

# A Consistently Fast and Globally Optimal Solution to the Perspective-n-Point Problem

George Terzakis<sup>1</sup> and Manolis Lourakis<sup>2</sup>

<sup>1</sup> Rovco, The Quorum, Bond Street, Bristol, BS1 3AE, UK  
george.terzakis@rovco.com

<sup>2</sup> Foundation for Research and Technology – Hellas  
N. Plastira 100, GR-70013 Heraklion, Greece  
lourakis@ics.forth.gr

**Abstract.** An approach for estimating the pose of a camera given a set of 3D points and their corresponding 2D image projections is presented. It formulates the problem as a non-linear quadratic program and identifies regions in the parameter space that contain unique minima with guarantees that at least one of them will be the global minimum. Each regional minimum is computed with a sequential quadratic programming scheme. These premises result in an algorithm that always determines the global minima of the perspective-n-point problem for any number of input correspondences, regardless of possible coplanar arrangements of the imaged 3D points. For its implementation, the algorithm merely requires ordinary operations available in any standard off-the-shelf linear algebra library. Comparative evaluation demonstrates that the algorithm achieves state-of-the-art results at a consistently low computational cost.

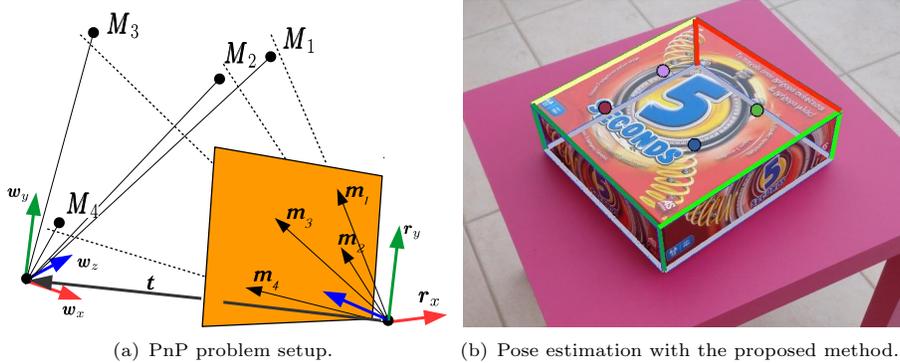
**Keywords:** Perspective-n-point problem, pose estimation, non-linear quadratic program, sequential quadratic programming, global optimality

## 1 Introduction

The perspective-n-point (PnP) problem concerns the recovery of 6D pose given the central projections of  $n \geq 3$  known 3D points on a calibrated camera (cf. Fig. 1). It arises often in vision and robotics applications involving localization, pose tracking and multi-view 3D reconstruction, e.g. [44,39,50,36,33,40,23,42,32].

We begin with the definition of the PnP problem in order to establish notation for the rest of the paper and remind the reader of the most typical cost function formulations associated with the problem. For a set of known Euclidean world points  $\mathbf{M}_i \in \mathbb{R}^3$ ,  $i \in \{1, \dots, n\}$  and their corresponding normalized projections  $\mathbf{m}_i$  on the  $Z = 1$  plane of an unknown camera coordinate frame, we seek to recover the rotation matrix  $\mathbf{R}$  and translation vector  $\mathbf{t}$  minimizing the cumulative squared projection error

$$\sum_{i=1}^n \left\| \mathbf{m}_i - \frac{\mathbf{R} \mathbf{M}_i + \mathbf{t}}{\mathbf{1}_z^T (\mathbf{R} \mathbf{M}_i + \mathbf{t})} \right\|^2, \quad (1)$$



**Fig. 1.** (a) The PnP setup for  $n = 4$  points. The unknown camera pose comprises the rotation matrix  $\mathbf{R} = [\mathbf{r}_x \ \mathbf{r}_y \ \mathbf{r}_z]^T$  and the vector  $\mathbf{t}$  from the camera center to the world frame. (b) Camera pose estimation with the proposed method for four coplanar points (colored dots). The inserted wireframe model reflects the pose estimate's quality.

where  $\mathbf{1}_z = [0 \ 0 \ 1]^T$  and  $(\mathbf{R}\mathbf{M}_i + \mathbf{t}) / (\mathbf{1}_z^T (\mathbf{R}\mathbf{M}_i + \mathbf{t}))$  is the projection of  $\mathbf{M}_i$  on the Euclidean plane  $Z = 1$  in the camera coordinate frame. The cost function (1) is often employed with non-linear least squares to iteratively refine an initial camera pose estimate; see, e.g., the use of the Gauss-Newton algorithm in [28,49].

Instead of comparing image projections, a slightly different formulation introduced in [17] considers the sum of squared differences between the measured direction vectors and those estimated from world points transformed in the camera frame, i.e.  $\sum_{i=1}^n \left\| \mathbf{u}_i - \frac{(\mathbf{R}\mathbf{M}_i + \mathbf{t})}{\|\mathbf{R}\mathbf{M}_i + \mathbf{t}\|} \right\|^2$ , where  $\mathbf{u}_i = \frac{\mathbf{m}_i}{\|\mathbf{m}_i\|}$  is the unit direction vector associated with the measured Euclidean projection  $\mathbf{m}_i$  in the camera frame.

### 1.1 Related work

Over the last two decades, numerous solutions to the PnP problem have been proposed. In many cases, authors target specializations of the problem for a specific number of points. Among them, of particular interest is the minimal case for  $n = 3$ , i.e. P3P, which can be solved analytically by assuming noise-free data [15,16,14,25]. When  $n > 3$ , multiple P3P solutions are typically used in the context of random sampling schemes [10] to identify mismatches and determine pose [42,32]. This approach is usually effective, however using minimal sets can also produce skewed estimates if the data points are very noisy [43].

For the generic (i.e.,  $n \geq 3$ ) PnP problem, Lepetit et al. [28] proposed EPnP, in which an initial pose estimate is obtained by rotating and translating the 3D points in their eigenspace and thereafter, solving the least squares (LS) formulation without the orthonormality constraints. Optionally, the estimate can be improved iteratively. Although EPnP can yield very good results, it nonetheless relies on an unconstrained LS estimate, which can be skewed by noise in the

data. It can also get trapped in local minima, particularly for small size inputs. There have been a few extensions to EPnP such as the one proposed by Ferraz et al. [9], whose main improvement upon the original algorithm is an iterative outlier rejection scheme that enforces a rank-1 constraint on the data matrix. A technique with procedural similarities to EPnP was proposed by Urban et al. [48] with a cost function that penalizes the misalignment of a reconstructed ray from the measured bearing and solved with ordinary least squares to obtain an initial solution. Amongst general solvers there is also a purely iterative method proposed by Lu et al. [34], which is initialized with a weak perspective approximation and refines the rotation matrix via successive solutions to the absolute orientation problem [20]. All the aforementioned methods can efficiently solve the general PnP problem, however they are all heuristic in nature and do not account for particular data configurations which yield multiple minima (e.g., P3P, P4P with coplanar points, etc.).

The largest class of PnP solvers in the literature comprises methods that render the problem unconstrained by utilizing a minimal degree of freedom (DoF) parameterization scheme for the rotation matrix [46,47]. The corresponding first order optimality conditions of the unconstrained optimization problem yield a complicated system of (at least cubic) polynomials. Solving polynomial systems with the aid of Gröbner basis solvers is a plausible approach [26,5,35]. Despite that these solvers tend to perform better in practice than their overall exponential rating [2], they cannot provide strict limits on execution time. This can be a serious drawback in applications such as robotic localization, which demand results within tight time constraints. One of the earliest such methods is that of Hesch and Roumeliotis [17], who proposed the Direct Least Squares (DLS) algorithm that employs the Cayley transform [8] to parametrize the rotation matrix and solve with resultants a quartic polynomial system obtained by the first order optimality conditions. DLS inherits a singularity for  $180^\circ$  rotations from its use of the Cayley transform [47]. A very similar solution was later proposed by Zheng et al. [51,52], with improved rotation parameterization for singularity avoidance and a Gröbner polynomial solver. In principle, both methods require a number of elimination steps to recover in the order of 40 solutions which are later substituted in the cost function to determine the global minimum.

Kukelova et al. [27] simplified the use of Gröbner basis solvers by introducing an automatic solver generator suited for computer vision problems. This generator revived the interest in polynomial solvers and instigated a new work cycle on the PnP problem. Bujnak et al. [6,7] used it to propose solutions for the P3P and P4P with unknown focal length problems and reported relatively short execution times. A more recent work that makes use of the automatic generator is that by Nakano [37]. He derives an optimality condition without Lagrange multipliers and proposes a globally optimal DLS method, parameterized by the Cayley representation. It is also worth mentioning here the RPnP method by Li et al. [29] which uses a relaxation of the PnP cost function by partitioning the points in triads so that each triad can minimize the P3P condition under a common unknown parameter. Owing to the partitioning, the first order condi-

tions of RPnP are an eighth-degree polynomial in a single variable that can be readily solved numerically. Despite its efficiency, the method is suboptimal due to the bias associated with the choice of two common points in all triads. This fact is acknowledged by Wang et al. [49], who extend RPnP by incorporating a Gauss-Newton refinement of its estimate.

## 1.2 The PnP as a quadratic program with quadratic constraints

The objective function (eq. (2)) that will be suggested in Section 2 along with the constraints on the elements of the rotation matrix can be used to cast the PnP problem as a quadratically constrained quadratic program (QCQP). Equality-constrained QCQPs are generally non-convex, NP-hard problems [41,21] and there has been a substantial number of contributions dealing with various manifestations of these problems in the optimization literature, e.g. [3,1,11,22,19].

To the best of our knowledge, only Schweighofer and Pinz [45] have approached the PnP problem as a QCQP. In that work, the problem is transformed into a semi-definite positive program that is solved with general purpose software. To achieve this, the original problem is substituted with a relaxation that seeks to maximize a lower bound of the global minimum expressed as a sum of squares polynomials. The method is generally effective, yet overly slow and requires slightly different approaches for special cases (e.g. coplanar points) as well as careful parameter tuning to achieve convergence.

## 1.3 Contributions

In this work we present an algorithm, called SQPnP, which casts PnP as a non-linear quadratic program (NLQP) with a cost function similar to these in [34,45]. However, instead of solving a relaxation of the problem or a polynomial system on the rotation parameters, our approach concentrates on special feasible points, from which it locates a small set of regional minima, guaranteed to contain the global one. Our contribution is two-fold:

1. Present a novel non-polynomial solver that possesses a number of desirable features generally not jointly present in existing methods:
  - It is truly generic and solves the PnP for any number and/or spatial arrangement of points without the need for any special treatment.
  - It is resilient to noise and recovers the global optima with the same or higher accuracy and consistency than those of state-of-the-art polynomial solvers.
  - Its complexity scales linearly with the number of points by virtue of invoking a small number of local searches, each completed in a bounded number of steps.
  - Its implementation is relatively short and simple, requiring only standard linear algebra operations.
2. Establish a novel mathematical framework which fully justifies the efficiency of the solver and provides a walk-through of the search space towards the global minimum.

## 2 Method

We consider the following cost function stemming from the sum of squared re-projection errors in (1):

$$\mathcal{E}^2 = \sum_{i=1}^n \left\| \mathbf{1}_z^T (\mathbf{R} \mathbf{M}_i + \mathbf{t}) \mathbf{m}_i - (\mathbf{R} \mathbf{M}_i + \mathbf{t}) \right\|^2 \quad (2)$$

The squared terms in eq. (2) penalize the distances between the reconstructed and the actual 3D points in the camera frame. This type of rearrangement in the cost function that replaces the reprojection error with a back-projection error is common in pertinent literature (e.g., [17,34]) and facilitates algebraic manipulations that lead to the elimination of the unknown translation  $\mathbf{t}$ .

Let now  $\mathbf{r} \in \mathbb{R}^9$  be the vector formed by stacking the rows of  $\mathbf{R}$ . For future use, we will denote the inverse operation as  $\text{mat}(\mathbf{r}) = \mathbf{R}$ . For a world point  $\mathbf{M}_i$ , we denote with  $\mathbf{A}_i \in \mathbb{R}^{3 \times 9}$  the matrix

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{M}_i^T & \mathbf{0}_3^T & \mathbf{0}_3^T \\ \mathbf{0}_3^T & \mathbf{M}_i^T & \mathbf{0}_3^T \\ \mathbf{0}_3^T & \mathbf{0}_3^T & \mathbf{M}_i^T \end{bmatrix}, \quad (3)$$

so that  $\mathbf{R} \mathbf{M}_i = \mathbf{A}_i \mathbf{r}$ . Substituting this into eq. (2) and making use of the fact that  $\mathbf{1}_z^T (\mathbf{R} \mathbf{M}_i + \mathbf{t}) \mathbf{m}_i = \mathbf{m}_i \mathbf{1}_z^T (\mathbf{R} \mathbf{M}_i + \mathbf{t})$ , the cost function can be factored into

$$\mathcal{E}^2 = \sum_{i=1}^n (\mathbf{A}_i \mathbf{r} + \mathbf{t})^T \mathbf{Q}_i (\mathbf{A}_i \mathbf{r} + \mathbf{t}), \quad (4)$$

where  $\mathbf{Q}_i$  is a symmetric positive semidefinite (PSD) matrix associated with the normalized Euclidean projection  $\mathbf{m}_i$  via  $\mathbf{Q}_i = (\mathbf{m}_i \mathbf{1}_z^T - \mathbf{I}_3)^T (\mathbf{m}_i \mathbf{1}_z^T - \mathbf{I}_3)$ .

By considering the first order optimality conditions,  $\mathbf{t}$  can be eliminated from the factorized cost of eq. (4). Zeroing the derivative of  $\mathcal{E}^2$  with respect to  $\mathbf{t}$  yields:

$$\sum_{i=1}^n \mathbf{Q}_i (\mathbf{A}_i \mathbf{r} + \mathbf{t}) = 0 \Leftrightarrow \left( \sum_{i=1}^n \mathbf{Q}_i \right) \mathbf{t} = - \left( \sum_{i=1}^n \mathbf{Q}_i \mathbf{A}_i \right) \mathbf{r}$$

**Proposition 1.** *The matrix  $\sum_{i=1}^n \mathbf{Q}_i$  is invertible.*

*Proof.* Provided in the supplementary material. □

Hence, the translation vector can be directly expressed in terms of rotation  $\mathbf{r}$  as

$$\mathbf{t} = \mathbf{P} \mathbf{r}, \quad (5)$$

where

$$\mathbf{P} = - \left( \sum_{i=1}^n \mathbf{Q}_i \right)^{-1} \left( \sum_{i=1}^n \mathbf{Q}_i \mathbf{A}_i \right). \quad (6)$$

Substituting eq. (5) in eq. (4) yields a squared error quadratic expression in terms of the rotation vector  $\mathbf{r}$  only, i.e.  $\mathcal{E}^2 = \mathbf{r}^T \boldsymbol{\Omega} \mathbf{r}$ , where  $\boldsymbol{\Omega}$  is the  $9 \times 9$  PSD matrix

$$\boldsymbol{\Omega} = \sum_{i=1}^n (\mathbf{A}_i + \mathbf{P})^T \mathbf{Q}_i (\mathbf{A}_i + \mathbf{P}). \quad (7)$$

Thus far,  $\mathbf{r}$  has been assumed to represent a valid rotation matrix. We may now cast the PnP problem as a NLQP<sup>3</sup> over an unknown vector denoted  $\mathbf{x}$  to emphasize that it can assume values in  $\mathbb{R}^9$  that do not correspond to valid rotation matrices. Unless otherwise specified, we reserve the notation  $\mathbf{r}$  to imply a rotation matrix. The NLQP is then

$$\underset{\mathbf{x} \in \mathbb{R}^9}{\text{minimize}} \quad \mathbf{x}^T \boldsymbol{\Omega} \mathbf{x} \quad \text{s.t.} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0}_6, \quad (8)$$

where  $\boldsymbol{\Omega}$  is given by eq. (7) and  $\mathbf{h}(\mathbf{x}) \in \mathbb{R}^6$  is the vector of constraints ensuring that when  $\mathbf{h}(\mathbf{x}) = \mathbf{0}_6$ , vector  $\mathbf{x}$  represents a rotation matrix:

$$\mathbf{h}(\mathbf{x}) = [\mathbf{x}_{1:3}^T \mathbf{x}_{1:3} - 1, \mathbf{x}_{4:6}^T \mathbf{x}_{4:6} - 1, \mathbf{x}_{1:3}^T \mathbf{x}_{4:6}, \mathbf{x}_{1:3}^T \mathbf{x}_{7:9}, \mathbf{x}_{4:6}^T \mathbf{x}_{7:9}, \det(\text{mat}(\mathbf{x})) - 1]^T,$$

where  $\mathbf{x}_{i:j}$  denotes the subvector of  $\mathbf{x}$  from the  $i^{\text{th}}$  to the  $j^{\text{th}}$  component. Note that the unit norm constraint for  $\mathbf{x}_{7:9}$  is redundant and therefore omitted from the components of  $\mathbf{h}$ . These constraints will be henceforth referred to as proper orthonormality constraints and  $\mathbf{h}(\mathbf{x})$  as the proper orthonormality function.

**Proposition 2.** Define  $\mathbf{H}_{\mathbf{x}} \equiv \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}} \in \mathbb{R}^{6 \times 9}$  to be the Jacobian matrix of the proper orthonormality function at  $\mathbf{x}$ . If  $\text{rank}(\text{mat}(\mathbf{x})) \geq 2$ , then  $\text{rank}(\mathbf{H}_{\mathbf{x}}) = 6$ .

*Proof.* Provided in the supplementary material.  $\square$

Proposition 2 will be useful in showing that the quadratic programming algorithm adapted for our method will always involve a non-singular system matrix (Section 2.2); additionally, it implies that when  $\text{rank}(\text{mat}(\mathbf{x})) \geq 2$ , the null space of  $\mathbf{H}_{\mathbf{x}}$  is 3-dimensional, which reflects the fact that rotations have 3 DoF. Note that the null space of  $\mathbf{H}_{\mathbf{x}}$  is also the tangent space of the 3D rotation group  $SO(3)$  at  $\mathbf{x}$ , hence the two terms will be interchangeable throughout this text.

## 2.1 Minima on the 8-sphere

Rather than parametrizing the rotation with a minimal representation, we next consider the problem of finding feasible solutions, i.e. proper orthogonal matrices, as 9-vectors for the NLQP of (8). Clearly, the feasible set of the constrained program lies on the hypersphere of radius  $\sqrt{3}$  centered at the origin of  $\mathbb{R}^9$ , henceforth referred to as the 8-sphere and simply denoted by  $\mathbb{S}^8$ . For simplicity, we assume that  $\boldsymbol{\Omega}$  has exactly nine non-vanishing eigenvalues. However, as will be explained in Section 2.3, the results that follow also hold in the general case

<sup>3</sup> For brevity, the determinant constraint we employ here is cubic. Alternatively, it can be imposed with 3 quadratic constraints, thereby the formulation becomes a QCQP.

where  $\mathbf{\Omega}$  is singular. Consider now a relaxed, more general problem, specifically that of finding the minima of the quadratic function defined on  $\mathbb{S}^8$ :

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{\Omega} \mathbf{x}, \mathbf{x} \in \mathbb{S}^8. \quad (9)$$

It is a well-known fact that the stationary points of a unit-norm constrained quadratic function are the eigenvectors of  $\mathbf{\Omega}$  [13, Thm. 8.5], which we will henceforth denote  $\mathbf{e}_1, \dots, \mathbf{e}_9$ , in descending order of the eigenvalues  $s_1 > \dots > s_9$ . Thus, the function  $f$  in eq. (9) has exactly 18 distinct stationary points  $\mathbf{x}_1, \dots, \mathbf{x}_{18} \in \mathbb{S}^8$  that correspond to the 9 eigenvectors of  $\mathbf{\Omega}$  scaled by  $\pm\sqrt{3}$ , i.e.

$$\mathbf{x}_1 = +\sqrt{3}\mathbf{e}_1, \dots, \mathbf{x}_9 = +\sqrt{3}\mathbf{e}_9, \mathbf{x}_{10} = -\sqrt{3}\mathbf{e}_1, \dots, \mathbf{x}_{18} = -\sqrt{3}\mathbf{e}_9. \quad (10)$$

In the rest of this section, we establish a close relationship between the feasible local minima and the solutions of the nearest orthogonal matrix problem (NOMP) [18] in a special spherical region around a minimizer of  $f$ . Through Propositions 3, 4, 5, we ensure that the aforementioned spherical region always contains feasible solutions of the fully constrained QCQP and, through Proposition 6, that the pertinent minima can be exhaustively traced from the solutions of the NOMP associated with the minimizer of  $f$ .

**Proposition 3.** *The function  $f(\mathbf{x}) = \mathbf{x}^T \mathbf{\Omega} \mathbf{x}$ ,  $\mathbf{x} \in \mathbb{S}^8$ , is convex in a region of the 8-sphere of radius  $\sqrt{3}$  that contains a local minimum and the spherical points that form angles less than  $90^\circ$  with the minimum.*

*Proof.* If  $\mathbf{e}$  is a local minimizer of  $f$ , then it is an eigenvector of  $\mathbf{\Omega}$ . Since  $\mathbf{\Omega}$  is a PSD matrix, its eigenvectors are mutually orthogonal. Thus, the nearest inflection point to  $\mathbf{e}$  must also be an eigenvector of  $\mathbf{\Omega}$  and forms an angle of at least  $90^\circ$  with  $\mathbf{e}$ .  $\square$

**Proposition 4.** *Let  $\mathbf{e} \in \mathbb{R}^9$  s.t.  $\|\mathbf{e}\| = 1$ . If  $\mathbf{r}$  represents a rotation minimizing*

$$\mathbf{r} = \underset{\text{mat}(\mathbf{x}) \in \mathcal{SO}(3)}{\text{argmin}} \left\| \mathbf{x} - \sqrt{3}\mathbf{e} \right\|^2, \quad (11)$$

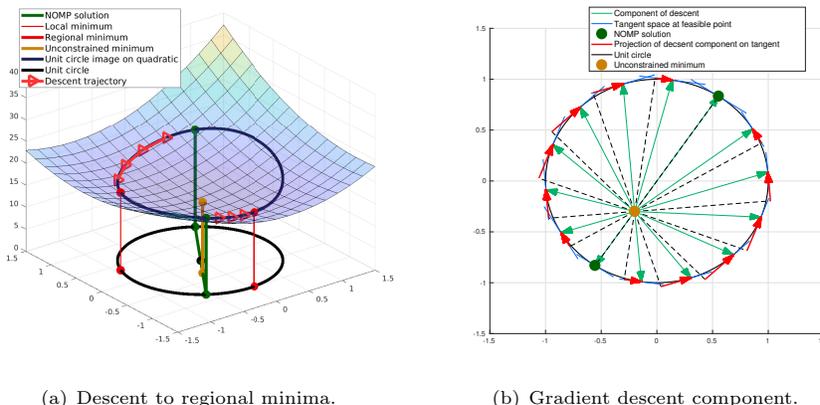
*then the angle between vectors  $\mathbf{e}$  and  $\mathbf{r}$  is strictly less than  $71^\circ$ .*

*Proof.* Provided in the supplementary material.  $\square$

Propositions 3 and 4 suggest that we can identify  $90^\circ$  regions of convexity of  $f$  which are guaranteed to contain a non-empty set of feasible solutions, i.e. rotations. For  $\mathbf{x} \in \mathbb{R}^9$ , the Euclidean norm  $\|\mathbf{x}\|$  equals the Frobenius norm  $\|\text{mat}(\mathbf{x})\|_{\text{F}}$ . Hence, eq. (11) amounts to finding the orthogonal matrix minimizing the Frobenius distance from a given matrix, i.e., the NOMP [18].

**Proposition 5.** *For  $\mathbf{e} \in \mathbb{R}^9$  with  $\|\mathbf{e}\|^2 = 1$ , there exist exactly 4 vectors  $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3$  and  $\boldsymbol{\xi}_4$  with  $\text{mat}(\boldsymbol{\xi}_i) \in \mathcal{O}(3)$  in the  $90^\circ$  region of  $\sqrt{3}\mathbf{e}$  for which the vectors  $\sqrt{3}\mathbf{e} - \boldsymbol{\xi}_i$  are orthogonal to the tangent space of  $\mathcal{O}(3)$  at  $\boldsymbol{\xi}_i$ .*

*Proof.* Provided in the supplementary material.  $\square$



**Fig. 2.** Illustration of regional search for minima from the solutions of the NOMP (antipodal dark green points on the unit circle) for the case where  $\mathbf{R} \in \mathcal{SO}(2)$ . (a) Descent from the two NOMP solutions leads to the two local minima shown in red. (b) The component of the gradient responsible for descent (light green arrows actually drawn in the direction of ascent for illustration purposes) at a feasible point is the direction vector from the minimum of the parabola to that point; the projection of this component on the tangent at the feasible point changes direction on the NOMP solutions, thus ensuring there can be at most one minimum and/or maximum in a feasible path between any two such solutions.

Note here that  $\mathcal{O}(3)$  is the orthogonal group of dimension 3, consisting of all  $3 \times 3$  orthogonal matrices. These matrices have determinant either 1 or -1. The PnP quadratic program of eq. (8) can be equivalently cast with orthogonality constraints, owing to the fact that the value of  $f$  for an orthogonal matrix  $\xi$  with determinant -1 is the same as the one for the rotation  $\mathbf{r} = -\xi$ . Thus, by focusing on simply orthogonal matrices in the region of  $\sqrt{3}\mathbf{e}$  allows us to study the original program in the  $90^\circ$  region of  $\sqrt{3}\mathbf{e}$  where the behavior of  $f$  is known.

In the special case where  $\mathbf{e}$  is a minimizing eigenvector, the direction of the projection of  $\sqrt{3}\mathbf{e} - \xi_i$  on the tangent space of the sphere at  $\xi_i$  should be the component of the gradient of  $f$  responsible for descent (Figure 2(b)), owing to the fact that  $f$  is convex in the  $90^\circ$  region of  $\sqrt{3}\mathbf{e}$ . It follows that the projection of the gradient's component of descent on the tangent space of  $\mathcal{O}(3)$  changes its direction at  $\xi_i$ ; based on the latter we conclude that for any feasible path between these solutions, there can be at most one minimum and/or maximum:

**Proposition 6.** *For a minimizing eigenvector  $\mathbf{e}$  of  $\Omega$ , the feasible minimum in  $\mathcal{O}(3)$  inside the  $90^\circ$  region of  $\sqrt{3}\mathbf{e}$  can be reached by descending from at least one of the vectors  $\xi_1, \xi_2, \xi_3$  and  $\xi_4$  of Proposition 5.*

*Proof.* Provided in the supplementary material.  $\square$

Proposition 6 is sufficient to enable safe navigation to the global minimum. The eigenvectors of  $\Omega$  divide the sphere in overlapping  $90^\circ$  regions associated

with either inflection or saddle points on the surface of  $f$ . Clearly, the best candidates to begin the search for the global minimum of the fully constrained problem are the global minima of  $f$  associated with eigenvector  $\mathbf{e}_9$ . The feasible regional minimizers are located by initiating a descent from the orthogonal matrix vector nearest to the minimum of  $f$ . Similar searches may be performed in the regions of eigenvectors that may simply be saddle points of  $f$ , owing to the overlap of the pertinent regions with regions associated with minimizers. Figure 2 visualizes this process in 3D for the special PnP case where  $\mathbf{R} \in \mathcal{SO}(2)$ .  $\mathcal{SO}(2)$  is effectively a 2D circle and  $f$  is a quadratic translated from the origin.

In practice, we do not need to thoroughly examine all eigenvectors as the fact that the corresponding stationary values of  $f$  are the eigenvalues of  $\mathbf{\Omega}$  scaled by 3 can be used to avoid unnecessary regional searches [13]. Thus, we begin from the region of  $\sqrt{3}\mathbf{e}_9$  and, if the recovered regional minimum has a value above  $3s_8$ , we examine the region  $\sqrt{3}\mathbf{e}_8$  and repeat in ascending order of eigenvalues, until one of the new minima is less than the remaining (scaled by 3) eigenvalues, or the set of eigenvectors is exhausted.

## 2.2 Sequential quadratic programming

Sequential quadratic programming (SQP) is an iterative technique for solving non-linear constrained optimization problems [12,38,4]. The core idea in SQP is to approximate the cost function with a quadratic and the constraints with linear functions in order to produce a linearly constrained quadratic program (LCQP) which can be solved analytically with a linear system that comprises the first order conditions of the Lagrangian function and the linearized constraints. The solution of the linearly constrained quadratic program yields a perturbation in the vector of unknowns and the vector of Lagrange multipliers at the solution. The non-linear program is subsequently approximated at the new estimate and the process is repeated until convergence.

In the case of the NLQP for the PnP problem (8), we introduce the SQP approximation at a feasible point  $\mathbf{r}$  on the 8-sphere of radius  $\sqrt{3}$ . Since the cost function is already a quadratic, we express it in terms of the difference  $\boldsymbol{\delta} = \mathbf{x} - \mathbf{r}$  and linearize the function  $\mathbf{h}(\mathbf{x})$  using its first order Taylor approximation:

$$\underset{\boldsymbol{\delta} \in \mathbb{R}^9}{\text{minimize}} \quad \boldsymbol{\delta}^T \mathbf{\Omega} \boldsymbol{\delta} + 2\mathbf{r}^T \mathbf{\Omega} \boldsymbol{\delta} \quad \text{s.t.} \quad \mathbf{H}_\mathbf{r} \boldsymbol{\delta} = -\mathbf{h}(\mathbf{r}). \quad (12)$$

The first order conditions of the Lagrangian function along with the linear constraints of the LCQP in eq. (12) yield a linear system, the solution of which is a descent direction that converges towards<sup>4</sup> or stays on a trajectory of feasible solutions. For the solution  $\hat{\boldsymbol{\delta}}$  of the linear system, the NLQP is approximated at a new point  $\mathbf{r}' = \mathbf{r} + \hat{\boldsymbol{\delta}}$  and a new descent direction is obtained. The process is repeated until the norm of  $\hat{\boldsymbol{\delta}}$  drops below a threshold.

<sup>4</sup> During the first few steps of SQP, the solutions may not be entirely feasible due to inaccuracies in the linear approximations of the constraints.

**Proposition 7.** *Let  $\mathbf{r} \in \mathbb{R}^9$  be the estimate of the rotation matrix which may not be feasible at some step of the SQP. If  $\text{rank}(\mathbf{\Omega}) \geq 3$ , then the linearly constrained quadratic program of eq. (12) has a unique solution in  $\mathbb{R}^9$ .*

*Proof.* Provided in the supplementary material.  $\square$

Proposition 7 ensures that the SQP linear system will have a unique solution in every step of the process. This is because in the presence of non-degenerate data,  $\text{rank}(\mathbf{\Omega})$  will be at least 3 for the problem to be fully constrained.

To solve the PnP, the aim is to recover the regional minima associated initially with the global minimizer  $\mathbf{e}_9$  of the quadratic  $f(\mathbf{x}) = \mathbf{x}^T \mathbf{\Omega} \mathbf{x}$  on the 8-sphere and, if necessary, proceed to repeat the process in the region of the next eigenvector in ascending order of eigenvalues (cf. Section 2.1).

As explained in Section 2.1, we may recover the regional minimum associated with  $\mathbf{e}$  by descending along the feasible path from the solutions  $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3$  and  $\boldsymbol{\xi}_4$  (cf. Props. 5, 6) of the NOMP associated with  $\sqrt{3}\mathbf{e}$ . We empirically determined that descending only from the two nearest of  $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3, \boldsymbol{\xi}_4$  to  $\sqrt{3}\mathbf{e}$  allows the method to converge to the global minimum. Finding the two nearest solutions of the NOMP related to eigenvector  $\mathbf{e}$  is equivalent to finding the rotation matrices  $\mathbf{r}_1, \mathbf{r}_2$  nearest to  $\sqrt{3}\mathbf{e}, -\sqrt{3}\mathbf{e}$  respectively. Thus, each inspected eigenvector  $\mathbf{e}$  contributes to the overall search with two minima recovered via SQP descent from the following rotations:

$$\mathbf{r}_1 = \underset{\text{mat}(\mathbf{x}) \in \mathcal{SO}(3)}{\text{argmin}} \left\| \mathbf{x} - \sqrt{3}\mathbf{e} \right\|^2, \quad \mathbf{r}_2 = \underset{\text{mat}(\mathbf{x}) \in \mathcal{SO}(3)}{\text{argmin}} \left\| \mathbf{x} + \sqrt{3}\mathbf{e} \right\|^2. \quad (13)$$

### 2.3 The general case

We have thus far assumed that the PSD data matrix  $\mathbf{\Omega}$  has an empty null space. However, this is not generally the case, particularly when  $n$  is small. In these cases, the intersection of the null space with the sphere of radius  $\sqrt{3}$ , referred to here as the null sphere, is treated as a generalized minimum. This “minimum” is a flat region for  $f$ , which can only contain a finite number of solutions. However, we know that the  $90^\circ$  regions of the null space basis vectors are not entirely flat due to overlap with the corresponding regions of eigenvectors with non-vanishing eigenvalues. Based on this, we generalize the approach of Section 2.2 and devise a number of SQPs equal to the number of null space vectors with feasible starting points obtained as in eq. (13). More formally, suppose that the last (in descending eigenvalue order)  $k$  eigenvectors of  $\mathbf{\Omega}$  are the null-space basis,  $\text{null}(\mathbf{\Omega}) = \langle \mathbf{e}_{10-k}, \dots, \mathbf{e}_9 \rangle$ ,  $k \geq 1$ . We then perform  $2k$  SQPs with starting points

$$\mathbf{r}_i = \underset{\text{mat}(\mathbf{x}) \in \mathcal{SO}(3)}{\text{argmin}} \left\| \mathbf{x} - (-1)^{\lfloor (i-1)/k \rfloor} \sqrt{3}\mathbf{e}_{9-k+i-\lfloor i/k \rfloor k} \right\|^2, \quad (14)$$

where  $i \in \{1, \dots, 2k\}$ . Note here that large null spaces (up to 6 basis vectors) are typically associated with small numbers of points (up to 6) and therefore multiple solutions may exist. In these cases, the typical treatment involving the positive depth test applies. The overall procedure is detailed in Algorithm 1.

### 3 The SQPnP algorithm

SQPnP is described in pseudocode as Algorithms 1 and 2 that jointly yield the global minima of the PnP problem. Algorithm 1 solves the PnP problem by delivering a list of minimizers that contains the global minima. The algorithm computes the PSD data matrix  $\mathbf{\Omega}$ , the matrix  $\mathbf{P}$  required by eq. (5) for the computation of the translation vector, and the feasible starting points from which it initiates iterative searches using SQP, as detailed by Algorithm 2. SQP typically converges within 10 iterations, hence the recommendation  $T \geq 15$ . Similarly, we empirically determined that  $10^{-5}$  suffices as the perturbation norm tolerance.

---

**Algorithm 1** SQPnP: SolvePnP

---

**Require:**

Number of points,  $n \geq 3$   
World points  $\mathbf{M}_i \in \mathbb{R}^3$ ,  $1 \leq i \leq n$   
Projections  $\mathbf{m}_i = [x_i \ y_i \ 1]^T$ ,  $1 \leq i \leq n$   
  
Perturbation tolerance,  $\epsilon \leq 10^{-5}$   
Maximum number of iterations,  $T \geq 15$

$\{\mathbf{\Omega}, \mathbf{P}\} \leftarrow$  eqs. (6), (7)  
 $\{\mathbf{e}_1, \dots, \mathbf{e}_9, s_1, \dots, s_9\} \leftarrow \text{SVD}(\mathbf{\Omega})$   
 $\{\mathbf{e}_{10-k}, \dots, \mathbf{e}_9\} \leftarrow \underset{\mathbf{x} \in \mathbb{S}^8}{\text{argmin}} \mathbf{x}^T \mathbf{\Omega} \mathbf{x}$   
**for**  $i \leftarrow 1$  **to**  $2k$  **do**  
     $\mu \leftarrow \lfloor (i-1)/k \rfloor$   
     $\nu \leftarrow 9 - k + i - \lfloor i/k \rfloor k$   
     $\mathbf{r}_i \leftarrow \underset{\mathbf{mat}(\mathbf{x}) \in \mathcal{SO}(3)}{\text{argmin}} \|\mathbf{x} - (-1)^\mu \sqrt{3} \mathbf{e}_\nu\|^2$   
     $\hat{\mathbf{r}}_i \leftarrow \text{SOLVESQP}(\mathbf{r}_i, \mathbf{\Omega}, \epsilon, T)$   
     $\mathcal{E}_i^2 \leftarrow \hat{\mathbf{r}}_i^T \mathbf{\Omega} \hat{\mathbf{r}}_i$   
**end for**  
  
**while**  $\min\{\mathcal{E}_1^2, \dots, \mathcal{E}_{2k}^2\} \geq s_{9-k}$  **do**  
    **for**  $i \leftarrow 1$  **to**  $2$  **do**  
         $\mathbf{r}_{2k+i} \leftarrow \underset{\mathbf{mat}(\mathbf{x}) \in \mathcal{SO}(3)}{\text{argmin}} \|\mathbf{x} - (-1)^i \sqrt{3} \mathbf{e}_{9-k}\|^2$   
         $\hat{\mathbf{r}}_{2k+i} \leftarrow \text{SOLVESQP}(\mathbf{r}_{2k+i}, \mathbf{\Omega}, \epsilon, T)$   
         $\mathcal{E}_{2k+i}^2 \leftarrow \hat{\mathbf{r}}_{2k+i}^T \mathbf{\Omega} \hat{\mathbf{r}}_{2k+i}$   
    **end for**  
     $k \leftarrow k + 1$   
**end while**  
**return**  $\hat{\mathbf{r}}_1, \dots, \hat{\mathbf{r}}_{2k}, \mathcal{E}_1^2, \dots, \mathcal{E}_{2k}^2$

---



---

**Algorithm 2** SQPnP: SolveSQP

---

**Require:**

Starting point  $\mathbf{r} \in \mathbb{R}^9$ , s.t.  $\text{mat}(\mathbf{r}) \in \mathcal{SO}(3)$   
Data matrix,  $\mathbf{\Omega} \in \mathbb{R}^{9 \times 9}$ ,  $\mathbf{\Omega} \succeq \mathbf{0}$   
Tolerance in perturbation estimate norm,  $\epsilon \leq 10^{-5}$   
Maximum number of iterations,  $T \geq 15$

step  $\leftarrow 0$   
 $\hat{\mathbf{r}} \leftarrow \mathbf{r}$   
**repeat**  
     $\mathbf{H}_{\hat{\mathbf{r}}} \leftarrow \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{r}}}$   
     $\begin{bmatrix} \hat{\delta} \\ \hat{\lambda} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{\Omega} & \mathbf{H}_{\hat{\mathbf{r}}}^T \\ \mathbf{H}_{\hat{\mathbf{r}}} & \mathbf{0}_{6 \times 6} \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{\Omega} \hat{\mathbf{r}} \\ -\mathbf{h}(\hat{\mathbf{r}}) \end{bmatrix}$   
     $\hat{\mathbf{r}} \leftarrow \hat{\mathbf{r}} + \hat{\delta}$   
    step  $\leftarrow$  step + 1  
**until**  $\|\hat{\delta}\| < \epsilon$  **or** step  $> T$   
**return**  $\hat{\mathbf{r}}$

---

## 4 Experiments

Based on Matlab implementations, we report next results from the comparison of SQPnP with several prominent PnP methods, namely DLS [17], LHM [34], RPnP [29], OPnP [51], MLPnP [48], REPPnP [9], EPnP [28] and optDLS [37]; both [17,37] include the three rotations by  $90^\circ$  preprocessing to avoid the Cayley singularity. Our investigation focuses primarily on the reprojection error achieved by these methods for increasing numbers of points  $n$  and amounts of noise. However, acknowledging that execution time is crucial for many practical applications, we also provide timing measurements with the forewarning that Matlab implementations are less efficient than implementations in compiled languages such as C++. We exclude UPnP [24] from our comparison since, as detailed in [37], it is a suboptimal method worse than optDLS and OPnP.

SQPnP calls for i) the computation of the data matrix  $\mathbf{\Omega}$  in eq. (7), which is linear in the number of points and ii) the search for minima via SQPs which are bounded by a finite number of iterations and starting points. Therefore, the execution time of SQPnP has a baseline offset which largely depends on the cost of the linear system solution required in every SQP iteration (cf. Algorithm 2). Our current implementation employs Matlab’s general-purpose linear system solver `linsolve`. However, this operation can be considerably accelerated by exploiting the special structure of the system’s matrix and  $\mathbf{H}_x$  in particular. The nearest orthogonal approximation problem in eq. (14) is solved without costly matrix factorizations using the FOAM algorithm, as discussed in [30,31].

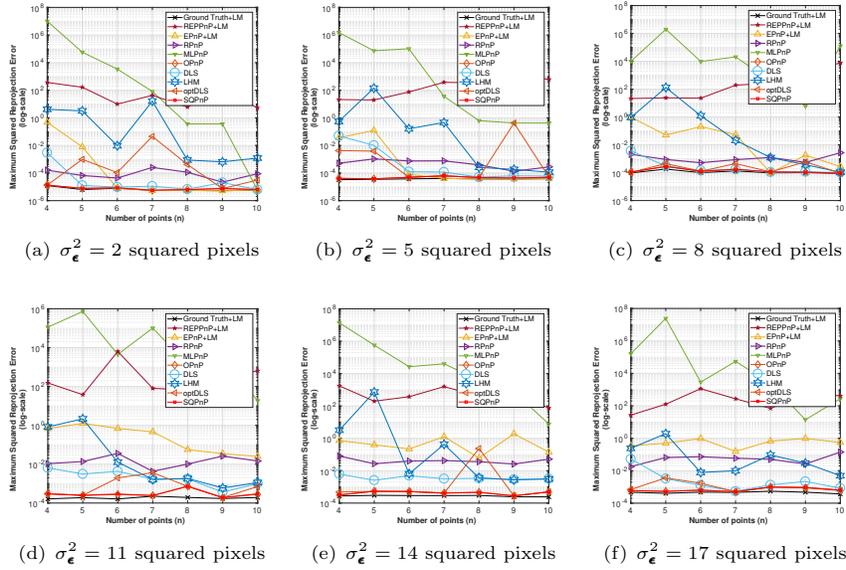
### 4.1 Synthetic experiments

**Procedure.** In our experiments, Euclidean quantities are expressed in units of meters. World 3D points were randomly sampled from an isotropic Gaussian distribution with SD 3, i.e.  $\mathbf{M}_i \sim \mathcal{N}(\bar{\mathbf{M}}, 3^2 \mathbf{I}_3)$ , where  $\bar{\mathbf{M}} \equiv [3/4 \ 3/4 \ 12]^T$ . Similarly, camera poses comprising position  $\mathbf{b}$  and MRP [47] orientation parameters  $\boldsymbol{\psi}$  in the world frame are sampled from a zero-mean 6D Gaussian distribution

$$\begin{bmatrix} \mathbf{b} \\ \boldsymbol{\psi} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}_6, \begin{bmatrix} \sigma_b^2 \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \sigma_\psi^2 \mathbf{I}_3 \end{bmatrix}\right). \quad (15)$$

The values chosen for the standard deviations were sufficiently small (i.e.,  $\sigma_b = 0.2$  and  $\sigma_\psi = 0.05$ ) to ensure that the generated points will always be in front of the simulated camera, assumed to have a focal length  $f = 1400$  pixels and image size  $1800 \times 1800$ .

Using six different levels of additive Gaussian noise with  $\sigma_\epsilon^2 \in \{2, 5, 8, 11, 14, 17\}$  squared pixels, we performed experiments summarized in the plots of Fig. 3. For each noise level, we generated 100 random 3D points. Then, for every  $n \in \{4, \dots, 10\}$  we randomly sampled a population of 500 sets of  $n$  3D points each, generated a camera pose with eq. (15) for each set, projected all sets on the image plane and perturbed the projections with noise  $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}_2, \sigma_\epsilon^2 \mathbf{I}_2)$ .



**Fig. 3.** Plots of maximum squared reprojection error for 500 executions of each PnP solver on  $n$  random points,  $4 \leq n \leq 10$ . For each  $n$ , points are repeatedly sampled from a previously generated point population contaminated with additive Gaussian noise. Each plot represents the results obtained by points drawn from a different population and whose projections were contaminated with zero-mean Gaussian noise of variance  $\sigma_{\epsilon}^2 \in \{2, 5, 8, 11, 14, 17\}$  squared pixels (top left to bottom right). Notice the different scales in the vertical axes of the plots.

For every set in a certain population, we executed all PnP solvers under comparison with default parameters<sup>5</sup> and calculated the maximum squared reprojection error for each solver across these executions. We also determined the reprojection error corresponding to the maximum likelihood estimate, obtained by minimizing the total reprojection error for each set’s noisy points with the Levenberg-Marquardt (LM) algorithm initiated at the true pose.

The maximum was preferred over the average squared reprojection error since we are primarily interested in demonstrating the consistency of our solver in reaching a squared error that is similar to that of the maximum likelihood estimate. Nevertheless, plots of the average squared reprojection errors corresponding to exactly the same experiments can be found in the supplementary material. For completeness, the supplementary material also includes plots of the pose translational and rotational errors for the same experiments.

It is finally noted that our experiments concern relatively small numbers of points since, in practice, these are the typical sample sizes that yield candidate solutions in sampling-based camera resectioning [10].

**Results.** The plots of Fig. 3 illustrate the maximum total squared reprojection errors (cf. (1)) in terms of  $n$  for each PnP solver applied to the samples drawn

<sup>5</sup> SQnPnP employed maximum iterations  $T = 15$  and tolerance  $\epsilon = 10^{-8}$ .

**Table 1.** Average and median execution times (ms) of several PnP solvers implemented in Matlab, computed across all executions for every  $4 \leq n \leq 10$ . Time for any additional non-linear refinement was not taken into account.

		<sup>(ours)</sup> PnP Solver								
		SQPnP	optDLS	LHM	DLS	OPnP	MLPnP	RPnP	EPnP	REPPnP
Time	Mean	2.7	3.5	4.7	3.5	14.0	2.3	0.7	1.1	2.1
	Median	2.0	3.5	4.7	3.5	14.1	2.4	0.8	1.1	2.1

from the populations generated for a certain level of additive noise. The results obtained by the EPnP and REPPnP were further improved with the LM algorithm. Moreover, the plots incorporate the reprojection error corresponding to the maximum likelihood estimates (labeled “Ground Truth+LM”).

Using as metric the infinity norm (i.e., maximum) of the reprojection errors pertaining to the multiple executions of the methods being compared, ensures that the plots will reflect the repeatability of the methods in approaching the minimum for a given  $n$ . As expected, we observe that SQPnP does not deviate from the ground truth more than  $10^{-3}$  in any of the 500 executions, regardless of the number of points or levels of additive noise. SQPnP consistently approaches the minimum similarly to the polynomial solvers OPnP, DLS and optDLS that provide strong theoretical guarantees of finding the global minimizer. In doing so, SQPnP attains better accuracy compared to DLS and optDLS, and very similar to that of OPnP. OPnP performs well in all experiments, albeit at a much higher computational cost in comparison to SQPnP (cf. Table 1). In contrast, methods such as EPnP, REPPnP and MLPnP that employ unconstrained LS formulations, tend to perform equally well for small amounts of noise (cf. Figs. 3(a), 3(b)), yet give rise to erratic convergence patterns when the noise increases. The execution times of the various PnP methods are in Table 1, showing that SQPnP is competitive also in terms of computational cost.

## 5 Conclusion

This paper presented SQPnP, a fast and globally convergent non-polynomial PnP solver. SQPnP casts the PnP problem as a quadratically constrained quadratic program and solves it by conducting local searches in the vicinity of special feasible points from which the global minima are located in a few steps. SQPnP admits a simple implementation that requires standard linear algebra operations and incurs a low computational cost. Comparative experiments confirm that SQPnP performs competitively to state-of-the-art PnP solvers and consistently recovers the true camera pose regardless of the noise and the spatial arrangement of input data. A C++ implementation of SQPnP is available at <https://github.com/terzakig/sqnpn>.

**Acknowledgements.** M. Lourakis has been funded by the EU H2020 Programme under Grant Agreement No. 826506 (sustAGE).

## References

1. Andersen, E.D., Roos, C., Terlaky, T.: On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming* **95**(2), 249–277 (2003)
2. Bardet, M., Faugere, J.C., Salvy, B.: On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In: *Proceedings of the International Conference on Polynomial System Solving*. pp. 71–74 (2004)
3. Beck, A., Eldar, Y.C.: Strong duality in nonconvex quadratic optimization with two quadratic constraints. *SIAM Journal on Optimization* **17**(3), 844–860 (2006)
4. Boggs, P.T., Tolle, J.W.: Sequential quadratic programming. *Acta Numerica* **4**, 1–51 (1995)
5. Buchberger, B., Kauers, M.: Groebner basis. *Scholarpedia* **5**(10), 7763 (2010)
6. Bujnak, M., Kukulova, Z., Pajdla, T.: A general solution to the P4P problem for camera with unknown focal length. In: *Conference on Computer Vision and Pattern Recognition*. pp. 1–8. IEEE (2008)
7. Bujnak, M., Kukulova, Z., Pajdla, T.: New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion. In: *Asian Conference on Computer Vision*. pp. 11–24. Springer (2010)
8. Cayley, A.: Sur quelques propriétés des déterminants gauches. *Journal für die reine und angewandte Mathematik* **32**, 119–123 (1846)
9. Ferraz, L., Binefa, X., Moreno-Noguer, F.: Very fast solution to the PnP problem with algebraic outlier rejection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 501–508 (2014)
10. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* **24**(6), 381–395 (1981)
11. Floudas, C.A., Visweswaran, V.: Quadratic optimization, pp. 217–269. Springer (1995)
12. Forst, W., Hoffmann, D.: Optimization — theory and practice. Springer Science & Business Media (2010)
13. Fraleigh, J., Beauregard, R.: Linear algebra. Addison-Wesley (1995)
14. Gao, X.S., Hou, X.R., Tang, J., Cheng, H.F.: Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(8), 930–943 (2003)
15. Grunert, J.: Das pothenotische Problem in erweiterter Gestalt nebst über seine Anwendungen in Geodäsie. *Grunerts Archiv für Mathematik und Physik* (1841)
16. Haralick, R.M., Joo, H., Lee, C.N., Zhuang, X., Vaidya, V.G., Kim, M.B.: Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man, and Cybernetics* **19**(6), 1426–1446 (1989)
17. Hesch, J.A., Roumeliotis, S.I.: A direct least-squares (DLS) method for PnP. In: *International Conference on Computer Vision*. pp. 383–390. IEEE (2011)
18. Higham, N.J.: *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics (2008)
19. Hmam, H.: Quadratic optimisation with one quadratic equality constraint. Tech. Rep. 2416, Defence Science and Technology Organisation, Australia (2010)
20. Horn, B.K., Hilden, H.M., Negahdaripour, S.: Closed-form solution of absolute orientation using orthonormal matrices. *JOSA A* **5**(7), 1127–1135 (1988)
21. Johnson, D.S., Garey, M.R.: *Computers and intractability: A guide to the theory of NP-completeness*. WH Freeman (1979)

22. Kim, S., Kojima, M.: Second order cone programming relaxation of nonconvex quadratic optimization problems. *Optimization Methods and Software* **15**(3-4), 201–224 (2001)
23. Klein, G., Murray, D.: Parallel tracking and mapping on a camera phone. In: *IEEE International Symposium on Mixed and Augmented Reality*. pp. 83–86. IEEE (2009)
24. Kneip, L., Li, H., Seo, Y.: UPnP: An optimal  $O(n)$  solution to the absolute pose problem with universal applicability. In: *European Conference on Computer Vision (ECCV)*. pp. 127–142 (2014)
25. Kneip, L., Scaramuzza, D., Siegwart, R.: A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2969–2976. IEEE (2011)
26. Kukulova, Z., Bujnak, M., Pajdla, T.: Polynomial eigenvalue solutions to minimal problems in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(7), 1381–1393 (2012)
27. Kukulova, Z., Bujnak, M., Pajdla, T.: Automatic generator of minimal problem solvers. In: *European Conference on Computer Vision*. pp. 302–315. Springer (2008)
28. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: An accurate  $O(n)$  solution to the PnP problem. *International Journal of Computer Vision* **81**(2), 155 (2009)
29. Li, S., Xu, C., Xie, M.: A robust  $O(n)$  solution to the perspective-n-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(7), 1444–1450 (2012)
30. Lourakis, M.: An efficient solution to absolute orientation. In: *International Conference on Pattern Recognition (ICPR)*. pp. 3816–3819 (Dec 2016)
31. Lourakis, M., Terzakis, G.: Efficient absolute orientation revisited. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 5813–5818 (2018)
32. Lourakis, M., Zabulis, X.: Model-based pose estimation for rigid objects. In: *International Conference on Computer Vision Systems*. pp. 83–92. Springer (2013)
33. Lourakis, M.I.A., Argyros, A.A.: Efficient, causal camera tracking in unprepared environments. *Comput. Vis. Image Underst.* **99**(2), 259–290 (2005)
34. Lu, C.P., Hager, G.D., Mjolsness, E.: Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(6), 610–622 (2000)
35. Mora, T.: *Solving polynomial equation systems II: Macaulay’s paradigm and Gröbner technology*, vol. 2. Cambridge University Press (2003)
36. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics* **31**(5), 1147–1163 (2015)
37. Nakano, G.: Globally optimal DLS method for PnP problem with Cayley parameterization. In: *British Machine Vision Conference*. pp. 78.1–78.11 (2015)
38. Nocedal, J., Wright, S.J.: *Sequential Quadratic Programming*, pp. 529–562. Springer New York (2006)
39. Nousias, S., Lourakis, M., Bergeles, C.: Large-scale, metric structure from motion for unordered light fields. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 3292–3301 (2019)
40. Ohayon, S., Rivlin, E.: Robust 3D head tracking using camera pose estimation. In: *International Conference on Pattern Recognition (ICPR)*. vol. 1, pp. 1063–1066. IEEE (2006)

41. Pardalos, P.M., Vavasis, S.A.: Quadratic programming with one negative eigenvalue is NP-hard. *Journal of Global Optimization* **1**(1), 15–22 (1991)
42. Romea, A.C., Torres, M.M., Srinivasa, S.: The MOPED framework: Object recognition and pose estimation for manipulation. *International Journal of Robotics Research* **30**(10), 1284–1306 (Sep 2011)
43. Rosten, E., Reitmayr, G., Drummond, T.: Improved RANSAC performance using simple, iterative minimal-set solvers. *CoRR* **abs/1007.1432** (2010)
44. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 4104–4113 (2016)
45. Schweighofer, G., Pinz, A.: Globally optimal  $O(n)$  solution to the PnP problem for general camera models. In: *British Machine Vision Conference*. pp. 1–10 (2008)
46. Shuster, M.: A survey of attitude representations. *Journal of the Astronautical Sciences* **41**(4), 439–517 (1993)
47. Terzakis, G., Lourakis, M., Ait-Boudaoud, D.: Modified Rodrigues parameters: an efficient representation of orientation in 3D vision and graphics. *Journal of Mathematical Imaging and Vision* **60**(3), 422–442 (2018)
48. Urban, S., Leitloff, J., Hinz, S.: MLPnP - a real-time maximum likelihood solution to the perspective-n-point problem. In: *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*. vol. 3, pp. 131–138 (2016)
49. Wang, P., Xu, G., Cheng, Y., Yu, Q.: A simple, robust and fast method for the perspective-n-point problem. *Pattern Recognition Letters* **108**, 31–37 (2018)
50. Zheng, E., Wu, C.: Structure from motion using structure-less resection. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 2075–2083 (2015)
51. Zheng, Y., Kuang, Y., Sugimoto, S., Åström, K., Okutomi, M.: Revisiting the PnP problem: A fast, general and optimal solution. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 2344–2351 (2013)
52. Zheng, Y., Sugimoto, S., Okutomi, M.: ASPnP: An accurate and scalable solution to the perspective-n-point problem. *IEICE Transactions on Information and Systems* **96**(7), 1525–1535 (2013)