Computer Vision and Image Understanding
journal homepage: www.elsevier.com

Certifiable Planar Relative Pose Estimation with Gravity Prior

Mercedes Garcia-Salguero[a,**], Javier Gonzalez-Jimenez[a]

[a]*Machine Perception and Intelligent Robotics (MAPIR) Group, System Engineering and Automation Department, University of Malaga, Campus de Teatinos, 29071 Malaga, Spain*

ABSTRACT

In this work we propose a certifiable solver for the relative pose problem between two calibrated cameras under the assumptions that the unknown 3D points lay on an unknown plane and the axis of rotation is given, *e.g.* by an IMU. The problem is stated in terms of the rotation, translation and plane parameters and solved iteratively by an on-manifold optimization. Since the problem is non-convex, we then try to certify this solution as the global optimum. For that, we leverage four different definitions for the search space that provide us with different certification capabilities. Since the formulations lack the Linear Independence Constraint Qualification and two of them have more constraints than variables, we cannot derive a closed-form certifier. Instead, we leverage the iterative algorithm proposed in our previous work Garcia-Salguero and Gonzalez-Jimenez (2023) that does not assume any condition on the problem formulation. Our evaluation on synthetic and real data shows that the smaller formulations are enough to certify most of the solutions, whereas the redundant ones certify all of them, including problem instances with highly noisy data. Code can be found in https://github.com/mergarsal.

1. Introduction

Estimating the relative pose between two cameras is the cornerstone of many computer vision tasks, from visual odometry to visual SLAM. Although the literature regarding this problem is vast, there exists a strong focus on using the essential matrix for this task. However, degenerate configurations lead to a family of solutions for the essential matrix, hence hindering the extraction of the (unique) pose from it. This is the case in which the 3D points that give rise to the observations lay on a plane, see figure 1. This configuration, which is predominant in man-made environments, requires the consideration of another relation between the observations via the so-called 2D homography matrix Hartley and Zisserman (2003). As for the general case, this point-to-point relation also allows us to retrieve the relative pose and, additionally, the information about the unknown plane π.

In the general case, the homography matrix is estimated through a linear solver from at least four points in correspondence Hartley and Zisserman (2003). However, when additional information is available, *e.g.* the gravity vector is inferred from an IMU, the degrees-of-freedom (DoF) of the homography matrix decreases from 8 to 6. Although this knowledge can be incorporated after the initial estimation, it also provides us with a convenient prior that may simplify robust algorithms, such as RANSAC Fischler and Bolles (1981), by reducing the number of iterations and the number of wrong inliers. The reduction of DoFs implies additional requirements on the solutions that are not fulfilled by general solvers, and therefore our interest in deriving an explicit solver for these cases.

In this work, we tackle the relative pose problem through the homography matrix assuming the axis of rotation is known, *e.g.* inferred from an IMU. Unfortunately, the homography matrix doesn't inherit a clear structure from this information as the essential matrix does and thus we pose the problem in terms of the rotation, translation and plane parameters in order to reflect the prior information. This approach does not require decomposing the homography matrix, thus avoiding the associated numerical instabilities *e.g.* Malis and Vargas (2007), although it depicts the nonconvexity of the problem, that is, it may have more than one local minimum. Our goal is to solve this problem with optimality guarantees, and while global solvers can be derived see

**Corresponding author
e-mail: mercedesgarsal@uma.es (Mercedes Garcia-Salguero), javiergonzalez@uma.es (Javier Gonzalez-Jimenez)

section 2, we are interested in solving it in a more efficient way. For that, we rely on the so-called certifiable algorithms, which are able to certificate the optimality of a solution, but not obtain it.

Contributions: In this work we propose an efficient optimality certifiable solver that first estimates the solution and then tries to certify whether the solution is the actual global optimum. Our first contribution is a nonminimal, iterative solver that obtains the pose (rotation and translation) and plane parameters (normal vector) through an on-manifold optimization that considers all the correspondences. This solution is, however, only a minimum of the problem and so we do not know a priori if it is the global optimum. Our second contribution is an optimality certifier based on the dual problem that tries to certify the optimality of this solution without modifying it. This certifier is only able to say whether the solution is the global optimum or is inconclusive about its optimality, information that can be leveraged by other applications to decide whether or not to use the solution. To derive this relaxation and eventually the certifier, we first formulate the problem to have quadratic cost and constraints, and since the tightness of the relaxation depends on the set of constraints, we propose four different sets with up to 97 constraints and 21 variables. Because the formulations don't fulfill the Linear Independence Constraint Qualification (LICQ) and two of them have more constraints than variables, we cannot derive the 'standard' closed-form certifiers, see section 2. Instead, we leverage the generic iterative algorithm for certification proposed in Garcia-Salguero and Gonzalez-Jimenez (2023) which does not assume any condition on the formulation. Notice that the problem and formulation of this work is different from Garcia-Salguero and Gonzalez-Jimenez (2023) as we seek the homography matrix, thus involving a different formulation of the problem. We evaluate the solver and certifier on synthetic and real data, considering degenerate configurations, and showcase that our proposal is able to estimate and certify the solution for most non-degenerate configurations, even with highly noisy data. We make the code available at `https://www.github.com/mergarsal`.

Notation: We introduce here the notation employed in the manuscript. Bold, upper-case letters denote matrices *e.g.* $\boldsymbol{H}$, whereas bold, lower-case denotes (column) vector *e.g.*, $\boldsymbol{t}$, $\boldsymbol{f}_i$ and normal font letters *e.g.*, a, b denote real scalar. We will denote with $\mathbb{R}^{n\times m}$ the set of $n \times m$ real-valued matrices, $\mathbb{S}^n \subset \mathbb{R}^{n\times n}$ the set of symmetric matrices of dimension $n\times n$ and $\mathbb{S}^n_+$ the cone of positive semidefinite (PSD) matrices of dimension $n\times n$. A PSD matrix will be also denoted by $\succeq$, *i.e.*, $\boldsymbol{S} \succeq 0 \Leftrightarrow \boldsymbol{S} \in \mathbb{S}^n_+$. The operator vec($\boldsymbol{E}$) vectorises the given matrix $\boldsymbol{E} \in \mathbb{R}^{m\times n}$ column-wise, and $\otimes$ denotes the kronecker product. We will denote the trace of a matrix as $\text{tr}(\boldsymbol{A}) = \sum_{i=1}^{n} a_{ii}$, $\boldsymbol{A} = [a_{ij}] \in \mathbb{R}^{n\times n}$. In this work, we identify 3D rotations with points in the rotation group $\text{SO(3)} \doteq \{\boldsymbol{R} \in \mathbb{R}^{3\times3} | \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}_3, \det(\boldsymbol{R}) = 1\}$ and define the 2-sphere as $\mathcal{S}^2 \doteq \{\boldsymbol{t} \in \mathbb{R}^3 | \boldsymbol{t}^T\boldsymbol{t} = 1\}$.

2. Related work

We summarize here the main works regarding the relative pose (RP) estimation with known direction of rotation for planar configurations and also include some works where optimality certifiers were proposed. Our previous work Garcia-Salguero and Gonzalez-Jimenez (2023) contains a more detailed related work, which includes the solvers for generic points. We refer interested readers to the references therein and the main books for computer vision, *e.g.* Hartley and Zisserman (2003), Ma et al. (2012) and convex optimization *e.g.* Boyd and Vandenberghe (2004).

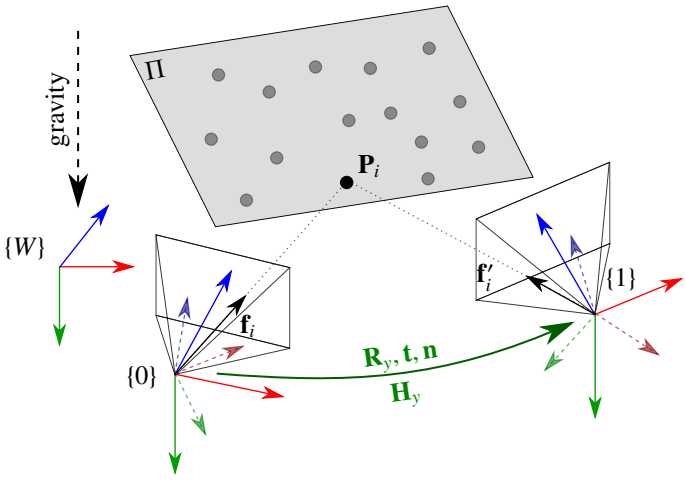

Fig. 1: In this work we aim to estimate the relative pose $(\boldsymbol{R}_Y, \boldsymbol{t})$ between cameras $\{0\} - \{1\}$ and the plane normal $\boldsymbol{n}$ from N pair-wise observations $\{\boldsymbol{f}_i, \boldsymbol{f}'_i\}$ assuming the unknown 3D points lay on a plane π. Original camera poses appear with dashed lines and the corrected ones (Y-axis aligned with the gravity vector) in solid lines.

2.1. Solvers for the RPp from Planar Observations

Given the observations of five generic points in the space we can uniquely compute the RP between the cameras that observed them, *e.g.* Nistér (2004); Stewenius et al. (2006); Kukelova et al. (2008). However, when the 3D points lay on a plane there exists a stronger relationship between the observations and only four correspondences between the images are needed to estimate the solution. This alternative approach is required for these planar scenes, since they are one of the degenerate configurations for the general solver Hartley and Zisserman (2003).

Nevertheless, when the direction of rotation is known, the number of required correspondences reduces to three, condition that is not imposed by general (planar and nonplanar) solvers. Thus, the returned solutions are not guaranteed to have the desired form when the data is corrupted by noise. For these cases, specific solvers that integrate this information should be employed instead. Saurer *et al.* in Saurer et al. (2016) propose a minimal, polynomial solver that requires only three correspondences and estimates the homography taking into account the information about the rotation. Ding *et al.* in Ding et al. (2019, 2020) improve the solver by exploiting the rank-one constraint and eliminating the rotation parameters from the resulting set, and mainly focus on (partially) uncalibrated configurations. Other solvers in the literature assume a particular configuration for the translation and/or plane normal, *e.g.* ground

plane, translation is parallel to plane or vertical plane, Ding et al. (2020); Guan et al. (2018); Wadenbäck et al. (2016). We refer the interested reader to these works and references therein for more information.

Although these solvers can be integrated into robust frameworks such as RANSAC Fischler and Bolles (1981) to detect and discard outliers, they come with a series of well-known drawbacks. First, they usually require computing a Gröbner basis and the roots of a high-order polynomial, which may become unstable and slow. Second, since they only employ the minimum number of correspondences, they are more sensitive to noise, whereas in real-world applications many correspondences are usually found, which can be employed to achieve robust pose estimations.

Nonminimal solvers do incorporate all the observations for the estimation, hence averaging and reducing the effect of the noise on the solution. The most common approach is to state a nonconvex optimization problem that minimizes certain cost function on the desired domain. This implies that the optimality of the solution cannot be guaranteed, since in general there exist many local minima. To the best of our knowledge, there don't exist in the literature non-minimal solvers for planar scenes and with a known rotation axis between the cameras.

2.2. Optimality certificate

There exist other approaches that are able to obtain and/or certify the optimality of the solution. Solvers based on branch-and-bound techniques have been proposed for the relative pose problem with generic points, see *e.g.* Kneip and Lynen (2013); Hartley and Kahl (2007) They have exponential complexity in the worst case due to their exploratory nature. An alternative technique relies on deriving a convex relaxation on the semidefinite positive cone. This relaxation can be solved by off-the-shelf solvers (SeDuMi Sturm (1999) or SDPT3 Toh et al. (1999)) in polynomial time, and under some circumstances may return the optimal solution to the original non-convex problem. The main drawback of these relaxations methods is that they may not remain tight in all problem instances. Empirically, it has been shown that increasing the number of constraints tightens the relaxations. This also increases the computational time required by the solvers, making them too slow to be used in practice.

Nonetheless, other approaches can be employed in these cases. In Bandeira (2016) Bandeira shows that faster algorithms can be leveraged to certify solutions to nonconvex problems, without actually obtaining these solutions. The potential optimal solution is estimated by any classic iterative method, and it is later certified or not as optimal, see *e.g.* Eriksson et al. (2018); Briales and Gonzalez-Jimenez (2017b); Carlone et al. (2015); Garcia-Salguero and Gonzalez-Jimenez (2021). Given the efficiency of iterative solvers, these optimality certifiers acquired special relevance in computer vision applications. These proposals compute a part of the certifier (the candidates to Lagrange multipliers) in closed-form and check whether the associated Hessian is positive semidefinite (PSD). This last step is usually the most time-consuming, especially if the matrix is dense and/or has a large size. Additionally, in order to derive these closed-form certifiers certain conditions must be fulfilled by the formulation, among them, having no more constraints than variables and the Jacobian of the constraints being full-rank at the tested solution, condition known as Linear Independence Constraint Qualification (LICQ). Failure of any of these conditions requires using another approach in general. In this work, all the formulations lack LICQ and we rely on our previous contribution in Garcia-Salguero and Gonzalez-Jimenez (2023) to certify optimality, where we proposed an iterative certifier that does not assume any condition on the problem.

3. Quadratic formulation of the RP problem

3.1. Generic problem formulation

The RPp consists of estimating the rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$ between two camera poses given a set of N pair-wise observations $(\boldsymbol{f}_i, \boldsymbol{f}'_i), i = 1, \dots, N$ originated from N unknown 3D points of the scene. For the planar configuration, we assume the 3D points lay on an unknown plane π (see Fig. 1), and therefore we can relate the observations by a full-rank 3×3 transformation, the so-called homography matrix $\boldsymbol{H} \in \mathbb{R}^{3\times3}$. Considering the normal vector to the plane $\tilde{\boldsymbol{n}} \in \mathcal{S}^2$ w.r.t. the first camera and the distance from the first camera to the plane d, the homography matrix for calibrated cameras has the form:

$$\boldsymbol{H} \sim \boldsymbol{R} + \frac{1}{d}\boldsymbol{t}\tilde{\boldsymbol{n}}^T, \tag{1}$$

where $\sim$ denotes up-to-scale. The translation $\boldsymbol{t}$ and distance d can be only estimated up to a global scale and thus we let the translation $\boldsymbol{t}$ absorbs the distance d, and the factor $\frac{1}{d}$ is omitted in the remaining of the manuscript. We further assume that the gravity vector is known, which is considered to be aligned with the global Y-axis, see figure 3. This allows us to rotate the two camera poses so that their Y-axes are also aligned with the global Y-axis, i.e. with the gravity vector. This reduces the degrees of freedom of the rotation $\boldsymbol{R}$ from three to one as we need to estimate only the angle of rotation around the new Y-axis. The new rotation, named here Y-rotation, has the form:

$$\boldsymbol{R}_\mathrm{Y} = \begin{pmatrix} c & 0 & s \\ 0 & 1 & 0 \\ -s & 0 & c \end{pmatrix} \in \mathrm{SO}(3)^\mathrm{Y} \subset \mathrm{SO}(3), \tag{2}$$

where $c, s \in \mathbb{R}$ such that $c^2 + s^2 = 1$, and $\mathrm{SO}(3)^\mathrm{Y}$ denotes the space of rotation matrices with the form in (2). Notice that we can also apply the information from the IMU by writing the 3D rotation $\boldsymbol{R}$ as $\boldsymbol{R} = \boldsymbol{R}_\mathrm{Y}\hat{\boldsymbol{R}}$ where $\hat{\boldsymbol{R}}$ is a XZ rotation that can be derived from the IMU measurements, and thus, $\boldsymbol{R}_\mathrm{Y} = \boldsymbol{R}\hat{\boldsymbol{R}}^T$. Then, the full homography matrix in eq. 1 can be written as $\boldsymbol{H} = \boldsymbol{H}_Y\hat{\boldsymbol{R}}$ with $\boldsymbol{H}_Y = \boldsymbol{R}_\mathrm{Y} + \boldsymbol{t}(\hat{\boldsymbol{R}}\tilde{\boldsymbol{n}})^T$ the homography with $\boldsymbol{R}_\mathrm{Y}$ as rotation and as normal vector $\hat{\boldsymbol{R}}\tilde{\boldsymbol{n}}$. As we only work with this form for the homography matrix, we re-define the plane normal $\boldsymbol{n} \doteq \hat{\boldsymbol{R}}\tilde{\boldsymbol{n}}$. We denote the set homography matrices with Y-rotation $\mathbb{H}^\mathrm{Y}$ by

$$\mathbb{H}^\mathrm{Y} \doteq \{\boldsymbol{H} \in \mathbb{R}^{3\times3} \,|\, \boldsymbol{H} = \boldsymbol{R}_\mathrm{Y} + \boldsymbol{t}\boldsymbol{n}^T, \boldsymbol{t} \in \mathbb{R}^3, \boldsymbol{n} \in \mathcal{S}^2, \boldsymbol{R}_\mathrm{Y} \in \mathrm{SO}(3)^\mathrm{Y}\}. \tag{3}$$

The RP problem is stated as the minimization of the geometric error over $\mathbb{H}^{\mathrm{Y}}$ Chum et al. (2005)

$$f^{\star} = \min_{\boldsymbol{H}\in\mathbb{H}^{\mathrm{Y}}} \mathrm{vec}(\boldsymbol{H})^T \boldsymbol{C}_{\mathrm{H}} \mathrm{vec}(\boldsymbol{H}) \qquad \text{(O)}$$

where $\boldsymbol{C}_{\mathrm{H}} = \sum_{i=1}^{N} (\boldsymbol{f}_i \otimes (\boldsymbol{B}_1 \boldsymbol{f}_i'))(\boldsymbol{f}_i \otimes (\boldsymbol{B}_1 \boldsymbol{f}_i'))^T + (\boldsymbol{f}_i \otimes (\boldsymbol{B}_2 \boldsymbol{f}_i'))(\boldsymbol{f}_i \otimes (\boldsymbol{B}_2 \boldsymbol{f}_i'))^T \in \mathbb{S}_+^9$, with $\boldsymbol{B}_1, \boldsymbol{B}_2 \in \mathbb{R}^{3\times 3}$ the generators

$$\boldsymbol{B}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}, \; \boldsymbol{B}_2 = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}. \qquad (4)$$

Appendix A includes this development. Notice that here, the pair-wise observations $(\boldsymbol{f}_i, \boldsymbol{f}_i')$ has been already corrected with the information about the axis of rotation. In practice, we can apply the rotation $\hat{\boldsymbol{R}}$ to the original relation $\boldsymbol{f}_i \sim \boldsymbol{H}\tilde{\boldsymbol{f}}_i'$ for general $\boldsymbol{H}$ through the relation $\boldsymbol{H} = \boldsymbol{H}_{\mathrm{Y}}\hat{\boldsymbol{R}}$ as $\boldsymbol{f}_i \sim \boldsymbol{H}_{\mathrm{Y}}(\hat{\boldsymbol{R}}\tilde{\boldsymbol{f}}_i')$, therefore considering the aligned pair-wise observations with $\boldsymbol{f}_i' = \hat{\boldsymbol{R}}\tilde{\boldsymbol{f}}_i'$ in Prob. O. Our goal is to solve problem O and obtain an optimality certificate that guarantees that the estimated solution is the global optimum.

3.2. *Sets of constraints for* $\mathbb{H}^{Y}$

In the general case, the Euclidean homography has eight degrees of freedom (DoF). Since we consider Y-rotations, the DoF are reduced to six and we need to introduce additional constraints for the solution to have the form in $\boldsymbol{H}_{\mathrm{Y}}$. In this part we aim to define the set $\mathbb{H}^{\mathrm{Y}}$ by a set of quadratic constraints, that will allow us to derive a convex relaxation for problem O.

Since the homography can be only defined up-to-scale, we consider the normalized homography matrix $\boldsymbol{H}$ with frobenius norm equals one. By fixing the scale we need to introduce the non-zero scalar $\alpha \in \mathbb{R}$ so that the equality $\boldsymbol{H} = \alpha\boldsymbol{R}_{\mathrm{Y}} + \alpha\boldsymbol{t}\boldsymbol{n}^T$ holds exactly. To simplify notation, we let the translation $\boldsymbol{t} \in \mathbb{R}^3$ absorb the unknown scale α and introduce α into the rotation matrix $\boldsymbol{R}_{\mathrm{Y}}$ with $\boldsymbol{R}_{\mathrm{Y}\alpha} \doteq \alpha\boldsymbol{R}_{\mathrm{Y}}$. The elements of $\boldsymbol{R}_{\mathrm{Y}\alpha}$ followed the same distribution as in Eq. (2), and with a little abuse of notation we denote the scaled cosine and sine of the rotation $\boldsymbol{R}_{\mathrm{Y}\alpha}$ by c, s again, and so $c^2 + s^2 = \alpha^2$. Further, to make the cost quadratic in the unknowns, we introduce the outer product $\boldsymbol{Q} = \boldsymbol{t}\boldsymbol{n}^T \in \mathbb{R}^{3\times 3}$, which makes the homography matrix linear in the unknowns $\boldsymbol{R}_{\mathrm{Y}\alpha}, \boldsymbol{Q}$, *i.e.* $\boldsymbol{H} = \boldsymbol{R}_{\mathrm{Y}\alpha} + \boldsymbol{Q}$.

We first follow the formulation in Ding et al. (2019) which exploits the rank-1 condition of $\boldsymbol{Q}$, and re-writes the equality associated with the homography as $\boldsymbol{H} - \boldsymbol{R}_{\mathrm{Y}\alpha} = \boldsymbol{Q}$. The rank-one condition of the matrix $\boldsymbol{Q}$ and thus also $\boldsymbol{H} - \boldsymbol{R}_{\mathrm{Y}\alpha}$, can be written as 9 quadratic constraints that come from all the 2×2 minors of the matrix being zero, *i.e.* all the 2×2 submatrices have determinant zero. These nine constraints, together with the Frobenius norm of $\boldsymbol{H}$ equals one and the rotation constraint of $\boldsymbol{R}_{\mathrm{Y}\alpha}$ give us 11 constraints that define the space $\mathbb{H}^{\mathrm{Y}}$ in terms of 12 variables: 9 for the homography, 2 for the rotation and 1 for the scale. We denote this set of constraints by HR, and since the homography matrix is one of the unknowns the cost function does not need to be modified. In order to accommodate the other variables, we padded the cost matrix with zeros when needed.

Problem	# variables	# constraints
HR	12	11
QR	12	11
HRQ	21	69
H-RED	21	97

Table 1: Number of variables and constraints for each definition of the set $\mathbb{H}^{\mathrm{Y}}$.

A similar formulation with the same number of constraints and variables arise by considering the expression $\boldsymbol{R}_{\mathrm{Y}\alpha} + \boldsymbol{Q}$ instead of $\boldsymbol{H} - \boldsymbol{R}_{\mathrm{Y}\alpha}$, thus removing the homography matrix $\boldsymbol{H}$ as variable. The norm constraint of $\boldsymbol{H}$ can be re-written as a quadratic form in terms of $\boldsymbol{R}_{\mathrm{Y}\alpha}$ and $\boldsymbol{Q}$:

$$\mathrm{tr}(\boldsymbol{H}^T\boldsymbol{H}) = \mathrm{tr}(\boldsymbol{R}_{\mathrm{Y}\alpha}^T\boldsymbol{R}_{\mathrm{Y}\alpha}) + \mathrm{tr}(\boldsymbol{Q}^T\boldsymbol{Q}) + 2\,\mathrm{tr}(\boldsymbol{Q}^T\boldsymbol{R}_{\mathrm{Y}\alpha}) \qquad (5)$$

The rotation constraint for $\boldsymbol{R}_{\mathrm{Y}\alpha}$ remains unchanged while now the nine rank-1 constraints only involve the matrix $\boldsymbol{Q}$, instead of $\boldsymbol{H} - \boldsymbol{R}_{\mathrm{Y}\alpha}$. This definition with 12 variables and 11 constraints is denoted by QR and since the homography is not a variable in the problem, we need to adjust the cost function to the variables $\boldsymbol{R}_{\mathrm{Y}\alpha}$ and $\boldsymbol{Q}$. Since the homography is linear in these variables, the cost function will remain quadratic, and the new expression for the cost can be derived by algebraic manipulation of $\boldsymbol{C}_{\mathrm{H}}$ (see *Supplementary material* Section A).

Last, we aim to find a redundant formulation for the set $\mathbb{H}^{\mathrm{Y}}$. We combine all the previous constraints in QR and HR to obtain a redundant formulation with $(H)9+(R)3+(Q)9 = 21$ variables. Further, from the relation $\boldsymbol{H} = \boldsymbol{R}_{\mathrm{Y}\alpha}+\boldsymbol{Q}$ we can derive additional quadratics constraints by multiplying $\boldsymbol{H}$ with the other variables $\boldsymbol{R}_{\mathrm{Y}\alpha}$ and $\boldsymbol{Q}$, for example, $\boldsymbol{H}\boldsymbol{H} = \boldsymbol{R}_{\mathrm{Y}\alpha}\boldsymbol{H} + \boldsymbol{Q}\boldsymbol{H}$. *Supplementary material* Section B shows all the employed expressions. In total we obtain 97 quadratic constraints, which defined the set denoted by H-RED, and from them we select a subset of 69 constraints, denoted by HRQ, which empirically appears to perform well in terms of tightness. This subset allows us to show the dependence of the proposed iterative certifier with the number of constraints, since both definitions H-RED and HRQ have 21 variables. The different formulations with the number of variables and constraints are found in Table 1.

4. Certifiable approach to the RPp

Since the problem and constraints for all the formulations are quadratic in the variables, problem O can be written as a Quadratically Constrained Quadratic Program (QCQP), which in general are NP-hard to solve. However, under some conditions we can derive convex relaxations that approximate well the original problem and even obtain its optimal solution. Among the different options, we leverage the relaxations with the form of semidefinite problems (SDP), which can be solved by off-the-shelf tools in polynomial time Sturm (1999), Toh et al. (1999). In this work, though, we aim to solve the problem in a more efficient way. Instead of solving the relaxation from scratch, we seek an optimality certificate that only certifies a given solution. However, two of our four formulations have more constraints than variables that precludes the

used of closed-form certifiers, and all the sets contain the rank-deficient constraint, that defines what is known as the Segre variety. From Bruns and Schwänzl (1990) we know that this set is defined by 9 equations. However, the dimension of the space is 5 and the Jacobian of the constraints has rank 4 Hauenstein et al. (2012). This puts an upper bound on the rank of the full Jacobian, which makes the LICQ condition fail even for the minimal HR and QR. A similar argument holds for the nonminimal formulations, as from (Decker and Schreyer, 2007, Th. 4.1.12) the rank of the Jacobian has as upper bound the difference between the number of variables 21 and the dimension of the space, which is 5 for all formulations. We empirically observe LICQ is not fulfilled for all problem instances. Hence, we leverage our previous proposal in Garcia-Salguero and Gonzalez-Jimenez (2023) and adapt it to this case. In summary, we employ an iterative algorithm to obtain the solution (homography) to problem O (section 4.1) and then employ the iterative certifier (section 4.2) that indicates whether or not the obtained solution is the global optimum.

4.1. Homography matrix estimation

We estimate the homography matrix by optimizing on its manifold, procedure that allows us to decouple the problem (model), the domain and the solver. For the latter we employ the (second order) Trust region solver (TNT)Boyd and Vandenberghe (2004), which has been shown to provide good convergence properties, *e.g.* Briales and Gonzalez-Jimenez (2017a); Garcia-Salguero and Gonzalez-Jimenez (2021). For the domain we employ the representation of the homography matrix in terms of rotation, translation and normal vector, as $\mathrm{SO}(3)^{\mathrm{Y}} \times \mathbb{R}^3 \times \mathcal{S}^2$. This representation has two main advantages: first, the three domains are well-known and well-studied, *e.g.* Absil et al. (2009); and second, the solution to the optimization is already a valid pose, which eliminates the decomposition of the homography matrix into its elements, process which we observe is unstable and prone to errors, see *e.g.* Malis and Vargas (2007). Last, the employed solver requires second-order information about the model, that is, gradient and (vector-product) Hessian. Since we consider the domain as the product of manifolds, we can define the gradient and Hessian separately for each variable. We refer the reader to Section C in the Appendices for further details about the domain operators and the second-order model.

4.2. Iterative optimality certifier

We include in this section the main aspects about the iterative certifier, and refer the reader to the original work in Garcia-Salguero and Gonzalez-Jimenez (2023) for further details. The algorithm relies on the dual problem associated to the primal in Problem O Boyd and Vandenberghe (2004). To derive the dual problem, let us consider the general form of the primal problem O under the set of m constraints $\boldsymbol{x}^T\boldsymbol{A}_i\boldsymbol{x} = c_i$ for $i = 1, \dots, m$ with $\boldsymbol{x} \in \mathbb{R}^n$ the vector with all the unknowns, and $\boldsymbol{A}_i \in \mathbb{S}^n, c_i \in \mathbb{R}$ the matricial form of the constraints. Similarly, we express the cost $\mathrm{vec}(\boldsymbol{H})^T\boldsymbol{C}_{\mathrm{H}}\mathrm{vec}(\boldsymbol{H})$ as $\boldsymbol{x}^T\boldsymbol{C}\boldsymbol{x}$ for $\boldsymbol{C} \in \mathbb{S}^n_+$. The primal is then re-written as

$$f^\star = \min_{\boldsymbol{x}\in\mathbb{R}^n} \boldsymbol{x}^T\boldsymbol{C}\boldsymbol{x} \text{ subject to } \boldsymbol{x}^T\boldsymbol{A}_i\boldsymbol{x} = c_i, i = 1, \dots, m. \qquad \text{(P)}$$

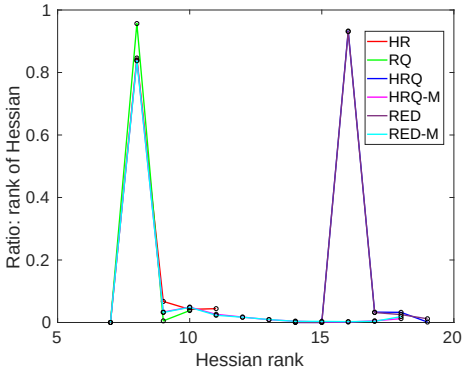

Fig. 2: Rank of the Hessian during the evaluation of the certifier in Section 5 for the different formulations (see legend), including those with different cost matrices.

The standard dual problem has the form

$$d^\star = \max_{\boldsymbol{\lambda}\in\mathbb{R}^m} \boldsymbol{c}^T\boldsymbol{\lambda} \text{ subject to } \boldsymbol{C} - \sum_{i=1}^m \lambda_i\boldsymbol{A}_i \succeq 0, \qquad \text{(D)}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ are the Lagrange multipliers, $\boldsymbol{c} \doteq [c_1, \dots, c_m]^T \in \mathbb{R}^m$ and $\boldsymbol{S} \doteq \boldsymbol{C} - \sum_{i=1}^m \lambda_i\boldsymbol{A}_i \succeq 0$ is the Hessian of the Lagrangian. Section D includes the development of problem D. Since the dual is a relaxation of the primal, we have that

$$d(\boldsymbol{\lambda}) \leq d^\star \leq f^\star \leq f(\boldsymbol{x}) \qquad (6)$$

for any feasible primal $\boldsymbol{x}$ and dual $\boldsymbol{\lambda}$ points, *i.e.* points feasible for their respective problems

The certifier tries to find a feasible dual point associated with a given homography matrix (a primal solution) assuming strong duality holds, that is, $d^\star = f^\star$. In practice, the certification solves the problem

$$g^\star = \min_{\boldsymbol{\lambda}\in\mathbb{R}^m, \boldsymbol{S}\in\mathbb{S}^n_+} \|\boldsymbol{S}\boldsymbol{x}\|_2^2 + \left\|\boldsymbol{S} - \boldsymbol{C} + \sum_{i=1}^m \lambda_i\boldsymbol{A}_i\right\|_{\mathrm{F}}^2 \qquad \text{(C)}$$

We solve the problem by introducing the decomposition $\boldsymbol{S} = \boldsymbol{Y}\boldsymbol{Y}^T$ for $\boldsymbol{Y} \in \mathbb{R}^{n\times k}$, which makes $\boldsymbol{S}$ PSD by construction, but turns the problem non-convex. However, contrary to the original RPp problem, we know that if the cost $g^\star$ for (C) is zero (up to some accuracy), then: (a) the primal solution is the global optimum; (b) the found dual solution is the optimum for the dual problem; and (c) strong duality holds for the given problem instance. In practice, we consider that the certification is positive if the cost normalized by the number of variables (*normalized cost*) is below $1e-9$.

An interesting peculiarity of the employed certifier is its dependence of the cost of the problem. We observe in Garcia-Salguero and Gonzalez-Jimenez (2023) and here that the certifier empirically tends to the Hessian with minimum rank. We also notice that changing the expression of the cost (and its rank) leads to different Hessians and performances. While the form of the cost cannot be modified for most problems, it is possible for the problem at hand. For the formulation HR the cost

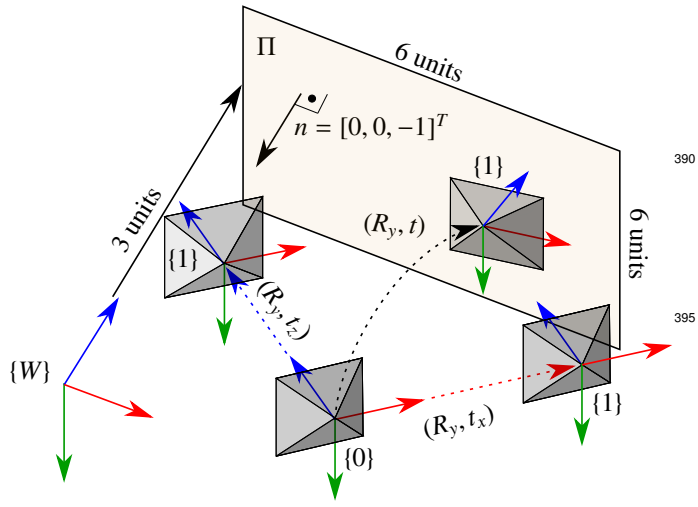

Fig. 3: Configuration of the points, cameras and parameters for the considered problem instances of the RPp for the simulated dataset. The lateral configuration has $\boldsymbol{t}_x$ as translation whereas the forward has $\boldsymbol{t}_z$.

acts on the homography $\boldsymbol{H}_{\mathrm{Y}}$, while the part for the rotation is zero. However, for QR the cost is dense since it depends on both $\boldsymbol{Q}, \boldsymbol{R}_{\mathrm{Y}}$. Therefore, for the redundant formulations HRQ and H-RED that have as variables $\boldsymbol{H}_{\mathrm{Y}}, \boldsymbol{Q}, \boldsymbol{R}_{\mathrm{Y}}$, we can obtain different combinations of the above-mentioned costs, making them sparse or dense. These redundant problems have 21 variables, and so the cost matrix has size 21×21. If we *only* employ the cost associated with $\boldsymbol{H}_{\mathrm{Y}}$ the cost matrix has several entries to zero. Further, the cost in terms of $\boldsymbol{H}_{\mathrm{Y}}$ has at most rank nine and at least rank eight for non-degenerate problem instances, which is also the rank for the 21×21 matrix. When the cost associated with $\boldsymbol{Q}, \boldsymbol{R}_{\mathrm{Y}}$ is included together with the one for $\boldsymbol{H}_{\mathrm{Y}}$, the rank of $\boldsymbol{C}$ increases up to 16 (maximum 21). Fig. 2 depicts the rank of the Hessian in our experiments in section 5.1 for the different formulations and costs: HRQ-M and H-RED-M are the formulations whose cost matrices have rank at most nine, and HRQ and H-RED the formulations with ranks up to 21. These results show that, in this case as well, the rank of the Hessian follows the one of the cost matrix. We therefore include these formulations on our evaluation since the computational time required by each of them during the certification may differ.

5. Evaluation

In this last section we evaluate the proposed pipeline for certification and the on-manifold estimation of the homography matrix. Our solver OURS-H and the initialization DLT-H are compared against the minimal solver in Sweeney et al. (2014), denoted by MIN. This minimal solver was devised for the essential matrix Saurer et al. (2016), although it was reported to perform similarly well even for the planar case. Since the one devised for the homography matrix is not publicly available, we use the former instead. To the best of our knowledge, there is no non-minimal solver for this problem published in the literature. Since MIN is minimal, we expect it to attain larger errors although it may be faster than the proposed iterative algorithm and so it is included here only to provide a baseline to our solver. For both synthetic and real data we run the solver on ten random samples of three features each and keep the solution with the lowest cost among them. We consider that all the correspondences are inliers, even when high noise is applied and thus we do not use RANSAC at this stage. The data for the real experiments have been previously filtered with the provided ground-truth pose. Last, notice that we use DLT-H method as initialization for our solver as it provides pose and plane parameters. Further, the result obtained with DLT-H is only an approximation and it is included to show the accuracy of the method as initial guess. On the other hand, the minimal MIN only returns the pose not the plane parameters, and therefore cannot be used to initialize our algorithm that estimates both.

5.1. Evaluation on synthetic data

We generate a set of random N world points lying on a plane with dimension 6×6 units. The plane has normal vector parallel to the world Z-axis and distance 3 units. We compute a random pose for the first camera w.r.t. the world reference with maximum angle of rotation 0.5 rad and 2 units for the translation magnitude. Since the homography matrix considers the relative normal and distance w.r.t. the first camera frame, this random pose makes the parameters associated to the plane different for each problem instance. The second camera pose is defined by a Y-rotation with maximum angle 0.5 rad and three configurations for the relative translation: (a) general; (b) forward; and (c) lateral. Additionally, we generate configurations with zero translation with: (1) points on a plane; and (2) points on a general position. Figure 3 shows the plane π with its parameters w.r.t. the world frame $\{\boldsymbol{W}\}$, and the two camera poses under general translation $\boldsymbol{t}$, forward translation $\boldsymbol{t}_z$ and lateral translation $\boldsymbol{t}_x$. We compute the observations by assuming a pin-hole camera model and perturb them adding Gaussian noise with deviation σ on the image plane considering a focal length of 512 pixels. We also consider perturbation on the relative rotation by pre-multiplying a random rotation with angle $\theta_{\boldsymbol{R}} \leq 0.01[rad]$ to the original Y-rotation. For the different setups we also vary the number of correspondences from $N \in \{10, 15, 50, 100, 200\}$ and noise $\sigma \in \{0.0, 0.5, 1.5, 3.0\}$`pix`, and for each type of configuration and set of parameters, we generate 300 random problem instances. Since we maintain the focal length fixed, varying the noise changes the Signal-to-noise ratio.

Error in rotation: Figure 4 shows the rotation error in degrees for the different solvers w.r.t. the ground-truth. The first row shows the errors for those cases with pure Y-rotation and the second those with noisy Y-rotation. From left to right, we have the problems with (1) general motion; (2) lateral motion along the X-axis; (3) forward motion along the Z-axis; (4) zero translation with points on plane; and (5) zero translation with points on general position. The X-axis indicates the level of noise applied to the correspondences and we fix the number of correspondences to $N = 50$. As expected the error w.r.t. the ground truth increases with the level of noise for all the solvers, while the minimal MIN attains the largest errors in most problem instances since it only considers a subset of the available data. Notice that for configurations with zero translation, the

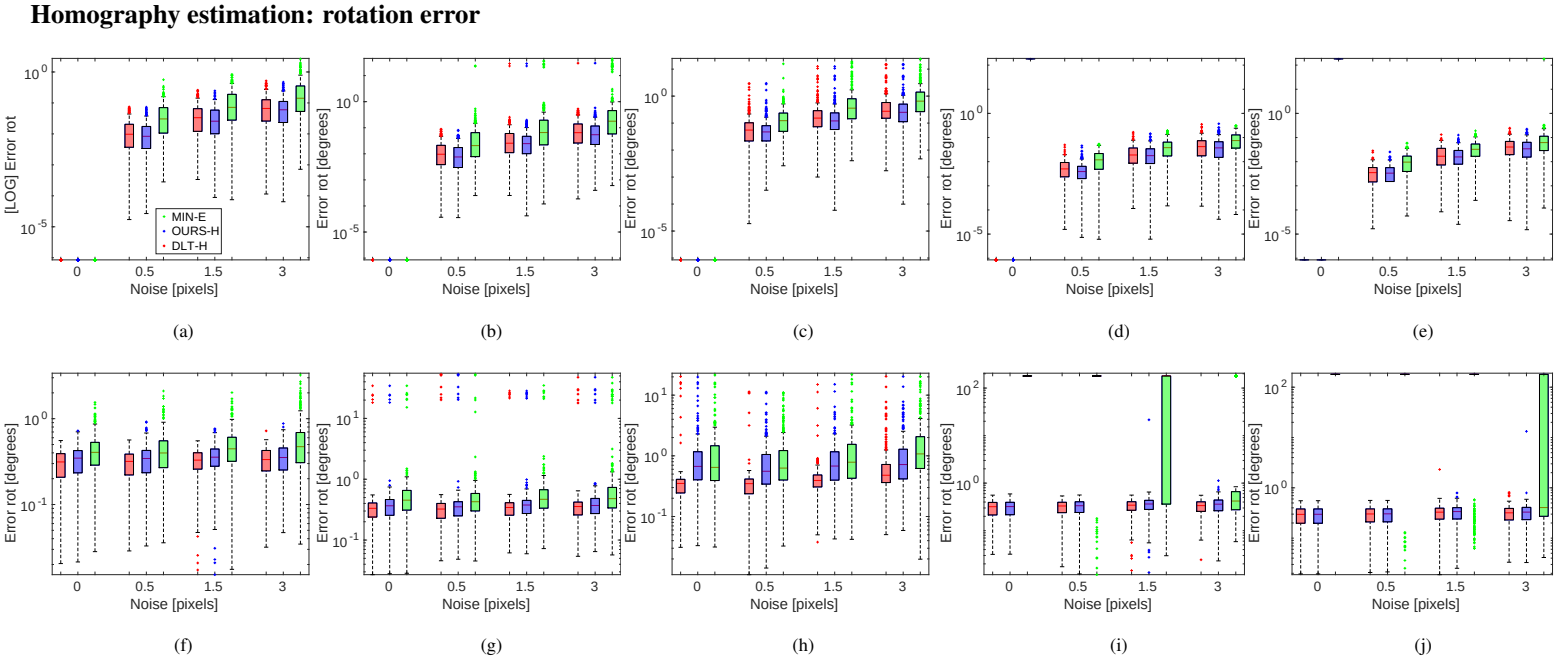

Fig. 4: **Homography estimation: rotation error in degrees (log-scale)**: Rotation error for the solutions returned by each solver in degrees (see legend). Top row shows the experiments with pure Y-rotation and bottom row where a small perturbation was added. From left to right: general motion (random translation); lateral motion (translation along the X-axis); forward motion (along the Z-axis); zero translation with points on plane; and zero translation with general points. The number of correspondences is fixed to $N = 50$. Notice the difference in the Y-scales.

error for the minimal solver is large until the noise applied to the observations is high enough to break the degeneracy, whereas for noisy Y-rotations the errors are large and independent of the noise for all the solvers. Errors in translation follow the same tendency, and we don't include them due to space limits.

Computational time: We conclude this part of the evaluation with the computational time required by the iterative estimation of the homography matrix. Table 2 includes the mean and median computational time for all the experiments with all the noise levels and number of correspondences. The reported time for OURS-H includes also the coefficient matrix creation, whereas the value for MIN considers only the time required to obtain the best solution. Notice that in general for highly noisy data the iterative refinement may take longer since the initial guess lays far from the optimum.

5.1.1. Results for the iterative certifier

Figure 5 shows the normalized cost for the certifier for $N = 50$ correspondences, following the format of figure 4. The ratio of certified solutions is included in the *Supplementary material* section F. The small formulation HR performs worse than all the other options, although it is able to certify optimality in most problem instances even for forward motions. Configurations with zero translation and general points hinder specially the performance of the HR certifier, and those with zero translation and planar points also affect QR, and for noisy Y-rotation this difference is emphasized. In general the minimal formulations detect fewer optimal solutions than the redundant ones which return optimality certificates even for large noise and low number of correspondences, highlighting the importance of this type of certifiers for these challenging scenarios.

Solver	$\boldsymbol{R}_Y$	$\boldsymbol{R}_Y\Delta\boldsymbol{R}$
MIN	7.03	7.52
DLT-H	49.819	58.3444
OURS-H	169.8	261.4

Table 2: Computational times (in microseconds) required by the different solvers for the configurations with pure Y-rotation $\boldsymbol{R}_Y$ and noisy Y-rotation $\boldsymbol{R}_Y\Delta\boldsymbol{R}$.

Computational time: For problem instances with an Y-rotation as ground-truth, the certifier only requires one iteration to return an optimality certificate for all the formulations. With zero translation, this number increases up to 3 for HR, 2 for QR and 1.5 for the redundant HRQ, whereas the median is one for all configurations and formulations. Table 3 shows the mean and median times for the different formulations for all the noise levels and number of correspondences. Notice that HR and QR require similar times since both have the same number of constraints (11) and rank of the Hessian (8). The redundant, however, with 97 constraints and rank 16 requires approximately ten times more computational time than the minimal formulations, although it remains circa 7 milliseconds. The problem instances with noisy Y-rotation require more computational time. HR requires more iterations, thus consuming more time than the similar formulation QR, while the redundant formulations follow the the tendency that was previously seen. As a baseline the SDP in H-RED implemented in MATLAB with CVX Grant and Boyd (2014) as modeling tool and SDPT3 Toh et al. (1999) as IPM requires from 0.5 to 1 second per problem instance.

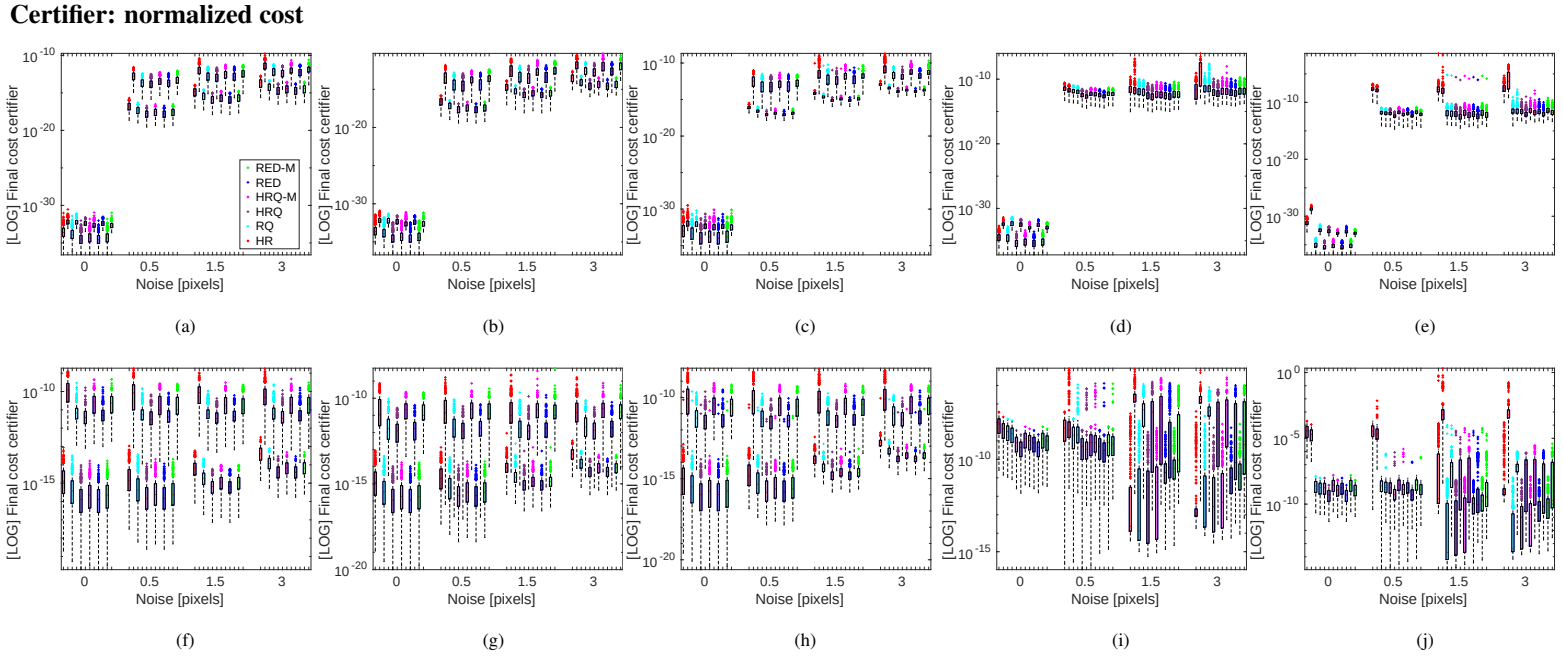

Fig. 5: **Certifier: normalized cost** Normalized cost of the certifier for the different formulations (see legend) and level of noise (X-axis). We also include the cost for the initialization with the same color, on the left of the final cost. Top row shows the experiments with pure Y-rotation and bottom row the experiments where a small perturbation was added to the rotation. From left to right: general motion (random translation); lateral motion (translation along the X-axis); forward motion (along the Z-axis); zero translation with points on plane; and zero translation with general points. The number of correspondences is fixed to $N = 50$. Notice the difference in the Y-scales. We consider the certification positive if the cost is below $1e-9$

Fig. 6: Figures 6a-6d show images from the real datasets, from left to right: TUM-FAR, TUM-NEAR, OURS-SHORT and OURS-LONG and Figure 6e depicts the cost for the associated essential matrix attained by the different solvers for the same sequences, including those certified as optimal OURS-OPT.

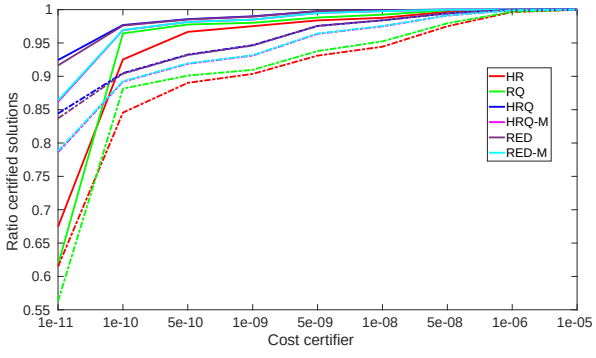

Fig. 7: Ratio of certified solution for first-order critical points (solid line) and all solutions (dashed line) for the different formulations for all the sequences.

5.2. *Evaluation on real data*

To conclude this section we evaluate our proposal on the real dataset TUM Sturm et al. (2012), sequences FR3/NOSTRUCTURE_TEXTURE_NEAR, denoted by TUM-NEAR and FR3/NOSTRUCTURE_TEXTURE_FAR, denoted by TUM-FAR, both publicly available and on our own two sequences which were recorded with a mobile phone. These last two sequences, denoted by OURS-SHORT and OURS-LONG, show two different posters placed on the floor, and hence contain a strong plane with texture. For the last two sequences we run the COLMAP Schönberger and Frahm (2016) to obtain the ground-truth poses. We include some of the images where we evaluate our algorithm in figure 6, from left to right: TUM-FAR, TUM-NEAR, OURS-SHORT and OURS-LONG.

For all the sequences, we extract and match URF features Bay et al. (2008) from consecutive images at least 5 cm apart, and keep only those images with more than 10 corre-

Rotation	Problem	Mean	Median
$\boldsymbol{R}_Y$	HR	0.851	0.768
	QR	0.881	0.763
	HRQ	6.3879	6.141
	HRQ-M	4.358	3.886
	H-RED	8.4163	8.4165
	H-RED-M	6.358	5.605
$\boldsymbol{R}_Y\Delta\boldsymbol{R}$	HR	3.379	2.970
	QR	2.8	1.59
	HRQ	11.07	6.161
	HRQ-M	13.2	6.895
	H-RED	18.326	8.410
	H-RED-M	19.03	8.768

Table 3: Computational times (in milliseconds) required by the certifier for each formulation for the configurations with pure Y-rotation $\boldsymbol{R}_Y$ and noisy Y-rotation $\boldsymbol{R}_Y\Delta\boldsymbol{R}$.

spondences. We don't contemplate wrong correspondences (outliers) on the data and filter the matches with the provided ground truth before feeding them to the solvers. Nevertheless, the proposed solvers are suitable for robust non-minimal paradigms, such as LO-SAC Chum et al. (2003). Last, we extract an approximate Y-rotation from the ground-truth rotation $\boldsymbol{R}$ by considering the first entry of the first row of the ground-truth rotation matrix as the cosine and the third entry of the first row as the sine. We consider the rotation formed by these values as the 'ground-truth' Y-rotation $\boldsymbol{R}_Y$ and use the XZ rotation $\hat{\boldsymbol{R}}$ given by $\hat{\boldsymbol{R}} = \boldsymbol{R}_Y^T\boldsymbol{R}$ to 'correct' the observations. Notice that this process carries numerical errors due to the rounding of the original rotation, approximation of the Y-rotation, transformation of the observations, etc

Figure 6e shows the cost of the returned solution for the different sequences by all the solvers, considering also only those solutions certified as optimal. For comparison, this cost only involves the rotation and normalized translation, *i.e.* the cost in terms of the essential matrix. Last, we observe a remarkable detail on these sequences in terms of the percentage of optimal certificates. Recall that the initialization of the homography matrix is estimated through the linear method that takes the eigenvector associated with the least eigenvalue of the data matrix. This is the general procedure, that is, it does not include the rotation constraint. Further, the algorithm outputs a homography matrix, from which the pose and plane equation are retrieved, in order to later project the general 3D rotation matrix to a 2D rotation. Recovering the pose from the homography is not a trivial task and several works have tackled it, see *e.g.* Malis and Vargas (2007) and references therein. Given these factors we observe that the initialization is not stable and different algorithms give different poses from the same homography matrix. However and to the best of our knowledge this is the only publicly available initialization, since MIN doesn't provide with the normal vector $\boldsymbol{n}$. In turn this implies that the iterative refinement of this initial guess may get stuck in local minima and/or in a solution that destabilizes the solver, that is, the solution is not a (first-order) critical point. Out of the 59139 problem instances tested the solution was a critical point for 53876 (91.1%). Figure 7 shows the ratio of certified solutions considering only critical points (solid lines) and for all solutions, including non-critical solutions (dashed lines). We notice fewer certified solutions when considering noncritical points independently of the formulation.

6. Conclusions and future work

This paper tackles the relative pose problem between two calibrated cameras with known gravity prior, assuming the 3D points that originated the observations belonged to an unknown plane π. These scenes, predominant in man-made environments, are degenerate configurations for approaches that rely on the essential matrix. Instead, in these cases the relative pose is retrieved through the homography matrix. We proposed a certifiable solver for this problem that first estimated the relative pose with an iterative algorithm and then, tried to certify this solution as the global optimum. We stated the homography estimation in terms of the relative pose and plane normal, and solved the problem on the associated manifold. The certification step relied on the dual problem, a convex relaxation that had been shown to depend on the employed formulation. Therefore, we proposed four different definitions of the search space that lead to certifiers with a different performance. Since the formulations lacked the Linear Independence Constraint Qualification and two of them had more constraints than variables, we employed our recent work for certification Garcia-Salguero and Gonzalez-Jimenez (2023). Our evaluation showed that the iterative algorithm returned solutions with the lowest cost, and that the smallest formulations certified most of the solutions. Redundant formulations certified all solutions, even for highly noisy problem instances. We made the code publicly available at `https://github.com/mergarsal`.

In future work, we envision introducing this solver into a robust paradigm and use the proposal in conjunction with essential matrix-based algorithms.

Acknowledgments

This work has been supported by the grant program FPU18/01526 and the research projects ARPEGGIO (PID2020-117057) and HOUNDBOT (UMA20-FEDERJA-056) funded by the Spanish and Andalusian Governments, respectively, and European Regional Development Fund (ERDF).

References

Absil, P.A., Mahony, R., Sepulchre, R., 2009. Optimization algorithms on matrix manifolds. Princeton University Press.

Bandeira, A.S., 2016. A note on probably certifiably correct algorithms. Comptes Rendus Mathematique 354, 329–333.

Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008. Speeded-up robust features (surf). Computer vision and image understanding 110, 346–359.

Boyd, S., Vandenberghe, L., 2004. Convex optimization. Cambridge university press.

Briales, J., Gonzalez-Jimenez, J., 2017a. Cartan-sync: Fast and global se (d)-synchronization. IEEE Robotics and Automation Letters 2, 2127–2134.

Briales, J., Gonzalez-Jimenez, J., 2017b. Convex global 3d registration with lagrangian duality, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4960–4969.

Bruns, W., Schwänzl, R., 1990. The number of equations defining a determinantal variety. Bulletin of the London Mathematical Society 22, 439–445.

Carlone, L., Rosen, D.M., Calafiore, G., Leonard, J.J., Dellaert, F., 2015. Lagrangian duality in 3d slam: Verification techniques and optimal solutions, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE. pp. 125–132.

Chum, O., Matas, J., Kittler, J., 2003. Locally optimized ransac, in: Pattern Recognition: 25th DAGM Symposium, Magdeburg, Germany, September 10-12, 2003. Proceedings 25, Springer. pp. 236–243.

Chum, O., Pajdla, T., Sturm, P., 2005. The geometric error for homographies. Computer Vision and Image Understanding 97, 86–102.

Decker, W., Schreyer, F.O., 2007. Varieties, groebner bases, and algebraic curves. To appear .

Ding, Y., Barath, D., Kukelova, Z., 2020. Homography-based egomotion estimation using gravity and sift features, in: Proceedings of the Asian Conference on Computer Vision.

Ding, Y., Yang, J., Ponce, J., Kong, H., 2019. An efficient solution to the homography-based relative pose problem with a common reference direction, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE. pp. 1655–1664.

Eriksson, A., Olsson, C., Kahl, F., Chin, T.J., 2018. Rotation averaging and strong duality, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 127–135.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24, 381–395.

Garcia-Salguero, M., Gonzalez-Jimenez, J., 2021. Fast and robust certifiable estimation of the relative pose between two calibrated cameras. Journal of Mathematical Imaging and Vision , 1–21.

Garcia-Salguero, M., Gonzalez-Jimenez, J., 2023. Fast certifiable relative pose estimation with gravity prior. Artificial Intelligence URL: http://mapir.isa.uma.es/papersrepo/2023/2023_mercedes_AI_priorRPp_doc.pdf, doi:https://doi.org/10.1016/j.artint.2023.103862.

Grant, M., Boyd, S., 2014. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx.

Guan, B., Vasseur, P., Demonceaux, C., Fraundorfer, F., 2018. Visual odometry using a homography formulation with decoupled rotation and translation estimation using minimal solutions, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 2320–2327.

Hartley, R., Zisserman, A., 2003. Multiple view geometry in computer vision. Cambridge university press.

Hartley, R.I., Kahl, F., 2007. Global optimization through searching rotation space and optimal estimation of the essential matrix, in: 2007 IEEE 11th International Conference on Computer Vision, IEEE. pp. 1–8.

Hauenstein, J., Rodriguez, J., Sturmfels, B., 2012. Maximum likelihood for matrices with rank constraints. arXiv preprint arXiv:1210.0198 .

Kneip, L., Lynen, S., 2013. Direct optimization of frame-to-frame rotation, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 2352–2359.

Kukelova, Z., Bujnak, M., Pajdla, T., 2008. Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems., in: BMVC, p. 2008.

Ma, Y., Soatto, S., Kosecka, J., Sastry, S.S., 2012. An invitation to 3-d vision: from images to geometric models. volume 26. Springer Science & Business Media.

Malis, E., Vargas, M., 2007. Deeper understanding of the homography decomposition for vision-based control. Ph.D. thesis. INRIA.

Nistér, D., 2004. An efficient solution to the five-point relative pose problem. IEEE transactions on pattern analysis and machine intelligence 26, 756–770.

Saurer, O., Vasseur, P., Boutteau, R., Demonceaux, C., Pollefeys, M., Fraundorfer, F., 2016. Homography based egomotion estimation with a common direction. IEEE transactions on pattern analysis and machine intelligence 39, 327–341.

Schönberger, J.L., Frahm, J.M., 2016. Structure-from-motion revisited, in: Conference on Computer Vision and Pattern Recognition (CVPR).

Stewenius, H., Engels, C., Nistér, D., 2006. Recent developments on direct relative orientation. ISPRS Journal of Photogrammetry and Remote Sensing 60, 284–294.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D., 2012. A benchmark for the evaluation of rgb-d slam systems, in: Proc. of the International Conference on Intelligent Robot Systems (IROS).

Sturm, J.F., 1999. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. Optimization methods and software 11, 625–653.

Sweeney, C., Flynn, J., Turk, M., 2014. Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem, in: 2014 2nd International Conference on 3D Vision, IEEE. pp. 483–490.

Toh, K.C., Todd, M.J., Tütüncü, R.H., 1999. Sdpt3—a matlab software package for semidefinite programming, version 1.3. Optimization methods and software 11, 545–581.

Wadenbäck, M., Åström, K., Heyden, A., 2016. Recovering planar motion from homographies obtained using a 2.5-point solver for a polynomial system, in: 2016 IEEE International Conference on Image Processing (ICIP), IEEE. pp. 2966–2970.

Appendices

Summary

This document includes details about the matrix form for the cost, the set of constraints employed in the main manuscript, the on-manifold optimization for the homography matrix and further results for our evaluation.

A. Cost function for the homography-based RPp

In this section we derive the cost function for the Relative Pose problem (RPp) based on the homography matrix $\boldsymbol{H}$ in section A.1, and in terms of the rotation $\boldsymbol{R}_{\mathrm{Y}}$ and the outer product $\boldsymbol{Q} = \boldsymbol{t}\boldsymbol{n}^T$ in section A.2.

A.1. Cost function in terms of homography matrix

The homography matrix imposes a point-to-point relation between corresponding observations as $\boldsymbol{f}_i \sim \boldsymbol{H}\boldsymbol{f}'_i$, where $\sim$ indicates equality up-to-scale. To remove the scale ambiguity, we take the cross-product as $\boldsymbol{f}_i \times \boldsymbol{H}\boldsymbol{f}'_i = \boldsymbol{0}_{3\times1}$, which gives three expressions, although only two of them are algebraically independent. For simplicity, we re-write the cross product $\boldsymbol{f}_i\times$ as the skew-symmetric matrix $[\boldsymbol{f}_i]_{\mathrm{x}} \in \mathbb{R}^{3\times3}$ with form

$$\begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix} \tag{7}$$

for $\boldsymbol{f}_i \doteq [a, b, c]^T$. The columns of $[\boldsymbol{f}_i]_{\mathrm{x}}$ are linear in $\boldsymbol{f}_i$ and we express them as $\boldsymbol{B}_k\boldsymbol{f}_i \in \mathbb{R}^{1\times3}$ for $k = 1, 2, 3$ and $\boldsymbol{B}_k \in \mathbb{R}^{3\times3}$, that is, $[\boldsymbol{f}_i]_{\mathrm{x}} = [\boldsymbol{B}_1\boldsymbol{f}_i \mid \boldsymbol{B}_2\boldsymbol{f}_i \mid \boldsymbol{B}_3\boldsymbol{f}_i]$. The matrices $\boldsymbol{B}_k$ are called generators and have the form

$$\boldsymbol{B}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix},\ \boldsymbol{B}_2 = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{8}$$

With these, we write the i-th entry of $\boldsymbol{f}_i \times \boldsymbol{H}\boldsymbol{f}'_i$ as the generic form $\epsilon^i_k \doteq (-\boldsymbol{B}_k\boldsymbol{f}_i)^T\boldsymbol{H}\boldsymbol{f}'_i$, where the unknown is the homography matrix $\boldsymbol{H}$ and we use the fact that the i-th row of $[\boldsymbol{f}_i]_{\mathrm{x}}$ equals the i-th column with a minus sign.

In this manuscript we treat the values ϵ^i_k as residuals and minimize the sum of the squared values, that is, ${\epsilon^i_1}^2 + {\epsilon^i_2}^2$ where we discard ϵ^i_3 for being dependent. In order to express this cost in its matricial form, we first re-formulate the error ϵ^i_k as

$$\epsilon^i_k = (-\boldsymbol{B}_k\boldsymbol{f}_i)^T\boldsymbol{H}\boldsymbol{f}'_i = -(\boldsymbol{f}'^T_i \otimes (\boldsymbol{B}_k\boldsymbol{f}_i)^T)\mathrm{vec}(\boldsymbol{H}) = -(\boldsymbol{f}'_i \otimes \boldsymbol{B}_k\boldsymbol{f}_i)^T\mathrm{vec}(\boldsymbol{H}) \tag{9}$$

with $\mathrm{vec}(\boldsymbol{H}) \in \mathbb{R}^9$ the column-wise vectorization of $\boldsymbol{H}$ and we use the identity $\mathrm{tr}(\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}) = (\boldsymbol{B}^T \otimes \boldsymbol{A})\mathrm{vec}(\boldsymbol{X})$.

To obtain the squared error, we consider that $r^2 = r^T r$ for $r \in \mathbb{R}$ and thus ${\epsilon^i_k}^2 = {\epsilon^i_k}^T\epsilon^i_k = \mathrm{vec}(\boldsymbol{H})^T(\boldsymbol{f}'_i \otimes \boldsymbol{B}_k\boldsymbol{f}_i)(\boldsymbol{f}'_i \otimes \boldsymbol{B}_k\boldsymbol{f}_i)^T\mathrm{vec}(\boldsymbol{H})$. Finally, the contribution to the total cost by the i-th pair of correspondences is

$$f_i = \mathrm{vec}(\boldsymbol{H})^T\Big((\boldsymbol{f}'_i \otimes \boldsymbol{B}_1\boldsymbol{f}_i)(\boldsymbol{f}'_i \otimes \boldsymbol{B}_1\boldsymbol{f}_i)^T + (\boldsymbol{f}'_i \otimes \boldsymbol{B}_2\boldsymbol{f}_i)(\boldsymbol{f}'_i \otimes \boldsymbol{B}_2\boldsymbol{f}_i)^T\Big)\mathrm{vec}(\boldsymbol{H}) \tag{10}$$

and the total cost is the sum of each contribution, *i.e.*

$$f(\boldsymbol{H}) = \sum_{i=1}^{N} \mathrm{vec}(\boldsymbol{H})^T\Big((\boldsymbol{f}'_i \otimes \boldsymbol{B}_1\boldsymbol{f}_i)(\boldsymbol{f}'_i \otimes \boldsymbol{B}_1\boldsymbol{f}_i)^T + (\boldsymbol{f}'_i \otimes \boldsymbol{B}_2\boldsymbol{f}_i)(\boldsymbol{f}'_i \otimes \boldsymbol{B}_2\boldsymbol{f}_i)^T\Big)\mathrm{vec}(\boldsymbol{H}) = \tag{11}$$

$$= \mathrm{vec}(\boldsymbol{H})^T \underbrace{\Big(\sum_{i=1}^{N}(\boldsymbol{f}'_i \otimes \boldsymbol{B}_1\boldsymbol{f}_i)(\boldsymbol{f}'_i \otimes \boldsymbol{B}_1\boldsymbol{f}_i)^T + (\boldsymbol{f}'_i \otimes \boldsymbol{B}_2\boldsymbol{f}_i)(\boldsymbol{f}'_i \otimes \boldsymbol{B}_2\boldsymbol{f}_i)^T\Big)}_{\boldsymbol{C}_{\mathrm{H}}}\mathrm{vec}(\boldsymbol{H}) \tag{12}$$

A.2. Cost function in terms of rotation and outer product

Consider the cost function for a homography $\boldsymbol{H}$ as $f(\boldsymbol{H}) = \mathrm{vec}(\boldsymbol{H})^T\boldsymbol{C}_{\mathrm{H}}\mathrm{vec}(\boldsymbol{H})$ and recall that $\boldsymbol{H} = \boldsymbol{R}_{\mathrm{Y}\alpha} + \boldsymbol{Q}$. Since $\boldsymbol{R}_{\mathrm{Y}\alpha}$ is an Y-rotation with form

$$\boldsymbol{R}_{\mathrm{Y}\alpha} = \begin{pmatrix} c & 0 & s \\ 0 & \alpha & 0 \\ -s & 0 & c \end{pmatrix} \tag{13}$$

we can simplify the expression of its vectorization as $\text{vec}(\boldsymbol{R}_{\text{Y}\alpha}) = \boldsymbol{P}\boldsymbol{r}$ by introducing the 3D vector $\boldsymbol{r} = [c, s, \alpha]^T \in \mathbb{R}^3$ and the matrix

$$\boldsymbol{P} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{9\times 3}. \tag{14}$$

Now we can re-write the cost as

$$\begin{aligned} \text{vec}(\boldsymbol{H})^T \boldsymbol{C}_{\text{H}} \text{vec}(\boldsymbol{H}) &= \text{vec}(\boldsymbol{R}_{\text{Y}\alpha} + \boldsymbol{Q})^T \boldsymbol{C}_{\text{H}} \text{vec}(\boldsymbol{R}_{\text{Y}\alpha} + \boldsymbol{Q}) = & (15)\\ &= \text{vec}(\boldsymbol{R}_{\text{Y}\alpha})^T \boldsymbol{C}_{\text{H}} \text{vec}(\boldsymbol{R}_{\text{Y}\alpha}) + \text{vec}(\boldsymbol{Q})^T \boldsymbol{C}_{\text{H}} \text{vec}(\boldsymbol{Q}) + 2\text{vec}(\boldsymbol{Q})^T \boldsymbol{C}_{\text{H}} \text{vec}(\boldsymbol{R}_{\text{Y}\alpha}) = & (16)\\ &= \boldsymbol{r}^T \underbrace{\boldsymbol{P}^T \boldsymbol{C}_{\text{H}} \boldsymbol{P}}_{\boldsymbol{C}_R} \boldsymbol{r} + \text{vec}(\boldsymbol{Q})^T \boldsymbol{C}_{\text{H}} \text{vec}(\boldsymbol{Q}) + 2\text{vec}(\boldsymbol{Q})^T \underbrace{\boldsymbol{C}_{\text{H}} \boldsymbol{P}}_{\boldsymbol{C}_{QR}} \boldsymbol{r}, & (17) \end{aligned}$$

where the matrices $\boldsymbol{C}_R \in \mathbb{S}^3_+, \boldsymbol{C}_{QR} \in \mathbb{R}^{9\times 3}$ are defined as

$$\boldsymbol{C}_R \doteq \begin{pmatrix} \boldsymbol{C}_{\text{H}1,1} + \boldsymbol{C}_{\text{H}1,9} + \boldsymbol{C}_{\text{H}9,1} + \boldsymbol{C}_{\text{H}9,9} & \boldsymbol{C}_{\text{H}1,7} - \boldsymbol{C}_{\text{H}1,3} - \boldsymbol{C}_{\text{H}3,9} + \boldsymbol{C}_{\text{H}7,9} & \boldsymbol{C}_{\text{H}1,5} + \boldsymbol{C}_{\text{H}5,9} \\ \boldsymbol{C}_{\text{H}7,1} - \boldsymbol{C}_{\text{H}3,1} - \boldsymbol{C}_{\text{H}9,3} + \boldsymbol{C}_{\text{H}9,7} & \boldsymbol{C}_{\text{H}3,3} - \boldsymbol{C}_{\text{H}3,7} - \boldsymbol{C}_{\text{H}7,3} + \boldsymbol{C}_{\text{H}7,7} & \boldsymbol{C}_{\text{H}5,7} + \boldsymbol{C}_{\text{H}5,3} \\ \boldsymbol{C}_{\text{H}9,5} + \boldsymbol{C}_{\text{H}5,1} & \boldsymbol{C}_{\text{H}7,5} - \boldsymbol{C}_{\text{H}3,5} & \boldsymbol{C}_{\text{H}5,5} \end{pmatrix} \in \mathbb{S}^3_+ \tag{18}$$

and

$$\boldsymbol{C}_{QR} \doteq \begin{pmatrix} \boldsymbol{C}_{\text{H}1,:} + \boldsymbol{C}_{\text{H}9,:} \\ \boldsymbol{C}_{\text{H}7,:} - \boldsymbol{C}_{\text{H}3,:} \\ \boldsymbol{C}_{\text{H}5,:} \end{pmatrix} \in \mathbb{R}^{9\times 3} \tag{19}$$

being $\boldsymbol{C}_{\text{H}i,j}$ the (i, j)-th entry of $\boldsymbol{C}_{\text{H}}$ and $\boldsymbol{C}_{\text{H}k,:}$ the k-th row of $\boldsymbol{C}_{\text{H}}$, indices starting at one.

B. Set of constraints for the homography matrix

This section lists the constraints used for each formulation. The minimal one HR is formed by

$$\begin{aligned} &\text{Nr. constraints: 9} && \text{rank}(\boldsymbol{H}_{\text{Y}} - \boldsymbol{R}_{\text{Y}\alpha}) = 1 && (20)\\ &\text{Nr. constraints: 1} && \text{tr}(\boldsymbol{H}_{\text{Y}}^T \boldsymbol{H}_{\text{Y}}) = 1 && (21)\\ &\text{Nr. constraints: 1} && c^2 + s^2 = \alpha, && (22) \end{aligned}$$

providing us with 11 constraints on 12 variables $(\boldsymbol{H}_{\text{Y}}, \boldsymbol{R}_{\text{Y}})$.

The second minimal formulation QR is formed by

$$\begin{aligned} &\text{Nr. constraints: 9} && \text{rank}(\boldsymbol{Q}) = 1 && (23)\\ &\text{Nr. constraints: 1} && \text{tr}\left((\boldsymbol{Q} + \boldsymbol{R}_{\text{Y}\alpha})^T (\boldsymbol{Q} + \boldsymbol{R}_{\text{Y}\alpha})\right) = 1 && (24)\\ &\text{Nr. constraints: 1} && c^2 + s^2 = \alpha, && (25) \end{aligned}$$

with 11 constraints and 12 variables $(\boldsymbol{Q}, \boldsymbol{R}_{\text{Y}\alpha})$.

For the redundant formulations, recall that the homography matrix has the form $\boldsymbol{H}_{\text{Y}} = \boldsymbol{R}_{\text{Y}\alpha} + \boldsymbol{Q}$, where $\boldsymbol{Q}$ is a rank-1 matrix and $\boldsymbol{R}_{\text{Y}\alpha} = \alpha \boldsymbol{R}_{\text{Y}}$ is the α-scaled of the Y-rotation $\boldsymbol{R}_{\text{Y}}$. We leverage the following relations, which follow from the definition of $\boldsymbol{H}_{\text{Y}}$ given

above:

$$\text{Nr. constraints: 9} \quad \boldsymbol{H}_{\mathrm{Y}}\boldsymbol{Q}^T = \boldsymbol{R}_{\mathrm{Y}\alpha}\boldsymbol{Q}^T + \boldsymbol{Q}\boldsymbol{Q}^T \tag{26}$$

$$\text{Nr. constraints: 9} \quad \boldsymbol{H}_{\mathrm{Y}}\boldsymbol{Q} = \boldsymbol{R}_{\mathrm{Y}\alpha}\boldsymbol{Q} + \boldsymbol{Q}\boldsymbol{Q} \tag{27}$$

$$\text{Nr. constraints: 9} \quad \boldsymbol{Q}^T\boldsymbol{H}_{\mathrm{Y}} = \boldsymbol{Q}^T\boldsymbol{R}_{\mathrm{Y}\alpha} + \boldsymbol{Q}^T\boldsymbol{Q} \tag{28}$$

$$\text{Nr. constraints: 9} \quad \boldsymbol{Q}\boldsymbol{H}_{\mathrm{Y}} = \boldsymbol{Q}\boldsymbol{R}_{\mathrm{Y}\alpha} + \boldsymbol{Q}\boldsymbol{Q} \tag{29}$$

$$\text{Nr. constraints: 9} \quad \boldsymbol{H}_{\mathrm{Y}}\boldsymbol{R}_{\mathrm{Y}\alpha}^T = \boldsymbol{R}_{\mathrm{Y}\alpha}\boldsymbol{R}_{\mathrm{Y}\alpha}^T + \boldsymbol{Q}\boldsymbol{R}_{\mathrm{Y}\alpha}^T \tag{30}$$

$$\text{Nr. constraints: 9} \quad \boldsymbol{H}_{\mathrm{Y}}\boldsymbol{R}_{\mathrm{Y}\alpha} = \boldsymbol{R}_{\mathrm{Y}\alpha}\boldsymbol{R}_{\mathrm{Y}\alpha} + \boldsymbol{Q}\boldsymbol{R}_{\mathrm{Y}\alpha} \tag{31}$$

$$\text{Nr. constraints: 9} \quad \boldsymbol{R}_{\mathrm{Y}\alpha}^T\boldsymbol{H}_{\mathrm{Y}} = \boldsymbol{R}_{\mathrm{Y}\alpha}^T\boldsymbol{R}_{\mathrm{Y}\alpha} + \boldsymbol{R}_{\mathrm{Y}\alpha}^T\boldsymbol{Q} \tag{32}$$

$$\text{Nr. constraints: 9} \quad \boldsymbol{R}_{\mathrm{Y}\alpha}\boldsymbol{H}_{\mathrm{Y}} = \boldsymbol{R}_{\mathrm{Y}\alpha}\boldsymbol{R}_{\mathrm{Y}\alpha} + \boldsymbol{R}_{\mathrm{Y}\alpha}\boldsymbol{Q} \tag{33}$$

$$\text{Nr. constraints: 9} \quad \boldsymbol{H}_{\mathrm{Y}}\boldsymbol{H}_{\mathrm{Y}}^T = \boldsymbol{R}_{\mathrm{Y}\alpha}\boldsymbol{H}_{\mathrm{Y}}^T + \boldsymbol{Q}\boldsymbol{H}_{\mathrm{Y}}^T \tag{34}$$

$$\text{Nr. constraints: 9} \quad \boldsymbol{H}_{\mathrm{Y}}^T\boldsymbol{H}_{\mathrm{Y}} = \boldsymbol{H}_{\mathrm{Y}}^T\boldsymbol{R}_{\mathrm{Y}\alpha} + \boldsymbol{H}_{\mathrm{Y}}^T\boldsymbol{Q} \tag{35}$$

$$\text{Nr. constraints: 9} \quad \boldsymbol{H}_{\mathrm{Y}}\boldsymbol{H}_{\mathrm{Y}} = \boldsymbol{R}_{\mathrm{Y}\alpha}\boldsymbol{H}_{\mathrm{Y}} + \boldsymbol{Q}\boldsymbol{H}_{\mathrm{Y}} \tag{36}$$

$$\text{Nr. constraints: 1} \quad \mathrm{tr}\left((\boldsymbol{H}_{\mathrm{Y}} - \boldsymbol{Q})(\boldsymbol{H}_{\mathrm{Y}} - \boldsymbol{Q})^T\right) = \mathrm{tr}(\boldsymbol{R}_{\mathrm{Y}\alpha}\boldsymbol{R}_{\mathrm{Y}\alpha}^T) = 3\alpha^2 \tag{37}$$

$$\text{Nr. constraints: 6} \quad (\boldsymbol{H}_{\mathrm{Y}} - \boldsymbol{Q})(\boldsymbol{H}_{\mathrm{Y}} - \boldsymbol{Q})^T = \boldsymbol{R}_{\mathrm{Y}\alpha}\boldsymbol{R}_{\mathrm{Y}\alpha}^T = \alpha^2\boldsymbol{I}_3 \tag{38}$$

$$\text{Nr. constraints: 6} \quad (\boldsymbol{H}_{\mathrm{Y}} - \boldsymbol{Q})^T(\boldsymbol{H}_{\mathrm{Y}} - \boldsymbol{Q}) = \boldsymbol{R}_{\mathrm{Y}\alpha}^T\boldsymbol{R}_{\mathrm{Y}\alpha} = \alpha^2\boldsymbol{I}_3 \tag{39}$$

Out of the 112 constraints, 36 are linear dependent. The set H-RED is formed by: (a) these 76 constraints; (b) the set HR with elevn (eqs. (20),(21),(22)); and (c) the set QR with nine (eqs. (23),(24)).

For the redundant formulation HRQ, we include those in: HR with eleven; QR with ten; nine from eq. (36) ($\boldsymbol{H}_{\mathrm{Y}}\boldsymbol{H}_{\mathrm{Y}}$); nine from eq. (27) ($\boldsymbol{H}_{\mathrm{Y}}\boldsymbol{Q}$); nine from eq. (29) ($\boldsymbol{Q}\boldsymbol{H}_{\mathrm{Y}}$); nine from eq. (30) ($\boldsymbol{H}_{\mathrm{Y}}\boldsymbol{R}_{\mathrm{Y}\alpha}^T$); nine from eq. (32) ($\boldsymbol{R}_{\mathrm{Y}\alpha}^T\boldsymbol{H}_{\mathrm{Y}}$); nine from eq. (31) ($\boldsymbol{H}_{\mathrm{Y}}\boldsymbol{R}_{\mathrm{Y}\alpha}$); and nine from eq (33) ($\boldsymbol{R}_{\mathrm{Y}\alpha}\boldsymbol{H}_{\mathrm{Y}}$). Out of the 85 constraints 16 are linear dependent.

C. On-manifold estimation of the homography matrix

In this section we include the necessary information for the on-manifold estimation of the homography matrix, which includes: the required operators associated with the geometry of the manifold (domain) and the quadratic model.

C.1. Domain

For the optimization we define the set as the product of manifolds $\mathbb{R}^3 \times \mathcal{S}^2 \times \mathrm{SO}(3)^{\mathrm{Y}}$, which allows us to tackle each component separately. There are four operators that have to be defined for the domain: retraction, projection from tangent space, gradient and (vector-product) Hessian. These functions are independent of the problem and can be found in the literature, *e.g.* Absil et al. (2009). We include them here for completeness, and refer the interested readers to this reference. Roughly, these operators allow us to pass from the Euclidean space where all the operations are computed to the Riemmanian space when needed.

Euclidean For the Euclidean space we do not have to take into account any additional operation, meaning that the retraction, projection, gradient and Hessian remain unchanged, *i.e.*, identity.

Sphere We employ a projection-based retraction defined as:

$$\mathrm{Retr}_{\boldsymbol{t}}(\boldsymbol{u}) = \frac{\boldsymbol{t} + \boldsymbol{u}}{\|\boldsymbol{t} + \boldsymbol{u}\|} \tag{40}$$

where $\boldsymbol{t} \in \mathbb{R}^3$ is a point on the manifold and $\boldsymbol{u} \in \mathbb{R}^3$ is a point of the tangent space.

With the same notation, the tangent space projector is

$$\mathrm{P}_{\boldsymbol{t}}(\boldsymbol{u}) = \boldsymbol{u} - \boldsymbol{t}^T\boldsymbol{t}\boldsymbol{u}. \tag{41}$$

The Riemmanian gradient at a point $\boldsymbol{t} \in \mathbb{R}^3$ on the manifold is obtained by projecting the Euclidean gradient $\nabla \mathrm{f}_{\boldsymbol{t}}(\boldsymbol{t})$ as

$$\mathrm{grad}\ \mathrm{f}_{\boldsymbol{t}}(\boldsymbol{t}) = \mathrm{P}_{\boldsymbol{t}}(\nabla \mathrm{f}_{\boldsymbol{t}}(\boldsymbol{t})). \tag{42}$$

Last, the Riemmanian vector-product Hessian depends on the point $\boldsymbol{t}$ on the manifold, the Euclidean gradient $\nabla \mathrm{f}_{\boldsymbol{t}}(\boldsymbol{t})$ and the Euclidean vector-product Hessian $\nabla^2 f_{\boldsymbol{t}}(\boldsymbol{t})[\boldsymbol{u}]$. It is computed as

$$\mathrm{Hess}\ \mathrm{f}_{\boldsymbol{t}}(\boldsymbol{t})[\boldsymbol{u}] = \mathrm{P}_{\boldsymbol{t}}(\nabla^2 f_{\boldsymbol{t}}(\boldsymbol{t})[\boldsymbol{u}]) - \boldsymbol{t}^T\nabla \mathrm{f}_{\boldsymbol{t}}(\boldsymbol{t})\boldsymbol{u} \tag{43}$$

Rotation The projection-like retraction is given by:

$$\text{Retr}_{\boldsymbol{R}}(\boldsymbol{Y}) = \boldsymbol{U}\boldsymbol{V}^T, \quad \boldsymbol{R} + \boldsymbol{Y} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T \tag{44}$$

where $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ is a point on the manifold and $\boldsymbol{Y} \in \mathbb{R}^{3\times3}$ is a point of the tangent space. To guarantee that the solution is a rotation matrix, we check the determinant of $\boldsymbol{U}\boldsymbol{V}^T$. If it is negative, we multiply the last column of $\boldsymbol{U}$ by -1.

The tangent space projector is

$$\text{P}_{\boldsymbol{R}}(\boldsymbol{Y}) = \boldsymbol{Y} - \boldsymbol{R}\text{symm}(\boldsymbol{R}^T\boldsymbol{Y}) \tag{45}$$

where symm(•) takes the symmetric part of the argument, and again $\boldsymbol{R}$ is a rotation matrix and $\boldsymbol{Y}$ is a matrix on the ambient space.

The Riemmanian gradient at a point $\boldsymbol{R}$ is obtained by projecting the Euclidean gradient $\nabla \text{f}_{\boldsymbol{R}}(\boldsymbol{R})$

$$\text{grad f}_{\boldsymbol{R}}(\boldsymbol{R}) = \text{P}_{\boldsymbol{R}}(\nabla \text{f}_{\boldsymbol{R}}(\boldsymbol{R})). \tag{46}$$

Last, the Riemmanian Hessian depends on the point $\boldsymbol{R}$, the Euclidean gradient $\nabla \text{f}_{\boldsymbol{R}}(\boldsymbol{R})$ and the Euclidean vector-product Hessian $\nabla^2 f_{\boldsymbol{R}}(\boldsymbol{R})[\boldsymbol{Y}]$. It is computed as

$$\text{Hess f}_{\boldsymbol{R}}(\boldsymbol{R})[\boldsymbol{Y}] = \text{P}_{\boldsymbol{R}}\Big(\nabla^2 f_{\boldsymbol{R}}(\boldsymbol{R})[\boldsymbol{Y}] - \boldsymbol{Y}\text{symm}(\boldsymbol{R}^T\nabla \text{f}_{\boldsymbol{R}}(\boldsymbol{R}))\Big) \tag{47}$$

C.2. Euclidean quadratic model

We provide next the Euclidean, quadratic model in terms of the components $\boldsymbol{R}, \boldsymbol{t}$ and $\boldsymbol{n}$. Only for this model, we consider the matrix $\boldsymbol{R}$ as the 3D extension of the 2D rotation, which allows the next development to be extended to any kind of 3D rotation, *e.g.* Y-rotation.

As in section A we can write the cost function in terms of the matrix $\boldsymbol{Q}$ and the 3D rotation $\boldsymbol{R}$.

$$\text{vec}(\boldsymbol{H})^T\boldsymbol{C}_{\text{H}}\text{vec}(\boldsymbol{H}) = \text{vec}(\boldsymbol{R})^T\boldsymbol{C}_{\text{H}}\text{vec}(\boldsymbol{R}) + \text{vec}(\boldsymbol{Q})^T\boldsymbol{C}_{\text{H}}\text{vec}(\boldsymbol{Q}) + 2\text{vec}(\boldsymbol{Q})^T\boldsymbol{C}_{\text{H}}\text{vec}(\boldsymbol{R}). \tag{48}$$

Let us define the matrices $\boldsymbol{N} = \boldsymbol{n} \otimes \boldsymbol{I}_3 \in \mathbb{R}^{9\times3}$ and $\boldsymbol{T} = \boldsymbol{I}_3 \otimes \boldsymbol{t} \in \mathbb{R}^{9\times3}$, such that $\text{vec}(\boldsymbol{Q}) = \boldsymbol{N}\boldsymbol{t} = \boldsymbol{T}\boldsymbol{n} \in \mathbb{R}^9$. Equivalently, the column-wise vectorization of $\boldsymbol{R}$ is $\boldsymbol{r}$.

The Euclidean gradient for each component has the form:

$$\nabla \text{f}_{\boldsymbol{R}}(\boldsymbol{R}) = 2\boldsymbol{C}_{\text{H}}\boldsymbol{r} + 2\boldsymbol{C}_{\text{H}}\boldsymbol{T}\boldsymbol{n} \quad \nabla \text{f}_{\boldsymbol{t}}(\boldsymbol{t}) = 2\boldsymbol{N}^T\boldsymbol{C}_{\text{H}}\boldsymbol{N}\boldsymbol{t} + 2\boldsymbol{N}^T\boldsymbol{C}_{\text{H}}\boldsymbol{r} \quad \nabla \text{f}_{\boldsymbol{n}}(\boldsymbol{n}) = 2\boldsymbol{T}^T\boldsymbol{C}_{\text{H}}\boldsymbol{T}\boldsymbol{n} + 2\boldsymbol{T}^T\boldsymbol{C}_{\text{H}}\boldsymbol{r}. \tag{49}$$

The Hessian vector-product reads

$$\nabla^2 f_{\boldsymbol{R}}(\boldsymbol{R})[\boldsymbol{V}_{\boldsymbol{R}}, \boldsymbol{V}_{\boldsymbol{t}}, \boldsymbol{V}_{\boldsymbol{n}}] = 2\boldsymbol{C}_{\text{H}}\text{vec}(\boldsymbol{V}_{\boldsymbol{R}}) + 2\boldsymbol{C}_{\text{H}}\boldsymbol{N}\boldsymbol{V}_{\boldsymbol{t}} + 2\boldsymbol{C}_{\text{H}}\boldsymbol{T}\boldsymbol{V}_{\boldsymbol{n}} \tag{50}$$

$$\nabla^2 f_{\boldsymbol{t}}(\boldsymbol{t})[\boldsymbol{V}_{\boldsymbol{R}}, \boldsymbol{V}_{\boldsymbol{t}}, \boldsymbol{V}_{\boldsymbol{n}}] = 2\boldsymbol{N}^T\boldsymbol{C}_{\text{H}}\text{vec}(\boldsymbol{V}_{\boldsymbol{r}}) + 2\boldsymbol{N}^T\boldsymbol{C}_{\text{H}}\boldsymbol{N}\boldsymbol{V}_{\boldsymbol{t}} + \boldsymbol{M}_{\text{t;n}}\boldsymbol{V}_{\boldsymbol{n}} \tag{51}$$

$$\nabla^2 f_{\boldsymbol{n}}(\boldsymbol{n})[\boldsymbol{V}_{\boldsymbol{R}}, \boldsymbol{V}_{\boldsymbol{t}}, \boldsymbol{V}_{\boldsymbol{n}}] = 2\boldsymbol{T}^T\boldsymbol{C}_{\text{H}}\text{vec}(\boldsymbol{V}_{\boldsymbol{r}}) + \boldsymbol{M}_{\text{t;n}}^T\boldsymbol{V}_{\boldsymbol{t}} + 2\boldsymbol{T}^T\boldsymbol{C}_{\text{H}}\boldsymbol{T}\boldsymbol{V}_{\boldsymbol{n}} \tag{52}$$

with the matrix $\boldsymbol{M}_{\text{t;n}} \doteq 4\boldsymbol{N}^T\boldsymbol{C}_{\text{H}}\boldsymbol{T} + 2\text{mat}(\boldsymbol{r}^T\boldsymbol{C}_{\text{H}}) \in \mathbb{R}^{3\times3}$, and mat(•) reshape the argument into a 3×3 matrix.

D. Dual problem and iterative certifier outline

This section includes the standard development of the dual problem and the basic aspects about the iterative certifier employed in this work, see Garcia-Salguero and Gonzalez-Jimenez (2023) for further details.

D.1. Standard dual problem

To formulate the original problem in (O) we first introduce the vector $\boldsymbol{x} \in \mathbb{R}^n$ with all the unknowns that appear in the formulation. Thus, for HR we stack the column-wise vectorization of $\boldsymbol{H}$ and $\text{vec}(\boldsymbol{R}_{\text{Y}\alpha}) = [c, s, \alpha]^T$ and for QR we use $\boldsymbol{Q} \in \mathbb{R}^9$ and $\text{vec}(\boldsymbol{R}_{\text{Y}\alpha})$. For the redundant formulations HRQ and H-RED we stack $\boldsymbol{H}, \boldsymbol{Q}, \text{vec}(\boldsymbol{R}_{\text{Y}\alpha})$. The cost is written in terms of this vector as $\text{vec}(\boldsymbol{H})^T\boldsymbol{C}_{\text{H}}\text{vec}(\boldsymbol{H}) = \boldsymbol{x}^T\boldsymbol{C}\boldsymbol{x}$. For HR, HRQ and H-RED the matrix $\boldsymbol{C} \in \mathbb{R}^n$ has $\boldsymbol{C}_{\text{H}}$ is its top-left corner and zero elsewhere. For QR we use the equivalent form given in section A, which is also used for the formulations denoted by HRQ-M and H-RED-M. Equivalently, the constraints for all the formulations are quadratic in their respective unknowns and can be written in the general form $\boldsymbol{x}^T\boldsymbol{A}_i\boldsymbol{x} = c_i$ for $i = 1, \dots m$ with $\boldsymbol{A}_i \in \mathbb{S}^n$ and $c_i \in \mathbb{R}$. This also includes the rank-one condition of $\boldsymbol{Q}$ and $\boldsymbol{H} - \boldsymbol{R}_{\text{Y}\alpha}$, which we express as the nine 2×2 minors of the matrix being zero. The primal (O) is re-formulated as a Quadratically Constrained Quadratic Problem (QCQP) in its standard form

$$f^\star = \min_{\boldsymbol{x}\in\mathbb{R}^n} \boldsymbol{x}^T\boldsymbol{C}\boldsymbol{x} \text{ subject to } \boldsymbol{x}^T\boldsymbol{A}_i\boldsymbol{x} = c_i, i = 1, \dots, m. \tag{53}$$

The dual problem is defined as Boyd and Vandenberghe (2004)

$$d^\star = \max_{\boldsymbol{\lambda}\in\mathbb{R}^m} \min_{\boldsymbol{x}\in\mathbb{R}^n} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}), \tag{54}$$

where $\boldsymbol{\lambda} \doteq [\lambda_1, \dots \lambda_m]^T \in \mathbb{R}^m$ are the Lagrange multipliers (one for each constraint) and $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ is the *Lagrangian*:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \doteq \boldsymbol{x}^T\boldsymbol{C}\boldsymbol{x} + \sum_{i=1}^{m} \lambda_i(c_i - \boldsymbol{x}^T\boldsymbol{A}_i\boldsymbol{x}) = \boldsymbol{x}^T(\boldsymbol{C} - \sum_{i=1}^{m} \lambda_i\boldsymbol{A}_i)\boldsymbol{x} + \boldsymbol{\lambda}^T\boldsymbol{c}, \tag{55}$$

with $\boldsymbol{c} \doteq [c_1, \dots, c_m]^T \in \mathbb{R}^m$. The inner optimization $\min_{\boldsymbol{x}\in\mathbb{R}^n} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ is the so-called *dual function*, and from the expression of the Lagrangian, we see the dual function doesn't have a finite minimum except when the matrix $\boldsymbol{S} \doteq \boldsymbol{C} - \sum_{i=1}^{m} \lambda_i\boldsymbol{A}_i \in \mathbb{S}^n$ is positive semidefinite (PSD). In that case, the minimum is achieved at $\boldsymbol{x}^\star$ such that $\boldsymbol{x}^{\star T}\boldsymbol{S}\boldsymbol{x}^\star = 0$ and the optimal cost of the minimization is $\boldsymbol{\lambda}^T\boldsymbol{c}$. Since we are interested only oi finite values, we restrict our analysis to this case, which allows us to re-formulate the dual problem as

$$d^\star = \max_{\boldsymbol{\lambda}\in\mathbb{R}^m} \boldsymbol{c}^T\boldsymbol{\lambda} \text{ subject to } \boldsymbol{C} - \sum_{i=1}^{m} \lambda_i\boldsymbol{A}_i \succeq 0, \tag{D}$$

which is convex by construction and an instance of a Semidefinite Positive Problem, which can be solved with off-the-shelf tools. The dual problem is a relaxation of the primal and therefore the optimal, dual cost $d^\star$ is a lower bound on the optimal, primal cost $f^\star$, formally:

$$d(\boldsymbol{\lambda}) \leq d^\star \leq f^\star \leq f(\boldsymbol{x}) \tag{56}$$

for any feasible primal $\boldsymbol{x}$ and dual $\boldsymbol{\lambda}$ points. We say that *strong duality* holds if $d^\star = f^\star$, and therefore there exists (at least) one dual feasible point that attains the same cost than the primal solution.

E. Outline of the iterative certifier

The employed certifier, as the ones available in the literature, aims to find a dual, feasible solution $\boldsymbol{\lambda}$ such that its associated cost is equal (up to some tolerance) to the cost attained by the given primal solution. If such point can be found, then we conclude that: (1) strong duality holds from $d^\star = f^\star$; (2) the dual point is the optimal solution to the dual problem; and (3) the given, primal solution is the optimal for the original, nonconvex problem.

In order to find this dual point, we assume strong duality holds and therefore, $d^\star = \boldsymbol{\lambda}^T\boldsymbol{c} = f^\star = \boldsymbol{x}^T\boldsymbol{C}\boldsymbol{x}$. Further, from the dual function we know that the optimal primal solution fulfills the condition $\boldsymbol{x}^T\boldsymbol{S}\boldsymbol{x} = 0$. Since the Hessian $\boldsymbol{S}$ is PSD by definition (the dual point is feasible), then $\boldsymbol{x}^T\boldsymbol{S}\boldsymbol{x} = 0 \Leftrightarrow \boldsymbol{S}\boldsymbol{x} = \boldsymbol{0}_{n\times 1}$. This last condition implies the first one while it provides with a direct relation between the primal $\boldsymbol{x}$ and dual $\boldsymbol{\lambda}$ solutions. Therefore, we seek the Lagrange multiplier $\boldsymbol{\lambda}$ that fulfills

$$\boldsymbol{S}\boldsymbol{x} = \boldsymbol{0}_{n\times 1} \text{ and } \boldsymbol{S} \succeq 0. \tag{57}$$

Since the Hessian $\boldsymbol{S}$ depends linearly on the vector $\boldsymbol{\lambda}$, we can re-write the expression $\boldsymbol{S}\boldsymbol{x} = \boldsymbol{0}_{n\times 1}$ as $\boldsymbol{J}(\boldsymbol{x})\boldsymbol{\lambda} = \boldsymbol{C}\boldsymbol{x}$, where $\boldsymbol{J}(\boldsymbol{x}) \in \mathbb{R}^{n\times m}$ is the Jacobian of the constraints $\{\boldsymbol{A}_i\}, i = 1, \dots, m$ evaluated at the primal point $\boldsymbol{x}$. Notice that we can obtain an unique solution $\boldsymbol{\lambda}$ in closed-form from this relation *provided*: (1) the number of constraints m is at most the number of variables n; and (2) the Jacobian $\boldsymbol{J}(\boldsymbol{x})$ is full rank (LICQ). If at least one of the conditions is not met, then there may exist a family of optimal, dual solutions. The formulations leveraged in this work lack LICQ and two of them have $m > n$, and thus, we rely on the iterative certifier in Garcia-Salguero and Gonzalez-Jimenez (2023) since it does not require these conditions.

The proposal states the certification as a feasibility problem that tries to find $\boldsymbol{\lambda} \in \mathbb{R}^m$ such that conditions (57) are met, without estimating $\boldsymbol{\lambda}$ in closed-form. To solve the problem, we minimize the errors $\|\boldsymbol{S}\boldsymbol{x}\|_2^2$ and $\left\|\boldsymbol{S} - \boldsymbol{C} + \sum_{i=1}^{m} \lambda_i\boldsymbol{A}_i\right\|_{\mathrm{F}}^2$ and we conclude the certification is positive if the optimal cost of this problem is zero (in practice, below $1e-09$). To solve the problem efficiently, we decompose the matrix $\boldsymbol{S} = \boldsymbol{Y}\boldsymbol{Y}^T$ with $\boldsymbol{Y} \in \mathbb{R}^{n\times r}$ being the low-rank decomposition of $\boldsymbol{S}$ with rank r.

$$g^\star = \min_{\boldsymbol{\lambda}\in\mathbb{R}^m, \boldsymbol{Y}\in\mathbb{R}^{n\times r}} \left\|\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{x}\right\|_2^2 + \left\|\boldsymbol{Y}\boldsymbol{Y}^T - \boldsymbol{C} + \sum_{i=1}^{m} \lambda_i\boldsymbol{A}_i\right\|_{\mathrm{F}}^2 \tag{C}$$

The problem is nonconvex and so there may exist more than one local minimum. However, we know that the optimal cost should be zero which allows us to discard the potential local minima. The problem is solved on the associated manifold that raises from the equivalence between points $\boldsymbol{Y}\boldsymbol{Y} = (\boldsymbol{Y}\boldsymbol{O})(\boldsymbol{Y}\boldsymbol{O})^T$ for all orthogonal matrices $\boldsymbol{O} \in \mathbb{R}^{r\times r}$. With the current implementation, this requires the value of r to be exact, that is, $\boldsymbol{Y}$ has to be full-rank during the process. Therefore, this rank is dynamically adjusted during the optimization to maintain the completeness of the space.

Homography estimation: translation error

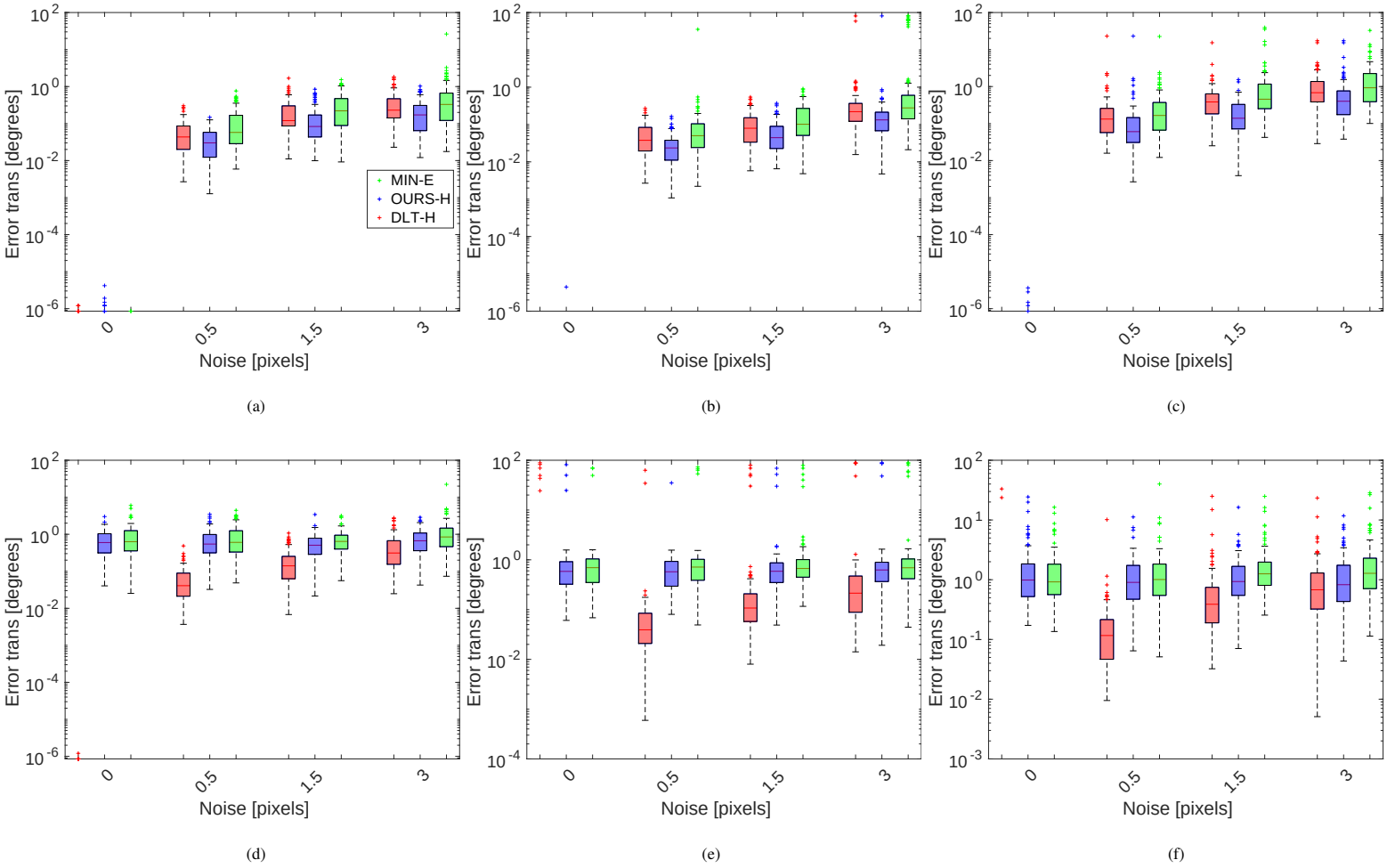

Fig. 8: Error for the translation direction in degrees (log-scale) for each solver with $N = 50$ correspondences. Top row shows the experiments with pure Y-rotation and bottom row where a small perturbation was added.

F. Further experiments on real synthetic data

In this section we include further results of our evaluation on synthetic data.

Translation error

Figure 8 includes the translation error only considering the direction for each solver for $N = 50$ correspondences for the synthetic experiments following the format in figure 4, from left to right: general motion (random translation); lateral motion (translation along the X-axis); and forward motion (along the Z-axis). We do not include the translation error for the problem instances with pure rotation as we are measuring direction error.

Quality of initialization of the certifier

We measure the quality of the initialization of the certifier by the sum of the eigenvalues which are zeroed, shown in figure 9.

For zero noise, all the formulations have zero ($\sim 1e-17$) as minimum eigenvalue. Generally the eigenvalue increases with the level of noise for all the camera configurations, although the effect is more apparent with the minimally constrained formulations HR and QR, although HR empirically performs worse than QR. The overconstrained HRQ reduces the eigenvalue by a factor of ten w.r.t. HR and QR. In most problem instances all the formulations attain minimum eigenvalues below $1e-04$, which shows that the initializations are close to a good minimum of the problem (the closer this value is to zero, the better the initialization). In terms of configurations lateral and forward movements hinder the performance of this stage. For noisy Y-rotation, the eigenvalue increases for all the configurations including with zero noise, being in average an order of magnitude larger than with the noiseless Y-rotation. All the configurations have the same effect over the certifiers (around $1e-04$), with HR having the worst outliers.

Ratio of certified solutions

Figure 10 shows the ratio of certified solutions for $N = 15$ (dashed lines) and $N = 50$ (solid lines) for all the formulations (see legend) under different level of noise (X-axis). We follow the format in figure 9. We consider the certification positive if this cost

Certifier: sum zeroed eigenvalues

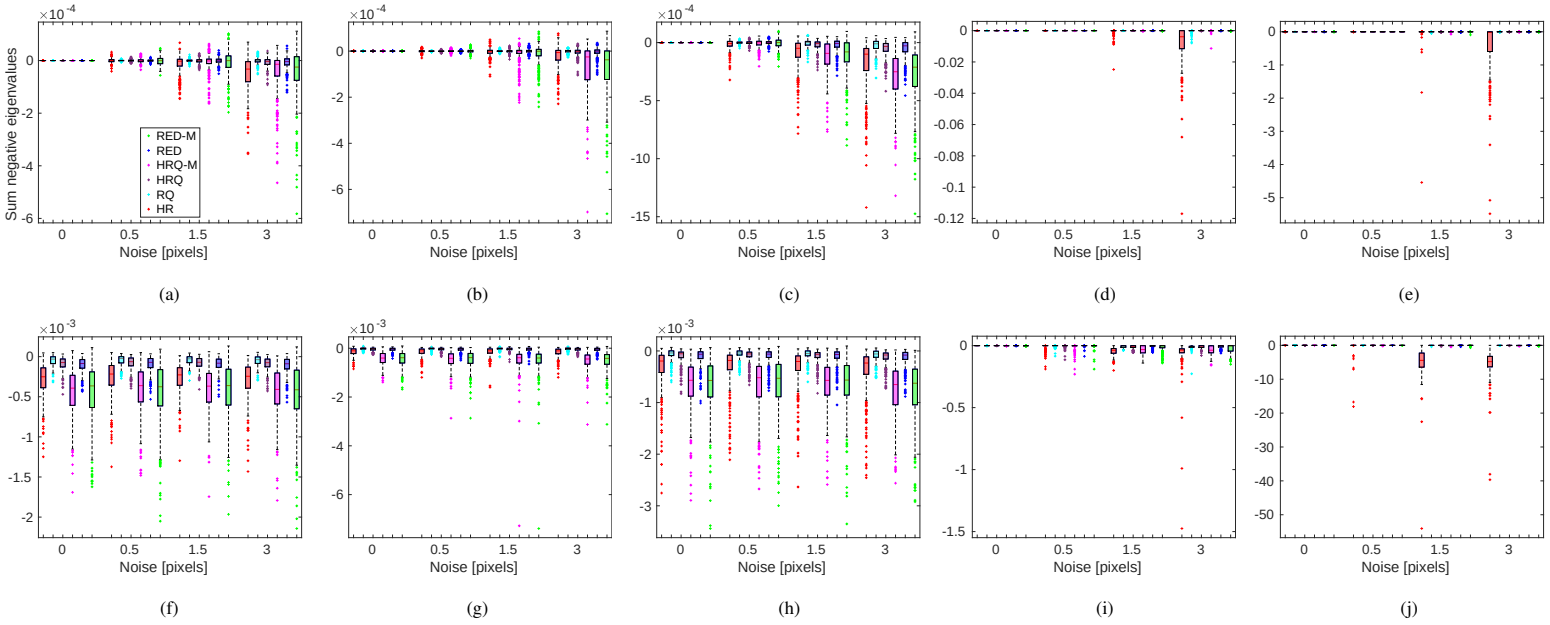

Fig. 9: **Certifier: sum zeroed eigenvalues** Sum of the zeroed eigenvalues of the solution from the certifier initialization. Legend shows the different formulations and the X-axis the applied level of noise. Top row shows the experiments with pure Y-rotation and bottom row the experiments where a small perturbation was added to the rotation. From left to right: general motion (random translation); lateral motion (translation along the X-axis); forward motion (along the Z-axis); zero translation with points on plane; and zero translation with general points. The number of correspondences is fixed to $N = 50$. Notice the difference in the Y-scales.

is below $1e-09$. We draw the next set of conclusions. First, for problems with noiseless Y-rotation, including those with zero translation, all the formulations perform well even for large noise 3 `pix` , except HR. For noisy Y-rotation, the certifiers break earlier. Redundant formulations are able to certify still most solutions when the translation is not zero, and the performance doesn't depend on the level of noise. Noisy Y-rotation and zero translation hinder the performance of the certifier, although the redundant formulations certify more than 70% of the solutions.

Certifier: ratio certifier solutions

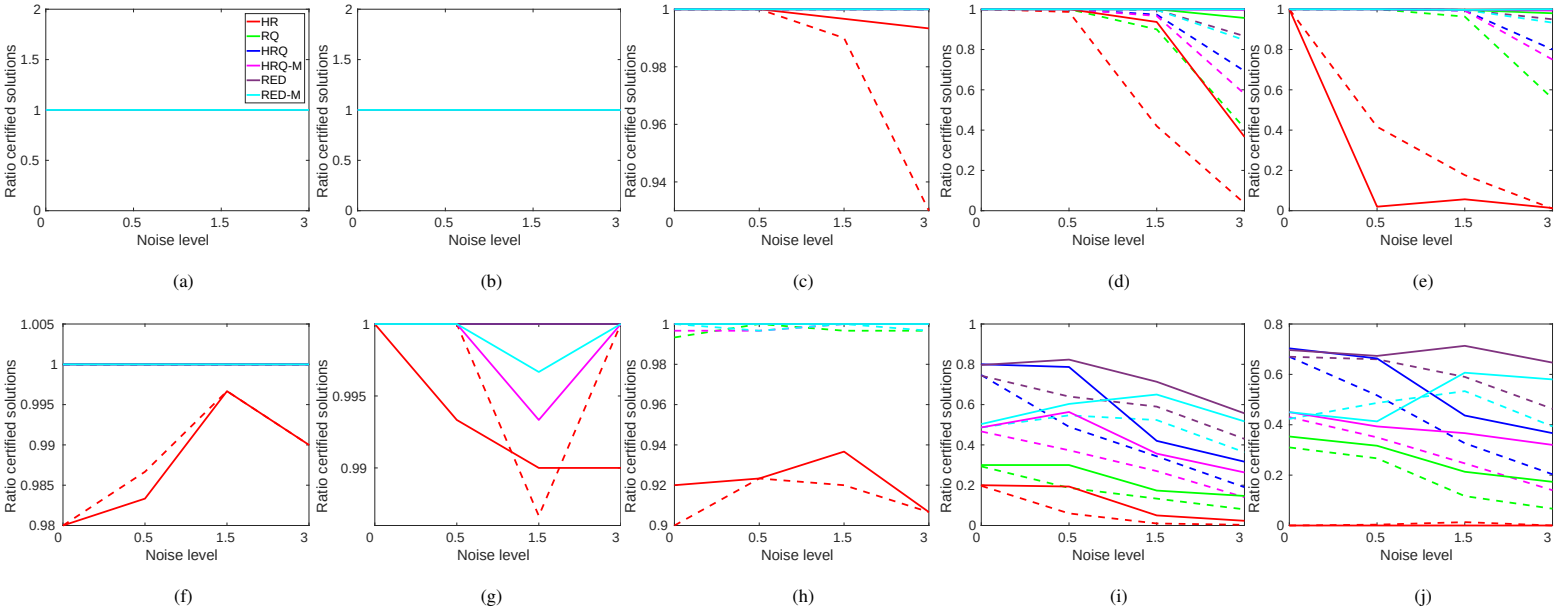

Fig. 10: **Certifier: ratio certifier solutions** Ratio of certified solutions for the different formulations (see legend) and level of noise (X-axis). Top row shows the experiments with pure Y-rotation and bottom row the experiments where a small perturbation was added to the rotation. From left to right: general motion (random translation); lateral motion (translation along the X-axis); forward motion (along the Z-axis); zero translation with points on plane; and zero translation with general points. We show the ratio for $N = 15$ as dashed lines and for $N = 50$ for solid lines.