

# Camera Pose Estimation Using Images of Planar Mirror Reflections

Rui Rodrigues, João P. Barreto, and Urbano Nunes

Institute of Systems and Robotics, Dept. of Electrical and Computer Engineering,  
University of Coimbra, 3030 Coimbra, Portugal  
{rrodrigues, jpbar, urbano}@isr.uc.pt

**Abstract.** The image of a planar mirror reflection (IPMR) can be interpreted as a virtual view of the scene, acquired by a camera with a pose symmetric to the pose of the real camera with respect to the mirror plane. The epipolar geometry of virtual views associated with different IPMRs is well understood, and it is possible to recover the camera motion and perform 3D scene reconstruction by applying standard structure-from-motion methods that use image correspondences as input. In this article we address the problem of estimating the pose of the real camera, as well as the positions of the mirror plane, by assuming that the rigid motion between  $N$  virtual views induced by planar mirror reflections is known. The solution of this problem enables the registration of objects lying outside the camera field-of-view, which can have important applications in domains like non-overlapping camera network calibration and robot vision. We show that the positions of the mirror planes can be uniquely determined by solving a system of linear equations. This enables to estimate the pose of the real camera in a straightforward closed-form manner using a minimum of  $N = 3$  virtual views. Both synthetic tests and real experiments show the superiority of our approach with respect to current state-of-the-art methods.

## 1 Introduction

It is well known that the image of a planar mirror reflection (IPMR) is equivalent to the image that would be acquired by a virtual camera located behind the mirror. Such virtual camera has a pose symmetric to the pose of the real camera with respect to the plane of reflection, and presents the same intrinsic parameters [1]. We can use a planar mirror for observing a scene from a viewpoint different from the actual viewpoint of the imaging device. This has been explored in the past for building planar catadioptric systems (PCS) able to provide multi-view imagery while using a single static camera, [2][3][4]. All these works assumed a specific mirror configuration for computing the rigid transformations between virtual views and achieve extrinsic calibration.

Gluckman and Nayar were the first authors studying the epipolar geometry between virtual views induced by planar mirror reflections with the mirrors being in an arbitrary configuration [5]. They proved that in general a pair of IPMRs is related by a fundamental matrix, and that the rigid displacement between the corresponding virtual cameras is always a planar motion. Since in a planar motion the translation is orthogonal to the rotation axis, then the fundamental matrix between IPMRs has only 6 degrees of freedom (DOF) [6].

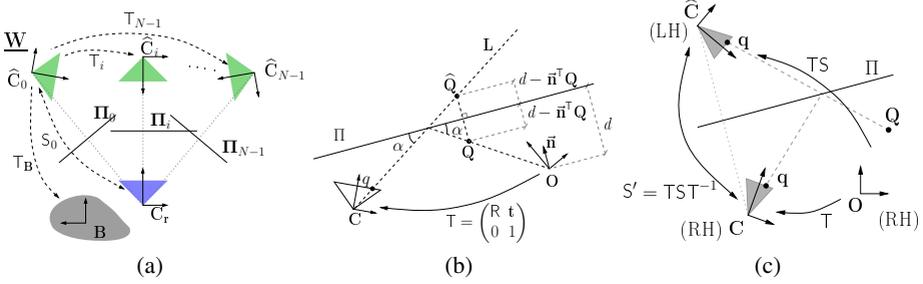
Without loss of generality, let's consider a sequence of IPMRs obtained by moving a planar mirror in front of a static camera, c.f. Fig. 1(a). Since there is an epipolar geometry relating the virtual views, it is possible to recover the virtual camera motion and compute the scene structure using a suitable structure-from-motion (SfM) approach [6],[7]. However, this approach, neither provides the pose of the real camera, nor determines the location of the mirror planes. This article proposes a robust closed-form method for accurately estimation of the 3D pose of the real camera using the rigid displacement between  $N \geq 3$  virtual views.

Knowing the pose of the real camera enables to determine the rigid transformation between the camera and scene objects lying outside of its FOV, which can be useful for many application scenarios. Hesch *et al.* consider a camera mounted on a robotic platform, and describe a method that uses planar mirror reflections for estimating the rigid transformation relating camera and robot reference frames [8]. They track points in hundreds of IPMRs acquired while moving the robot in front of a mirror, and use a complicated iterative ML estimator to find the transformation. Still in the context of hand-eye calibration, Mariotinni *et al.* propose several geometric methods for localizing the real camera and the mirror planes in stereo PCS [9]. However, all the methods require the real camera and the virtual views to simultaneously observe a part of the scene, which somehow undermines the usefulness of the approach.

Closely related to our work is the article by Sturm *et al.* [10], that determines the position of an object lying outside the camera FOV. They observe that the planar motion between pairs of virtual views [5] can be described as a *fixed-axis rotation*, with the rotation axis being the intersection of the two mirror planes. Thus, they propose a constructive approach that, for a certain virtual view, estimates the fixed-axes of rotation in a closed form manner. It determines the mirror position by fitting a plane to the estimated 3D lines, and finally it recovers the real camera pose by reflecting the virtual view with respect to the mirror. The article proves for the first time that the position of mirrors and camera can be uniquely determined from a minimum of  $N = 3$  virtual views and, in addition, it discusses singular configurations. The main difference to our work is that we recover the mirror position in a single estimation step by solving a new system of linear equations. The experimental results show that our formulation significantly improves accuracy and increases robustness.

Another related work is the one of Kumar *et al.* [11], concerning the calibration of camera networks with non-overlapping FOV. The extrinsic calibration of non-overlapping nodes is achieved by registering them with respect to a reference object that is observed through planar mirror reflections. The poses of the real cameras are obtained by solving a system of linear equations derived from orthogonality relations between axes of different reference frames. The linear estimator proposed by Kumar *et al.* requires a minimum of  $N = 5$  views and provides a closed form solution that is sub-optimal. This solution is used as an initial estimate for a subsequent refinement step using bundle adjustment. We present tests, with both synthetic and real data, that show that the estimations obtained with this approach are substantially less accurate and robust than the ones achieved with our algorithm.

**Notation:** Matrices are represented in sans serif font, e.g.  $M$ , vectors in bold, e.g.  $\mathbf{Q}$ ,  $\mathbf{q}$ , and scalars in italic, e.g.  $d$ . We do not distinguish between a linear transformation



**Fig. 1.** The geometry of IPMR: (a) The object  $B$  is observed by camera  $C_r$  through  $N$  planar mirror reflections, with each mirror plane  $\Pi_i$  giving rise to a virtual camera  $\hat{C}_i$ . The  $N - 1$  rigid transformations  $T_i$  (and possibly the pose of the object  $T_B$ ) are known 'a priori'. We recover the position of mirror  $\Pi_0$  and localize the real camera  $C_r$  by reflecting  $\hat{C}_0$ . (b) The 3D point  $Q$  is seen through a planar mirror reflection. The line of back-projection  $L$  always goes through point  $\hat{Q}$ , that is symmetric to  $Q$  with respect to the mirror plane  $\Pi$ . (c) The virtual camera  $\hat{C}$  and the real camera  $C$  are symmetric to each other with respect to the mirror  $\Pi$ . Remark that any symmetry transformation causes a change in the handedness of the coordinate systems.

and the matrix representing it. If nothing is said 3D points are represented in non-homogeneous coordinates, and the vector product is carried using the skew symmetric matrix, e.g.  $\mathbf{t} \times \mathbf{n} = [\mathbf{t}]_{\times} \mathbf{n}$ . The symbol  $\hat{\cdot}$  signals virtual entities induced by planar mirror reflections, e.g.  $\hat{Q}$ , and vectors topped with an arrow are versors with unitary norm, e.g.  $\vec{\mathbf{n}}$ .

## 2 Projection Model for Images of Planar Mirror Reflections

This section shows how to model the projection of a camera seeing a scene through a planar mirror reflection. We re-formulate some background concepts that have been introduced in the past [5],[9],[11], [10].

### 2.1 Projection of a 3D Point

Consider the scheme of Fig. 1(b) showing a camera with projection center  $C$  and a 3D planar mirror  $\Pi$ . Both entities are expressed with respect to a world reference frame with origin in  $O$ . The mapping of world coordinates into camera coordinates is carried by a  $4 \times 4$  matrix  $T$  in the special euclidean group  $se(3)$  [12].

$$T = \begin{pmatrix} R & \mathbf{t} \\ 0 & 1 \end{pmatrix}$$

with  $R$  being a rotation matrix, and  $\mathbf{t}$  a  $3 \times 1$  translation vector. Plane  $\Pi$  is uniquely defined by its normal, represented by the unitary vector  $\vec{\mathbf{n}}$ , and the scalar euclidean distance  $d$ . A generic point  $\mathbf{X}$  lies on the plane  $\Pi$  iff it satisfies the following equation

$$\vec{\mathbf{n}}^T \mathbf{X} = d$$

Let  $\mathbf{Q}$  be the coordinates of a 3D point that is observed through a planar mirror reflection. The reflection follows Snell’s law, and  $\mathbf{Q}$  is projected into the image point  $\mathbf{q}$ . It is easy to see that the back-projection line  $\mathbf{L}$  always goes through point  $\hat{\mathbf{Q}}$ , that is the symmetric of  $\mathbf{Q}$  with respect to  $\mathbf{\Pi}$ . Since every point on  $\mathbf{L}$  is imaged into  $\mathbf{q}$ , we can use the symmetry to go around the Snell’s reflection. Thus, the projection of  $\mathbf{Q}$  into  $\mathbf{q}$  can be expressed in homogeneous coordinates by

$$\mathbf{q} \sim \mathbf{K} \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} \mathbf{T} \begin{pmatrix} \hat{\mathbf{Q}} \\ 1 \end{pmatrix}, \tag{1}$$

with  $\mathbf{I}$  being the  $3 \times 3$  identity matrix, and  $\mathbf{K}$  the matrix of intrinsic parameters [6].

From Fig. 1(b) it follows that the world coordinates of the symmetric point  $\hat{\mathbf{Q}}$  are

$$\hat{\mathbf{Q}} = \mathbf{Q} + 2(d - \mathbf{\bar{n}}^T \mathbf{Q})\mathbf{\bar{n}}$$

The equation can be re-written in the following matrix form:

$$\begin{pmatrix} \hat{\mathbf{Q}} \\ 1 \end{pmatrix} = \mathbf{S} \begin{pmatrix} \mathbf{Q} \\ 1 \end{pmatrix} \tag{2}$$

with  $\mathbf{S}$  being a *symmetry transformation* induced by  $\mathbf{\Pi}$

$$\mathbf{S} = \begin{pmatrix} \mathbf{I} - 2\mathbf{\bar{n}}\mathbf{\bar{n}}^T & 2d\mathbf{\bar{n}} \\ 0 & 1 \end{pmatrix} \tag{3}$$

### 2.2 Symmetry Matrices

Let’s denote by  $ss(3)$  the set of all  $4 \times 4$  symmetry matrices  $\mathbf{S}$  whose structure is given in equation 3. The top-left  $3 \times 3$  sub-matrix of any element in  $ss(3)$  is always a Householder matrix [13]. Taking into account the properties of the Householder matrices, it is straightforward to prove that the following holds:

- (i) The symmetry transformations are involutory:  $\mathbf{S}^{-1} = \mathbf{S}, \forall \mathbf{S} \in ss(3)$
- (ii) The product of two symmetry transformations is a rigid transformation:  $\mathbf{S}_1 \mathbf{S}_2 \in se(3), \forall \mathbf{S}_1 \mathbf{S}_2 \in ss(3)$  ( $ss(3)$  is not an algebraic group)
- (iii) In general the product of a symmetry transformation  $\mathbf{S}$  by a rigid transformation  $\mathbf{T}$  is not an element of  $ss(3)$

### 2.3 The Virtual Camera

Consider the projection equation 1. Replacing  $\hat{\mathbf{Q}}$  by the result of equation 2 yields:

$$\mathbf{q} \sim \mathbf{K} \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} \mathbf{T} \mathbf{S} \begin{pmatrix} \mathbf{Q} \\ 1 \end{pmatrix} \tag{4}$$

We can think on  $\mathbf{T} \mathbf{S}$  as being a transformation that maps world coordinates into coordinates in a certain reference frame  $\hat{\mathbf{C}}$ . It follows from equation 4, that the IPMR can be

modeled as the image that would be directly acquired by a *virtual camera* placed in  $\widehat{\mathbf{C}}$  and having intrinsic parameters  $\mathbf{K}$  [5]. It is also easy to realize that this virtual camera  $\widehat{\mathbf{C}}$  and the *real camera*  $\mathbf{C}$  are symmetric to each other with respect to the mirror plane  $\mathbf{II}$ , c. f. Fig 1(c). The transformation from the real camera  $\mathbf{C}$  to the virtual camera  $\widehat{\mathbf{C}}$  is carried by the symmetry matrix  $S'$ :

$$S' = \mathbf{TST}^{-1} \quad (5)$$

Since  $S'$  is involutory, it also maps the virtual camera into the real camera. Moreover, and as pointed out by Kumar et al. [11], if the reference frame  $\mathbf{C}$  is right-handed, then the reference frame  $\widehat{\mathbf{C}}$  is left-handed, and vice-versa.

### 3 Assumptions and Problem Formulation

Without loss of generality, let's consider a situation where an object, lying outside the FOV of a static camera, is observed through planar mirror reflections obtained by freely moving the mirror in front of the camera<sup>1</sup>. As discussed in section 2.3, for each position of the mirror plane  $\mathbf{II}_i$ , with  $i = 0, 1, \dots, N - 1$ , there is a virtual camera  $\widehat{\mathbf{C}}_i$  that models the projection, c. f. Fig. 1(a). For convenience, it will be assumed that the world coordinate system is coincident with the reference frame of the virtual camera  $\widehat{\mathbf{C}}_0$ , and that geometric entities are expressed in world coordinates by default.

The rigid motions  $\mathbf{T}_i$  of the virtual cameras  $\widehat{\mathbf{C}}_i$ , as well as the pose of the object  $\mathbf{T}_B$ , are assumed to be known 'a priori'. They can be computed from image data using a suitable SfM approach [5][6]. The choice of the most suitable method for the problem at hands, as well as eventual adaptations to take into account the constraints of planar camera motion [5], are beyond the scope of the work. Our objective is to estimate the pose of the real camera  $\mathbf{C}_r$ , using the rigid displacements  $\mathbf{T}_i$ . Remark that, if the position of the real camera is known, then the mirror plane  $\mathbf{II}_i$  can be easily determined by finding the orthogonal plane that bisects the line that joins the centers of  $\mathbf{C}_r$  and  $\widehat{\mathbf{C}}_r$ . On the other hand, if the position of the mirror plane  $\mathbf{II}_i$  is known, then the real camera can be determined by a symmetry transformation of  $\widehat{\mathbf{C}}_i$  (equation 3). Therefore, we can solve the formulated problem by either estimating directly the pose of the real camera, or by finding the position of one of the mirror planes. We will pursue the latter strategy and look for the position of  $\mathbf{II}_0$ .

Finally, remark in Fig. 1(c) that the system of coordinates  $\mathbf{O}$  is right-handed, while the virtual camera reference frame is left-handed. Since most implementations of popular SfM algorithms provide as output rigid transformations that preserve the handiness, it will be henceforth assumed that the scene is represented using left-handed coordinates. In this manner, the transformations between the object  $\mathbf{B}$  and the virtual camera  $\widehat{\mathbf{C}}_i$  can be carried by an element of  $ss(3)$ , at the expenses of considering a symmetry transformation between the object and the real camera, c.f. Fig 1(a).

<sup>1</sup> This is not geometrically equivalent to keeping the mirror stationary and moving the camera in an unconstrained manner. In this case the rigid displacement between virtual views is not necessarily a planar motion.

## 4 Searching for the Mirror Planes

As pointed out by Gluckman and Nayar, the rigid transformation between two virtual cameras is always a planar motion [5]. This section derives the constraints that each planar motion  $T_i$  imposes on the position of the planar mirrors  $\mathbf{II}_0$  and  $\mathbf{II}_i$ .

Consider the scheme of Fig. 1(a), with the reference frame of camera  $\widehat{\mathbf{C}}_0$  being the world coordinate system in which the different planes are represented. Let  $\widehat{\mathbf{Q}}_0$  and  $\mathbf{Q}_r$  be the coordinates of the same 3D point expressed with respect to  $\widehat{\mathbf{C}}_0$  and  $\mathbf{C}_r$  respectively. It follows that,

$$\begin{pmatrix} \mathbf{Q}_r \\ 1 \end{pmatrix} = S_0 \begin{pmatrix} \widehat{\mathbf{Q}}_0 \\ 1 \end{pmatrix}$$

with  $S_0$  being the symmetry transformation defined by plane  $\mathbf{II}_0$  (equation 3). If  $\widehat{\mathbf{Q}}_i$  represents the same 3D point in the coordinate system of  $\widehat{\mathbf{C}}_i$ , it comes from equation 5 that

$$\begin{pmatrix} \mathbf{Q}_r \\ 1 \end{pmatrix} = T_i S_i T_i^{-1} \begin{pmatrix} \widehat{\mathbf{Q}}_i \\ 1 \end{pmatrix}$$

By equaling the two equations above, replacing  $\widehat{\mathbf{Q}}_i$  by  $T_i \widehat{\mathbf{Q}}_0$ , and considering the properties of the symmetry matrices, we conclude that

$$T_i = S_0 S_i$$

Considering explicitly the rotation  $R_i$ , the translation  $\mathbf{t}_i$ , and expressing the symmetries in terms of the plane parameters, it yields

$$\begin{pmatrix} R_i \mathbf{t}_i \\ 0 \ 1 \end{pmatrix} = \begin{pmatrix} 1 - 2\vec{\mathbf{n}}_0 \vec{\mathbf{n}}_0^T & 2d_0 \vec{\mathbf{n}}_0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 - 2\vec{\mathbf{n}}_i \vec{\mathbf{n}}_i^T & 2d_i \vec{\mathbf{n}}_i \\ 0 & 1 \end{pmatrix} \quad (6)$$

### 4.1 Geometric Properties

Taking into account the result of equation 6, we can state the following property that relates the rotation  $R_i$  between virtual cameras, to the normals  $\vec{\mathbf{n}}_0$  and  $\vec{\mathbf{n}}_i$  of the corresponding mirror planes.

**Statement 1:** Let  $\vec{\omega}_i$  and  $\theta_i$  denote, respectively, the direction of the rotation axis and the rotation angle of  $R_i$ . If  $\alpha_i$  is the angle between the normals to the mirror planes such that  $\vec{\mathbf{n}}_0^T \vec{\mathbf{n}}_i = \cos(\alpha_i)$ , then the following equalities hold:

$$\vec{\omega}_i = \frac{\vec{\mathbf{n}}_i \times \vec{\mathbf{n}}_0}{\sin(\alpha_i)} \wedge \theta_i = 2\alpha_i$$

*Proof:* Let  $R'_i$  be a rotation by an angle of  $2\alpha_i$  around an axis with unit direction  $(\vec{\mathbf{n}}_i \times \vec{\mathbf{n}}_0) \sin(\alpha_i)^{-1}$ . By applying Rodrigues' formula [12], it follows that:

$$R'_i = I + \frac{\sin(2\alpha_i)}{\sin(\alpha_i)} [\vec{\mathbf{n}}_i \times \vec{\mathbf{n}}_0]_{\times} + \frac{(1 - \cos(2\alpha_i))}{\sin(\alpha_i)^2} [\vec{\mathbf{n}}_i \times \vec{\mathbf{n}}_0]_{\times}^2$$

We conclude after some algebraic manipulation that:

$$[\vec{n}_i \times \vec{n}_0]_{\times} = \vec{n}_0 \vec{n}_i^T - \vec{n}_i \vec{n}_0^T$$

$$[\vec{n}_i \times \vec{n}_0]_{\times}^2 = \cos(\alpha_i) (\vec{n}_0 \vec{n}_i^T + \vec{n}_i \vec{n}_0^T) - \vec{n}_0 \vec{n}_0^T - \vec{n}_i \vec{n}_i^T$$

Replacing the above results in the expression of  $R'_i$ , and performing some simplifications, it yields that

$$R'_i = I + 4 \cos(\alpha_i) (\vec{n}_0 \vec{n}_i^T) - 2 (\vec{n}_0 \vec{n}_i^T + \vec{n}_i \vec{n}_0^T)$$

Finally, taking into account that  $\cos(\alpha_i) = \vec{n}_0^T \vec{n}_i$ , it follows that

$$R'_i = (I - 2\vec{n}_0 \vec{n}_0^T) (I - 2\vec{n}_i \vec{n}_i^T)$$

Thus,  $R'_i$  and  $R_i$  are the same rotation matrix (see equation 6), and the correctness of the statement has been proved.  $\square$

The next result was originally stated in [5], and it is presented in here for the sake of completeness

**Statement 2:** The rigid transformation  $T_i$  is a planar motion, with the translation component  $\mathbf{t}_i$  being orthogonal to the rotation axis  $\vec{\omega}_i$ .

*Proof:* From equation 6, and taking into account that  $\vec{n}_i^T \vec{n}_0 = \cos(\frac{\theta_i}{2})$ , we can write the translation as

$$\mathbf{t}_i = 2(d_0 - 2d_i \cos(\frac{\theta_i}{2})) \vec{n}_0 + 2d_i \vec{n}_i \tag{7}$$

Since  $\vec{\omega}_i$  is orthogonal to both  $\vec{n}_0$  and  $\vec{n}_i$  (property 1) it is obvious that  $\mathbf{t}_i^T \vec{\omega}_i = 0$ , which proves the orthogonality statement.  $\square$

## 4.2 Linear Constraints

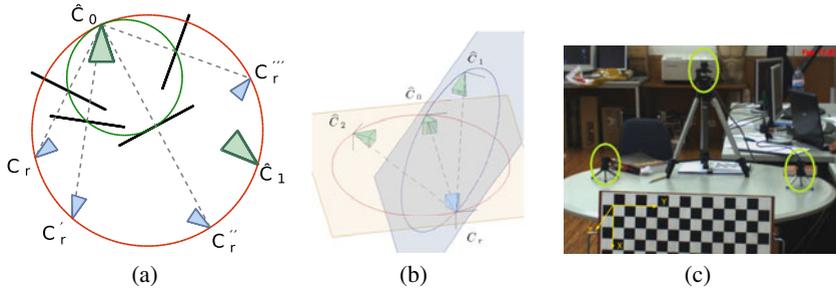
As stated in section 3, it is sufficient to estimate the position of one mirror plane for solving for the pose of the real camera and position of remaining planes. It is shown in here that each planar motion  $T_i$  gives rise to a pair of independent linear constraints on the position of the plane  $\Pi_0$ , c.f. Fig. 1(a). Henceforth, the rigid transformation is expressed in terms of  $\vec{\omega}_i$ ,  $\theta_i$  and  $\mathbf{t}_i$ , while the planes are parametrized by their normals  $\vec{n}$  and scalar distances  $d$ .

Consider the transpose of equation 7 with both sides being multiplied by  $\vec{n}_0$ . It arises after some algebraic manipulation that

$$\mathbf{t}_i^T \vec{n}_0 - 2d_0 + 2 \cos(\frac{\theta_i}{2}) d_i = 0 \tag{8}$$

A different equation can be derived by considering the cross product between  $\mathbf{t}_i$  and  $\vec{n}_0$ . From equation 7 and property 1, we obtain that

$$[\mathbf{t}_i]_{\times} \vec{n}_0 - 2 \sin(\frac{\theta_i}{2}) \vec{\omega}_i d_i = 0 \tag{9}$$



**Fig. 2.** (a) The space of possible solutions.  $\widehat{C}_0$  and  $\widehat{C}_1$  are two virtual cameras. The figure shows the planes  $\Pi_0$  satisfying equation 10, as well as the corresponding solutions for the real camera  $C_r$ . The red circle is the locus of camera centers, while the green circle is the locus of points where  $\Pi_0$  bisects  $C_r\widehat{C}_0$ . It can be verified that there are an infinite number of solutions for the pose of the real camera that are consistent with the two virtual cameras. (b) Determining the real camera  $C_r$  for the case of  $N = 3$ . Consider the two pairs of virtual cameras  $\widehat{C}_0, \widehat{C}_1$  and  $\widehat{C}_0, \widehat{C}_2$ . Each pair has a space of possible solutions for the real camera. The two loci intersect in two points: the common virtual camera  $\widehat{C}_0$ , and the correct solution for the real camera  $C_r$ . (c) Camera cluster setup used to evaluate the algorithms performance using IPMRs.

Equations 8 and 9 provide 4 linear constraints on the unknown plane parameters  $\vec{n}_0, d_0$  and  $d_i$ . However, and since  $[\mathbf{t}_i]_\times$  is a singular matrix, only 3 constraints are independent. It is interesting to see that equation 9 implicitly enforces orthogonality between the rotation axis  $\vec{\omega}_i$  and vectors  $\vec{n}_0$  (property 1) and  $\mathbf{t}_i$  (property 2).

By solving equation 8 in order to  $d_i$ , and replacing the result in equation 9, it arises

$$([\mathbf{t}_i]_\times + \tan(\frac{\theta_i}{2}) \vec{\omega}_i \mathbf{t}_i^\top) \vec{n}_0 - 2 \tan(\frac{\theta_i}{2}) \vec{\omega}_i d_0 = 0 \tag{10}$$

Equation 10 provides 2 independent linear constraints on the parameters of plane  $\Pi_0$ . Since the camera pose and mirror planes can be uniquely recovered by finding the position of one of those planes, the constrains of equation 10 constitute a minimal formulation of the problem in terms of the number of unknowns. However, these two independent constraints are insufficient to determine the 3 DOF of the mirror plane. As pointed out by Sturm and Bonfort [10] the problem is under-determined when using just a pair of virtual views. In this case the real camera can be at any location in a circular locus on the plane orthogonal to the axis of the relative rotation. This is illustrated in Fig. 2(a).

### 5 Determining the Real Camera from $N \geq 3$ Virtual Cameras

Three virtual cameras,  $\widehat{C}_0, \widehat{C}_1$  and  $\widehat{C}_2$ , define two independent planar motions  $T_1$  and  $T_2$ . Each motion gives rise to two independent linear constraints on the parameters  $\vec{n}_0$  and  $d_0$  (equation 10). Since we have a total of 4 equations and just 3 unknowns, then it is possible to uniquely determine the plane  $\Pi_0$  and estimate the pose of the real camera. Fig. 2(b) provides a geometric insight about localizing the real camera using  $N = 3$  virtual cameras.

### 5.1 The System of Linear Equations

Consider  $N$  virtual cameras defining  $N - 1$  independent planar motions, c.f. Fig. 1(a). From section 4.2 follows that each motion  $T_i$  gives rise to a set of linear constraints on the parameters of the mirror plane  $\mathbf{II}_0$ . Stacking the constraints of the  $N - 1$  motions leads to a system of linear equations that can be solved using a DLT estimation approach. We consider two alternative formulations for the system of equations: *Method 1* that is based on the result of equation 10, which becomes singular whenever  $\theta_i = \pi$ ; and *Method 2* that uses the constraints derived in equations 8-9.

**Method 1.** The system of equations can be written as

$$\underbrace{\begin{pmatrix} A_1 \\ \vdots \\ A_{N-1} \end{pmatrix}}_A \begin{pmatrix} \vec{\mathbf{n}}_0 \\ d_0 \end{pmatrix} = 0 \tag{11}$$

where each planar motion  $T_i$  defines a sub-matrix  $A_i$  with dimension  $3 \times 4$  (equation 10)

$$A_i = ([\mathbf{t}_i]_{\times} + \tan(\frac{\theta_i}{2})\vec{\omega}_i\mathbf{t}_i^T - 2 \tan(\frac{\theta_i}{2})\vec{\omega}_i)$$

**Method 2.** This second method uses the linear constraints derived in equations 8-9. Remark that the formulation is non-minimal in the sense that involves the scalar parameters  $d_i$ . The system of equations can be written in the following form

$$\underbrace{\begin{pmatrix} B_1 & \mathbf{b}_1 & 0 & \dots & 0 \\ B_2 & 0 & \mathbf{b}_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B_{N-1} & 0 & 0 & \dots & \mathbf{b}_{N-1} \end{pmatrix}}_B \begin{pmatrix} \vec{\mathbf{n}}_0 \\ d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{N-1} \end{pmatrix} = 0 \tag{12}$$

with  $B_i$  being  $4 \times 4$  sub-matrices

$$B_i = \begin{pmatrix} \mathbf{t}_i^T & -2 \\ [\mathbf{t}_i]_{\times} & 0 \end{pmatrix}$$

and  $\mathbf{b}_i$  vectors with dimension 4

$$\mathbf{b}_i = \begin{pmatrix} 2 \cos(\frac{\theta_i}{2}) \\ -2 \sin(\frac{\theta_i}{2})\vec{\omega}_i \end{pmatrix}$$

### 5.2 Outline of the Algorithm

1. Let  $T_i$  be the transformations mapping  $\widehat{C}_0$  into  $\widehat{C}_i$ , with  $i = 1, 2, \dots, N - 1$ . For each  $T_i$  obtain the rotation axis  $\vec{\omega}_i$ , the rotation angle  $\theta_i$  and the translation vector  $\mathbf{t}_i$

2. Build the  $3(N - 1) \times 4$  matrix A of equation 11 (for *method 2* build the  $4(N - 1) \times (N + 3)$  matrix B of equation 12)
3. Apply SVD to find a least square solution for the system of equations (the vector result is denoted by  $\mathbf{x}$ ).
4. Compute plane  $\Pi_0$  by making  $\vec{\mathbf{n}}_0 = \frac{\mathbf{x}_{1\dots 3}}{\|\mathbf{x}_{1\dots 3}\|}$  and  $d_0 = \frac{x_4}{\|\mathbf{x}_{1\dots 3}\|}$
5. Determine the symmetry matrix  $S_0$  using  $\vec{\mathbf{n}}_0$  and  $d_0$  (equation 3). This matrix maps  $\widehat{C}_0$  into  $C_r$ , enabling the localization of the real camera, c.f. Fig. 1(a).
6. Compute the position of remaining mirror planes  $\Pi_i$ , by finding the plane orthogonal to the line  $C_r \bar{C}_i$  that bisects it.

### 5.3 Singular Configurations

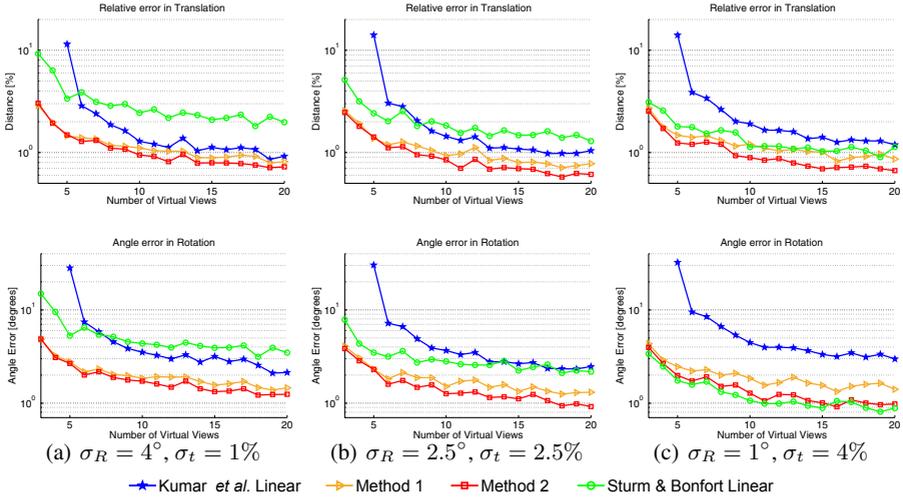
It has already been referred that the linear equations of method 1 present an algebraic singularity for the case of  $\theta_i = \pi$ . In addition, the null spaces of matrices A (*method 1*) and B (*method 2*) become multi-dimensional whenever all the mirror planes intersect into a single line in 3D. This is a singular configuration for which there are multiple solutions for the real camera pose. It can occur by either rotating the mirror around a fixed-axis, or by translating the mirror in a manner that all the reflection planes are parallel between them (the intersection line is at infinity). This degeneracy has been originally observed by Sturm and Bonfort [10].

## 6 Performance Evaluation with Synthetic Data

This section evaluates the accuracy of the algorithm outlined in section 5.2. The two linear formulations (*method 1* and *method 2*) are compared against the closed-form linear estimation approaches proposed by Sturm [10] and Kumar *et al.* [11]. For the latter we used the code implementation made publicly available by the authors, while for the former we re-implemented the method following the steps in the paper.

In order to have reliable ground truth we defined a simulation environment for generating synthetic data. We consider a working volume with a static camera. A set of  $N$  planes, simulating the mirrors, are generated assuming a convenient random uniform distribution. Each plane gives rise to a virtual camera using the symmetry transformation of equation 3. The positions of the virtual cameras are disturbed both in translation and rotation. The direction of the translation and rotation axis are drawn from a random uniform distribution, while the amount of disturbance is given by a zero mean normal distribution with variable standard deviation (the noise power). For convenience the errors in translation are relative errors defined as a percentage of the modulus of the true translation vector. The disturbed pose of the virtual cameras are the input to the different methods. The translation and rotation angle of the relative rigid motion between the estimate and ground truth are used as error metrics. We performed tests for different noise conditions. For each test the simulation was ran 100 times and the root mean square error was computed. The results are exhibited in Fig. 3.

According to the synthetic experiments the two linear methods presented in this paper outperform the linear approach proposed by Kumar *et al.* [11]. This is not surprising

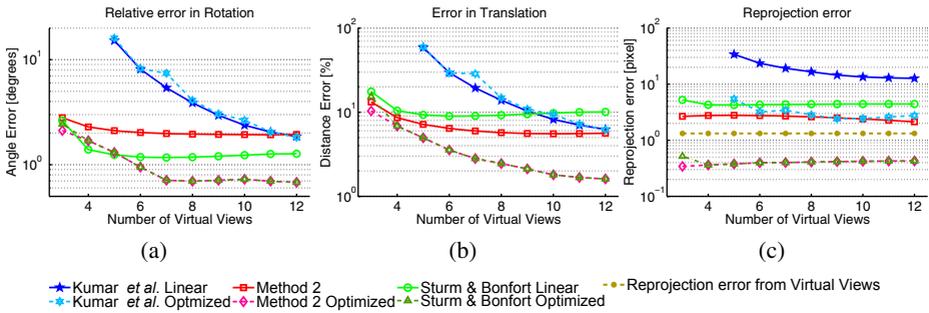


**Fig. 3.** Synthetic experiment for comparing the performance in estimating the real camera pose for an increasing number  $N$  of images of planar mirror reflections. The top row shows the RMS of the relative error in translation, while the bottom row concerns the angular RMS error in rotation. Each column concerns results assuming different standard deviations  $\sigma_R$  and  $\sigma_t$  for the additive noise in rotation and translation.

because the latter provides a sub-optimal solution in the sense that it estimates the rigid displacement between camera and object without taking into account the rotation constraints. The approach by Sturm and Bonfort is in general less accurate than both *method 1* and *method 2*, and seems to be highly sensitive to noise in the rotation between virtual views. However, it presents better results in the estimation of the real camera rotation for the case of  $\sigma_R = 1^\circ$ . This behavior can be explained by the fact that their algorithm fits the mirror to the estimated fixed rotation axes by first determining the normal orientation of the plane, and then its distance to the origin. Under negligible noise in rotation, the direction of the axes is accurately determined, and the normal to the mirror is correctly computed. Since the real camera rotation only depends on the mirror orientation, the final estimation error is very small. When the noise in rotation increases, the estimation of the direction of the fixed rotation axes becomes inaccurate, and the final results suffer a quick degradation.

Our two methods carry the estimation of the mirror position by simultaneously solving for the orientation and distance to the origin. This leads to a better trade-off in terms of sensitivity to noise. It also worth mentioning that during the simulations there were situations for which both Sturm’s and Kumar’s algorithm diverged in the sense that the output estimate was too off from the ground truth. In the case of Sturm’s approach this happened occasionally for  $N = 3$ , while for Kumar’s approach this happened quite often for  $N \leq 6$ . The situations of divergence were discarded and the random generation of data was repeated to produce the results of Fig. 3.

Somehow surprising is the fact that *method 2* systematically over-performs the minimal formulation of *method 1*. From synthetic results and experiences with real data,



**Fig. 4.** Extrinsic calibration of a camera cluster with 3 nodes (Fig. 2(c)). The graphics compare the errors in estimating the relative displacement between camera nodes using different approaches. Since the cameras have overlapping FOV's, we determined an approximate ground truth by using a conventional extrinsic calibration approach [6].

*method 2* remained the most accurate approach. In order to not overload the next graphs only the results using *Method 2* will be shown.

## 7 Experiments in Extrinsic Calibration

In this experiment we use IPMRs to perform the extrinsic calibration of a small cluster with 3 camera nodes. Fig. 2(c) shows the setup where a calibration grid is placed outside of the FOV of the cameras. Several images of the grid are acquired with the help of a mirror, and the corresponding virtual cameras are localized in the grid reference frame by homography factorization [7]. This is performed independently for each camera node. The virtual views are used as input for the different methods that return a pose estimate for the real camera. Since we are using a common object, the 3 cameras are registered in the same reference frame which enables extrinsic calibration. The results are subsequently refined by optimizing the re-projection error with respect to the mirror positions and real camera pose.

Fig. 4 shows the difference between the reference ground truth and the extrinsic calibration from mirror reflections. The graphics compare the performance of the different methods for an increasing number of virtual views  $N = 3, \dots, 12$ . Since we collected a total of 12 images per node, we provide the RMS error taking into account the estimation results for all possible  $N$  image combinations. This real experiment confirms the conclusions reached in synthetic environment: Kumar *et al.* algorithm is in general inferior to both our method and Sturm's approach. It needs a larger number of views to provide accurate estimations, and it diverges for  $N \leq 8$  views. For  $N < 4$  views Sturm's approach also diverges in the cases that the mirrors are close to a singular parallel configuration. Table 1 shows the number of diverging cases for all methods. These situations were not considered in the statistical treatment of the results shown in Fig. 4. It is also important to refer that, since the determination of motion from planar homographies typically presents a good accuracy in rotation, this is an experiment that favors Sturm's approach, explaining the excellent performance in estimating the camera rotation.

**Table 1.** The table summarizes the number of situations for which each method fails in providing a suitable initial estimate. Our approach shows an impressive robustness even when the number of virtual views is small and/or are close to a singular parallel configuration.

Number of views	3	4	5	6	7	8	9	10	11	12
Total number of combinations	220	495	792	924	792	495	220	66	12	1
<i>Method 2</i>	0	0	0	0	0	0	0	0	0	0
Sturm and Bonfort Linear	8	2	0	0	0	0	0	0	0	0
Kumar <i>et al.</i> Linear			756	441	146	19	0	0	0	0

Fig. 4 also shows the optimized results from each method. It can be observed that Kumar's accuracy is not brilliant for  $N \leq 8$  because of the poor closed-form initialization. Also Sturm's initialization for  $N=3$  does not provide good convergence results because of the inaccuracy in the translation estimate. For  $N \geq 3$  both Sturm's and our method converge towards the same solution.

## 8 Conclusions

This article provides a geometric insight about IPMRs and shows that the position of the mirror planes can be recovered in a straightforward manner by solving a system of linear equations. This leads to a new closed-form estimation method that is able to recover the pose of the real camera from a minimum of  $N = 3$  views. Extensive experimental results clearly show the superiority of our approach with respect to the state-of-the-art. Our approach proved to have a well balanced sensitivity to noise in translation and rotation, a good behavior under situations of quasi-singularity, and an excellent accuracy even when the number of images is minimum.

## Acknowledgments

The authors are grateful to the Portuguese Science Foundation by generous funding through grant PTDC/EEA-ACR/72226/2006. João P. Barreto also acknowledges the support through grant PTDC/EEA-ACR/68887/2006.

## References

1. Baker, S., Nayar, S.K.: A theory of single-viewpoint catadioptric image formation. *Int. Journal of Computer Vision* 35(2), 175–196 (1999)
2. Goshtasby, A., Gruver, W.: Design of a single-lens stereo camera system. *Pattern Recognition* (1993)
3. Inaba, M., Hara, T., Inoue, H.: A stereo viewer based on a single camera with view-control mechanisms. In: *Intelligent Robots and Systems* (1993)
4. Ramsgaard, B., Balslev, I., Arnspar, J.: Mirror-based trinocular systems in robot-vision. *Pattern Recognition* 4, 499–502 (2000)
5. Gluckman, J., Nayar, S.: Catadioptric stereo using planar mirrors. *IJCV* (2001)
6. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York (2003)

7. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.S.: *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer, Heidelberg (2003)
8. Hesch, J., Mourikis, A., Roumeliotis, S.: Determining the camera to robot-body transformation from planar mirror reflections. In: *IROS* (2008)
9. Mariottini, G., Scheggi, S., Morbidi, F.: Planar catadioptric stereo: Single and multi-view geometry for calibration and localization. In: *ICRA* (2009)
10. Sturm, P., Bonfort, T.: How to compute the pose of an object without a direct view? In: *ACCV*, vol. II, pp. 21–31 (2006)
11. Kumar, R., Ilie, A., Frahm, J.M., Pollefeys, M.: Simple calibration of non-overlapping cameras with a mirror. In: *CVPR*, pp. 1–7 (2008)
12. Murray, R.M., Sastry, S.S., Zexiang, L.: *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc, Boca Raton (1994)
13. Golub, G.H., Van Loan, C.F.: *Matrix computations*, 3rd edn. Johns Hopkins Univ. Press, Baltimore (1996)