Reliable Camera-Based Positioning Robust to Initial Pose Error

Antonino Triolo, Chen Zhu, Michael Meurer

Institute of Communications and Navigation,
German Aerospace Center (DLR), Oberpfaffenhofen, Germany
Email: {Antonino.Triolo, Chen.Zhu, Michael.Meurer}@dlr.de

BIOGRAPHY

Antonino Triolo is a researcher at the German Aerospace Center (DLR, Institute of Communication and Navigation). He graduated in Computer Engineering in 2020 and in Automation and Control Engineering in 2022 from the University of Catania (Italy, Sicily), and did a short research period there. In the same year, after finishing his studies, he started his work at DLR. His research concerns the applicability of the concept of integrity in the case of camera-based navigation in the context of Unmanned Aerial Vehicles (UAVs).

Chen Zhu Dr. Chen Zhu is a senior research fellow and the head of Visual and Terrestrial Augmentation group at the Institute of Communications and Navigation, German Aerospace Center (DLR). He received his Ph.D. degree (Dr.-Ing.) and Master's degree (M.Sc.) from Technical University of Munich, Germany and his Bachelor's degree from Tsinghua University, Beijing, China. He is interested in the research fields of visual navigation, multi-sensor fusion, and robotic swarm navigation, currently focusing on the system integrity.

Michael Meurer is the director of the Department of Navigation at the German Aerospace Center (DLR) and of the Center of Excellence for Satellite Navigation. In addition, he is a full professor of electrical engineering and director of the Chair of Navigation at the RWTH Aachen University. His current research interests include GNSS signals, GNSS receivers and navigation for safety-critical applications.

ABSTRACT

In GNSS degraded environments, cameras have great potential to be used as navigation sensors in various applications such as UAV landing, autonomous driving, indoor navigation, etc. Using a set of visible features with known locations on the map, the 6 degrees of freedom (DOF) pose (position and attitude) of the camera can be estimated. This is known as the Perspective-n-Point (PnP) problem. However, the performance of pose estimation is very sensitive to the initial rough estimate of the camera pose, as the measurement equation is highly non-linear. This problem degrades the performance of visual navigation and has limited its use in applications that require high reliability. In this work, we propose an innovative algorithm based on Feasible Pursuit Point and Successive Convex Approximation (FPP-SCA) method by Mehanna et al. (2015) to solve the PnP problem. The algorithm exhibits a significant global convergence property so that the camera pose can be accurately estimated even when the initial position error is large. The overall performance of the proposed method is shown to outperform state-of-the-art approaches and to be more resistant to incorrect initial conditions and measurement noise levels from simulations and experiments in the context of UAV landing.

I. INTRODUCTION

In recent years there has been an increase in the use of cameras for navigation in a variety of systems ranging from automotive to robotics and aerospace. Each context has different requirements on the navigation system. For example, in augmented reality and many robotics applications, a fast and accurate camera-based localization algorithm is usually preferred. The real-time feasibility and the continuity of the system have a higher priority than the system integrity in the algorithm design. However, in safety-critical applications such as urban air mobility (UAM), which targets on integrating cargo delivery drones and future air-taxis into the urban airspace, the navigation system also needs to provide high integrity solutions. In urban areas, the performance of navigation solutions based on Global Navigation Satellite System (GNSS) may degrade or even become unavailable due to the blockage of the satellite signals, radio interference and multipath effects. In such scenarios, computer vision based navigation methods can play an important role in the landing of UAVs (Unmanned Aerial Vehicles) at vertiports, and the integrity of the camera based position estimation is crucial to the operation safety.

Exploiting georeferenced landmarks, e.g., reference markers or distinct features in the map, is an intuitive and reliable visual positioning method for UAV landing. Using a set of 2D visible features with known 3D locations, the six degrees of freedom (DOF) pose (position and attitude) of the camera can be estimated in the reference frame of the 3D points with a single image.

Figure 1: *Urban Air Mobility Concept. ©DLR.*

This is known as the Perspective-n-Point (PnP) problem.

A few state-of-the-art algorithms can solve the PnP problem analytically, e.g., P3P in Gao et al. (2003), AP3P in Ke and Roumeliotis (2017), EPNP in Lepetit et al. (2009), IPPE in Collins and Bartoli (2014), and SQPNP in Terzakis and Lourakis (2020). They have already been implemented in the prestigious open source library OpenCV (Bradski (2000)) and widely used in various visual navigation applications. However, most of the analytical PnP solvers are based on the noise-free assumption, but in practice the measurements are always noisy. Nonlinear optimisation is required to solve the PnP problem with noisy measurements. Gradient or Jacobian-based methods such as Gauss-Newton or the Levenberg-Marquardt algorithms are widely used to estimate camera poses iteratively. Nevertheless, the performance of the nonlinear optimization is significantly dependent on the initial pose estimation, since the measurement equations for solving the PnP problem are highly nonlinear. If the error in the initial guess is large, there is a high possibility that the optimization algorithm converges to a local minimum of the cost function, which can result in large errors in the estimated pose. It has been shown that with exactly the same measurements and associations of 2D-3D points, the estimation results may differ for distinct initial guesses of the camera pose (Zhu et al. (2022)). It is in many cases challenging to validate whether the solver has converged to the global optimum or not. This introduces integrity risks to the navigation system and strongly limits their use in safety-critical applications.

Researchers have proposed advanced methods to mitigate the impact of the initial pose errors. Rosen et al. (2014) and Jia et al. (2023) proposed methods applying the trust region algorithm from Conn et al. (2000) to solve the nonlinear optimization. Semi-Definite Programming (SDP) is another tool to cope with the global convergence problem of PnP (Schweighofer and Pinz (2008)) and has been used in visual navigation for UAVs in Jubran et al. (2022).

In this work we proposed SEC-PnP (Slack-Eigen-Convexification-PnP) algorithm to estimate the camera pose reliably, even if the initial pose estimation is erroneous and the measurements are noisy. SEC-PnP models the PnP problem as a Quadratic Constrained Quadratic Problem (QCQP), and convexify the problem using slack variables and linearizing the quadratic constraints based on the FPP-SCA algorithm. The use of slack variables relaxing the constraints during the iterative process gives the proposed algorithm greater flexibility, and the possibility of ”avoiding” local minima. The proposed approach has greater robustness and better convergence properties than the state-of-the-art algorithms, so it is shown to outperform other methods in pose estimation error using both simulation and real experiment data from UAV, at a cost of slightly increased computational time. The paper is organized as follows: in Section II, the system model and reference frame definitions are introduced, followed by a categorization of the state-of-the-art methods to solve the problem. In Section III, the proposed SEC-PnP algorithm is introduced in detail. Validations of the algorithm are shown in Section IV, using both simulated scenarios and real measurement data collected from UAV flight tests. Conclusions are drawn in Section V.

II. SYSTEM MODEL

In this section, the basics of camera-based positioning and the system models are briefly introduced. More detailed introduction of the theory behind visual navigation can be found in Hartley and Zisserman (2004), and the error sources and uncertainty in the visual positioning process are reviewed in Zhu et al. (2022).

The raw sensor measurement of digital cameras is a discrete image $\boldsymbol{I}$ which represents an amplitude measure of the illuminance during the exposure time on the image plane $\Omega \subset \mathbb{R}^2$. For visual navigation applications, it is essential to extract geometric information from the luminance of the images, and the processing should be in real time. For navigation purposes, gray-scale images are typically used because of computational reasons. It is important to underline that the intensity values in the image $\boldsymbol{I}$ can be noisy due to various error sources.

The location of pixels in the image plane and the corresponding light sources follows projective geometry. Most systems can be approximated using the classic pinhole model. For this model, the two-dimensional (2D) position of an image point $\boldsymbol{P}_i^{2D} = [x_i^{2D}, y_i^{2D}]^\top \in \Omega \subset \mathbb{R}^2$ can be related to the corresponding 3D point $\boldsymbol{P}_i^C = [x_i^C, y_i^C, z_i^C]^\top$ as :

$$\boldsymbol{P}_i^{2D} = \left[f_x \frac{x_i^C}{z_i^C} + c_x, f_y \frac{y_i^C}{z_i^C} + c_y \right]^\top \tag{1}$$

with $f_{x/y}$ denoting the focal length in x and y direction, (c_x, c_y) denotes the coordinate of the projected principal point in the 2D image. The transformation between the camera frame C and the world reference frame O is dependent on the position and attitude of the camera. For a camera at position $\boldsymbol{t}_O \in \mathbb{R}^3$ and with attitude represented by $\boldsymbol{R}_O \in SO(3)$ in the world frame, the projective geometry of a 3D point in the world frame is described by:

$$\widetilde{\boldsymbol{P}}_i^{2D} = d_i \begin{bmatrix} \boldsymbol{P}_i^{2D} \\ 1 \end{bmatrix} = \boldsymbol{K}\, \boldsymbol{R}_O [\boldsymbol{I}| - \boldsymbol{t}_O] \begin{bmatrix} \boldsymbol{P}_i^O \\ 1 \end{bmatrix} = \boldsymbol{P}_{cam} \widetilde{\boldsymbol{P}}_i^O \tag{2}$$

where $\widetilde{\boldsymbol{P}}_i^O$ is expressed in homogeneous coordinates in the world frame, and $\boldsymbol{K}$ is the intrinsic matrix. A simple example with four points in the space projected to the image plane is shown in Figure 2. By rewriting the last equation with variables in Euclidean space, we denote the 2D coordinates of a point in the image by a function of the camera pose and 3D position of the point as:

$$\boldsymbol{P}_i^{2D} = \pi(\widetilde{\boldsymbol{P}}_i^O, \boldsymbol{x}) \tag{3}$$

where $\boldsymbol{x} \in \mathbb{R}^6$ is the camera pose parameterized with a six degrees-of-freedom vector. The camera pose $\boldsymbol{x}$ can be estimated given a set of associated 3D points and their corresponding 2D projections in the image. The coordinates of the 3D points can be obtained, for example, by using georeferenced landmarks (this is the approach applied in our simulations and real tests). By stacking all the N successfully associated 2D-3D points, the pose of the camera can be estimated by solving the following nonlinear optimization problem iteratively:

$$\boldsymbol{x}_{est} = arg\, min_x \left\| \boldsymbol{\mu} - \pi(\boldsymbol{x}) \right\|^2 \tag{4}$$

with $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_N]^\top$ and $\boldsymbol{\mu}_i = \boldsymbol{P}_i^{2D} + \boldsymbol{n}$, being $\boldsymbol{\mu}$ the vector composed by the stacked 2D measurements and $\boldsymbol{\mu}_i$ the single 2D measurement composed by the sum of the correct value given by the camera model and a noise vector $\boldsymbol{n}$ derived by multiple noise sources and disturbances, respectively. Equation (4) is the fundamental optimization problem for visual pose estimation. It is easy to verify that the measurement function $\pi(\boldsymbol{x})$ is highly nonlinear. Therefore, a good initial estimate of $\boldsymbol{x}$ is essential to ensure that the iterative process does not lead to local optima, if state-of-the-art methods like Levenberg-Marquardt (Eade (2013)) are applied.

In has been shown in Zhu et al. (2022) that distinct initial guesses of the camera pose can result in different optimization results. If the initial guess of the nonlinear optimization is close enough to the true position, the process can generate the correct estimation results. However, if the initialization is not sufficiently accurate the estimated camera pose is biased from the true value, even if all the measurements are noise-free and all the associations are correct. The error is caused by the convergence issue of nonlinear optimization.

The state-of-the-art approaches to solve the PnP problem can be categorized into the following groups:

1. *Analytical PnP algorithms*. In noise-free cases, $\boldsymbol{x}$ can be solved analytically by using $N \geq 3$ measurements. For the real cases where different sources of noise are present, there are several analytical PnP algorithms, each with different assumptions about the minimal number of features or geometry of 2D-3D associations. The algorithms compared in this work are the P3P algorithm in Gao et al. (2003), AP3P in Ke and Roumeliotis (2017), EPNP in Lepetit et al. (2009), IPPE in Collins and Bartoli (2014), and SQPNP in Terzakis and Lourakis (2020).
2. *Gradient-based iterative approaches*. Gauss-Newton methods (Madsen et al. (2004)) and Levenberg-Marquardt are among the most classical and popular non-linear optimisation algorithms. As mentioned above, starting from an initial estimate of the parameters and having a well-defined cost function based on the 2D-3D associations, a gradient-based approach is used with the assumption of already being in the domain of the global minimum of the cost function.
3. *Solvers for non-linear constrained problems*. As will be shown in Section III, the PnP problem can be modelled as a non-convex Quadratic Constrained Quadratic Problem. It is possible to solve QCQP using state-of-the-art approaches, such as Trust-Region methods in Conn et al. (2000) or the Sequential Least Squares Programming (SLSQP) in Fu et al. (2019). Comparing the SEC-PnP algorithm with the state-of-the-art solvers is necessary to show the performance of the proposed approach under the same the QCQP modelling of the problem.
4. *SDP relaxation*. SDP (Semi-Definite Programming) relaxation in Fu et al. (2013) is the state-of-the-art algorithm to solve QCQP. SDP algorithm transforms the non-convex QCQP problem into one in which only semi-definite positive functions exist, by making appropriate assumptions. This makes the problem convex by definition and therefore easy to solve with global convergence property.

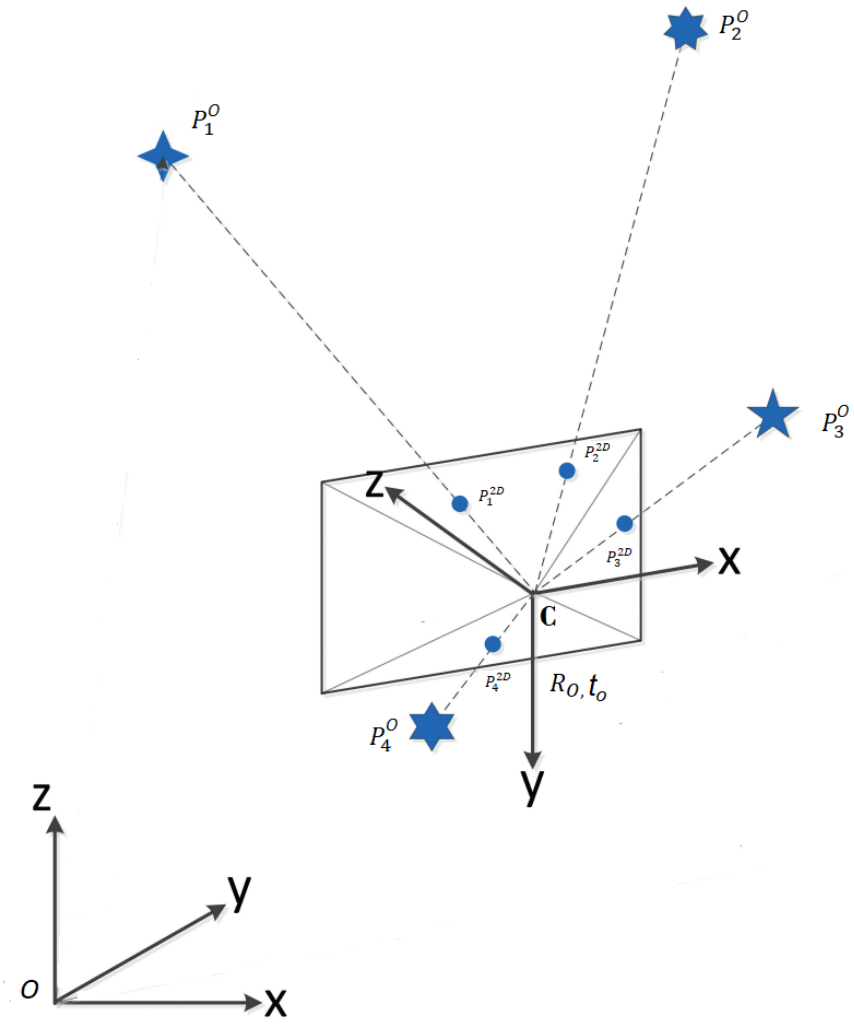


Figure 2: *Reference frames in visual positioning.*

III. THE SEC-PNP ALGORITHM

In this work, we propose an innovative solution to the PnP problem, which aims to improve the robustness to the errors in the initial camera pose. The acronym of the algorithm, SEC-PnP, stands for *Slack-Eigen-Convexification-PnP*, which describes the most important aspects of the processing procedure. Following the problem formulation derived from the work of Sun and Deng (2020), the PnP problem can be formulated into a non-convex QCQP. With the help of *Slack* variables and *Eigen*value decomposition based *Convexification*, the proposed algorithm approaches the non-convex QCQP using a series of convex QCQPs, which can be solved with guaranteed global convergence.

The detailed cost function construction and convexification process is described in the following part of this section.

1. A QCQP Formulation of the PnP Problem

In the mapping process of a calibrated pinhole camera, consider a number of points in the 3D world and their corresponding projections on the 2D image plane of the camera, both with known coordinates. With these 3D-to-2D point correspondences, as already discussed in the last section, the PnP problem consists in estimating the camera's 3D pose w.r.t. the world frame that can be mathematically represented as a rotation matrix $\boldsymbol{R}_O \in SO(3)$ containing the orientation information and a translation vector $\boldsymbol{t}_O \in \mathbb{R}^3$ containing the position information.

Given a point

$$\boldsymbol{P}_i^O = [x_i^O, y_i^O, z_i^O]^\top \tag{5}$$

with $\boldsymbol{P}_i^O \in \mathbb{R}^3$ in the 3D world frame O, the extrinsic absolute pose of the camera $\boldsymbol{R}_O$ and $\boldsymbol{t}_O$, and the intrinsic matrix $\boldsymbol{K}$, point $\boldsymbol{P}_i^O$'s corresponding coordinate in the camera frame C can be written as

$$\boldsymbol{P}_i^C = [x_i^C, y_i^C, z_i^C]^\top = \boldsymbol{K}\,\boldsymbol{R}_O\,[\boldsymbol{I}|-\boldsymbol{t}_O]\,\widetilde{\boldsymbol{P}}_i^O, \tag{6}$$

where $\widetilde{\boldsymbol{P}}_i^O = [\boldsymbol{P}_i^{O\top}, 1]^\top$ is the homogenized expression of the point $\boldsymbol{P}_i^O$, and its 2D coordinate on the image plane can be expressed as

$$\boldsymbol{P}_i^{2D} = [x_i^C/z_i^C, y_i^C/z_i^C, 1]^\top = [x_i^{2D}, y_i^{2D}, 1]^\top. \tag{7}$$

Due to noise, the 2D projections of 3D points cannot be precise, generating projection errors. This error for the point $\boldsymbol{P}_i^O$ can be geometrically quantified as a distance from the theoretical projection $\boldsymbol{P}_i^{2D}$ to the actual one $\boldsymbol{P}_i^{2D*}$ obtained on the real image. We can take the squares of these distances for each n point so that the total projection error caused by noise can be globally formulated as

$$\sum_{i=1}^{n} d^2(\boldsymbol{P}_i^{2D}, \boldsymbol{P}_i^{2D*}) = \sum_{i=1}^{n} (\boldsymbol{P}_i^{2D} - \boldsymbol{P}_i^{2D*})^\top (\boldsymbol{P}_i^{2D} - \boldsymbol{P}_i^{2D*}), \tag{8}$$

where $d(,)$ is the planar distance of two 2D points and $i = 1, 2, ..., n$. And by multiplying the depth z_i^C in the Z-axis, that means to amplify the distance according to the depth, it is possible to have the following expression:

$$\zeta_{points} = \sum_{i=1}^{n} d^2(\boldsymbol{P}_i^C, z_i^C \boldsymbol{P}_i^{2D*}) = \sum_{i=1}^{n} (\boldsymbol{P}_i^C - z_i^C \boldsymbol{P}_i^{2D*})^\top (\boldsymbol{P}_i^C - z_i^C \boldsymbol{P}_i^{2D*}), \tag{9}$$

where ζ_{points} represents now the sum of the geometric distances for all n point correspondences. Equation (6) can be reformulated as

$$\boldsymbol{P}_i^C = \boldsymbol{K}\,\boldsymbol{R}_O\,[\boldsymbol{I}|-\boldsymbol{t}_O]\,\widetilde{\boldsymbol{P}}_i^O = \boldsymbol{K}\,\boldsymbol{M}\,\widetilde{\boldsymbol{P}}_i^O = \boldsymbol{K}\,(\widetilde{\boldsymbol{P}}_i^{O\top} \otimes \boldsymbol{I}_3)\,vec(\boldsymbol{M}), \tag{10}$$

where $\otimes$ is the Kronecker product and $\boldsymbol{I}_3$ is the identity matrix. $vec(\boldsymbol{M})$ is the vectorialization of the matrix $\boldsymbol{M}$, that will contain the parameters to be found during the optimization process. The Z-axis depth z_i^C in (9) can be written as

$$z_i^C = (\widetilde{\boldsymbol{P}}_i^{O\top} \otimes \boldsymbol{I}_3)_{(3)}\,vec(\boldsymbol{M}), \tag{11}$$

where the $(\cdot)_{(3)}$ indicates the third row of the 3×12 matrix taken in consideration. Furthermore, the geometric distances in (9) can be rewritten as a quadratic function such that

$$d^2(\boldsymbol{P}_i^C, z_i^C \boldsymbol{P}_i^{2D*}) = \boldsymbol{x}^\top \boldsymbol{D}_i^\top \boldsymbol{D}_i \boldsymbol{x} = \boldsymbol{x}^\top \boldsymbol{N}_i \boldsymbol{x} \tag{12}$$

where vector $\boldsymbol{x} = vec(\boldsymbol{M}) = [r_{11},\, r_{12},\, r_{13},\, m_{14},\, r_{21},\, r_{22},\, r_{23},\, m_{24},\, r_{31},\, r_{32},\, r_{33},\, m_{34}]^\top$, with $m_{14},\, m_{24},\, m_{34}$ representing the fourth column of the matrix $\boldsymbol{M}$, matrix $\boldsymbol{D}_i = (\boldsymbol{K}(\widetilde{\boldsymbol{P}}_i^{O\top} \otimes \boldsymbol{I}_3) - \boldsymbol{P}_i^{2D*}(\widetilde{\boldsymbol{P}}_i^{O\top} \otimes \boldsymbol{I}_3)_{(3)})$, and $\boldsymbol{N}_i$ is a 12×12 Positive Semi-Definite (PSD) matrix that can be computed knowing 3D-to-2D measurements of point correspondences using the aforementioned formula.
Exploiting the quadratic nature of the formulation of (12), we can superpose all the cumulative geometric distances from the n point matches as

$$\zeta_{points} = \sum_{i=1}^{n} d^2(\boldsymbol{P}_i^C, z_i^C \boldsymbol{P}_i^{2D*}) = \boldsymbol{x}^\top (\sum_{i=1}^{n} \boldsymbol{N}_i)\, \boldsymbol{x} = \boldsymbol{x}^\top \boldsymbol{N}_{tot}\, \boldsymbol{x}\,. \tag{13}$$

This formulation is independent from the point correspondence number n.

So far we have analysed the construction of the cost function. The variables representing the rotation matrix are contained in the vector $\boldsymbol{x}$, but it is now necessary to define the constraints that a correct rotation matrix has to respect. As we mentioned before, $\boldsymbol{R}_O \in SO(3)$, so $\boldsymbol{R_O}$ is a 3×3 orthogonal matrix with determinant 1. These two properties can be briefly summarized in the expression $\boldsymbol{R}_O^\top \boldsymbol{R}_O = \boldsymbol{I}$. We can then write the following quadratic functions representing the same properties

$$\begin{aligned}
&r_{11}^2 + r_{21}^2 + r_{31}^2 = 1,\\
&r_{12}^2 + r_{22}^2 + r_{32}^2 = 1,\\
&r_{13}^2 + r_{23}^2 + r_{33}^2 = 1,\\
&r_{11}^2 r_{12}^2 + r_{21}^2 r_{22}^2 + r_{31}^2 r_{32}^2 = 0,\\
&r_{12}^2 r_{13}^2 + r_{22}^2 r_{23}^2 + r_{31}^2 r_{33}^2 = 0,\\
&r_{11}^2 r_{13}^2 + r_{21}^2 r_{23}^2 + r_{31}^2 r_{33}^2 = 0.
\end{aligned} \tag{14}$$

Each of the above quadratic functions can be briefly expressed in matrix form using the vector of variables $\boldsymbol{x}$, so we can write:

$$\boldsymbol{x}^\top \boldsymbol{P}_i \boldsymbol{x} = c_i, \tag{15}$$

with $\boldsymbol{c} = [1, 1, 1, 0, 0, 0]^\top$ and $i = 1, .., 6$. Finally, taking into account the cost function and the six quadratic constraints derived from the orthonormality of the rotation matrix, we can define the PnP problem as a non-convex Quadratic Constrained Quadratic Problem (QCQP)

$$\begin{aligned}
\min_{x} \quad & \boldsymbol{x}^\top \boldsymbol{N}_{tot} \boldsymbol{x}\\
\text{s.t.} \quad & \boldsymbol{x}^\top \boldsymbol{P}_i \boldsymbol{x} = c_i, \qquad i = 1, ..., 6.
\end{aligned} \tag{16}$$

2. Convexification of the QCQP Formulation

Non-convex QCQPs are NP-hard in general. Existing approaches relax the non-convexity using semi-definite relaxation (SDR) or linearize the non-convex part and solve the resulting convex problem. However, these techniques are seldom successful in even obtaining a feasible solution when the QCQP matrices are indefinite (Mehanna et al. (2015)).

We propose to employ the FPP-SCA iterative algorithm for obtaining the optimal solution to the QCQP formulation in Equation (22), where we approximate the feasible region through a linear restriction of the non-convex parts of the constraints. In order to guarantee feasibility of the modified problem, slack variables are added, and a penalty is used to ensure that slacks are sparingly used. The solution of the resulting optimization problem is then used to compute a new linearization, and the procedure is repeated until convergence.

As a first step, we represent each equality constraint in the original problem as two inequality constraints

$$\begin{aligned} \min_{x} \quad & \boldsymbol{x}^\top \boldsymbol{N}_{tot} \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{x}^\top \boldsymbol{P}_i \boldsymbol{x} \leq c_i, \\ & \boldsymbol{x}^\top \boldsymbol{P}_i \boldsymbol{x} \geq c_i, \qquad i = 1, ..., 6. \end{aligned} \tag{17}$$

In order to have more flexibility during the minimisation process, we introduce slack variables $\boldsymbol{s} \in \mathbb{R}^6$ in the constraints and a slack penalty in the cost function

$$\begin{aligned} \min_{x,\, s} \quad & \boldsymbol{x}^\top \boldsymbol{N}_{tot} \boldsymbol{x} + \lambda \|\boldsymbol{s}\| \\ \text{s.t.} \quad & \boldsymbol{x}^\top \boldsymbol{P}_i \boldsymbol{x} \leq c_i + s_i, \\ & \boldsymbol{x}^\top \boldsymbol{P}_i \boldsymbol{x} \geq c_i - s_i, \qquad i = 1, ..., 6. \end{aligned} \tag{18}$$

where λ trades off the original objective function and the slack penalty term, and $\| \cdot \|$ is the euclidean norm. Problem (18) is always feasible, and if $(\boldsymbol{x}^*, \boldsymbol{s}^*)$ is an optimal solution of (18) and it so happens that $\boldsymbol{s}^* = 0$, then $\boldsymbol{x}^*$ is an optimal solution of Problem (17), and clearly of (16). We must underline that Problem (18) remains non-convex and NP-hard in general.

Using eigenvalue decomposition, the matrix $\boldsymbol{P}_i$ can be expressed as $\boldsymbol{P}_i = \boldsymbol{P}_i^{(+)} + \boldsymbol{P}_i^{(-)}$, where $\boldsymbol{P}_i^{(+)} \succcurlyeq 0$ and $\boldsymbol{P}_i^{(-)} \preccurlyeq 0$. For any $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^{12}$, $(\boldsymbol{x} - \boldsymbol{z})^\top \boldsymbol{P}_i^{(-)} (\boldsymbol{x} - \boldsymbol{z}) \leq 0$. Expanding the last expression, we obtain

$$\boldsymbol{x}^\top \boldsymbol{P}_i^{(-)} \boldsymbol{x} \leq 2 \boldsymbol{z}^\top \boldsymbol{P}_i^{(-)} \boldsymbol{x} - \boldsymbol{z}^\top \boldsymbol{P}_i^{(-)} \boldsymbol{z}. \tag{19}$$

By using the linear restriction (19) around the point $\mathbf{z}$ we may replace the i-th non-convex constraint of Problem (16) with the convex constraints

$$\begin{aligned} & \boldsymbol{x}^\top \boldsymbol{P}_i^{(+)} \boldsymbol{x} + 2 \boldsymbol{z}^\top \boldsymbol{P}_i^{(-)} \boldsymbol{x} \leq c_i + \boldsymbol{z}^\top \boldsymbol{P}_i^{(-)} \boldsymbol{z} + s_i \\ & \boldsymbol{x}^\top \boldsymbol{P}_i^{(-)} \boldsymbol{x} + 2 \boldsymbol{z}^\top \boldsymbol{P}_i^{(+)} \boldsymbol{x} \geq c_i + \boldsymbol{z}^\top \boldsymbol{P}_i^{(+)} \boldsymbol{z} - s_i. \end{aligned} \tag{20}$$

And finally we can present the basic algorithm that is used during the optimization process of SEC-PnP as follows:

1. Set $k = 0$ and generate a initial point $\boldsymbol{z}_0$ by knowing an initial estimate of the rotation $\boldsymbol{R}_0$ and the translation vector $\boldsymbol{t}_0$.
2. Solve the following convex QCQP:

$$\begin{aligned} \min_{x,\, s} \quad & \boldsymbol{x}^\top \boldsymbol{N}_{tot} \boldsymbol{x} + \lambda \|\boldsymbol{s}\| \\ \text{s.t.} \quad & \boldsymbol{x}^\top \boldsymbol{P}_i^{(+)} \boldsymbol{x} + 2 \boldsymbol{z}_k^\top \boldsymbol{P}_i^{(-)} \boldsymbol{x} \leq c_i + \boldsymbol{z}_k^\top \boldsymbol{P}_i^{(-)} \boldsymbol{z}_k + s_i, \\ & \boldsymbol{x}^\top \boldsymbol{P}_i^{(-)} \boldsymbol{x} + 2 \boldsymbol{z}_k^\top \boldsymbol{P}_i^{(+)} \boldsymbol{x} \geq c_i + \boldsymbol{z}_k^\top \boldsymbol{P}_i^{(+)} \boldsymbol{z}_k - s_i, \qquad i = 1, ..., 6. \end{aligned} \tag{21}$$

3. Let $\boldsymbol{x}_k$ denote the optimal x obtained in Problem (17). at the k iteration, and set $\boldsymbol{z}_{k+1} = \boldsymbol{x}_k$.
4. Set $k = k + 1$ and repeat the procedure from point 2. until convergence.

The Problem (21) is convex and can be easily formulated as a second-order cone program (SOCP). The solver used in our tests is CVXOPT (Vandenberghe (2010)) coupled with the CVXPY interface (Diamond and Boyd (2016), Agrawal et al. (2018)). In the original FPP-SCA algorithm it is proposed to use a generic $\lambda \gg 1$ to force the slack variables toward zero, in order to push the iterates towards the feasible region of Problem (16). The main advantage of FPP-SCA over conventional SCA is the ability to find a feasible point with higher probability during the iteration process, giving at the same time more flexibility to the parameters space.

Instead of using a fixed λ value in the optimization (22) as proposed by FPP-SCA, we further optimize the algorithm by using an adaptive function depending on the current iteration number and on the norm of the slack variables at the first iteration. This approach shows more robustness to measurements noise and to initial position and rotation uncertainties while solving the PnP problem. So the SEC-PnP algorithm can be obtained by modifying the aforementioned algorithm as:

1. Set $k = 0$ and generate a initial point $\boldsymbol{z}_0$ by knowing an initial estimate of the rotation $\boldsymbol{R}_0$ and the translation vector $\boldsymbol{t}_0$.
2. Solve the convex QCQP (21) by setting $\lambda = 1$.
3. Save the value of $\|\boldsymbol{s}\|$ of the solved problem as $\|\boldsymbol{s}_0\|$.
4. Solve the following problem:

$$\begin{aligned} \min_{x,\, s} \quad & \boldsymbol{x}^\top \boldsymbol{N}_{tot} \boldsymbol{x} + f(k, \|\boldsymbol{s}_0\|)\, \|\boldsymbol{s}\| \\ \text{s.t.} \quad & \boldsymbol{x}^\top \boldsymbol{P}_i^{(+)} \boldsymbol{x} + 2\boldsymbol{z}_k^\top \boldsymbol{P}_i^{(-)} \boldsymbol{x} \leq c_i + \boldsymbol{z}_k^\top \boldsymbol{P}_i^{(-)} \boldsymbol{z}_k + s_i, \\ & \boldsymbol{x}^\top \boldsymbol{P}_i^{(-)} \boldsymbol{x} + 2\boldsymbol{z}_k^\top \boldsymbol{P}_i^{(+)} \boldsymbol{x} \geq c_i + \boldsymbol{z}_k^\top \boldsymbol{P}_i^{(+)} \boldsymbol{z}_k - s_i, \qquad i = 1, ..., 6. \end{aligned} \tag{22}$$

5. Let $\boldsymbol{x}_k$ denote the optimal x obtained in Problem (22). at the k iteration, and set $\boldsymbol{z}_{k+1} = \boldsymbol{x}_k$.
6. Set $k = k + 1$ and repeat the procedure from point 4. until convergence.

If the FPP-SCA algorithm converges, it converges to a KKT point for the Problem (18), according to Ye and Zhang (2003). If the converged slack variables turn out being all zero, then it is easy to show that the remaining variables satisfy the KKT conditions for the original Problem (16). For more information on convex optimisation theory, the definition of a KKT point and KKT conditions, we recommend reading the book "Convex Optimisation" by Boyd and Vandenberghe (2004).

The adaptive modification in the SEC-PnP algorithm increases the chances of obtaining solutions with slack variables equal to 0 (with tiny numerical effects in implementation) and thus feasible solutions for Problem (16). The initial information about the norm of the slack variables at step 3 is necessary to determine the extent of constraint violation the problem has during the optimisation process. Once such extent of violation is quantified for the scenario, we can appropriately calibrate the coefficient that determines the weight in the cost function of the slack variable minimisation part. Moreover, this coefficient will increase in magnitude in proportion to the number of iterations in order to obtain an accurate solution that fully respects the constraints imposed by the orthonormality of the rotation matrix.

IV. EXPERIMENTAL RESULTS

Our proposed SEC-PnP algorithm are compared to the state-of-the-art methods reviewed in Section II using both simulated data and real measurements. For clearness of the result, only selected methods from each category (P3P, IPPE, Levenberg-Marquart, SLSQP) are shown in the plots. The SDP algorithm is analysed separately due to its unavailability. Although the SDP relaxation formulation is elegant and easy to understand, it has been shown in Mehanna et al. (2015) that the success rate of the SDP relaxation is significantly lower compared to the FPP-SCA algorithm (on which SEC-PnP is based) for generic QCQPs. We have verified such performance behavior for solving PnP problems in our simulations.

Using the same simulation scenario in terms of altitude and noise level in measurements, and using the same software framework (CVXPY modeling language and CVXOPT/ ECOS/ SDPA solvers), the SDP relaxation does indeed provide an infeasible problem. Given the complexity of the problem (12 variables and 12 inequality constraints), the infeasibility of solving the PnP problem using the same cost function and set of constraints as the aforementioned formulation was predictable, given this table from Mehanna et al. (2015):

Table 1: Generic QCQPs solved with SDP relaxation with $n = 20$

M	32	40	48
Rank-1 solution	11.5%	2.9%	0.4%
No feasible sol. after randomization	84.7%	96.4%	99.5%
Feasible sol. after randomization	3.8%	0.7%	0.1%

M is the number of constraints in the generic problem and n is the number of variables. After obtaining a solution $\boldsymbol{X}$ from the SDP relaxation, it is necessary to derive a solution vector $\boldsymbol{x}$ from $\boldsymbol{X}$ if it has $rank(\boldsymbol{X}) = 1$. If this does not happen, it is necessary to use randomization techniques to obtain rank-1 solutions. Even after using randomization, it is not obvious to obtain a feasible solution. The percentages of feasible and rank-1 solutions are given in the Table 1. The results reported are averaged over 1000 Monte-Carlo simulation runs. The entries of the matrices in the QCQPs are randomly and independently generated from a complex Gaussian distribution (with zero-mean and variance 2), then symmetrized.

Our instance of QCQP, which represents a highly nonlinear problem where the constraints represent the orthonormality of the rotation matrix and the cost function represents a geometric distance subject to numerous sources of error, is hard be solved globally without a feasible starting point. The SDP approach could be useful for problems of small size or mild non-linearity.

Another important factor is the conditioning of the problem data: in the generic QCQPs of Table 1, the data are well conditioned because their order of magnitude is comparable between the cost function and the constraints, so a small percentage of problems was solved successfully. In the case of the PnP problem, it is easy to fall into the infeasible region due to the nature of the data. The data conditioning problem and the condition number are well-studied topics in the world of numerical optimization (Bianchi et al. (2012), Zolezzi (2003)).

Both the simulations and the real tests consist of measurements from a camera mounted on a UAV during the take-off and landing phases on a vertiport. The movement of the UAV is mainly vertical. The test scenario is useful for analysing the possible use of camera-based navigation for applications such as air-taxis. Nevertheless, the SEC-PnP algorithm can be used in many other contexts, as it has no prerequisites on the geometry of the 2D-3D associations.

1. From Simulation

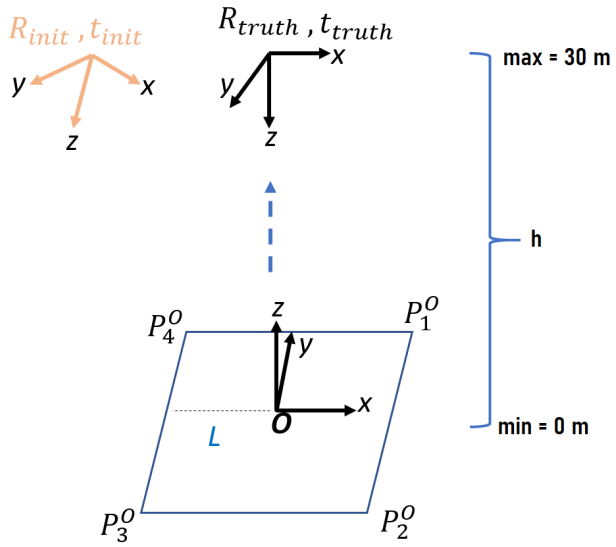


Figure 3: *Simulation setting.*

The simulation scenario is set up as in Figure 3:

- Four 3D points in the world frame O are defined as: $\boldsymbol{P}_1^O = [L, L, 0]$, $\boldsymbol{P}_2^O = [L, -L, 0]$, $\boldsymbol{P}_3^O = [-L, -L, 0]$, $\boldsymbol{P}_4^O = [-L, L, 0]$ with $L = 0.5$. The units for each point coordinate are meters [m]. This arrangement of points represents the shape of a marker of dimension $2L \times 2L$ m with its 4 corners. The world coordinate frame O is centered at the point $\boldsymbol{P}^O = [0, 0, 0]$.
- The camera attitude is represented by the rotation vector $\boldsymbol{r}_{truth} = [-\pi, 0, 0]$ and its translation vector is given by $\boldsymbol{t}_{truth} = [0, 0, h]$ where h represents the z-coordinate varying during the simulation. This means that the camera is simulating a vertical take-off trajectory while pointing down to the marker.
- h will vary in the interval $[0, 30]$ meters. This range will be divided in 60 steps (granularity of 0.5 m), and for each step 300 runs for each solver will be carried out.
- For each run, each i-th coordinate of the translation vector $\boldsymbol{t}_{truth}$ will be perturbed by a normal distribution with $\mu_t = 0$ m and $\sigma_t = 10$ m. The resulting vector will be the initial estimate for the position $\boldsymbol{t}_{init}$. Regarding $\boldsymbol{r}_{init}$, it will be generated by rotating $\boldsymbol{R}_{truth}$ by a certain angle given by a normal distribution with $\mu_r = 0$ rad and $\sigma_r = \frac{\pi}{12}$ rad, with a random rotation axis.
- The $\boldsymbol{P}_i^{2D}$ image points generated by applying the pinhole camera model to the 3D points will be perturbed in both coordinates with noise modelled as a normal distribution with $\mu_n = 0$ pixels and $\sigma_n = 2$ pixels.

- The calibration matrix of the camera is:

$$\boldsymbol{K} = \begin{bmatrix} 1363.58692 & 0 & 948.00583 \\ 0 & 1365.00925 & 609.90681 \\ 0 & 0 & 1 \end{bmatrix}. \tag{23}$$

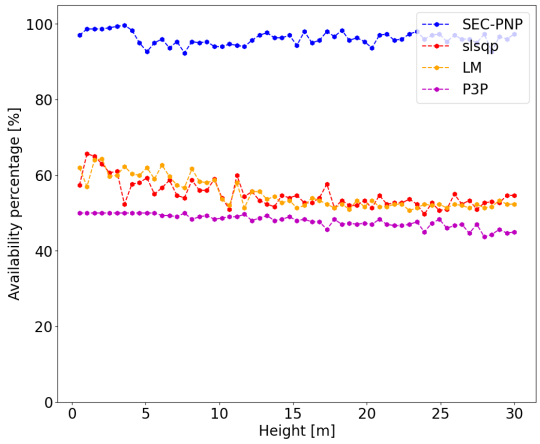
The availability, computational times, rotational error and translation error of the algorithms are compared based on the simulated measurements.

Figure 4a illustrates the percentage of availability of the algorithms out of the 300 completed runs is calculated at each height level. The SEC-PnP algorithm is available for 95 % of the runs regardless of height. It is important to emphasise that an algorithm is also considered unavailable when it provides negative z-coordinate in the translation vector as a result, and consequently a result inconsistent with the geometry of the simulated scenario. This is what happens very frequently in state-of-the-art algorithms; this phenomenon does not occur in SEC-PnP.

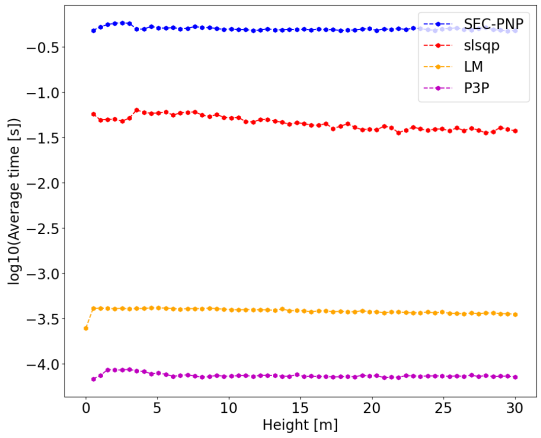
Figure 4b shows the runtime required to estimate the camera pose on a logarithmic scale (in second). SEC-PnP takes approximately half a second. The computational time using SEC-PnP is higher compared to the state-of-the-art algorithms. However, this is still within acceptable limits to potentially support real-time processing. The accuracy of the results of the SEC-PnP algorithm comes at a cost in time that may not be comparable to other geometric methods that are faster but less accurate. Moreover, it is important to underline that the code used has not yet been optimised from a computational time point of view, and this can be further improved in the future, while the state-of-the-art methods are all well-implemented in prestigious open source libraries, e.g. OpenCV, and they require computation times ranging from 1 millisecond to 0.1 millisecond.

Regarding the attitude estimation (Figure 4c), the SEC-PnP algorithm is more accurate than other solvers. At a height of 30 meters, the average error is 0.2 radians. The other algorithms far exceed this value. Most of them settle around the value of 0.5 radians.

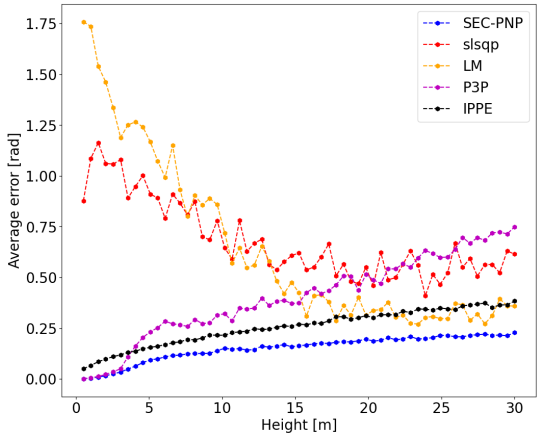
With regard to the estimation of the translation vector (Figure 4d), it can also be seen here that the SEC-PnP algorithm is the most accurate (roughly twice as accurate). On the other hand, with regard to the Levenberg-Marquardt algorithm, it can be observed that this is highly unstable in estimating the position.



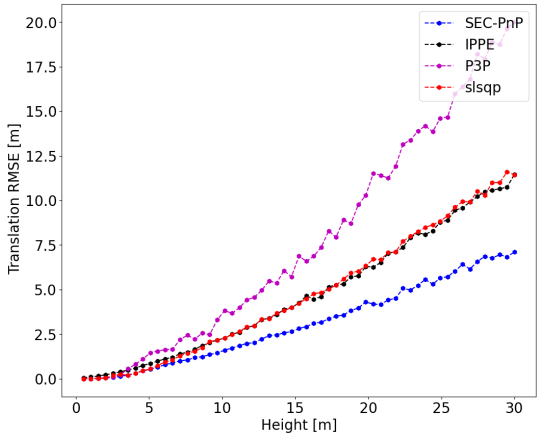
(a) *Availability of the algorithms.*



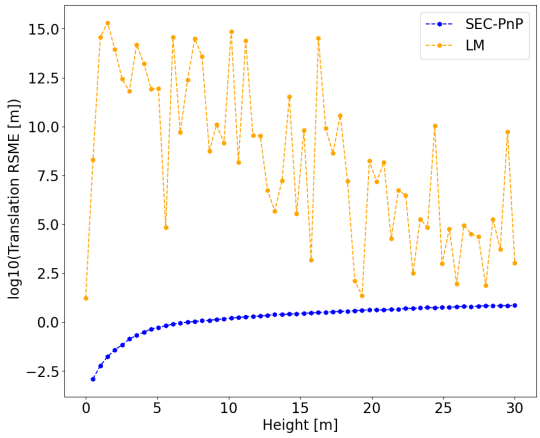
(b) *Computational time of the algorithms.*



(c) *Average rotational error of the algorithms.*



(d) *Translation RMSE of the algorithms.*



(e) *Translation RMSE in log scale of the algorithms.*

Figure 4: Simulation results.

2. From Real-Data

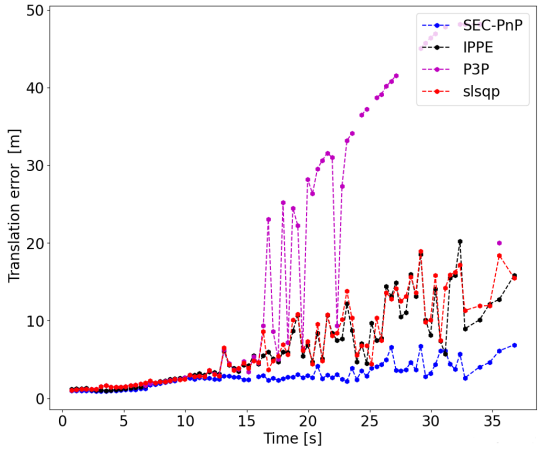
Plots will be generated representing translation error, using the ground-truth given by a local GNSS-system as reference during take-off and landing phases of a drone. This test was carried out in Cochstedt (Germany) in November 2022 for a measurement campaign of DLR. The setup is the following:

- Four 3D points in the world frame O are geo-referenced by using the local GNSS-system: $\boldsymbol{P}_0^O = [L, -L, 0]$, $\boldsymbol{P}_1^O = [-L, -L, 0]$, $\boldsymbol{P}_2^O = [-L, L, 0]$, $\boldsymbol{P}_3^O = [L, L, 0]$ with $L = 0.3925$. The units for each point coordinate are meters [m]. Those points are the four corners of an April-Tag detected by using the April-Tag detector (Olson (2011)). This means that the 2D points are subject to various types of errors, such as association error, camera thermal error, errors due to varying lighting conditions, etc. (more details about these error sources in Zhu et al. (2022)).
  These sources of error will affect the 2D coordinates obtained during the association process, but unlike simulations, the error projected onto the pixels cannot be quantified analytically. The world coordinate frame O is centered at the point $\boldsymbol{P}^O = [0, 0, 0]$. Refer to Figure 6a.
- Information on the camera translation vector $\boldsymbol{t}_{truth}$ will be given by the local GNSS-system. Two different phases of the flight will be analysed: take-off and landing. The z-coordinate of $\boldsymbol{t}_{truth}$ will vary in the $[0, 50]$ meter range. Even in this case the camera will be pointing down to the April-Tag. Unfortunately, no ground-truth data are available for the attitude of the camera, and for this reason no comparison will be carried out to quantify the rotational error of each algorithm. Refer to Figure 6b.
- Since this is a real flight test, the trajectory of the drone will be heavily influenced by abrupt movements due to weather factors such as wind. These sudden and abrupt movements can lead to large errors in the previously mentioned marker detection phase. We will see later that the SEC-PnP algorithm will perform better than the state-of-the-art ones despite these external disturbances.
- Considering both landing and take-off phases, around 200 camera poses have to be estimated. For every estimation process, each i-th coordinate of the translation vector $\boldsymbol{t}_{truth}$ will be perturbed by a normal distribution with $\mu_t = 0$ m and $\sigma_t = 10$ m. The resulting vector will be the initial estimate for the position $\boldsymbol{t}_{init}$. Regarding $\boldsymbol{r}_{init}$, due to the missing of attitude data, we will simply make the assumption that the camera is constantly pointing down, so $\boldsymbol{r}_{init} = [-\pi, 0, 0]$.
- The calibration matrix of the camera is:

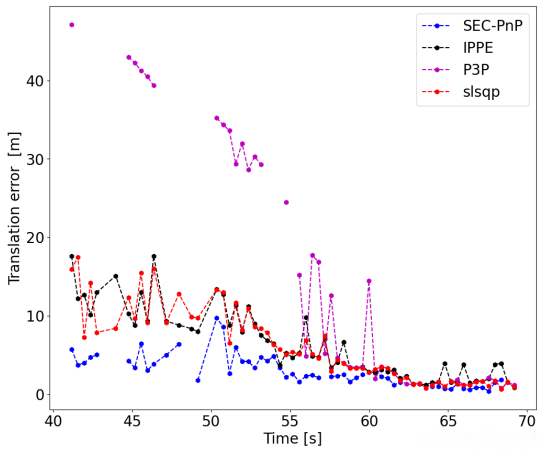
$$\boldsymbol{K} = \begin{bmatrix} 1328.82588 & 0 & 955.46177 \\ 0 & 1335.60937 & 605.3018 \\ 0 & 0 & 1 \end{bmatrix}. \tag{24}$$

  It is important to note that the calibration matrix was obtained through a calibration process prior to the flight tests. This was done using some tools from ROS (Robot Operating System, Quigley et al. (2019)) as well as some custom software from DLR's Visual and Terrestrial Augmentation Group. This means that the $\boldsymbol{K}$ matrix is not free from different types of errors resulting from the calibration, which will inevitably affect the optimization process performed by the SEC-PnP algorithm.

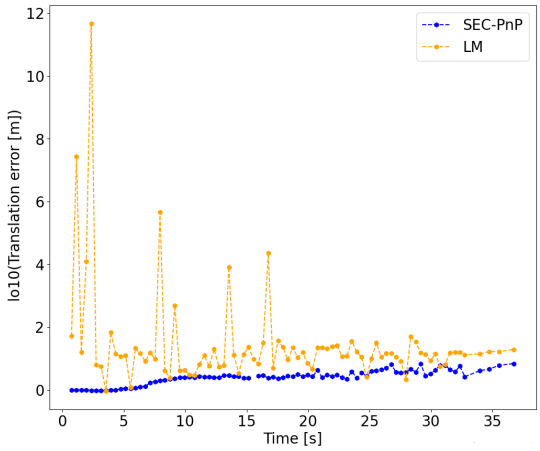
The performance of the different algorithms in the real UAV landing scenario is compared in Figure 5. It can be seen from the plot that SEC-PnP outperforms other algorithms in accuracy and robustness. This statement is valid for both the take-off (Figure 5a) and the landing (Figure 5b) procedure. It is also important to note that the P3P algorithm actually begins to be very inaccurate after a height of approximately 30 m (it is very imprecise after 15 seconds during the take-off phase and becomes reliable again after 60 seconds during the landing phase) and that the Levenberg-Marquardt approach alternates between very accurate and completely incorrect results. This highly unstable behaviour justifies the results obtained for the latter algorithm during simulations, where high levels of RMSE are present.



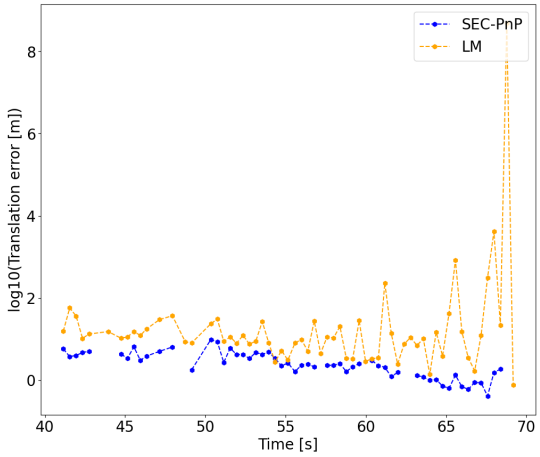
(a) *Translation error during take-off phase [m].*



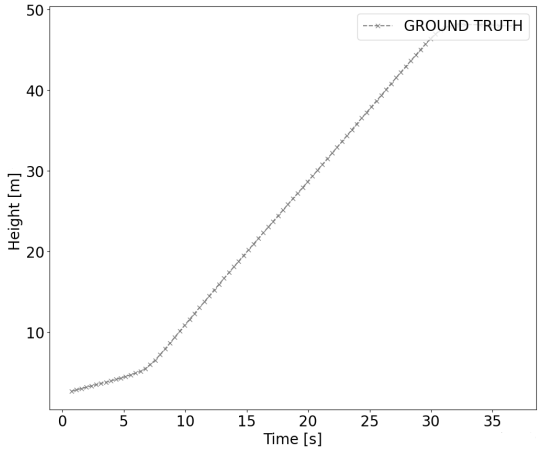
(b) *Translation error during landing phase [m].*



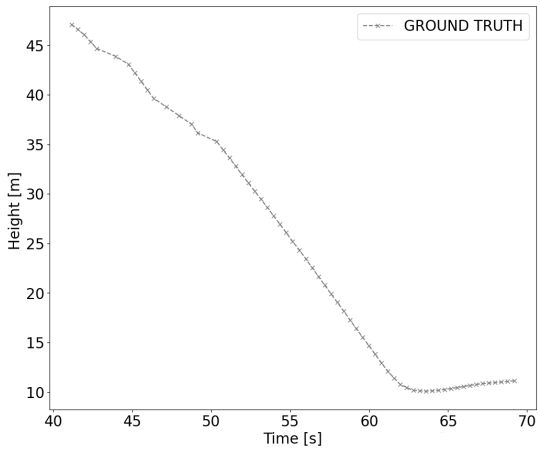
(c) *Translation error during take-off phase in log scale.*



(d) *Translation error during landing phase in log scale.*



(e) *Height during take-off phase.*

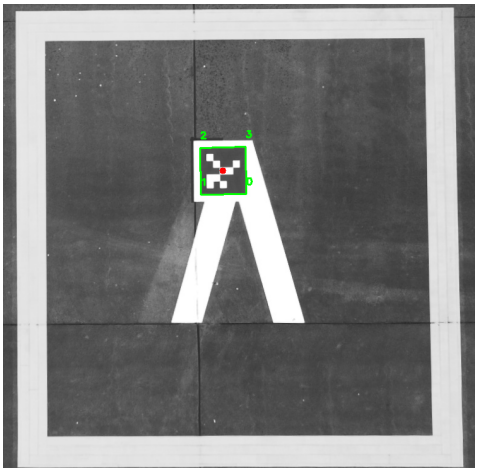


(f) *Height during landing phase.*

Figure 5: Real flights results.

V. CONCLUSION AND OUTLOOK

In this work, an innovative visual positioning algorithm, SEC-PnP, is proposed to improve the global convergence property in pose estimation. When the initial pose estimation is inaccurate, the SEC-PnP algorithm outperforms the state-of-the-art methods in accuracy due to its remarkable global convergence capability. This algorithm has great potential to be exploited in applications with high reliability requirements, such as urban air mobility, autonomous driving, etc. The next step will be to define protection levels for SEC-PnP in order to have integrity standards that other navigation technologies possess and that camera-based navigation does not yet have.
Furthermore, the algorithm has the great potential that an analytical way to validate the global optimality of results obtained through SEC-PnP can be developed in the future work. This concept will be of fundamental importance in securing the integrity of output, so that a certifiable visual navigation solution can be developed.



(a) *April-Tag used during flight tests.*

(b) *Flight tests in Cochstedt (Nov. 2022).*

Figure 6: Images from the real experimental setup.

ACKNOWLEDGEMENTS

I would like to thank my colleague Young-Hee Lee from the *Institute of Communications and Navigation, German Aerospace Center (DLR)*, (Oberpfaffenhofen, Germany) for providing me with valuable flight data from the tests made in November 2022 in Cochstedt, Germany.

REFERENCES

Agrawal, A., Verschueren, R., Diamond, S., and Boyd, S. (2018). A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60.

Bianchi, M., Kassay, G., and Pini, R. (2012). *Nonlinear Analysis*.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambdridge.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Collins, T. and Bartoli, A. (2014). Infinitesimal plane-based pose estimation. *International Journal of Computer Vision*, 109:252–286.

Conn, A. R., Gould, N. I. M., and Toint, P. L. (2000). *Trust Region Methods*. SIAM, Philadelphia.

Diamond, S. and Boyd, S. (2016). Cvxpy: A python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5.

Eade, E. (2013). Gauss-newton /levenberg-marquardt optimization.

Fu, Z., Liu, G., and Guo, L. (2013). Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3):20–34.

Fu, Z., Liu, G., and Guo, L. (2019). Sequential quadratic programming method for nonlinear least squares estimation and its application. *Mathematical Problems in Engineering*, 2019.

Gao, X.-S., Hou, X.-R., Tang, J., and Cheng, H.-F. (2003). Complete solution classification for the perspective-three-point problem. *IEEE Transactions on pattern analysis and Machine Intelligence*, 25(8).

Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambdridge.

Jia, N., Wei, Z., and Li, B. (2023). Trust region nonlinear optimization algorithm for orientation estimator and visual measurement of inertial magnetic sensor. *Drones*, 7(6).

Jubran, I., Fares, F., Alfassi, Y., Ayoub, F., and Feldman, D. (2022). Newton-pnp: Real-time visual navigation for autonomous toy-drones. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13363–13370.

Ke, T. and Roumeliotis, S. I. (2017). An efficient algebraic solution to the perspective-three-point problem. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81:155–166.

Madsen, K., Nielsen, H. B., and Tingleff, O. (2004). Methods for non-linear least squares problems (2nd ed.).

Mehanna, O., Huang, K., Gopalakrishnan, B., Konar, A., and Sidiropoulos, N. D. (2015). Feasible point pursuit and successive approximation of non-convex qcqps. *IEEE Signal Processing Letters*, 22(7).

Olson, E. (2011). Apriltag: A robust and flexible visual fiducial system.

Quigley, M., Gerkey, B., Conley, K., and Faust, J. (2019). ROS.

Rosen, D. M., Kaess, M., and Leonard, J. J. (2014). Rise: An incremental trust-region method for robust online sparse least-squares estimation. *IEEE Transactions on Robotics*, 30(5):1091–1108.

Schweighofer, G. and Pinz, A. (2008). Globally optimal o(n) solution to the pnp problem for general camera models. In *British Machine Vision Conference*.

Sun, L. and Deng, Z. (2020). Certifiably optimal and robust camera pose estimation from points and lines. *IEEE Access*, 8.

Terzakis, G. and Lourakis, M. (2020). A consistently fast and globally optimal solution to the perspective-n-point problem. *European Conference on Computer Vision*.

Vandenberghe, L. (2010). The cvxopt linear and quadratic cone program solvers.

Ye, Y. and Zhang, S. (2003). New results on quadratic minimization. *SIAM Journal on Optimization*, 14(1).

Zhu, C., Meurer, M., and Günther, C. (2022). Integrity of visual navigation —developments, challenges, and prospects. *Journal of the Institute of Navigation*, 69(2).

Zolezzi, T. (2003). Condition number theorems in optimizations. *SIAM Journal on Optimization*, 14(2):2.