LETTER *Special Section on Machine Vision and its Applications*

Speeding Up the Orthogonal Iteration Pose Estimation

Junying XIA[†], **Xiaoquan XU**[†], **Qi ZHANG**[†], *Nonmembers*, *and* **Jiulong XIONG**[†a)], *Student Member*

SUMMARY Existing pose estimation algorithms suffer from either low performance or heavy computation cost. In this letter, we present an approach to improve the attractive algorithm called Orthogonal Iteration. A new form of fundamental equations is derived which reduces the computation cost significantly. And paraperspective camera model is used instead of weak perspective camera model during initialization which improves the stability. Experiment results validate the accuracy and stability of the proposed algorithm and show that its computational complexity is favorably compare to the $O(n)$ non-iterative algorithm.

key words: *pose estimation, orthogonal iteration, paraperspective camera model*

1. Introduction

Pose estimation of a calibrated camera with respect to a reference frame is a basic and important problem in computer vision, which is widely used in visual navigation, robot localization, photogrammetry and other areas. Usually, it is also known as Perspective-n-Point (PnP) problem, where the objective is to estimate the position and orientation of the camera based on a set of correspondences between 3D reference points and their images.

Many solutions for this problem have been developed during the past decades, which can be divided into two categories: non-iterative and iterative algorithms. The traditional non-iterative algorithms adopt various methods that yield algebraic solutions, with the complexity varies between $O(n^2)$ [1] and $O(n^8)$ [2]. They are sensitive to additive noise and possible outliers. The classical iterative algorithms use nonlinear optimization algorithms to improve the accuracy of pose estimation. However, they rely on a good initial guess to converge to the correct solution and are time consuming. To overcome the problem of traditional algorithms, linear iterative algorithms have been presented which are both faster and more robust.

The Orthogonal Iteration (OI) algorithm proposed by Lu [3] is one of the fastest and most accurate among these iterative algorithms. But it is still not fast enough in real-time applications with larger number of points, besides, it sometimes converges to local minima as pointed out in [4]. Recently, Noguer [4] introduced the non-iterative Efficient Perspective-n-Point algorithm (EPnP) to compute the camera pose in $O(n)$ time. However, the pose estimation accuracy remains lower than linear iterative algorithms. Hatem [5] present a convex relaxation method that globally solves for the camera pose in $O(n)$ time. But their experiments show that the computing cost is lower than OI only when the number of points is larger than about 100.

In this letter, we propose an approach that improves both the speed and stability of OI. First, by reforming the equation for computing the optimal translation vector and other equations, the speed of the proposed approach is favorable compare to EPnP. Second, by introduce an initial approach based on paraperspective camera model, it is more stable than OI with a higher converging speed.

2. Orthogonal Iteration Pose Estimation

For making this letter more self-contained, we briefly explain OI algorithm in this section.

Assume the camera is calibrated, given a set of correspondences between reference points $\mathbf{p}_i$ $(i = 1, \ldots, n)$ in 3D space and their associated images, the image point $\mathbf{v}_i = (x_i, y_i, 1)^T$ which is the projection of $\mathbf{p}_i$ on the normalized image plane can be computed [3]. Then the objective of pose estimation can be achieved through minimizing the object space error as follows

$$\min_{R,t} \; E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{n} \|e_i\|^2 = \sum_{i=1}^{n} \|(\mathbf{I} - \mathbf{V}_i)(\mathbf{R}\mathbf{p}_i + \mathbf{t})\|^2 \quad (1)$$

where $\mathbf{R}$ is a rotation matrix, $\mathbf{t}$ represents a 3D translation vector, and $\mathbf{V}_i = (\mathbf{v}_i\mathbf{v}_i^T)/(\mathbf{v}_i^T\mathbf{v}_i)$ is the line-of-sight projection matrix of $\mathbf{v}_i$.

Since Eq. (1) is quadratic in $\mathbf{t}$, given a fixed $\mathbf{R}$, the optimal value for $\mathbf{t}$ can be computed in closed form as

$$\begin{aligned} \mathbf{t}(\mathbf{R}) &= -\left(\sum_{i=1}^{n}(\mathbf{I} - \mathbf{V}_i)\right)^{-1} \sum_{i=1}^{n}(\mathbf{I} - \mathbf{V}_i)\mathbf{R}\mathbf{p}_i \\ &\stackrel{def}{=} \mathbf{F}\sum_{i=1}^{n}(\mathbf{I} - \mathbf{V}_i)\mathbf{R}\mathbf{p}_i \end{aligned} \quad (2)$$

The initial value $\mathbf{R}^0$ is obtained uses a weak perspective approximation [3]. Then $\mathbf{R}$ is computed iteratively as follows: First, assume the kth estimate of $\mathbf{R}$ is $\mathbf{R}^k$, $\mathbf{t}^k = \mathbf{t}(\mathbf{R}^k)$ and $\mathbf{q}_i = \mathbf{V}_i(\mathbf{R}^k\mathbf{p}_i + \mathbf{t}^k)$. The next estimate $\mathbf{R}^{k+1}$ is determined by solving the following absolute orientation problem

$$\mathbf{R}^{k+1} = arg \; \min_R \sum_{i=1}^{n} \|(\mathbf{R}\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i\|^2 \quad (3)$$

Manuscript received November 7, 2011.
Manuscript revised January 16, 2012.

[†]The author are with the College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha, China.
a) E-mail: xiong@nudt.edu.cn

DOI: 10.1587/transinf.E95.D.1827

Define

$$\mathbf{M} = \sum_{i=1}^{n} \mathbf{q}_i' \mathbf{p}_i'^T = \sum_{i=1}^{n} (\mathbf{q}_i - \bar{\mathbf{q}})(\mathbf{p}_i - \bar{\mathbf{p}})^T \tag{4}$$

where $\bar{\mathbf{p}}$ and $\bar{\mathbf{q}}$ are the centroid of $\{\mathbf{p}_i\}$ and $\{\mathbf{q}_i\}$ respectively.

Let $\mathbf{U}^T\mathbf{M}\mathbf{V} = \mathbf{S}$ be a SVD of $\mathbf{M}$, the solution to Eq. (3) is $\mathbf{R}^{k+1} = \mathbf{V}\mathbf{U}^T$. When the data is severely corrupted, this may give a reflection with $det(\mathbf{R}_{k+1}) = -1$, which can be corrected use the method in [6]. Then the next estimate of translation is computed using Eq. (2), as $\mathbf{t}^{k+1} = \mathbf{t}(\mathbf{R}^{k+1})$, and the process is repeated.

3. Improvements of Orthogonal Iteration Algorithm

The mainly redundancy of OI is in the computer process to obtain $\mathbf{t}^k$,which need n matrix multiplications. By introducing Kronecker product ($\otimes$) and the *vec* operator [7], Eq. (2) is reformed as

$$\mathbf{t}(\mathbf{R}) = \mathbf{F} \sum_{i=1}^{n} (\mathbf{I} - \mathbf{V}_i)(\mathbf{p}_i^T \otimes \mathbf{I}) vec(\mathbf{R}) \stackrel{def}{=} \mathbf{G} vec(\mathbf{R}) \tag{5}$$

The value of $\mathbf{G}$ can be precomputed before the iterations, thus the computation of $\mathbf{t}^k$ is simplified to only one multiplication between a matrix and a vector.

Besides, there are another two equations can be reformed, which slightly cut down the overall computing cost: First, $\mathbf{q}_i$ can be effectively computed as

$$\mathbf{q}_i = \frac{\mathbf{v}_i \mathbf{v}_i^T}{\mathbf{v}_i^T \mathbf{v}_i} (\mathbf{R}^k \mathbf{p}_i + \mathbf{t}^k) = (\mathbf{v}_i^T (\mathbf{R}^k \mathbf{p}_i + \mathbf{t}^k)) \frac{\mathbf{v}_i}{\mathbf{v}_i^T \mathbf{v}_i} \tag{6}$$

Second, computation time cost by the calculation of $\mathbf{q}_i'$ can be totally omitted by using Eq. (7) instead of Eq. (4).

$$\begin{aligned} \mathbf{M} &= \sum_{i=1}^{n} \mathbf{q}_i (\mathbf{p}_i - \bar{\mathbf{p}})^T - \sum_{i=1}^{n} \bar{\mathbf{q}} (\mathbf{p}_i - \bar{\mathbf{p}})^T \\ &= \sum_{i=1}^{n} \mathbf{q}_i (\mathbf{p}_i - \bar{\mathbf{p}})^T = \sum_{i=1}^{n} \mathbf{q}_i \mathbf{p}_i'^T \end{aligned} \tag{7}$$

Whenever the 3D reference points are projected onto a small region on the side of the image, the weak perspective approximation used in the initial stage of OI is poor and may lead to instability [4] or a slowly converging process. However, in this situation, the paraperspective camera model can provide a good approximation given a proper choice of the origin [8]. So we use paraperspective instead of weak perspective to calculate the initial value of $\mathbf{R}$.

Select the closest image point to the centroid of $\{\mathbf{v}_i\}$ as the origin $\mathbf{v}_0 = (x_0, y_0, 1)^T$, and the corresponding reference point as the origin $\mathbf{p}_0$. The initial value of $\mathbf{R}$, $\mathbf{R}^0 = [\mathbf{i}\ \mathbf{j}\ \mathbf{k}]^T$, can be computed as follows [8]: First, solve the two over-constrained linear system of Eq. (8) to get an estimation of intermediate variables $\mathbf{I}_p$ and $\mathbf{J}_p$

$$\begin{cases} x_i - x_0 = \mathbf{I}_p \cdot (\mathbf{p}_i - \mathbf{p}_0) \\ y_i - y_0 = \mathbf{J}_p \cdot (\mathbf{p}_i - \mathbf{p}_0) \end{cases} \tag{8}$$

Then, the rows of $\mathbf{R}^0$ can be calculated as

$$\begin{cases} \mathbf{k} = (\mathbf{I} - t_z y_0 S(\mathbf{I}_p) + t_z x_0 S(\mathbf{J}_p))^{-1} t_z^2 (\mathbf{I}_p \times \mathbf{J}_p) \\ \mathbf{i} = t_z \mathbf{I}_p + x_0 \mathbf{k} \\ \mathbf{j} = t_z \mathbf{J}_p + y_0 \mathbf{k} \end{cases} \tag{9}$$

where

$$t_z = \frac{1}{2} \left(\frac{\sqrt{1 + x_0^2}}{\|\mathbf{I}_p\|} + \frac{\sqrt{1 + y_0^2}}{\|\mathbf{J}_p\|} \right) \tag{10}$$

4. Experimental Results

In this letter we consider a comparative performance analysis of: the OI algorithm of [3], denoted as LHM, the EPnP algorithm of [4] denoted as EPnP, and the proposed algorithm denoted as SUOI. All algorithms are carried out on PC with Pentium 2.4 GHz CPU, Matlab2010b environment. Programs for the LHM and EPnP are obtained from their respective web sources. The LHM algorithm is enhanced with the method in [6], and the EPnP algorithm is always executed with the Gauss-Newton fine-tuning enabled. As our algorithm is coded in a vectorizing form, it is inherently more efficient than LHM and EPnP which use loops. To make the comparison fair, the loops of these two algorithms are vectorized.

We produced synthetic 2D image points in a 640×480 image acquired using a virtual calibrated camera with an effective focal length of $f_u = f_v = 750$ and a principal point at $(u_c, v_c) = (320, 240)$. The 3D reference points are selected using a uniform distribution within a cubic box and then transformed to a randomly generated pose. The images points must also lie within the field of view, therefore any pose that makes any points project outside the image plane is discarded. Each projected image point is then subjected to Gaussian noise of varying standard deviations and then quantized by rounding its coordinate to the nearest integer.

Given the true camera rotation $\mathbf{R}_{true}$ and translation $\mathbf{t}_{true}$, the error metric of the estimated rotation $\mathbf{R}$ is defined by $E_{rot} = 2 \arccos(q_0)$, where $q_0 = 0.5\sqrt{1 + tr(\mathbf{R}\mathbf{R}_{true}^T)}$ [5], and the relative error of the estimated translation $\mathbf{t}$ is determined by $E_{trans} = \|\mathbf{t}_{true} - \mathbf{t}\|/|\mathbf{t}\|$.

We first analyze the speed of the three algorithms by recording the average computation time of 1000 runs. In fact, the speed comparison in [4] may be unfair to LHM, because the original LHM codes suffer greatly from the poor coding form, while the original EPnP codes do not have so much trouble due to its simplicity. However, as shown in the left side of Fig. 1, the computational cost of LHM remains much higher than EPnP after vectorization. The proposed method is even a little more fast than EPnP, which is known to be fast, when the number of reference points is small than about 200. When the number of reference points becomes even larger, EPnP gradually shows better efficiency. For further comparison with LHM, we plot the average iteration numbers of LHM and our method in the right side of

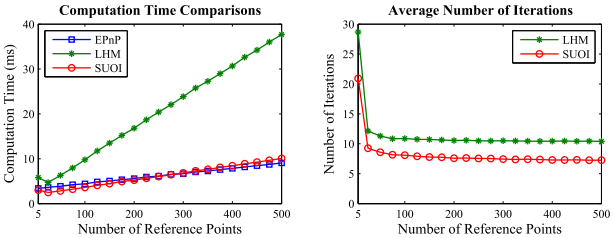

Fig. 1 Average computation times and number of iterations for experiments with different number of points under a fixed noise level of 5 pixels.

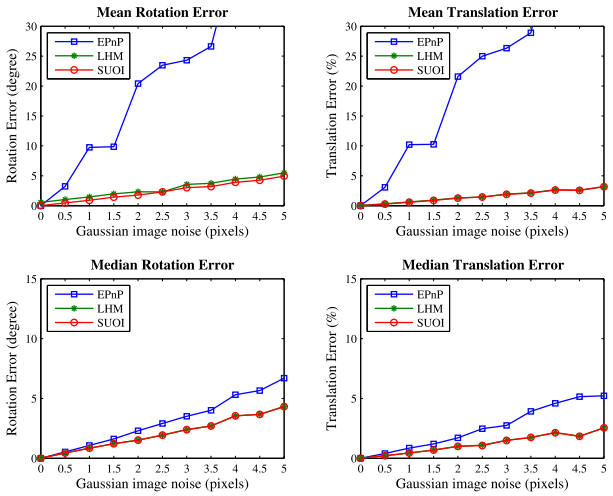

Fig. 2 Mean and median rotation and translation errors for experiments of a fixed six-point configuration under different noise levels.

Fig. 1. Thanks to the paraperspective based initial method, it is clearly that our method needs fewer iterations to converge to the optimal pose than LHM.

To compare the accuracy and stability of the three algorithms, experiments were performed under 11 different noise levels from 0 to 5 pixels, and all the plots discussed here were created by running 300 independent simulations. Figure 2 displays the pose estimation accuracy for a fixed six-point configuration. From Fig. 2 the EPnP performance accuracy is clearly poor even for pixel errors as small as 0.5, while LHM and our method give a favorably similar accuracy. Actually, the trouble that LHM may produce negative $det(\mathbf{R})$ has been corrected well by the method in [6], while EPnP sometimes gives wrong result with negative t_z, especially when the point number is small. This may be the main reason that the mean errors of EPnP are so glaring compared to LHM. The mean rotation errors of our method is small than LHM with almost equivalent median rotation errors, which validates the instability of LHM. Figure 3 displays the pose estimation accuracy for configurations with different point numbers under a fixed noise level of 5 pixels. As point number grows, the performances of all the three methods become better, especially EPnP. One thing should be noticed is that, the advantage of our method compare to LHM exhibits only when point number is very small.

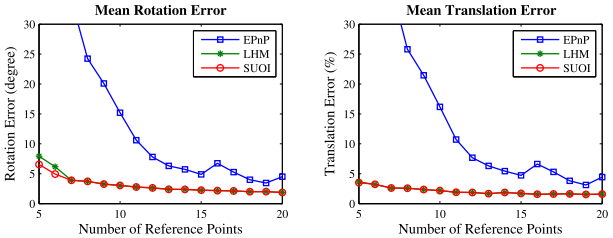

Fig. 3 Mean rotation and translation errors for experiments with different number of points under a fixed noise level of 5 pixels.

5. Conclusion

In this letter, we proposed to improve the performance of the Orthogonal Iteration method by reforming its equations and adopting a different initialization method. Through the experimental results with synthetic camera images, we verified that the proposed method gives as accuracy but more stable results than OI method, and its computation cost is even less than the most recent $O(n)$ non-iterative EPnP method with a moderate point set. This approach is well-suited for vision applications where both efficiency and accuracy are desired.

This algorithm can also be easily extended to line-based Orthogonal Iteration pose estimation problems [9]. Currently, we are trying to extend the proposed method to the planar case and analyze the performance of it in more detail.

References

[1] L. Quan and Z. Lan, "Linear N-point camera pose determination," IEEE Trans. Pattern Anal. Mach. Intell., vol.21, no.7, pp.774–780, 1999.

[2] A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," IEEE Trans. Pattern Anal. Mach. Intell., vol.24, no.5, pp.578–589, 2003.

[3] C. Lu, G.D. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," IEEE Trans. Pattern Anal. Mach. Intell., vol.22, no.6, pp.610–622, 2000.

[4] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," Int. J. Comput. Vis., vol.81, no.2, pp.155–166, 2009.

[5] H. Hmam and J. Kim, "Optimal non-iterative pose estimation via convex relaxation," Int. J. Comput. Vis., vol.28, pp.1515–1523, 2010.

[6] Z.Y. Zhang, D.Y. Zhu, and J. Zhang, "An improved pose estimation algorithm for real-time vision applications," 2006 International Conference on Communications, Circuits and Systems, pp.402–406, June 2006.

[7] J.R. Magnus and H. Neudecker, Matrix differential calculus with applications in statistics and econometrics, 3rd ed., pp.31–35, John Wiley & Sons, New York, 2007.

[8] R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy, "Object pose: The link between weak perspective, paraperspective, and full perspective," Int. J. Comput. Vis., vol.22, no.2, pp.173–189, 1997.

[9] X.H. Zhang, K.P. Wang, Z. Zhang, and Q.F. Yu, "A new line-based orthogonal iteration pose estimation algorithm," 2009 International Conference on Information Engineering and Computer Science, pp.1–4, Dec. 2009.