Problem. Proc. IEEE Int'l Conf. Computer Vision (2007)
27. Oberkampf, D., Dementhon, D.F., Davis, L.S.: Iterative Pose Estimation Using Coplanar Feature Points. Computer Vision and Image Understanding (1996)
28. Pégard, C., Mouaddib, E.: A mobile robot using a panoramic view. IEEE Proc. Int'l Conf. Robotics & Automation (ICRA) (1996)
29. Pottmann, H., Wallner, J.: Computational Line Geometry. Springer–Verlag (2001)
30. Ramalingam, S., Bouaziz, S., Sturm, P.: Pose Estimation using Both Points and Lines for Geo–Localization. Proc. IEEE Int'l Conf. Robotics and Automation (2011)
31. Schweighofer, G., Pinz, A.: Robust Pose Estimation from a Planar Target. IEEE Trans. Pattern Analysis and Machine Intelligence (2006)
32. Schweighofer, G., Pinz, A.: Globally Optimal $O(n)$ Solution to the PnP Problem for General Camera Models. Proc. British Machine Vision Conf. (2008)
33. Chu Song, C., Wen Yan, C.: On Pose Recovery for Generalized Visual Sensers. IEEE Trans. Pattern Analysis and Machine Intelligence (2004)
34. Sturm, J.F.: Using SeDuMi 1.02, a MATLAB Toolbox for Optimization Over Symmetric Cones (1999)
35. Sturm, P., Ramalingam, S.: A Generic Concept for Camera Calibration. Proc. European Conf. Computer Vision (2004)
36. Swaminathan, R., Grossberg, M.D., Nayar, S.K.: Caustics of Catadioptric Cameras. IEEE Proc. International Conference on Computer Vision (ICCV) (2001)