Certifiable solver for real-time N-view triangulation

Mercedes Garcia-Salguero[1] and Javier Gonzalez-Jimenez[1]

***Abstract*—Cutting-edge field robotic systems, such as UAV or autonomous cars, demand fast and optimal solutions for any component at the core of their critical navigational tasks. Among them, we focus on the triangulation of image points from multiple views, which is a cornerstone for more complex tasks such as visual localization and SLAM. In this paper we present a fast and certifiable solver for the N-view triangulation problem that doesn't require any specific optimization software package and can be implemented with any linear algebra library. The proposal relies on a series of linear convexifications which, in the limit, recovers the original problem, allowing us to solve problem instances with $N = 10$ views in 150 microseconds on a standard desktop computer. On real data our solver obtains and certifies the optimal solution in more than 99% of the problem instances. We make the code available at https://github.com/mergarsal.**

***Index Terms*—Mapping, Optimization and Optimal Control, Computational Geometry, Optimality Certification, Convex Relaxation**

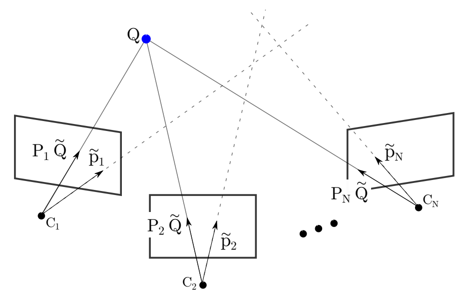


Fig. 1: Scheme of the N-view triangulation problem. Given a set of N projection matrices $\boldsymbol{P}_1, \dots, \boldsymbol{P}_N$, we aim to correct the error-corrupted observations $\hat{\boldsymbol{p}}_1, \dots, \hat{\boldsymbol{p}}_N$ so that they triangulate exactly the (unknown) 3D point $\boldsymbol{Q}$.

I. Introduction

CAMERAS are prevalent sensors in mobile systems, such as autonomous cars, UAVs and robots in general. The information provided by the camera(s) is employed by many visual (detection, recognition, etc) and critical navigational tasks, *i.e.* the localization of the robot and the creation of the environment's map. This map, whose accuracy is crucial for the correct and safe operation of the system, can be estimated together with the localization, task known as Simultaneous Localization and Mapping (SLAM) [1], [2], or can be stored to estimate only the localization later. Maps based on observed features are formed usually by 3D world points, whose coordinates are defined in general by these features and the observers' poses. The estimation of the coordinates is known as *triangulation*, and when the features are obtained from images, the information required to triangulate the point is reduced to the observation on the image and the camera pose (or projection matrix). For the above-mentioned robotic applications based on visual information, the number of views is usually above four, which can be considered as the last "minimal" case [3]. These non-minimal problem instances make the triangulation more robust to noise in the observations but also more complex, as it was shown in previous works [4], [5]. Arguably, the simpler approach is the linear method [3] that leverages a singular value decomposition of the matrix formed by the projection matrices and observations. This method finds the intersection on the rays from the camera centers toward the observations (see Fig. 1), and thus returns the solution only when the observations are noiseless. When the data is corrupted by noise, this intersection doesn't exist and the linear method only obtains suboptimal solutions since the minimized error does not have any geometric meaning. An alternative algorithm is the so-called midpoint method, that returns the point in the middle of these rays. The midpoint method tends to obtain good solutions, specially when each observation is properly weighted [6]. Nevertheless, it is usually considered that the 'optimal' approach for the triangulation problem is to correct the noisy observations so that the 3D point is recovered exactly by, for example, the above-mentioned linear method [3], [4]. This correction is understood as modifying these observations, seeking the 'minimum' change that assures the uniqueness of the 3D point. Since the definition of 'minimum' is not fixed, different solvers based on different norms, *e.g.* $\ell_1, \ell_2, \ell_\infty$ have been proposed in the literature.

In this manuscript we work with the ℓ_2 norm that turns the triangulation problem into a non-convex problem even for two views [4]. This implies the existence of local optima, where iterative algorithms can be trapped without notice. Whereas polynomial solvers, as the one proposed in [4] for the 2-view triangulation, are guaranteed to find the global optimum, they quickly become intractable and/or unstable for more than 2 views, see [5]. Nonetheless, there exist other alternatives to these polynomial solvers that are also able to obtain and certify optimal solutions. In particular, convex relaxations of the original non-convex problem seem to perform well in most problem instances and can be solved by off-the-shelf tools such as SeDuMi [7] or SDPT3 [8] in polynomial time, and have

This work has been supported by the grant program FPU18/01526 and the research project ARPEGGIO (PID2020-117057), both funded by the Spanish Government and the research project HOUNDBOT (P20-01302) funded by the Regional Government of Andalusia with support from the ERDF (European regional development funds).

[1] Mercedes Garcia-Salguero (corresponding author) and Javier Gonzalez-Jimenez are with the Machine Perception and Intelligent Robotics (MAPIR) Group, Malaga Institute for Mechatronics Engineering and Cyber-Physical Systems (IMECH.UMA). University of Malaga, Spain mercedesgarsal@uma.es, javiergonzalez@uma.es

been recently leveraged for other computer vision problems, *e.g.* [9]–[13]. This was the approach followed by Aholt *et al.* in [14] for the N-view triangulation, in which the authors stated the problem in terms of the bilinear constraints that fully identify the space for non-coplanar cameras [15]. However, since the number of constraints is $\binom{N}{2}$, and problem instances with more than 10 views are not rare in real applications (see Section IV), the problem soon acquires dimensions that cannot be handled by current off-the-shelf tools and the proposal becomes too slow or runs out of memory, precluding its usage in real-time applications and/or systems with limited resources. Cifuentes [16] proposed another approach that didn't rely on the bilinear constraints but still can be solved by the above-mentioned tools. The proposal performs well even for coplanar cameras although it is slower than [14]. Whereas recent efforts have been directed to develop certifiable solvers that can handle large-scale problems, they all assume some conditions regarding the constraints of the problems. One of these conditions implies the Jacobian of the constraints being full-rank, the so-called Linear Independence Constraint Qualification (LICQ) [17, Def. 12.4], which doesn't hold for the N-view triangulation problem, thus making those solvers unsuitable. Further, most iterative algorithms that aim to solve constrained problems also assume this condition, *e.g.* IPOPT [18] and SNOPT [19], and in general Sequential Quadratic Programming (SQP) [17, Assumption 18.1]. Indeed, previous works have proposed modifications for these general algorithms to handle lack of LICQ, *e.g.* [20], [21] [1]

Contribution: In this work we contribute a novel solver for the N-view triangulation problem and a fast optimality certifier that says whether the solution is the optimum or it is inconclusive, both being faster than existing realizations [14]. Our solver corrects the observations under the ℓ_2 norm so that they are originated from a unique (unknown) 3D point, condition that for non-coplanar configurations is enforced by $\binom{N}{2}$ bilinear constraints [15]. Our proposal consists of relaxing the original, non-convex problem into linear problems that can be solved by any algebraic library, such as Eigen [22], through a rank revealing decomposition, *e.g.* Complete Orthogonal Decomposition (COD) or Singular Value Decomposition (SVD), and it is able to solve problem instances with $N = 10$ views in less than 150 microseconds and even problems with 500 views (1000 variables and more than 127000 constraints). On real data, we obtain the same results that the other ℓ_2 optimal solver in [14] in a fraction of time. These two features make our proposal suitable for real-time applications with limited resources.

II. Related work

In this Section we mainly focus on the proposals for the N-view triangulation problem with more than four views. The 3D world point that originates the N observations in N different views can be recovered with a set of linear equations if the observations are noiseless. Since error is always present, this linear method only provides a suboptimal solution [3]. As an alternative, we may select the 3D point in the middle of the rays, method which is known as the *midpoint method* [3]. This approach, that can be also extended from two to N views, may provide good results specially when a weight for each observation is introduced [6]. The linear and midpoint methods have in common that they don't modify the observations, but try to explain the data through the 'best' 3D point. Another way to tackle the problem is thus to modify the observations so that they give rise to a single 3D world point. This last approach is the so-called 'optimal' method in the literature [3]. The term 'optimal' here shouldn't be understood as the global optimum is always achieved by these solvers since in general the triangulation problem is non-convex with multiple local minima [4]. Whereas previous works have tackled the minimal problems with less than five views under different norms, *e.g.* [4], [23], the literature for the non-minimal configurations is limited, see [14], [24], [25] and [26] for a recent review.

In this work we seek the minimum correction under the ℓ_2 norm, thus making the triangulation problem non-convex in general. This implies that iterative algorithms cannot guarantee that the returned solution is the global optimum [25], which may affect negatively the applications that leverage them. Aholt *et al.* in [14] propose to solve the N-view triangulation problem by describing the domain only through bilinear constraints [27] and relaxing the problem as a semidefinte program (SDP) that can be solved with guarantees. Cifuentes in [16] proposed an alternative method and formulation for the N-view triangulation problem that overcomes some of the drawbacks in [14], although it's slower due to the employed formulation. Since the number of constraints increases with the number of views, the relaxations [14], [16] soon become too computational expensive in terms of memory and time to be used or even solved given the current limits of off-the-shelf tools. While recent efforts have been put into developing faster tools for these convex relaxations, they usually assume a set of conditions that are not met for the problem at hand.

Large-scale problems aren't new in the literature of convex relaxations, being more predominant in the robotic field, *e.g.* [28], [29]. These previous works try to overcome the dimensionality problem by leveraging alternative certifiable algorithms. Among these alternatives, we focus on optimality certifiers based on Lagrangian duality, which certify the optimality of the solution but do not obtain it and have been shown to perform well in practice while being computational efficient, see *e.g.* [28] (2D SLAM), [29] (3D SLAM) or [30] (rotation averaging).

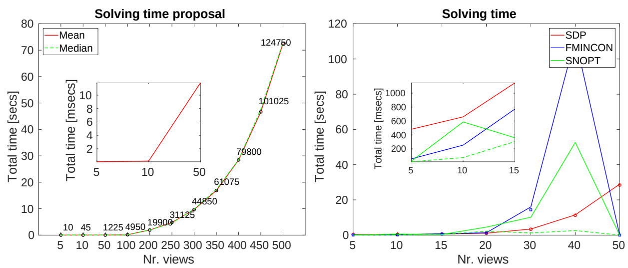
III. Problem formulation and resolution

In this work we leverage the theory about the "multiview ideal" that can be found in *e.g.* [15], [27]. In order to keep the manuscript short and accessible, we don't include any of these theoretical aspects and we refer the interested reader to those references for further details.

A. Problem formulation

The N-view triangulation problem aims to obtain the 3D coordinates of the point $\boldsymbol{Q}$ that generates a set of N observations $\boldsymbol{p}_{\mathrm{i}} \in \mathbb{R}^2$ according to their projection matrices

[1] Since the problem at hand only has equality constraints, LICQ is equivalent to Mangasarian-Fromovitz Constraint Qualification (MFCQ) [17, Def. 12.6].



(a) (b)

Fig. 2: Mean and median computational time (Y-axis) in seconds required to solve a single problem instance with N views (X-axis) and noise $\sigma = 3.0$ pix (a) with our solver (number of constraints M are shown for each N); (b) with the SDP in [14] with CVX [31] and SDPT3 [8], using the function "fmincon" and the commercial software SNOPT [19] through their MATLAB interface. Mean and median are practically the same for all the solvers except for SNOPT, whose median is shown as the green, dashed line. We don't provide the data for $N = 50$ views for "fmincon" and SNOPT. The costs for our solver and the SDP [14] were the same, but "fmincon" and SNOPT return solutions with larger costs.

$\boldsymbol{P}_i \in \mathbb{R}^{3\times 4}, i = 1, \ldots, N$. Due to noisy data, the point $\boldsymbol{Q}$ cannot be recovered exactly, and thus we seek the minimum correction $\boldsymbol{x}^i \in \mathbb{R}^2$ in the ℓ_2 sense of the observations $\boldsymbol{p}_{\text{i}}$ that assures the existence of $\boldsymbol{Q}$. For that we leverage the $M = \binom{N}{2}$ bilinear or epipolar constraints, one per each pair of distinct poses, through the epipolar matrix $\boldsymbol{E}_{\text{i,j}} \in \mathbb{R}^{3\times 3}$ as it was done in [14]. This set of constraints has been shown to fully define the relation between the observations under a general configuration of cameras although it fails for coplanar motions, including collinear setups, see *e.g.* [15], [27]. Empirically, though, we show in Section IV, that this exception only holds for *exact* configurations and a small perturbation in the configuration can break the degeneracy.

Formally, we seek the correction $\boldsymbol{x}^i \in \mathbb{R}^2$ associated with the i-th observation $\tilde{\boldsymbol{p}}_{\text{i}} \in \mathbb{R}^3$ such that $\tilde{\boldsymbol{p}}_{\text{i}} + \boldsymbol{S}\boldsymbol{x}^i$ is the exact projection of the 3D point $\boldsymbol{Q}$, with $\boldsymbol{S} = [\boldsymbol{I}_2|\boldsymbol{0}_{2\times 1}]^T \in \mathbb{R}^{3\times 2}$ and $\tilde{\boldsymbol{p}}_{\text{i}} \in \mathbb{R}^3$ is the homogeneous form of $\boldsymbol{p}_{\text{i}}$ (last entry to one). By concatenating all the corrections in a single column vector $\boldsymbol{x} = [{\boldsymbol{x}^1}^T, \ldots, {\boldsymbol{x}^N}^T]^T \in \mathbb{R}^{2N}$, the N-view triangulation problem based only on bilinear constraints has the form [14]

$$f^\star = \min_{\boldsymbol{x}\in\mathbb{R}^{2N}} \|\boldsymbol{x}\|_2^2 \qquad \text{(O)}$$
$$\text{subject to } (\tilde{\boldsymbol{p}}_{\text{i}} + \boldsymbol{S}\boldsymbol{x}^i)^T \boldsymbol{E}_{\text{i,j}} (\tilde{\boldsymbol{p}}_{\text{j}} + \boldsymbol{S}\boldsymbol{x}^j) = 0, {\scriptstyle i\neq j = 1,\ldots,N}$$

We assume that a (real) solution exists, *i.e.*, a point that generates the observations exists, and that *not all* the observations lie close to the epipoles [3]. Problem O is an instance of a Quadratically Constrained Quadratic Problem (QCQP), that is non-convex and in general NP-hard to solve. In order to estimate the global optimum of this problem in an efficient way, we first re-write it in a more standard form. All the constraints can be written in terms of the variable vector $\boldsymbol{x}$ by padding with zeros when necessary as $(\tilde{\boldsymbol{p}}_{\text{i}} + \boldsymbol{S}\boldsymbol{x}^i)^T \boldsymbol{E}_{\text{i,j}} (\tilde{\boldsymbol{p}}_{\text{j}} + \boldsymbol{S}\boldsymbol{x}^j) = \boldsymbol{x}^T \boldsymbol{A}_k \boldsymbol{x} + 2\boldsymbol{a}_k^T \boldsymbol{x} + b_k = 0, k = 1, \ldots, M$, with $\boldsymbol{A}_k \in \mathbb{S}^{2N}$, $\boldsymbol{a}_k \in \mathbb{R}^{2N}$ and $b_k \in \mathbb{R}$. The generic QCQP is

$$f^\star = \min_{\boldsymbol{x}\in\mathbb{R}^{2N}} \|\boldsymbol{x}\|_2^2 \qquad \text{(QCQP)}$$
$$\text{subject to } \boldsymbol{x}^T \boldsymbol{A}_k \boldsymbol{x} + 2\boldsymbol{a}_k^T \boldsymbol{x} + b_k = 0, k = 1, \ldots, M$$

Notice that the form of the constraints in Prob. O defines a pattern on $\boldsymbol{A}_i, \boldsymbol{a}_i, b_i$ and some cases that are not possible, *e.g.* $b_i \neq 0$ *and* $\boldsymbol{a}_i = \boldsymbol{0}_{2N}$. From problem QCQP we can derive convex relaxations in the form of semidefinite problems that can be solved in polynomial time by off-the-shelf solvers [14]. Unfortunately, these tools have polynomial time complexity in the number of variables and constraints, making them too slow to be employed in real-time systems. Our own implementation of these relaxations shows that for more than 50 views the solver becomes very slow, requiring more than 30 seconds to solve a single problem. To overcome this limitation, we propose a fast solver that is able to certify optimality but runs in microseconds for $N = 10$ views. Figure 2 shows the computational time in seconds required by the different solvers (Y-axis) as a function of the number of views (cameras) (X-axis): Fig. 2a with our proposal; and Fig. 2b with the other solvers through their matlab interface (the SDP in [14] with CVX [31] as modeling tool and SDPT3 as solver; using "fmincon" with 'sqp' as algorithm; and using SNOPT [19]). Notice that the solvers in Fig. 2b are slower than our proposal, requiring more than 80 milliseconds to obtain a solution for $N = 10$ views, whereas our solver requires less than 0.15 milliseconds. The mean and median times are the same for all the solvers, except for SNOPT. The median for $N = 10$ is greater than 80 ms.

B. Fast solver

The key idea behind our proposal is to approximate the feasible set of the original problem QCQP with linear constraints, which allows us to solve the approximation via, *e.g.* a complete orthogonal decomposition (COD) or Singular Value Decomposition (SVD), available in most mathematical libraries *e.g.* Eigen [22], making our proposal suitable for most devices. In the limit and under some conditions, these approximations recover the original problem, hence returning a feasible solution for QCQP. The approximations are obtained by taking the first-order approximation of the quadratic constraints at the current solution $\boldsymbol{x}_0$, that is, the Taylor expansion at this point. Formally, the problem to solve reads

$$f_l^\star = \min_{\boldsymbol{x}\in\mathbb{R}^{2N}} \|\boldsymbol{x}\|_2^2, \qquad \text{(L-K)}$$
$$\text{subject to}$$
$$(2\boldsymbol{a}_k^T + 2\boldsymbol{x}_0^T \boldsymbol{A}_k)\boldsymbol{x} + (b_k - \boldsymbol{x}_0^T \boldsymbol{A}_k \boldsymbol{x}_0) = 0, {\scriptstyle k=1,\ldots,M}.$$

We form the data matrices

$$\boldsymbol{C}_l = [2\boldsymbol{a}_1 + 2\boldsymbol{A}_1\boldsymbol{x}_0| \ldots |2\boldsymbol{a}_M + 2\boldsymbol{A}_M\boldsymbol{x}_0]^T \in \mathbb{R}^{M\times 2N}, \quad (1)$$
$$\boldsymbol{d}_l = [b_1 - \boldsymbol{x}_0^T \boldsymbol{A}_1 \boldsymbol{x}_0, \ldots, b_M - \boldsymbol{x}_0^T \boldsymbol{A}_M \boldsymbol{x}_0]^T \in \mathbb{R}^M. \quad (2)$$

The solution $\boldsymbol{x}_l$ to problem L-K is the minimum-norm minimizer of $\|\boldsymbol{C}_l\boldsymbol{x}_l + \boldsymbol{d}_l\|_2$. We assume that there exists at least one solution x_l for which the error $\|C_l x_l + d_l\|_2$ is small. Empirically, this error is always zero. The algorithm consists in solving linear systems and updating the coefficient matrix C_l and vector d_l with the previous solution x_{l-1} to obtain the new solution x_l, stopping when two consecutive solutions are the same. Note that the feasible set of the subproblems coincides with the polynomial solved by Newton's method for multivariate problems for rank deficient matrices that pretends to find one of the roots of the system [32]. If the algorithm converges to $x_l = x_{l-1}$ and the residual of $C_l x_l + d_l = 0_{M\times 1}$ is zero (up to some accuracy), then the final solution is feasible for the original prob. O. Assuming that the original problem has more than one feasible, real solution (see [4]), we are left to certify that the solution is also the optimum for Prob. O.

Remark 1: Assuming that the solution x_l is feasible for the original QCQP, then the matrix C_l is the Jacobian of the quadratic constraints in Prob. QCQP evaluated at the point x_l. The rank of the Jacobian is associated with the codimension of the space defined by the constraints (from the Jacobian criterion [33, Th 2.3]), that, as it was shown in [27], is $2N-3$ for all *feasible* solutions, while the Jacobian is a $M \times 2N$ matrix. Since the Jacobian is not full-rank the Linear Independence Constraint Qualification (LICQ) doesn't hold, which is the condition assumed by most solvers (Section II). This condition is also assumed by iterative algorithms for constrained problems, *e.g*. Sequential Quadratic Programming (SQP) [34, Ch. 18]. While previous works have tackled constraint degeneracy in SQP and have proposed modifications of the algorithm, our evaluation showed that these modifications make the algorithms slow (see Fig. 2). We must say, however, that the rank-deficiency of the Jacobian is more apparent the closer the estimation is to the feasible set.

C. Optimality certification

Whereas it is possible to employ one of the convex relaxations proposed in [14] for the certification, in this work we aim to provide fast alternatives. Thus, we rely on the dual problem of Prob. QCQP (Appendix A and [14]), but derive an optimality certificate, which is sufficient but not necessary for optimality, see Sec. II. Thus, we don't solve the dual problem from scratch, but rather leverage it to certify a given solution. Assuming that strong duality holds and that the global optimum x is given, we are left to find Lagrange multipliers $\lambda \in \mathbb{R}^M$ such that $C_l^T \lambda = 2x$ for which the Hessian $H(\lambda)$ is positive semidefinite (PSD)

$$H(\lambda) \doteq \begin{pmatrix} I_{2N} - \sum_{i=0}^{M} \lambda_i A_i & -\sum_{i=0}^{M} \lambda_i a_i \\ -\sum_{i=0}^{M} \lambda_i a_i^T & -\sum_{i=0}^{M} \lambda_i b_i - x^T x \end{pmatrix}. \quad (3)$$

Recall that $C_l \in \mathbb{R}^{M\times 2N}$ is the Jacobian of the constraints evaluated at the feasible primal point x, hence being rank deficient for $N \geq 4$, thus making the solution to $C_l^T \lambda = 2x$ not unique. Empirically we observe that the minimum-norm solution is actually feasible for the dual problem, *i.e.* the Hesian $H(\lambda)$ in Eq. (3) is PSD. Although this is just an observation we show in Section IV that this solution certifies *all* the problem instances. Our certifier has therefore two steps: (1) compute the Lagrange multipliers in closed-form; and (2) check the eigenvalues of the Hessian evaluated at that point. With this approach, the certifier will fail to certify optimality if at least one of the next three conditions is met: (1) the solution x is not optimal; (2) strong duality doesn't hold; or (3) the minimum-norm λ isn't the optimal solution.

Algorithm 1: N-view triangulation solver

Data: N noisy observations $\tilde{p}_i$ and projections P_i
Result: Corrected observations $\tilde{p}_i + S x^i$; cert. IsOpt
`// Create constraints`
Create $M = \binom{N}{2}$ constraints between pairs of images;
$l \leftarrow 1$, $x_0 \leftarrow 0_{2N}$;
repeat
    `// Update coefficient matrices`
    Update C_l, d_l with x_{l-1} from Eqs. (1),(2);
    `// Compute new solution`
    Solve $\|C_l x_l + d_l\|_2$ for minimum-norm x_l;
    $l \leftarrow l + 1$;
until *convergence or max. iters*;
Compute Q from corrected observations;
if *solution is feasible for* (O) AND Q *is exact* **then**
    `// Check suff. cond. optimality`
    if *Hessian in certifier (Eq. (3)) is PSD* **then**
        `// Solution is optimal`
        IsOpt = True ;
    else
        `// assumptions don't hold`
        IsOpt = inconclusive ;
else
    `// Solution not feasible`
    IsOpt = inconclusive ;

D. Algorithm

Our proposal is summarized in Alg. 1. We define convergence when two consecutive solutions x_l and x_{l-1} are close in the sense $\|x_l - x_{l-1}\|_2^2 \leq 3 \cdot 10^{-10}$. We limit the number of iterations to five, and for feasibility we require that the ℓ_2 norm of the constraints for the final solution x_l is below the threshold $\epsilon_5 = 5 \cdot 10^{-11}$. The last condition is checked for sanity: due to numerical errors on the matrices and the threshold applied to the rank of C_l, the decomposition doesn't necessarily match *exactly* the coefficient matrix, which may lead to non-zero residuals for the linear system. We consider that the corrected observations are originated by a real point Q if the least singular value of the matrix from the linear method is below the threshold $\epsilon_5 = 5 \cdot 10^{-11}$. Last, we consider that the solution is optimal if the minimum eigenvalue of the Hessian is greater than $\epsilon_{\min} = -1 \cdot 10^{-09}$.

IV. Evaluation

We performed the evaluation on a standard desktop with 32 GB RAM, i7 − 3770 CPU, 3.4 GHz and Ubuntu 16.04.

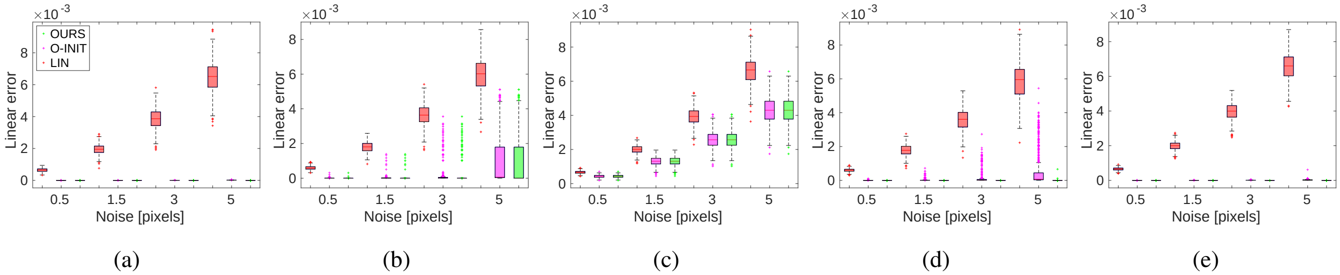


Fig. 3: Error for the 3D reconstruction of the point (least singular value) with $N = 10$ views as a function of the noise level (X-axis) in the configurations with (a) general; (b) planar; (c) linear; (d) noisy planar; and (e) noisy linear motion.

A. *Evaluation on synthetic data*

We follow the procedure in [14] to generate the synthetic data: we define a set of $N = 10$ random 3D points at a distance d units from the origin on a ball with radius $d/4$ and avoid points far away from the cameras by limiting this radius to 8 units. Next we generate sets of N poses that define the configuration (motion) of the problem. In the first set, we generate random poses with translation within a ball of 5 units and angle of rotation bounded above by 0.5 rads. In the second set (planar motion), the camera follows a circular path of radius 5 units centered at the point $[0, 0, d]$, that is, the center of the motion is also the center of the point cloud, and all the cameras point towards this same point. In the last set (linear motion), the camera moves horizontally along the X-axis with zero rotation and maximum parallax of 5 units. The last two sets are degenerate configurations for our formulation, see [27], although they are common in autonomous cars, *e.g.* a long road. However, we notice that this degeneracy is only present for these *exact* paths and a small perturbation on the translation and/or rotation can break it. In practice, as we see on the evaluation on real data, *e.g.* the CORRIDOR dataset, the noiseless setup is not usually found and real-world situations are better modeled by these perturbations. To show this behavior, we run our algorithm also with the noisy motions, which were generated by adding a rotation of 0.002 rads to the original rotation. For all the configurations we obtain the observation on each camera by assuming a pinhole camera model with focal length 512 and image size 512^2. We perturb the observations by adding Gaussian noise with standard deviation σ in pixels on the image plane. We consider the noise levels $\sigma \in \{0.5, 1.5, 3.0, 5.0\}\texttt{pix}$ and generate $N \in \{5, 10, 50, 100, 200\}$ different camera poses, that is, $\binom{N}{2} = \{10, 45, 1225, 4950, 19900\}$ constraints. For each configuration we generate 50 problem instances.

In all the problem instances, the ℓ_2 norm of the constraints was numerically zero (under $1e - 20$), thus the solution returned by our solver was feasible for prob. O. In Figure 3 we report the least singular value for the original observations DLT, the result with our initialization O-INIT without refinement (line 2 in Alg. 1); and the final result of our algorithm OURS. In this figure the first three columns show from left to right the general, planar and collinear cameras, whereas the last two columns stand for noisy planar and collinear cameras. Notice that for general and noisy cameras the solver returns always observations that generate exactly a 3D point, while this is not the case for some of the problems for planar and linear paths (the degenerate configurations). Last, the sufficient condition in Sec. III-C certified all the solutions where the algorithm converges to a feasible solution (that is, all but pure planar and linear motions) and we obtain minimum eigenvalues of the Hessian of the order of $1 \cdot 10^{-12}$. Empirically, we obtain non-feasible and non-certified solutions for highly noisy problem instances, *e.g.* $\sigma = 100\texttt{pix}$ and $N = 5, 6$ views. For problems with $N = 9$ views failure of the algorithm starts from $\sigma = 200\texttt{pix}$. We also notice that the proposal becomes unstable for noise $\sigma = 1000\texttt{pix}$, even for N large.

Computational time: Figure 2a shows the mean computational time in seconds required by the proposed certifiable solver for different number of views (X-axis) with general motion and noise $\sigma = 3\ \texttt{pix}$, averaging the results for 20 random problem instances. Above each black circle we report the number of constraints $M = \binom{N}{2}$. We estimate the solution for problem instances with $N = 10$ views are solved in 60 microseconds and in 20 microseconds for $N = 5$ views. We provide next the computational time required for problem with up to 200 views. The initialization (line 2 in Alg. 1) and the refinement (lines $7 - 9$) are the same operation, and thus they take the same time with independence of the camera configuration (provided it's not degenerate) and noise. As a function of the number of views N, we have in average (μs): $N = 5$ goes to 9.655; $N = 10$ to 21.922; $N = 50$ to 1892; $N = 100$ to 22984.8; and $N = 200$ to 316516.0. The number of iterations ranges between one and three depending on the level of noise (increasing) and the number of views (decreasing). For lateral motions we obtain similar times independently of noise to those for general motions with low noise. On the other hand, for orbital paths we obtain the times of general motion for large noise. For the certifier, we observe that the required computational time only depends on the number of views (hence the number of constraints), but not on the noise level nor the configuration. In average for $N = 5$ views the certifier requires 30 microseconds, for $N = 10$ it takes 89 microseconds, for $N = 50$ goes to 5.12 milliseconds, for 100 views to 54 milliseconds and for $N = 200$ views to 748 milliseconds. Note that this condition is only evaluated on the final solution, and involves the estimation of the M Lagrange multipliers, which is the most consuming step, requiring the next values: for $N = 5$ it takes 20 microseconds, for $N = 10$ goes to 42 microseconds, for $N = 50$ to 4.3 milliseconds, $N = 100$ to 57.5 milliseconds and $N = 200$ to 695 milliseconds. Although the certifier increases the computational time of Alg. 1, our proposal is still faster than the state-of-the-art.

B. Evaluation on real data

To conclude we evaluate our algorithm on real data with the sequences: CORRIDOR, DINOSAUR, MODEL HOUSE [2] and NOTREDAME [3]. Table I collects the information about these sequences, including the number of problem instances with more than 4 and 10 views. The time required by our algorithm to triangulate *all* the points is reported in column TIME SOL., whereas column TIME CERT. reports the time required to certify these solutions, both in milliseconds. Note that the **total time** is obtained by adding both columns. The most consuming problem instance in terms of total time (estimation and certification) in μs is included in column MAX. TIME, followed by the number of views of the problem instance. Note that our proposal is faster than other certifiable methods (see the reported results in [14]). Further, the algorithm converges to a feasible solution for more than 99% of the problem instances (column ϵ_5-OPTIMAL), whereas the certifier was able to certify all these feasible solutions as the global optima.

V. Conclusions and Future Work

Robotic systems, such as autonomous cars or UAVs, require fast, optimal solutions for their critical navigation tasks, *e.g.* localization and mapping. Typically, the map is represented by a set of 3D points originated from their observations. Estimating the coordinates of such points given N views, the so-called N-view triangulation problem, is key for having a precise and consistent world representation. In this work we proposed a fast and certifiable solver for this problem that relied on a series of linear approximations of the triangulation problem which in the limit, provided a solution for the original problem, that was later certified as optimal by the proposed certifier. The proposal is based on basic operations that are available in any linear algebra library, *e.g.* Eigen and was able to find and certify the optimal solution in 150 microseconds for $N = 10$ views. The evaluation on synthetic and real data showed that the proposal consistently found and certified optimal solutions.

As future work, we project to improve the present algorithm since currently the decomposition of the coefficient matrix has to be computed at each iteration, even though the matrix $\boldsymbol{C}_l$ is just a perturbation of the previous $\boldsymbol{C}_{l-1}$. Additionally, we are interested in analyzing wherever the proposed algorithm can be extended to other problems.

[2] https://www.robots.ox.ac.uk/~vgg/data/mview/
[3] http://phototour.cs.washington.edu/datasets/

References

[1] Ruben Gomez-Ojeda, Francisco-Angel Moreno, David Zuñiga-Noël, Davide Scaramuzza, and Javier Gonzalez-Jimenez. Pl-slam: A stereo slam system through the combination of points and line segments. *IEEE Transactions on Robotics*, 35(3), 2019.

[2] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5), 2015.

[3] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[4] Richard I Hartley and Peter Sturm. Triangulation. *Computer vision and image understanding*, 68(2), 1997.

[5] H. Stewenius, F. Schaffalitzky, and D. Nister. How hard is 3-view triangulation really? In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 686–693 Vol. 1, 2005.

[6] Kui Yang, Wei Fang, Yan Zhao, and Nianmao Deng. Iteratively reweighted midpoint method for fast multiple view triangulation. *IEEE Robotics and Automation Letters*, 4(2):708–715, 2019.

[7] Jos F Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999.

[8] Kim-Chuan Toh, Michael J Todd, and Reha H Tütüncü. Sdpt3—a matlab software package for semidefinite programming, version 1.3. *Optimization methods and software*, 11(1-4):545–581, 1999.

[9] Heng Yang and Luca Carlone. A quaternion-based certifiably optimal solution to the wahba problem with outliers. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[10] Ji Zhao, Wanting Xu, and Laurent Kneip. A certifiably globally optimal solution to generalized essential matrix estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12034–12043, 2020.

[11] Jesus Briales and Javier Gonzalez-Jimenez. Convex global 3d registration with lagrangian duality. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] Jesus Briales, Laurent Kneip, and Javier Gonzalez-Jimenez. A certifiably globally optimal solution to the non-minimal relative pose problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

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

[14] Chris Aholt, Sameer Agarwal, and Rekha Thomas. A qcqp approach to triangulation. In *European Conference on Computer Vision*, pages 654–667. Springer, 2012.

[15] Sameer Agarwal, Andrew Pryhuber, and Rekha R Thomas. Ideals of the multiview variety. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[16] Diego Cifuentes. A convex relaxation to compute the nearest structured rank deficient matrix. *SIAM Journal on Matrix Analysis and Applications*, 42(2):708–729, 2021.

[17] Stephen Wright, Jorge Nocedal, et al. Numerical optimization. *Springer Science*, 35(67-68):7, 1999.

[18] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.

[19] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 2005.

[20] Stephen J Wright. Modifying sqp for degenerate problems. *SIAM Journal on Optimization*, 13(2):470–497, 2002.

[21] Alexey F Izmailov and Mikhail V Solodov. Newton-type methods for optimization problems without constraint qualifications. *SIAM Journal on Optimization*, 15(1):210–228, 2004.

[22] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[23] Seong Hun Lee and Javier Civera. Closed-form optimal two-view triangulation based on angular errors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[24] Richard Hartley and Fredrik Kahl. Optimal algorithms in multiview geometry. In *Asian conference on computer vision*. Springer, 2007.

[25] Seong Hun Lee and Javier Civera. Robust uncertainty-aware multiview triangulation. *arXiv preprint arXiv:2008.01258*, 2020.

[26] Jia Chen, Dongli Wu, Peng Song, Fuqin Deng, Ying He, and Shiyan Pang. Multi-view triangulation: Systematic comparison and an improved method. *IEEE Access*, 8:21017–21027, 2020.

Name seq.	Nr. views	Nr. points	$N > 4$	$N > 10$	time sol. [ms]	time cert. [ms]	max. time [μs] (N)	ϵ_5-optimal (%)
MODEL HOUSE	10	672	289	7	16.118	17.519	469.07 (10)	100
CORRIDOR	11	737	396	104	35.089	30.204	472.3 (9)	99.86
DINOSAUR	36	4983	1516	63	101.30	94.74	1043.2 (21)	100
NOTREDAME	715	127431	71481	13633	30354	34020	1550150 (192)	99.2

TABLE I: **Real data** Information about the employed datasets. TIME SOL.: required time to triangulate *all* the points; and TIME CERT. time to certify *all* the solutions, both in ms (the **total time** is the sum of both columns). MAX. TIME: required time by the most time consuming problem instance (followed by the number of views) in μs considering solution estimation and certification. ϵ_5-OPTIMAL: percentage of certified optimal solutions.

[27] Matthew Trager, Martial Hebert, and Jean Ponce. The joint image handbook. In *Proceedings of the IEEE international conference on computer vision*, pages 909–917, 2015.

[28] Luca Carlone and Frank Dellaert. Duality-based verification techniques for 2d slam. In *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015.

[29] Jesus Briales and Javier Gonzalez-Jimenez. Fast global optimality verification in 3d slam. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016.

[30] Anders Eriksson, Carl Olsson, Fredrik Kahl, and Tat-Jun Chin. Rotation averaging and strong duality. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[31] Michael Grant and Stephen Boyd. Cvx: Matlab software for disciplined convex programming, version 2.1, 2014.

[32] Stepan Yu Gatilov. Using low-rank approximation of the jacobian matrix in the newton–raphson method to solve certain singular equations. *Journal of Computational and Applied Mathematics*, 272, 2014.

[33] Daniel Perrin. *Algebraic geometry: an introduction*. Springer Science & Business Media, 2007.

[34] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

APPENDIX A
DUAL PROBLEM FOR THE PRIMAL PROB. QCQP

To derive the dual problem, we introduce an homogeneous variable $y \in \mathbb{R}$ into the primal problem (QCQP) as

$$f^\star = \min_{\boldsymbol{x} \in \mathbb{R}^{2N}, y \in \mathbb{R}} \boldsymbol{x}^T \boldsymbol{x}, \tag{D}$$

$$\text{subject to } [\boldsymbol{x}^T, y] \underbrace{\begin{pmatrix} \boldsymbol{A}_i & \boldsymbol{a}_i \\ \boldsymbol{a}_i^T & b_i \end{pmatrix}}_{\tilde{\boldsymbol{A}}_i} [\boldsymbol{x}^T, y]^T = 0, y^2 = 1.$$

Then, the dual problem has the form $d^\star = \max_{\boldsymbol{\lambda} \in \mathbb{R}^M, \rho \in \mathbb{R}} d(\boldsymbol{\lambda}, \rho)$, where $d(\boldsymbol{\lambda}, \rho)$ is the dual function defined as $d(\boldsymbol{\lambda}, \rho) = \min_{\boldsymbol{x} \in \mathbb{R}^{2N}} \mathcal{L}(\boldsymbol{x}, y, \boldsymbol{\lambda}, \rho)$ and the Lagrangian $\mathcal{L}(\boldsymbol{x}, y, \boldsymbol{\lambda}, \rho) = [\boldsymbol{x}^T, y] \boldsymbol{H} [\boldsymbol{x}^T, y]^T + \rho$, which has a finite minimum w.r.t. $\boldsymbol{x}$ equal to zero *iff* the Hessian $\boldsymbol{H}(\boldsymbol{\lambda}, \rho)$ is positive semidefinite (PSD) with

$$\boldsymbol{H}(\boldsymbol{\lambda}, \rho) \doteq \begin{pmatrix} \boldsymbol{I}_{2N} - \sum_{i=0}^M \lambda_i \boldsymbol{A}_i & -\sum_{i=0}^M \lambda_i \boldsymbol{a}_i \\ -\sum_{i=0}^M \lambda_i \boldsymbol{a}_i^T & -\sum_{i=0}^M \lambda_i b_i - \rho \end{pmatrix}. \tag{4}$$

Restringing our attention to finite values, the dual problem $d^\star = \max_{\boldsymbol{\lambda} \in \mathbb{R}^M, \rho \in \mathbb{R}} \rho$, subject to $\boldsymbol{H}(\boldsymbol{\lambda}, \rho) \succeq 0$. Assuming strong duality, *i.e.* $d^\star = \rho = f^\star$ and that the given solution $\boldsymbol{x} \in \mathbb{R}^{2N}$ is the global optimum, and so $f^\star = \boldsymbol{x}^T \boldsymbol{x}$ we have the relation $\boldsymbol{H}(\boldsymbol{\lambda}, \boldsymbol{x}^T \boldsymbol{x})[\boldsymbol{x}^T, y]^T = \boldsymbol{0}_{M \times 1}$ with $y = 1$, from which we derive the expression for $\boldsymbol{\lambda} \in \mathbb{R}^M$ as $2[\boldsymbol{A}_i \boldsymbol{x} + \boldsymbol{a}_i, \ldots, \boldsymbol{A}_M \boldsymbol{x} + \boldsymbol{a}_M] \boldsymbol{\lambda} = 2\boldsymbol{x}$. If the solution $\boldsymbol{\lambda}$ to this linear system is feasible for the dual (Hessian is PSD), then by weak duality: (1) strong duality holds; (2) $\boldsymbol{\lambda}$ and $\boldsymbol{x}$ are the global optima for their respective problems.