2021 International Conference on Communications, Information System and Computer Engineering (CISCE) | 978-1-6654-0352-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/CISCE52179.2021.9445890

An Optimized POSIT Algorithm Based on Mean Convergence

Xibing Ni
Beijing Jiaotong University
School of Electronic Engineering
Beijing,China
19125041@bjtu.edu.cn

Chunyue Zhou
Beijing Jiaotong University
School of Electronic Engineering
Beijing,China
chyzhou@bjtu.edu.cn

Hui Tian
Griffith University
School of Information and Communication Technology
Australia
huitian@griffith.edu.au

***Abstract*—Proportional Orthogonal Projection Iteration (POSIT) is the mainstream iterative algorithm in camera pose estimation. The traditional POSIT algorithm has many iterations, too long operating time, and low algorithm robustness. In response to these problems, it proposes a POSIT optimization algorithm based on mean convergence in this paper. The algorithm is based on the traditional POSIT, which the input of each iteration is not the last output, but is determined by the weighted sum of all the previous outputs. The weights are determined by the differences between the outputs and the mean. In order to reduce the number of iterations of the algorithm and the time complexity, the algorithm pre-stores some of the key-value pairs of input and output in the actual scene into the hash table, and uses Bloom filters to save memory space. In theory, the POSIT algorithm can be used down to O(1). Finally, the algorithm is verified by experiments with other existing algorithms in this paper. The results show that the optimized POSIT has significantly improved accuracy and robustness, and its time complexity is lower than that of the traditional POSIT.**

Keywords-machine pose estimation; POSIT; mean convergence

I. INTRODUCTION

Machine pose estimation is an important technical in computer vision, and has a wide range of applications in the fields of augmented reality, photogrammetry, and robot positioning. It refers to determining the direction and position of the camera through the known n (n≥3) 3D reference points and their corresponding 2D projections. It is also called the perspective-n-point (PnP) problem. The PnP problem can generally be divided into two solutions, iterative method and non-iterative method. In the non-iterative method, DLT (direct linear transformation) technology tarts from the projection matrix to obtain the parameters of the machine pose, the time complexity is O(n^5) [1], because of the ignored calibration Parameters, resulting in poor robustness. EPnP technology introduced virtual control points to linearly represent all three-dimensional reference points in the space, and successfully reduced the algorithm complexity to O(n)[4]. However, due to the limited number of virtual control points, it is easy to fall into an infinite loop and the accuracy is low. In terms of iterative method, Raveendran integrates division and square root operations on the basis of traditional POSIT (Proportional Orthogonal Projection Iteration), and develops this technical module on Verilog HDL, which achieves less data path delay and more less hardware usage[8]. The direct least squares (DLS) method can obtains all the stationary points by solving the first-order optimal cost function[9]. But the plan uses Cayley to Parameterize the rotation matrix. When the camera pose is close to the singularity, the accuracy is severely reduced. An accelerated orthogonal iterative algorithm (LHM+) based on the orthogonal iterative algorithm pre-calculates the formula that appears repeatedly in the iteration, reduces the number of iterations, and reduces the iteration calculation time to O(1)[12].

In summary, non-iterative technologies such as DLT and EPnP have low time complexity, but the results are less accurate and robust; iterative algorithms such as DLS, LHM+, and POSIT have high accuracy and good convergence, but it is easy to cause high time complexity when the number of iterations is large. Therefore, this paper proposes an optimized POSIT algorithm based on mean convergence to take into account both accuracy and time complexity, and reduces the time complexity of the algorithm as the accuracy is improved.

II. POSIT ALGORITHM

Assuming that there are n 3D-2D matching point pairs in the camera, the rotation matrix R and the translation vector t of the system is,

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} r_1^T \\ r_2^T \\ r_3^T \end{bmatrix}$$
$$t = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (1)$$

for a point on the object p, let its world coordinate system be (x_w, y_w, z_w), the pixel coordinates be (u, v). The point p normalized coordinates in the camera coordinate system can be calculated from the pixel point $(x_m, y_m, 1)$, as shown in(2),

978-0-7381-1215-2/21/$31.00 ©2021 IEEE

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix} \quad (2)$$

among them， f_x， f_y， u_0， v_0 is internal camera parameters, which is a conversion vector from camera coordinates to pixel coordinates.it can be obtained from (2),

$$y_m = \frac{v - v_0}{f_y}$$
$$x_m = \frac{u - u_0}{f_x} \quad (3)$$

After finding the normalized camera coordinates, just multiply by the depth vectorz_c. Therefore, the point $p$can be obtained from the conversion between the world coordinate system and the camera coordinate system, as shown in the following formula,

$$\begin{bmatrix} z_c x_m \\ z_c y_m \\ z_c \end{bmatrix} = \begin{bmatrix} r_1^T & t_1 \\ r_2^T & t_2 \\ r_3^T & t_3 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (4)$$

to facilitate subsequent calculations, both sides of the equation need to be divided by t_3，

$$\begin{bmatrix} w x_m \\ w y_m \\ w \end{bmatrix} = \begin{bmatrix} s r_1^T & s t_1 \\ s r_2^T & s t_2 \\ s r_3^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (5)$$

among them， $w = \frac{z_c}{t_3}, s = \frac{1}{t_3}$.

As mentioned in the previous article, the depth of an object in the z_c axis direction can be replaced by the geometric center of the object. When the geometric center is taken as the origin of the world coordinate system, the depth value of the object in the z_c axis direction is approximately equal to the t_3 component, $z_c = t_3$. At this point, the perspective projection model becomes a weak perspective projection model， $w = z_c / t_z \approx 1$. In this way, (5) can be simplified, and the last row of the matrix can be reduced to obtain the linear equations (6)，

$$\begin{cases} w x_m = s r_{11} x_w + s r_{12} y_w + s r_{13} z_w + s t_1 \\ w y_m = s r_{21} x_w + s r_{22} y_w + s r_{23} z_w + s t_2 \end{cases} \quad (6)$$

there are 8 variables in the above formula, which are， r_{11}， r_{12}， r_{13}， t_1， r_{21}， r_{22}， r_{23}， t_2， according to the nature of the linear equation solution, 8 variables need at least 4 pairs of matching points, and these 4 pairs of matching points cannot be coplanar. In addition, it is necessary to obtain specific R and t based on the unit orthogonality of the rotation matrix. under the surface enters the iterative process. The purpose of this step is to precisely rotate the matrix and translate the value of the vector. Update the depth z_c value And coefficient w from (6),

$$z_c = \begin{bmatrix} r_3^T & t_3 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \qquad w = \frac{z_c}{t_3} \quad (7)$$

this will generate new depth information for the matching point, and correspondingly, w will also have a new value, (6) will be updated. Continue to solve the system of equations to get more accurate R and t, and R and t can get more accurate z_c and w. Repeated update iterations can gradually approach the exact solution.

III. POSIT ALGORITHM BASED ON MEAN CONVERGENCE

It can be seen from the iterative process of the traditional POSIT algorithm that the rotation matrix and the translation vector need to be continuously updated to continuously solve the linear equations. The algorithm complexity can reach $O(n^k)$, where k is the number of iterations. Therefore, in order to improve the efficiency of the algorithm, this paper proposes an optimized POSIT algorithm based on mean convergence. The algorithm process is as follows:

Suppose the output value of the i-th iteration of the POSIT algorithm being R_i and t_i， keep the value. The input of the i+1th iteration is related to the outputs of the previous i. First find out the mean of all previous output values, namely,

$$\bar{R} = \frac{1}{i}(R_1 + R_2 \ldots + R_i),$$
$$\bar{t} = \frac{1}{i}(t_1 + t_2 + \cdots + t_i). \quad (8)$$

calculate the difference between $R_i$and t_i and the mean $\bar{R}$ and $\bar{t}$ respectively. At the same time, the algorithm will set the corresponding threshold. If the difference exceeds the threshold, it will be judged as an abnormal value and the corresponding parameter value will be eliminated. Assuming that the eligible parameters are R_j and t_j， $j \in i$. Next, assign the corresponding weights $\widehat{R_j}$ and $\widehat{t_j}$ to each $R_j$and t_j. This is their contribution to the final input. The weight is inversely proportional to the difference between them and the expected value, that is,

$$\widehat{R_j} = \frac{1}{R_i - \bar{R_j}} / \sum_j \frac{1}{R_i - \bar{R_j}},$$
$$\widehat{t_j} = \frac{1}{t_j - \bar{t_j}} / \sum_j \frac{1}{t_j - \bar{t_j}} \quad (9)$$

finally, the input value of the i+1th iteration can be obtained,

$$R_{i+1} = \sum_j \widehat{R_j} R_j,$$
$$t_{i+1} = \sum_j \widehat{t_j} t_j \quad (10)$$

the algorithm ends when the differences between R_{i+1} and t_{i+1} and the outputs of the previous iteration converge to a minimum. In this way, each iteration will ensure that the solution is within a reasonable range, avoid the appearance of outliers, and reduce unnecessary iterations; each iteration will adjust the convergence scale toward the final stable value, which improves the speed of the algorithm convergence.

It can be observed that the algorithm has to calculate (10) for each iteration, which will consume some time. In order to reduce a large number of repetitive calculations, this paper preliminarily stores the results of some values in the hash table. In this way, in each iteration, first check whether there is any knowledged results in the hash table. If not, calculate (10) and update the hash table in time. Through hash search, the algorithm time complexity can be reduced to O(1). This will cause another problem, requiring additional memory space, and the hash table has only 50% utilization. In order to save memory space and improve utilization, this paper uses Bloom filter to optimize the hash table. Each key-value pair is

randomly mapped into 8 bits, and these 8 bits are set to 1. The space complexity is 1/8 of the original. POSIT optimization based on mean convergence is shown in algorithm 1.

Algorithm 1 POSIT based on expected convergence

Input: matching point pair p q
Output:Rotation matrix R translation vector t 1：function "APOSIT"
2: while($\Delta R < \varepsilon, \Delta t < \varepsilon$)
3: if i==1 then
4: hash[1] ←{R,t}
5: else then
6: for j=1 to sum of hash
7: s←s+hash[i]
8: end for
9: end if
10: mean←s/sum of hash
11: sumhash←sum of hash
12: for i=1 to sumhash
13: if(R[i]-mean)>∂ then
14: delete R[i]
15: else then
16: s1=s1+1/(R[i]-mean)
17: end for
18: for i=1 to sumhash
19: if [R t] not in the hash then
20: Rs=R[i]/[(R[i]-mean)*s1]
21: hash[k++]={Rs,ts};
22: end if
23: end for
24: w=[R[3] t[3]]*p/t[3]
25: find Rd and td in terms of w
26: ΔR=Rd-R, Δt=td-t
27: end function

IV. EXPERIMENTAL RESULTS AND ANALYSIS

This section will verify the performance of the algorithm proposed in this through simulation experiments, and conduct experiments on simulated data and real images. Camera with virtual calibration， effective focal length $f_x = f_y = 800$, the optical center of the camera is $(u_c, v_c) = (320,240)$. For simulation data, the three-dimensional reference points are evenly distributed in $x, y, z,$ interval $[-2,2] \times [-2,2] \times [4,8]$. Taking into account the noise interference in the actual work scene, this paper adds Gaussian noise to the two-dimensional coordinates corresponding to the 2D reference points. In order to verify the effect of the improved algorithm， this paper makes a detailed comparison with DLT, EPnP, DLS and LHM+ algorithms in terms of accuracy, robustness and time complexity. In order to facilitate the observation of experimental results, the algorithm in this paper is represented by APOSIT(POSIT based on expected convergence). The experimental simulation environment is: processor Inter Core i7 10510U@1.8GHz， operating system win10， memory 16GB，simulation software MATLAB2020Rb.

Under the virtual calibration camera, randomly generate camera external parameters R_{true} and t_{true} ,according to the conversion relationship between the world coordinate system and the pixel coordinate system, the corresponding two-dimensional image coordinates can be calculated by randomly generated three-dimensional coordinates. The error between the rotation matrix R and the translation vector t and the true value is shown in the following:

$$E_{rot} = \|q - q_{true}\| / \|q_{true}\|$$
$$E_{trans} = \|t - t_{true}\| / \|t_{true}\| \quad (11)$$

among them， q_{true} and $q \in [a, b, c, d]$, They are the unit quaternions of R_{ture} and R, respectively. The following paper will simulate the accuracy, robustness and time complexity of the experiment from the two aspects of the number of reference point pairs and the Gaussian noise scale.

A Number of reference point pairs

In the experiment, the range of the number of reference points selected is [6,36], and the step size is 6. Gaussian noise is set to a mean value of 0 and a standard deviation of 1.7. In order to eliminate the accidental errors of the system, 100 simulation experiments were performed on each pair of matching points, and the average and standard variance of each algorithm error were measured. The experimental results are shown in Figure 1.

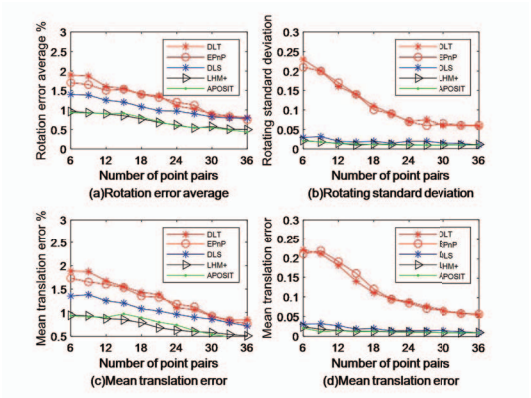

Figure 1. Rotation and translation errors at different points in logarithm

It can be seen from the experimental results that the APOSIT rotation and translation error accuracy is significantly better than the DLS and DLT algorithms, which is due to the effect of POSIT optimization. Secondly, compared with the EPnP algorithm, the rotation and translation errors of APOSIT are also lower. EPnP selects four non-coplanar virtual control points as the base of the three-dimensional space. It is inevitable to produce error to replace all points in the space. It can be seen that LHM+ and PDLS are very close in accuracy, which is due to the good convergence of the iterative algorithm. When the number of point pairs is greater than or equal to 24, it can be seen that the advantages of APOSIT are more obvious. This is based on the advantages of the POSIT algorithm based on expected convergence over the orthogonal iterative algorithm, and will not produce abnormal values with excessive deviations. It can be seen from the variation of variance that as the number of point pairs increases, the rotation and translation errors of each algorithm also gradually decrease. This is also in line with the basic nature of the solution of linear algebraic equations, and the number of constraints is proportional to the accuracy of the solution. Continue to observe the iterative algorithm and the non-iterative algorithm, it can be concluded that the iterative algorithm is more stable. This is because the iterative method has good convergence, and the non-iterative method may have derivation errors. Among the iterative algorithms, APOSIT and EPnP are the most stable.

B Gaussian noise scale

In this section, various algorithms will be tested according to different Gaussian noise scales to verify the robustness of different algorithms. The average value of Gaussian noise used in the experiment is 0, the standard deviation is σ, the variation range of σ is [0.3, 3], the step size is 0.3, and the number of reference point pairs is set to 24. Perform 100 simulation experiments on each scale noise, and count the average and standard variance of each algorithm. The experimental results are shown in Figure 2.

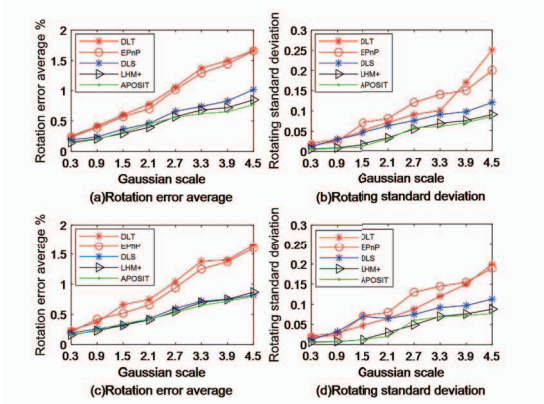

Figure 2. Rotation and translation errors of noise at different scales

From the experimental results, it can be concluded that under different noise scales, APOSIT has significantly improved accuracy compared with EPnP and DLT in terms of rotation matrix and translation vector. This is because the iterative method has good convergence. From the perspective of variance changes, APOSIT is more robust than DLS and EPnP, and has a stronger ability to adapt to noise. Under the influence of different noises, the errors of LHM+ and APOSIT are basically the same. However, by observing their growth rates, PDLS has a slower growth rate and stronger stability.

C Algorithm time complexity

This section conducts simulation experiments on the time complexity of the above-mentioned algorithms, taking the number of reference point pairs as the independent variable, and the range of the number of point pairs is divided into [3,24] and [36,120], and the step length is 3 and 12. The purpose of the two intervals is to verify the time complexity of the algorithm for small number pairs and large number pairs, and to compare the performance of the algorithms in different scenarios. Set the mean value of Gaussian noise to zero and the standard deviation to 1.7. Each algorithm performs 100 experiments under different logarithms to calculate the average and standard variance of the calculation time. The experimental simulation results are shown in Figure 3.

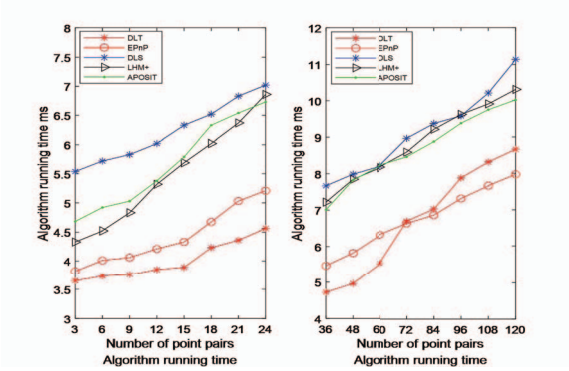

Figure 3. Calculation time verification at different point logarithms

First of all, from the overall analysis, DLT and EPnP tend to be stable under different number of point pairs, and the algorithm consumes less time. On the contrary, as the number of point pairs increases, DLS, LHM+ and APOSIT have a larger time increase. This is because the iterative algorithm, as the number of point pairs increases, repeated iterations will inevitably increase time consumption. It can be seen from this that the computational time complexity of the non-iterative method is lower than that of the iterative method, and this gap will increase exponentially as the number of point pairs increases. In the linear algorithm, EPnP and DLT are in the interval [3,24], and the time consumption is lower than that of LHM+ and APOSIT. In the interval [36,120], the time complexity of EPnP is significantly lower. The time complexity of LHM+ and PDLS is basically the same. The time complexity of DLS and DLT has increased significantly. This shows that the traditional DLS and DLT algorithms have high computational complexity when the number of point pairs is large, which affects the performance of the algorithm. In the iterative algorithm, APOSIT uses POSIT optimization based on expected convergence, which has certain advantages over LHM+ and DLS in terms of time, especially when the number of point pairs is relatively large.

In summary, the optimized POSIT machine pose estimation algorithm based on mean convergence proves that it has high performance in accuracy, robustness and computational time complexity through the above three experimental analysis.

V. CONCLUSION

It investigates the current machine pose estimation methods, analyzes the pros and cons of various pose estimation algorithms, and proposes an optimized POSIT based on mean convergence in this paper. The basic idea is: the input of each iteration of the algorithm is determined by the weighted sum of the previous output values, which improves the convergence speed and robustness of the algorithm; It uses a hash table to pre-store some key-value pairs which reduces the number of iterations and uses bloom filter to solve the problem of low memory usage of the hash table which saves memory space in this algorithm. Compared with the traditional POSIT algorithm, this algorithm avoids the appearance of outliers, accelerates the optimization speed, and reduces the time complexity. Finally, this paper uses MATLAB simulation software to verify the calculation error

and calculation time of the simulated data with Gaussian noise. Experimental results show that the algorithm proposed by this paper has obvious improvements in accuracy, robustness and time complexity. Future research work will focus on the impact of optimized POSIT on computer vision applications such as augmented reality.

ACKNOWLEDGMENT

This work was supported by the Fundamental Research Funds for the Central Universities(2019JBZ001), for which we grateful.

REFERENCES

[1] R. Frohlich, L. Tamas and Z. Kato, "Absolute Pose Estimation of Central Cameras Using Planar Regions," IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 43, no. 2, pp. 377-391, 1 Feb. 2021.

[2] L.Kneip, D.Scaramuzza and R.Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation ， "Proc. CVPR,Colorado Springs,2011，pp.2969–2976.

[3] A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol,25, no,5, pp. 578-589, May 2003.

[4] Vincent Lepetit; Francesc Moreno-Noguer; Pascal Fua (2009). "EPnP: An AccurateO(n) Solution to the PnP Problem,".vol,81,no,2, pp.155–166.

[5] H. Tjaden, U. Schwanecke, E. Schömer and D. Cremers, "A Region-Based Gauss-Newton Approach to Real-Time Monocular Multiple Object Tracking," IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol,41,no,8,pp. 1797-1812, 1 Aug. 2019.

[6] H. Meng, F. Yuan, T. Yan and M. Zeng, "Indoor Positioning of RBF Neural Network Based on Improved Fast Clustering Algorithm Combined With LM Algorithm,". IEEE Access, vol.7,pp,5932-5945,2019.

[7] B. M. Wilamowski and H. Yu, "Improved Computation for Levenberg–Marquardt Training,". *IEEE Transactions on Neural Networks*, vol,21,no,6,pp,930-937, June 2010.

[8] A. Raveendran, S. Jean, J. Mervin, D. Vivian and D. Selvakumar,". A Novel Parametrized Fused Division and Square-Root POSIT Arithmetic Architecture," 2020 33rd International Conference on VLSI Design and 2020 19th International Conference on Embedded Systems (VLSID),Bangalore,India, 2020,pp.207-212.

[9] J. A. Hesch and S. I. Roumeliotis, "A Direct Least-Squares (DLS) method for PnP,". 2011 International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 383-390.

[10] J. Wu, "MARG Attitude Estimation Using Gradient-Descent Linear Kalman Filter,". IEEE Transactions on Automation Science and Engineering, Vol,17,no,4,pp,1777-1790, Oct. 2020.

[11] C. -. 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, June 2000.

[12] Li Xin, Long Gucan, Liu Jinbo, Zhang Xiaohu and Yu Qifeng."Accelerated orthogonal iterative algorithm for camera pose estimation".Acta Optica Sinica, 2015, vol.35 no.01,pp,266-273.

[13] S. Li, C. Xu and M. Xie, "A Robust O(n) Solution to the Perspective-n-Point Problem,". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol,34, no,7,pp,1444-1450, July 2012.

[14]S. H. Fatemi Langroudi, T. Pandit and D. Kudithipudi, "Deep Learning Inference on Embedded Devices: Fixed-Point vs Posit," 2018 1st Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications (EMC2), Williamsburg, VA, USA, 2018, pp. 19-23.