Hindawi Publishing Corporation
Journal of Robotics
Volume 2013, Article ID 692838, 7 pages
http://dx.doi.org/10.1155/2013/692838

Research Article

Computationally Efficient Iterative Pose Estimation for Space Robot Based on Vision

Xiang Wu and Ning Wu

Shenzhen Graduate School, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China

Correspondence should be addressed to Ning Wu; aning.wu@gmail.com

Received 21 May 2012; Accepted 19 December 2012

Academic Editor: Farhad Aghili

In postestimation problem for space robot, photogrammetry has been used to determine the relative pose between an object and a camera. The calculation of the projection from two-dimensional measured data to three-dimensional models is of utmost importance in this vision-based estimation however, this process is usually time consuming, especially in the outer space environment with limited performance of hardware. This paper proposes a computationally efficient iterative algorithm for pose estimation based on vision technology. In this method, an error function is designed to estimate the object-space collinearity error, and the error is minimized iteratively for rotation matrix based on the absolute orientation information. Experimental result shows that this approach achieves comparable accuracy with the SVD-based methods; however, the computational time has been greatly reduced due to the use of the absolute orientation method.

1. Introduction

Vision based methods have been applied to estimate the pose of space robot since 1990s. In these methods, the relative position and orientation between a camera and a robot target are determined with a set of n feature points expressed in the three dimensional (3D) object coordinates and their two dimensional (2D) projection in the camera coordinate. The error in position and orientation is usually optimized using the noniterative or iterative algorithms. The noniterative algorithms give an analytical solution for the optimization [1–3], and a typical example of these algorithms includes the method to represent feature points as a linear combination of four virtual control points based on their coordinates [4]. The noniterative methods are generally less time consuming than the iterative methods with acceptable accuracy; however, they are sensitive to observation noise such as image noise, different lighting conditions, and even occlusion by outliers. The iterative approaches, however, achieve better accuracy than the noniterative methods by solving the rotation matrix with a nonlinear least-square method iteratively.

A typical iterative method is the Levenberg-Marquardt (L-M) algorithm [5–7], and it has been widely used and accepted as a standard algorithm for least-square problem in photogrammetry. The L-M method is in essentially the combination of the steepest descent method and the Gauss-Newton method in different optimization stages. The steepest descent method is used at the early stage of optimization when the current value of error is still far from the minimum, while the Gauss-Newton method is used at the later stage of optimization when the solution is relatively close to the target. The combination of the two methods at different stages is in fact a coarse search from a globalwise followed by a fine search within a local area. The use of the steepest descent method in the early stage of optimization helps to find a guaranteed convergence direction and locate the solution within a small area, while the Gauss-Newton approach finds the optimized solution with fast speed. The L-M algorithm offers a way to find an optimized solution for the iterative approach; however, since the L-M method is a general-purposed optimization method, it can be improved significantly to suit the specific requirement of the pose estimation for a faster converging speed and a better noise-tolerant solution.

The iterative optimization method specially designed for pose estimation purpose has been considered based on the target pose and the depths of the feature points [8]. This method calculates the depth information and absolute orientation of target, respectively, and the perspective nonlinearity can be reduced by introducing the depth variable. However, this method requires hundreds of iterations before it can reach a convergence point [8].

Another type of iterative algorithm, orthogonal iterative (OI), was proposed by Lu et al. [9] to estimate the object-space collinearity error with a different objective function. Instead of using the depth of the object, this algorithm uses scene points to improve the calculation of the translation vector, and it achieves a higher accuracy and more computation efficiency. However, the corruption in the input data can cause a considerable error in the rotation matrix, and thus the accuracy of the OI method is affected [10]. The OI algorithm was further developed by Zhang et al. by introducing depth update in the computation of the translation vector in a two-stage iterative process [11, 12]. Higher accuracy than the OI algorithm can be achieved from this method by refining the error of absolute orientation [10].

The essential process of the above optimization algorithms specially designed for pose estimation is to solve the absolute orientation problem, which can be applied with quaternions [13–15] and Singular Value Decomposition (SVD) [15, 16] methods, respectively. The SVD method has achieved a great performance and has been used extensively because of its closed form solution and enhanced orthogonality; however, the computational load makes it difficult to implement in real-time system.

In order to overcome this problem, this paper introduces the FOAM method [17] to calculate the absolute orientation for pose estimation. The experimental result shows that the performance of accuracy and noise resistance will be shown to be comparable with the SVD method; however, the computational efficiency is considerably better. The structure of this paper has been organized as follows. Section 2 of this paper further introduces the problem, and Section 3 presents our solution for the problem. Section 4 shows the experimental result, and finally a conclusion is drawn.

2. Theory

In pose estimation of space robot, we usually have a target coordinate frame $OXYZ$ and a camera coordinate frame $O'X'Y'Z'$, and they are defined as illustrated in Figure 1, respectively.

It can be seen that the center of the projection from the object is at the origin O' and the optical axis points to the positive Z' axis. Supposed that a lens with the focal length of f is located at the origin, the plane $Z' = f$ is then considered as the image plane of the camera on which the feature points are projected. If the coordinates of feature points, P_i $(i = 1, \ldots, n)$ on the target are denoted as $\mathbf{l}_i = (x_i, y_i, z_i)^T$ in the target coordinate frame, and its corresponding projection on the camera axis is expressed as $\mathbf{m}_i = (x_i', y_i', z_i')^T$, then the rotation matrix and the translation vector from the target coordinate frame to the camera coordinate frame can be written in the relationship such as

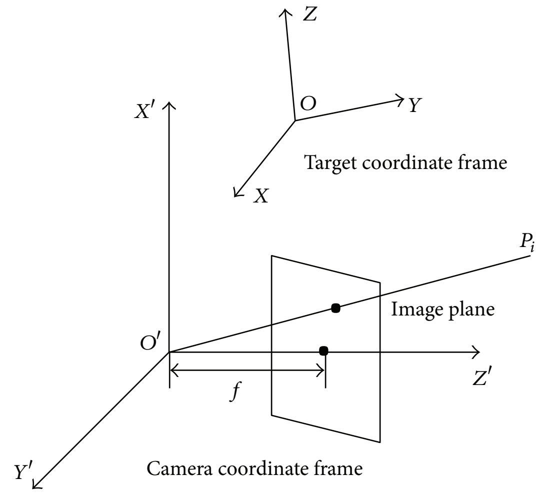

Figure 1: Coordinate frames in the pose estimation problem.

$$\mathbf{m}_i = A\mathbf{l}_i + \mathbf{t}, \tag{1}$$

where $A = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)^T$ denotes the rotation matrix and $\mathbf{t} = (t_x, t_y, t_z)^T$ is the translation vector.

If the image point P_i' in camera axis frame represents the projection from the feature point P_i, and its coordinate is written as $\mathbf{w}_i = (u_i, v_i, f)^T$, according to the idealized pinhole camera model, the relationship between the coordinates of P_i^T and P_i can be expressed as

$$\mathbf{w}_i = \frac{f}{\mathbf{a}_3^T \mathbf{l}_i + t_z}\mathbf{m}_i = \frac{f}{\mathbf{a}_3^T \mathbf{l}_i + t_z}\left(A\mathbf{l}_i + \mathbf{t}\right), \tag{2}$$

and $\mathbf{w}_i$ in (2) is regarded as a collinearity equation in image space, and the orthogonal projection of $\mathbf{m}_i$ to $\mathbf{w}_i$ can be written as

$$A\mathbf{l}_i + \mathbf{t} = W_i\left(A\mathbf{l}_i + \mathbf{t}\right), \tag{3}$$

where

$$W_i = \frac{\mathbf{w}_i \mathbf{w}_i^T}{\mathbf{w}_i^T \mathbf{w}_i} \tag{4}$$

denotes the projection matrix to the vector $\mathbf{w}_i$.

The object-space collinearity error for each feature point is then formulated for optimization, such as

$$e_i = \mathbf{m}_i - \widehat{W}_i \mathbf{m}_i = \left(A\mathbf{l}_i + \mathbf{t}\right) - \widehat{W}_i\left(A\mathbf{l}_i + \mathbf{t}\right), \tag{5}$$

where

$$\widehat{W}_i = \frac{\widehat{\mathbf{w}_i \mathbf{w}_i^T}}{\widehat{\mathbf{w}_i^T}\widehat{\mathbf{w}}_i} \tag{6}$$

represents the observed projection matrix to vector $\widehat{\mathbf{w}}_i$, and $\widehat{\mathbf{w}}_i$ denotes the observation of $\mathbf{w}_i$. Considering each of the feature points, the following objective function is defined:

$$\begin{aligned} E(A,\mathbf{t}) &= \sum_{i=1}^{n} \|e_i\|^2 \\ &= \sum_{i=1}^{n} \left\| (A\mathbf{l}_i + \mathbf{t}) - \widehat{W}_i (A\mathbf{l}_i + \mathbf{t}) \right\|^2. \end{aligned} \tag{7}$$

The objective function in (7) can be minimized by finding a suitable A and $\mathbf{t}$.

Since the objective function in (7) is the second norm of both the translation vector $\mathbf{t}$ and the rotation matrix A, the error of the absolute orientation can be found by taking the derivative of (7) with respect to $\mathbf{t}$ and making it equal to zero, such as

$$\frac{\partial E(A,\mathbf{t})}{\partial \mathbf{t}} = 0 \Longrightarrow n\left(I - \frac{1}{n}\sum_{i=1}^{n} \widehat{W}_i \right)\mathbf{t} = \sum_{i=1}^{n} \left(\widehat{W}_i - I \right) A\mathbf{l}_i. \tag{8}$$

The left side of (7) can be rewritten as,

$$\begin{aligned} \mathbf{x}^T \left(I - \frac{1}{n}\sum_{i=1}^{n} \widehat{W}_i \right)\mathbf{x} &= \frac{1}{n}\sum_{i=1}^{n} \left(\|\mathbf{x}\|^2 - \mathbf{x}^T \widehat{W}_i \mathbf{x} \right) \\ &= \frac{1}{n}\sum_{i=1}^{n} \left(\|\mathbf{x}\|^2 - \mathbf{x}^T \widehat{W}_i^T \widehat{W}_i \mathbf{x} \right) \\ &= \frac{1}{n}\sum_{i=1}^{n} \left(\|\mathbf{x}\|^2 - \left\| \widehat{W}_i \mathbf{x} \right\|^2 \right). \end{aligned} \tag{9}$$

Since $\widehat{W}_i\mathbf{x}$ is the projection of $\mathbf{t}$ on $\widehat{\mathbf{w}}_i$, and in this case $\mathbf{t}$ cannot project on itself, we have $\|\mathbf{x}\| \geq \|\widehat{W}_i\mathbf{x}\|$, or $\|\mathbf{x}\| - \|\widehat{W}_i\mathbf{x}\| \geq 0$; therefore, the term $n(I - (1/n)\sum_{i=1}^{n} \widehat{W}_i)$ in (8) is nonsingular, and the optimal position of $\mathbf{t}$ can be written as a function of the rotation matrix A such as

$$\mathbf{t}(A) = \frac{1}{n}\left(I - \frac{1}{n}\sum_{i=1}^{n} \widehat{W}_i \right)^{-1} \sum_{i=1}^{n} \left(\widehat{W}_i - I \right) A\mathbf{l}_i. \tag{10}$$

If $\widehat{\mathbf{m}}_i(A) = \widehat{W}_i(A\mathbf{l}_i + \mathbf{t}(A))$, then (7) can be rewritten as a function of A, such that

$$E(A) = \sum_{i=1}^{n} \|A\mathbf{l}_i + \mathbf{t}(A) - \widehat{\mathbf{m}}_i(A)\|^2. \tag{11}$$

It can be seen from (11) that the object-space collinearity error function can be minimized with respect to the rotation matrix A only.

In the same nonlinear least-square format as shown in (11), another objective function can be formulated in order to find the optimal orientation matrix A and translation vector $\mathbf{t}$, such that

$$L(A,\mathbf{t}) = \sum_{i=1}^{n} \|\mathbf{m}_i - (A\mathbf{l}_i + \mathbf{t})\|^2. \tag{12}$$

If the mean values of the feature points in object coordinate and camera coordinate are calculated as $\bar{\mathbf{l}} = (1/n)\sum_{i=1}^{n} \mathbf{l}_i$ and $\overline{\mathbf{m}} = (1/n)\sum_{i=1}^{n} \mathbf{m}_i$ respectively, (12) can be rewritten in the form of,

$$\begin{aligned} L(A,\mathbf{t}) &= \sum_{i=1}^{n} \|\mathbf{m}_i - (A\mathbf{m}_i + \mathbf{t})\|^2 \\ &= \sum_{i=1}^{n} \left\| \mathbf{m}_i - \overline{\mathbf{m}} - A\left(\mathbf{l}_i - \bar{\mathbf{l}} \right) + \left(\overline{\mathbf{m}} - A\bar{\mathbf{l}} - \mathbf{t} \right) \right\|^2 \\ &= \sum_{i=1}^{n} \left\| \mathbf{m}_i' - A\mathbf{l}_i' \right\|^2 + n\left\| \overline{\mathbf{m}} - A\bar{\mathbf{l}} - \mathbf{t} \right\|^2, \end{aligned} \tag{13}$$

where $\mathbf{l}_i' = \mathbf{l}_i - \bar{\mathbf{l}}$ and $\mathbf{m}_i' = \mathbf{m}_i - \overline{\mathbf{m}}$. Since the second term of (13) can be set to zero by assigning $\mathbf{t} = \overline{\mathbf{m}} - A\bar{\mathbf{l}}$, the objective function $L(A,\mathbf{t})$ can be optimized by minimizing the first term, such that

$$\begin{aligned} G(A) &= \sum_{i=1}^{n} \left\| \mathbf{m}_i' - A\mathbf{l}_i' \right\|^2 \\ &= \sum_{i=1}^{n} \left\| \mathbf{m}_i' \right\|^2 + \sum_{i=1}^{n} \left\| A\mathbf{l}_i' \right\|^2 - 2\,\mathrm{tr}\left(A^T \sum_{i=1}^{n} \mathbf{m}_i' \mathbf{l}_i'^T \right) \\ &= \sum_{i=1}^{n} \left\| \mathbf{m}_i' \right\|^2 + \sum_{i=1}^{n} \left\| A\mathbf{l}_i' \right\|^2 - 2\,\mathrm{tr}\left(A^T B \right), \end{aligned} \tag{14}$$

where $B = \sum_{i=1}^{n} \mathbf{m}_i' \mathbf{l}_i'^t$. It is obvious that $G(A)$ in (14) can be minimized with a maximum value of $\mathrm{tr}(A^T B)$, and it can be found by using the SVD method. In traditional SVD method, B can be decomposed as (U, Σ, V), where U and V are orthogonal to each other, and $\Sigma = \mathrm{diag}[\sigma_1, \sigma_2, \sigma_3]$ ($\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$ are the singular values of B), such that

$$B = U' S V'^T, \tag{15}$$

where $S = \mathrm{diag}[s_1, s_2, s_3]$, ($s_1 = \sigma_1; s_2 = \sigma_2; s_3 = (\det U)(\det V)\sigma_3$), $U' = U\,\mathrm{diag}[1, 1, \det U]$, and $V' = V\,\mathrm{diag}[1, 1, \det V]$. Therefore, we have

$$\begin{aligned} \mathrm{tr}\left(A^T B \right) &= \mathrm{tr}\left(A' U' S V'^T \right) \\ &= \mathrm{tr}\left(S U'^T A V' \right). \end{aligned} \tag{16}$$

It can be seen that the maximum of $\mathrm{tr}(A^T B)$ in (16) is $\mathrm{tr}(S)$ with $U'^T A V' = E$, where E is an identity matrix. Therefore, the optimal A can be obtained by using $A_{\mathrm{opt}} = U' V'^T$. However, the computation of the matrices U' and V' is too time consuming and this method cannot be implemented for real-time applications.

It is noted that the calculation of the singular values of B can be replaced by the combination of $\det(B)$, $\|B\|$ and $\mathrm{adj}(B')$,

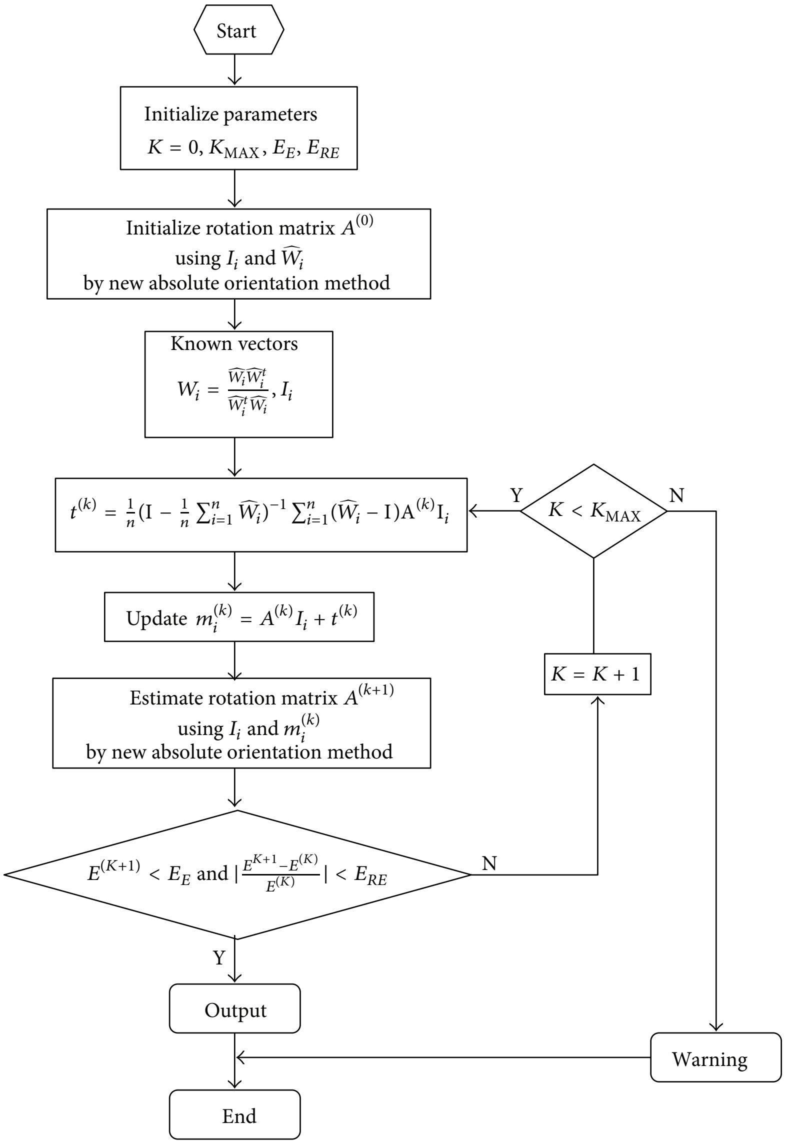
Start
Initialize parameters
$K = 0, K_{\text{MAX}}, E_E, E_{RE}$
Initialize rotation matrix $A^{(0)}$
using I_i and $\widehat{W}_i$
by new absolute orientation method
Known vectors
$W_i = \frac{\widehat{W}_i\widehat{W}_i^t}{\widehat{W}_i^t\widehat{W}_i}, I_i$
$t^{(k)} = \frac{1}{n}(\mathrm{I} - \frac{1}{n}\sum_{i=1}^{n}\widehat{W}_i)^{-1}\sum_{i=1}^{n}(\widehat{W}_i - \mathrm{I})\mathrm{A}^{(k)}\mathrm{I}_i$
Y
$K < K_{\text{MAX}}$
N
Update $m_i^{(k)} = A^{(k)}I_i + t^{(k)}$
$K = K + 1$
Estimate rotation matrix $A^{(k+1)}$
using I_i and $m_i^{(k)}$
by new absolute orientation method
$E^{(K+1)} < E_E$ and $|\frac{E^{K+1} - E^{(K)}}{E^{(K)}}| < E_{RE}$
N
Y
Output
Warning
End

Figure 2: Flow chart of the algorithm.

and therefore A_{opt} can be calculated without the need for the SVD operation, such that

$$\begin{aligned}
\det(B) &= s_1 s_2 s_3, \\
\|B\|^2 &= s_1^2 + s_2^2 + s_3^2, \\
\operatorname{adj}\left(B'\right) &= U' \operatorname{diag}\left[s_2 s_3, s_1 s_3, s_1 s_2\right] V'^T, \\
BB^T B &= U' \operatorname{diag}\left[s_1^2, s_2^2, s_3^2\right] V'^T.
\end{aligned} \tag{17}$$

If we let

$$\lambda = \operatorname{tr}\left(A_{\text{opt}}^T B\right) = s_1 + s_2 + s_3, \tag{18}$$

then the following equation can be formed in order to obtain A_{opt}, such that

$$\left(\lambda^2 - \|B\|^2\right)^2 - 8\lambda \det(B) - 4\|\operatorname{adj}(B)\|^2 = 0, \tag{19}$$

It is noted that (18) is one of the four solutions and the largest root of (19). Therefore, A_{opt} can be derived from (19) such that

$$A_{\text{opt}} = \frac{\left(\alpha + \|B\|^2\right)B + \lambda \operatorname{adj}\left(B'\right) - BB^T B}{\alpha\lambda - \det(B)}, \tag{20}$$

where $\alpha = (1/2)(\lambda^2 - \|B\|^2)$. With the optimal A being found, the minimum $L(A, \mathbf{t})$ in (13) can be calculated as,

$$L_{\min} = \sum_{i=1}^{n}\|\mathbf{m}_i\|^2 + \sum_{i=1}^{n}\|\mathbf{l}_i\|^2 - 2\sum_{i=1}^{3}s_i. \tag{21}$$

The procedures of the computationally efficient post estimation method can be summarized as follows.

First of all, calculate the kth optimal translation vector $\mathbf{t}^{(k)}(A^{(k)})$ using the kth rotation matrix $A^{(k)}$; secondly, update

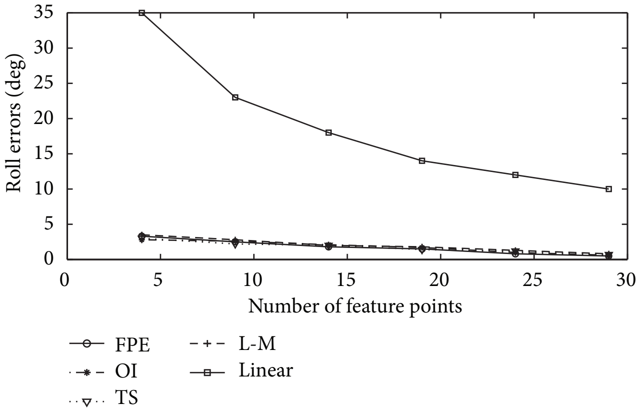

Figure 3: Rotation errors against a number of feature points. FPE denotes the proposed Fast Pose Estimation algorithm, OI means Lu's orthogonal iterative algorithm, TS is Zhang's two stage algorithm, L-M represents the Levenberg-Marquardt method, and the classic linear method is denoted as Linear. The same abbreviations are applied to the figures below too.

the camera-frame coordinates $\mathbf{m}_i^{(k)}(A^{(k)})$ with $\mathbf{m}_i^{(k)}(A^{(k)}) = A^{(k)}\mathbf{p}_i + \mathbf{t}^{(k)}(A^{(k)})$; thirdly, calculate the $(k+1)$th optimal rotation matrix $A^{(k+1)}$ with $\mathbf{l}_i$ and $\mathbf{m}_i^{(k)}$ based on the proposed absolute orientation method. Repeat the process above until the absolute and relative object-space collinearity errors are less than the predefined thresholds. The initial value $A^{(0)}$ is computed with $\mathbf{w}_i$ and $\mathbf{l}_i$ based on the weak perspective approximation [9]. The flowchart of the iterative algorithm is shown in Figure 2.

3. Experiment

In this section, the performance-like accuracy, noise resistance, and computation efficiency for the proposed method are tested and compared with the methods such as Lu's OI algorithm, Zhang's two-stage algorithm, the classic linear transform method [18], and the L-M method, respectively.

The above methods are tested in programming languages such as Matlab 7.0 and VC++ 2008 with LAPACK 3.3.1 (linear algebra package), respectively, and the tests are run on a PC with a CPU of Intel Pentium D E5500 (with clock frequency 2.8 GHz and with 2 GB of random access memory). The operating system is Microsoft Windows XP Professional with service pack 3.

The feature points for the testing are generated randomly within a space of $[-4,4] \times [-4,4] \times [-4,4]$ in the object coordinate frame. The rotation matrix A is chosen randomly by the Euler angles: yaw, pitch, and roll within $(-90, 90) \times (-90, 90] \times [0, 360)$ degree. The translation vector $\mathbf{t}$ is distributed randomly within a space of $[5,15] \times [5,15] \times [10,200]$ in the camera coordinate frame. The focal length on Figure 1 is set to $f = 1$. The coordinates of the image points $\mathbf{w}_i$ are calculated based on $\mathbf{l}_i$, A, and $\mathbf{t}$, and Gaussian white noise is added to $\mathbf{w}_i$ to generate observed points $\widehat{\mathbf{w}}_i$. The standard deviation σ of the noise is a function of the SNR (Signal-to-Noise Ratio) defined by SNR $= -20\lg(\sigma(t_z/10))$. In this way, 1000 sets of $\widehat{\mathbf{w}}_i$ are generated for each set of $\mathbf{l}_i$, A, and $\mathbf{t}$, and the final result is the average value of the 1000 set of $\widehat{\mathbf{w}}_i$.

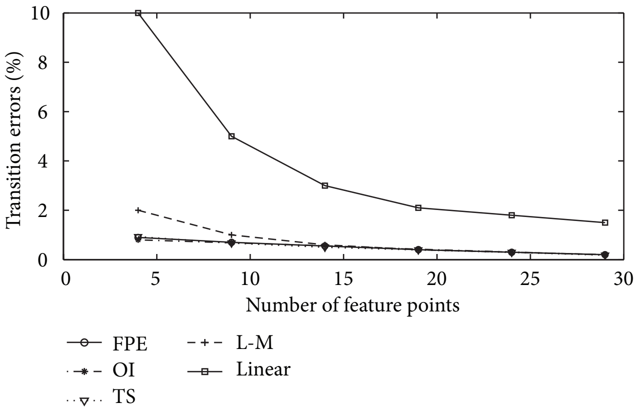

Figure 4: Translation errors versus the number of feature points.

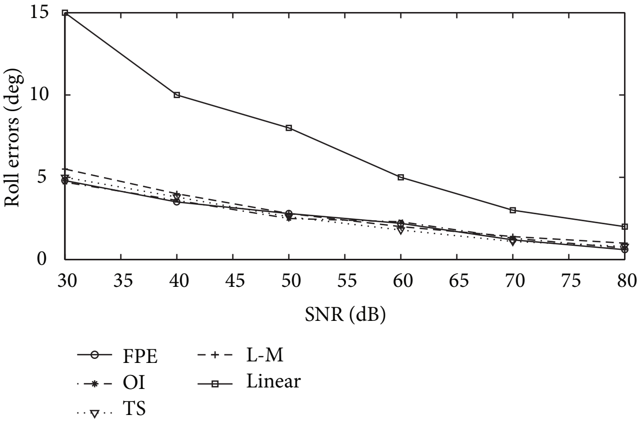

Figure 5: Rotation errors versus SNR.

In the first test, we select $n = 10$ feature points for the optimization, and the SNR is varied from 30 dB to 80 dB with the interval of 10 dB. The errors of Euler angles and translation vector are recorded on the camera, and the result of each of the steps is averaged from 500 sets of $\mathbf{l}_i$, A, and $\mathbf{t}$.

In the second test, the SNR is set to be 60 dB, and the number of feature pints selected varies from 4 to 29 with the interval of 5. The time of computation is recorded as well as the errors of Euler angles and translation vector. The result of each step is taken from the average value of 500 sets of $\mathbf{l}_i$, A and $\mathbf{t}$.

Figure 3 shows the rotation error against the number of feature points, and Figure 4 gives the translation error versus the number of feature points. Figure 5 describes the rotation error against the SNR, and Figure 6 shows the translation error versus the SNR. In these figures, the rotation errors are represented by the roll angle errors because both yaw and pitch have the same effect. It can be seen from these figures that four of the approaches have comparable accuracy and noise-resistance capability, but the performance of the Linear method is relatively poor.

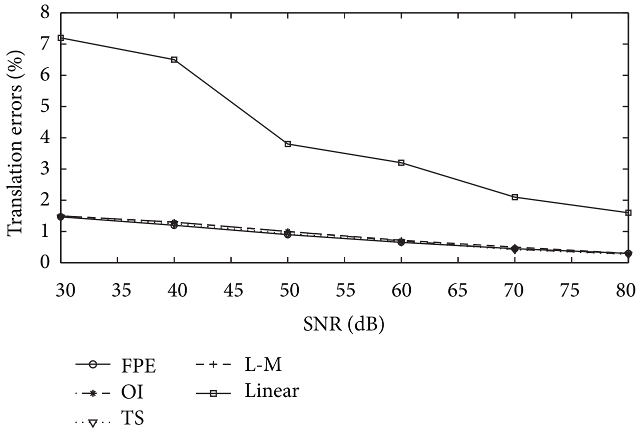

Figure 6: Translation errors against SNR.

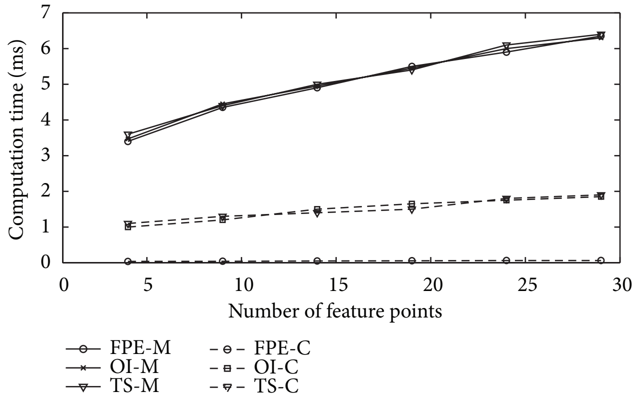

Figure 7: Computation time versus a number of feature points. "FPE-M", "OI-M", and "TS-M" mean that the algorithms are tested in MATLAB language, while "FPE-C", "OI-C", and "TS-C" mean that the algorithms are tested in C++ language.

Figure 7 shows the computation time of three of the algorithms required to complete the calculation in both Matlab and C++ environments. Since the accuracy and noise resistance of the Linear method are not satisfactory, this method is not considered in the comparison of computation time. Also, because the computation time of the L-M method is too time consuming, the time data shown in Figure 7 is far beyond the range of the time coordinate, and therefore the L-M method is not included in the comparison as well. It can be seen from Figure 7 that the programmes run on C++ platform are generally faster than in Matlab environment. The proposed Fast Pose Estimation algorithm does not show its advantage in Matlab because the SVD calculation has been optimized in the built-in function, while the matrix operations in the Fast Pose Estimation algorithm have not been improved. However, when the tests are carried on based on the LAPACK in C++, the SVD calculation and other matrix operations are optimized equally, and it can be seen from the result that the proposed Fast Pose Estimation algorithm performs much faster than the other two approaches. Since, in real world application, our software for embedded system is generally designed using C++ language, the performance of the algorithms in C++ environment would be concerned the most.

4. Conclusions

In this paper, a computationally efficient pose estimation algorithm is proposed based on vision data. In this approach, a new absolute orientation method is designed to replace the time-consuming SVD calculation with the operations such as Frobenius norm, determinant, and adjoint of matrix. Experimental results show that the computation time required for pose estimation with the proposed method is much less than the original SVD approach in C++ programming language, while the performance such as accuracy and noise resistance is maintained as similar with the original method. Therefore, the proposed fast post estimation method is more suitable for real world applications such as the embedded systems used in satellites or other space missions.

References

[1] X. S. Gao, X. R. Hou, J. Tang, and H. F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

[2] Z. Y. Hu and F. C. Wu, "A note on the number of solutions of the noncoplanar P4P problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 550–555, 2002.

[3] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[4] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: an accurate O(n) solution to the PnP problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[5] D. G. Lowe, "Fitting parameterized three-dimensional models to images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 5, pp. 441–450, 1991.

[6] J. Weng, N. Ahuja, and T. S. Huang, "Optimal motion and structure estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 864–884, 1993.

[7] X. B. Cao and S. J. Zhang, "An iterative method for vision-based relative pose parameters of RVD spacecrafts," *Journal of Harbin Institute of Technology*, vol. 37, no. 8, pp. 1123–1126, 2005.

[8] R. M. Haralick, H. Joo, C. N. Lee, X. Zhuang, V. G. Vaidya, and M. B. Kim, "Pose estimation from corresponding point data," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 6, pp. 1426–1446, 1989.

[9] C. P. Lu, G. D. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 610–622, 2000.

[10] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991.

[11] Z. Shijie, L. Fenghua, C. Xibin, and H. Liang, "Monocular vision-based two-stage iterative algorithm for relative position and attitude estimation of docking spacecraft," *Chinese Journal of Aeronautics*, vol. 23, no. 2, pp. 204–210, 2010.

[12] S. J. Zhang, X. B. Cao, F. Zhang, and L. He, "Monocular vision-based iterative pose estimation algorithm from corresponding feature points," *Science in China, Series F*, vol. 53, no. 8, pp. 1682–1696, 2010.

[13] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour, "A closed-form solution of absolute orientation using orthonomal matrices," *Journal of the Optical Society of America A*, vol. 5, pp. 1127–1135, 1988.

[14] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternion," *Journal of the Optical Society of America A*, vol. 4, pp. 629–642, 1987.

[15] M. W. Walker, L. Shao, and R. A. Volz, "Estimating 3-D location parameters using dual number quaternions," *CVGIP: Image Understanding*, vol. 54, no. 3, pp. 358–367, 1991.

[16] K. S. Arun, T. S. Huang, and S. D. Blostein, "A least-squares fitting of two 3-D point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 698–700, 1987.

[17] F. L. Markley, "Attitude determination using vector observations: a fast optimal matrix algorithm," *Journal of the Astronautical Sciences*, vol. 41, no. 2, pp. 261–280, 1993.

[18] Y. I. Abdel-Aziz and H. M. Karara, "Direct linear transformation into object space coordinates in close-range photogrammetry," in *Proceedings of the Symposium Close-Range Photogrammetry*, pp. 1–18, Urbana, Ill, USA, 1971.

Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
International Journal of Distributed Sensor Networks
Advances in Civil Engineering
Journal of Control Science and Engineering
Hindawi
Submit your manuscripts at
http://www.hindawi.com
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration