2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

# Non-Minimal Solvers for Relative Pose Estimation with a Known Relative Rotation Angle

Deshun Hu[1,*]

***Abstract*— Knowing the relative rotation angle improves relative pose estimation accuracy. We consider the problem of computing relative motion from a non-minimal number of correspondences with a known relative rotation angle. While several solvers for minimum correspondences have been proposed, no non-minimal solver for this problem currently exists. In this work, we propose two non-minimal solvers for this problem. The first solver solves the problem using convex relaxation and semidefinite programming, yielding certifiable solutions. The second method approaches the problem through local eigenvalue optimization with random initialization. Increasing the number of initial guesses lowers the chances of missing the correct solution. We conduct experiments on synthetic and real data, confirming our methods' advantages over competing methods.**

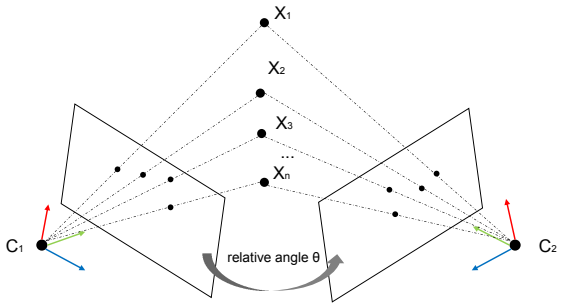



Fig. 1. Two cameras $\mathbf{C}_1$ and $\mathbf{C}_2$ with known relative angle $\theta$ observing n points $\mathbf{X}_1$, $\mathbf{X}_2$, ..., $\mathbf{X}_n$. The objective is to determine the relative pose of the cameras.

## I. INTRODUCTION

Estimating the relative camera pose between each frame pair is a key step in visual odometry (VO), structure from motion (SFM) and simultaneous localization and mapping (SLAM). Relative pose for calibrated cameras is characterized by the epipolar geometry, and commonly the observed point movements are used to estimate the essential matrix between two frames [1]. Due to scale ambiguity of translation, the essential matrix has only five degrees of freedom [2], [3]. The so-called minimal solvers use least point correspondences possible to efficiently calculate the relative pose and are typically used in a hypothesis and test framework (RANSAC) [4] to gain robustness against outliers.

Using measurements from other motion sensors can help to simplify the pose estimation problem [9]. Sensors such as an odometer or an inertial measurement unit (IMU) can provide motion priors such as known gravity direction or known rotation angle. If the sensors are properly calibrated, the IMU data can be used to calculate the vertical direction in the camera coordinate frame, decreasing the degrees of freedom from five to three [5]–[8], [10]. Using rotation angles from motion sensors, on the other hand, eliminates the need for extrinsic calibration between different sensors because the relative rotation angles of rigidly connected sensors are the same. In this case, minimal solvers can take four point correspondences from an image pair and the rotation angle from any relative rotation sensor to identify the relative pose [11]–[13].

The minimum solution, which uses the least points for each hypothesis instantiation, may not necessarily lead to an accurate estimate and may trap further refining procedures in poor minima. Hypotheses generated from larger-than-minimal subsets of the consensus set also benefit robust estimating systems. Non-minimal solvers frequently employ global optimization approaches that require specific relative pose parameterizations for efficiency and performance [14]–[18]. Non-minimal solvers are usually much more computationally complex than minimal solvers.

In this work, we present two non-minimal solvers for the relative pose problem with a known rotation angle. In the first solver, the underlying algebraic error minimization formulation is modeled as a non-convex quadratic constraint quadratic program (QCQP), which is then solved using certifiable convex relaxation [19]. The second solver utilizes a different epipolar constraint to optimize the least eigenvalue of the cost matrix directly over the relative rotation, regardless of the relative translation [20], [21]. A random variation of the starting point prevents the optimization from reaching local minima.

The main technical contributions of our work are:

1) We introduce two QCQP formulations of the problem and solve them using semidefinite programming (SDP);
2) We present a fast method for the problem based on local eigenvalue optimization with random initialization;
3) We validate the effectiveness of the proposed non-minimal solvers on both synthetic and real datasets.

## II. RELATED WORK

During the last two decades, minimal solvers for relative pose estimation have received a lot of attention and have been extensively discussed. Nister's five-point method utilizes Gauss-Jordan elimination as well as the hidden variable

[1]Deshun Hu is with Department of Communication Engineering, Harbin Institute of Technology, Harbin, China hudeshun1993@gmail.com
*Corresponding author, E-mail: hudeshun1993@gmail.com

technique to efficiently reduce the problem to one variable root extraction [2]. Stewenius suggests using Groebner bases to solve the relative pose problem, which have been shown to be more numerically accurate [3]. The authors in [22] introduce the polynomial eigenvalue formulation of relative pose, which is solved using standard numerical algorithms to achieve a stable estimate. Kneip proposes the idea of estimating rotation and translation independently in [20], allowing for accurate rotation estimation even with diminishing translation magnitudes. A solver for small rotations is proposed in [23].

Recently, non-minimal relative pose solvers have gained popularity, showing more accurate results than five point methods. The most widely used and well-known non-minimal relative pose solver is the normalized direct linear transform (DLT) [1], though the solutions are not on the essential manifold. Kneip's method [21] uses an eigenvalue formulation [20] to directly optimize over the relative rotation, which can be performed in a constant amount of time regardless of the number of correspondences. Accounting for point uncertainties in the optimization improves estimation accuracy [24]. Another approach is to reformulate the original problem as a QCQP and then relax it to SDP using Shor's relaxation [19]. If the convex relaxation is tight, the original non-convex problem is solved with global optimality in polynomial time. Using the eigenvalue-based formulation, the authors in [15] formulate the non-minimal relative pose problem as a QCQP whose convex relaxation is always tight empirically. The authors in [14] present an alternative QCQP formulation with a significantly reduced number of variables and constraints by using an equivalent parameterization of the essential matrix. Including redundant constraints in the QCQP helps to tighten the relaxation [25]. Iterative local optimization paired with a quick optimality certifier is another viable approach [26]. However, for that approach to converge quickly, a good initial value is generally required.

Many solvers utilize motion priors, such as known gravity direction and known rotation angle. When the direction of gravity is known, rotation has only one degree of freedom, and the relative pose can be estimated using only three point correspondences [10]. A non-minimal solver based on closed-form eigenvalue minimization for known gravity cases is presented in [18]. Despite having one fewer degree of freedom, solvers for known rotation angles are more complex than 5-point approaches. In [11], the authors propose two solutions to the problem, the first of which has a large elimination template and thus is impractical. The second method optimizes a set of random initial estimations to find the right relative pose [11]. The authors in [12] use quaternion parameterization and Schur elimination in the Groebner bases to create a significantly faster minimum solver. The first minimal generalized relative pose solver with a known rotation angle is also proposed in [12].

Modern motion sensors can provide very accurate relative angles with simple processing. See Fig. 2 for an example. To the best of our knowledge, this is the first work to present non-minimal relative pose solvers with a known rotation angle. In our first solver, the relative angle constraint is expressed as a quadratic constraint on the essential matrix. The second method separates rotation from translation and optimizes directly over the rotation axis to find the best rotation.

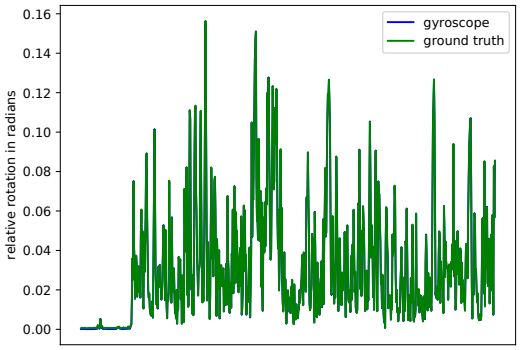



Fig. 2. Relative rotation angles between consecutive images in downsampled EuRoC sequence MH01 [27]. We downsample the sequence by taking every fourth image. The $x$-axis represents images sorted by time. The relative angles calculated from the gyroscope are numerically close to those obtained from the ground truth. Section V contains details on computing relative angles from the gyroscope.

## III. PRELIMINARIES

This section begins by explaining the notations used in this work. The epipolar constraint and the essential matrix are then briefly discussed, as they are required to proceed with the development of our SDP solvers. Finally, we investigate how the problem can be treated as an eigenvalue optimization problem, as the second method does.

### A. Notation

We denote matrices with upper case boldface letters and vectors with lower case boldface letters (e.g. $\mathbf{A}$, $\mathbf{x}$). The identity matrix is written as $\mathbf{I}$, whose dimension can be inferred from the context. We write $\mathbf{A} \succ \mathbf{0}$ (resp. $\mathbf{A} \succeq \mathbf{0}$) to denote that the symmetric matrix $\mathbf{A}$ is positive definite (resp. positive semidefinite). We use $\mathcal{S}^n$ to denote the space of symmetric matrix with dimension $n \times n$. By MATLAB syntax, $\mathbf{A}_{[a:b,c:d]}$ stands for the submatrix of $\mathbf{A}$ constructed by rows $a$ to $b$ and columns $c$ to $d$, and $\mathbf{x}_{[a:b]}$ stands for the entries of vector $\mathbf{x}$ indexed from $a$ to $b$. $\mathrm{A}_{ij}$ represents the element of $\mathbf{A}$ in the $i$th row and $j$th column, while $\mathrm{x}_i$ represents the $i$th element of $\mathbf{x}$. $\mathrm{vec}(\mathbf{A})$ stacks matrix entries by column-first order. $\mathrm{trace}(\mathbf{A})$ denotes the trace of matrix $\mathbf{A}$. We denote by $\lfloor \mathbf{t} \rfloor_\times$ the matrix form for the cross-product with a 3D vector $\mathbf{t}$. The Kronecker product between two matrices is denoted as $\mathbf{A} \otimes \mathbf{B}$. The identity $\mathrm{vec}(\mathbf{AXB}) = (\mathbf{B}^\top \otimes \mathbf{A})\mathrm{vec}(\mathbf{X})$ is used in later formulations.

### B. Epipolar geometry and essential matrix

We represent the relative pose between two given calibrated cameras as $3 \times 4$ matrix of the form

$$\mathbf{P} = \left[\mathbf{R}, \mathbf{t}\right], \tag{1}$$

where $\mathbf{R}$ is the rotation matrix and $\mathbf{t}$ is the translation vector. The essential matrix $\mathbf{E}$ is defined as

$$\mathbf{E} = \lfloor \mathbf{t} \rfloor_{\times} \mathbf{R}. \tag{2}$$

Representing pair-wise correspondences with pairs of (noisy) unit bearing vectors $(\mathbf{f}_i, \mathbf{f}_i^{'})$ which point from the corresponding camera center towards the same world point, the epipolar constraint reads

$$\mathbf{f}_i^{'\top} \mathbf{E} \mathbf{f}_i = 0. \tag{3}$$

Using convex relaxation to estimate relative pose requires a quadratic parameterization of $\mathbf{E}$. Directly parameterizing $\mathbf{E}$ in $\mathbf{R}$ and $\mathbf{t}$ via (2) is possible [15], but inefficient because the rotation matrix imposes a large number of constraints. To simplify the resulting SDP problem, we define $\mathbf{E}$ utilizing equivalent conditions proposed by [14]

$$\mathbf{E}\mathbf{E}^{\top} = \mathbf{t}\mathbf{t}^{\top}, \ \mathbf{t}^{\top}\mathbf{t} = 1. \tag{4}$$

As the scale is not observable, we require normalized translation in (4). This parameterization considerably reduces the number of variables and constraints in the resulting SDP problem [14].

### C. Problem formulation

Under the presence of noise, we pursue the optimal pose by minimizing algebraic error of the epipolar constraint

$$\min_{\mathbf{E}} \sum_{i=1}^{N} (\mathbf{f}_i^{'\top} \mathbf{E} \mathbf{f}_i)^2, \tag{5}$$

where $N$ is the number of point correspondences. The cost function (5) is quadratic in $\mathbf{E}$, and the problem can be turned into a QCQP with a quadratic parameterization of $\mathbf{E}$ (4). In our case, additional quadratic constraints must be imposed on $\mathbf{E}$ to specify the known rotation angle. The resulting QCQP is relaxed and then solved by off-the-shelf SDP solvers.

Another approach is to eliminate translation from (5) and obtain a rotation-only expression [21], which is then optimized directly on the rotation manifold. Due to the shift invariance of scalar triplet product, the epipolar constraint can be rewritten as

$$\mathbf{f}_i^{'\top} \mathbf{E} \mathbf{f}_i = \mathbf{f}_i^{'\top} \lfloor \mathbf{t} \rfloor_{\times} \mathbf{R} \mathbf{f}_i = \mathbf{t}^{\top} \lfloor \mathbf{R}\mathbf{f}_i \rfloor_{\times} \mathbf{f}_i^{'}. \tag{6}$$

And (5) can be written as

$$\begin{aligned} \min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{N} (\mathbf{f}_i^{'\top} \mathbf{E} \mathbf{f}_i)^2 &= \sum_{i=1}^{N} \mathbf{t}^{\top} (\lfloor \mathbf{R}\mathbf{f}_i \rfloor_{\times} \mathbf{f}_i^{'}) (\lfloor \mathbf{R}\mathbf{f}_i \rfloor_{\times} \mathbf{f}_i^{'})^{\top} \mathbf{t}, \\ &= \mathbf{t}^{\top} (\sum_{i=1}^{N} (\lfloor \mathbf{R}\mathbf{f}_i \rfloor_{\times} \mathbf{f}_i^{'}) (\lfloor \mathbf{R}\mathbf{f}_i \rfloor_{\times} \mathbf{f}_i^{'})^{\top}) \mathbf{t}. \end{aligned} \tag{7}$$

Denoting $\mathbf{M}(\mathbf{R}) = \sum_{i=1}^{N} (\lfloor \mathbf{R}\mathbf{f}_i \rfloor_{\times} \mathbf{f}_i^{'}) (\lfloor \mathbf{R}\mathbf{f}_i \rfloor_{\times} \mathbf{f}_i^{'})^{\top}$, we can now rewrite (7) as

$$\min_{\mathbf{R},\mathbf{t}} \mathbf{t}^{\top} \mathbf{M}(\mathbf{R}) \mathbf{t}. \tag{8}$$

Because the scale is not observable, we only optimize over the normalized translation. For one fixed rotation, the translation can be set as the eigenvector corresponding to the least eigenvalue of $\mathbf{M}(\mathbf{R})$. As a result, the original problem (5) is reduced to minimizing the smallest eigenvalue of $\mathbf{M}(\mathbf{R})$, which only depends on the rotation matrix

$$\min_{\mathbf{R}} \lambda_{min}(\mathbf{M}(\mathbf{R})). \tag{9}$$

Since the rotation angle is already determined, all that remains is to parameterize and optimize the rotation matrix over the rotation axis.

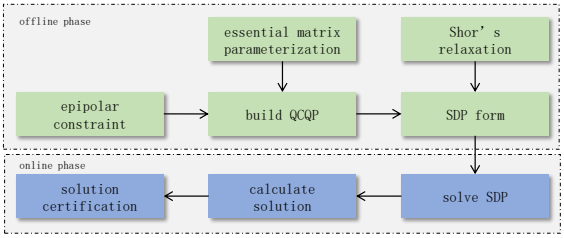



Fig. 3. The process of our SDP algorithm. During the offline phase, we write the problem as a QCQP and apply Shor's relaxation. Online, we substitute point correspondences and perform SDP optimization, after which the solution is checked for certification.

## IV. SOLUTION PROCEDURE

In this section, we solve the non-minimal relative pose problem with a known rotation angle. The first method approaches the problem as a QCQP, which is subsequently relaxed into an SDP and solved using off-the-shell solvers. The second method employs local eigenvalue optimization with random starts to find a global solution.

### A. SDP solvers

In this subsection, we present two SDP solvers for the problem. Fig. 3 depicts the solvers' steps.

Using the identity $\mathrm{vec}(\mathbf{AXB}) = (\mathbf{B}^{\top} \otimes \mathbf{A})\mathrm{vec}(\mathbf{X})$, the objective in (5) can be reformulated as a quadratic form

$$\sum_{i=1}^{N} (\mathbf{f}_i^{'\top} \mathbf{E} \mathbf{f}_i)^2 = \mathbf{e}^{\top} \mathbf{C} \mathbf{e}, \tag{10}$$

where $\mathbf{e} = \mathrm{vec}(\mathbf{E})$ and

$$\mathbf{C} = \sum_{i=1}^{N} (\mathbf{f}_i \otimes \mathbf{f}_i^{'}) (\mathbf{f}_i \otimes \mathbf{f}_i^{'})^{\top}. \tag{11}$$

Using (4), the equivalent QCQP form of minimizing the algebraic error is

$$\begin{aligned} &\min_{\mathbf{E},\mathbf{t}} \ \mathbf{e}^{\top} \mathbf{C} \mathbf{e} \\ &\text{s.t. } \mathbf{E}\mathbf{E}^{\top} = \mathbf{t}\mathbf{t}^{\top}, \ \mathbf{t}^{\top}\mathbf{t} = 1. \end{aligned} \tag{12}$$

We introduce a new optimization variable

$$\mathbf{x} = \begin{bmatrix} \mathrm{vec}(\mathbf{E})^{\top} & \mathbf{t}^{\top} \end{bmatrix}^{\top}, \tag{13}$$

and variables $\mathbf{x}$ and $\mathrm{vec}(\mathbf{E})$ are related by

$$\mathrm{vec}(\mathbf{E}) = \mathbf{T}\mathbf{x}, \tag{14}$$

where $\mathbf{T} \in \mathrm{R}^{9\times 12}$ denotes a constant linear transformation with an identity matrix in the left 9 columns and zero else. Combining (10) and (14), we can reformulate the objective as

$$\min_{\mathbf{x}} \mathbf{x}^\top \mathbf{A}_0 \mathbf{x}, \tag{15}$$

where $\mathbf{A_0} = \mathbf{T}^\top \mathbf{C} \mathbf{T}$. Considering the constraints in (12), the QCQP form can be written as

$$\begin{aligned} &\min_{\mathbf{x}} \ \mathbf{x}^\top \mathbf{A}_0 \mathbf{x} \\ &\text{s.t. } \mathbf{x}^\top \mathbf{A}_i \mathbf{x} = b_i, \quad i = 1, \cdots, 7, \end{aligned} \tag{16}$$

where $\mathbf{A}_i$ and $\mathbf{b}_i$, $i = 1, \cdots, 7$, are from 7 equalities introduced in (4). See [14] for more details. There are 12 variables and 7 constraints in this problem.

Let $\theta$ be the relative rotation angle between two calibrated camera frames. $\theta$ can be calculated from other odometry sensors. The trace of the rotation matrix can be calculated as $\tau = \mathrm{trace}(\mathbf{R}) = 2 * \cos\theta + 1$. This introduces another quadratic constraint on the essential matrix

$$\frac{1}{2}(\tau^2 - 1)\mathrm{trace}(\mathbf{E}\mathbf{E}^\top) + (\tau + 1)\mathrm{trace}(\mathbf{E}\mathbf{E}) + \mathrm{trace}^2(\mathbf{E}) = 0. \tag{17}$$

See [28] for the proof. If one essential matrix satisfies (17), then one of its factorized rotation matrices will have the desired rotation angle [1], [28]. The QCQP with the known rotation angle constraint is

$$\begin{aligned} &\min_{\mathbf{x}} \ \mathbf{x}^\top \mathbf{A}_0 \mathbf{x} \\ &\text{s.t. } \mathbf{x}^\top \mathbf{A}_i \mathbf{x} = b_i, \quad i = 1, \cdots, 8, \end{aligned} \tag{18}$$

where $\mathbf{A}_8$ and $\mathbf{b}_8$ are from the rotation angle quadratic constraint introduced above.

Introducing a new matrix variable $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$, with the identity $\mathrm{trace}(\mathbf{A}_i\mathbf{X}) = \mathbf{x}^\top \mathbf{A}_i \mathbf{x}$, the problem (18) is identical to

$$\begin{aligned} \min_{\mathbf{X}\in\mathcal{S}^{12}} \quad & \mathrm{trace}(\mathbf{A}_0\mathbf{X}) \\ \text{s.t.} \quad & \mathrm{trace}(\mathbf{A}_i\mathbf{X}) = b_i, \quad i = 1, \cdots, 8 \\ & \mathbf{X} \succeq \mathbf{0}, \\ & \mathrm{rank}(\mathbf{X}) = 1. \end{aligned} \tag{19}$$

(19) is a non-convex problem owning to the non-trivial rank constraint. We derive Shor's relaxation of (19) by dropping the rank constraint as in [14], [15]

$$\begin{aligned} \min_{\mathbf{X}\in\mathcal{S}^{12}} \quad & \mathrm{trace}(\mathbf{A}_0\mathbf{X}) \\ \text{s.t.} \quad & \mathrm{trace}(\mathbf{A}_i\mathbf{X}) = b_i, \quad i = 1, \cdots, 8 \\ & \mathbf{X} \succeq \mathbf{0}. \end{aligned} \tag{20}$$

The problem (20) is an SDP problem, which can be easily solved in polynomial time by off-the-shell solvers [29]. It turns out that the dual problems of (19) and (20) are the same following the derivation of Lagrange duality [19]. The dual problem is

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \quad & -\boldsymbol{\lambda}^\top \mathbf{b} \\ \text{s.t.} \quad & \mathbf{Q}(\boldsymbol{\lambda}) = \mathbf{A}_0 + \sum_{i=1}^{8} \lambda_i \mathbf{A}_i \succeq \mathbf{0}, \end{aligned} \tag{21}$$

where $\mathbf{b} = \begin{bmatrix} b_1, \cdots, b_8 \end{bmatrix}^\top$ and $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1, \cdots, \lambda_8 \end{bmatrix}^\top$. Because the block sparsity structure of the problem remains unchanged, the procedure for obtaining the solution and checking for tightness is identical to [14]. The whole process of the first SDP solver is illustrated in Alg. 1.

**Algorithm 1:** Non-Minimal Relative Pose Estimation with a Known Rotation Angle via SDP

**Input:** normalized point correspondences $(\mathbf{f}_i, \mathbf{f}_i^{'})$, $i = 1, \cdots, N$, relative rotation angle $\theta$
**Output:** rotation $\mathbf{R}^*$, translation $\mathbf{t}^*$

```
Construct C by equation (11);
Construct A_0 in (15);
Calculate τ and construct A_8 in (18);
Solve SDP (20) and obtain optimal solution X*;
Calculate the numerical rank of X*_[1:9,1:9] and
  X*_[10:12,10:12];
if the numerical rank is 1 then
    The SDP relaxation is tight;
else
    The SDP relaxation is not tight;
end
Set vec(E) as the eigenvector corresponding to the
  biggest eigevalue of X*_[1:9,1:9];
Decompose E to obtain R* and t*.
```

Now we briefly present the second SDP solver. In the second SDP solver, similar to [30], [31], we obtain the quadratic constraint for the known rotation angle from the elimination theory [32]. Adding the equality $\tau = \mathrm{trace}(\mathbf{R})$ to the constraints of the essential matrix and rotation matrix gives various equations in $\mathbf{E}$, $\mathbf{t}$ and $\mathbf{R}$. By eliminating $\mathbf{R}$ from these equations, we can extract the following quadratic constraint containing $\tau$

$$\begin{aligned} &2\tau(\mathrm{E}_{23} - \mathrm{E}_{32})\mathrm{t}_1 + 2\tau(\mathrm{E}_{31} - \mathrm{E}_{13})\mathrm{t}_2 + \\ &2\tau(\mathrm{E}_{12} - \mathrm{E}_{21})\mathrm{t}_3 + \mathrm{E}_{12}^2 + \mathrm{E}_{13}^2 - \\ &2\mathrm{E}_{12}\mathrm{E}_{21} - \mathrm{E}_{22}^2 - 2\mathrm{E}_{13}\mathrm{E}_{31} - \\ &2\mathrm{E}_{23}\mathrm{E}_{32} - \mathrm{E}_{33}^2 + \mathrm{t}_1^2 + \tau^2 = 0. \end{aligned} \tag{22}$$

We use the code in Listing 1 to obtain (22).

Cross terms between $\mathbf{E}$ and $\mathbf{t}$ in (22), like $\mathrm{E}_{32}\mathrm{t}_1$, break the original block sparsity pattern [14]. And we can add redundant quadratic constraints $\mathbf{t}^\top \mathbf{E} = \mathbf{0}$ on the cross terms. More redundant constraints [25] can be added, but we will not discuss this in the work. We now have an SDP problem with 11 constraints after using the convex relaxation. As we no longer have the block sparse structure as in [14], we need to decompose the entire matrix $\mathbf{X}$ and check the rank for tightness [33], [34]. The condition $\tau = \mathrm{trace}(\mathbf{R})$ is also checked because (22) is only a necessary condition according to the elimination theory [32]. The entire procedure is similar to Alg. 1 and is omitted for space.

```
R = QQ[r11,r12,r13,r21,r22,r23,r31,r32,r33,t1,t2,t3
    ,e11,e12,e13,e21,e22,e23,e31,e32,e33,tau];

Rot = matrix{{r11,r12,r13},{r21,r22,r23},{r31,r32,
    r33}};
t = matrix{{t1},{t2},{t3}};
tc = matrix{{0_R,-t3,t2},{t3,0_R,-t1},{-t2,t1,0_R
    }};
Ess = matrix{{e11,e12,e13},{e21,e22,e23},{e31,e32,
    e33}};
E = matrix{{1_R,0_R, 0_R},{0_R,1_R,0_R},{0_R,0_R,1
    _R}};

I = ideal(minors(1,Ess-tc*Rot) + minors(1,Rot*
    transpose(Rot)-E) + minors(1,transpose(Rot)*Rot
    -E), transpose(t)*t-1_R, trace(Rot)-tau);
Il = eliminate({r11,r12,r13,r21,r22,r23,r31,r32,r33
    },I);
mingens Il
```

Listing 1. Macaulay2 code

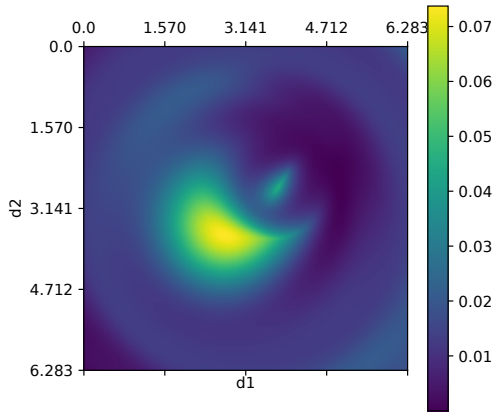



Fig. 4. A sample cost surface of the function in (25). In this figure, $\epsilon$ is set to 0 to show only the minimal eigenvalue. There are only a few local minima in the cost function (25).

### B. Optimization solver

We provide another non-minimal solver based on local eigenvalue optimization in this section. The Rodrigues formula for rotation matrix is

$$\mathbf{R} = \mathbf{R}(\theta, \mathbf{r}) = \cos\theta \mathbf{I} + (1 - \cos\theta)\mathbf{r}\mathbf{r}^\top + \sin\theta \lfloor \mathbf{r} \rfloor_\times, \quad (23)$$

where $\theta$ is the rotation angle and $\mathbf{r}$ is the unit rotation axis. To optimize over $\mathbf{r}$, we parameterize the unit rotation axis as

$$\mathbf{r}(\mathbf{d}) = \mathbf{r}(d_1, d_2) = [\sin(\text{norm}(\mathbf{d})) * \cos(\arctan(d_2/d_1)), \sin(\text{norm}(\mathbf{d})) * \sin(\arctan(d_2/d_1), \cos(\text{norm}(\mathbf{d}))]^\top, \quad (24)$$

where $\mathbf{d} = [d_1, d_2]^\top$ is the logarithm parameter for the unit axis [35].

We use the Levenberg-Marquardt (LM) algorithm to efficiently optimize (9). To apply the algorithm, the number of cost terms must be at least equal to the number of unknowns. Following Kneip's trick [21], we add two more terms to the cost function in (9)

$$\min_{\mathbf{d}} \lambda_{min}(\mathbf{M}(\mathbf{R}(\mathbf{d}))) + \epsilon * ((\frac{\partial \lambda_{min}(\mathbf{M}(\mathbf{R}(\mathbf{d})))}{\partial d_1})^2 + (\frac{\partial \lambda_{min}(\mathbf{M}(\mathbf{R}(\mathbf{d})))}{\partial d_2})^2), \quad (25)$$

where $\epsilon$ is the weight factor for the derivative terms. Because the derivatives are zero at the minimum, the two extra terms have no influence on the global (local) optimum, while allowing us to solve it effectively using the LM approach. Since the initial value could be very poor, a large number of iterations are needed in the LM method.

To prevent local minima, random starts are required. The initial rotation axis is created at random in 3D space rather than logarithmic space, and it is then normalized to represent a uniform random distribution. From our experiments, 7 random starts are sufficient to yield a satisfactory result. Fig. 4 depicts a sample cost surface of (25) from our simulation data in section V. The MATLAB symbolic computing toolbox is used to simplify (25). The weight factor $\epsilon$ is set to $0.2$ in all our experiments in section V.

## V. EXPERIMENTS

In this section, we conduct experiments on synthetic and real data to evaluate the performance of the proposed solvers. To compare the relative rotation accuracy, we use the geodesic distance criteria for quantitative evaluation. For rotation, geodesic distance between the estimated rotation $\mathbf{R}_{est}$ and ground truth $\mathbf{R}_{gt}$ is defined as

$$\epsilon_{rot}[degree] = \frac{180}{\pi} \arccos(\frac{\text{trace}(\mathbf{R}_{gt}^\top \mathbf{R}_{est}) - 1}{2}). \quad (26)$$

The rotation errors are measured in degrees. We skip the translation error analysis for space. Because the SDP solvers have similar relaxation tightness for a sufficient number of points, we compare the solutions from the relevant decomposition directly and skip the tightness analysis. We mainly focus on the median errors because all solvers are used in a RANSAC [4] framework in practice.

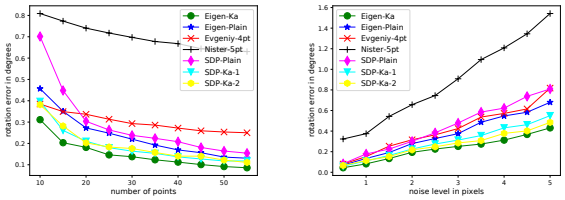
The first and second proposed SDP solvers are referred to as SDP-Ka-1 and SDP-Ka-2, respectively. The proposed eigenvalue solver is denoted as Eigen-Ka. We compare our methods against the following existing methods: Nister-5pt is the essential matrix solver in [2], Evgeniy-4pt is the solver in [12], Eigen-Plain is the local eigenvalue solver in [21], and SDP-Plain is the SDP method proposed by [14]. For a fair comparison, we run Nister-5pt and Evgeniy-4pt for 10 RANSAC times each and keep the solution with the least algebraic error. Eigen-Plain uses the ground truth value as the starting point, whereas Eigen-Ka applies local optimization with 7 random starts. For Nister-5pt and Eigen-Plain, we use the implementations in [36]. All the SDP solvers are implemented in MATLAB. Our eigenvalue solver is written in C++.

For synthetic data, we generate random problems in a similar manner to [15]. The random translation is constrained within a spherical shell with radius $[1, 3]$. We generate a

TABLE I

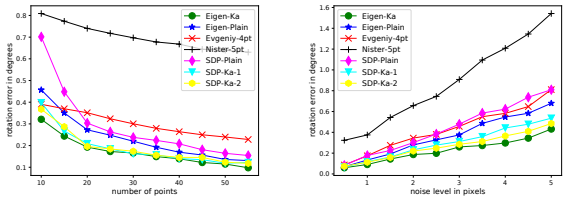
ACCURACY OF RELATIVE POSE SOLVERS ON THE EUROC DATASET. ROTATION ERRORS ARE MEASURED IN DEGREES. MH01 DENOTES THE SEQUENCE MACHINE HALL 01, AND SIMILARLY FOR OTHERS.

| Method | MH01 | MH02 | MH03 | MH04 | MH05 | V101 | V102 | V103 | V201 | V202 | V203 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SDP-Plain | 0.4821 | 0.4465 | 0.7671 | 0.6485 | 0.5499 | 0.8462 | 1.4486 | 1.3364 | 0.6832 | 1.3357 | 1.2794 |
| SDP-Ka-1 | 0.2854 | 0.2626 | 0.5132 | 0.3683 | 0.3486 | 0.6270 | 1.1270 | 0.8747 | 0.4490 | 0.9341 | 1.0159 |
| SDP-Ka-2 | 0.2873 | 0.2902 | 0.5105 | 0.3738 | 0.3435 | 0.6029 | 1.0806 | 0.8405 | 0.4509 | 0.9706 | 1.0327 |
| Eigen-Plain | 0.3667 | 0.3607 | 0.6912 | 0.4901 | 0.4601 | 0.6894 | 1.290 | 1.124 | 0.5517 | 1.2117 | 1.1091 |
| Eigen-Ka | 0.2678 | 0.2521 | 0.4934 | 0.3481 | 0.3236 | 0.5566 | 1.061 | 0.7661 | 0.3918 | 0.8702 | 0.8544 |
| Nister-5pt | 0.5199 | 0.4865 | 0.9367 | 0.6938 | 0.6822 | 1.064 | 1.924 | 1.5309 | 0.7604 | 1.5082 | 1.5175 |
| Evgeniy-4pt | 0.3312 | 0.3153 | 0.5496 | 0.4237 | 0.6227 | 0.6022 | 1.162 | 0.9371 | 0.5118 | 1.0949 | 1.0747 |




(a) (b)

Fig. 5. Relative rotation errors for various methods. Figure (a) shows the errors with different numbers of points. Figure (b) depicts the errors with varying levels of noise. Relative rotation angles are assumed to be perfectly known.




(a) (b)

Fig. 6. Relative rotation errors for various methods with noisy relative angles. Figure (a) shows the errors with different numbers of points. Figure (b) depicts the errors with varying levels of noise.

set of 3D world points randomly around the origin with distances varying between 2 and 20, retaining points that lie fully inside the Field of View (FOV) of both cameras. Both cameras are assumed to have the same focal length 800, image width 1600, and image height 900. The default number of points is 25, and the default noise level is 2 pixel. By running all solvers 200 times, the median errors are calculated, and the results are shown in Fig. 5 and 6. In Fig. 5, the relative rotation angle is assumed to be perfectly known. In Fig. 6, we perturb the relative angle $\theta$ with uniform noise in the range $[-0.02\theta, 0.02\theta]$. We observe only a minor degradation in performance for methods that use noisy relative angles in Fig. 6. Not surprisingly, non-minimal solvers outperform their competing minimal solvers by a large margin, and knowing relative angles significantly improves the accuracy. The eigenvalue solver Eigen-Plain beats the competing SDP solver SDP-Plain in our experiments, which differs from the findings in [14]. This is because we initialize the relative rotation with the ground truth value in Eigen-Plain. Furthermore, increasing the number of points benefits non-minimum solvers significantly while improving minimal solvers' performance only mildly. SDP-Ka-1 is less accurate than SDP-Ka-2 in high noise cases, demonstrating the benefit of including redundant constraints.

To test the proposed techniques on real-world data, we choose sequences from the Euroc dataset [27]. To construct the correspondences, we extract FAST [37] features and find feature matches between consecutive images in the left camera from forward and backward optical flow [38]. Point correspondences with correct optical flow are further filtered by the fundamental matrix [1]. Using the intrinsic parameter provided in [27], we obtain the corresponding bearing vectors. To avoid having too small baselines between consecutive images, we reduce the number of images in the sequences by taking only every fourth image. For each image pair, we sample 60 point correspondences for later computation. The relative angle is calculated by integrating gyroscope data. We remove a fixed bias $[-0.00239, 0.0222, 0.0768]^\top$ from gyroscope data in all sequences. This bias is calculated by averaging the gyroscope data over one static period. This simple processing yields very accurate relative angle measurements, as seen in Fig. 2. Table I shows the mean rotation errors of the compared methods. We again see that employing more correspondences improves accuracy, and knowing the relative angle boosts the estimation considerably. Besides, despite the fact that they converge globally, SDP solvers are less accurate than local eigenvalue solvers [16].

## VI. CONCLUSIONS

In this work, we present non-minimal solutions to the relative pose problem with a known rotation angle. In SDP solvers, the angle constraint is expressed as a quadratic constraint on the essential matrix. The eigenvalue solver, however, optimizes directly along the rotation axis to find the optimal solution. We show that gyroscope data can produce highly accurate relative angles. We illustrate how employing non-minimal correspondences improves accuracy. Experiments also demonstrate that solvers who know the relative angle provide more accurate estimates than those who do not. One could argue that, in the same way that gravity data from accelerators is widely used, relative angle data from gyroscopes should not be neglected when determining relative pose.

## REFERENCES


[1] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. New York, NY, USA: Cambridge University Press, 2 ed., 2003.

[2] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[3] H. Stewenius, C. Engels, and D. Nistér, "Recent developments on direct relative orientation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, no. 4, pp. 284–294, 2006.

[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[5] Y. Ding, J. Yang, J. Ponce, and H. Kong, "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)*, pp. 1655–1664, IEEE, 2019.

[6] Y. Ding, J. Yang, J. Ponce, and H. Kong, "Homography-based minimal-case relative pose estimation with known gravity direction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 196–210, 2020.

[7] Y. Ding, J. Yang, J. Ponce, and H. Kong, "Minimal solutions to relative pose estimation from two views sharing a common direction with unknown focal length," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7045–7053, 2020.

[8] Y. Ding, D. Barath, J. Yang, and Z. Kukelova, "Relative pose from a calibrated and an uncalibrated smartphone image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12766–12775, 2022.

[9] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," 2007.

[10] C. Sweeney, J. Flynn, and M. Turk, "Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem," in *2014 2nd International Conference on 3D Vision*, vol. 1, pp. 483–490, IEEE, 2014.

[11] B. Li, L. Heng, G. H. Lee, and M. Pollefeys, "A 4-point algorithm for relative pose estimation of a calibrated camera with a known relative rotation angle," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1595–1601, IEEE, 2013.

[12] E. Martyushev and B. Li, "Efficient relative pose estimation for cameras and generalized cameras in case of known relative rotation angle," *Journal of Mathematical Imaging and Vision*, vol. 62, no. 8, pp. 1076–1086, 2020.

[13] B. Li, E. Martyushev, and G. H. Lee, "Relative pose estimation of calibrated cameras with known se (3) invariants," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX*, pp. 215–231, Springer, 2020.

[14] J. Zhao, "An efficient solution to non-minimal case essential matrix estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[15] J. Briales, L. Kneip, and J. Gonzalez-Jimenez, "A certifiably globally optimal solution to the non-minimal relative pose problem," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 145–154, 2018.

[16] J. Zhao, W. Xu, and L. Kneip, "A certifiably globally optimal solution to generalized essential matrix estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12034–12043, 2020.

[17] S. Agostinho, J. Gomes, and A. Del Bue, "Cvxpnpl: A unified convex solution to the absolute pose estimation problem from point and line correspondences," *arXiv preprint arXiv:1907.10545*, 2019.

[18] Y. Ding, D. Barath, J. Yang, H. Kong, and Z. Kukelova, "Globally optimal relative pose estimation with gravity prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 394–403, 2021.

[19] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[20] L. Kneip, R. Siegwart, and M. Pollefeys, "Finding the exact rotation between two images independently of the translation," in *European conference on computer vision*, pp. 696–709, Springer, 2012.

[21] L. Kneip and S. Lynen, "Direct optimization of frame-to-frame rotation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2352–2359, 2013.

[22] Z. Kukelova, M. Bujnak, and T. Pajdla, "Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems.," in *BMVC*, vol. 2, p. 2008, 2008.

[23] D. Hu, "Approximating the polynomial system for effective relative pose estimation," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 374–380, IEEE, 2022.

[24] D. Muhle, L. Koestler, N. Demmel, F. Bernard, and D. Cremers, "The probabilistic normal epipolar constraint for frame-to-frame rotation optimization under uncertain feature positions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1819–1828, 2022.

[25] M. Garcia-Salguero, J. Briales, and J. Gonzalez-Jimenez, "A tighter relaxation for the relative pose problem between cameras," *Journal of Mathematical Imaging and Vision*, vol. 64, no. 5, pp. 493–505, 2022.

[26] M. Garcia-Salguero, J. Briales, and J. Gonzalez-Jimenez, "Certifiable relative pose estimation," *Image and Vision Computing*, vol. 109, p. 104142, 2021.

[27] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[28] E. Martyushev, "Self-calibration of cameras with euclidean image plane in case of two views and known relative rotation angle," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[29] M. Grant and S. Boyd, "Cvx: Matlab software for disciplined convex programming, version 2.1," 2014.

[30] Y. Ding, Y. Su, C. Xu, J. Yang, and H. Kong, "A general elimination strategy for camera motion estimation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9333–9339, IEEE, 2021.

[31] S. Bhayani, T. Sattler, D. Barath, P. Beliansky, J. Heikkilä, and Z. Kukelova, "Calibrated and partially calibrated semi-generalized homographies," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5936–5945, 2021.

[32] D. Cox, J. Little, and D. OShea, *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.

[33] C. Aholt, S. Agarwal, and R. Thomas, "A qcqp approach to triangulation," in *European Conference on Computer Vision*, pp. 654–667, Springer, 2012.

[34] X. Zheng, X. Sun, D. Li, and Y. Xu, "On zero duality gap in nonconvex quadratic programming problems," *Journal of Global Optimization*, vol. 52, no. 2, pp. 229–242, 2012.

[35] Y. Liu, G. Chen, and A. Knoll, "Globally optimal vertical direction estimation in atlanta world," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[36] L. Kneip and P. Furgale, "Opengv: A unified and generalized approach to real-time calibrated geometric vision," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, IEEE, 2014.

[37] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 105–119, 2008.

[38] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International journal of computer vision*, vol. 56, no. 3, pp. 221–255, 2004.