

• U

C •

# UNIVERSIDADE DE COIMBRA

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

INSTITUTO DE SISTEMAS E ROBÓTICA

3030-290 COIMBRA, PORTUGAL

## ARTICLE: Planar Pose Estimation for General Cameras using Known 3D Lines

---

*Author:*

Pedro MIRALDO

miraldo@isr.uc.pt

*Co-Author:*

Helder ARAUJO

helder@isr.uc.pt

---

### CONTENTS

<b>I</b>	<b>INTRODUCTION</b>	1
I-A	Notations and Background . . . . .	1
<b>II</b>	<b>Planar Pose using Known Coordinates of 3D Lines</b>	2
<b>III</b>	<b>Experimental Results using Synthetic Data</b>	4
<b>IV</b>	<b>Robot Localization—Experiments with Real Data</b>	4
<b>V</b>	<b>Conclusions</b>	5
V-A	Analysis of the Experimental Results . . . . .	5
V-B	Closure . . . . .	7
<b>References</b>		7

---

# Planar Pose Estimation for General Cameras using Known 3D Lines

Pedro Miraldo and Helder Araujo

**Abstract**—In this article, we address the pose estimation for planar motion under the framework of generalized camera models. We assume the knowledge of the coordinates of 3D straight lines in the world coordinate system. Pose is estimated using the images of the 3D lines. This approach does not require the determination of correspondences between pixels and 3D world points. Instead, and for each pixel, it is only required that we determine to which 3D line it is associated with. Instead of identifying individual pixels, it is only necessary to establish correspondences between the pixels that belong to the images of the 3D lines, and the 3D lines. Moreover and using the assumption that the motion is planar, this paper presents a novel method for the computation of the pose using general imaging devices and assuming the knowledge of the coordinates of 3D straight lines. The approach is evaluated and validated using both synthetic data and real images. The experiments are performed using a mobile robot equipped with a non-central camera.

## I. INTRODUCTION

The problem of the absolute pose (or localization) consists in the estimation of a rotation and translation, that define the rigid transformation between the world and camera coordinate systems. Most of the methods proposed in the literature were developed for perspective cameras [1], [2], [3] (cameras that are modeled by a central projection [4]). The main reason for the use of these cameras are their simplicity, wide availability and well-known mathematical model. However, in the last few years, new types of imaging devices have started to be used due to several advantages related to their visual fields. Some of these new imaging devices that provide wide fields of view are designed and built to comply with a central projection model, eg., [5], [6]. Others are not modeled by a central projection, e.g., [7], [8]. To deal with all the cases (central and non-central cameras), Grossberg and Nayar [9] proposed the general camera model – individual association between image pixels and unconstrained 3D straight lines.

In this paper we address the problem of the planar pose of a camera, for general camera models. As far as we know, the problem of the absolute pose for general camera models was studied by Chen & Chang at [10] and [11] and Schweighofer & Pinz at [12] (we specifically do not include the works dealing with relative pose for general camera models). Both approaches use matching between known coordinates of 3D points (in the world coordinate frame) and their corresponding pixels in the image. Chen & Chang proposed a solution for the minimal case (where only three 3D points and their corresponding pixels are known) and

then derived an algorithm that computes the pose in a least-squared-error manner, using the minimal case. Schweighofer & Pinz proposed an iterative globally optimal  $O(n)$  solution to the problem. We also note that alternative solutions for the minimal case were proposed [13], [14]. In this paper, we address the non-minimal case using projection of lines.

Previous methods used the matching between known 3D points and their corresponding image points to compute the pose. However, the determination of point correspondences is still a difficult problem and current solutions are error-prone. The goal is to relax this procedure by using coordinates of 3D straight lines (one dimensional elements) defined in the world coordinate system, instead of 3D points, which constitutes one of the advantages of the proposed approach.

From the framework of general camera models, we can use the assumption that the image of a 3D straight line is made up by a set of image pixels. If we are using a smooth camera model (the variation between image pixels and the projection lines varies smoothly [15]), this set of pixels will define a continuous curve in the image plane. However, when considering non-smooth cameras, we cannot make this assumption. As a result, and since we are considering the general case, we only assume that we have a set of pixels whose coordinates are known and which correspond to a given 3D straight line in the world. We note that, in practice, the only data-set requirement is the determination of the association between a set of image pixel and a 3D straight line. There is no need to establish correspondences between pixels and 3D point features and, therefore, the matching problem becomes easier.

In our previous work [16], we used the known coordinates of 3D lines to derive a solution for the general pose – six degrees of freedom. However, specially in mobile robotics, in many instances the pose or localization of a robot are determined for the case of planar motion. In this case, we have three degrees of freedom – see Fig. 1. In this paper we address the planar pose estimation problem within the framework of the generalized camera models, using coordinates of 3D lines in the world.

In the remaining of this section, we describe the notation and background that we use in the rest of the paper. In Sec. II, we derive the proposed solution. In Sec. III, we show experimental results and, in Sec. V, we present the conclusions.

### A. Notations and Background

By default, we use small bold letters, eg.  $\mathbf{a}$ , to represent vectors. Matrices are denoted with bold capital letters, eg.  $\mathbf{A}$ . Regular small letters represent one dimensional elements. We

Both authors are with the Institute for Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, 3030-290 Coimbra, Portugal. E-Mail: {miraldo, holder}@isr.uc.pt

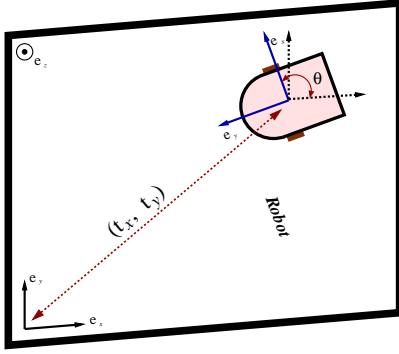


Fig. 1: Depiction of the problem addressed in this paper. The goal is to compute the translation parameters  $t_x$  and  $t_y$  and the rotation angle  $\theta$ .

use  $\sim$  to express an algebraic relation up to a scale factor. We use superscripts  $(\mathcal{W})$  and  $(\mathcal{C})$  to distinguish between the same feature in the world and camera coordinate systems, respectively.

We use the operator  $s(\cdot)$  to represent the skew-symmetric  $3 \times 3$  matrix that linearizes the exterior product, such that  $\mathbf{a} \times \mathbf{b} = s(\mathbf{a}) \mathbf{b}$ .

To represent 3D straight lines, we use the six-tuple *Plücker* coordinates [17]. A line is thus denoted by  $\mathbf{g} \sim (\hat{\mathbf{g}}, \tilde{\mathbf{g}})$  where  $\hat{\mathbf{g}} \in \mathbb{R}^3$  and  $\tilde{\mathbf{g}} \in \mathbb{R}^3$  are respectively the direction and moment of the line. Let us consider two lines  $\mathbf{g} \sim (\hat{\mathbf{g}}, \tilde{\mathbf{g}})$  and  $\mathbf{h} \sim (\hat{\mathbf{h}}, \tilde{\mathbf{h}})$ , defined in the same coordinate system. One can see that they intersect, if and only if,

$$\Omega(\mathbf{g}, \mathbf{h}) = 0 \Leftrightarrow \hat{\mathbf{g}} \cdot \tilde{\mathbf{h}} + \tilde{\mathbf{g}} \cdot \hat{\mathbf{h}} = 0. \quad (1)$$

Let us consider the rigid transformation between the world and camera coordinate systems defined by the rotation  $\mathbf{R} \in \text{SO}(3)$  and translation  $\mathbf{t} \in \mathbb{R}^3$ . Let us consider a line represented in two distinct coordinate systems  $\mathbf{g}^{(\mathcal{W})} \sim (\hat{\mathbf{g}}^{(\mathcal{W})}, \tilde{\mathbf{g}}^{(\mathcal{W})})$  and  $\mathbf{g}^{(\mathcal{C})} \sim (\hat{\mathbf{g}}^{(\mathcal{C})}, \tilde{\mathbf{g}}^{(\mathcal{C})})$ . According to [18], [19], it is possible to derive the linear operator

$$\mathbf{g}^{(\mathcal{C})} \doteq \Psi(\mathbf{g}^{(\mathcal{W})}), \quad (2)$$

that can be defined as

$$\begin{bmatrix} \hat{\mathbf{g}}^{(\mathcal{C})} \\ \tilde{\mathbf{g}}^{(\mathcal{C})} \end{bmatrix} \sim \underbrace{\begin{bmatrix} \mathbf{R} & \mathbf{0}_{3 \times 3} \\ s(\mathbf{t})\mathbf{R} & \mathbf{R} \end{bmatrix}}_{\mathbf{H} \in \mathbb{R}^{6 \times 6}} \begin{bmatrix} \hat{\mathbf{g}}^{(\mathcal{W})} \\ \tilde{\mathbf{g}}^{(\mathcal{W})} \end{bmatrix}. \quad (3)$$

## II. PLANAR POSE USING KNOWN COORDINATES OF 3D LINES

Pose estimation consists in finding the rotation and translation parameters ( $\mathbf{R}$  and translation  $\mathbf{t}$  respectively), that define the rigid transformation between the world and camera coordinate systems. In this paper, we address the case where the coordinates of 3D straight lines are known in the world coordinate system. To represent lines, we use *Plücker* coordinates (see Sec.I-A).

Assume that the coordinates of  $M$  3D straight lines are known (in the world coordinate system). As discussed in the

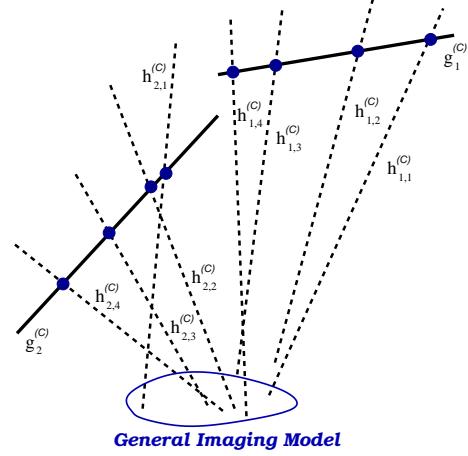


Fig. 2: Representation of the data that we use to compute the pose.  $\mathbf{g}_i^{(\mathcal{C})}$  are the known coordinates of the 3D straight lines in the world and  $\mathbf{h}_{i,j}^{(\mathcal{C})}$  are the 3D straight lines that are estimated from the pixels using the general camera model – denoted here as “projection lines”.

introduction, and since we are considering general camera models, we can only assume that the image of any 3D straight line is made up of a set of image pixels (depending on the imaging device, this set of pixels can or cannot define a continuous curve). Formally, let us consider the  $i^{\text{th}}$  known 3D straight line in the world coordinate denoted by  $\mathbf{g}_i^{(\mathcal{W})}$ . For each of the  $i^{\text{th}}$  lines, it is possible to associate a set of image pixels (here denoted by  $\{\mathbf{u}_{i,j}\}$ , for all  $j$ ), making up the data-set  $\mathbf{g}_i^{(\mathcal{W})} \leftrightarrow \{\mathbf{u}_{i,j}\}$  for all  $i$ . This set of image pixels make up the image of the 3D lines.

Moreover, and since we consider that the camera is calibrated (according to the general camera model), we know, for each and all image pixels, the corresponding 3D straight line, here denoted as “projection line”, in the camera coordinate system. To distinguish these 3D lines (in the camera coordinate system) from the known 3D lines in the world coordinate system, we denote these lines as “projection lines”. However, we would like to stress out that, since we are considering general camera models, these lines will not verify the conventional perspective projection model. Thus and formally,  $\mathbf{u}_{i,j} \mapsto \mathbf{h}_{i,j}^{(\mathcal{C})}$ , where  $\mathbf{h}_{i,j}^{(\mathcal{C})}$  denotes the “projection line” in the camera coordinate system. To conclude, the data-set is made up with  $\mathbf{g}_i^{(\mathcal{W})} \mapsto \{\mathbf{h}_{i,j}^{(\mathcal{C})}\}$  for all  $i$ , which is depicted in Fig. 2. Note that, in this framework, we never use coordinates of 3D points.

From Sec. I-A, and for each and all of the  $M$  3D straight lines ( $i^{\text{th}}$  line), and the associated  $j^{\text{th}}$  “projection lines”, the following constraint must be verified

$$\Omega(\mathbf{g}_i^{(\mathcal{C})}, \mathbf{h}_{j,i}^{(\mathcal{C})}) = 0, \quad \forall j. \quad (4)$$

Note that, for this constraint, the lines must be represented in the same coordinate system, in this case the camera coordinate system. The coordinates of the “projection lines”  $\mathbf{h}_{j,i}^{(\mathcal{C})}$  are known. On the other hand, the known coordinates of

TABLE I: In this table we show the elements of the vector  $\mathbf{c}_{i,j} \in \mathbb{R}^{18}$ .  $c_{i,j}^{(k)}$  denotes the  $k^{\text{th}}$  element of  $\mathbf{c}_{i,j}$  and  ${}^k\hat{g}_i^{(\mathcal{W})}$  denotes the  $k^{\text{th}}$  element of the vector  $\hat{\mathbf{g}}_i^{(\mathcal{W})}$ . The same denomination is used for  $\tilde{\mathbf{g}}_i^{(\mathcal{W})}$ ,  $\hat{\mathbf{h}}_{i,j}^{(\mathcal{W})}$  and  $\tilde{\mathbf{h}}_{i,j}^{(\mathcal{W})}$

$\kappa_1^{(i,j)} = {}^3\hat{h}_{i,j}^{(\mathcal{C})}(1)\hat{g}_i^{(\mathcal{W})}$	$\kappa_3^{(i,j)} = -{}^2\hat{h}_{i,j}^{(\mathcal{C})}(3)\hat{g}_i^{(\mathcal{W})}$	$\kappa_5^{(i,j)} = -{}^3\hat{h}_{i,j}^{(\mathcal{C})}(1)\hat{g}_i^{(\mathcal{W})}$
$\kappa_2^{(i,j)} = {}^3\hat{h}_{i,j}^{(\mathcal{C})}(2)\hat{g}_i^{(\mathcal{W})}$	$\kappa_4^{(i,j)} = {}^3\hat{h}_{i,j}^{(\mathcal{C})}(2)\hat{g}_i^{(\mathcal{W})}$	$\kappa_6^{(i,j)} = {}^1\hat{h}_{i,j}^{(\mathcal{C})}(3)\hat{g}_i^{(\mathcal{W})}$
$\kappa_7^{(i,j)} = {}^2\hat{h}_{i,j}^{(\mathcal{C})}(1)\tilde{g}_i^{(\mathcal{W})} - {}^1\hat{h}_{i,j}^{(\mathcal{C})}(2)\tilde{g}_i^{(\mathcal{W})} + {}^1\hat{g}_i^{(\mathcal{W})}(2)\hat{h}_{i,j}^{(\mathcal{C})} - {}^2\hat{g}_i^{(\mathcal{W})}(1)\tilde{h}_{i,j}^{(\mathcal{C})} - c^{te}(1)\hat{h}_{i,j}^{(\mathcal{C})}(1)\hat{g}_i^{(\mathcal{W})} - c^{te}(2)\hat{h}_{i,j}^{(\mathcal{C})}(2)\hat{g}_i^{(\mathcal{W})}$		
$\kappa_8^{(i,g)} = {}^1\hat{h}_{i,j}^{(\mathcal{C})}(1)\tilde{g}_i^{(\mathcal{W})} + {}^1\hat{g}_i^{(\mathcal{W})}(1)\tilde{h}_{i,j}^{(\mathcal{C})} + {}^2\hat{h}_{i,j}^{(\mathcal{C})}(2)\tilde{g}_i^{(\mathcal{W})} + {}^2\hat{g}_i^{(\mathcal{W})}(2)\tilde{h}_{i,j}^{(\mathcal{C})} - c^{te}(1)\hat{h}_{i,j}^{(\mathcal{C})}(2)\hat{g}_i^{(\mathcal{W})} + c^{te}(2)\hat{h}_{i,j}^{(\mathcal{C})}(1)\hat{g}_i^{(\mathcal{W})}$		
$\kappa_9^{(i,g)} = {}^3\hat{h}_{i,j}^{(\mathcal{C})}(3)\tilde{g}_i^{(\mathcal{W})} + {}^3\hat{g}_i^{(\mathcal{W})}(3)\tilde{h}_{i,j}^{(\mathcal{C})}$		

the 3D straight lines are expressed in the world coordinate system  $\mathbf{g}_i^{(\mathcal{W})}$ . However and as described in Sec. I-A, we can apply the rigid transformation defined in (3), to change the coordinates of these lines from the world to the camera coordinate system  $\mathbf{g}_i^{(\mathcal{W})} \mapsto \mathbf{g}_i^{(\mathcal{C})}$ . Formally, the constraint defined in (4) can be rewritten as

$$\Omega \left( \Psi \left( \mathbf{g}_i^{(\mathcal{W})} \right), \mathbf{h}_{j,i}^{(\mathcal{C})} \right) = 0, \quad \forall j. \quad (5)$$

Developing (5) using (1) and (3), for each and all of the  $i^{\text{th}}$  3D straight lines  $\mathbf{g}_i^{(\mathcal{W})}$ , we get

$$(\mathbf{R}\hat{\mathbf{g}}_i^{(\mathcal{W})}) \cdot \tilde{\mathbf{h}}_{j,i}^{(\mathcal{C})} + (\mathbf{E}\hat{\mathbf{g}}_i^{(\mathcal{W})}) \cdot \hat{\mathbf{h}}_{j,i}^{(\mathcal{C})} + (\mathbf{R}\hat{\mathbf{g}}_i^{(\mathcal{W})}) \cdot \hat{\mathbf{h}}_{j,i}^{(\mathcal{C})} = 0, \quad \forall j. \quad (6)$$

For the general pose, the six degrees of freedom have to be estimated (three for the translation and three for the rotation). However, in this paper, we are considering the case of planar motion. For this case, we have one degree of freedom for the rotation (rotation angle  $\phi$  about the  $z$ -axis) and two for the translation (translation elements  $t_x$  and  $t_y$ ) as shown in Fig. 1. Thus, the rotation matrix and translation parameters can be rewritten as

$$\mathbf{R} = \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ c^{te} \end{bmatrix}, \quad (7)$$

where  $c\theta$  and  $s\theta$  are the  $\cos(\theta)$  and  $\sin(\theta)$  respectively and  $c^{te}$  is a known constant. Thus, the unknowns of the problem are the angle  $\theta$ , and the translation elements  $t_x$  and  $t_y$ . Moreover, from (3), one can see that

$$\mathbf{s}(\mathbf{t}) \mathbf{R} = \begin{bmatrix} -c^{te}s\theta & -c^{te}c\theta & t_y \\ c^{te}c\theta & -c^{te}s\theta & -t_x \\ -t_yc\theta + t_xs\theta & t_ys\theta + t_xc\theta & 0 \end{bmatrix} \quad (8)$$

Developing (6) using  $\mathbf{E}$ , from (8), and  $\mathbf{R}$ , from (7), we obtain the constraint

$$\boldsymbol{\kappa}^{(i,j)} \mathbf{v} = 0 \quad (9)$$

where  $\mathbf{v}$ ,  $\boldsymbol{\kappa}^{(i,j)} \in \mathbb{R}^9$ ,

$$\mathbf{v} = [t_xs\theta \quad t_xc\theta \quad t_x \quad t_ys\theta \quad t_yc\theta \quad t_y \quad s\theta \quad c\theta \quad 1]^T \quad (10)$$

are the unknown vector, and the parameters of the vector  $\boldsymbol{\kappa}^{(i,j)}$  are shown in Tab. I.

Using this result, for a set of correspondences between known coordinates of 3D straight lines in the world coordinate system and a set of associated “projections lines” in the camera coordinate system, we can define the system

$$\underbrace{\begin{bmatrix} \boldsymbol{\kappa}^{(1,1)} \\ \boldsymbol{\kappa}^{(1,2)} \\ \vdots \\ \boldsymbol{\kappa}^{(1,9)} \end{bmatrix}}_{\mathbf{K}} \mathbf{v} = \mathbf{0} \quad (11)$$

This problem, however, can not be computed using a simple “least-squares solution of the homogeneous equation (11)” [4]. From the unknown vector (10), the following constraints  $c_i(\mathbf{v}) = 0$  can be easily derived

$$c_1(\mathbf{v}) = v_1 - v_3v_7 = 0 \quad (12)$$

$$c_2(\mathbf{v}) = v_2 - v_3v_8 = 0 \quad (13)$$

$$c_3(\mathbf{v}) = v_4 - v_6v_7 = 0 \quad (14)$$

$$c_4(\mathbf{v}) = v_5 - v_6v_8 = 0. \quad (15)$$

where  $v_i$  are the  $i^{\text{th}}$  element of the vector  $\mathbf{v}$ , of (10). Moreover and from the trigonometric constraint  $s\theta^2 + c\theta^2 = 1$ , a new constraint can be defined

$$c_5(\mathbf{v}) = v_7v_7 + v_8v_8 - 1 = 0. \quad (16)$$

To conclude, the solution for the problem can be computed, in the least-squares sense, using the formulation

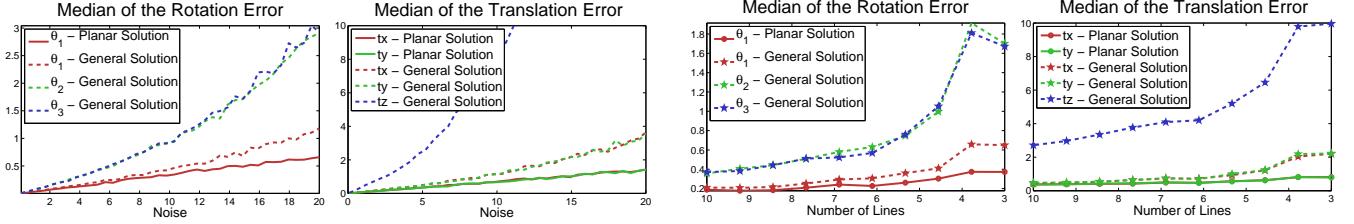
$$\begin{aligned} & \underset{\mathbf{v}}{\operatorname{argmin}} \quad f(\mathbf{v}) \\ & \text{subject to} \quad c_i(\mathbf{v}) = 0, \quad i = 1, \dots, 5. \end{aligned} \quad (17)$$

where

$$f(\mathbf{v}) = \|\mathbf{K}\mathbf{v}\|^2 = \mathbf{v}^T \mathbf{K}^T \mathbf{K} \mathbf{v}. \quad (18)$$

This problem is well known and it can be easily solved using standard toolboxes, such as the MATLAB optimization toolbox or the NLOPT toolbox [20]. The computation time of the problem is analyzed in the section relative to the analysis of experimental results, Sec. V-A.

To evaluate and validate the proposed approach, the method was tested using both synthetic and real data. We used synthetic data-sets to evaluate the method subject to noise and to different number of known 3D lines, in the world coordinate system – see Sec. III. To validate the approach, we also used real data. We used a Pioneer 3-DX (from MobileRobots) in the experiments. The localization



(a) Evaluation of the proposed approach as a function of the Noise variable. For this purpose, we consider the data-set with  $M = 6$ ,  $N_i = 20$ ,  $\forall i$ . For more information, see Sec. III

Fig. 3: In this figure, we evaluate the method proposed in this article (planar pose estimation) against our previous approach [16] (general pose estimation). We consider the evaluation of both data with noise Fig. (a) and with different number 3D lines in the world, denoted by  $g_i^{(W)}$ , Fig. (b). While for the method proposed in this paper, we only have three degrees of freedom (here denoted by the rotation angle  $\theta_1$  and translation elements  $t_x$  and  $t_y$ ), for the general case one needs to evaluate the six degrees of freedom (one has to consider more two rotation angles  $\theta_2$ ,  $\theta_3$  and an additional translation parameter  $t_z$ ). For each value of the evaluation variable, we consider  $10^3$  trials as described in the text. Moreover, the errors associated with the rotation parameters are in degrees.

of the robot on a path, and also the control its motion, were performed using data provided by this algorithm. Control was performed using well-known methods for motion control of non-holonomic robot [21]. Additional details regarding these experiments are presented in Sec. IV.

### III. EXPERIMENTAL RESULTS USING SYNTHETIC DATA

To evaluate the proposed approach, we consider both data with noise and also a different number of known 3D straight lines  $g_i^{(W)}$ . As a comparison measure, we evaluate the method proposed in this paper – which computes the pose for the planar case – against our previous approach [16] – which computes the pose for the general case. We use Matlab to obtain the experimental results presented in this section (the code will be available on the author’s page).

To generate the synthetic data-set, we used the following procedure:  $M$  3D straight lines were randomly generated  $g_i^{(C)}$ . To obtain these lines we randomly generate 3D points  $\tilde{g}_i^{(C)} \in \mathbb{R}^3$  (in a cube with 200 units of side length) and random directions  $\hat{g}_i^{(C)}$  (with norm equal to one). Using this framework and from [17], the *Plücker* coordinates of the known 3D straight lines are given by

$$g_i^{(C)} \doteq (\hat{g}_i^{(C)}, \tilde{g}_i^{(C)} \times \hat{g}_i^{(C)}), \quad \forall i = 1, \dots, M. \quad (19)$$

The next step of the data-set generation is the computation of  $N_i$  “projection lines”  $h_{j,i}^{(C)}$ . We randomly chose parameters  $\check{\mu}_{i,j}$  and compute the coordinates of 3D points  $\check{h}_{j,i}^{(C)} = \tilde{g}_i^{(C)} + \check{\mu}_{i,j} \hat{g}_i^{(C)}$  (note that this belongs to the line  $g_i^{(C)}$ ). The set of parameters  $\check{\mu}_{i,j}$  is randomly chosen, from  $-100$  to  $100$ . The set of directions  $\hat{h}_{j,i}^{(C)}$  is randomly computed too. Using this data, for the  $i^{\text{th}}$  single 3D straight line (denoted as  $g_i^{(C)}$ ), *Plücker* coordinates of the “projection lines” are thus computed as

$$h_{j,i}^{(C)} \doteq (\hat{h}_{j,i}^{(C)}, \check{h}_{j,i}^{(C)} \times \hat{h}_{j,i}^{(C)}), \quad \forall j = 1, \dots, N_i. \quad (20)$$

From the definition, the set of lines  $h_{j,i}^{(C)}$  will intersect the  $i^{\text{th}}$  line  $g_i^{(C)}$ .

(b) Evaluation of the proposed approach as a function of the number of 3D straight lines used, here denoted as  $M$ . We use  $N_i = 20$ ,  $\forall i$  and Noise variable with value equals to 7.

To conclude, random ground truth rotations and translation parameters (7) are generated and applied to the 3D straight lines  $g_i^{(C)} \mapsto g_i^{(W)}$ , using  $g_i^{(W)} = \Psi^{-1}(g_i^{(C)})$  – see (3). Note that, here, rotation and translation have three degrees of freedom. The pose is thus computed using the association between  $g_i^{(W)} \leftrightarrow \{h_{1,i}^{(C)}, \dots, h_{N_i,i}^{(C)}\}$ , for all  $i$ . With this, we conclude the data-set generation as is shown in Fig. 2.

The goal of this section is to evaluate the proposed approach for data with noise (for different levels of noise and for different number of known 3D lines). As a result, we consider the following procedure. Instead of considering “projection lines” as described in (20), we use

$$h_{j,i}^{(C)} \doteq (\hat{h}_{j,i}^{(C)}, (\check{h}_{j,i}^{(C)} + e_{j,i}^{(C)}) \times \hat{h}_{j,i}^{(C)}). \quad (21)$$

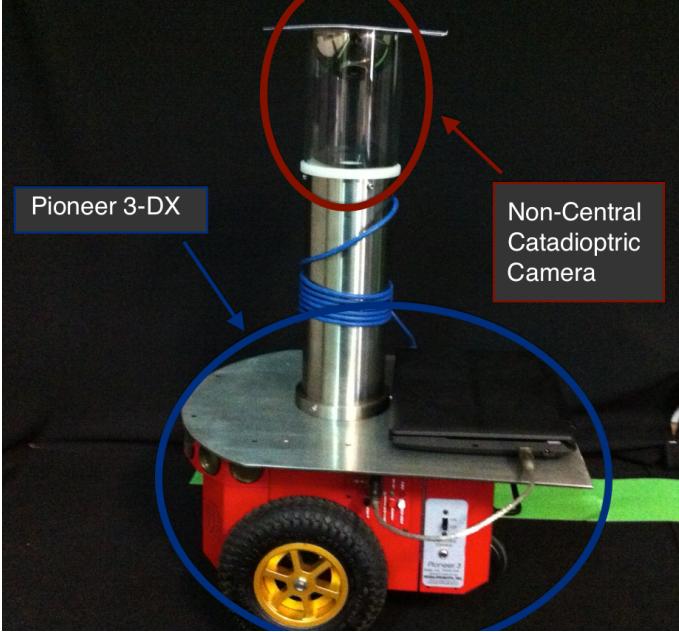
The vector  $e_{j,i}^{(C)}$  has random direction and random norm, with standard deviation equal to the Noise variable. When this variable increases, the noise of the data-set will be bigger and vice versa.

The results are shown in Fig. 3. Let us first consider the evaluation for different levels of noise. (we vary the Noise variable from 0 to 20). For these experiments, we consider a fixed value for the number of known 3D lines in the world ( $M = 6$ ) and a fixed value for the number of “projection lines” ( $N_i = 20$ ). The results are shown in Fig. 3(a).

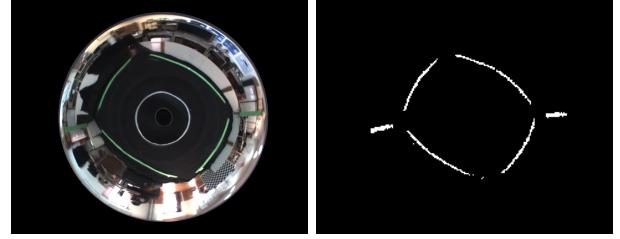
For the case of the data with different number of 3D lines in the world (variable  $N$  as described in (19)), we consider a fixed value for the number of “projection lines” ( $N_i = 20$  in (20)) and a Noise variable with the value of 7. The results are shown in Fig. 3(b).

### IV. ROBOT LOCALIZATION–EXPERIMENTS WITH REAL DATA

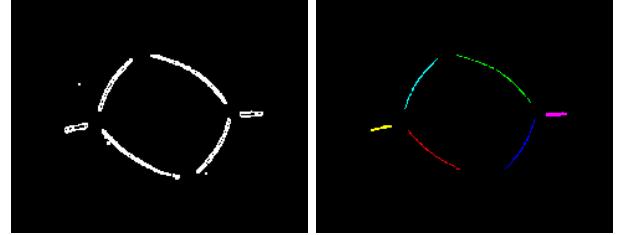
For experiments with real data, we considered a problem of visual navigation. We used a Pioneer-3DX mobile robot (from Mobile Robots [22]) with a non-central catadioptric camera, made up by a perspective camera and a spherical mirror. For the calibration of the non-central catadioptric



(a) In this figure, we show the setup used in the robot localization experiments – Pioneer 3-DX robot and a non-central catadioptric imaging system.



(b) The picture on the left is an original image, taken from the non-central camera model shown in Fig. (a). On the right, we show the results of the application of the color filter.



(c) In the picture on the left, we show an example of the contour extraction of the color filtered image from (b). The contour is here used to define the Blobs. On the right, we show an example of the association of the Blobs/curves by colors – these curves will be associated with 3D straight lines in the world.

Fig. 4: In Fig. (a), we show the robot setup – the Pioneer 3-DX mobile robot with a non-central catadioptric camera. In Figs. (b) and (c) we show an example of the image processing steps that we used to get the data-set – association between a set of image pixels with 3D straight lines in the world.

camera, we used the method proposed by Perdigoto and Araujo in [23]. The setup is shown in Fig. 4(a).

To get the data-set, we considered six green 3D straight lines in the world – four 3D lines on the ground and two 3D vertical lines. An example of an image of these lines is shown in Fig. 4(b). To complete the acquisition of the data-set, we have to associate pixels in the image to these 3D straight lines. To get these pixels, we used the following procedure:

- 1) *Color Filtering*: The first step of the image processing is to apply a color filter (we used green lines for that purpose). In addition, and to remove errors from the filtering process, we apply morphological operators – an example is shown in Fig. 4(b);
- 2) *Object Identification*: We detect/identify Binary Large OBjects (Blobs) by extracting the closed contours in the image obtained in the previous step – an example is shown in Fig. 4(c);
- 3) *Associate Blobs with 3D lines*: For the first frame, we associate the Blobs to the 3D lines manually. For the next frames, this procedure is performed automatically, by selecting the Blob in the new frame which is closest to the corresponding Blob in the previous frame;
- 4) *Associate 2D pixels with 3D lines*: For each 3D straight line, we select a set of random 75 image pixels from its associated Blob. As a result, we have a set of 2D image pixels associated with 3D straight lines

$(g_i^{(W)} \leftrightarrow \{h_{1,i}^{(C)}, \dots, h_{75,i}^{(C)}\}$ , for all  $i$ ), concluding the acquisition of the data-set.

All of these image processing operations were implemented in C++, using OpenCV. We defined two types motions for the robot: a circular motion and a S-motion. In a different process, we implemented the optimization procedure described in (17), using the C++ NLOpt optimization toolbox [20] (the computation time of the complete algorithm is detailed in Sec. V-A).

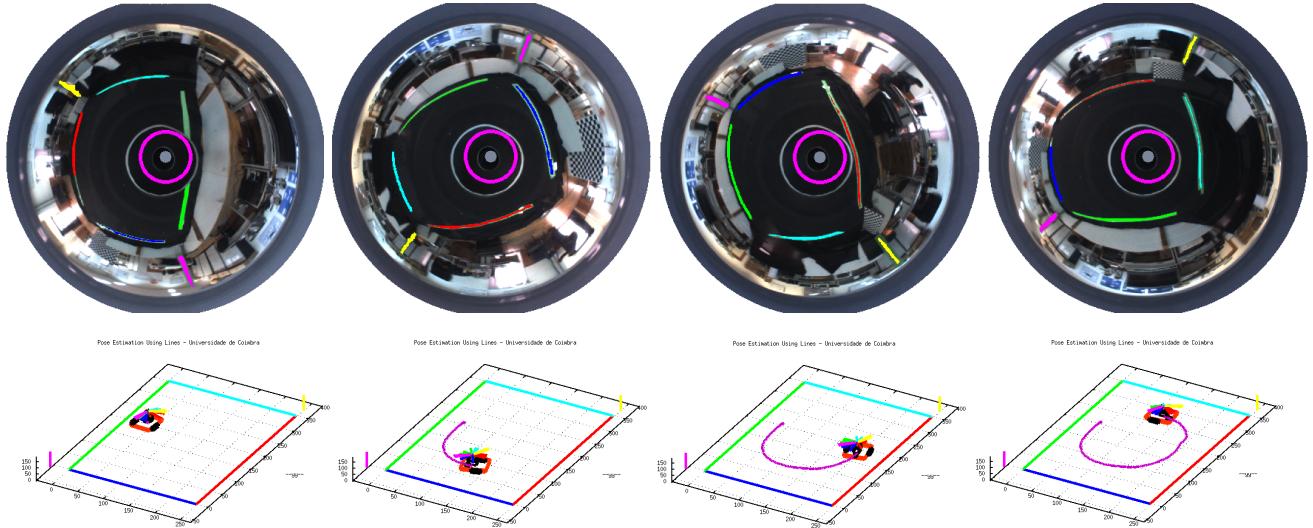
The results of these experiments are shown in Fig. 5. Let us first consider the circular motion. In the first row of Fig. 5(a), we show four examples of the images acquired by the non-central catadioptric camera with colors identifying the blobs associated with the 3D lines. The 3D straight lines are shown in the second row of Fig. 5(a), with the respective colors. To conclude, also in the 3D plots, we show the reconstruction of the motion of the robot in the world. In Fig. 5(b), we have the same information but for an S-motion.

In addition to the images of Fig. 5, we recorded a video, with the reconstruction of the position through the path, which we attach as supplementary material. This video will also be available on the author's page.

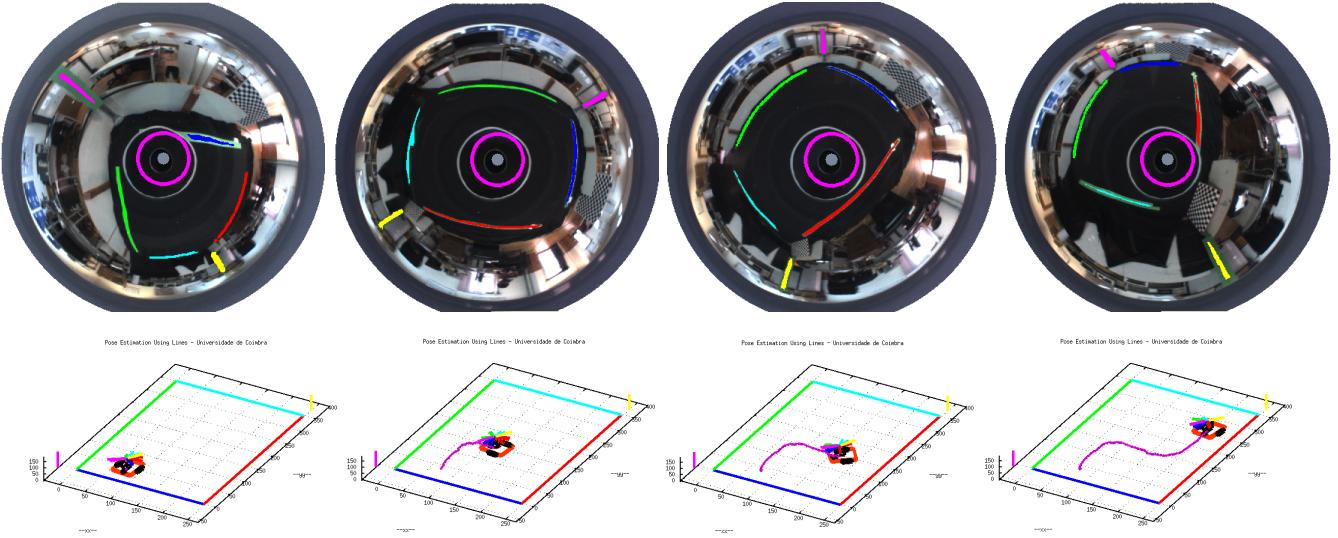
## V. CONCLUSIONS

### A. Analysis of the Experimental Results

In this section, we analyze the experimental results of Sec. III and IV.



(a) In this figure, we show the results for the localization of the robot with a circular motion. In the first row we show four images taken with the non-central catadioptric camera – in which we can see the tracking curves associated with the 3D straight lines– and, in the second row, we show the reconstruction of the motion in 3D.



(b) Similar to the case described in (a) but, in this case, for a mixture between a straight and circular motion.

Fig. 5: In this figure, we show the results of the application of the proposed method in a robot localization problem. We use a Pioneer 3-DX robot with a non-central catadioptric system, made up by a perspective camera and a spherical mirror. We consider a circular motion (a) and a mixture between straight and circular motion (b).

The goal of the experiments with synthetic data is to evaluate the robustness of the proposed approach against noise and also as a function of different numbers of known 3D straight lines in the world. To better understand the advantage to study the planar case, we compare the proposed method with our previous approach, general case [16]. From the experimental evaluation shown in Fig. 3(a), we can see that, for noisy data, the proposed approach performs significantly better when compared to the general case. Of course, it is obvious that in the case of planar motion only three degrees of freedom exist, which simplifies the problem. As we can see, from Fig. 3(a), the errors associated with the additional degrees of freedom (two angles  $\theta_2$ ,  $\theta_3$  and a translation element  $t_z$ ) are highly sensitive to noise.

Moreover and as we can see from these figures, even in the other cases (the angle  $\theta_1$  and the two translation elements  $t_x$  and  $t_y$ ) the planar solution is much more robust than the general solution. For example, for the translation parameters, the planar solution is up to two times more robust than the general solution. The same analysis can be made for the experiments using a different number of known 3D lines in the world coordinates, Fig. 3(b). As we can see, in the general solution, the three additional degrees of freedom are highly sensitive to noise. Similar to the noise evaluation, the planar solution is also less sensitive to the variation on the number of 3D lines used.

To conclude the analysis of the experiments using synthetic data, we will discuss the computation time spent in the

computation of both the planar motion parameters and also in the case of general motion parameters. We recorded all the computation times for the experiments of Fig. 3(a). For these experiments, we used the Matlab toolbox and an *intel i7-3930K* CPU. The average of the computation times for planar motion model proposed in this paper is 0.0225 [s]. On the other hand, the computation time for the general case was 0.0517 [s]. Thus, we can conclude that the method proposed in this paper is significantly faster than the general case (more than twice faster). Note that, since these methods are all iterative, the initial solution used for the optimization process is very important. We ensured that the initial solutions used for both types of motion models were the same.

In what concerns the experiments with real data, Sec. IV, we considered robot navigation examples. From the results of Fig. 5 and from the video sent as supplementary material, one can see that the method has good performance. In this analysis, we want to emphasize that, with the exception of the non-central catadioptric system, no further sensory information was used. For the experiments with the robot, we used an *intel i7-3630QM* CPU. Both image processing and pose estimation procedures were implemented in C++. The average of the computation time spent on the optimization step was 0.00531 [s].

### B. Closure

In this article, we presented a novel approach to problem of planar pose estimation for generalized camera models. This method can be applied to any imaging device, such as: the conventional perspective cameras, central catadioptric cameras, non-central catadioptric camera, or even cameras with refraction elements. To the best of our knowledge, this problem (using 3D straight lines) was not addressed before.

One of the main advantages of the proposed approach is the use of correspondences between known 3D straight lines and pixels that belong to their images. This approach significantly reduces the difficulties in the acquisition of the data set in particular because it does not require the determination of correspondences between 3D points and their images.

To conclude, when considering planar motion, we prove that the planar solution proposed in this paper is more robust against noise and also against the variation of the data-set than the general solution. To validate the method, we tested our approach in a visual navigation problem using a mobile robot.

### REFERENCES

- [1] J. Hesch and S. Roumeliotis, “A Direct Least-Squares (DLS) Method for PnP,” *IEEE Proc. Int'l Conf. Computer Vision (ICCV)*, 2011.
- [2] F. Moreno Noguer, V. Lepetit, and P. Fua, “Accurate Non-Iteractive  $O(n)$  Solution to the PnP Problem,” *IEEE Proc. Int'l Conf. Computer Vision (ICCV)*, 2007.
- [3] 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.
- [4] R. Hartley and A. Zisserman, *Multiple View Geometry*. Cambridge University Press, 2000.
- [5] S. Baker and S. K. Nayar, “A Theory of Single-Viewpoint Catadioptric Image Formation,” *Int'l J. of Computer Vision*, 1999.
- [6] C. Geyer and K. Daniilidis, “A unifying theory for central panoramic systems and practical implications,” *Proc. European Conf. Computer Vision (ECCV)*, 2000.
- [7] R. Swaminathan, M. D. Grossberg, and S. K. Nayar, “Non-Single Viewpoint Catadioptric Cameras: Geometry and Analysis,” *Int'l J. of Computer Vision*, 2006.
- [8] A. Agrawal, Y. Taguchi, and S. Ramalingam, “Beyond Alhazens Problem: Analytical Projection Model for Non-Central Catadioptric Cameras with Quadric Mirrors,” *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [9] M. D. Grossberg and S. K. Nayar, “The Raxel Imaging Model and Ray-Based Calibration,” *Int'l J. of Computer Vision*, 2005.
- [10] C.-S. Chen and W.-Y. Chang, “Pose Estimation for Generalized Imaging Device via Solving Non-Perspective N-Point Problem,” *IEEE Proc. Int'l Conf. Robotics & Automation (ICRA)*, 2002.
- [11] ———, “On Pose Recovery for Generalized Visual Sensors,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2004.
- [12] G. Schreiberhofer and A. Pinz, “Globally Optimal  $O(n)$  Solution to the PnP Problem for General Camera Models,” *Proc. British Machine Vision Conf. (BMVC)*, 2008.
- [13] D. Nistér and H. Stewnius, “A Minimal Solution to the Generalised 3-Point Pose Problem,” *J. Math. Imaging Vis.*, 2007.
- [14] P. Miraldo and H. Araujo, “A Simple and Robust Solution to the Minimal General Pose Estimation,” *IEEE Proc. Int'l Conf. Robotics & Automation*, 2014.
- [15] ———, “Calibration of Smooth Camera Models,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2013.
- [16] ———, “Pose Estimation for General Cameras using Lines,” *Tech. Report – Please see the author's page*, 2013.
- [17] H. Pottmann and J. Wallner, *Computational Line Geometry*. Springer-Verlag, 2001.
- [18] R. Pless, “Using Many Cameras as One,” *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2003.
- [19] P. Sturm, “Multi-View Geometry for General Camera Models,” *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2005.
- [20] S. G. Johnson, J.-S. Roy, and M. J. D. Powell, “The NLOpt Nonlinear-Optimization Package.” [Online]. Available: <http://ab-initio.mit.edu/nlopt>
- [21] J. J. Park and B. Kuipers, “A smooth control law for graceful motion of differential wheeled mobile robots in 2d environment,” *IEEE Proc. Int'l Conf. Robotics & Automation*, 2011.
- [22] MobileRobots, Inc., “Pioneer 3-DX.” [Online]. Available: [http://www.mobilerobots.com/Mobile\\_Robots.aspx](http://www.mobilerobots.com/Mobile_Robots.aspx)
- [23] L. Perdigoto and H. Araujo, “Calibration of mirror position and extrinsic parameters in axial non-central catadioptric systems,” *Computer Vision and Image Understanding*, 2013.