

Globally Optimal DLS Method for PnP Problem with Cayley parameterization

Gaku Nakano
g-nakano@cq.jp.nec.com

Information and Media Processing Labs
NEC Corporation
Kawasaki, Japan

Abstract

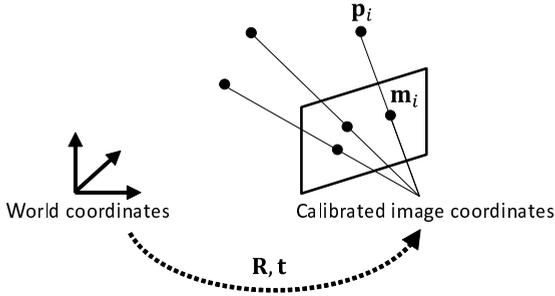
This paper proposes a globally optimal direct least squares (DLS) method for the PnP problem with Cayley parameterization. First we derive a new optimality condition without Lagrange multipliers, which is independent of any rotation representations. Then, we show that the new equation can be solved by several types of parameterizations and among them, Cayley parameterization is the most efficient. According to the experimental results, the proposed method represented by Cayley parameterization is more than three times faster than the state-of-the-art method while maintaining equivalent accuracy.

1 Introduction

The perspective- n -point (PnP) problem, which estimates 3D rotation and translation of a calibrated camera from n pairs of known 3D points and corresponding 2D points on an image, is a classical problem but still fundamental in the computer vision community. It is well studied that the PnP problem can be solved by at least three points [3, 4, 5]. If the number of the points is greater than or equal to four, the PnP problem becomes a nonlinear problem where the number of the solutions depend on n and the shape of the scene. Such conditions are varied on different applications, *e.g.* augmented reality, robot navigation, incremental structure-from-motion, etc. Therefore, the following features are required for a practical usage: efficiency for changing the number of the points, scalability for applications in both planar and non-planar scenes, and global optimality for avoiding local minima.

Since Lepetit *et al.* has introduced the first linear complexity method with respect to the number of the points n , named EPnP [6], many $O(n)$ methods have been proposed in the literature. EPnP is actually computationally efficient; however, it does not assure optimality and suffers from convergence to a local minimum especially for $n \leq 5$ situations [6]. In addition, different implementations are required for planar and non-planar scenes, respectively. For ensuring scalability and global optimality, two iterative convex relaxation methods have been proposed by Schweighofer and Pinz [7] and Hmam and Kim [8], respectively. Their methods using semi-definite programming (SDP) take more than 100 msec/frame for solving the PnP problem. Despite the slow computation, the SDP based methods often fail to converge to global optimum due to the difficulty in controlling the relaxation.

To overcome the computational cost issue, direct least squares (DLS) approach was developed by Hesch and Roumeliotis [9]. The concept of DLS is to find all stationary points of the first optimality condition represented as a system of multivariate polynomials. Solving

Figure 1: The PnP problem

the polynomial system results in eigenvalue decomposition, which can be computed efficiently using a linear algebra library. Since Cayley parameterization, which is used in the Hesch’s method, has a singularity for any 180 degree rotations, OPnP [16] and UPnP [8] adopt a quaternion-based parameterization for avoiding the singularity. However, for efficiency, those two methods require a special implementation to eliminate the sign ambiguity of quaternion, or 2-fold symmetry [11].

This paper proposes an efficient, scalable, and globally optimal DLS method parameterized by Cayley representation, which has been regarded as a unsuitable parameterization due to its singularity. First we derive a new optimality condition without Lagrange multipliers. The number of the solutions to the new equation is not changed for any types of rotational parameterization. Then, we show that Cayley parameterization is the most compact representation and can be used for calculating the optimal solution with avoiding the singularity. Due to no 2-fold symmetry, the proposed method can be implemented by using Kukulova’s automatic generator [9] and is more than three times faster than OPnP.

2 Theoretical Background

2.1 PnP Problem

This section briefly describes a mathematical formulation of the PnP problem. Figure 1 shows a concept of the PnP problem.

Letting $\mathbf{p}_i = [x_i, y_i, z_i]^T$ be an i -th 3D point and $\mathbf{m}_i = [u_i, v_i, 1]^T$ be the corresponding calibrated image point in homogeneous coordinates, the perspective projection from \mathbf{p}_i to \mathbf{m}_i can be expressed by

$$\mathbf{m}_i \propto \mathbf{R}\mathbf{p}_i + \mathbf{t}, \quad i = 1, \dots, n, \quad (1)$$

where \propto denotes equality up to scale, $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$ are the rotation matrix and the translation vector from the world coordinates to the camera coordinates, respectively.

The PnP problem aims to find the six unknown parameters, \mathbf{R} and \mathbf{t} , from known point correspondences, \mathbf{p}_i and \mathbf{m}_i . Although the dimension of Equation (1) is three, one point pair gives only two constraints due to the scale ambiguity. Therefore, at least three correspondences are required to solve \mathbf{R} and \mathbf{t} . In the case of $n \geq 4$, the PnP problem can be formulated

as a nonlinear optimization

$$\begin{aligned} \min_{\mathbf{R}, \mathbf{t}} \quad & \sum_{i=1}^n \left\| [\mathbf{m}_i]_{\times} (\mathbf{R}\mathbf{p}_i + \mathbf{t}) \right\|^2 \\ \text{s.t.} \quad & \mathbf{R}^T \mathbf{R} = \mathbf{I}, \quad \det(\mathbf{R}) = 1 \end{aligned} \quad (2)$$

where $[\cdot]_{\times}$ denotes a matrix representation of the vector cross product. The cost function in Equation (2) is based on a minimization of algebraic error.

We can express \mathbf{t} as a function of \mathbf{R} since \mathbf{t} is linear and does not have any constraints in Equation (2). Hence, the PnP problem can be rewritten as

$$\begin{aligned} \min_{\mathbf{R}} \quad & \mathbf{r}^T \mathbf{M} \mathbf{r} \\ \text{s.t.} \quad & \mathbf{R}^T \mathbf{R} = \mathbf{I}, \quad \det(\mathbf{R}) = 1 \end{aligned} \quad (3)$$

where $\mathbf{r} \in \mathbb{R}^9$ is a vector form of \mathbf{R} and $\mathbf{M} \in \mathbb{R}^{9 \times 9}$ is a symmetric coefficient matrix computed from the known parameters, \mathbf{p}_i and \mathbf{m}_i . Zheng *et al.* [14] proposes an efficient method for calculating \mathbf{M} , named vectorization technique. For the derivation details, see Appendix A in the supplemental material.

2.2 Gröbner basis solver

Not only the PnP problem but also many computer vision problems result in solving a system of multivariate polynomial equations. One way to solve the polynomial equations is to compute the Gröbner basis, which is a special set of multivariate polynomials so that it has the same solutions as the original equations and is easy to solve. Gröbner basis method has been widely known since Stewenius *et al.* [15] introduced it for solving the two-view geometry. Early works try to compute Gröbner basis manually. However, it takes a lot of time and is difficult to assure reliability.

Kukelova *et al.* [6] proposed an automatic generator, which provides a MATLAB code for Gröbner basis solvers. The generator analyses the Gröbner basis and the number of solutions in the finite field, then outputs the code for constructing an elimination template and an action matrix. The elimination template is a coefficient matrix from the initial equations to determine which equations are required for the Gröbner basis. The action matrix is extracted from the elimination template and its eigenvalues are identical to the solutions of the original polynomial equations. The readers can refer to [6] for more details of Gröbner basis.

Once the code is given, the actual computations are decomposition of the elimination template by QR or Gaussian elimination and the eigenvalue decomposition of the action matrix. As the number of the unknowns and the maximum degree become larger, the computation becomes less efficient and unstable because the size of the elimination template and the action matrix become large. Therefore, the key to the numerical efficiency and stability is to find a minimal set of parameterization for describing the original problem.

3 Proposed Method

3.1 New optimality condition without Lagrange multipliers

In this section we formulate a new optimality condition that satisfies Equation (3). Assuming that \mathbf{R} is a general rotation matrix parameterized by nine unknowns, Lagrange function of the PnP problem can be written as

$$L(\mathbf{R}, \mathbf{S}, \lambda) = \frac{1}{2} \mathbf{r}^T \mathbf{M} \mathbf{r} - \frac{1}{2} \text{trace}(\mathbf{S}(\mathbf{R}^T \mathbf{R} - \mathbf{I})) - \lambda(\det(\mathbf{R}) - 1). \quad (4)$$

Here, λ is a Lagrange multiplier and $\mathbf{S} \in \mathbb{R}^{3 \times 3}$ is a symmetric matrix of Lagrange multipliers. The multiplier $1/2$ is merely for convenience. Then, the first-order optimality condition is given by

$$\frac{\partial L}{\partial \mathbf{R}} = \text{mat}(\mathbf{M} \mathbf{r}) - \mathbf{R} \mathbf{S} - \lambda \mathbf{R} = \mathbf{0}, \quad (5)$$

$$\frac{\partial L}{\partial \mathbf{S}} = \mathbf{R}^T \mathbf{R} - \mathbf{I} = \mathbf{0}, \quad (6)$$

$$\frac{\partial L}{\partial \lambda} = \det(\mathbf{R}) - 1 = 0, \quad (7)$$

where $\text{mat}(\cdot)$ is a reshaping operator from a 9×1 vector to a 3×3 square matrix. From Equation (5), we have

$$\text{mat}(\mathbf{M} \mathbf{r}) = \mathbf{R}(\mathbf{S} + \lambda \mathbf{I}). \quad (8)$$

Multiplying \mathbf{R}^T from the left-hand and the right-hand side, respectively, we have

$$\mathbf{R}^T \text{mat}(\mathbf{M} \mathbf{r}) = \mathbf{S} + \lambda \mathbf{I}, \quad (9)$$

$$\text{mat}(\mathbf{M} \mathbf{r}) \mathbf{R}^T = \mathbf{R}(\mathbf{S} + \lambda \mathbf{I}) \mathbf{R}^T. \quad (10)$$

Since $\mathbf{S} + \lambda \mathbf{I}$ is a symmetric matrix, the left-hand side of Equations (9) and (10) must be symmetric matrices. Hence, we obtain the following two equations where the Lagrange multipliers are eliminated:

$$\mathbf{P} = \mathbf{R}^T \text{mat}(\mathbf{M} \mathbf{r}) - \text{mat}(\mathbf{M} \mathbf{r})^T \mathbf{R} = \mathbf{0}, \quad (11)$$

$$\mathbf{Q} = \text{mat}(\mathbf{M} \mathbf{r}) \mathbf{R}^T - \mathbf{R} \text{mat}(\mathbf{M} \mathbf{r})^T = \mathbf{0}. \quad (12)$$

Equations (11) and (12) are the proposed new optimality condition for the PnP problem. Let $P_{j,k}$ and $Q_{j,k}$ be the element of \mathbf{P} and \mathbf{Q} in the j -th row and k -th column, respectively. Obviously the diagonal elements are zeros, $P_{j,j} = Q_{j,j} = 0$. On the other hand, the non-diagonal elements are second degree polynomials in \mathbf{R} . Due to the symmetry, $P_{j,k} = P_{k,j}$ and $Q_{j,k} = Q_{k,j}$, we have six polynomials in total:

$$\begin{aligned} P_{1,2} = 0, & \quad P_{1,3} = 0, & \quad P_{2,3} = 0, \\ Q_{1,2} = 0, & \quad Q_{1,3} = 0, & \quad Q_{2,3} = 0. \end{aligned} \quad (13)$$

Finally, the solution of the PnP problem can be obtained by solving Equation (13) together with the constraints for rotation matrix, Equations (6) and (7).

3.2 Comparison of parameterizations

Although Equation (13) is derived from a general rotation parameterization, any parameterization satisfying Equations (6), (7), and (13) is usable for solving the PnP problem. First we will discuss about three typical parameterizations that are expressed by polynomial: general rotation matrix, quaternion, and Cayley. To compare the parameterizations, we use an automatic Gröbner basis solver by Kukelova *et al.* [9]. Then, we will show that Cayley parameterization is the most compact representation. The readers can find MATLAB code for using the automatic generator in Appendix B in the supplemental material.

3.2.1 Rotation Matrix

As shown in the previous section, a general rotation matrix is represented by nine unknowns with seven constraint equations. Intuitively we have $6 + 7 = 13$ equations and the maximum degree is three. Note that $\mathbf{R}^T \mathbf{R} - \mathbf{I}$ is symmetric and $\det(\mathbf{R})$ is a cubic equation. However, we found that the use of the third-order determinant constraint is too complicated to run the automatic generator since it requires to multiply eighth-order monomials. To overcome this, we introduced quadratic equations that are equivalent to $\det(\mathbf{R}) = 1$ under $\mathbf{R}^T \mathbf{R} = \mathbf{I}$:

$$\mathbf{r}_i - \mathbf{r}_j \times \mathbf{r}_k = \mathbf{0}, \quad (i, j, k) = \begin{cases} (1, 2, 3) \\ (2, 3, 1) \\ (3, 1, 2) \end{cases} \quad (14)$$

where the subscripts i , j , and k denote the column number of \mathbf{R} . Also we added $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ for further reduction of the elimination template. Finally, $6 + 21 = 27$ equations are used for the automatic generator and those equations give 40 solutions as with OPnP. This fact implies that the solution space is identical each other. Although this is one of the simplest representation without singularity nor p -fold symmetry, the size of the elimination template, 1936×1976 , is the largest among the three parameterizations.

3.2.2 Quaternion

Letting $\mathbf{q} = [a, b, c, d]^T$ be a unit quaternion, the rotation matrix is given by

$$\mathbf{R} = \begin{bmatrix} a^2 + b^2 - c^2 - d^2 & 2(bc - ad) & 2(bd + ac) \\ 2(bc + ad) & a^2 - b^2 + c^2 - d^2 & 2(cd - ab) \\ 2(bd - ac) & 2(cd + ab) & a^2 - b^2 - c^2 + d^2 \end{bmatrix}. \quad (15)$$

Equations (6) and (7) are replaced by the following equation

$$\|\mathbf{q}\|^2 - 1 = 0. \quad (16)$$

Therefore, the total number of equations is $6 + 1 = 7$ and the maximum degree is four.

Quaternion does not have a singularity but sign ambiguity, i.e., \mathbf{q} and $-\mathbf{q}$ give the same rotation matrix. This is called 2-fold symmetry [8, 16]. For this reason, the number of solutions is 80, which is doubled for the general rotation matrix representation. The elimination template is expected to be smaller than 630×710 if we introduce Ask *et al.*'s technique [10] for eliminating the 2-fold symmetry. However, the implementation is not easy because the generator is not open publicly, unlike Kukelova's automatic generator. Therefore, only the result from the automatic generator is discussed in this paper.

3.2.3 Cayley parameterization

This is equivalent to replace the first element a in \mathbf{q} by 1. The sign ambiguity is eliminated, therefore, the number of unknowns is reduced from four to three. Also, the number of equations is reduced from seven to six since the unit norm constraint, Equation (16), does not hold anymore. Instead of $\|\mathbf{q}\|^2 = 1$, normalization is required for expressing a rotation matrix:

$$\mathbf{R} = \frac{1}{s} \begin{bmatrix} 1 + b^2 - c^2 - d^2 & 2(bc - d) & 2(bd + c) \\ 2(bc + d) & 1 - b^2 + c^2 - d^2 & 2(cd - b) \\ 2(bd - c) & 2(cd + b) & 1 - b^2 - c^2 + d^2 \end{bmatrix}, \quad (17)$$

where $s = 1 + b^2 + c^2 + d^2$.

It appears that the fractional term $1/s$ changes the solution space. However, \mathbf{P} and \mathbf{Q} are not affected by any types of normalization. Let us consider a case of \mathbf{P} . Substituting Equation (17) into Equation (11), we have

$$\begin{aligned} \left[\frac{1}{s} \mathbf{R} \right]^T \text{mat} \left(\mathbf{M} \left[\frac{1}{s} \mathbf{r} \right] \right) - \text{mat} \left(\mathbf{M} \left[\frac{1}{s} \mathbf{r} \right] \right)^T \left[\frac{1}{s} \mathbf{R} \right] &= \frac{1}{s^2} \left(\mathbf{R}^T \text{mat}(\mathbf{M}\mathbf{r}) - \text{mat}(\mathbf{M}\mathbf{r})^T \mathbf{R} \right) \\ &= \frac{1}{s^2} \mathbf{P}. \end{aligned} \quad (18)$$

Because of $1/s^2 > 0$, we have

$$\frac{1}{s^2} \mathbf{P} = \mathbf{P} = \mathbf{0}. \quad (19)$$

Actually the number of the solutions is 40 in spite of Cayley parameterization. This fact implies that the solution space is the same as that of OPnP. Furthermore, the size of the elimination template is the smallest, 124×164 , among the three representations.

As pointed out by Zheng *et al.* [16, 17], this parameterization is singular at $a = 0$, which occurs at any 180 degree rotations. However, Hesch and Roumeliotis [8] propose an easy trick to avoid it by applying a random rotation to \mathbf{p}_i as a preprocessing. This approach has not been published in a paper but a source code is available on the website¹.

3.2.4 Choosing the best parameterization

Table 1 is a comparison of the above three parameterizations with existing methods. The computational cost and stability highly depend on the size of the elimination template and the action matrix. Therefore, this paper selects Cayley parameterization for solving the PnP problem.

One may ask why higher degree of the polynomial by multiplying \mathbf{R}^T to Equation (8) does not result in a larger number of possible solutions instead of 40. The derivation of Equation (13) is equivalent to a manual elimination of the Lagrange multipliers, that is, the multiplication of \mathbf{R}^T can be interpreted to be a part of operations to build Gröbner basis of the PnP problem with the constraints. Even without the constraints like Cayley parameterization, Equation (13) still holds as proved in Equations (18) and (19). Therefore, the number of the solutions is not changed.

¹<http://www-users.cs.umn.edu/~joel/>

Table 1: Comparison of rotation parameterizations

	Existing methods			Proposed params		
	DLS [8] (Cayley)	OPnP [16] (Non-unit Quat.)	UPnP [8] (Quaternion)	Rotation Matrix	Quaternion	Cayley
# of unknowns	3	4	4	9	4	3
# of equations	3	4	8	27	7	6
max degree	3	3	3	3	4	4
# of solutions	27	40 (w/o 2-fold)	8 (w/o 2-fold)	40	80 (w/ 2-fold)	40
singularity	yes	no	no	no	no	yes
elim. templ.	120 × 120*	348 × 376	141 × 149	1936 × 1976	630 × 710	124 × 164
action matrix	27 × 27	40 × 40	8 × 8	40 × 40	80 × 80	40 × 40

*Hesch *et al.* use a Macaulay resultant method instead of Kukulova’s automatic generator.

4 Experiments on Synthetic Data

This section compares the proposed method using Cayley parameterization, referred to as **optDLS**, with existing methods, **SP+LHM** [16], **EPnP+GN** [16], **SDP** [16], **RPnP** [16], **DLS+++** [8], **OPnP** [16] and **UPnP** [8]. We had not evaluated Lu *et al.*’s iterative method, called **LHM** [16], since its accuracy is not as good as the other latest methods [8, 16]. All the methods are MATLAB implementations except for **UPnP**, a C++ implementation.

We evaluated the above methods in terms of robustness against varying the number of the points with fixed image noise, robustness against varying image noise with fixed the number of points, and computational time. In the two robustness tests, two point distributions, planar and non-planar, and two Cayley parameterizations, degenerate ($a = 0$) and non-degenerate ($a \neq 0$), were configured. Thus, the total number of the test is four for each.

We modified a MATLAB code² provided by Zheng *et al.* [16] for evaluating methods for the PnP problem. We assume a virtual perspective camera with image resolution 640×480 [pixels], focal length 800 [pixels], and principal point at the coordinate $(320, 240)$. 3D points were randomly distributed in the x -, y -, and z -range of $[-2, 2] \times [-2, 2] \times [4, 8]$ in the camera coordinates for the non-planar scene and $[-2, 2] \times [-2, 2] \times [0, 0]$ in the world coordinates for the planar scene. Those points were projected onto the virtual camera by the ground truth of \mathbf{R}_{true} and \mathbf{t}_{true} , which were randomly generated. Then, the rotation and translation errors were measured by

$$e_{rot} = \max_{k \in \{1,2,3\}} \cos(\mathbf{r}_k^T \mathbf{r}_{k,true})^{-1} \times 180/\pi \quad [\text{degrees}],$$

$$e_{trans} = \|\mathbf{t}_{true} - \mathbf{t}\| / \|\mathbf{t}\| \times 100 \quad [\%], \quad (20)$$

where $\mathbf{r}_{k,true}$ and \mathbf{r}_k are the k -th column of \mathbf{R}_{true} and \mathbf{R} , respectively. We ran 500 independent trials for each evaluation on Core-i7 3770 with 16GB RAM. We compared the errors and the elapsed time by taking median and mean, respectively.

In the implementation of **optDLS**, we used the one-time random rotation preprocessing to avoid the Cayley singularity. Also, inspired by ASPnP [16], we replaced `ref()` with `\` in the original code by the automatic generator so that the solver becomes faster.

4.1 Robustness against varying number of points

In this test, the number of the points n were varied from 4 to 20. The image noise was zero-mean Gaussian noise with fixed deviation $\sigma = 2$ [pixels] on the image points. Figure 2

²<https://sites.google.com/site/yinqiangzheng/>

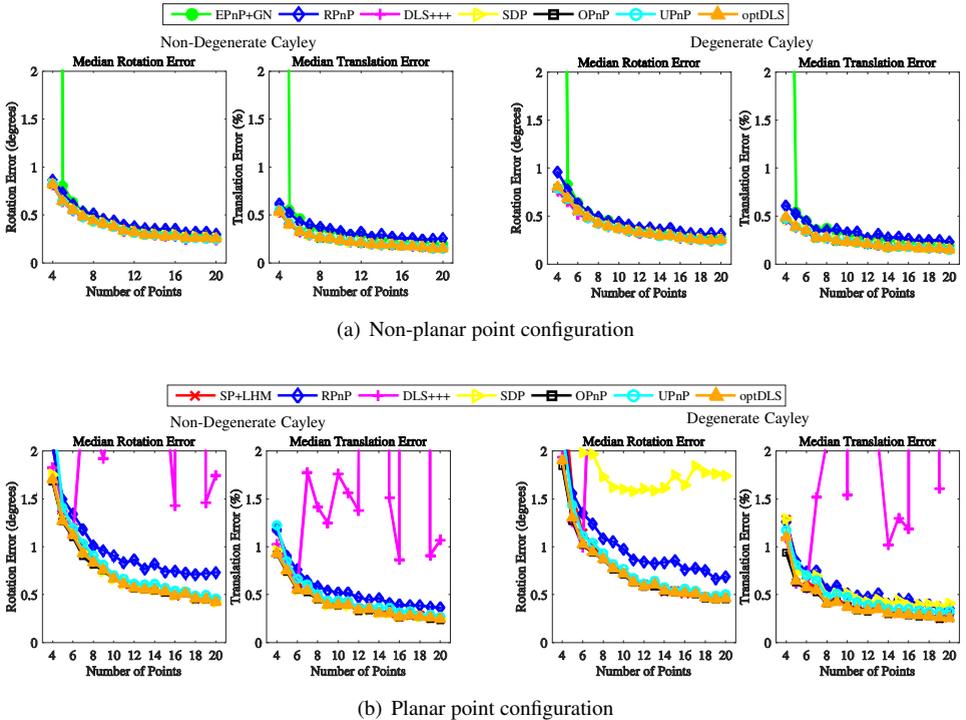


Figure 2: Robustness for varying $4 \leq n \leq 20$ and fixed image noise $\sigma = 2.0$

shows the result of the evaluation. Most of the methods are overlapped in the non-planar configuration since they give the optimal solution for $n \geq 6$ points. **DLS+++**, which uses a random rotation preprocessing three times, is not stable for planar scenes. On the contrary, the accuracy of **optDLS** is comparable to **OPnP** for both planar and non-planar configurations. Interestingly, **UPnP** is slightly worse than **optDLS** and **OPnP**. This will be discussed further in the next experiment.

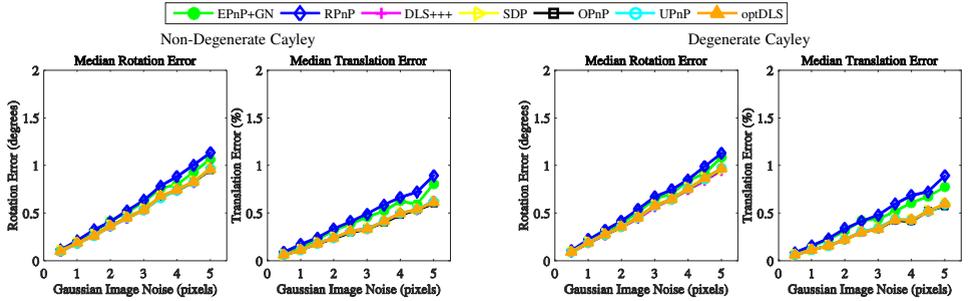
4.2 Robustness against image noise

Figure 3 shows the result with fixed $n = 10$ and varying $0.5 \leq \sigma \leq 5.0$ [pixels]. Similar to the previous experiment, **optDLS** and **OPnP** are comparable with each other. However, **UPnP** has almost the same accuracy with **RPnP**, which is worse than that of **optDLS** and **OPnP**. The results from the previous and this experiment imply that **UPnP** is not a optimal method but a kind of suboptimal method like **RPnP**.

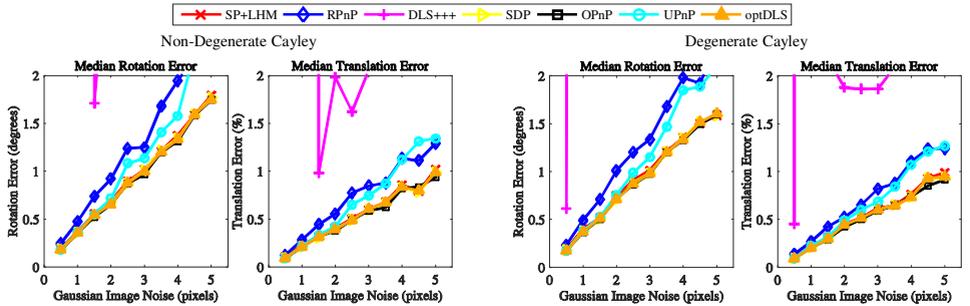
4.3 Computational Time

Figure 4 is a plot of computational time with varying $4 \leq n \leq 2000$ and fixed $\sigma = 2$. Note that the result of **SDP** is out of the range.

The computational time of **optDLS** is less than 3 msec for almost all cases and is the fastest especially for $n \geq 400$ points, even though **UPnP** is a mex function. This result indicates that **optDLS** is suitable for realtime applications, such as augmented reality and



(a) Non-planar point configuration



(b) Planar point configuration

Figure 3: Robustness for varying $0 \leq \sigma \leq 5.0$ and fixed $n = 10$.

visual SLAM, where $n \geq 400$ is not a rare situation. In addition, faster performance is expected for **optDLS** by a C++ implementation.

It is seen that **optDLS** and **OPnP** are $O(1)$ methods rather than $O(n)$. Due to the vectorization technique by [14], the dominant part is the decomposition of the elimination template and the action matrix. The reason why **optDLS** is faster than **OPnP** is that **optDLS** does not require a special implementation for eliminating 2-fold symmetry, which accounts for 75% of the computational time of **OPnP**.

5 Conclusion

This paper has presented a globally optimal DLS method for the PnP problem with Cayley parameterization. The contribution is to derive a new optimality condition without Lagrange multipliers, which can be applied for any rotational representations. Considering the size of solvers of several representations, we have found Cayley parameterization is the smallest one. As shown in the experiment, the proposed method is as accurate as the state-of-the-art method and the fastest for almost all the cases since it does not require a 2-fold symmetric elimination.

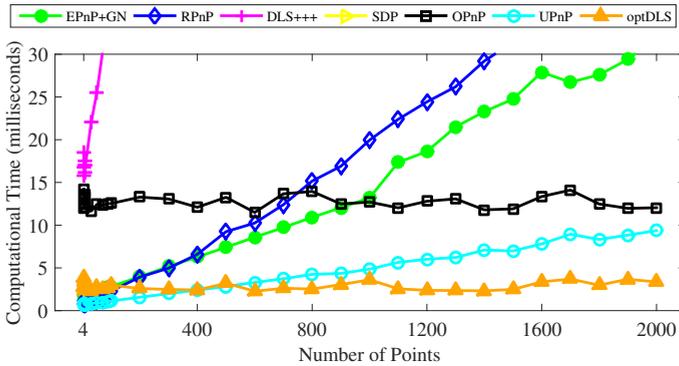


Figure 4: Computational time for varying $4 \leq n \leq 2000$ and fixed $\sigma = 2.0$

References

- [1] Erik Ask, Yubin Kuang, and Kalle Astrom. Exploiting p-fold symmetries for faster polynomial equation solving. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3232–3235. IEEE, 2012.
- [2] David A Cox, John Little, and Donal O’shea. *Using algebraic geometry*, volume 185. Springer Science & Business Media, 2006.
- [3] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [4] Bert M Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International journal of computer vision*, 13(3):331–356, 1994.
- [5] Joel A Hesch and Stergios I Roulmliotis. A direct least-squares (dls) method for pnp. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 383–390. IEEE, 2011.
- [6] Hatem Hmam and Jijoong Kim. Optimal non-iterative pose estimation via convex relaxation. *Image and Vision Computing*, 28(11):1515–1523, 2010.
- [7] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2969–2976. IEEE, 2011.
- [8] Laurent Kneip, Hongdong Li, and Yongduek Seo. Upnp: An optimal o (n) solution to the absolute pose problem with universal applicability. In *Computer Vision–ECCV 2014*, pages 127–142. Springer, 2014.
- [9] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Automatic generator of minimal problem solvers. In *Computer Vision–ECCV 2008*, pages 302–315. Springer, 2008.

- [10] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate $o(n)$ solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009.
- [11] S. Li, C. Xu, and M. Xie. A robust $o(n)$ solution to the perspective- n -point problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1444–1450, 2012.
- [12] Chien-Ping Lu, Gregory D Hager, and Eric Mjolsness. Fast and globally convergent pose estimation from video images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(6):610–622, 2000.
- [13] Gerald Schweighofer and Axel Pinz. Robust pose estimation from a planar target. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):2024–2030, 2006.
- [14] Gerald Schweighofer and Axel Pinz. Globally optimal $o(n)$ solution to the pnp problem for general camera models. In *BMVC*, pages 1–10, 2008.
- [15] Henrik Stewenius, Christopher Engels, and David Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, 2006.
- [16] Yinqiang Zheng, Yubin Kuang, Shigeki Sugimoto, Kalle Astrom, and Masatoshi Okutomi. Revisiting the pnp problem: A fast, general and optimal solution. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2344–2351. IEEE, 2013.
- [17] Yinqiang Zheng, Shigeki Sugimoto, and Masatoshi Okutomi. Aspnp: An accurate and scalable solution to the perspective- n -point problem. *IEICE TRANSACTIONS on Information and Systems*, 96(7):1525–1535, 2013.