

# Plane-Based Resection for Metric Affine Cameras

Adrien Bartoli      Toby Collins

The ENCOV group (Endoscopy and Computer Vision)  
Université Clermont Auvergne and CNRS, Institut Pascal

Corresponding author: Adrien Bartoli

`Adrien.Bartoli@gmail.com`

Accepted by the *Journal of Mathematical Imaging and Vision*, January 2018

## Abstract

We study the problem of resecting the metric affine camera models from at least three non-colinear point correspondences. A direct application is plane pose estimation. We consider the three most popular metric affine cameras, namely the paraperspective, weak-perspective and orthographic cameras. For each model, we give an algebraic procedure which finds the optimal solution, where by optimal we mean the global minimizer of the reprojection error under the Euclidean norm. Our algebraic procedures cover both the minimal case of three points and the redundant cases of more than three points. They always return two solutions, as the problem has a two-way ambiguity on the rotation and translation for the three cameras in the general case. The scale of the paraperspective and weak-perspective cameras is however recovered uniquely. The orthographic case is the most involved and has not been solved analytically in the literature. We characterize its intrinsic complexity by showing that it reduces to finding the roots of an irreducible and non-solvable by radicals sextic polynomial. The previous algorithms for the paraperspective and weak-perspective cases have singularities, while, in contrast, our algebraic procedures do not.

**Code release.** We have prepared a public release version of our code, which has been released with the article.

**Acknowledgements.** This research has received funding from the EU's FP7 through the ERC research grant 307483 FLEXABLE. We thank Florian Bugarin and Didier Henrion for their help in using Glopoly.

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Background</b>	<b>7</b>
2.1	Notation . . . . .	7
2.2	Camera Models . . . . .	7
<b>3</b>	<b>Problem Statement and Formulations</b>	<b>9</b>
3.1	Initial Formulation . . . . .	9
3.2	Translationless Formulation . . . . .	9
3.3	First Canonical Formulation . . . . .	9
3.4	Second Canonical Formulation . . . . .	10
3.5	Formulation Group . . . . .	11
3.6	Algebraic Procedure . . . . .	12
<b>4</b>	<b>Resection of the Orthographic Camera</b>	<b>12</b>
4.1	The Canonical Formulations . . . . .	12
4.2	A Difficult Quartically Constrained Problem . . . . .	13
4.3	Solving via Multiple Cases . . . . .	13
4.4	Cases 2.x.x . . . . .	15
4.5	Case 1.2.2 . . . . .	16
4.5.1	Parameterization for Cases 1.x.x . . . . .	17
4.5.2	Reformulation . . . . .	17
4.5.3	Introduction of the Orthonormality Constraints . . . . .	18
4.5.4	Solution Strategy . . . . .	19
4.5.5	Expressing $\mathbf{q}$ as a Function of $\beta$ . . . . .	19
4.5.6	Deriving a Univariate Polynomial in $\beta$ . . . . .	20
4.5.7	Solving for $\beta$ . . . . .	20
4.6	Cases 0.x.x . . . . .	23
4.7	Algebraic Procedure . . . . .	24
<b>5</b>	<b>Resection of the Weak-Perspective and Paraperspective Cameras</b>	<b>24</b>
5.1	Overview, Relationship to IPPE and Optimality . . . . .	26
5.2	Computing $\mathbf{B} \in GL_2(\mathbb{R})$ . . . . .	26

---

5.3	Computing $\gamma \in \mathbb{R}^+$ and $\mathbf{R} \in \text{SO}_3$ . . . . .	27
5.3.1	Computing $\gamma \in \mathbb{R}^+$ and $\mathbf{u} \in \mathbb{R}^2$ . . . . .	27
5.3.2	Computing $\mathbf{R} \in \text{SO}_3$ . . . . .	27
5.4	Computing $\mathbf{t} \in \mathbb{R}^3$ . . . . .	28
5.5	Algebraic Procedures . . . . .	28
<b>6</b>	<b>Experimental Results</b>	<b>28</b>
6.1	Compared Methods . . . . .	29
6.2	Synthetic Data . . . . .	30
6.2.1	Comparison of the Orthographic and Weak-Perspective Cameras . . . . .	30
6.2.2	Comparison of the Orthographic Algebraic Procedure with Alternative Methods . . . . .	31
6.2.3	Performance with Perspective Projection and Uncalibrated Images . . . . .	34
6.2.4	Performance in Plane-Based Structure-from-Motion . . . . .	35
6.3	Real Data . . . . .	37
6.3.1	Pure Rotation . . . . .	37
6.3.2	General Motion . . . . .	39
6.4	Computation Time Analysis . . . . .	41
<b>7</b>	<b>Conclusion</b>	<b>43</b>
<b>A</b>	<b>Translationless Formulation</b>	<b>43</b>
<b>B</b>	<b>Analytic Expression of the Cholesky Factor of <math>\mathbf{I} + \mathbf{d}\mathbf{d}^\top</math></b>	<b>44</b>
<b>C</b>	<b>Aligning the Projection Direction to the <math>Z</math>-Axis</b>	<b>44</b>
<b>D</b>	<b>Solution of the Rank-1 Equation of Type 3, <math>z\mathbf{u}\mathbf{u}^\top + z\mathbf{G} = \mathbf{K}</math></b>	<b>45</b>
<b>E</b>	<b>Solution of the Rank-1 Equation of Type 4, <math>-z\mathbf{u}\mathbf{u}^\top + z\mathbf{G} = \mathbf{K}</math></b>	<b>46</b>
<b>F</b>	<b>Intersecting a Centred Ellipse and the Unit Circle</b>	<b>47</b>
F.1	Problem Statement . . . . .	47
F.2	Finding the Degenerate Conic as a Pair of Lines . . . . .	47
F.3	Intersecting the Pair of Lines with the Unit Circle . . . . .	48
<b>G</b>	<b>Impossibility of Cases 1.x.x Other Than 1.2.2</b>	<b>49</b>
G.1	Case 1.1.2 . . . . .	49

---

G.1.1	Specializing the Parameterization . . . . .	49
G.1.2	Reducing the Unknowns to $\alpha, \nu \in \mathbb{R}$ . . . . .	49
G.1.3	Impossibility . . . . .	50
G.2	Case 1.2.1 . . . . .	51
G.3	Case 1.1.1 . . . . .	51
G.4	Cases 1.0.x . . . . .	52
G.5	Cases 1.x.0 . . . . .	52
<b>H</b>	<b>Algebraic Procedure for the Weak-Perspective Camera</b>	<b>52</b>
<b>I</b>	<b>Derivation Details for Case 2.x.x</b>	<b>53</b>

# 1 Introduction

Camera resection is the problem of finding the camera parameters from 3D-2D correspondences of geometric primitives, typically points and lines. It has been thoroughly studied in the literature for the dominant camera models and the typical types of primitives. The most common case is the resection of the calibrated perspective camera from point correspondences, which is also called the  $PnP$  problem (Fischler and Bolles, 1981). The general problem of camera resection is important, as it forms a building block for SfM (Structure-from-Motion) and SLAM (Simultaneous Localization and Mapping) engines (Klein and Murray, 2007; Snavely et al., 2008), and SfT (Shape-from-Template) to some extent (Bartoli et al., 2015).

We here study the specific case of resecting a metric affine camera from coplanar point correspondences. A metric affine camera is one of the orthographic, weak-perspective and paraperspective models. We use the commonly adopted assumption that the data likelihood follows a Gaussian IID model. Under this assumption, the reprojection error, defined as the sum of squared Euclidean distances between the measured and the predicted image points, can be minimized to find the statistically optimal solution. We came across this problem when we studied SfM with metric affine cameras and coplanar points (Collins and Bartoli, 2017). We demonstrated that without additional prior knowledge, plane-based affine SfM is solvable only with the orthographic camera, which is thus the most important practical case. We gave a comprehensive SfM framework, where we used Gloptipoly to solve plane-based orthographic resection. Gloptipoly is a general purpose polynomial optimization tool (Henrion et al., 2009). Plane-based orthographic resection with Gloptipoly finds the cost’s global minimum. However, it is too slow for realtime applications and unstable in a very small fraction of cases. Iterative solutions were proposed that run extremely fast (Cardoso and Zietak, 2015; Steger, 2017a). However, none of these guarantees one to find the cost’s global minimum. In practice, they may fall in a local minimum for smaller numbers of point correspondences, though this happens in rare instances (Steger, 2017a).

On the other hand, the weak-perspective and paraperspective models received optimal solutions in the literature (Collins and Bartoli, 2014; Horaud et al., 1997; Oberkampf et al., 1996). However, the proposed methods have singularities. The existing weak-perspective solution is part of the POSIT framework for planar objects (Oberkampf et al., 1996). The solution requires one to solve a complex bivariate quadratic, whose solution requires one to divide by a possibly vanishing quantity  $\|\mathbf{J}_0\|_2^2 - \|\mathbf{I}_0\|_2^2$ . Our algebraic procedure is much simpler and compact to derive, and does not have singularities. The existing paraperspective solutions are twofold. The first one came out of a generalization of the POSIT framework (Horaud et al., 1997). This generalized framework may be instantiated with the weak-perspective camera, in which case it directly uses the weak-perspective solution from (Oberkampf et al., 1996). It may also be instantiated with

the paraperspective camera but using the same formalism as (Oberkampff et al., 1996). The most recent existing paraperspective solution is called IPPE and is part of a complete perspective plane-based pose estimation framework (Collins and Bartoli, 2014). IPPE requires one to align the paraperspective viewing direction to the camera’s optical axis, and fails if these are already close in space. Our algebraic procedure has similarities with IPPE, but does not have singularities. We give a proof of optimality of our algebraic procedure. Though IPPE turns out to be optimal away from the singularity, this was not proved.

Our main findings also include that, even if the weak-perspective and paraperspective models are relatively easy to solve analytically, the orthographic model involves solving a non-trivial polynomial system, as was already observed in (Collins and Bartoli, 2017; Steger, 2017a). This is because the number of unknowns in the weak-perspective and paraperspective models can be simply reduced to four, which are directly constrained by quadratic equations up to a change of variable. In the orthographic model however, the number of degrees of freedom is reduced to three, which are expressed in terms of four parameters related by a quartic constraint. These four parameters indeed form a matrix in the group of sub-Stiefel matrices in  $\mathbb{R}^{2 \times 2}$ . We show that the polynomial system can be reduced, in the most complicated case, to finding the roots of an irreducible and non-solvable by radicals univariate sextic polynomial. These roots cannot be found symbolically, but can be found numerically in a very stable and fast manner.

We give a complete algebraic procedure for each of the three camera models. We experimentally compare our algebraic procedure for the orthographic camera to the previous solution based on Gloptipoly (Collins and Bartoli, 2017). Our algebraic procedure runs in about 0.76 ms for fewer than 100 points and 1.2 ms for 1,000 points while Gloptipoly takes about 0.25 s in both cases. Our algebraic procedure has practical implications, and we briefly sketch three of them. The first practical implication is in SfM with planar structures and affine cameras. The slowest step is currently optimal camera resection (Collins and Bartoli, 2017). Using the proposed algebraic procedure would speed up the whole pipeline by an order of magnitude. The second and third practical implications are in deformable surface 3D reconstruction from monocular images. Our algebraic procedure can be used to reconstruct a surface piecewise, using local orthographic camera resection (Collins and Bartoli, 2010; Taylor et al., 2010). Because this process must be run hundreds or thousands of times per image, the proposed algebraic procedure would considerably improve computation speed. Our algebraic procedure could also be used to rapidly detect where tearing occurs on a deformable surface from a monocular image. This is because at a tearing region, a local camera resection will have an unusually high reprojection error (Taylor et al., 2010).

## 2 Background

This section gives our notation and the three metric affine camera models.

### 2.1 Notation

We use italics for scalar ( $a, \lambda$ ), bold for vectors ( $\mathbf{a}, \boldsymbol{\lambda}$ ) and typewriter fonts for matrices ( $\mathbf{M}$ ). For dimension  $d \in \{2, 3\}$ , we write  $GL_d(\mathbb{R})$  the general linear group over  $\mathbb{R}$ , containing the invertible ( $d \times d$ ) matrices,  $\mathbb{O}_d \subset GL_d(\mathbb{R})$  the group of orthonormal matrices and  $\mathbb{SO}_d \subset \mathbb{O}_d$  the group of orthonormal matrices with positive determinant. We write  $\mathbb{SS}_{23}$  and  $\mathbb{SS}_{22}$  the Stiefel and sub-Stiefel manifolds of  $\mathbb{O}_3$ . We write  $\mathbb{S}$  the set of symmetric positive definite matrices. We write the vectors of the standard basis as  $\mathbf{e}_1, \dots, \mathbf{e}_d$ , where  $\mathbf{e}_j$  is an all zero vector with a one at the  $j$ -th position. We write the vector and matrix two-norm as  $\|\mathbf{a}\|_2$  and  $\|\mathbf{M}\|_{\mathcal{F}}$  respectively. We write the elementwise, Hadamard product as  $\odot$ . We write vertical stacking for scalars/vectors/matrices as  $\text{stk}(\mathbf{M}, \mathbf{N}) \stackrel{\text{def}}{=} [\mathbf{M}^\top \ \mathbf{N}^\top]^\top$ . We define  $\lambda_i(\mathbf{M})$  and  $\epsilon_i(\mathbf{M})$  for the eigenvalues and eigenvectors of  $\mathbf{M} \in \mathbb{R}^{d \times d}$ . We write  $\mu_{\pm}(\mathbf{M})$  for the two singular values of  $\mathbf{M} \in \mathbb{R}^{2 \times 2}$ , with  $\mu_+(\mathbf{M}) \geq \mu_-(\mathbf{M}) \geq 0$ . We define the operator which factorizes a rank-1 matrix as  $\text{rank}_1(\mathbf{u}\mathbf{u}^\top) = \pm\mathbf{u}$ , for  $\mathbf{u} \in \mathbb{R}^d$ .

We define  $\mathbf{S} \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \in \mathbb{SO}_2$  and  $\Pi_{\mathbf{d}} \stackrel{\text{def}}{=} [\mathbf{I} \ \mathbf{d}]$ , where  $\mathbf{I}$  is the identity matrix of the right size. We write  $\mathbf{X} \in \mathbb{R}^{3 \times m}$  and  $\mathbf{Y} \in \mathbb{R}^{2 \times m}$  for the  $m$  point correspondences. More specifically, each column of  $\mathbf{X}$  holds the coordinates of a model point, with the image coordinates of the corresponding observed point held in the corresponding column in  $\mathbf{Y}$ . The points in  $\mathbf{X}$  are coplanar. The points in  $\mathbf{X}$  and  $\mathbf{Y}$  are not colinear.

### 2.2 Camera Models

An affine camera preserves parallelism (Faugeras et al., 2001; Hartley and Zisserman, 2003). The most general affine camera is represented by a matrix  $\mathbf{P} \in \mathbb{R}^{2 \times 3}$  for the rotational part and a vector  $\mathbf{t} \in \mathbb{R}^2$  for the translational part. It projects a 3D point  $\mathbf{Q} \in \mathbb{R}^3$  to an image point  $\mathbf{q} \in \mathbb{R}^2$  as  $\mathbf{q} = \mathbf{P}\mathbf{Q} + \mathbf{t}$ . A metric affine camera has its rotational part in a specific subspace. We consider the three most common instances of metric affine cameras. For the orthographic camera we have  $\mathbf{P} = \bar{\mathbf{R}}$ , with  $\bar{\mathbf{R}} \in \mathbb{SS}_{23}$ . For the weak-perspective camera we have  $\mathbf{P} = \gamma\bar{\mathbf{R}}$ , with  $\gamma \in \mathbb{R}^+$  and  $\bar{\mathbf{R}} \in \mathbb{SS}_{23}$ . We recall that  $\Pi_{\mathbf{d}} \stackrel{\text{def}}{=} [\mathbf{I} \ \mathbf{d}]$ . For the paraperspective camera we have  $\mathbf{P} = \gamma\Pi_{\mathbf{d}}\mathbf{R}$ , with  $\gamma \in \mathbb{R}^+$ ,  $\mathbf{d} \in \mathbb{R}^2$  and  $\mathbf{R} \in \mathbb{SO}_3$ , and with  $\det(\mathbf{P}\Pi_{\mathbf{0}}^\top) \neq 0$ . While the orthographic and weak-perspective cameras project along the camera's principal ray, the paraperspective camera projects along the direction represented by  $\mathbf{d}$ . We define the set of admissible matrices for these three metric affine

cameras' rotational parts as:

$$\begin{aligned} \mathbb{M}_{\text{OR}} &\stackrel{\text{def}}{=} \mathbb{SS}_{23} \\ \mathbb{M}_{\text{WP}} &\stackrel{\text{def}}{=} \{ \mathbf{P} \in \mathbb{R}^{2 \times 3} \mid \exists \gamma \in \mathbb{R}^+, \bar{\mathbf{R}} \in \mathbb{SS}_{23} \text{ s.t. } \mathbf{P} = \gamma \bar{\mathbf{R}} \} \\ \mathbb{M}_{\text{PP}} &\stackrel{\text{def}}{=} \{ \mathbf{P} \in \mathbb{R}^{2 \times 3} \mid \exists \gamma \in \mathbb{R}^+, \mathbf{R} \in \mathbb{SO}_3 \text{ s.t. } \mathbf{P} = \gamma [\mathbf{I} \ \mathbf{d}] \mathbf{R} \}. \end{aligned}$$

Camera resection is to compute the rotational and translational parts of the camera models, as well as the scale factor if any. More specifically, this means that we consider our unknowns to be  $\mathbf{R}$  and  $\mathbf{t}$  for the orthographic camera and  $\gamma$ ,  $\mathbf{R}$  and  $\mathbf{t}$  for the weak-perspective and paraperspective cameras. The direction  $\mathbf{d}$  used in the paraperspective camera is assumed known. In practice, it is chosen following (Horaud et al., 1997) as the sightline passing through the centroid of the image points in  $\mathbf{Y}$ .

We now discuss the question of whether intrinsic camera parameters are needed to apply the above three models. We define a camera to be calibrated when its usual perspective intrinsic parameter matrix  $\mathbf{K}$  is known, which contains in particular its effective focal length  $f \in \mathbb{R}$  and its principal point  $\mathbf{q}_0 \in \mathbb{R}^2$ , both in px unit. The weak-perspective camera does not have intrinsic parameters and may be fitted directly to the image data. The notion of calibration does therefore not apply to this model. The orthographic and paraperspective cameras however require one to use some intrinsic parameters. More specifically, the orthographic camera has one intrinsic parameter. It is a scale factor, which acts similarly as the scale factor of the weak-perspective camera. It is defined as  $\gamma = \frac{f}{d}$ , where  $d$  is the average camera to object distance in metric unit. Even though  $d$ , and thus  $\gamma$ , are data-dependent, we deem the orthographic camera to be calibrated when  $\gamma$  is known and uncalibrated otherwise. The paraperspective camera requires one to estimate the direction  $\mathbf{d}$ , as explained above, which represents two intrinsic parameters. The process of mapping the centroid of the image points to the sightline requires matrix  $\mathbf{K}$  to be known. Even though  $\mathbf{d}$  is data-dependent we deem the paraperspective camera to be calibrated when  $\mathbf{d}$  is known and uncalibrated otherwise. Consequently, the paraperspective camera is calibrated for any set of image points  $\mathbf{Y}$  and a known  $\mathbf{K}$  matrix.

Approximating the perspective camera is not the only use-case for the affine camera. There exists another use-case, which is to model telecentric lenses, which in effect perform a parallel projection (Steger, 2017b). Conversely, an affine camera can be interpreted as a camera mounted with a bilateral telecentric lens (Steger, 2017b, Theorem 1). Therefore, if a camera is calibrated and mounted with a telecentric lens, the pose computation problem can always be reduced to the orthographic case, which is thus the most important practical case.

### 3 Problem Statement and Formulations

We start from an initial formulation, from which we derive a translationless formulation, following the usual principle when working with an affine camera of centering the point sets. We then use the SVD (Singular Value Decomposition) to simplify the formulation and arrive at the first and second canonical formulations, respectively depending on the camera's full rotation part and its  $(2 \times 2)$  leading submatrix. Our derivations are kept generic in that they apply to the three metric affine cameras. The two canonical formulations are equivalent but both are needed to derive our algebraic procedures. Three equivalent formulations were derived using the QR decomposition in (Steger, 2017a).

#### 3.1 Initial Formulation

We define  $* \in \{\text{OR}, \text{WP}, \text{PP}\}$  as a variable indicating that we work with the three metric affine cameras. The generic  $L_2$ -optimal affine resection problem is written as:

$$\min_{\substack{\mathbf{P} \in \mathbb{M}_* \\ \mathbf{t} \in \mathbb{R}^2}} \mathcal{O}_1(\mathbf{P}, \mathbf{t}) \quad \text{with} \quad \mathcal{O}_1(\mathbf{P}, \mathbf{t}) \stackrel{\text{def}}{=} \left\| \mathbf{P}\mathbf{X} + \mathbf{t}\mathbf{1}^\top - \mathbf{Y} \right\|_{\mathcal{F}}^2. \quad (1)$$

#### 3.2 Translationless Formulation

We define  $\mathbf{X}' \in \mathbb{R}^{3 \times m}$  and  $\mathbf{Y}' \in \mathbb{R}^{2 \times m}$  as the centred model and observed point sets  $\mathbf{X}$  and  $\mathbf{Y}$  respectively, and  $\mathbf{x} \in \mathbb{R}^3$  and  $\mathbf{y} \in \mathbb{R}^2$  as their centres of gravity. Following the derivation given in appendix A, the initial formulation (1) can be rewritten without loss of generality under the translationless formulation:

$$\min_{\mathbf{P} \in \mathbb{M}_*} \mathcal{O}_3(\mathbf{P}) \quad \text{with} \quad \mathcal{O}_3(\mathbf{P}) \stackrel{\text{def}}{=} \left\| \mathbf{P}\mathbf{X}' - \mathbf{Y}' \right\|_{\mathcal{F}}^2. \quad (2)$$

Once the rotational part  $\mathbf{P}$  is estimated, the translation is given by:

$$\mathbf{t} = \mathbf{y} - \mathbf{P}\mathbf{x}. \quad (3)$$

#### 3.3 First Canonical Formulation

Because the model points are coplanar, but not colinear, and because  $\mathbf{X}'$  is centred, we have  $\text{rank}(\mathbf{X}') = 2$ . We use the SVD  $\mathbf{U}\Sigma\mathbf{V}^\top \stackrel{\text{def}}{=} \mathbf{X}'$ , with  $\mathbf{U} \in \mathbb{O}_3$ ,  $\mathbf{V} \in \mathbb{O}_m$  and  $\Sigma \stackrel{\text{def}}{=} [\bar{\Sigma} \ \mathbf{0}] \in \mathbb{R}^{3 \times m}$ , with  $\bar{\Sigma} \stackrel{\text{def}}{=} \text{stk}(\mathbf{W}, \mathbf{0}^\top) \in \mathbb{R}^{3 \times 2}$  and  $\mathbf{W} \stackrel{\text{def}}{=} \text{diag}(\sigma_1, \sigma_2)$ . We have, thanks to the non-colinearity assumption,  $\sigma_1 \geq \sigma_2 > 0$ . We modify the SVD to ensure  $\mathbf{U} \in \mathbb{S}\mathbb{O}_3$  by defining  $\mathbf{U}' \stackrel{\text{def}}{=} \det(\mathbf{U})\mathbf{U}$  and  $\mathbf{V}' \stackrel{\text{def}}{=} \det(\mathbf{U})\mathbf{V}$ , so that  $\mathbf{X}' = \mathbf{U}'\Sigma\mathbf{V}'^\top$  and  $\mathbf{V}' \in \mathbb{O}_m$ . Substituting

this modified SVD into the cost function of the translationless formulation, we obtain:

$$\mathcal{O}_3(\mathbf{P}) = \left\| \mathbf{P}\mathbf{U}'\Sigma\mathbf{V}'^\top - \mathbf{Y}' \right\|_{\mathcal{F}}^2 = \left\| \mathbf{P}\mathbf{U}'\Sigma - \mathbf{Y}'\mathbf{V}' \right\|_{\mathcal{F}}^2.$$

Because  $\mathbf{U}' \in \mathbb{S}\mathbb{O}_3$  we have  $\mathbf{A} = \mathbf{P}\mathbf{U}' \in \mathbb{M}_*$ . We can thus rewrite the problem as:

$$\min_{\mathbf{A} \in \mathbb{M}_*} \mathcal{O}_4(\mathbf{A}) \quad \text{with} \quad \mathcal{O}_4(\mathbf{A}) \stackrel{\text{def}}{=} \left\| \mathbf{A}\Sigma - \mathbf{Y}'\mathbf{V}' \right\|_{\mathcal{F}}^2.$$

We define  $[\mathbf{Z} \ \bar{\mathbf{Z}}] \stackrel{\text{def}}{=} \mathbf{Y}'\mathbf{V}' \in \mathbb{R}^{2 \times m}$  with  $\mathbf{Z} \in \mathbb{R}^{2 \times 2}$  and  $\bar{\mathbf{Z}} \in \mathbb{R}^{2 \times (m-2)}$  respectively. The cost function is expanded as:

$$\mathcal{O}_4(\mathbf{A}) = \left\| \mathbf{A}\bar{\Sigma} - \mathbf{Z} \right\|_{\mathcal{F}}^2 + \left\| \bar{\mathbf{Z}} \right\|_{\mathcal{F}}^2.$$

The second term is constant, and we thus arrive at the first canonical problem formulation as:

$$\min_{\mathbf{A} \in \mathbb{M}_*} \mathcal{O}_5(\mathbf{A}) \quad \text{with} \quad \mathcal{O}_5(\mathbf{A}) \stackrel{\text{def}}{=} \left\| \mathbf{A}\bar{\Sigma} - \mathbf{Z} \right\|_{\mathcal{F}}^2. \quad (4)$$

The cost function  $\mathcal{O}_5$  is related to the initial cost by:

$$\mathcal{O}_5(\mathbf{A}) = \mathcal{O}_4(\mathbf{A}) - \mathcal{O} \quad \text{with} \quad \mathcal{O} \stackrel{\text{def}}{=} \left\| \bar{\mathbf{Z}} \right\|_{\mathcal{F}}^2. \quad (5)$$

### 3.4 Second Canonical Formulation

The last row of  $\bar{\Sigma}$  is zero, meaning that the cost function does not depend on the last column of  $\mathbf{A}$ . We may thus define the first two columns of  $\mathbf{A}$  as a matrix  $\mathbf{B} \stackrel{\text{def}}{=} \mathbf{A}\Pi_0^\top$ , and arrive at the second canonical problem formulation as:

$$\min_{\mathbf{B} \in \bar{\mathbb{M}}_*} \mathcal{O}_6(\mathbf{B}) \quad \text{with} \quad \mathcal{O}_6(\mathbf{B}) \stackrel{\text{def}}{=} \left\| \mathbf{B}\mathbf{W} - \mathbf{Z} \right\|_{\mathcal{F}}^2, \quad (6)$$

and  $\bar{\mathbb{M}}_*$  is the space of the leading  $(2 \times 2)$  submatrices of members of  $\mathbb{M}_*$ , defined as:

$$\bar{\mathbb{M}}_* \stackrel{\text{def}}{=} \left\{ \mathbf{B} \in \mathbb{R}^{2 \times 2} \mid \exists \mathbf{b} \in \mathbb{R}^2 \text{ s.t. } [\mathbf{B} \ \mathbf{b}] \in \mathbb{M}_* \right\}.$$

We have:

$$\mathcal{O}_6 \left( \mathbf{A}\Pi_0^\top \right) = \mathcal{O}_5(\mathbf{A}). \quad (7)$$

### 3.5 Formulation Group

An optimization problem's formulation generally has multiple very similar instances sharing the same parameterization. These instances are obtained by applying a *formulation transformation* with which the solutions covary or are left invariant. This is related but different from the gauge, which represents the parameterization's internal degrees of freedom. Identifying the formulation transformations and the gauge is important as they may then be used to simplify the formulation or the expression of its solutions. The canonical formulations (4) and (6) have multiple instances but do not have gauge freedoms. They share the same set of formulation transformations, which forms a group of dimension two, which we call the *formulation group*. One of these dimensions represents a positive rescaling. This holds for two reasons. First, as for any optimization problem, multiplying the cost by a positive scalar does not change its extrema. Second, this also preserves the parameterization in both canonical formulations. The other dimension in the formulation group represents an orthonormal transformation. This holds for two reasons. First, the canonical formulations are based on the Frobenius norm, which is invariant to orthonormal transformations. Second, left-multiplying the unknown by an orthonormal transformation also preserves the parameterization. We aggregate the two degrees of freedom in a scaled orthonormal transformation representing a similarity. The formulation group can thus be parameterized by three variables  $a, b \in \mathbb{R}$  and  $s \in \{-1, 1\}$ , with its members defined as:

$$\mathbf{N} \stackrel{\text{def}}{=} \begin{bmatrix} sa & sb \\ -b & a \end{bmatrix}.$$

We also define  $\varepsilon \stackrel{\text{def}}{=} \sqrt{a^2 + b^2}$ . The cost functions in the canonical formulations then implicitly depend on parameters  $a, b$  and  $s$ , as:

$$\mathcal{O}_5(\mathbf{A}') = \|\mathbf{A}'\bar{\Sigma} - \frac{1}{\varepsilon}\mathbf{NZ}\|_{\mathcal{F}}^2 \quad \text{and} \quad \mathbf{A}' \stackrel{\text{def}}{=} \frac{1}{\varepsilon}\mathbf{NA} \in \mathbb{M}_*, \quad (8)$$

and:

$$\mathcal{O}_6(\mathbf{B}') = \|\mathbf{B}'\bar{\mathbf{W}} - \frac{1}{\varepsilon}\mathbf{NZ}\|_{\mathcal{F}}^2 \quad \text{and} \quad \mathbf{B}' \stackrel{\text{def}}{=} \frac{1}{\varepsilon}\mathbf{NB} \in \bar{\mathbb{M}}_*. \quad (9)$$

Therefore,  $\sqrt{\varepsilon}$  represents the cost functions' rescaling,  $\text{atan2}(b, a)$  a rotation angle and  $s$  the similarity of the formulation. We are free to choose  $a$  and  $b$  with  $\varepsilon \neq 0$  and  $s$  to simplify the problem's formulation or solution.

### 3.6 Algebraic Procedure

We give the algebraic procedure that prepares the canonical formulations' parameters in table 1. This procedure does not exploit the formulation group, as it is only used for the orthographic camera.

<p style="text-align: center;">Function <b>PrepareData</b>(<math>\mathbf{X} \in \mathbb{R}^{3 \times m}</math>, <math>\mathbf{Y} \in \mathbb{R}^{2 \times m}</math>)</p> <ul style="list-style-type: none"> <li>• Set <math>\mathbf{x} \leftarrow \frac{1}{m}\mathbf{X}\mathbf{1}</math> and <math>\mathbf{y} \leftarrow \frac{1}{m}\mathbf{Y}\mathbf{1}</math></li> <li>• Set <math>\mathbf{X}' \leftarrow \mathbf{X} - \mathbf{x}\mathbf{1}^\top</math> and <math>\mathbf{Y}' \leftarrow \mathbf{Y} - \mathbf{y}\mathbf{1}^\top</math></li> <li>• Set <math>(\mathbf{U}, \Sigma, \mathbf{V}) \leftarrow \text{SVD}(\mathbf{X}')</math></li> <li>• Set <math>\mathbf{Z} \leftarrow \det(\mathbf{U})\mathbf{Y}'[\mathbf{v}_1 \ \mathbf{v}_2]</math></li> <li>• Set <math>\mathcal{O} \leftarrow \ \mathbf{Y}'[\mathbf{v}_3 \ \dots \ \mathbf{v}_m]\ _2^2</math></li> <li>• Set <math>\sigma_1 \leftarrow \Sigma_{11}</math> and <math>\sigma_2 \leftarrow \Sigma_{22}</math> (we have <math>\mathbf{W} = \text{diag}(\sigma_1, \sigma_2)</math>)</li> </ul> <p style="text-align: center;">Output <math>\sigma_1, \sigma_2 \in \mathbb{R}^+</math>, <math>\mathbf{Z} \in \mathbb{R}^{2 \times 2}</math>, <math>\mathcal{O} \in \mathbb{R}^+</math>, <math>\mathbf{x}, \mathbf{y} \in \mathbb{R}^3</math>, <math>\mathbf{U} \in \mathbb{O}_3</math></p>
--

Table 1: **Algebraic procedure to prepare metric affine planar resection.** The procedure prepares the data for all metric affine cameras.

## 4 Resection of the Orthographic Camera

The case of the orthographic camera is the most involved. We start with specializing the canonical formulations.

### 4.1 The Canonical Formulations

For the orthographic camera we have  $\mathbb{M}_{\text{OR}} = \mathbb{SS}_{23}$  and  $\bar{\mathbb{M}}_{\text{OR}} = \mathbb{SS}_{22}$ . The canonical formulations for the orthographic camera are thus respectively given by:

$$\min_{\mathbf{A} \in \mathbb{SS}_{23}} \mathcal{O}_5(\mathbf{A}), \quad (10)$$

and:

$$\min_{\mathbf{B} \in \mathbb{SS}_{22}} \mathcal{O}_6(\mathbf{B}). \quad (11)$$

The  $\mathbb{SS}_{22}$  group is formed of those matrices whose largest singular value is one (Cardoso and Zietak, 2015). Our analytic solution uses both canonical formulations. It starts with the first canonical formulation, and shows that multiple cases arise. It may seem more natural at first to solve the second canonical formulation directly. We show next the difficulty in trying to do so.

## 4.2 A Difficult Quartically Constrained Problem

The difficulty of formulation (11) stems from the constraints characterizing  $\mathbb{SS}_{22}$ . We have:

$$\mathbf{B} \in \mathbb{SS}_{22} \Leftrightarrow \mu_+(\mathbf{B}) = 1. \quad (12)$$

This constraint implies by definition of  $\mu_{\pm}(\mathbf{B})$  that  $0 \leq \mu_-(\mathbf{B}) \leq 1$ . The singular values  $\mu_{\pm}(\mathbf{B})$  are given as the square root of the eigenvalues of  $\mathbf{B}^\top \mathbf{B}$ , and we thus have:

$$\mu_{\pm}(\mathbf{B}) \stackrel{\text{def}}{=} \sqrt{\frac{1}{2} \left( \|\mathbf{B}\|_{\mathcal{F}}^2 \pm \sqrt{(\|\mathbf{b}_1\|_2^2 - \|\mathbf{b}_2\|_2^2)^2 + 4(\mathbf{b}_1^\top \mathbf{b}_2)^2} \right)}.$$

Substituting this expression into the characterization (12) of  $\mathbb{SS}_{22}$ , squaring, multiplying by 2 and subtracting  $\|\mathbf{B}\|_{\mathcal{F}}^2$ , we obtain:

$$\mathbf{B} \in \mathbb{SS}_{22} \Leftrightarrow \sqrt{(\|\mathbf{b}_1\|_2^2 - \|\mathbf{b}_2\|_2^2)^2 + 4(\mathbf{b}_1^\top \mathbf{b}_2)^2} = 2 - \|\mathbf{B}\|_{\mathcal{F}}^2. \quad (13)$$

The right-hand side of this equation must be positive,  $2 - \|\mathbf{B}\|_{\mathcal{F}}^2 \geq 0$ , and so  $\|\mathbf{B}\|_{\mathcal{F}}^2 \leq 2$ . Squaring, we rewrite the characterization of  $\mathbb{SS}_{22}$  as:

$$\mathbf{B} \in \mathbb{SS}_{22} \Leftrightarrow \begin{cases} (\|\mathbf{b}_1\|_2^2 - \|\mathbf{b}_2\|_2^2)^2 + 4(\mathbf{b}_1^\top \mathbf{b}_2)^2 = (2 - \|\mathbf{B}\|_{\mathcal{F}}^2)^2 \\ \|\mathbf{B}\|_{\mathcal{F}}^2 \leq 2. \end{cases} \quad (14)$$

Expanding and simplifying the first equation, we arrive at:

$$\mathbf{B} \in \mathbb{SS}_{22} \Leftrightarrow \begin{cases} \|\mathbf{B}\|_{\mathcal{F}}^2 - \det(\mathbf{B})^2 - 1 = 0 \\ \|\mathbf{B}\|_{\mathcal{F}}^2 \leq 2. \end{cases} \quad (15)$$

The equality constraint and an equivalent inequality constraint were also derived in (Steger, 2017a). The implication of equation (15) is that, though the cost function  $\mathcal{O}_6$  is convex in  $\mathbf{B}$ , the latter must meet a *quartic* equality constraint and a *quadratic* inequality constraint. An analytic solution to the formulation (11) is therefore extremely difficult to derive. A numerical solution to a very similar problem was described in (Steger, 2017a).

## 4.3 Solving via Multiple Cases

We use the first canonical formulation (10) to show that the problem has multiple cases which are each solved with a specific procedure. Let  $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{R}^3$  be the two rows of  $\mathbf{A}$  and  $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^2$  be the two rows of  $\mathbf{Z}$ .

We rewrite the cost function as:

$$\mathcal{O}_5(\mathbf{A}) = \left\| \bar{\Sigma}^\top \mathbf{a}_1 - \mathbf{z}_1 \right\|_2^2 + \left\| \bar{\Sigma}^\top \mathbf{a}_2 - \mathbf{z}_2 \right\|_2^2.$$

The constraint  $\mathbf{A} \in \mathbb{SS}_{23}$  is made of three orthonormality constraints, namely  $\|\mathbf{a}_1\|_2 = 1$ ,  $\|\mathbf{a}_2\|_2 = 1$  and  $\mathbf{a}_1^\top \mathbf{a}_2 = 0$ . So as to form the problem's Lagrangian, we introduce a vector  $\boldsymbol{\ell} = \text{stk}(\ell_1, \ell_2, \ell_3) \in \mathbb{R}^3$  of three Lagrange multipliers, giving:

$$\mathcal{L}(\mathbf{A}, \boldsymbol{\ell}) \stackrel{\text{def}}{=} \left\| \bar{\Sigma}^\top \mathbf{a}_1 \right\|_2^2 + \|\mathbf{z}_1\|_2^2 - 2\mathbf{a}_1^\top \bar{\Sigma} \mathbf{z}_1 + \left\| \bar{\Sigma}^\top \mathbf{a}_2 \right\|_2^2 + \|\mathbf{z}_2\|_2^2 - 2\mathbf{a}_2^\top \bar{\Sigma} \mathbf{z}_2 + \ell_1 (\|\mathbf{a}_1\|_2^2 - 1) + \ell_2 (\|\mathbf{a}_2\|_2^2 - 1) + 2\ell_3 \mathbf{a}_1^\top \mathbf{a}_2.$$

Differentiating with respect to  $\mathbf{a}_1$  and  $\mathbf{a}_2$  and nullifying gives:

$$\begin{aligned} \frac{1}{2} \frac{\partial \mathcal{L}}{\partial \mathbf{a}_1} &= \bar{\Sigma} \bar{\Sigma}^\top \mathbf{a}_1 - \bar{\Sigma} \mathbf{z}_1 + \ell_1 \mathbf{a}_1 + \ell_3 \mathbf{a}_2 = \mathbf{0} \\ \frac{1}{2} \frac{\partial \mathcal{L}}{\partial \mathbf{a}_2} &= \bar{\Sigma} \bar{\Sigma}^\top \mathbf{a}_2 - \bar{\Sigma} \mathbf{z}_2 + \ell_2 \mathbf{a}_2 + \ell_3 \mathbf{a}_1 = \mathbf{0}. \end{aligned}$$

This is equivalent to the following system:

$$\begin{bmatrix} \bar{\Sigma} \bar{\Sigma}^\top + \ell_1 \mathbf{I} & \ell_3 \mathbf{I} \\ \ell_3 \mathbf{I} & \bar{\Sigma} \bar{\Sigma}^\top + \ell_2 \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} = \begin{bmatrix} \bar{\Sigma} \mathbf{z}_1 \\ \bar{\Sigma} \mathbf{z}_2 \end{bmatrix}.$$

This system cannot be used directly to solve for  $\mathbf{A}$  because it involves the unknown  $\boldsymbol{\ell}$ , and would require us to reintroduce the three nonlinear orthonormality constraints to find a solution. We can however use this system to derive multiple cases with restricted solution spaces. In this system, the design matrix is  $(6 \times 6)$ , symmetric and made of four blocks, each being  $(3 \times 3)$  and diagonal. With this special structure, defining:

$$\mathbf{M} \stackrel{\text{def}}{=} \begin{bmatrix} \ell_1 & \ell_3 \\ \ell_3 & \ell_2 \end{bmatrix},$$

and naming the three columns of  $\mathbf{A}$  as  $\mathbf{b}_1$ ,  $\mathbf{b}_2$  and  $\mathbf{b}$ , and the two columns of  $\mathbf{Z}$  as  $\mathbf{z}'_1$  and  $\mathbf{z}'_2$ , the system is equivalent to:

$$\mathbf{M}_1 \mathbf{b}_1 = \sigma_1 \mathbf{z}'_1 \quad \text{with} \quad \mathbf{M}_1 \stackrel{\text{def}}{=} \mathbf{M} + \sigma_1^2 \mathbf{I} \quad (16)$$

$$\mathbf{M}_2 \mathbf{b}_2 = \sigma_2 \mathbf{z}'_2 \quad \text{with} \quad \mathbf{M}_2 \stackrel{\text{def}}{=} \mathbf{M} + \sigma_2^2 \mathbf{I} \quad (17)$$

$$\mathbf{M} \mathbf{b} = \mathbf{0}. \quad (18)$$

The multiple cases are derived by analyzing the rank of the three  $(2 \times 2)$  design matrices  $\mathbf{M}$ ,  $\mathbf{M}_1$  and  $\mathbf{M}_2$  involved in the three subsystems. We name each case with three numbers in  $\{0, 1, 2\}$  representing the rank of these three matrices, and replace the number by ‘x’ to mean any rank in  $\{0, 1, 2\}$ . For instance, our first cases is 2.x.x, which means that  $\text{rank}(\mathbf{M}) = 2$ , while  $\text{rank}(\mathbf{M}_1)$  and  $\text{rank}(\mathbf{M}_2)$  can be different and any of  $\{0, 1, 2\}$ . Because the Lagrange multipliers forming the entries of matrix  $\mathbf{M}$  are unknown, the matrices  $\mathbf{M}_1$  and  $\mathbf{M}_2$  are unknown too. It is thus necessary in our algebraic procedure of §4.7 to test all feasible cases and select the actual one a posteriori. The following table summarizes our findings on the feasibility of the different cases:

Cases 2.x.x	Implies $\mathbf{B} \in \mathbb{O}_2$ , solved in §4.4
Cases 1.x.x	Handled with one subcase
Case 1.2.2	Solved in §4.5
Case 1.1.2	Impossible, shown in §G.1
Case 1.2.1	Impossible, shown in §G.2
Case 1.1.1	Impossible, shown in §G.3
Cases 1.0.x	Impossible, shown in §G.4
Cases 1.x.0	Impossible, shown in §G.5
Cases 0.x.x	Implies $\mathbf{B} \in \mathbb{SS}_{22} \setminus \mathbb{O}_2$ , handled with one subcase
Case 0.2.2	Solved in §4.6, handled by the solution of case 1.2.2
	All other subcases impossible, shown in §4.6

Case 1.2.2 is the most general, and also the most difficult to solve. Cases 1.x.x and 0.x.x are similar in the sense that they both have two solutions in general, are represented by only one feasible subcase with the others subcases leading to contradictions, and can be solved by the same algebraic procedure. Cases 2.x.x however always possess a single solution, and require a dedicated algebraic procedure. This is because, as shown in the next section, they imply  $\mathbf{B} \in \mathbb{O}_2$  and thus  $\mathbf{A} = \text{diag}(\mathbf{B}, \det(\mathbf{B}))$ . In other words, the optimal rotation in this case must be about the  $z$  axis. The next two sections solve cases 2.x.x and 1.2.2 respectively. The impossibility of the other cases in 1.x.x is shown in appendix G. The remaining sections study cases 0.x.x and give a complete algebraic procedure covering all cases.

#### 4.4 Cases 2.x.x

We first give the parameterization we use for this specific case and then our solution to the problem. We finally assemble an expression for  $\mathbf{R}$  and  $\mathbf{t}$ .

**Solving for B.** We have  $\text{rank}(\mathbf{M}) = 2$  and we can thus use equation (18) to obtain  $\mathbf{b} = \mathbf{0}$ . Consequently,  $\mathbf{A} = [\mathbf{B} \ \mathbf{0}]$  and  $\mathbf{A} \in \mathbb{SS}_{23}$  is written as  $\mathbf{B}\mathbf{B}^\top = \mathbf{I}$ , which means  $\mathbf{B} \in \mathbb{O}_2 \subset \mathbb{SS}_{22}$ . The second canonical formulation (11) is thus simplified to:

$$\min_{\mathbf{B} \in \mathbb{O}_2} \mathcal{O}_6(\mathbf{B}).$$

This is an instance of the absolute orientation or orthogonal Procrustes problem in 2D, which can be solved by adapting existing 3D methods (Arun et al., 1987; Horn et al., 1988). The other existing methods (Haralick et al., 1989; Umeyama, 1991) solve a similar problem but under the constraint  $\mathbf{B} \in \mathbb{SO}_2$ , meaning  $\det(\mathbf{B}) = 1$ , whereas in our case we have to determine if  $\det(\mathbf{B}) = 1$  or  $\det(\mathbf{B}) = -1$  from the data. The problem has an elegant and compact solution, based on parameterizing  $\mathbf{B} \in \mathbb{O}_2$  as:

$$\mathbf{B} = \begin{bmatrix} sa & -b \\ sb & a \end{bmatrix} \quad \text{with} \quad a, b \in \mathbb{R}, \quad a^2 + b^2 = 1, \quad s \in \{-1, 1\}. \quad (19)$$

The problem thus becomes:

$$\min_{\substack{a, b, s \\ a^2 + b^2 = 1 \\ s \in \{-1, 1\}}} \mathcal{O}_6(a, b, s).$$

We give an optimized solution in appendix I.

**Forming R and t.** Starting from the parameterization (19) of  $\mathbf{B}$ , we have:

$$\mathbf{A} = \begin{bmatrix} \mathbf{B} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} sa & -b & 0 \\ sb & a & 0 \end{bmatrix},$$

which leads to:

$$\mathbf{R} = \det(\mathbf{U}) \begin{bmatrix} sa & -b & 0 \\ sb & a & 0 \\ 0 & 0 & s \end{bmatrix} \mathbf{U}^\top.$$

and:

$$\mathbf{t} = \mathbf{y} - \Pi_0 \bar{\mathbf{R}} \mathbf{x}.$$

#### 4.5 Case 1.2.2

This is the most general and the most involved case. We first give a specific parameterization for cases 1.x.x and reformulate the problem to express the orthonormality constraints on a reduced set of parameters. This results in a system of two polynomials and two rational functions, which leads us to our solution strategy.

We then derive a univariate polynomial, study its characteristics, and show how to subsequently recover  $\mathbf{B}$ .

#### 4.5.1 Parameterization for Cases 1.x.x

We have  $\text{rank}(\mathbf{M}) = 1$  and, since  $\mathbf{M}$  is symmetric, we can parameterize it by  $\mathbf{q} \in \mathbb{R}^2$ ,  $\|\mathbf{q}\|_2 = 1$  and  $\beta \in \mathbb{R}^*$  as:

$$\mathbf{M} = \beta \mathbf{q} \mathbf{q}^\top. \quad (20)$$

We have that the kernel of  $\mathbf{M}$  is  $\ker(\mathbf{M}) \propto \mathbf{q}^\perp \propto \mathbf{S} \mathbf{q}$  and its eigenvector is  $\mathbf{q}$ , with eigenvalue  $\beta$ . We can then parameterize  $\mathbf{b}$  using equation (18), which implies  $\mathbf{b} \in \ker(\mathbf{M})$  or  $\mathbf{b} = \mathbf{0}$ , with a free parameter  $\alpha \in \mathbb{R}$  as:

$$\mathbf{b} = \alpha \mathbf{S} \mathbf{q}. \quad (21)$$

We note that from equations (16) and (17)  $\text{rank}(\mathbf{M}_j) = 1$  is equivalent to  $\beta = -\sigma_j^2$ , for  $j \in \{1, 2\}$ . This is because we have  $\mathbf{M}_j = \beta \mathbf{q} \mathbf{q}^\top + \sigma_j^2 \mathbf{I}$ , and a short calculation shows that  $\lambda_1(\mathbf{M}_j) = \sigma_j^2 > 0$  and  $\lambda_2(\mathbf{M}_j) = \beta + \sigma_j^2$ , which vanishes for  $\beta = -\sigma_j^2$ . Our goal is now to find  $\mathbf{q}$ ,  $\beta$ ,  $\alpha$  and  $\mathbf{B}$  for each subcase.

#### 4.5.2 Reformulation

Our first step is to reformulate the problem in terms of  $\mathbf{Q}^\top \mathbf{A} = [\mathbf{Q}^\top \mathbf{B} \mathbf{Q}^\top \mathbf{b}]$ , with  $\mathbf{Q} \stackrel{\text{def}}{=} [\mathbf{q} \ \mathbf{S} \mathbf{q}] \in \mathbb{SO}_2$ . This allows us to express the orthonormality constraints on  $\mathbf{q}$ ,  $\beta$  and  $\alpha$  in the next section. We start by forming the equations concerning  $\mathbf{Q}^\top \mathbf{B}$ . By substituting the parameterization (20) into equations (16) and (17), and with  $j \in \{1, 2\}$ , we have:

$$\left( \beta \mathbf{q} \mathbf{q}^\top + \sigma_j^2 \mathbf{I} \right) \mathbf{b}_j = \sigma_j \mathbf{z}'_j,$$

which forms a set of quartics in  $\mathbf{q}$ ,  $\beta$  and  $\mathbf{B}$ . By multiplying to the left by  $\mathbf{q}^\top \mathbf{S}$  and  $\mathbf{q}^\top$ , we obtain, respectively:

$$\begin{aligned} \sigma_j \mathbf{q}^\top \mathbf{S} \mathbf{b}_j &= \mathbf{q}^\top \mathbf{S} \mathbf{z}'_j \\ (\beta + \sigma_j^2) \mathbf{q}^\top \mathbf{b}_j &= \sigma_j \mathbf{q}^\top \mathbf{z}'_j, \end{aligned}$$

which forms a set of quadratics and cubics in  $\mathbf{q}$ ,  $\beta$  and  $\mathbf{B}$ , that we rewrite in matrix form as:

$$\begin{aligned} \mathbf{q}^\top \mathbf{S} \mathbf{B} \mathbf{W} &= \mathbf{q}^\top \mathbf{S} \mathbf{Z} \\ \mathbf{q}^\top \mathbf{B} (\mathbf{W}^2 + \beta \mathbf{I}) &= \mathbf{q}^\top \mathbf{Z} \mathbf{W}, \end{aligned}$$

and transform in the following two vector equations, where  $\mathbf{Q}^\top \mathbf{B}$  eventually appears. This is achieved by multiplying the first equation by  $-\mathbf{W}^{-1}$  and using the property  $-\mathbf{S} = \mathbf{S}^\top$ , and multiplying the second equation by  $(\mathbf{W}^2 + \beta \mathbf{I})^{-1}$ :

$$\begin{aligned}\mathbf{q}^\top \mathbf{S}^\top \mathbf{B} &= \mathbf{q}^\top \mathbf{S}^\top \mathbf{Z} \mathbf{W}^{-1} \\ \mathbf{q}^\top \mathbf{B} &= \mathbf{q}^\top \mathbf{Z} \mathbf{W} (\mathbf{W}^2 + \beta \mathbf{I})^{-1}.\end{aligned}$$

The first equation is still quadratic, but because of the matrix inverse, the second equation is quadratic and linear-fractional. We gather these two equations in a single matrix equation:

$$\mathbf{Q}^\top \mathbf{B} = \text{stk} \left( \mathbf{q}^\top \mathbf{Z} \mathbf{W} (\mathbf{W}^2 + \beta \mathbf{I})^{-1}, \mathbf{q}^\top \mathbf{S}^\top \mathbf{Z} \mathbf{W}^{-1} \right).$$

By observing using equation (21) that  $\mathbf{Q}^\top \mathbf{b} = \text{stk}(0, \alpha)$ , we arrive at:

$$\mathbf{Q}^\top \mathbf{A} = \begin{bmatrix} \mathbf{Q}^\top \mathbf{B} & \mathbf{Q}^\top \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{q}^\top \mathbf{Z} \mathbf{W} (\mathbf{W}^2 + \beta \mathbf{I})^{-1} & 0 \\ \mathbf{q}^\top \mathbf{S}^\top \mathbf{Z} \mathbf{W}^{-1} & \alpha \end{bmatrix}. \quad (22)$$

### 4.5.3 Introduction of the Orthonormality Constraints

The orthonormality constraints are  $\mathbf{A} \mathbf{A}^\top = \mathbf{I}$ . Since  $\mathbf{Q} \in \mathbb{S}\mathbb{O}_2$ , we also have  $\mathbf{Q}^\top \mathbf{A} \mathbf{A}^\top \mathbf{Q} = \mathbf{I}$ . By substituting  $\mathbf{Q}^\top \mathbf{A}$  from equation (22), we arrive at the following system of equations, also including the unit norm constraint of vector  $\mathbf{q}$  as last equation:

$$\left\{ \begin{aligned} \left\| (\mathbf{W}^2 + \beta \mathbf{I})^{-1} \mathbf{W} \mathbf{Z}^\top \mathbf{q} \right\|_2^2 &= 1 & (23) \end{aligned} \right. \quad (23)$$

$$\left\{ \begin{aligned} \left\| \mathbf{W}^{-1} \mathbf{Z}^\top \mathbf{S} \mathbf{q} \right\|_2^2 + \alpha^2 &= 1 & (24) \end{aligned} \right. \quad (24)$$

$$\left\{ \begin{aligned} \mathbf{q}^\top \mathbf{Z} (\mathbf{W}^2 + \beta \mathbf{I})^{-1} \mathbf{Z}^\top \mathbf{S} \mathbf{q} &= 0 & (25) \end{aligned} \right. \quad (25)$$

$$\left\{ \begin{aligned} \|\mathbf{q}\|_2^2 &= 1. & (26) \end{aligned} \right. \quad (26)$$

Equation (25) represents orthogonality, and is a simplification of  $\mathbf{q}^\top \mathbf{Z} \mathbf{W} (\mathbf{W}^2 + \beta \mathbf{I})^{-1} \mathbf{W}^{-1} \mathbf{Z}^\top \mathbf{S} \mathbf{q}$  obtained from the property that the product of diagonal matrices is commutative. We now have to solve this system of 4 equations in the 4 variables  $\mathbf{q}$ ,  $\alpha$  and  $\beta$ . Equations (23) and (25) are rational functions, respectively quadratic-quadratic and quadratic-linear, and equations (24) and (26) are quadratic.

#### 4.5.4 Solution Strategy

Our strategy is to reduce the system to a single polynomial equation in  $\beta$ . We first make a few observations on equations (23), (24), (25) and (26):

- Only equation (24) depends on  $\alpha$ ; we can thus ignore it to find  $\beta$  and  $\mathbf{q}$  in a first stage
- The signs of  $\mathbf{q}$  and  $\alpha$  are unconstrained
- The norm of  $\mathbf{q}$  does not change equation (25)

These observations suggest the following solution steps:

1. Express  $\mathbf{q}$  as a function of  $\beta$  from equations (23) and (26); even though this fixes the norm of  $\mathbf{q}$ , we only express  $\mathbf{q}$  up to scale at this stage as it yields a simpler expression and does not affect the next step
2. Substitute this expression for  $\mathbf{q}$  into equation (25) to construct the polynomial equation in  $\beta$ , and find candidate solutions for  $\beta$
3. Retrieve  $\mathbf{q}$  by backsubstituting  $\beta$  into the expression obtained in step 1 and renormalization
4. Solve for  $\alpha$  from equation (24)
5. Keep the solution which minimizes the cost

The next section presents the derivation of step 1, and the two sections after that the derivation of step 2 in detail. The first of these show that step 2 involves solving a polynomial  $p_1$  of degree 18. The second of these show that 12 roots of  $p_1$  can be found analytically and discarded, leaving a polynomial  $p_3$  of degree 6, called a sextic, to be solved. It shows that this sextic is not solvable by radicals and cannot be simplified using the formulation group. The last section summarizes the proposed solution method in an algebraic procedure.

#### 4.5.5 Expressing $\mathbf{q}$ as a Function of $\beta$

Equation (26) constrains  $\mathbf{q}$  to lie on the unit circle, while equation (23) constrains it to lie on an ellipse centred at the origin. We may thus have between 0 and 4 solutions for  $\mathbf{q}$ . However, it is obvious that if  $\mathbf{q}$  is a solution,  $-\mathbf{q}$  is a solution too, and we thus have between 0 and 2 solutions up to sign for  $\mathbf{q}$ . We derive the solution in appendix F. The two solutions for  $\mathbf{q}$  are given by:

$$\mathbf{q}_{\pm} \propto \left( \mathbf{S}(\bar{\mathbf{E}} - \mathbf{I}) \pm \sqrt{\text{tr}(\bar{\mathbf{E}}) - \det(\bar{\mathbf{E}}) - \mathbf{1}\mathbf{I}} \right) \mathbf{e}_1 \quad \text{with} \quad \bar{\mathbf{E}} = \mathbf{Z}\mathbf{W}(\mathbf{W}^2 + \beta\mathbf{I})^{-2}\mathbf{W}\mathbf{Z}^{\top}. \quad (27)$$

#### 4.5.6 Deriving a Univariate Polynomial in $\beta$

In order to form the polynomial in  $\beta$ , we first need to cancel the fractions with  $\beta$  in the denominator in the expression of  $\mathbf{q}_{\pm}$  and in equation (25), and then to substitute  $\mathbf{q}_{\pm}$  into equation (25). We define:

$$\Gamma \stackrel{\text{def}}{=} \mathbf{W}^2 + \beta \mathbf{I}, \quad \Gamma' \stackrel{\text{def}}{=} -\mathbf{S}\Gamma\mathbf{S} \quad \text{and} \quad \tau \stackrel{\text{def}}{=} \det(\Gamma),$$

and observe that  $\tau\Gamma^{-1} = \Gamma'$ . We multiply the expression of  $\mathbf{q}_{\pm}$  by  $\tau^2$ , giving:

$$\mathbf{q}_{\pm} \propto (\mathbf{S}\mathbf{L} \pm \tau\sqrt{r}\mathbf{I}) \mathbf{e}_1, \quad (28)$$

with:

$$\mathbf{L} \stackrel{\text{def}}{=} \mathbf{Z}\mathbf{W}\Gamma'^2\mathbf{W}\mathbf{Z}^{\top} - \tau^2\mathbf{I} \quad \text{and} \quad r \stackrel{\text{def}}{=} \text{tr}(\Gamma'^2\mathbf{W}\mathbf{Z}^{\top}\mathbf{Z}\mathbf{W}) - \det(\mathbf{W})^2 \det(\mathbf{Z})^2 - \tau^2.$$

We multiply equation (25) by  $\tau$ , obtaining:

$$\mathbf{q}^{\top}\mathbf{K}\mathbf{q} = 0 \quad \text{with} \quad \mathbf{K} \stackrel{\text{def}}{=} \mathbf{Z}\Gamma'\mathbf{Z}^{\top}\mathbf{S}. \quad (29)$$

Combining equations (28) and (29), we obtain:

$$\mathbf{e}_1^{\top} (\mathbf{L}\mathbf{S}^{\top} \pm \tau\sqrt{r}\mathbf{I}) \mathbf{K} (\mathbf{S}\mathbf{L} \pm \tau\sqrt{r}\mathbf{I}) \mathbf{e}_1 = 0,$$

which we expand to:

$$\mathbf{e}_1^{\top} (\mathbf{L}\mathbf{S}^{\top}\mathbf{K}\mathbf{S}\mathbf{L} + \tau^2 r \mathbf{K}) \mathbf{e}_1 = -\pm \tau\sqrt{r} \mathbf{e}_1^{\top} (\mathbf{K}\mathbf{S}\mathbf{L} + \mathbf{L}\mathbf{S}^{\top}\mathbf{K}) \mathbf{e}_1.$$

Squaring to cancel the radical, we arrive at the following polynomial equation of degree 18:

$$p_1(\beta) = 0 \quad \text{with} \quad p_1(\beta) \stackrel{\text{def}}{=} \left( \mathbf{e}_1^{\top} (\mathbf{L}\mathbf{S}^{\top}\mathbf{K}\mathbf{S}\mathbf{L} + \tau^2 r \mathbf{K}) \mathbf{e}_1 \right)^2 - \tau^2 r \left( \mathbf{e}_1^{\top} (\mathbf{K}\mathbf{S}\mathbf{L} + \mathbf{L}\mathbf{S}^{\top}\mathbf{K}) \mathbf{e}_1 \right)^2. \quad (30)$$

#### 4.5.7 Solving for $\beta$

We use Matlab's symbolic toolbox and Maple to understand the structure of  $p_1$ .

**Factoring.** We first observe that  $p_1$  has two simple roots,  $-\sigma_1^2$  and  $-\sigma_2^2$ , each repeated once, and that the remaining part can be factored as the product of two polynomials of lower degrees, a quartic  $p_2$  and a sextic

$p_3$ , with the quartic being repeated once:

$$p_1(\beta) = (\beta + \sigma_1^2)^2 (\beta + \sigma_2^2)^2 p_2(\beta)^2 p_3(\beta).$$

As discussed in §4.5.1, the simple root  $-\sigma_j^2$  implies that  $\text{rank}(\mathbf{M}_j) = 1$ , and thus is not dealt with in the present case, but in cases 1.1.2 and 1.2.1 for  $j = 1$  and  $j = 2$  respectively. The computational cost in solving for  $\beta$  primarily lies in finding the roots of the sextic  $p_3$ . We thus first try to use the formulation group in order to simplify the sextic and examine the quartic afterwards.

**Analysis of the sextic.** The general expression of the sextic's coefficients is large, and there is no special interest to give it at this stage. Anticipating slightly the result of our analysis of the next two paragraphs, we rather give the sextic's specialization under a particular formulation transformation, in table 2. In short, our analysis leads to a formulation transformation which implies  $Z_{12} = 0$  and  $\sigma_1 = 1$ , which shortens the expression of the sextic's coefficients, but does not change its complexity and solvability as reflected by its Galois group. The Galois group of a polynomial is the group of permutation that preserves any algebraic equation relating the polynomial's roots. The Galois group is important as its solvability allows one to determine if the polynomial is solvable by radicals (Stewart, 2015). Using Maple, we found that our sextic's Galois group is  $S_6$ , the symmetric group of degree 6. This is the most general permutation group, and it is not solvable. This implies that our sextic is not solvable by radicals (Hagedorn, 2000).

**Can we simplify the sextic using the formulation group?** A natural question which arises is whether one can simplify the sextic by a suitable choice of the formulation transformation. By simplify, we mean that (i) one of the sextic's coefficients would vanish, or (ii) its Galois group would change to a solvable subgroup of  $S_6$ . To answer this question, we expressed the coefficients  $c_i$ ,  $i \in [0, 6]$ , with  $p_3(\beta) = \sum_{i=0}^6 c_i \beta^i$ , as functions of the formulation transformation's parameters  $a$ ,  $b$  and  $s$ . We found that the coefficients change according to the following rule:

$$c_i \rightarrow \varepsilon^{10-i} c_i, \quad i \in [0, 6]. \quad (31)$$

Because  $\varepsilon \neq 0$ , we can answer part (i) of the question, and conclude that the formulation transformation cannot be used to cancel a power of  $\beta$ . We define a formulation-transformation-dependent version of the sextic using equation (31) as:

$$p'_3(\beta) \stackrel{\text{def}}{=} \sum_{i=0}^6 \varepsilon^{10-i} c_i \beta^i = \varepsilon^{10} p_3\left(\frac{\beta}{\varepsilon}\right).$$

$$\begin{aligned}
c_6 &= a^4 + c^4 + b^4 + 2a^2c^2 - 2a^2b^2 + 2c^2b^2 \\
c_5 &= 2a^4 + 2c^4 + 4b^4 + 4a^2c^2 - 6a^2b^2 + 6c^2b^2 + 4a^4d^2 + 4c^4d^2 + 2b^4d^2 + 8a^2c^2d^2 - 6a^2b^2d^2 + 6c^2b^2d^2 \\
c_4 &= a^4 - a^6 + c^4 + 6b^4 - c^6 + 2a^2c^2 - 6a^2b^2 - 3a^2c^4 - 3a^4c^2 - a^2b^4 + 2a^4b^2 + 6c^2b^2 - c^2b^4 - 2c^4b^2 \\
&\quad + 8a^4d^2 + 6a^4d^4 + 8c^4d^2 + 8b^4d^2 + 6c^4d^4 + b^4d^4 - b^6d^2 + 16a^2c^2d^2 - 18a^2b^2d^2 + 12a^2c^2d^4 \\
&\quad - 6a^2b^2d^4 + 2a^2b^4d^2 - a^4b^2d^2 + 18c^2b^2d^2 + 6c^2b^2d^4 - 2c^2b^4d^2 - c^4b^2d^2 - 2a^2c^2b^2d^2 \\
c_3 &= 4b^4 - 2a^2b^2 - 2a^2b^4 + 2a^4b^2 + 2c^2b^2 - 2c^2b^4 - 2c^4b^2 + 4a^4d^2 + 12a^4d^4 - 4a^6d^2 + 4a^4d^6 + 4c^4d^2 \\
&\quad + 12b^4d^2 + 12c^4d^4 - 4c^6d^2 + 4b^4d^4 - 4b^6d^2 + 4c^4d^6 + 8a^2c^2d^2 - 18a^2b^2d^2 + 24a^2c^2d^4 - 12a^2c^4d^2 \\
&\quad - 12a^4c^2d^2 - 18a^2b^2d^4 + 4a^2b^4d^2 + 4a^4b^2d^2 + 8a^2c^2d^6 - 2a^2b^2d^6 + 2a^2b^4d^4 - 2a^4b^2d^4 + 18c^2b^2d^2 \\
&\quad + 18c^2b^2d^4 - 8c^2b^4d^2 - 8c^4b^2d^2 + 2c^2b^2d^6 - 2c^2b^4d^4 - 2c^4b^2d^4 - 4a^2c^2b^2d^2 - 4a^2c^2b^2d^4 \\
c_2 &= b^4 - a^2b^4 - c^2b^4 + 6a^4d^4 + 8a^4d^6 - 6a^6d^4 + a^4d^8 + 8b^4d^2 + 6c^4d^4 + 6b^4d^4 - 6b^6d^2 + 8c^4d^6 - 6c^6d^4 \\
&\quad + c^4d^8 + a^2c^2b^4 - 6a^2b^2d^2 + 12a^2c^2d^4 - 18a^2b^2d^4 + 2a^2b^4d^2 + 5a^4b^2d^2 + 16a^2c^2d^6 - 18a^2c^4d^4 \\
&\quad - 18a^4c^2d^4 - 6a^2b^2d^6 + 5a^2b^4d^4 + a^2b^6d^2 + 2a^4b^2d^4 - 2a^4b^4d^2 + a^6b^2d^2 + 2a^2c^2d^8 - a^4b^2d^6 \\
&\quad + 6c^2b^2d^2 + 18c^2b^2d^4 - 10c^2b^4d^2 - 7c^4b^2d^2 + 6c^2b^2d^6 - 7c^2b^4d^4 - 10c^4b^2d^4 - c^4b^2d^6 - 2a^2c^2b^2d^2 \\
&\quad - 8a^2c^2b^2d^4 + a^2c^4b^2d^2 + 2a^4c^2b^2d^2 - 2a^2c^2b^2d^6 + a^2c^2b^4d^4 \\
c_1 &= 4a^4d^6 + 2a^4d^8 - 4a^6d^6 + 2b^4d^2 + 4b^4d^4 - 4b^6d^2 + 4c^4d^6 + 2c^4d^8 - 4c^6d^6 - 6a^2b^2d^4 + 8a^2c^2d^6 \\
&\quad - 6a^2b^2d^6 + 4a^2b^4d^4 + 2a^2b^6d^2 + 4a^4b^2d^4 - 2a^4b^4d^2 + 4a^2c^2d^8 - 12a^2c^4d^6 - 12a^4c^2d^6 - 2a^4b^4d^4 \\
&\quad + 2a^6b^2d^4 + 6c^2b^2d^4 - 4c^2b^4d^2 + 6c^2b^2d^6 - 8c^2b^4d^4 - 8c^4b^2d^4 - 4c^4b^2d^6 - 4a^2c^2b^2d^4 + 2a^2c^2b^4d^2 \\
&\quad - 4a^2c^2b^2d^6 + 2a^2c^2b^4d^4 + 2a^2c^4b^2d^4 + 4a^4c^2b^2d^4 \\
c_0 &= a^4d^8 - a^6d^8 + b^4d^4 - b^6d^2 + c^4d^8 - c^6d^8 - 2a^2b^2d^6 + a^2b^4d^4 + a^2b^6d^2 + 2a^2c^2d^8 + a^4b^2d^6 \\
&\quad - 2a^4b^4d^4 - 3a^2c^4d^8 - 3a^4c^2d^8 + a^6b^2d^6 + 2c^2b^2d^6 - 3c^2b^4d^4 - 3c^4b^2d^6 - 2a^2c^2b^2d^6 + 2a^2c^2b^4d^4 \\
&\quad + a^2c^4b^2d^6 + 2a^4c^2b^2d^6
\end{aligned}$$

Table 2: Coefficients of the sextic  $p_3$  under the chosen formulation transformation implying  $Z_{12} = 0$  and  $\sigma_1 = 1$ . We use  $a \stackrel{\text{def}}{=} Z_{11}$ ,  $b \stackrel{\text{def}}{=} Z_{22}$ ,  $c \stackrel{\text{def}}{=} Z_{21}$  and  $d \stackrel{\text{def}}{=} \sigma_2$ .

This means that the roots of  $p'_3$  are simply obtained from the roots of  $p_3$  by multiplying by a factor  $\varepsilon$ . We may thus say that the sextic is *rotation-invariant*, *reflection-invariant* and *scale-covariant*. It follows that changing  $\varepsilon$ , and thus acting on the formulation transformation, does not change the sextic's Galois group, which answers part (ii) of the question. In conclusion, one cannot simplify the sextic by the formulation group, which we thus use to simplify the quartic  $p_2$ .

**Analysis of the quartic, choice of the formulation transformation.** We found the quartic to be rotation-dependent: using the same principle as for the sextic, we found that the constant term could be cancelled by a suitable choice of the rotation. However, we also found that cancelling the top-right element of matrix  $Z$  allows us to find very simple formulas to solve the quartic, which are given in the next paragraph. This is achieved by choosing the formulation transformation with  $a = Z_{2,2}$ ,  $b = -Z_{1,2}$  and  $s = 1$ , and we thus have  $N^\top = [-S\mathbf{z}'_2 \ \mathbf{z}'_2]$  and  $\varepsilon = \|\mathbf{z}'_2\|_2$ . In addition, we divide the cost by  $\sigma_1^2$ , so that  $W$  becomes  $\text{diag}\left(1, \frac{\sigma_2}{\sigma_1}\right)$ . In practice, we simply perform the following updates:

$$Z \leftarrow \frac{1}{\sigma_1 \|\mathbf{z}'_2\|_2} \begin{bmatrix} \mathbf{z}'_2{}^\top S \mathbf{z}'_1 & 0 \\ \mathbf{z}'_2{}^\top \mathbf{z}'_1 & \|\mathbf{z}'_2\|_2^2 \end{bmatrix}, \quad \sigma_2 \leftarrow \frac{\sigma_2}{\sigma_1} \quad \text{and} \quad \sigma_1 \leftarrow 1.$$

Following equation (9), the estimated matrix  $\mathbf{B}$  will then be updated by  $\mathbf{B} \leftarrow \frac{1}{\epsilon} \mathbf{N}^\top \mathbf{B}$ , and the cost multiplied by  $\sigma_1^2$ .

**Factorizing and discarding the quartic's roots.** Under the chosen formulation transformation, we observe that the quartic  $p_2$  has the simple root  $-\sigma_2^2$  repeated once, and the two roots  $-\sigma_1 (\sigma_1 \pm Z_{1,1})$ :

$$p_2(\beta) = (\beta + \sigma_2^2)^2 (\beta + \sigma_1 (\sigma_1 + Z_{1,1})) (\beta + \sigma_1 (\sigma_1 - Z_{1,1})).$$

As already discussed, the simple root  $-\sigma_2^2$  corresponds to  $\text{rank}(\mathbf{M}_2) = 1$  and is not dealt with in this case as it corresponds to case 1.2.1. The two other roots can be discarded as well. In order to show why, we first evaluate the condition  $\det(\bar{\mathbf{E}} - \mathbf{I}) \leq 0$ , which is always met. This is because  $\det(\bar{\mathbf{E}} - \mathbf{I}) = -\frac{Z_{2,1}^2}{Z_{1,1}^2}$ , for both roots. This means that we always have one or two real solutions for  $\mathbf{q}$ . However, we can show that none of them meets the orthogonality constraint (25). We give the detailed derivation of this result for one of the four cases, with root  $-\sigma_1 (\sigma_1 + Z_{1,1})$  and  $\mathbf{q}_+$  from equation (28). The other three cases are derived similarly. Defining  $t \stackrel{\text{def}}{=} 1 - Z_{1,1} - \sigma_2^2$ , we have:

$$\mathbf{q}_+ = \begin{bmatrix} -Z_{1,1} Z_{2,1} t^2 (1 + \text{sign}(Z_{2,1} t)) \\ 0 \end{bmatrix},$$

and the orthogonality constraint (25) is simplified to:

$$Z_{1,1}^2 Z_{2,1}^3 t^3 (1 + \text{sign}(Z_{2,1} t))^2 = 0.$$

Because  $\mathbf{q} \neq 0$  is impossible we cannot fulfill the orthogonality constraint.

## 4.6 Cases 0.x.x

In that case, we have  $\mathbf{M} = 0$ , and thus:

$$\mathbf{M}_1 = \sigma_1^2 \mathbf{I} \quad \text{and} \quad \mathbf{M}_2 = \sigma_2^2 \mathbf{I}.$$

This implies  $\text{rank}(\mathbf{M}_1) = 2$ , as otherwise  $\sigma_1$  would vanish, and contradicts an hypothesis. Similarly, this implies  $\text{rank}(\mathbf{M}_2) = 2$ . Equations (16) and (17) thus imply:

$$\mathbf{b}_1 = \frac{\mathbf{z}'_1}{\sigma_1} \quad \text{and} \quad \mathbf{b}_2 = \frac{\mathbf{z}'_2}{\sigma_2},$$

which we rewrite as:

$$\mathbf{B} = \mathbf{Z}\mathbf{W}^{-1}.$$

This solution holds if it verifies equation (15). The general procedure of §4.5 handles this case.

## 4.7 Algebraic Procedure

We give our algebraic procedure for the orthographic camera in table 3. Because the value of the Lagrange multipliers is unknown, we do not know the case at hand in advance. We thus have to try all cases and keep the best estimate. The procedure first computes the best estimate of  $\mathbf{B}$  in  $\mathbb{O}_2$ , corresponding to cases 2.x.x. It then computes and tries all possible estimates in  $\mathbb{SS}_{22}$ , corresponding to cases 1.2.2 and 0.2.2, and keeps the best one. The roots of the sextic  $p_3$  are computed using Matlab’s `roots` function, which uses the eigenvalues of the companion matrix. All 6 roots are then considered in turn. Because numerical errors may induce non-zero imaginary parts in some roots, we do not try to discard the complex roots, but simply use their real parts. The algorithm then selects the root whose real part leads to the lowest cost. Similarly, because recovering  $\mathbf{q}$  involves the square root of a possibly negative number, we compose the square root with function  $\rho$ , with  $\rho(x) \stackrel{\text{def}}{=} \max(x, 0)$ . The estimates  $\mathbf{B}_{\pm}$  corresponding to spurious values of  $\beta$  may not lie in  $\mathbb{SS}_{22}$ . Instead of introducing the membership of  $\mathbb{SS}_{22}$  as a hard requirement, which because of numerical errors would require us to define a threshold and possibly discarding the optimal solution wrongly, we simply ‘project’ the estimates  $\mathbf{B}_{\pm}$  on  $\mathbb{SS}_{22}$  and select the one with the lowest cost. Projecting a non-zero  $(2 \times 2)$  matrix  $\mathbf{B}$  on  $\mathbb{SS}_{22}$  can be achieved by dividing it by its largest singular value as  $\mathbf{B} \leftarrow \frac{\mathbf{B}}{\mu_+(\mathbf{B})}$ . This is slightly different from the optimal projector (Cardoso and Zietak, 2015, theorem 5.5). The implementation returns the projected solutions, thus ensuring all results are exact members of  $\mathbb{SO}_3$  up to machine precision. Once the estimate of  $\mathbf{B}$  is obtained, it is finalized by forming the two rotations and the translation, and the cost is updated. If the output variable ‘singlesolution’ is true, this means that the solution was found in  $\mathbb{O}_2$  and that the two rotations and translations being returned match. If it is false however, the two solutions are different, but may still be numerically close.

## 5 Resection of the Weak-Perspective and Paraperspective Cameras

Resecting the weak-perspective camera is a special case of the paraperspective camera obtained with  $\mathbf{d} = \mathbf{0}$ . We derive the procedure for the paraperspective camera and then specialize it to the weak-perspective camera.

<p>Function <b>OPR</b>(<math>\mathbf{X} \in \mathbb{R}^{3 \times m}</math>, <math>\mathbf{Y} \in \mathbb{R}^{2 \times m}</math>)</p> <ul style="list-style-type: none"> <li>• Set <math>(\sigma_1, \sigma_2, \mathbf{Z}, \mathcal{O}, \mathbf{x}, \mathbf{y}, \mathbf{U}) \leftarrow \text{PrepareData}(\mathbf{X}, \mathbf{Y})</math>, <math>\mathbf{W} \leftarrow \text{diag}(\sigma_1, \sigma_2)</math></li> <li>• Recover <math>\mathbf{B}</math> in cases 2.x.x <ul style="list-style-type: none"> <li>– Set <math>s \leftarrow \text{sign}(\det(\mathbf{Z}))</math>, <math>a \leftarrow s\sigma_1 Z_{1,1} + \sigma_2 Z_{2,2}</math>, <math>b \leftarrow s\sigma_1 Z_{2,1} - \sigma_2 Z_{1,2}</math></li> <li>– Set <math>\begin{bmatrix} a \\ b \end{bmatrix} \leftarrow \frac{1}{\sqrt{a^2+b^2}} \begin{bmatrix} a \\ b \end{bmatrix}</math>, <math>\mathbf{B} \leftarrow \begin{bmatrix} sa &amp; -b \\ sb &amp; a \end{bmatrix}</math>, <math>\hat{\mathcal{O}} \leftarrow \ \mathbf{B}\mathbf{W} - \mathbf{Z}\ _{\mathcal{F}}^2</math>, singlesolution <math>\leftarrow</math> true</li> </ul> </li> <li>• Check if <math>\mathbf{B}</math> has a better solution in cases 1.2.2 and 0.2.2 <ul style="list-style-type: none"> <li>– Apply the formulation transformation: Set <math>\varepsilon \leftarrow \ \mathbf{z}'_2\ _2</math>, <math>\mathbf{N}^\top \leftarrow [-\mathbf{S}\mathbf{z}'_2 \ \mathbf{z}'_2]</math></li> <li><math>f \leftarrow \sigma_1^2</math>, <math>\mathbf{Z} \leftarrow \frac{1}{\varepsilon\sigma_1} \begin{bmatrix} \mathbf{z}'_2{}^\top \mathbf{S}\mathbf{z}'_1 &amp; 0 \\ \mathbf{z}'_2{}^\top \mathbf{z}'_1 &amp; \varepsilon^2 \end{bmatrix}</math>, <math>\sigma_2 \leftarrow \frac{\sigma_2}{\sigma_1}</math>, <math>\sigma_1 \leftarrow 1</math>, <math>\mathbf{W} \leftarrow \text{diag}(1, \sigma_2)</math></li> <li>– Form the sextic <math>p_3</math> from table 2, store the real part of the roots in <math>\beta \in \mathbb{R}^6</math></li> <li>– For <math>\beta \in \beta</math> <ul style="list-style-type: none"> <li>* If <math>\beta = \sigma_1^2</math> or <math>\beta = \sigma_2^2</math>, continue</li> <li>* Set <math>\bar{\mathbf{E}} \leftarrow \mathbf{Z}\mathbf{W}(\mathbf{W}^2 + \beta\mathbf{I})^{-2}\mathbf{W}\mathbf{Z}^\top</math></li> <li>* Set <math>\mathbf{q}_\pm \leftarrow \left( \mathbf{S}(\bar{\mathbf{E}} - \mathbf{I}) \pm \sqrt{\rho(\text{tr}(\bar{\mathbf{E}}) - \det(\bar{\mathbf{E}}) - 1)\mathbf{I}} \right) \mathbf{e}_1</math></li> <li>* Set <math>\mathbf{q}_\pm \leftarrow \frac{\mathbf{q}_\pm}{\ \mathbf{q}_\pm\ _2}</math></li> <li>* Set <math>\mathbf{M}_\pm \leftarrow \beta\mathbf{q}_\pm\mathbf{q}_\pm^\top</math></li> <li>* Set <math>\mathbf{B}_\pm \leftarrow \left[ \sigma_1(\mathbf{M}_\pm + \sigma_1^2\mathbf{I})^{-1}\mathbf{z}'_1 \ \sigma_2(\mathbf{M}_\pm + \sigma_2^2\mathbf{I})^{-1}\mathbf{z}'_2 \right]</math></li> <li>* Set <math>\mathbf{B}_\pm \leftarrow \frac{\mathbf{B}_\pm}{\mu_+(\mathbf{B}_\pm)}</math></li> <li>* Set <math>\mathcal{O}'_\pm \leftarrow f\ \mathbf{B}_\pm\mathbf{W} - \mathbf{Z}\ _{\mathcal{F}}^2</math></li> <li>* If <math>\mathcal{O}'_\pm &lt; \hat{\mathcal{O}}</math>, Set <math>\hat{\mathcal{O}} \leftarrow \mathcal{O}'_\pm</math>, <math>\mathbf{B} \leftarrow \frac{1}{\varepsilon}\mathbf{N}^\top\mathbf{B}_\pm</math>, singlesolution <math>\leftarrow</math> false</li> </ul> </li> </ul> </li> <li>• Finalize the estimate <ul style="list-style-type: none"> <li>– If singlesolution, Set <math>\mathbf{R}_\pm \leftarrow \det(\mathbf{U}) \begin{bmatrix} \mathbf{B} &amp; \mathbf{0} \\ \mathbf{0}^\top &amp; s \end{bmatrix} \mathbf{U}^\top</math>, <math>\mathbf{t}_\pm \leftarrow \mathbf{y} - \Pi_0\mathbf{R}_\pm\mathbf{x}</math></li> <li>Else <ul style="list-style-type: none"> <li>* Set <math>\bar{\mathbf{Q}} \leftarrow [\mathbf{B} \ \text{rank}_1(\mathbf{I} - \mathbf{B}\mathbf{B}^\top)]</math></li> <li>* Form <math>\mathbf{Q}_+</math> from <math>\bar{\mathbf{Q}}</math> and the cross-product of its two rows</li> <li>* Set <math>\mathbf{Q}_- \leftarrow \mathbf{Q}_+ \odot \begin{bmatrix} 1 &amp; 1 &amp; -1 \\ 1 &amp; 1 &amp; -1 \\ -1 &amp; -1 &amp; 1 \end{bmatrix}</math></li> <li>* Set <math>\mathbf{R}_+ \leftarrow \det(\mathbf{U})\mathbf{Q}_+\mathbf{U}^\top</math> and <math>\mathbf{R}_- \leftarrow \det(\mathbf{U})\mathbf{Q}_-\mathbf{U}^\top</math></li> <li>* Set <math>\mathbf{t}_+ \leftarrow \mathbf{y} - \Pi_0\mathbf{R}_+\mathbf{x}</math> and <math>\mathbf{t}_- \leftarrow \mathbf{y} - \Pi_0\mathbf{R}_-\mathbf{x}</math></li> </ul> </li> </ul> </li> <li>• Update the cost, Set <math>\mathcal{O} \leftarrow \mathcal{O} + \hat{\mathcal{O}}</math></li> </ul> <p>Output <math>\mathbf{R}_+, \mathbf{R}_- \in \mathbb{S}\mathbb{O}_3</math>, <math>\mathbf{t}_+, \mathbf{t}_- \in \mathbb{R}^2</math>, <math>\mathcal{O}</math>, singlesolution</p>
---

Table 3: **Algebraic procedure solving orthographic planar resection.** The procedure may return one or two rotation and translation pairs and a single cost. This is because the problem may have one or two solutions in general. In the latter case, the two solutions become equal for  $\mathbf{B} \in \mathbb{O}_2$ . We have by construction that the involved denominators are never zero ( $a^2 + b^2 \neq 0$ ,  $\varepsilon \neq 0$ ,  $\sigma_1 \neq 0$  and  $\|\mathbf{q}_\pm\|_2 \neq 0$ ). We recall that  $\rho(x) \stackrel{\text{def}}{=} \max(x, 0)$ .

## 5.1 Overview, Relationship to IPPE and Optimality

After preprocessing the data points to obtain the second canonical formulation (6), we follow the solution strategy we developed for IPPE (Collins and Bartoli, 2014). This is as follows. First, matrix  $\mathbf{B}$  is estimated, which represents the optimal affine transform between transformed point correspondences. It is then decomposed into the scale  $\gamma$  and the rotation  $\mathbf{R}$  from  $\mathbf{B} = \gamma \Pi_{\mathbf{d}} \mathbf{R} \mathbf{U}' \Pi_{\mathbf{0}}^{\top}$ . The translation is then computed from equation (3).

To solve the decomposition, IPPE uses a coordinate transform which simplifies the problem by converting the paraperspective into a weak-perspective resection problem. The coordinate transform is a rotation  $\mathbf{R}_{\mathbf{d}}$  of the camera to align its optical axis with the camera's projection direction given by  $\mathbf{d}$ . Once solved, the coordinate transform is undone in a final step. In IPPE,  $\mathbf{R}_{\mathbf{d}}$  is computed with Rodrigues's formula. If the camera's optical axis is already aligned with its projection direction, Rodrigues's formula fails due to a division by zero. This could be handled by detecting the event, for instance by thresholding  $\|\mathbf{d}\|_2$ , and setting  $\mathbf{R}_{\mathbf{d}} = \mathbf{I}$ . We have however developed a better approach to compute  $\mathbf{R}_{\mathbf{d}}$  that handles this situation seamlessly and also requires fewer floating point operations.

The overall approach's optimality is proved as follows. First, for any  $\mathbf{B} \in \mathbb{R}^{2 \times 2}$ ,  $\mathbf{B} \neq \mathbf{0}$ , there exist a decomposition which is unique in the scale  $\gamma$  and two-way ambiguous in the rotation  $\mathbf{R}$  (Collins and Bartoli, 2014). Second, the optimal affine transform  $\mathbf{B}$  is also the optimal solution of the resection problem. This is shown in the next paragraph.

## 5.2 Computing $\mathbf{B} \in GL_2(\mathbb{R})$

The second canonical formulation (6) involves finding matrix  $\mathbf{B} \in \bar{\mathbb{M}}_{\text{PP}}$ . We establish the following result:

$$\left( \arg \min_{\mathbf{B} \in \bar{\mathbb{M}}_{\text{PP}}} \mathcal{O}_6(\mathbf{B}) \right) = \mathbf{Z} \mathbf{W}^{-1}.$$

Our proof is based on showing that  $\bar{\mathbb{M}}_{\text{PP}} \equiv GL_2(\mathbb{R})$ . This relationship holds for any  $\mathbf{d} \in \mathbb{R}^2$  specifying  $\bar{\mathbb{M}}_{\text{PP}}$ . The proof forms the basis of the algebraic procedure given in the next section. Proving the forward implication  $\mathbb{M}_{\text{PP}} \subset GL_2(\mathbb{R})$  is straightforward. Let  $\mathbf{B} \in \bar{\mathbb{M}}_{\text{PP}}$ . By definition,  $\mathbf{B}$  is the leading submatrix of a paraperspective projection. It is thus full-rank and therefore  $\mathbf{B} \in GL_2(\mathbb{R})$ . Proving the backward implication  $GL_2(\mathbb{R}) \subset \mathbb{M}_{\text{PP}}$  is very simple, by invoking a result from (Collins and Bartoli, 2014). Let  $\mathbf{F} \in GL_2(\mathbb{R})$ . We can always find  $\gamma \in \mathbb{R}^+$  and  $\mathbf{Q} \in \mathbb{S}\mathbb{O}_3$  such that  $\mathbf{F} = \gamma \Pi_{\mathbf{d}} \mathbf{Q} \Pi_{\mathbf{0}}^{\top}$ . This stems from theorems 3 and 4 in (Collins and Bartoli, 2014). Given that matrix  $\mathbf{B} \in \mathbb{M}_{\text{PP}}$  and  $\mathbb{M}_{\text{PP}} \equiv GL_2(\mathbb{R})$ , the cost in the second canonical formulation vanishes by setting  $\mathbf{B} = \mathbf{Z} \mathbf{W}^{-1} \in GL_2(\mathbb{R})$ . A direct consequence is that the overall cost of the

estimate is  $\mathcal{O}$  from equation (5). This is obtained because  $\mathcal{O}_6(\mathbf{Z}\mathbf{W}^{-1}) = 0$ , by applying equations (7) and (5).

### 5.3 Computing $\gamma \in \mathbb{R}^+$ and $\mathbf{R} \in \mathbb{S}\mathbb{O}_3$

We wish to recover  $\gamma$  and  $\mathbf{R}$ , forming the parameters of  $\mathbf{P} = \gamma\Pi_{\mathbf{d}}\mathbf{R} \in \mathbb{M}_{\text{PP}}$ , from  $\mathbf{B}$ . Recall that we defined  $\mathbf{A} = \mathbf{P}\mathbf{U}' \in \mathbb{M}_{\text{PP}}$  and  $\mathbf{B} = \mathbf{A}\Pi_{\mathbf{0}} \in \bar{\mathbb{M}}_{\text{PP}}$ . We proceed in two steps. First, we recover  $\gamma$  and  $\mathbf{u} \in \mathbb{R}^2$ , with  $\gamma\mathbf{u}$  defined as the third column of  $\mathbf{A}$ , from  $\mathbf{B}$ . Second, we recover  $\mathbf{R}$  from  $\mathbf{A}$  and  $\gamma$ .

#### 5.3.1 Computing $\gamma \in \mathbb{R}^+$ and $\mathbf{u} \in \mathbb{R}^2$

We wish to compute  $\gamma$  and complete  $\mathbf{A}$  as  $\mathbf{A} = [\mathbf{B} \ \gamma\mathbf{u}]$ . Indeed, given that  $\mathbf{B} = \mathbf{A}\Pi_{\mathbf{0}}$  we have:

$$[\mathbf{B} \ \gamma\mathbf{u}] = \gamma\Pi_{\mathbf{d}}\mathbf{R}\mathbf{U}'. \quad (32)$$

Multiplying each side of this equation to the right by its transpose gives:

$$\mathbf{B}\mathbf{B}^\top + \gamma^2\mathbf{u}\mathbf{u}^\top = \gamma^2(\mathbf{I} + \mathbf{d}\mathbf{d}^\top).$$

This is a rank-1 equation of type 4, whose solution is given in appendix E. Proposition 2 with  $z = \gamma^2$ ,  $\mathbf{K} = \mathbf{B}\mathbf{B}^\top$  and  $\mathbf{G} = \mathbf{I} + \mathbf{d}\mathbf{d}^\top$  shows that  $\gamma$  has a unique solution, and that  $\mathbf{u}$  has two, defined by a simple sign change, which we represent by  $s \in \{-1, 1\}$ . Computing the solution requires the Cholesky decomposition  $\mathbf{G} = \mathbf{V}\mathbf{V}^\top$ , which is given analytically in appendix B.

#### 5.3.2 Computing $\mathbf{R} \in \mathbb{S}\mathbb{O}_3$

We first introduce  $\mathbf{R}_{\mathbf{d}} \in \mathbb{S}\mathbb{O}_3$  defined such that there exist  $\mathbf{H} \in GL_2(\mathbb{R})$  such that  $\mathbf{H}\Pi_{\mathbf{0}} \stackrel{\text{def}}{=} \Pi_{\mathbf{d}}\mathbf{R}_{\mathbf{d}}$ . In other words, we rotate the camera to align the projection direction with the  $Z$ -axis. We give an efficient way of choosing  $\mathbf{R}_{\mathbf{d}}$  from  $\mathbf{d}$  in appendix C, which involves the Cholesky factor  $\mathbf{V}$  and leads to  $\mathbf{H} = \mathbf{V}$ . By introducing  $\mathbf{R}_{\mathbf{d}}\mathbf{R}_{\mathbf{d}}^\top$  in the right-hand side of equation (32), we obtain:

$$[\mathbf{B} \ s\gamma\mathbf{u}] = \gamma\mathbf{H}\Pi_{\mathbf{0}}\mathbf{R}_{\mathbf{d}}^\top\mathbf{R}\mathbf{U}'. \quad (33)$$

The fact that  $s \in \{-1, 1\}$  is undetermined leads to two solutions. Defining  $\mathbf{Q} \stackrel{\text{def}}{=} \mathbf{R}_{\mathbf{d}}^\top\mathbf{R}\mathbf{U}' \in \mathbb{S}\mathbb{O}_3$ , we have:

$$\mathbf{H}^{-1} \begin{bmatrix} 1 \\ \frac{1}{\gamma}\mathbf{B} \ s\mathbf{u} \end{bmatrix} = \Pi_{\mathbf{0}}\mathbf{Q}_s,$$

where the index in  $\mathbf{Q}_s$  emphasizes the dependency on  $s$ . This equation directly gives the two top rows of  $\mathbf{Q}_s$ . The bottom row  $\mathbf{q}_s$  is obtained as their cross-product. With  $\mathbf{B} = [\mathbf{b}_1 \ \mathbf{b}_2]$ , this is given by:

$$\mathbf{q}_s^\top = \det(\mathbf{H}^{-1}) \frac{1}{\gamma} \begin{bmatrix} s \det([\mathbf{b}_2 \ \mathbf{u}]) & s \det([\mathbf{u} \ \mathbf{b}_1]) & \frac{1}{\gamma} \det(\mathbf{B}) \end{bmatrix}.$$

Recalling that  $\mathbf{U}' = \det(\mathbf{U})\mathbf{U}$ , we finally obtain two solutions for  $\mathbf{R}$ , depending on which value of  $s$  is used in assembling  $\mathbf{Q}_s$ , as:

$$\mathbf{R}_s = \det(\mathbf{U})\mathbf{R}_d\mathbf{Q}_s\mathbf{U}^\top. \quad (34)$$

#### 5.4 Computing $\mathbf{t} \in \mathbb{R}^3$

We compute the translation from the rotation using equation (3). We thus obtain a different translation  $\mathbf{t}_s$  for each of the two rotation solutions  $\mathbf{R}_s$ ,  $s \in \{-1, 1\}$ , as:

$$\begin{aligned} \mathbf{t}_s &= \mathbf{y} - \gamma\Pi_d\mathbf{R}_s\mathbf{x} \\ &= \mathbf{y} - \gamma\det(\mathbf{U})\Pi_d\mathbf{R}_d\mathbf{Q}_s\mathbf{U}^\top\mathbf{x} \\ &= \mathbf{y} - \det(\mathbf{U})[\mathbf{B} \ s\gamma\mathbf{u}]\mathbf{U}^\top\mathbf{x}. \end{aligned}$$

The second equation is obtained by replacing  $\mathbf{R}_s$  by its expression (34). The third equation is obtained by substituting the definition  $\Pi_d\mathbf{R}_d = \mathbb{H}\Pi_0$  of  $\mathbb{H}$ , and by introducing equation (33).

#### 5.5 Algebraic Procedures

We give our algebraic procedure for the paraperspective camera in table 4. The weak-perspective case is obtained from the paraperspective case by setting  $\mathbf{d} = \mathbf{0}$ . This introduces the following simplifications in the procedure: (i) line two is removed, (ii)  $\tilde{\mathbf{B}}$  becomes  $\mathbf{B}$  and  $\tilde{\mathbf{u}}$  becomes  $\mathbf{u}$ , and (iii)  $\mathbf{R}_d$  is the identity matrix and can be ignored in the products computing  $\mathbf{R}_\pm$ . We give the simplified procedure in appendix H.

## 6 Experimental Results

Our experimental results compare the proposed algebraic procedures with alternative solutions based on polynomial global optimization and gradient-based refinement, with the perspective camera model in some experiments, with uncalibrated versions of these in some other experiments and even with bundle adjustment in SfM experiments. An emphasis is given to the orthographic model because it forms the most complex resection problem and algebraic procedure we have derived. It is therefore important to understand if using

<p>Function <b>PPR</b>(<math>\mathbf{X} \in \mathbb{R}^{3 \times m}</math>, <math>\mathbf{Y} \in \mathbb{R}^{2 \times m}</math>, <math>\mathbf{d} \in \mathbb{R}^2</math>)</p> <ul style="list-style-type: none"> <li>• Set <math>(\sigma_1, \sigma_2, \mathbf{Z}, \mathcal{O}, \mathbf{x}, \mathbf{y}, \mathbf{U}) \leftarrow \text{PrepareData}(\mathbf{X}, \mathbf{Y})</math>, <math>\mathbf{B} \leftarrow \mathbf{Z} \text{diag}(\sigma_1^{-1}, \sigma_2^{-1})</math></li> <li>• Set <math>(\mathbf{H}, \mathbf{H}^{-1}, \mathbf{R}_d) \leftarrow \text{CholeskyAndProjectionAxisToZAxis}(\mathbf{d})</math>, <math>\tilde{\mathbf{B}} \leftarrow \mathbf{H}^{-1}\mathbf{B}</math></li> <li>• Compute <math>\gamma, \mathbf{u}</math> by solving the rank-1 equation of type 4</li> </ul> <p>Set <math>\mathbf{A} \leftarrow \tilde{\mathbf{B}}\tilde{\mathbf{B}}^\top</math>, <math>\gamma^2 \leftarrow \lambda_1(\mathbf{A})</math>, <math>\tilde{\mathbf{u}} \leftarrow \text{rank}_1\left(\mathbf{I} - \frac{1}{\gamma^2}\mathbf{A}\right)</math>, <math>\mathbf{u} \leftarrow \mathbf{H}\tilde{\mathbf{u}}</math></p> <ul style="list-style-type: none"> <li>• Set <math>\bar{\mathbf{Q}} \leftarrow \begin{bmatrix} \frac{1}{\gamma}\tilde{\mathbf{B}} &amp; \tilde{\mathbf{u}} \end{bmatrix}</math></li> <li>• Form <math>\mathbf{Q}_+</math> from <math>\bar{\mathbf{Q}}</math> and the cross-product of its two rows</li> <li>• Set <math>\mathbf{Q}_- \leftarrow \mathbf{Q}_+ \odot \begin{bmatrix} 1 &amp; 1 &amp; -1 \\ 1 &amp; 1 &amp; -1 \\ -1 &amp; -1 &amp; 1 \end{bmatrix}</math></li> <li>• Set <math>\mathbf{R}_+ \leftarrow \det(\mathbf{U})\mathbf{R}_d\mathbf{Q}_+\mathbf{U}^\top</math> and <math>\mathbf{R}_- \leftarrow \det(\mathbf{U})\mathbf{R}_d\mathbf{Q}_-\mathbf{U}^\top</math></li> <li>• Set <math>\mathbf{t}_+ \leftarrow \mathbf{y} - \det(\mathbf{U})[\mathbf{B} \ \gamma\mathbf{u}]\mathbf{U}^\top \mathbf{x}</math> and <math>\mathbf{t}_- \leftarrow \mathbf{y} - \det(\mathbf{U})[\mathbf{B} \ -\gamma\mathbf{u}]\mathbf{U}^\top \mathbf{x}</math></li> </ul> <p>Output <math>\gamma \in \mathbb{R}^+</math>, <math>\mathbf{R}_+, \mathbf{R}_- \in \mathbb{SO}_3</math>, <math>\mathbf{t}_+, \mathbf{t}_- \in \mathbb{R}^2</math>, <math>\mathcal{O} \in \mathbb{R}^2</math></p>
---

Table 4: **Algebraic procedure solving paraperspective planar resection.** The procedure always returns two rotation and translation pairs and a single cost. This is because the problem has two solutions in general. These two solutions become equal for  $\mathbf{u} = \mathbf{0}$ .

this procedure is worthwhile in practice, in particular given that resection with the weak-perspective model turned out to be much simpler to solve and implement, and that the weak-perspective model can be directly used in place of the orthographic model in almost all use cases. This is not the case for the paraperspective and perspective models however, which both require the camera’s perspective intrinsics to be known in order to compute pose. We use the rotation error extensively, which is measured in degrees, denoted ‘deg’. It is computed as the smallest angle required to align the estimated to the groundtruth pose. We also use the reprojection error, given in number of pixels, denoted ‘px’.

## 6.1 Compared Methods

Our algebraic procedures are abbreviated as ALG, followed by two letters indicating the camera model, OR, WP or PP. The alternative global polynomial optimization methods are based on a direct formulation of the second canonical formulation (6) in Gloptipoly (Henrion et al., 2009). Three types of implementation are introduced in §6.2.2 and tested. These methods are abbreviated as POLY, followed by the two camera model letters. We also used the best result in terms of distance to groundtruth among ALG and POLY to initialize direct iterative minimization with Levenberg-Marquardt using Matlab’s optimization toolbox. This is abbreviated as REF, followed by the two camera model letters. We consider a method to be calibrated by default, which means that the intrinsic parameters of the model it involves are known, as defined in §2.2.

On the contrary, an uncalibrated method estimates some of its intrinsic parameters from the data. More specifically, the intrinsics to be estimated for each model and the estimation methods are discussed in §6.2.3, which contains all our experiments done in an uncalibrated setting. Calibrated and uncalibrated methods are indicated by CAL and UNC when required.

## 6.2 Synthetic Data

We provide four batches of experiments with synthetic data, each with a specific purpose: comparing the orthographic and weak-perspective cameras, comparing our algebraic procedure with alternative methods, assessing performance under perspective projection in the uncalibrated setting, and assessing performance in orthographic SfM.

### 6.2.1 Comparison of the Orthographic and Weak-Perspective Cameras

The goal of this experiment is to assess the benefit of using the orthographic camera to solve resection compared to the weak-perspective camera. In other words, if knowing the camera’s scale factor a priori has a real benefit. We thus simulated a set of  $m$  coplanar points observed by an orthographic camera. We work in px units directly. The points are chosen so that they have an average distance to their centroid of  $50\sqrt{2} \approx 70.7$  px. The points are projected to form an image and gaussian noise is added, whose standard deviation varies between 0 to 2.5 px. The experiment is repeated 5,000 times for each noise level and the average error is reported. For each trial, the coplanar point set and camera pose are randomly drawn. The generated camera pose has a tilt angle of 80 degrees at most and no translation, as this does not influence the result. The scale of the generated points can be understood from their convex hull, and more precisely by the square root of their convex hull’s area, which we measured to be approximately 54.7 px with standard deviation 11.4 px. The highest noise level of 2.5 px thus represents a relative noise of approximately 4.6 %. We run our algebraic procedure for the orthographic and weak-perspective cameras, respectively denoted ALGOR and ALGWP. The next experiment shows very clearly that there is no need to report the results of alternative orthographic resection methods in this experiment. The results of rotation error in degrees for  $m \in \{3, 5, 20\}$  point sets are given in figure 1.

We observe that, as expected, the error increases linearly with the level of noise and decreases as the number of points increases. The orthographic camera consistently performs significantly better than the weak-perspective camera. Intuitively, this is explained by the fact that the orthographic camera has fewer parameters than the weak-perspective camera, namely 5 against 6 parameters. It is thus better constrained by the data, which allows it to reduce the effect of noise on the estimate with an improved efficiency. This

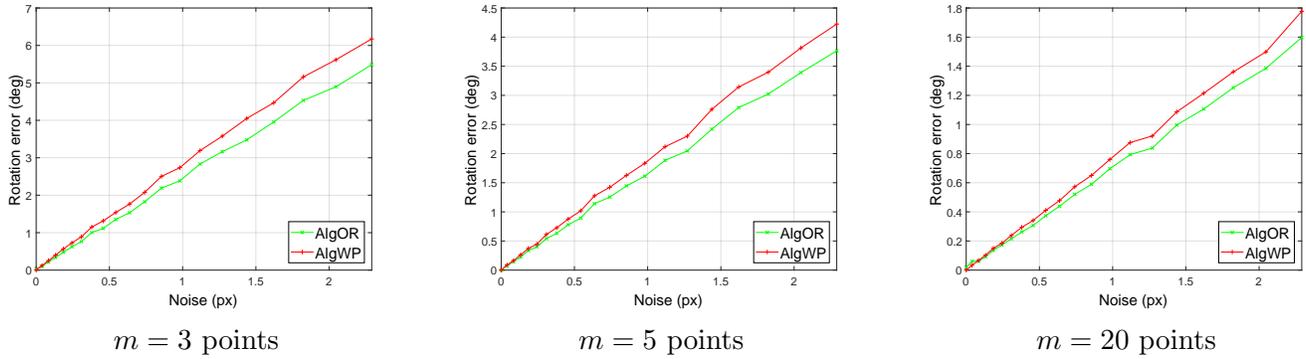


Figure 1: Results obtained from synthetic data to compare the orthographic and weak-perspective cameras using the proposed algebraic procedures.

also means that knowing the scale factor benefits to the rotation estimate. Interestingly, we do not observe a crossover point between the two methods as the number of points increases. This means that even for stable configurations with low noise or large point sets the orthographic camera still performs noticeably better than the weak-perspective camera.

### 6.2.2 Comparison of the Orthographic Algebraic Procedure with Alternative Methods

The goal of this experiment is to assess the benefit of the algebraic procedure for the orthographic camera compared to solutions based on polynomial global optimization and gradient-based nonlinear refinement. The latter is denoted as REFOR. It uses Matlab’s `lsqnonlin` procedure with ‘FunctionTolerance’ and ‘StepTolerance’ set to  $10^{-8}$ . As for polynomial global optimization, we used Gloptipoly and tested three different ways of handling the rotation, which is where the polynomial constraints stem in this problem. The first implementation, POLYOR1, uses a unit quaternion parameterization. The second implementation, POLYOR2, uses the quartic constraints (15). The third implementation, POLYOR3, uses an SVD-based parameterization of the leading  $(2 \times 2)$  block of the orthographic projection matrix. More specifically, recall that in the second canonical formulation (11) for the orthographic camera, the unknown is matrix  $\mathbf{B} \in \mathbb{S}\mathbb{S}_{22}$ . Let the SVD of  $\mathbf{B}$  be  $\mathbf{B} = \mathbf{U}_\mathbf{B} \mathbf{\Sigma}_\mathbf{B} \mathbf{V}_\mathbf{B}^\top$ . We know from equation (12) that  $\mathbf{\Sigma}_\mathbf{B} = \text{diag}(1, \zeta)$  with  $0 \leq \zeta \leq 1$ . We can then represent matrix  $\mathbf{B}$  by  $\mathbf{U}, \mathbf{V} \in \mathbb{O}_2$  and  $\zeta \in [0, 1]$  or, equivalently, by  $\mathbf{U}, \mathbf{V} \in \mathbb{S}\mathbb{O}_2$  and  $|\zeta| \in [0, 1]$ . Concretely, we parameterize  $\mathbf{U} \in \mathbb{S}\mathbb{O}_2$  by  $u_1, u_2 \in \mathbb{R}$  as  $\mathbf{U} = \begin{bmatrix} u_1 & -u_2 \\ u_2 & u_1 \end{bmatrix}$  with  $u_1^2 + u_2^2 = 1$  and  $\mathbf{V} \in \mathbb{S}\mathbb{O}_2$  likewise. We also add the constraint  $u_1 \geq 0$  to raise the common sign ambiguity between  $\mathbf{U}$  and  $\mathbf{V}$ . Our final parameterization thus has five parameters, two quadratic equality constraints and three linear inequality constraints. In the three polynomial optimization methods, the cost polynomial constructed from the cost function from formulation (11) was normalized by dividing its coefficients by the norm of its coefficient vector. The simulated data are similar to the first experiment described in §6.2.1 with a noise standard

deviation of 1.0 px.

Running the experiments, we found that the methods based on polynomial optimization failed in rare occasions for numerical reasons. More specifically, we found that this happens for problem instances where the cost polynomial’s coefficients were unbalanced and caused either the failure of SeDuMi to solve the SDP Gloptipoly relies on or the failure of Gloptipoly to extract the solutions from the measure vector. We use ‘unbalanced’ to mean that there is a significant discrepancy between the orders of magnitude of the polynomial’s coefficients. We found that SeDuMi’s precision parameter could be left at its default value  $1 \times 10^{-8}$ , as setting it to 0, meaning that the iterations continue until no improvement is made, did not change our results. However, Gloptipoly’s relaxation order was found to be a very important parameter, as it trades-off success rate and speed. We thus ran Gloptipoly with relaxation order 1, which is very fast, and increased the relaxation order in case of failure, with a maximum value of 5. In order to quantify the performance of each parameterization, we chose to run three batches of experiments and only keep the trials for which the tested method completed properly. For each batch we report the success rate of the tested polynomial optimization based method. We also report, for all three methods tested in a batch, the absolute reprojection error difference and the reprojection error rank. The former is the absolute difference between a method’s reprojection error and the best reprojection error over all methods. The latter is the rank of a method according to its reprojection error. The rank varies between 1, which is best, and 3, which is worst. We chose to use the reprojection error because this is the criterion which we minimize in the resection problem (1). The stability of a method is thus directly indicated by monitoring how small it renders the reprojection error. Our results are given in table 5. Because the size of the imaged point set is 54.7 px on average, we may multiply the value of the reprojection error in px by a factor of 2 to obtain a rough estimate of the relative error in %.

We first make the following two general comments. First, all methods are in good agreement in terms of absolute reprojection error. This agreement is especially strong for ALGOR and REFOR. Second, all methods have good accuracy. More specifically, ALGOR and REFOR outperform the polynomial optimization based methods consistently by several orders of magnitudes, though the latter still give acceptable results, with an absolute reprojection error difference consistently lower than or equal to  $10^{-7}$ .

We observe differences between the three implementations of global polynomial optimization. In terms of absolute reprojection error difference, the SVD-based implementation POLYOR3 gives the best results, followed by the quaternion-based implementation POLYOR1 and the constraint-based implementation POLYOR2. The difference with the other two methods is nonetheless always mild, being approximately  $10^{-10}$ ,  $10^{-8}$  and  $10^{-7}$  respectively for POLYOR3, POLYOR1 and POLYOR2. In terms of reprojection error

Number of points	3	4	5	10	20
<b>Absolute reprojection error difference in px</b>					
POLYOR1	$4.61 \times 10^{-8}$	$2.31 \times 10^{-8}$	$1.99 \times 10^{-8}$	$1.50 \times 10^{-8}$	$2.37 \times 10^{-8}$
ALGOR	0	$8.05 \times 10^{-16}$	0	0	0
REFOR	$1.11 \times 10^{-15}$	0	0	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$
<b>Reprojection error rank</b>					
POLYOR1	3.00	3.00	3.00	3.00	3.00
ALGOR	1.47	1.50	1.48	1.47	1.46
REFOR	1.53	1.50	1.52	1.53	1.54
<b>Success rate in %</b>					
POLYOR1	100	100	99.9	100	100

Number of points	3	4	5	10	20
<b>Absolute reprojection error difference in px</b>					
POLYOR2	$2.14 \times 10^{-7}$	$1.11 \times 10^{-7}$	$1.17 \times 10^{-7}$	$9.84 \times 10^{-8}$	$1.06 \times 10^{-7}$
ALGOR	$1.91 \times 10^{-15}$	$1.78 \times 10^{-15}$	$1.77 \times 10^{-15}$	$8.88 \times 10^{-16}$	$1.77 \times 10^{-15}$
REFOR	$2.50 \times 10^{-15}$	0	$4.44 \times 10^{-16}$	$1.78 \times 10^{-15}$	$2.66 \times 10^{-15}$
<b>Reprojection error rank</b>					
POLYOR2	2.90	2.91	2.89	2.86	2.81
ALGOR	1.54	1.52	1.54	1.54	1.55
REFOR	1.56	1.57	1.57	1.60	1.64
<b>Success rate in %</b>					
POLYOR2	99.7	100	100	100	98.5

Number of points	3	4	5	10	20
<b>Absolute reprojection error difference in px</b>					
POLYOR3	$1.75 \times 10^{-9}$	$4.87 \times 10^{-10}$	$8.18 \times 10^{-10}$	$4.29 \times 10^{-10}$	$5.48 \times 10^{-10}$
ALGOR	$5.78 \times 10^{-14}$	$2.59 \times 10^{-14}$	$1.86 \times 10^{-14}$	$1.15 \times 10^{-14}$	$9.77 \times 10^{-15}$
REFOR	$7.19 \times 10^{-14}$	$4.77 \times 10^{-14}$	$3.19 \times 10^{-14}$	$4.31 \times 10^{-14}$	$1.48 \times 10^{-11}$
<b>Reprojection error rank</b>					
POLYOR3	2.13	2.14	2.15	2.23	2.32
ALGOR	1.92	1.90	1.91	1.86	1.79
REFOR	1.95	1.96	1.94	1.91	1.89
<b>Success rate in %</b>					
POLYOR3	99.5	99.6	99.5	99.6	99.7

Table 5: Results obtained from synthetic data to compare the proposed algebraic procedure for the orthographic camera with three implementations of polynomial optimization and nonlinear refinement. We use 0 for numbers lower than  $10^{-16}$ . POLYOR1 uses a unit quaternion parameterization, POLYOR2 uses quartic constraints and POLYOR3 uses an SVD-based parameterization.

rank, POLYOR1 is consistently last while POLYOR2 and POLYOR3 are generally last but in a noticeable number of cases. In contrast ALGOR and REFOR are first in very similar numbers of times. This may be explained by the fact that Gloptipoly uses a relaxation of the original constraints.

Importantly, in terms of success rate, the quaternion-based POLYOR1 is the best method, with a 100 % success rate in almost all tested cases. Both the SVD-based POLYOR3 and the constraint-based POLYOR2 follow with almost 100 % successes. More specifically, POLYOR2 reaches 100 %, except for  $m = 3$  and  $m = 20$  points, while POLYOR3 never reaches 100 %, but is always greater than or equal to 99.5 %. This may be explained by the fact that POLYOR3 has a singularity when  $\xi \rightarrow 1$ , as  $\mathbf{U}, \mathbf{V}$  are then not unique, contrarily to the general case. The success rate of ALGOR and REFOR is not shown in the result table as it is always 100 %. Overall, POLYOR1 then appears as the best-behaved polynomial method.

### 6.2.3 Performance with Perspective Projection and Uncalibrated Images

The goal of this experiment is to assess which camera model performs best in realistic conditions close to affine projection. We used the same simulation setup to simulate coplanar points as in the first experiment described in §6.2.1. However, we simulated a perspective camera with a long focal length of 4,000 px and an image of  $500 \times 500$  px. The scene plane is then located at a distance of approximately twice the focal length from the camera. As above, we assessed the proposed algebraic procedures for the three metric affine camera models. However, we also implemented an uncalibrated version of these and methods for the perspective camera model, which we now describe. For the orthographic camera, the calibrated version is denoted ALGOR-CAL and the uncalibrated version ALGOR-UNC. The latter estimates both the pose and the scale factor  $\gamma$  described in §2.2 by running the algebraic procedure ALGWP for the weak-perspective camera on 10 different images and keeping the average scale factor. For the weak-perspective camera, as described in §2.2, the notion of calibration does not apply and the algebraic procedure is simply denoted ALGWP. For the paraperspective camera, the calibrated version is denoted ALGPP-CAL and the uncalibrated version REFPP-UNC. The latter estimates both the pose and the camera’s effective focal length  $f$ . It uses the true value of the other entries of the perspective camera intrinsic matrix  $\mathbf{K}$ . The estimation is carried out by a joint refinement of pose and the effective focal length using Levenberg-Marquardt on the same 10 different image batches as for ALGOR-UNC. For the perspective camera, the calibrated version is denoted REFPE-CAL and the uncalibrated version REFPE-UNC. The former estimates pose and is solved by Levenberg-Marquardt refinement on each image independently. The latter estimates pose and the camera’s effective focal length and is solved similarly to REFPP-UNC. The results of rotation error in degrees for  $m \in \{3, 5, 20\}$  point sets are given in figure 2.

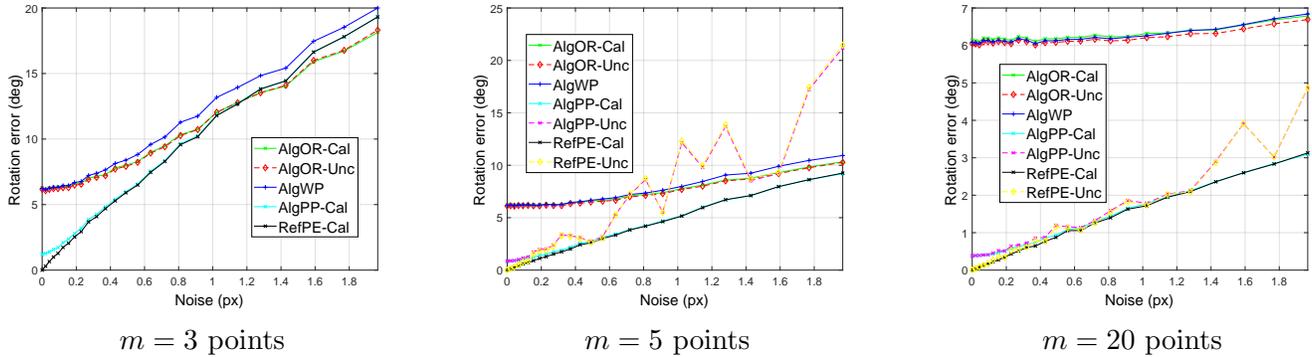


Figure 2: Results obtained from synthetic data to compare calibrated and uncalibrated versions of our algebraic procedures with the perspective camera model.

The uncalibrated paraperspective and perspective camera methods REFPP-UNC and REFPE-UNC are not shown in the  $m = 3$  point case as they both require at least 4 points. We observe that without noise, the modeling error of the orthographic and weak-perspective methods leads to an approximate rotation error of 6 deg, while for the calibrated paraperspective method ALGPP-CAL it leads to 1.2 deg and for the calibrated perspective method REFPE-CAL it leads to 0 deg, as there is no modeling error. Interestingly, we have that the weak-perspective camera performs always slightly worse than the orthographic camera, which strengthens the conclusions of §6.2.1 stating that one cannot simply swap the weak-perspective model in place of the orthographic camera to ease the solution method. The paraperspective and perspective cameras perform better than the orthographic camera in their calibrated versions. It is interesting to see however that for slightly more than a 1 px noise a crossover point appears in the case of  $m = 3$  points, beyond which the orthographic camera outperforms. We do not observe this crossover point for  $m \in \{5, 20\}$  points, at least within the  $[0, 2]$  px noise span that we simulated, because the influence of noise is then reduced. It is also interesting to see that for  $m \in \{3, 5\}$  the calibrated and uncalibrated versions of orthographic resection are undistinguishable and that for  $m = 20$  for uncalibrated version performs better. The uncalibrated version of the paraperspective and perspective camera models lead to different conclusions. They turned out to be much more sensitive to noise as compared to their calibrated versions. This is especially true for a low number of points, typically less than 10, and for a noise level larger than about 0.5 px, in which case the orthographic camera should be preferred.

#### 6.2.4 Performance in Plane-Based Structure-from-Motion

The goal of this experiment is to assess if using the proposed algebraic procedure for the orthographic camera model is beneficial to perform resection in a plane-based orthographic SfM pipeline which we described in (Collins and Bartoli, 2017). Our SfM pipeline solves for structure first and then resects all cameras, one

at a time. We originally solved camera resection with polynomial optimization, specifically using the quartic constraints leading to implementation POLYOR2, which was slow and failed at times, as also observed in §6.2.2. We compare the algebraic procedure ALGOR and nonlinear refinement REFOR for the orthographic camera and the algebraic procedure ALGWP for the weak-perspective camera as the resection method embedded in our SfM pipeline. We also compare with orthographic bundle adjustment BUNDADJOR, which comes as an optional final step of our SfM pipeline. We used the same simulation setup to simulate coplanar points as in the first experiment described in §6.2.1, but consider the object structure as unknown and use several images to run SfM. We varied noise, the number of points and the number of views.

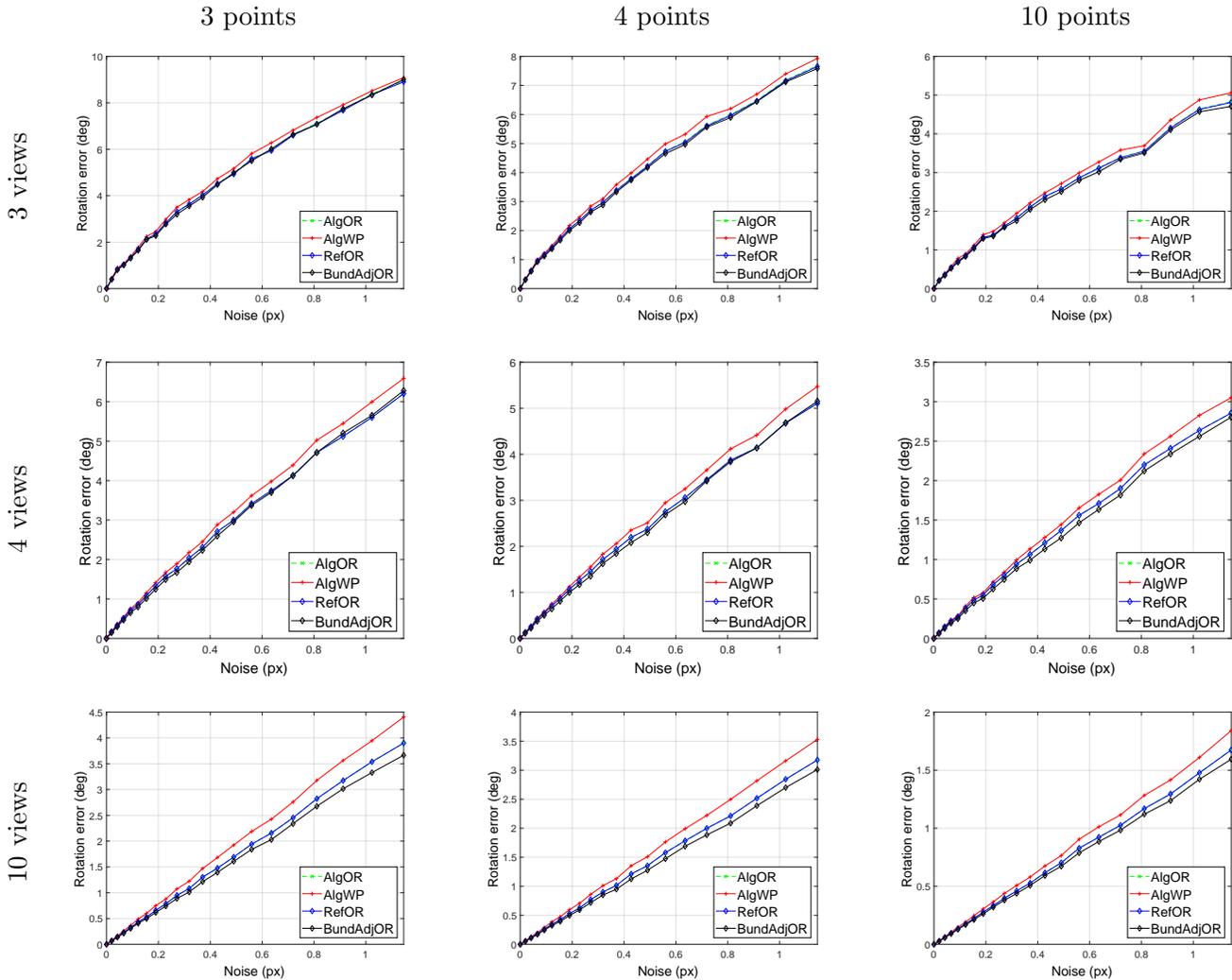


Figure 3: Results obtained from synthetic data to show how our algebraic procedures benefit to our SfM pipeline (Collins and Bartoli, 2017) and compare to bundle adjustment.

We observe that the rotation error for all methods degrades gracefully with an increasing amount of noise. Using resection ALGWP with the weak-perspective camera always performs significantly worse than the other models. There is a small gap between orthographic bundle adjustment BUNDADJOR and our

initialization pipeline based on orthographic resection. This difference is negligible for a small number of views but increases with the number of views, though remaining limited for 10 views. For the orthographic camera the algebraic procedure ALGOR yields no differences with the nonlinear refinement REFOR in all cases. This confirms that nonlinear refinement of the resection is not necessary and that our algebraic procedure ALGOR can be safely used in our SfM pipeline.

### 6.3 Real Data

The goal of our real data experiments is to compare the accuracy of the orthographic, weak-perspective and paraperspective camera models at computing pose. Our methodology is to compute absolute pose and use it to form relative pose between image pairs for which groundtruth is available. We consider two scenarios: pure rotation and general motion. In all experiments, we used the camera calibration toolbox for Matlab to compute the camera’s intrinsic parameter matrix  $K$  and to undistort the images. We used a high resolution, full-frame DSLR camera fitted with a 70-300 mm focal length zoom lens.

#### 6.3.1 Pure Rotation

We estimate the rotation of a small object undergoing a pure rotation about a fixed axis. The effective focal length from matrix  $K$  was estimated at  $f = 31,025$  px. The object is a wooden stick on which we drew a square of 10 mm side length. We measured the distance between the stick and the camera using a measuring tape and found 5.667 m. More precisely, we used the corner of the stick serving as the model’s origin as the start point and the centre of the front of the camera’s lens as the end point. We then precisely rotated the stick with 10 deg steps using a protractor between 0 and 90 deg with a precision lower than 1 deg. We took three pictures for every angle, which resulted in a total of 30 pictures. The stick and an excerpt of the images for every angle are shown in figure 4. For each image we manually clicked the 4 corners of the squares. We first ran pose estimation with the weak-perspective camera. This gave an estimate of the scale factor for each of the 30 images, from which, using the focal length, we could compute the object to camera distance. We then computed pose for both the weak-perspective and orthographic cameras. For each pair of images we found the rotation angle and compared it to groundtruth, from which we obtained the rotation error for the  $\frac{1}{2}(30^2 - 30) = 435$  image pairs. The statistics are reported in table 6 for the 5 possible combinations of  $m \geq 3$  point sets. We first comment on the distance estimation for the weak-perspective and paraperspective cameras. We observe that they produce almost indistinguishable estimates. In particular, they have the same average estimate and average error, which are  $|566.7 - 567.7| = 1.0$  cm for the average and  $|566.7 - 568.4| = 1.7$  cm for the median. This may be found to be accurate. Recall

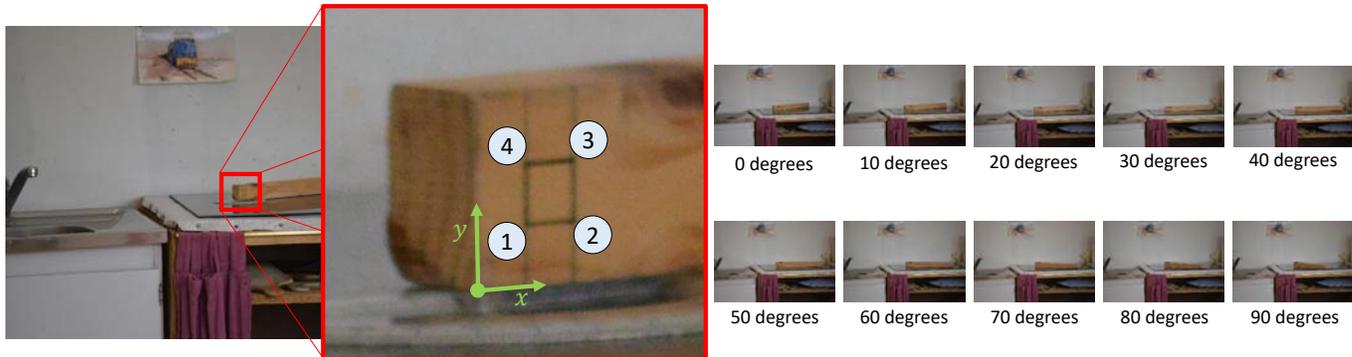


Figure 4: The wooden stick object we used for comparing rotation estimation. (left) the stick has a square of 10 mm side length located 10 mm away along  $x$  and  $y$  of the origin. (right) the stick was rotated between 0 and 90 deg.

	Point id sets					Average
	234	134	124	123	1234	
<b>Weak-perspective camera</b>						
Distance in m, average	5.662	5.703	5.687	5.649	5.682	5.677
Distance in m, median	5.664	5.703	5.698	5.667	5.690	5.684
Rotation error in deg, average	3.85	3.53	3.06	3.62	3.16	3.44
Rotation error in deg, median	2.76	3.12	2.56	2.80	2.45	2.74
<b>Paraperspective camera</b>						
Distance in m, average	5.662	5.703	5.687	5.649	5.683	5.677
Distance in m, median	5.664	5.703	5.698	5.667	5.690	5.684
Rotation error in deg, average	3.80	3.48	3.01	3.57	3.11	3.39
Rotation error in deg, median	2.67	3.03	2.52	2.74	2.40	2.67
<b>Orthographic camera</b>						
Rotation error in deg, average	3.13	3.01	2.55	3.15	2.51	2.46
Rotation error in deg, median	2.18	2.29	1.71	2.17	1.62	1.99

Table 6: Results obtained on the wooden stick object shown in figure 4.

nonetheless that the measured distance of 566.7 cm is already uncertain for the camera model, because the entrance pupil is the virtual image of the centre of the aperture stop, typically located within the lens (Steger, 2017b). As for the rotation estimation, we observe that the weak-perspective camera has the highest error and is closely followed by the paraperspective camera, both for the average and median angle errors. In spite of the accuracy of the distance estimates for the weak-perspective and paraperspective cameras, the orthographic camera produces significantly better estimates of the rotation. For instance, the relative average and median error differences between the weak-perspective and orthographic cameras are respectively  $2\frac{3.44-2.46}{3.44+2.46} = 33.2\%$  and  $\frac{2.74-1.99}{2.74+1.99} = 31.7\%$ . We conclude that first, the difference of the order of centimeters in the distance estimate has a significant influence on the rotation estimate, second, that the paraperspective approximation does not compensate for this error and third, that the orthographic camera is able to reduce noise significantly in spite of the distance measurement uncertainty.

### 6.3.2 General Motion

We estimate the pose of a smartphone using its screen’s four corners. The effective focal length from matrix  $K$  was estimated at  $f = 12,759$  px. In this experiment the object was static but the camera moved. In order to acquire the camera’s groundtruth displacement magnitude, we mounted it on a tripod to which a regular calibration checkerboard was rigidly attached. We named this camera A and introduced a second camera named camera B. Camera B is fixed and used to track camera A’s groundtruth displacement by computing the checkerboard’s relative pose. We took 7 pictures with camera A and camera B in this setup, which are shown in figure 5. For each image we measured the distance from the smartphone to camera A. We then clicked the 4 corners of the smartphone’s screen in each image from camera A. We used the 4 correspondences to estimate the weak-perspective and paraperspective cameras. Their scale factor then gives an estimate of the distance, which we compare to the measured distance in the following table:

Image id	1	2	3	4	5	6	7
Measured distance in m	2.5065	2.5010	2.4163	2.3052	2.3553	2.1479	2.6375
Weak-perspective distance in m	2.4781	2.4574	2.3753	2.2498	2.3238	2.1203	2.5900
Paraperspective distance in m	2.4824	2.4612	2.3775	2.2503	2.3250	2.1203	2.5974

The average discrepancy between the measured distance and the weak-perspective and paraperspective cameras are 3.9 cm and 3.6 cm respectively. We then computed the pose for all combinations of the  $\frac{1}{2} (7^2 - 7) = 21$  image pairs. For each pair of images we then found the rotation angle and translation magnitude and compared them to groundtruth, from which we obtained the rotation and translation errors. The statistics are reported in table 7. We observe that the best rotation estimates are obtained with the

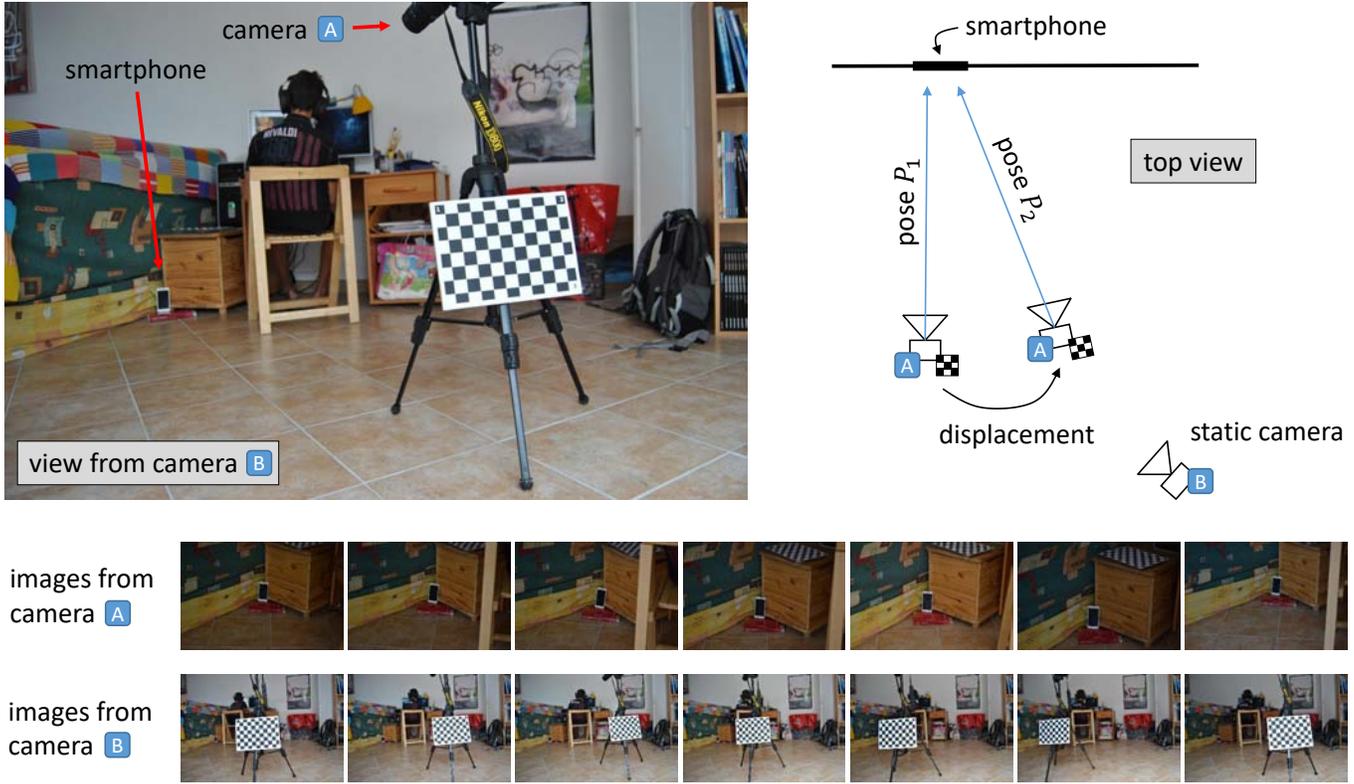


Figure 5: The smartphone we used for comparing full pose estimation. Camera A acquires the images used for pose estimation and camera B is used to find camera A’s groundtruth displacement. (top left) the view from camera A. (top right) the setup. (bottom) the seven images from camera A and camera B.

	Point id sets					Average
	234	134	124	123	1234	
<b>Weak-perspective camera</b>						
Rotation error in deg, average	3.33	3.48	4.14	3.90	3.55	3.68
Rotation error in deg, median	3.48	3.69	3.95	4.06	3.18	3.67
Translation error in %, average	7.85	3.43	10.85	9.59	5.55	7.45
Translation error in %, median	6.13	3.15	5.86	6.24	4.31	5.14
<b>Paraperspective camera</b>						
Rotation error in deg, average	2.61	2.75	2.88	2.85	2.50	2.71
Rotation error in deg, median	2.41	2.69	2.65	2.68	2.18	2.52
Translation error in %, average	7.72	7.27	15.95	13.62	8.35	10.58
Translation error in %, median	6.57	6.00	11.79	13.02	6.75	8.82
<b>Orthographic camera</b>						
Rotation error in deg, average	3.14	3.21	3.05	2.72	2.85	2.99
Rotation error in deg, median	3.07	3.09	3.17	2.71	2.97	3.00
Translation error in %, average	4.34	6.47	9.15	6.29	2.14	5.67
Translation error in %, median	3.26	6.08	6.51	4.45	0.58	4.17

Table 7: Pose results obtained on the smartphone shown in figure 5.

paraperspective camera, followed by the orthographic camera and then by the weak-perspective camera. For translation however, we have that the paraperspective camera produces the worst results, followed by the weak-perspective camera and then by the orthographic camera. We conclude that the orthographic camera is able to exploit the known distance in a very beneficial way, as it significantly outperforms the weak-perspective camera. The paraperspective approximation slightly improves the rotation part of pose but significantly degrades its translational part compared to the orthographic approximation.

## 6.4 Computation Time Analysis

We now evaluate the runtime of our algebraic procedure ALGOR for the orthographic camera and compare it to the alternative solutions POLYOR1, POLYOR2 and POLYOR3 based on Gloptipoly. Our implementation uses Matlab and our tests were run on a 1.8 GHz Intel Core i7-4500U CPU under Windows 10. For these experiments, we specifically fixed Gloptipoly’s relaxation order to 3 and left SeDuMi’s precision to its default value. These were chosen to ensure convergence for the majority of test cases. We used simulated data generated as described in §6.2.1 with a 1 px noise and varied the number of point correspondences  $m$ . The results were obtained by averaging over 10,000 trials. The four compared methods share the same two steps. The first step is to prepare the data for the canonical formulation (11) using the PrepareData procedure from table 1. The second step is to solve the canonical formulation. Our results are shown in figure 6.

The top-left graph shows that for  $m \leq 100$  the second step dominates the computation time in ALGOR. This second step has a fairly constant computation time. A figure for the overall computation time is 0.76 ms. We measured the standard deviation for  $m \in \{3, 5, 20, 100\}$  and obtained 0.041 ms. The bottom-left graph shows that for  $m > 100$  however, the first step dominates the computation time in ALGOR. For instance, for  $m = 1,000$ , we have an overall computation time of 1.2 ms. The two graphs of the right column show that for the Gloptipoly based methods the second step has a fairly constant computation time which however always dominates the overall computation time at 0.25 s, 0.36 s and 1.42 s for POLYOR1, POLYOR2 and POLYOR3 respectively.

Lower overall runtimes were reported in (Steger, 2017a). Though it is difficult to figure out exactly why this is so, this may be explained by at least two reasons: (Steger, 2017a) uses a C implementation whereas ours uses Matlab and (Steger, 2017a) used a more powerful CPU than ours and ran under Linux whereas we used Windows. In any case, our tests show that our algebraic procedure is significantly faster than the Gloptipoly based solutions, which remains true independently of the test platform.

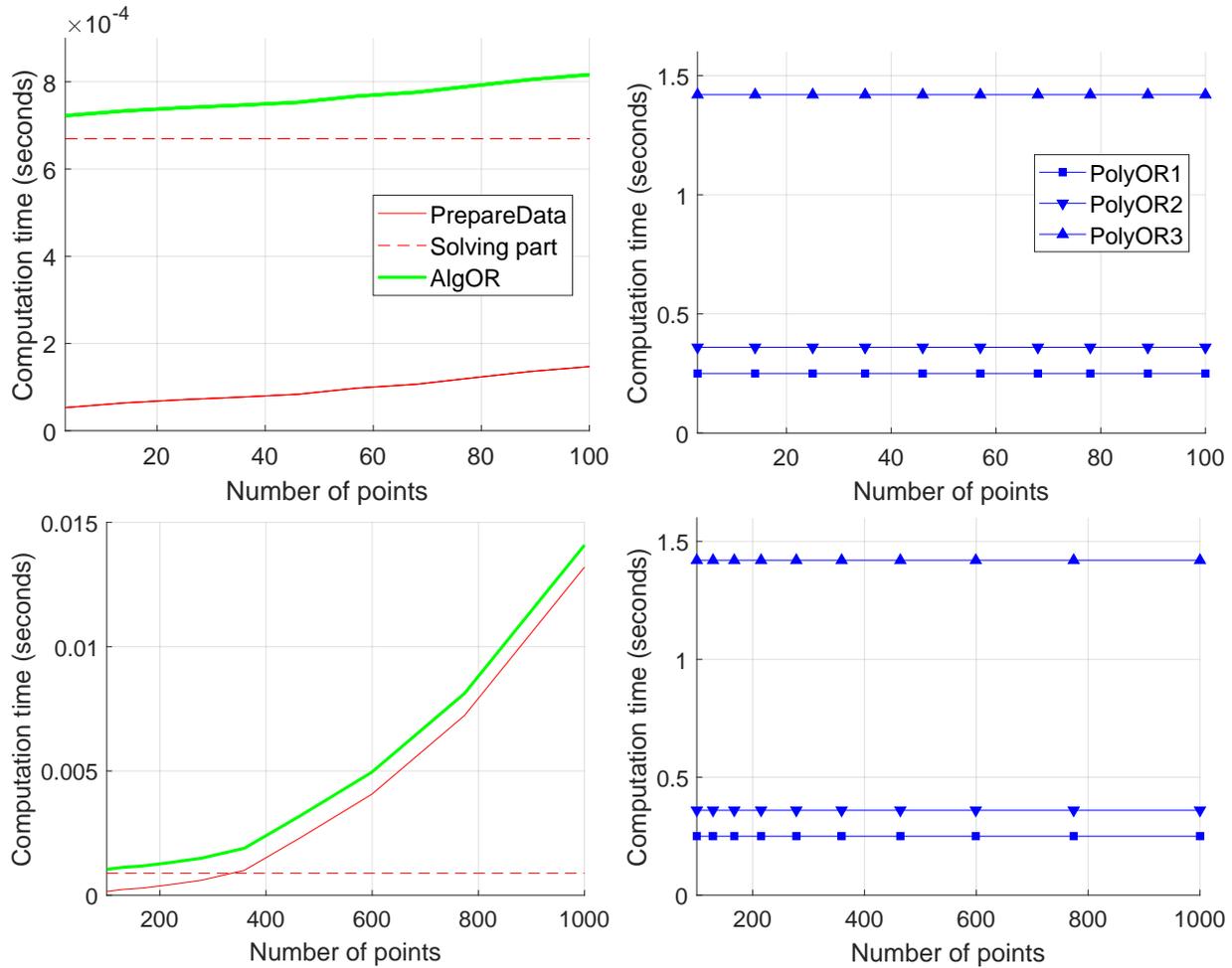


Figure 6: Computation time evaluation results. The top and bottom rows are for smaller and larger numbers of points respectively. The left and right columns are for PrepareData and ALGOR, and POLYOR1, POLYOR2 and POLYOR3 respectively.

## 7 Conclusion

We have studied the plane-based camera resection problem for the metric affine camera models. We have given a detailed analysis of its solution space, and showed that it could be solved exactly and efficiently. We found that the orthographic camera formed the most difficult case to handle for this problem. Nonetheless, we found that it led to an irreducible and non-solvable univariate sextic, whose roots form the problem's solution. We have given algebraic procedures which can be readily translated to a programming language. This paper means that all three main metric affine camera models now have an analytic, singularity-free and optimal resection solution with planar structures.

## A Translationless Formulation

We show how to eliminate the translation from the problem's initial formulation (1). We compute the centre of gravity of the model points as:

$$\mathbf{x} \stackrel{\text{def}}{=} \frac{1}{m} \mathbf{X} \mathbf{1}.$$

We then define a new set of model points by centering the initial ones, meaning that we translate their centre of gravity to the origin:

$$\mathbf{X}' \stackrel{\text{def}}{=} \mathbf{X} - \mathbf{x} \mathbf{1}^\top.$$

Defining the translation for the new set of model points as  $\mathbf{t}' \stackrel{\text{def}}{=} \mathbf{t} + \mathbf{P} \mathbf{x}$ , the problem is rewritten as:

$$\min_{\substack{\mathbf{P} \in \mathbb{M}_* \\ \mathbf{t}' \in \mathbb{R}^2}} \mathcal{O}_2(\mathbf{P}, \mathbf{t}') \quad \text{with} \quad \mathcal{O}_2(\mathbf{P}, \mathbf{t}') \stackrel{\text{def}}{=} \left\| \mathbf{P} \mathbf{X}' + \mathbf{t}' \mathbf{1}^\top - \mathbf{Y} \right\|_{\mathcal{F}}^2.$$

Taking the partial derivatives of  $\mathcal{O}_2$  with respect to  $\mathbf{t}'$ , we obtain:

$$2m \mathbf{t}' - 2m \mathbf{y} + 2m \mathbf{P} \mathbf{x}' = \mathbf{0},$$

where  $\mathbf{y} = \frac{1}{m} \mathbf{Y} \mathbf{1}$  is the centre of gravity of the observed points and  $\mathbf{x}' = \frac{1}{m} \mathbf{X}' \mathbf{1}$ . Because the new model points in  $\mathbf{X}'$  were obtained by centering the model points in  $\mathbf{X}$ , we have  $\mathbf{x}' = \mathbf{0}$ . We thus obtain:

$$\mathbf{t}' = \mathbf{y}.$$

Defining  $\mathbf{Y}' \stackrel{\text{def}}{=} \mathbf{Y} - \mathbf{y}\mathbf{1}^\top$  as a new set of observed points by centering the initial ones, we arrive at the translationless formulation:

$$\min_{\mathbf{P} \in \mathbb{M}_*} \mathcal{O}_3(\mathbf{P}) \quad \text{with} \quad \mathcal{O}_3(\mathbf{P}) \stackrel{\text{def}}{=} \|\mathbf{P}\mathbf{X}' - \mathbf{Y}'\|_{\mathcal{F}}^2.$$

Once the rotational part  $\mathbf{P}$  is estimated, the translation is given by:

$$\mathbf{t} = \mathbf{y} - \mathbf{P}\mathbf{x}. \quad (35)$$

## B Analytic Expression of the Cholesky Factor of $\mathbf{I} + \mathbf{d}\mathbf{d}^\top$

The Cholesky factor  $\mathbf{V}$  defined as  $\mathbf{I} + \mathbf{d}\mathbf{d}^\top = \mathbf{V}\mathbf{V}^\top$  and its inverse are given by:

$$\mathbf{V} = \begin{bmatrix} \sqrt{1+d_1^2} & 0 \\ \frac{d_1 d_2}{\sqrt{1+d_1^2}} & \frac{\sqrt{1+d_1^2+d_2^2}}{\sqrt{1+d_1^2}} \end{bmatrix} \quad \text{and} \quad \mathbf{V}^{-1} = \begin{bmatrix} \frac{1}{\sqrt{1+d_1^2}} & 0 \\ -\frac{d_1 d_2}{\sqrt{1+d_1^2}\sqrt{1+d_1^2+d_2^2}} & \frac{\sqrt{1+d_1^2}}{\sqrt{1+d_1^2+d_2^2}} \end{bmatrix}.$$

## C Aligning the Projection Direction to the $Z$ -Axis

We need to choose  $\mathbf{R}_\mathbf{d} \in \mathbb{S}\mathbb{O}_3$  such that  $\Pi_\mathbf{d}\mathbf{R}_\mathbf{d} = [\mathbf{H} \mathbf{0}]$  for some matrix  $\mathbf{H} \in GL_2(\mathbb{R})$ . This means finding a rotation which aligns the projection direction  $\mathbf{d}$  with the  $z$ -axis  $\text{stk}(0, 0, 1)$ . In IPPE (Collins and Bartoli, 2014), we chose the smallest angle rotation, using Rodrigues's formula. This has three limitations. *(i)* The formula fails for  $\mathbf{d} = \mathbf{0}$ , and is unstable for  $\|\mathbf{d}\|_2$  small. *(ii)* The formula involves trigonometry. *(iii)* The analytic expression of the required  $\mathbf{H}^{-1}$  and  $\det(\mathbf{H}^{-1})$  are very involved.

We propose a solution in radicals which addresses the above three limitations. This is a general solution, which has the smallest rotation solution as a special case. The requirement on  $\mathbf{R}_\mathbf{d}$  is rewritten as  $\Pi_\mathbf{d} = [\mathbf{H} \mathbf{0}]\mathbf{R}_\mathbf{d}^\top$ . Multiplying each side of this equation to the right by its transpose leads to  $\Pi_\mathbf{d}\Pi_\mathbf{d}^\top = \mathbf{H}\mathbf{H}^\top$ , or equivalently to  $\mathbf{I} + \mathbf{d}\mathbf{d}^\top = \mathbf{H}\mathbf{H}^\top$ . The general solution to this equation has one free degree of freedom, and is given by  $\mathbf{H} = \mathbf{V}\mathbf{F}$ , with  $\mathbf{F} \in \mathbb{O}_2$  an arbitrary orthonormal matrix and  $\mathbf{V} \in GL_2(\mathbb{R})$  a lower triangular matrix representing a Cholesky factor of  $\mathbf{I} + \mathbf{d}\mathbf{d}^\top = \mathbf{V}\mathbf{V}^\top$ . The first two rows of  $\mathbf{R}_\mathbf{d}^\top$  are obtained from  $\Pi_\mathbf{d} = \mathbf{H}\Pi_0\mathbf{R}_\mathbf{d}^\top$  as  $\Pi_0\mathbf{R}_\mathbf{d}^\top = \mathbf{F}^\top\mathbf{V}^{-1}\Pi_\mathbf{d}$ , and its third row is then obtained as their cross-product. A simple solution is obtained

by choosing  $\mathbf{F} = \mathbf{I}$  as:

$$\mathbf{R}_{\mathbf{d}} \stackrel{\text{def}}{=} \begin{bmatrix} 1 & -d_1d_2 & -d_1 \\ 0 & 1 + d_1^2 & -d_2 \\ d_1 & d_2 & 1 \end{bmatrix} \begin{bmatrix} a'' \\ a''b'' \\ b'' \end{bmatrix} \quad \text{with} \quad a'' \stackrel{\text{def}}{=} \frac{1}{\sqrt{1 + d_1^2}} \quad \text{and} \quad b'' \stackrel{\text{def}}{=} \frac{1}{\sqrt{1 + d_1^2 + d_2^2}}.$$

We have that  $\mathbf{H}^{-1}$  is the leading  $(2 \times 2)$  sub-matrix of  $\mathbf{R}_{\mathbf{d}}^\top$ , and  $\det(\mathbf{H}^{-1})$  is its bottom-right element. We propose an efficient algorithm to compute the Cholesky factor and its inverse, and matrix  $\mathbf{R}_{\mathbf{d}}$ , given in table 8.

<p>Function <b>CholeskyAndProjectionAxisToZAxis</b>(<math>\mathbf{d} \in \mathbb{R}^2</math>)</p> <ul style="list-style-type: none"> <li>• Set <math>a \leftarrow 1 + d_1^2</math>, <math>b \leftarrow a + d_2^2</math></li> <li>• Set <math>a' \leftarrow \sqrt{a}</math>, <math>b' \leftarrow \sqrt{b}</math></li> <li>• Set <math>a'' \leftarrow \frac{1}{a'}</math>, <math>b'' \leftarrow \frac{1}{b'}</math></li> <li>• Set <math>c \leftarrow a''d_1</math>, <math>d \leftarrow b''d_2</math></li> <li>• Set <math>\mathbf{H} \leftarrow \begin{bmatrix} a' &amp; 0 \\ cd_2 &amp; a''b' \end{bmatrix}</math>, <math>\mathbf{H}^{-1} \leftarrow \begin{bmatrix} a'' &amp; 0 \\ -cd &amp; a'b'' \end{bmatrix}</math></li> <li>• Set <math>\mathbf{R}_{\mathbf{d}}^\top \leftarrow \begin{bmatrix} \mathbf{H}^{-1} &amp; c \\ -b''d_1 &amp; -d &amp; b'' \end{bmatrix}</math></li> </ul> <p>Output <math>\mathbf{H}, \mathbf{H}^{-1} \in GL_2(\mathbb{R})</math>, <math>\mathbf{R}_{\mathbf{d}} \in \mathbb{SO}_3</math></p>
---

Table 8: **Algebraic procedure to align the projection axis to the Z axis.** The procedure computes the rotation matrix required to solve paraperspective resection.

## D Solution of the Rank-1 Equation of Type 3, $z\mathbf{u}\mathbf{u}^\top + z\mathbf{G} = \mathbf{K}$

**Proposition 1** (Rank-1 equation, type 3). *Let  $\mathbf{G} \in \mathbb{S}$  and  $\mathbf{K} \in \mathbb{S}$  be two known matrices. The following matrix equation defines three constraints on three unknowns in  $\mathbf{u} \in \mathbb{R}^2$  and  $z \in \mathbb{R}^+$ :*

$$z\mathbf{u}\mathbf{u}^\top + z\mathbf{G} = \mathbf{K}. \tag{36}$$

Equation (36) has always a unique solution for  $z$  and one or two solutions for  $\mathbf{u}$ . Let  $\mathbf{V} \in \mathbb{R}^{2 \times 2}$  be an upper triangular full rank matrix obtained from the Cholesky decomposition  $\mathbf{G} = \mathbf{V}\mathbf{V}^\top$  and set  $\mathbf{A} \leftarrow \mathbf{V}^{-1}\mathbf{K}\mathbf{V}^{-\top}$ . The solution for  $z$  is given by  $z = \lambda_2(\mathbf{A})$ . The two solutions for  $\mathbf{u}$  are given by  $\mathbf{u} = \pm \text{rank}_1\left(\frac{1}{\lambda_2(\mathbf{A})}\mathbf{K} - \mathbf{G}\right)$ . They both vanish if  $\lambda_1(\mathbf{A}) = \lambda_2(\mathbf{A})$ .

*Proof.* Because  $\mathbf{G} \in \mathbb{S}$ , we can always compute its Cholesky decomposition and rewrite equation (36) as:

$$z\mathbf{e}\mathbf{e}^\top + z\mathbf{I} = \mathbf{A}, \quad (37)$$

with  $\mathbf{e} \stackrel{\text{def}}{=} \mathbf{V}^{-1}\mathbf{u}$  and  $\mathbf{A} \stackrel{\text{def}}{=} \mathbf{V}^{-1}\mathbf{K}\mathbf{V}^{-\top}$ ,  $\mathbf{A} \in \mathbb{S}$ . Since  $z\mathbf{e}\mathbf{e}^\top = \mathbf{A} - z\mathbf{I}$  is rank-1 positive semi-definite, equation (37) is equivalent to:

$$\lambda_1(\mathbf{A} - z\mathbf{I}) \geq 0 \quad (38)$$

$$\lambda_2(\mathbf{A} - z\mathbf{I}) = 0. \quad (39)$$

Equation (39) implies  $\det(\mathbf{A} - z\mathbf{I}) = 0$  and so  $\exists j \in \{1, 2\}$ ,  $z = \lambda_j(\mathbf{A})$ . Equation (38) and lemma 2 in (Bartoli et al., 2015) then imply that  $z$  has a single solution given by  $z = \lambda_2(\mathbf{A})$ . Substituting the solution for  $z$  into equation (36), we obtain:

$$\mathbf{u}\mathbf{u}^\top = \frac{1}{\lambda_2(\mathbf{A})}\mathbf{K} - \mathbf{G},$$

and that  $\mathbf{u}$  is then obtained by invoking rank-1 factorization. Multiplying this equation by  $\mathbf{V}^{-1}$  to the left and  $\mathbf{V}^{-\top}$  to the right, we obtain:

$$\mathbf{e}\mathbf{e}^\top = \frac{1}{\lambda_2(\mathbf{A})}\mathbf{A} - \mathbf{I},$$

whose right-hand side vanishes for  $\lambda_1(\mathbf{A}) = \lambda_2(\mathbf{A})$ , leading to  $\mathbf{e} = \mathbf{u} = \mathbf{0}$ . □

## E Solution of the Rank-1 Equation of Type 4, $-z\mathbf{u}\mathbf{u}^\top + z\mathbf{G} = \mathbf{K}$

**Proposition 2** (Rank-1 equation, type 4). *Let  $\mathbf{G} \in \mathbb{S}$  and  $\mathbf{K} \in \mathbb{S}$  be two known matrices. The following matrix equation defines three constraints on three unknowns in  $\mathbf{u} \in \mathbb{R}^2$  and  $z \in \mathbb{R}^+$ :*

$$-z\mathbf{u}\mathbf{u}^\top + z\mathbf{G} = \mathbf{K}. \quad (40)$$

*Equation (40) has always a unique solution for  $z$  and one or two solutions for  $\mathbf{u}$ . Let  $\mathbf{V} \in \mathbb{R}^{2 \times 2}$  be an upper triangular full rank matrix obtained from the Cholesky decomposition  $\mathbf{G} = \mathbf{V}\mathbf{V}^\top$  and set  $\mathbf{A} \leftarrow \mathbf{V}^{-1}\mathbf{K}\mathbf{V}^{-\top}$ . The solution for  $z$  is given by  $z = \lambda_1(\mathbf{A})$ . The two solutions for  $\mathbf{u}$  are given by  $\mathbf{u} = \pm \text{rank}_1\left(\frac{1}{\lambda_1(\mathbf{A})}\mathbf{K} - \mathbf{G}\right)$ . They both vanish if  $\lambda_1(\mathbf{A}) = \lambda_2(\mathbf{A})$ .*

*Proof.* The proof is very similar to the proof of proposition 1. The conditions are now:

$$\lambda_1(\mathbf{A} - z\mathbf{I}) = 0 \quad (41)$$

$$\lambda_2(\mathbf{A} - z\mathbf{I}) \leq 0, \quad (42)$$

from which we obtain  $z = \lambda_1(\mathbf{A})$ , and allows us to use rank-1 factorization to retrieve  $\mathbf{u}$ .  $\square$

## F Intersecting a Centred Ellipse and the Unit Circle

### F.1 Problem Statement

Given  $\bar{\mathbf{E}} \in \mathbb{R}^{2 \times 2}$ ,  $\bar{\mathbf{E}} \succeq 0$ , representing the centred ellipse, the problem is to solve for the intersection points with coordinates  $\mathbf{q} \in \mathbb{R}^2$  such that:

$$\begin{cases} \mathbf{q}^\top \bar{\mathbf{E}} \mathbf{q} = 1 & (43) \\ \mathbf{q}^\top \mathbf{q} = 1. & (44) \end{cases}$$

The intersection points of two conics are generally found in two steps. The first step finds a degenerate member of the pencil defined by the two conics. This degenerate member is a pair of lines, and the second step is to recover and intersect these with one of the conics to retrieve the intersection points.

### F.2 Finding the Degenerate Conic as a Pair of Lines

By expanding and equating equations (43) and (44), we have:

$$\mathbf{q}^\top \bar{\mathbf{E}} \mathbf{q} = \mathbf{q}^\top \mathbf{q},$$

and thus:

$$\mathbf{q}^\top \bar{\mathbf{P}} \mathbf{q} = 0 \quad \text{with} \quad \bar{\mathbf{P}} \stackrel{\text{def}}{=} \bar{\mathbf{E}} - \mathbf{I},$$

which represents one of the sought degenerate conics in the pencil. The degenerate conic  $\mathbf{P} \stackrel{\text{def}}{=} \text{diag}(\bar{\mathbf{P}}, 0)$  generally represents a pair of lines, but degenerates to a single line if  $\bar{\mathbf{P}}$  is rank-1. These two lines intersect at the origin, as  $\ker(\mathbf{P}) = \text{stk}(0, 0, 1)$ . The two lines have homogeneous coefficient vectors  $\mathbf{m}, \mathbf{l} \in \mathbb{R}^3$ , with  $m_3 = l_3 = 0$ , as they contain the origin. We have:

$$\bar{\mathbf{P}} \stackrel{\text{def}}{=} \begin{bmatrix} a' & c' \\ c' & b' \end{bmatrix} = \bar{\mathbf{m}}\bar{\mathbf{l}}^\top + \bar{\mathbf{l}}\bar{\mathbf{m}}^\top,$$

which we expand to:

$$\begin{aligned} a' &= 2m_1l_1 \\ b' &= 2m_2l_2 \\ c' &= m_1l_2 + m_2l_1. \end{aligned}$$

Because the role of  $\mathbf{m}$  and  $\mathbf{l}$  is perfectly similar, we can find both by solving for only one of them. We chose to eliminate  $\bar{\mathbf{l}}$  and rewrite the first two equations as  $l_1 = \frac{a'}{2m_1}$  and  $l_2 = \frac{b'}{2m_2}$ . Substituting into the third equation we have:

$$m_1^2b' + m_2^2a' - 2m_1m_2c' = 0.$$

This is a homogeneous quadratic in two variables. We solve for  $m_2$  by fixing  $m_1 = 1$ , and rescale the result for convenience. Of the two solutions of the quadratic, one is associated to  $\bar{\mathbf{m}}$  and one to  $\bar{\mathbf{l}}$ , giving:

$$\bar{\mathbf{m}} = \frac{\text{sign}(a')}{\sqrt{2|a'|}} \left( \begin{bmatrix} a' \\ c' \end{bmatrix} + \begin{bmatrix} 0 \\ \sqrt{-\det(\bar{\mathbf{P}})} \end{bmatrix} \right) \quad \text{and} \quad \bar{\mathbf{l}} = \frac{1}{\sqrt{2|a'|}} \left( \begin{bmatrix} a' \\ c' \end{bmatrix} - \begin{bmatrix} 0 \\ \sqrt{-\det(\bar{\mathbf{P}})} \end{bmatrix} \right).$$

The sign of  $\det(\bar{\mathbf{P}})$  gives the number of real intersection points up to symmetry about the origin:  $\det(\bar{\mathbf{P}}) < 0$  corresponds to two real intersection points,  $\det(\bar{\mathbf{P}}) = 0$  corresponds to one real intersection point and  $\det(\bar{\mathbf{P}}) > 0$  corresponds to two complex intersection points.

### F.3 Intersecting the Pair of Lines with the Unit Circle

By intersecting the two lines with the unit circle, we retrieve the two solution points up to symmetry, with coordinates  $\mathbf{q}_1, \mathbf{q}_2 \in \mathbb{R}^2$ . We have:

$$\bar{\mathbf{m}}^\top \mathbf{q}_1 = 0 \quad \text{and} \quad \|\mathbf{q}_1\|_2 = 1,$$

and:

$$\bar{\mathbf{l}}^\top \mathbf{q}_2 = 0 \quad \text{and} \quad \|\mathbf{q}_2\|_2 = 1.$$

Because we only need  $\mathbf{q}_1$  and  $\mathbf{q}_2$  up to scale, we have:

$$\mathbf{q}_1 \propto \mathbf{S}\bar{\mathbf{m}} \propto \mathbf{S} \left( \begin{bmatrix} a' \\ c' \end{bmatrix} + \begin{bmatrix} 0 \\ \sqrt{-\det(\bar{\mathbf{P}})} \end{bmatrix} \right),$$

and:

$$\mathbf{q}_2 \propto \mathbf{S}\bar{\mathbf{I}} \propto \mathbf{S} \left( \begin{bmatrix} a' \\ c' \end{bmatrix} - \begin{bmatrix} 0 \\ \sqrt{-\det(\bar{\mathbf{P}})} \end{bmatrix} \right).$$

Expanding, and using  $\det(\bar{\mathbf{P}}) = 1 + \det(\bar{\mathbf{E}}) - \text{tr}(\bar{\mathbf{E}})$ , we rewrite these compactly in terms of  $\bar{\mathbf{E}}$  as:

$$\begin{aligned} \mathbf{q}_1 &\propto \left( \mathbf{S}(\bar{\mathbf{E}} - \mathbf{I}) - \sqrt{\text{tr}(\bar{\mathbf{E}}) - \det(\bar{\mathbf{E}}) - 1} \mathbf{I} \right) \mathbf{e}_1 \\ \mathbf{q}_2 &\propto \left( \mathbf{S}(\bar{\mathbf{E}} - \mathbf{I}) + \sqrt{\text{tr}(\bar{\mathbf{E}}) - \det(\bar{\mathbf{E}}) - 1} \mathbf{I} \right) \mathbf{e}_1. \end{aligned}$$

## G Impossibility of Cases 1.x.x Other Than 1.2.2

Case 1.2.2 is studied in §4.5. We show that all other subcases of 1.x.x are impossible under the problem's inputs.

### G.1 Case 1.1.2

We first specialize the parameterization of cases 1.x.x and reduce it to two unknowns. We then show that the orthonormality constraint yields a pair of incompatible equations on one of these two unknowns.

#### G.1.1 Specializing the Parameterization

We have  $\text{rank}(\mathbf{M}) = \text{rank}(\mathbf{M}_1) = 1$ . This is equivalent to  $\beta = -\sigma_1^2$ , as shown in §4.5.1. Using parameterization (20), we thus have:

$$\mathbf{M} = -\sigma_1^2 \mathbf{q}\mathbf{q}^\top.$$

#### G.1.2 Reducing the Unknowns to $\alpha, \nu \in \mathbb{R}$

**Solving for  $\mathbf{q}$ .** We have from equation (16) that:

$$\mathbf{M}_1 = -\sigma_1^2 \mathbf{q}\mathbf{q}^\top + \sigma_1^2 \mathbf{I} = \sigma_1^2 (\mathbf{I} - \mathbf{q}\mathbf{q}^\top).$$

It is easy to verify that this matrix has  $\mathbf{q}$  as nullvector and  $\mathbf{S}\mathbf{q}$  as eigenvector, with eigenvalue  $\sigma_1^2$ . Still using equation (16), we thus have:

$$\sigma_1 (\mathbf{I} - \mathbf{q}\mathbf{q}^\top) \mathbf{b}_1 = \mathbf{z}'_1. \quad (45)$$

Left-multiplying by  $\mathbf{q}^\top$  gives  $\mathbf{q}^\top \mathbf{z}'_1 = 0$ . Because  $\|\mathbf{q}\|_2 = 1$ , we obtain:

$$\mathbf{q} = \pm \frac{\mathbf{S}\mathbf{z}'_1}{\|\mathbf{z}'_1\|_2}. \quad (46)$$

**Solving for  $\mathbf{b}_2$ .** We have from equation (17) that:

$$\mathbf{b}_2 = \sigma_2 \mathbf{M}_2^{-1} \mathbf{z}'_2.$$

By substituting  $\mathbf{M}_2 = \mathbf{M} + \sigma_2^2 \mathbf{I} = -\sigma_1^2 \mathbf{q} \mathbf{q}^\top + \sigma_2^2 \mathbf{I}$ , and  $\mathbf{q}$  from equation (46), we obtain:

$$\mathbf{b}_2 = \sigma_2 \left( \sigma_2^2 \mathbf{I} - \sigma_1^2 \frac{\mathbf{S} \mathbf{z}'_1 \mathbf{z}'_1{}^\top \mathbf{S}}{\|\mathbf{z}'_1\|_2^2} \right)^{-1} \mathbf{z}'_2. \quad (47)$$

**Solving for  $\mathbf{b}_1$ .** We left-multiply equation (45) by its eigenvector  $\mathbf{q}^\top \mathbf{S}^\top$ , giving:

$$\sigma_1 \mathbf{q}^\top \mathbf{S}^\top \mathbf{b}_1 = \mathbf{q}^\top \mathbf{S}^\top \mathbf{z}'_1.$$

Substituting  $\mathbf{q}$  from equation (46), we obtain:

$$\sigma_1 \mathbf{z}'_1{}^\top \mathbf{S}^\top \mathbf{S}^\top \mathbf{b}_1 = \mathbf{z}'_1{}^\top \mathbf{S}^\top \mathbf{S}^\top \mathbf{z}'_1.$$

Because  $\mathbf{S}^\top \mathbf{S}^\top = -\mathbf{I}$ , we arrive at:

$$\sigma_1 \mathbf{z}'_1{}^\top \mathbf{b}_1 = \|\mathbf{z}'_1\|_2^2.$$

The general solution to this equation has one free parameter  $\nu \in \mathbb{R}$  and takes the form:

$$\mathbf{b}_1 = \frac{\mathbf{z}'_1}{\sigma_1} + \nu \mathbf{S} \mathbf{z}'_1. \quad (48)$$

**Expressing  $\mathbf{b}$ .** We use the expression  $\mathbf{b} = \alpha \mathbf{S} \mathbf{q}$  from the parameterization (21), giving, using equation (46):

$$\mathbf{b} = \alpha \frac{\mathbf{z}'_1}{\|\mathbf{z}'_1\|_2}. \quad (49)$$

### G.1.3 Impossibility

We have a reduced expression of  $\mathbf{A}$  in terms of the two unknowns  $\alpha, \nu \in \mathbb{R}$ , from the expressions (48) and (47) of  $\mathbf{b}_1$  and  $\mathbf{b}_2$ , and (49) of  $\mathbf{b}$ . We observe that they have a stronger dependency on  $\mathbf{z}'_1$  than on  $\mathbf{z}'_2$ . We thus use the formulation group to fix  $Z_{2,1} = 0$ , giving the following simplified expression for  $\mathbf{A}$ :

$$\mathbf{A} = \begin{bmatrix} \frac{Z_{1,1}}{\sigma_1} & \frac{Z_{1,2}}{\sigma_2} & \alpha \operatorname{sign}(Z_{1,1}) \\ \nu Z_{1,1} & \frac{\sigma_2 Z_{2,2}}{\sigma_2^2 - \sigma_1^2} & 0 \end{bmatrix}.$$

It is then straightforward to see that the orthonormality constraint  $\mathbf{A}\mathbf{A}^\top = \mathbf{I}$  yields three equations, one quadratic in  $\alpha$ ,  $\|[1\ 0]\mathbf{A}\|_2^2 = 1$ , one quadratic in  $\nu$ ,  $\|[0\ 1]\mathbf{A}\|_2^2 = 1$ , and one linear in  $\nu$ ,  $[1\ 0]\mathbf{A}\mathbf{A}^\top \text{stk}(0, 1) = 0$ .

The linear equation in  $\nu$  gives:

$$\nu_0 = \frac{\sigma_1 Z_{1,2} Z_{2,2}}{(\sigma_1^2 - \sigma_2^2) Z_{1,1}^2},$$

and the quadratic equation gives two complex solutions:

$$\nu_{\pm} = \pm \frac{\sigma_2 Z_{2,2} i}{(\sigma_1^2 - \sigma_2^2) Z_{1,1}^2},$$

with  $i^2 = \sqrt{-1}$ . The linear and the quadratic equations are thus compatible only if they both vanish, which occurs if  $\sigma_2 = 0$ , which contradicts the hypothesis of non-colinearity of the model points, or  $Z_{2,2} = 0$ , which, because  $Z_{2,1} = 0$ , contradicts the hypothesis that  $\text{rank}(\mathbf{Z}) = 2$ .

## G.2 Case 1.2.1

The proof of impossibility is obtained very similarly to case 1.1.2. We have:

$$\begin{aligned} \mathbf{q} &= \pm \frac{\mathbf{S}\mathbf{z}'_2}{\|\mathbf{z}'_2\|_2} \\ \mathbf{b}_1 &= \sigma_1 \mathbf{M}_1^{-1} \mathbf{z}'_1 \\ \mathbf{b}_2 &= \frac{\mathbf{z}'_2}{\sigma_2} + \nu \mathbf{S}\mathbf{z}'_2 \\ \mathbf{b} &= \alpha \frac{\mathbf{z}'_2}{\|\mathbf{z}'_2\|_2}. \end{aligned}$$

We use to formulation group to set  $Z_{1,2} = 0$ , which leads, using the orthonormality constraint, to two incompatible equations in  $\nu$ .

## G.3 Case 1.1.1

We have  $\text{rank}(\mathbf{M}) = \text{rank}(\mathbf{M}_1) = \text{rank}(\mathbf{M}_2) = 1$  and  $\sigma_1 = \sigma_2$ . Following cases 1.1.2 and 1.2.1, we have:

$$\mathbf{q} = \pm \frac{\mathbf{S}\mathbf{z}'_1}{\|\mathbf{z}'_1\|_2} \quad \text{and} \quad \mathbf{q} = \pm \frac{\mathbf{S}\mathbf{z}'_2}{\|\mathbf{z}'_2\|_2}.$$

Therefore, we must have  $\mathbf{z}'_1 \propto \mathbf{z}'_2$ , which contradicts the hypothesis that  $\text{rank}(\mathbf{Z}) = 2$ .

### G.4 Cases 1.0.x

We have  $M_1 = 0$ , and from equation (16) and parameterization (20):

$$\beta \mathbf{q} \mathbf{q}^\top + \sigma_1^2 \mathbf{I} = 0.$$

The off-diagonal equation reduces to  $\beta q_1 q_2 = 0$ , which means that one of these three variables must vanish. However, the diagonal equations are  $\beta q_1^2 + \sigma_1^2 = 0$  and  $\beta q_2^2 + \sigma_1^2 = 0$ . Because  $\sigma_1 \neq 0$ , they respectively imply  $\beta \neq 0$ ,  $q_1 \neq 0$  and  $\beta \neq 0$ ,  $q_2 \neq 0$ , which contradicts the off-diagonal equation.

### G.5 Cases 1.x.0

We have  $M_2 = 0$ , and follow cases 1.0.x to discard this case.

## H Algebraic Procedure for the Weak-Perspective Camera

We give our algebraic procedure for the weak-perspective camera, obtained as a simplification of the paraperspective camera case, in table 9.

<p>Function <math>\mathbf{WPR}(\mathbf{X} \in \mathbb{R}^{3 \times m}, \mathbf{Y} \in \mathbb{R}^{2 \times m})</math></p> <ul style="list-style-type: none"> <li>• Set <math>(\sigma_1, \sigma_2, \mathbf{Z}, \mathcal{O}, \mathbf{x}, \mathbf{y}, \mathbf{U}) \leftarrow \text{PrepareData}(\mathbf{X}, \mathbf{Y})</math>, <math>\mathbf{B} \leftarrow \mathbf{Z} \text{diag}(\sigma_1^{-1}, \sigma_2^{-1})</math></li> <li>• Compute <math>\gamma, \mathbf{u}</math> by solving the rank-1 equation of type 4</li> </ul> <p style="padding-left: 2em;">Set <math>\mathbf{A} \leftarrow \mathbf{B} \mathbf{B}^\top</math>, <math>\gamma^2 \leftarrow \lambda_1(\mathbf{A})</math>, <math>\mathbf{u} \leftarrow \text{rank}_1\left(\mathbf{I} - \frac{1}{\gamma^2} \mathbf{A}\right)</math></p> <ul style="list-style-type: none"> <li>• Set <math>\bar{\mathbf{Q}} \leftarrow \begin{bmatrix} \frac{1}{\gamma} \mathbf{B} &amp; \mathbf{u} \end{bmatrix}</math></li> <li>• Form <math>\mathbf{Q}_+</math> from <math>\bar{\mathbf{Q}}</math> and the cross-product of its two rows</li> <li>• Set <math>\mathbf{Q}_- \leftarrow \mathbf{Q}_+ \odot \begin{bmatrix} 1 &amp; 1 &amp; -1 \\ 1 &amp; 1 &amp; -1 \\ -1 &amp; -1 &amp; 1 \end{bmatrix}</math></li> <li>• Set <math>\mathbf{R}_+ \leftarrow \det(\mathbf{U}) \mathbf{Q}_+ \mathbf{U}^\top</math> and <math>\mathbf{R}_- \leftarrow \det(\mathbf{U}) \mathbf{Q}_- \mathbf{U}^\top</math></li> <li>• Set <math>\mathbf{t}_+ \leftarrow \mathbf{y} - \det(\mathbf{U}) [\mathbf{B} \ \gamma \mathbf{u}] \mathbf{U}^\top \mathbf{x}</math> and <math>\mathbf{t}_- \leftarrow \mathbf{y} - \det(\mathbf{U}) [\mathbf{B} \ -\gamma \mathbf{u}] \mathbf{U}^\top \mathbf{x}</math></li> </ul> <p>Output <math>\gamma \in \mathbb{R}^+</math>, <math>\mathbf{R}_+, \mathbf{R}_- \in \mathbb{SO}_3</math>, <math>\mathbf{t}_+, \mathbf{t}_- \in \mathbb{R}^2</math>, <math>\mathcal{O} \in \mathbb{R}^+</math></p>
--

Table 9: **Algebraic procedure solving weak-perspective planar resection.** The procedure always returns two rotation and translation pairs and a single cost. This is because the problem has two solutions in general. These two solutions become equal for  $\mathbf{u} = \mathbf{0}$ .

## I Derivation Details for Case 2.x.x

We derive an optimized solution for case 2.x.x using the formulation give in §4.4.

**Solving for B.** We define  $\mathbf{z} \stackrel{\text{def}}{=} \text{vect}(\mathbf{Z})$ ,  $\mathbf{g} \stackrel{\text{def}}{=} \text{stk}(a, b)$  and:

$$\mathbf{K} \stackrel{\text{def}}{=} \begin{bmatrix} s\sigma_1 & & & \\ & s\sigma_1 & & \\ & & -\sigma_2 & \\ \sigma_2 & & & \end{bmatrix}.$$

This allows us to rewrite the cost function as:

$$\mathcal{O}_7(a, b, s) = \|\mathbf{K}\mathbf{g} - \mathbf{z}\|_2^2.$$

We find the optimal solution for  $\mathbf{g}$ . We define a Lagrange multiplier  $\ell$  to enforce the constraint  $a^2 + b^2 = 1$  and the Lagrangian:

$$\mathcal{L}'(a, b, s, \ell) \stackrel{\text{def}}{=} \mathcal{O}_7(a, b, s) + \ell (\|\mathbf{g}\|_2^2 - 1).$$

Differentiating with respect to  $\mathbf{g}$  and nullifying, we obtain:

$$\frac{\partial \mathcal{L}'}{\partial \mathbf{g}} = 2\mathbf{K}^\top \mathbf{K}\mathbf{g} - 2\mathbf{K}^\top \mathbf{z} + 2\ell \mathbf{g} = \mathbf{0}.$$

We obtain  $\mathbf{g}$  as a function of  $\ell$  and  $s$  as:

$$\mathbf{g} = \left( \mathbf{K}^\top \mathbf{K} + \ell \mathbf{I} \right)^{-1} \mathbf{K}^\top \mathbf{z} = \frac{1}{\sigma_1^2 + \sigma_2^2 + \ell} \mathbf{K}^\top \mathbf{z} \quad \text{with} \quad \mathbf{K}^\top \mathbf{z} = \begin{bmatrix} s\sigma_1 Z_{1,1} + \sigma_2 Z_{2,2} \\ s\sigma_1 Z_{2,1} - \sigma_2 Z_{1,2} \end{bmatrix}.$$

We determine two possible values for  $\ell$  by substituting the solution for  $\mathbf{g}$  into the constraint, giving:

$$\left\| \mathbf{K}^\top \mathbf{z} \right\|_2^2 = (\sigma_1^2 + \sigma_2^2 + \ell)^2,$$

which we expand to the following quadratic in  $\ell$ :

$$\ell^2 + 2(\sigma_1^2 + \sigma_2^2)\ell + (\sigma_1^2 + \sigma_2^2)^2 - \left\| \mathbf{K}^\top \mathbf{z} \right\|_2^2 = 0.$$

The discriminant is  $\delta = 4 \|\mathbf{K}^\top \mathbf{z}\|_2^2 > 0$  and we thus obtain:

$$\ell = -\sigma_1^2 - \sigma_2^2 + r \|\mathbf{K}^\top \mathbf{z}\|_2 \quad \text{with} \quad r \in \{-1, 1\}.$$

As expected, we obtain  $\mathbf{g}$  with unitary norm as:

$$\mathbf{g} = \frac{r\mathbf{K}^\top \mathbf{z}}{\|\mathbf{K}^\top \mathbf{z}\|_2}.$$

The final step is to find the optimal solution for  $s$  and  $r$ . Substituting the solution for  $\mathbf{g}$  into the cost function yields:

$$\min_{s,r \in \{-1,1\}} \mathcal{O}_8(s,r) \quad \text{with} \quad \mathcal{O}_8(s,r) \stackrel{\text{def}}{=} \left\| \frac{r\mathbf{K}\mathbf{K}^\top \mathbf{z}}{\|\mathbf{K}^\top \mathbf{z}\|_2} - \mathbf{z} \right\|_2^2.$$

Expanding the cost function, we obtain:

$$\mathcal{O}_8(s,r) = \frac{\|\mathbf{K}\mathbf{K}^\top \mathbf{z}\|_2^2}{\|\mathbf{K}^\top \mathbf{z}\|_2^2} + \|\mathbf{z}\|_2^2 - 2r \frac{\mathbf{z}^\top \mathbf{K}\mathbf{K}^\top \mathbf{z}}{\|\mathbf{K}^\top \mathbf{z}\|_2}.$$

The first and second terms do not depend on  $r$ . The third term may be rewritten as:

$$-2r \frac{\mathbf{z}^\top \mathbf{K}\mathbf{K}^\top \mathbf{z}}{\|\mathbf{K}^\top \mathbf{z}\|_2} = -2r \frac{\|\mathbf{K}^\top \mathbf{z}\|_2^2}{\|\mathbf{K}^\top \mathbf{z}\|_2} = -2r \|\mathbf{K}^\top \mathbf{z}\|_2,$$

and is thus minimized for  $r = 1$ . We now solve for  $s$ . The problem becomes:

$$\min_{s \in \{-1,1\}} \mathcal{O}_8(s,1).$$

We have  $\|\mathbf{K}\mathbf{h}\|_2^2 = (\sigma_1^2 + \sigma_2^2)\|\mathbf{h}\|_2^2$ , and we can thus expand the cost function as:

$$\mathcal{O}_8(s,1) = \sigma_1^2 + \sigma_2^2 + \|\mathbf{z}\|_2^2 - 2 \|\mathbf{K}^\top \mathbf{z}\|_2.$$

Only the last term depends on  $s$ . Because of the negative sign, we have to maximize it, and because it is always positive, we can maximize its square. We thus have:

$$\max_{s \in \{-1,1\}} \mathcal{O}_9(s) \quad \text{with} \quad \mathcal{O}_9(s) \stackrel{\text{def}}{=} \|\mathbf{K}^\top \mathbf{z}\|_2^2.$$

The cost function is expanded to:

$$\mathcal{O}_9(s) = \sigma_1^2 \|\mathbf{z}'_1\|_2^2 + \sigma_2^2 \|\mathbf{z}'_2\|_2^2 + 2s\sigma_1\sigma_2 \det(\mathbf{Z}).$$

Only the last term depends on  $s$ , and we thus have  $s = \text{sign}(\det(\mathbf{Z}))$ .

## References

- K. Arun, T. Huang, and S. Blostein. Least-squares fitting of two 3-D points sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, September 1987.
- A. Bartoli, Y. Gérard, F. Chadebecq, T. Collins, and D. Pizarro. Shape-from-template. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2099–2118, October 2015.
- J. R. Cardoso and K. Zietak. On a substiefel procrustes problem arising in computer vision. *Numerical Linear Algebra With Applications*, 22(3):523–547, May 2015.
- T. Collins and A. Bartoli. Locally affine and planar deformable surface reconstruction from video. In *International Workshop on Vision, Modeling and Visualization*, 2010.
- T. Collins and A. Bartoli. Infinitesimal plane-based pose estimation. *International Journal of Computer Vision*, 109(3):252–286, September 2014.
- T. Collins and A. Bartoli. Planar structure-from-motion with affine camera models: Closed-form solutions, ambiguities and degeneracy analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1237–1255, 2017.
- O. Faugeras, Q.-T. Luong, and T. Papadopoulos. *The Geometry of Multiple Images*. MIT Press, 2001.
- M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Computer Vision, Graphics and Image Processing*, 24(6):381–395, June 1981.
- T. R. Hagedorn. General formulas for solving solvable sextic equations. *Journal of Algebra*, 233(2):704–757, November 2000.
- R. M. Haralick, H. Joo, C. Lee, X. Zhuang, V. G. Vaidya, and M. B. Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man and Cybernetics*, 6(19):1426–1446, November 1989.
- R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. Second Edition.
- D. Henrion, J. B. Lasserre, and J. Loeferberg. Gloptipoly 3: Moments, optimization and semidefinite programming. *Optimization Methods and Software*, 24(4-5):761–779, 2009.

- R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy. Object pose: The link between weak perspective, paraperspective and full perspective. *International Journal of Computer Vision*, 22(2):173–189, March 1997.
- B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127–1135, July 1988.
- G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *International Symposium on Mixed and Augmented Reality*, 2007.
- D. Oberkampf, D. F. DeMenthon, and L. S. Davis. Iterative pose estimation using coplanar feature points. *Computer Vision and Image Understanding*, 63(3):495–511, May 1996.
- N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–201, November 2008.
- C. Steger. Algorithms for the orthographic- $n$ -point problem. *Journal of Mathematical Imaging and Vision*, 2017a. To appear.
- C. Steger. A comprehensive and versatile camera model for cameras with tilt lenses. *International Journal of Computer Vision*, 123(2):121–159, 2017b.
- I. Stewart. *Galois Theory*. CRC Press, 2015. Fourth Edition.
- J. Taylor, A. D. Jepson, and K. Kutulakos. Non-rigid structure from locally-rigid motion. In *International Conference on Computer Vision and Pattern Recognition*, 2010.
- S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.