The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-1/W1, 2017
ISPRS Hannover Workshop: HRIGI 17 – CMRT 17 – ISA 17 – EuroCOW 17, 6–9 June 2017, Hannover, Germany

RELATIVE CAMERA POSE ESTIMATION METHOD USING OPTIMIZATION ON THE MANIFOLD

Chuanqi. Cheng [a, *], Xiangyang. Hao [a], Jiansheng. Li [a, b]

[a] School of Navigation and Aerospace Engineering, Information Engineering University, Zhengzhou 450001, China
[b] Institute of Photogrammetry and GeoInformation, Leibniz Universität Hannover, Germany
(legend3q, xiangyanghao2004, ljszhx)@163.com

Commission VI, ICWG I/IV

KEY WORDS: Pose Estimation, Optimization, Manifold, Lie Group/Algebra, Levenberg-Marquardt

ABSTRACT:

To solve the problem of relative camera pose estimation, a method using optimization with respect to the manifold is proposed. Firstly from maximum-a-posteriori (MAP) model to nonlinear least squares (NLS) model, the general state estimation model using optimization is derived. Then the camera pose estimation model is applied to the general state estimation model, while the parameterization of rigid body transformation is represented by Lie group/algebra. The jacobian of point-pose model with respect to Lie group/algebra is derived in detail and thus the optimization model of rigid body transformation is established. Experimental results show that compared with the original algorithms, the approaches with optimization can obtain higher accuracy both in rotation and translation, while avoiding the singularity of Euler angle parameterization of rotation. Thus the proposed method can estimate relative camera pose with high accuracy and robustness.

1. INTRODUCTION

In general, there are three kinds of relative camera pose estimation models: 2D-2D, 3D-2D, and 3D-3D, whereas the kind of 3D-2D model is the most widely used in photogrammetry, incremental structure-from-motion (SFM), visual simultaneous localization and mapping (V-SLAM), augmented reality, autonomous navigation and so on. It can be described as that how to determine the orientation and position of a fully calibrated perspective camera, given n ($n \geqslant 3$) 3D points in the world framework and their corresponding 2D image points, which is also known as the perspective-n-point (PnP) problem (Hartley, Richard, 2003).

Considering the importance of PnP problem, a large amount of work has been done in the past few decades. The P3P problem attracts a lot of researchers' interests, such as (Li, 2011) and (Rieck, M. Q., 2014). Usually P3P solutions are implemented with RANSAC outlier rejection scheme. In practice, there are often more than 3 points and considering the redundancy can generally improve accuracy, most of recent works on PnP problem concentrate on the situations with more than 3 points. Roughly, the state-of-the-art solutions on PnP problem can be divided into two types – the multi-stage method and the direct minimization method.

Typically, the multi-stage methods first estimate the points coordinates in the camera framework, and transform the PnP (3D-2D) problem into 3D-3D pose estimation problem. Also, the linear methods usually have closed-form solutions. Latest progress in linear methods includes EPnP (Lepetit, V., 2009), RPnP (Li, S., 2012), OPnP (Zheng, Y., M, 2013). Lepetit, et al. (Lepetit, V., 2009) expresses the n 3D points as a weighted sum of four virtual control points, making the PnP problem reduce to estimate the coordinates of these control points in the camera referential, which reduces the complexity to $O(n)$. Li, et al. (Li, S., 2012) points out that due to underlying linearization scheme, EPnP performs poor for slightly redundant cases with $n = 4$ or $n = 5$. Then, Li, et al propose another non-iterative $O(n)$ solution which retrieves the optimum by solving a seventh order polynomial. Zheng, et al (Zheng, Y., M, 2013) put forward a non-iterative $O(n)$ solution which transforms the PnP problem to an unconstrained optimization problem solved by a Grobner basis solver. Moreover, the well-known direct linear transformation (DLT) is also a multi-stage method, because it first estimates the projection matrix and extracts the camera pose. However due to ignoring the orthogonal constraint of rotation matrix, its accuracy is poor.

The second type of PnP solutions is direct minimization methods. Its main idea is to minimize a defined energy function (or cost function), either in the image space or in the object space, which contains all nonlinear constraints. There exists some representative direct minimization methods. Lu et al. (Lu, C. P., 2000) propose an orthogonal iteration method to minimize the object space collinearity error, while Garro et al. (Garro, V., 2012) propose an alternative minimization method to minimize the 3D space geometric error. Hesch, et al. (Hesch, J. A., 2011) present a direct least squares (DLS) method for computing all solutions of the PnP problem by solving a system of three third-order polynomials. However, due to the Gayley representation of rotation, there are degeneration cases. Then they provide a remedy to conquer the degeneracy of the Gayley representation by solving DLS three times under different rotated 3D points, whereas the computational time is tripled.

To sum up, all the mentioned multi-stage methods are generally poor in accuracy, while the direct minimization methods suffer from the risk of getting trapped into local minimum. So in practise, we often firstly acquire an initial guess about the PnP

* Corresponding author

solution using multi-stage methods such as EPnP, DLT or RPnP, and then we use optimization method such as Gauss-Newton or Levenberg-Marquardt scheme to generate an optimal result.
However, in optimization, especially in computer vision and robotics, the correct treatment of angles consistently causes confusion. On one hand, a minimal parameterization is desired, but also singularities should be avoided. Interpolation of angles is not straightforward, since the group of rotation is only locally Euclidean. Probably the most elegant way to represent rigid body transformation is using a Lie group/algebra representation.

Thus we present an approach for relative camera pose estimation using optimization method with respect to the Lie group, which can avoid the singularity of Euler angle parameterization of rotation, and make the optimization method such as Gauss-Newton or Levenberg-Marquardt (Moré, J. J., 1978) more robust and convenient.

2. METHODS FOR STATE ESTIMATION USING OPTIMIZATION

In this section, we give a brief review of state estimation using optimization. This section defines the common notation and technology for the rest of the paper and introduces different types of optimization our method uses.

2.1 Maximum-a-posteriori (MAP) estimation and Least Square Problems

In general, we want to estimate a set of unknown variables $\mathbf{p}$ given a set of measurements $\mathbf{f}$, where we know the likelihood function $p(\mathbf{f}|\mathbf{p})$. We estimate $\mathbf{p}$ by computing the assignment of variables $\mathbf{p}^*$ that attains the maximum of the posterior $p(\mathbf{p}|\mathbf{f})$:

$$\mathbf{p}^* = \arg\max_{\mathbf{p}} p(\mathbf{p}|\mathbf{f}) = \arg\max_{\mathbf{p}} p(\mathbf{f}|\mathbf{p})\,p(\mathbf{p}) \quad (1)$$

In case no prior knowledge is available, $p(\mathbf{p})$ becomes a constant (uniform distribution) which is inconsequential and can be dropped. Then MAP estimation reduces to maximum likelihood estimation (MLE). Assuming that the measurements are independent, problem (1) factorizes into:

$$\mathbf{p}^* = \arg\max_{\mathbf{p}} p(\mathbf{p}|\mathbf{f}) = \arg\max_{\mathbf{p}} \prod_i p(\mathbf{f}_i|\mathbf{p}_i) \quad (2)$$

In order to write (2) in a more explicit but still widely applicable form, assume that the measurement noise is a zero-mean Gaussian noise with information matrix $\Sigma_{\mathbf{f}}^{-1}$. Then, the measurement likelihood in (2) becomes:

$$p(\mathbf{f}_i|\mathbf{p}_i) \propto \exp\left(-\frac{1}{2}\left\|\mathbf{f}_i - \hat{\mathbf{f}}_i(\mathbf{p}_i)\right\|^2_{\Sigma^{-1}_{\mathbf{f},i}}\right) \quad (3)$$

Since maximizing the posterior is the same as minimizing the negative log-posterior, or energy, the MAP estimate in (2) becomes:

$$\begin{aligned}\mathbf{p}^* &= \arg\min_{\mathbf{p}} \chi^2(\mathbf{p}) \\ &= \arg\min_{\mathbf{p}} -\log(p(\mathbf{f}|\mathbf{p})) \\ &= \arg\min_{\mathbf{p}} \sum_i \frac{1}{2}\left\|\mathbf{f}_i - \hat{\mathbf{f}}_i(\mathbf{p}_i)\right\|^2_{\Sigma^{-1}_{\mathbf{f},i}}\end{aligned} \quad (4)$$

which is a nonlinear least squares problem.

2.2 Optimization Methods

$\chi^2(\mathbf{p})$ is simply a sum of squares, and to minimize it is called nonlinear least squares optimization. A common technique for nonlinear least squares optimization is the Gauss-Newton (GN) method. The Gauss-Newton method performs iteratively, starting from a given initial guess $\mathbf{p}_0$ and updates by the rule:

$$\mathbf{p}_{(i+1)} = \mathbf{p}_{(i)} + \delta \quad (5)$$

where at each step the update vector δ is found by solving the normal equation:

$$(\mathbf{J}_{\mathbf{p}}^{\mathrm{T}}\Sigma_{\mathbf{f}}^{-1}\mathbf{J}_{\mathbf{p}})\delta = -\mathbf{J}_{\mathbf{p}}^{\mathrm{T}}\Sigma_{\mathbf{f}}^{-1}\mathbf{r} \quad (6)$$

Here, $\mathbf{J}_{\mathbf{p}} = \frac{\partial \mathbf{r}}{\partial \mathbf{p}}$ and $\mathbf{r} = \mathbf{f} - \hat{\mathbf{f}}(\mathbf{p})$ is the residual error.

A widely used optimization method is a variant of GN called Levenberg-Marquardt (LM), which alters the normal equation as follows:

$$(\mathbf{J}_{\mathbf{p}}^{\mathrm{T}}\Sigma_{\mathbf{f}}^{-1}\mathbf{J}_{\mathbf{p}} + \mu\,\mathrm{diag}(\Sigma_{\mathbf{f}}^{-1}))\delta = -\mathbf{J}_{\mathbf{p}}^{\mathrm{T}}\Sigma_{\mathbf{f}}^{-1}\mathbf{r} \quad (7)$$

The parameter μ rotates the update vector δ towards the direction of the steepest descent. Thus, if $\mu \to 0$, pure GN is performed, whereas if $\mu \to \infty$, gradient descent is used. In LM, the update step is performed only if it can significantly reduce the residual error. The parameter μ is self-adapted in the LM method.

3. RELATIVE CAMERA POSE ESTIMATION MODEL

3.1 The Camera Projection Function and Camera Poses

Points in the world $x_j \in \mathbf{R}^3$ are mapped to the camera image using the observation function:

$$\hat{z}(T_i, x_j) = \mathbf{proj}(K \cdot T_i \cdot x_j) \quad (8)$$

Here, the x_j is homogeneous point, T_i is the rigid body transformation which consists of the rotation matrix $\mathbf{R}$ and translation vector $\boldsymbol{t}$, and K is the camera calibration matrix (which we assume is known from prior calibration) and $\mathbf{proj}(\cdot)$ is the 3D-2D projection function:

$$\mathbf{proj}(\boldsymbol{a}) := \frac{1}{a_3}(a_1, a_2)^{\mathrm{T}},\ \forall \boldsymbol{a} \in \mathbf{R}^3 \quad (9)$$

The camera pose at a time-step i is represented as the rigid body transformation T_i.

3.2 Pose Estimation Model

Given a set of 3D points $x_j \in \boldsymbol{x}$ which are associated with 2D measurements z_{ij}, to estimate the camera pose a time-step i T_i, we minimize the following energy function using LM algorithm:

$$\chi^2(T_i) = \sum_j \frac{1}{2} \left\| z_j - \hat{z}(T_i, x_j) \right\|^2_{\Sigma_j^{-1}} \tag{10}$$

with respect to the rigid body transformation T_i. We use the Huber cost function as a robust kernel to guard against spurious matches.

3.3 Pose Optimization with respect to Lie Groups

The optimization methods presented in the previous section are applicable for scalar fields which are defined on Euclidean vector spaces $\mathbb{R}^n$. However, we want to minimize the re-projection error with respect to the rigid body transformation T_i, which includes the rotation $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$ in three dimensional space. $\boldsymbol{\omega}$ can be any parameterization of rotation in 3D (such as Euler angles or the rotation vector). Performing a rotation by δ and then by $\boldsymbol{\omega}$ is in general not equivalent to performing a rotation of $\boldsymbol{\omega} + \delta$. Vector addition is simply not the right operation to concatenate rotations. Thus, rotations cannot be modelled as Euclidean vector space, but as a Lie group.

3.3.1 Lie group and Lie algebra

A rigid body transformation in $\mathbb{R}^3$ can be expressed as an 4×4 matrix which can be applied to homogeneous position vectors (H. Strasdat., 2010):

$$T = \begin{bmatrix} \mathbf{R} & \boldsymbol{t} \\ 0 & 1 \end{bmatrix}, \text{ with } \mathbf{R} \in \mathrm{SO}(3), \boldsymbol{t} \in \mathbf{R}^3 \tag{11}$$

Here, $\mathrm{SO}(3)$ is the Lie group of rotation matrices. The rigid body transformations in $\mathbb{R}^3$ form a smooth manifold and therefore a Lie group, which is called the Special Euclidean Group ($\mathrm{SE}(3)$). The group operator is the matrix multiplication.

A minimal representation of this transformation is defined by the corresponding Lie algebra $\mathfrak{se}(3)$ which is the tangent space of $\mathrm{SE}(3)$ at the identity. In $\mathbb{R}^3$, the algebra elements are 6-vectors $(\boldsymbol{\omega}, \boldsymbol{v})^{\mathrm{T}}$: $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$ is the axis-angle representation for rotation, and $\boldsymbol{v}$ is a rotated version of the translation $\boldsymbol{t}$.

Elements of the $\mathfrak{se}(3)$ algebra can be mapped to the $\mathrm{SE}(3)$ group via the exponential mapping $\exp_{\mathrm{SE}(3)}$:

$$\exp_{\mathrm{SE}(3)}(\boldsymbol{\omega}, \boldsymbol{v}) := \begin{bmatrix} \exp_{\mathrm{SO}(3)}(\boldsymbol{\omega}) & \boldsymbol{V}\boldsymbol{v} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \boldsymbol{t} \\ 0 & 1 \end{bmatrix}. \tag{12}$$

Here,

$$\exp_{\mathrm{SO}(3)}(\boldsymbol{\omega}) = I + \frac{\sin(\theta)}{\theta}(\boldsymbol{\omega})_\times + \frac{1-\cos(\theta)}{\theta^2}(\boldsymbol{\omega})_\times^2. \tag{13}$$

Equation (13) is the Rodrigues' formula.

$$\boldsymbol{V} = I + \frac{1-\cos(\theta)}{\theta^2}(\boldsymbol{\omega})_\times + \frac{\theta - \sin(\theta)}{\theta^3}(\boldsymbol{\omega})_\times^2 \tag{14}$$

Here, $\theta = \left\| \boldsymbol{\omega} \right\|_2$, and $(\cdot)_\times$ is an operator which maps a 3-vector to its skew-symmetric matrix. Since $\exp_{\mathrm{SE}(3)}$ is surjective, there is also an inverse mapping $\log_{\mathrm{SE}(3)}$.

3.3.2 Pose-Point Transformation Jacobian

Using the chain rule, the partial derivative of the residual $\mathbf{r} = z - \hat{z}(T, x)$ with respect to T is:

$$\begin{aligned}
\frac{\partial \mathbf{r}}{\partial \delta} &= \frac{\partial (z - \hat{z}(T, x))}{\partial \delta} \\
&= -\frac{\partial \hat{z}(y)}{\partial y}\Big|_{y=\mathbf{R}\cdot x+t} \cdot \frac{\partial \mathbf{proj}(\exp_{\mathrm{SE}(3)}(\delta) T \cdot x)}{\partial \delta}\Big|_{\delta=0} \\
&= -\frac{\partial \hat{z}(y)}{\partial y}\Big|_{y=\mathbf{R}\cdot x+t} \cdot \frac{\partial \mathbf{proj}(\exp_{\mathrm{SE}(3)}(\delta) y)}{\partial \delta}\Big|_{\substack{\delta=0 \\ y=\mathbf{R}\cdot x+\boldsymbol{t}}} \\
&= -\frac{\partial \hat{z}(y)}{\partial y}\Big|_{y=\mathbf{R}\cdot x+t} \cdot \frac{\partial \mathbf{proj}(\mathbf{q})}{\partial \mathbf{q}}\Big|_{\mathbf{q}=y} \frac{\partial \exp_{\mathrm{SE}(3)}(\delta) y}{\partial \delta}\Big|_{\delta=0} \\
&= -\frac{f}{y_3} \begin{bmatrix} 1 & 0 & \frac{-y_1}{y_3} \\ 0 & 1 & \frac{-y_2}{y_3} \end{bmatrix} \cdot \begin{bmatrix} I_{3\times 3} & -(y)_\times \end{bmatrix}
\end{aligned} \tag{15}$$

We can get the update vector δ from (7) in the tangent space around identity $\mathfrak{se}(3)$ and mapped back onto the manifold $\mathrm{SE}(3)$, leading to a modified update step:

$$T_{i+1} = \exp_{\mathrm{SE}(3)}(\delta) \cdot T_i \tag{16}$$

Thus, we can use LM algorithm to solve the pose estimation problem.

4. EXPERIMENTAL RESLULTS

In this section, we experimentally investigate the LM algorithm for camera pose optimization on the manifolds, and compare the original state-of-the-art solutions to PnP problem with their corresponding optimization versions, including the well-known iterative approach by Lu et al. (Lu, C. P., 2000), denoted as LHM in short, the multi-stage method, put forward by Li et al. (Li, S., 2012), denoted as RPnP, and OPnP method proposed by Zheng (Zheng, Y., 2013). Their optimization versions are denoted as LHM+LM, RPnP+LM and OPnP+LM respectively. Also the direct minimization based method, DLS+++ (Hesch, J. A., 2011), is included.

The source codes of LHM, RPnP, OPnP and DLS++ are publicly available on the internet provided in (Zheng, Y., 2013). All the experiments are performed in MATLAB on a laptop with 2.4GHz CPU and 8GB RAM.

To acquire a quantitative analysis, all the experiments are implemented with simulated data. We generate a virtual perspective camera, and *n* 3D reference points in the camera

framework, which are randomly distributed in a specific range. All the simulated data parameter settings are below in Table 1.

Parameters	Settings
Focal length	800 pixels
Principle point	(320 pixels, 240 pixels)
Image solution	640 pixels × 480 pixels
3D point x-range	[-2, 2]
3D point y-range	[-2, 2]
3D point z-range	[4, 8]

Table 1. Parameter settings for simulated data

We rotate and translate the 3D points with the ground-truth transformation T_{true}, including the rotation $\mathbf{R}_{true}$ and translation $\boldsymbol{t}_{true}$. Then the absolute error in degrees between $\mathbf{R}_{true}$ and the estimated rotation $\mathbf{R}$ is measured. The rotation error is defined as:

$$err_{rot}(\deg ree) = \max_{k=1}^{3} a\cos(r_{true}^{k}, r^{k}) \cdot 180/\pi \quad (17)$$

where r_{true}^{k} and r^{k} are the k-th column of $\mathbf{R}_{true}$ and $\mathbf{R}$ respectively.

The translation error is the relative difference between $\boldsymbol{t}_{true}$ and the estimated $\boldsymbol{t}$, which is defined as:

$$err_{trans}(\%) = (\|\boldsymbol{t}_{true} - \boldsymbol{t}\| / \|\boldsymbol{t}\|) \cdot 100 \quad (18)$$

4.1 Varying Number of Points with Fixed Noise Level

Firstly, we vary the number of points n from 4 to 49, and add zero-mean Gaussian noise with fixed deviation (2 pixels) onto the projection images. At each n, 100 independent tests are performed. The average rotation and translation error are presented in Fig. 1 to Fig. 4.

Results show that all the solutions with LM algorithm optimized outperform their original versions, especially the accuracy of RPnP is improved effectively, which demonstrates the effectivity and efficiency of the proposed optimization strategy. We can find that the RPnP is not accurate enough, even in the presence of redundant correspondences, and the major reason lies in its underlying approximation schemes.

Also, we can find that compared with other methods, the LHM is not so accurate due to its possible local optimum. Moreover, with the number of points increasing, the accuracy of all solutions are all improved effectively.

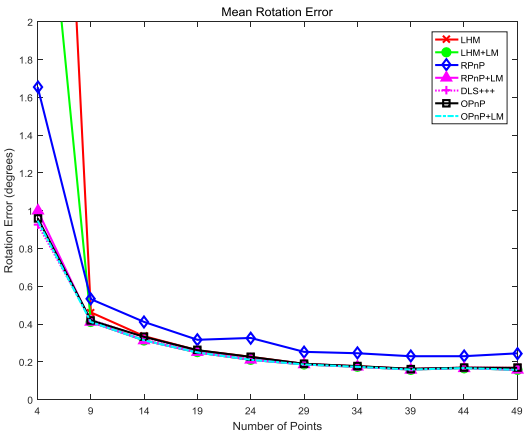

Figure 1. Mean rotation error w.r.t. varying number of points

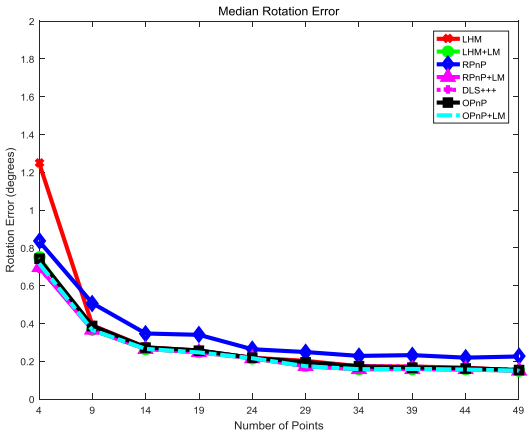

Figure 2. Median rotation error w.r.t. varying number of points

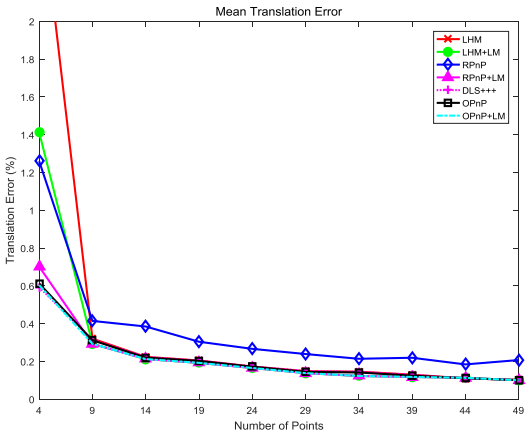

Figure 3. Mean translation error w.r.t. varying number of points

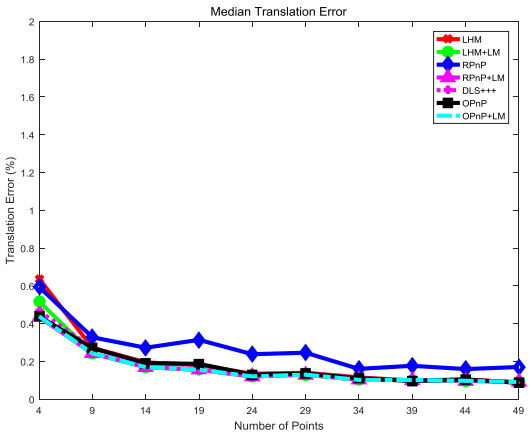

Figure 4. Median translation error w.r.t. varying number of points

4.2 Varying Noise Levels with Fixed Number of Points

Then, we fix the number of points n to be 10, and add zero-mean Gaussian noise with varying deviation levels (from 0.5 to 5 pixels) onto the projection images. At each noise level, 100 independent tests are performed and the average results are

reported. The average rotation and translation error are presented in Fig. 5 to Fig. 8.

As shown in Fig. 5 to Fig. 8, the proposed optimization strategy is efficient and effective, as the accuracy of all solutions is improved, especially the RPnP method. Also, we can find that with the noise increasing, the accuracy of all the method decreases.

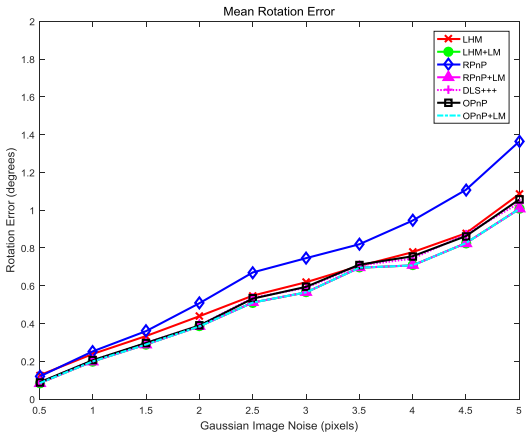

Figure 5. Mean rotation error w.r.t. varying noise levels

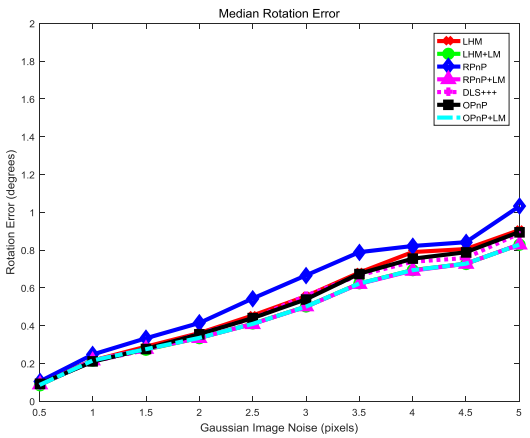

Figure 6. Median rotation error w.r.t. varying noise levels

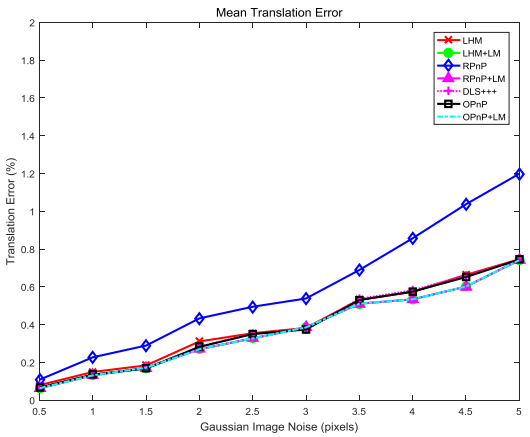

Figure 7. Mean translation error w.r.t. varying noise levels

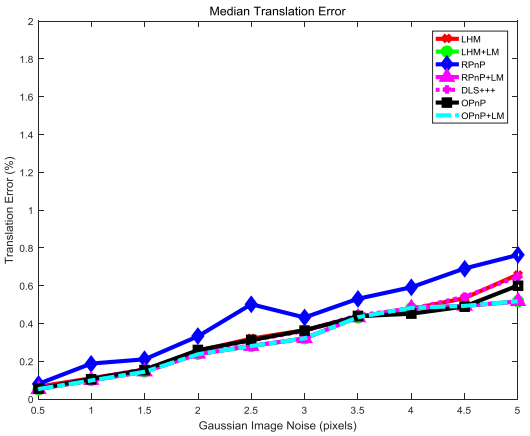

Figure 8. Median translation error w.r.t. varying noise levels

5. CONCLUSION

We propose an approach for relative camera pose estimation using optimization method with respect to the Lie group, which can avoid the singularity of Euler angle parameterization of rotation, and make the optimization method such as Gauss-Newton or Levenberg-Marquardt more robust and convenient. Experimental results show that the proposed approach outperform the original method without optimization which is capable of estimate relative camera pose with high accuracy, and it can be widely used in Photogrammetry .

REFERENCES

Garro, V., Crosilla, F., & Fusiello, A., 2012. Solving the pnp problem with anisotropic orthogonal procrustes analysis. *Proc. 3DIM/3DPVT*, pp. 262-269.

Hartley, Richard, 2003. *Multiple view geometry in computer vision*. Cambridge University Press, 2nd edtion.

Hesch, J. A., & Roumeliotis, S. I., 2011. A Direct Least-Squares (DLS) method for PnP. *IEEE International Conference on Computer Vision, ICCV 2011*, Barcelona, Spain,. pp.383-390. DBLP.

H. Strasdat, J. M. M. Montiel, and A. J. Davison. 2010. Scale drift-aware large scale monocular SLAM. *in Robotics: Science and Systems (RSS)*, Zaragoza, Spain.

Li, Xu, 2011. A stable direct solution of perspective-three-point problem. *International Journal of Pattern Recognition & Artificial Intelligence*, 25(5), pp. 627-642.

Li, S., Xu, C., & Xie, M, 2012. A robust o(n) solution to the perspective-n-point problem. *Pattern Analysis & Machine Intelligence IEEE Transactions on*, 34(7), pp. 1444-1450.

Lu, C. P., Hager, G. D., & Mjolsness, E., 2000. Fast and globally convergent pose estimation from video images. *IEEE TPAMI*, 22(6), pp. 610-622.

Lepetit, V., Moreno-Noguer, F., & Fua, P, 2009. Epnp: an accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2), pp. 155-166.

Moré, J. J. 1978. The levenberg-marquardt algorithm: implementation and theory. Lecture Notes in Mathematics, 630, pp. 105-116.

Rieck, M. Q. 2014. A fundamentally new view of the perspective three-point pose problem. *Journal of Mathematical Imaging and Vision*, 48(3), pp. 499-516.

Zheng, Y., Kuang, Y., Sugimoto, S., Åström, Kalle, & Okutomi, M, 2013. Revisiting the PnP Problem: A Fast, General and Optimal Solution. *IEEE International Conference on Computer Vision*. IEEE, pp.2344-2351.