



**HAL**  
open science

# An improved method for Rotation Estimation Using Photometric Spherical Moments

Yao Du, Omar Tahri, Hicham Hadj-Abdelkader

► **To cite this version:**

Yao Du, Omar Tahri, Hicham Hadj-Abdelkader. An improved method for Rotation Estimation Using Photometric Spherical Moments. 16th International Conference on Control, Automation, Robotics and Vision, Mar 2021, Shenzhen, China. 10.1109/icarcv50220.2020.9305374 . hal-03183387

**HAL Id: hal-03183387**

**<https://hal.science/hal-03183387v1>**

Submitted on 27 Mar 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/344667310>

# An improved method for Rotation Estimation Using Photometric Spherical Moments

Conference Paper · December 2020

DOI: 10.1109/ICARCV50220.2020.9305374

CITATIONS

0

READS

16

3 authors:



Yao du

University of Burgundy

3 PUBLICATIONS 2 CITATIONS

SEE PROFILE



Omar Tahri

University of Burgundy

62 PUBLICATIONS 888 CITATIONS

SEE PROFILE



Hadj Hicham

Université Clermont Auvergne

48 PUBLICATIONS 288 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



vision-based navigation for mobile robots [View project](#)



phd thesis [View project](#)

# An improved method for Rotation Estimation Using Photometric Spherical Moments

Yao Du, Omar Tahri and Hicham Hadj-Abdelkader

**Abstract**—Using spherical moments to estimate the rotation of camera is an effective method. However, after the camera rotates, the scene in the captured image will change, which will cause a certain error. This paper proposes a method to reduce this error by removing the non-common part between the two images during rotation. Through multiple iterations, the error caused by scene changing will become very small. The experimental results using manual generated image are provided to show the efficiency of this proposed method.

## I. INTRODUCTION

It has always been a human dream to make machines that can replace human labor, but currently even the most intelligent robots in the world have very limited ability to adapt to changes in the external environment. There is still a great distance from people's expected goals, which has greatly affected the promotion and use of robots. One of the important reasons is that robots lack human-like perception. To solve this problem, researchers began to add various external sensors to the robot [1], one of the most important ones is the visual sensor.

In the fields of national defense and aerospace, machine vision has very important application significance [2], such as automatic tracking [3] and recognition of moving targets [4], autonomous tank navigation, automatic landing of the lunar module [5], [6], and visual control of space robots. In the modern industrial automated production process, machine vision is also becoming a key technology to improve production efficiency and ensure product quality [7]. Such as automatic detection of mechanical parts, intelligent robot control and automatic monitoring of production lines.

Estimating their posture is a significant task for many applications of robot including drones [8]. As early as 1987, [9] used finite Fourier transform to process images to estimate translation and rotation motion. The commonly used methods for estimating translation and rotation motion are mainly feature point method [10], [11] and [12], direct method and optical flow method [13], [14], and [15]. Traditional method depends on handcrafted features as object representations for rotation estimation [16]. Actually, the feature point method uses feature point matching to track points, calculate the geometric relationship to estimate rotation and translation, and then obtain rotation and translation motion through an

optimization algorithm [17]. This kind of approaches using features is limited to the texture-rich scene, and if too many or too few features they do not work. The feature point methods rely on a highly repeatable feature extractor and correct feature matching in order to correctly calculate the camera motion. Besides, it spends a lot of time on calculating descriptors and matching. For the direct method, there is no need to calculate geometric descriptors, and then no matching is required, which saves a lot of calculation. It works as long as the key point has a gradient [18]. Among direct methods, [19] proposed an approach of estimation using optical flow. The disadvantage is that when the camera moves or rotates at a large scale, it cannot track well. Several methods can be used at the same time to achieve better results. For instance [20] use the direct method and the optical flow method alternately according to the number of feature points.

Recently, [21] has exploited a method based on spherical moment, which has provided an analytical solution to rotation. This is a fast and elegant algorithm, which avoids the extraction and calculation of feature points, and can meet most scene and real-time requirements.

Since the camera is rotated before and after, it will produce a part of the non-common areas in the photo, which may cause a certain error. However, this part of the source of error can be avoided by adding an optimization module to remove the non-common area between the two pictures. As the camera rotates, there will be a part of the non-common area, which may cause a certain error. This part of the error source can be reduced or even removed by adding an optimization module. In section 2, the principle and improvement of this method are introduced.

## II. ROTATION ESTIMATION PRINCIPLE

A. *Rotation estimation using spherical moments as proposed in [22]*

In this section, first the necessary steps of the proposed algorithm in previous work are described. For specific theoretical derivation, the readers can refer to [22].

The spherical moments  $m_{ijk}$  are computed based on the following formula:

$$m_{ijk} = \iint_{\text{Image}} x_s^i y_s^j z_s^k \frac{(\xi + z_s)^3}{1 + \xi z_s} I(x, y, t) dx dy \quad (1)$$

where  $I(x, y, t)$  is the image at time  $t$ .  $x_s, y_s, z_s$  is the coordinate of the corresponding point of the image on the unit sphere,  $\xi$  is the distortion parameter, and  $(x, y)$  are the

This work was supported by China Scholarship Council  
Y. DU is with VIBOT ERL CNRS 6000, ImViA, Université de Bourgogne Franche-Comté, France [yao.du@insa-cvl.fr](mailto:yao.du@insa-cvl.fr)  
O. Tahri is with VIBOT ERL CNRS 6000, ImViA, Université de Bourgogne Franche-Comté, France [omar.tahri@u-bourgogne.fr](mailto:omar.tahri@u-bourgogne.fr)  
H. Hadj-Abdelkader is with IBISC lab. EA4526, University of Evry-Val-d'Essonne-Paris Sacleay, France

coordinates of a point in the metric image. Which means that they can be computed directly on the 2D image. From the spherical moments, the triplets  $P_1, P_2, P_3$  to be used in rotation estimation is calculated by (2) as in [22].

$$P_1 = A_1 + B_1 + C_1$$

$$A_1 = \begin{bmatrix} m_{003}m_{101} + m_{012}m_{110} + m_{021}m_{101} \\ m_{003}m_{011} + m_{011}m_{021} + m_{012}m_{020} \\ m_{002}m_{003} + m_{002}m_{021} + m_{011}m_{012} \end{bmatrix}$$

$$B_1 = \begin{bmatrix} m_{030}m_{110} + m_{101}m_{201} + m_{102}m_{200} \\ m_{020}m_{030} + m_{011}m_{201} + m_{102}m_{110} \\ m_{011}m_{030} + m_{002}m_{201} + m_{101}m_{102} \end{bmatrix}$$

$$C_1 = \begin{bmatrix} m_{110}m_{210} + m_{120}m_{200} + m_{200}m_{300} \\ m_{020}m_{210} + m_{110}m_{120} + m_{110}m_{300} \\ m_{011}m_{210} + m_{101}m_{120} + m_{101}m_{200} \end{bmatrix}$$

$$P_2 = A_2 + B_2 + C_2$$

$$A_2 = \begin{bmatrix} m_{002}m_{120} - 2m_{011}m_{111} + m_{020}m_{102} \\ m_{002}m_{030} - 2m_{011}m_{021} + m_{012}m_{020} \\ m_{002}m_{021} + m_{003}m_{020} - 2m_{011}m_{012} \end{bmatrix}$$

$$B_2 = \begin{bmatrix} m_{002}m_{300} - 2m_{101}m_{201} + m_{102}m_{200} \\ m_{002}m_{210} + m_{012}m_{200} - 2m_{101}m_{111} \\ m_{002}m_{201} + m_{003}m_{200} - 2m_{101}m_{102} \end{bmatrix}$$

$$C_2 = \begin{bmatrix} m_{020}m_{300} - 2m_{110}m_{210} + m_{120}m_{200} \\ m_{020}m_{210} + m_{030}m_{200} - 2m_{110}m_{120} \\ m_{020}m_{201} + m_{021}m_{200} - 2m_{110}m_{111} \end{bmatrix}$$

$$P_3 = A_3 + B_3$$

$$A_3 = \begin{bmatrix} m_{002}m_{102} + 2m_{011}m_{111} + m_{020}m_{120} \\ m_{002}m_{012} + 2m_{011}m_{021} + m_{020}m_{030} \\ m_{002}m_{003} + 2m_{011}m_{012} + m_{020}m_{021} \end{bmatrix}$$

$$B_3 = \begin{bmatrix} 2m_{101}m_{201} + 2m_{110}m_{210} + m_{200}m_{300} \\ 2m_{101}m_{111} + 2m_{110}m_{120} + m_{200}m_{210} \\ 2m_{101}m_{102} + 2m_{110}m_{111} + m_{200}m_{201} \end{bmatrix} \quad (2)$$

Through the triplets in (2), a set of orthogonal bases can be obtained.

$$P_{n1} = \frac{P_1}{\|P_1\|}, P_{n2} = \frac{P_2}{\|P_2\|} \quad (3)$$

$$V_1 = \frac{P_{n1} + P_{n2}}{\|P_{n1} + P_{n2}\|}, V_2 = \frac{P_{n1} - P_{n2}}{\|P_{n1} - P_{n2}\|}, V_3 = V_1 \times V_2 \quad (4)$$

In the last step, the 3D rotation  $R$  between two images can be obtained through their corresponding orthogonal bases  $V$  and  $V'$  by:

$$R = VV'^T \quad (5)$$

Fig. 1 is a flowchart of every step in the original algorithm.

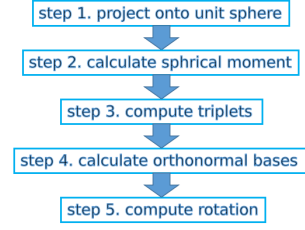


Fig. 1. Flowchart of the original algorithm

### B. Contribution and outline of the paper

It can be seen from the above algorithm that the entire image is used when initially calculating the spherical moments of the image. However, in practical situations, not all scenes between the two images are in common, which will cause a certain error.

Fortunately, it can be found by studying the formula (1) that the integrated term can be divided into two parts. The first part is the product of the high-order power of coordinates  $x, y, z$  of the points on the spherical surface, and the second part is the corresponding gray value of the pixel in image. The essence of spherical moments is to integrate the product of the above two items on the whole image, which is a summation operation. Therefore, spherical moments of the whole image is equal to the sum of the spherical moments of the common area and the non-common area, which is the part should be removed. That is to say, the spherical moments of the common part of two pictures is equal to the difference between the spherical moments of the entire image and the spherical moments of the non-common part, as follows.

$$M_{common} = M_{whole} - M_{non-common} \quad (6)$$

where  $M_{common}$  is the spherical moments of common part,  $M_{whole}$  is the spherical moments of entire image,  $M_{non-common}$  is the spherical moments of non-common area.

Therefore, the rotation estimation can be improved by removing the non-common part between the two images as shown on the flowchart in Fig. 2.

It should be paid attention to that the added part can be iterated multiple times, and in theory, the more iterations, the smaller the error caused by the non-common part between the two images.

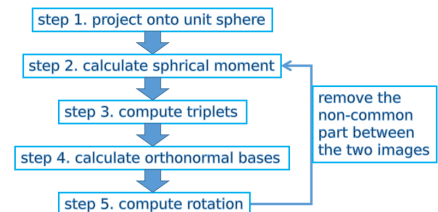


Fig. 2. Flowchart of the algorithm with added optimization module

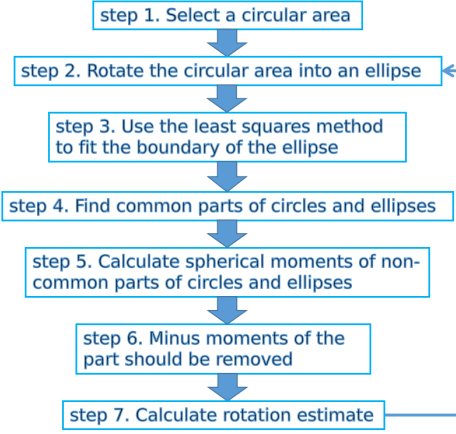


Fig. 3. Algorithm flow chart for removing non-common parts

For the specific steps of removing non-common part between two images, algorithm flow chart can be found in Fig. 3. Firstly, we follow the original method to obtain the first estimation of rotation by calculating the moments and triplets of the two images. Then according to the first estimation, the circular boundary of the second image is rotated to get an ellipse. The area within the elliptical boundary is the field of view of the second image in the first image.

As seen in Fig. 4. The inside of the red circular border is the field of view of the first image, and the area inside the blue ellipse is the field of view of the second image in the first image.

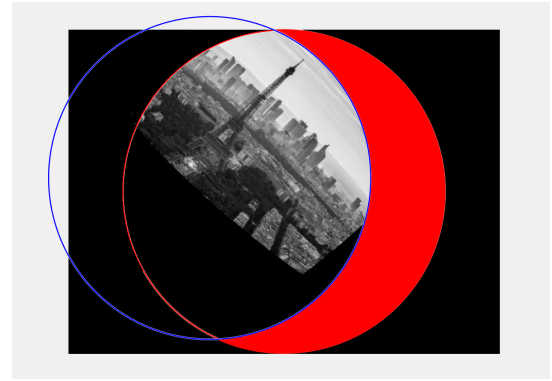
Therefore, the public area in the first image is easy to find, and then the non-common part in the first image can be removed. As shown in Fig. 5(a), the red area is the part should be removed in first image.

In the same way, the non-common part of the second image can be found. As shown in Fig. 5(b), the blue area is the part should be removed in second image. Only not with the  $R$  matrix in the first image situation but the inverse of  $R$ .

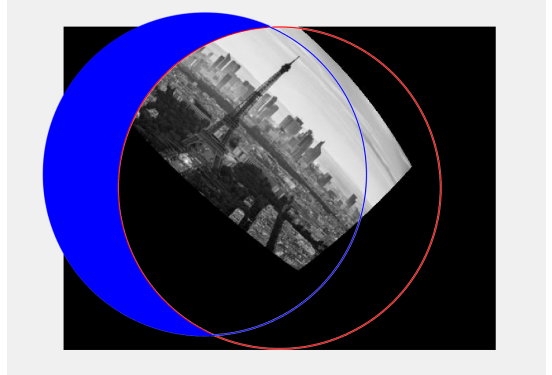
The spherical moments of the first and second images in the circle calculated in the first step are subtracted from the



Fig. 4. The fitted circular boundary and rotated ellipse



(a) The part should be removed from first image



(b) The part should be removed from second image

Fig. 5. The area should be removed

corresponding spherical moments of the parts that should be removed, and the new spherical moments of the two images are obtained to calculate a new rotation estimate between the two images.

After obtaining a new rotation estimate, repeat the above steps to get a more accurate rotation estimate, and iterate multiple times.

The above describes how to calculate the rotation estimation  $R_n^{n+1}$  between two adjacent pictures in an image sequence. The following will explain how to calculate the relative rotation of the first image and each image from the second image in the image sequence.

In order to increase the common part between two pictures as much as possible, (7) is used to calculate the rotation of the  $(n+1)^{th}$  image and the  $n^{th}$  image.

$$R_0^{n+1} = R_n^{n+1} \times R_0^n \quad (7)$$

where  $R_0^{n+1}$  is the rotation of the  $(n+1)^{th}$  ( $n \geq 1$ ) picture relative to the first image,  $R_0^n$  is the rotation between the  $n^{th}$  image and the first image,  $R_n^{n+1}$  is the rotation of the  $(n+1)^{th}$  ( $n \geq 1$ ) image and the  $n^{th}$  image.

Fig. 6 is a schematic diagram of calculating the relative rotation between the  $(n+1)^{th}$  ( $n \geq 1$ ) image  $I_{n+1}$  and the first image  $I_1$ .

The experiments in next section will demonstrate that the convergence of this improvement.

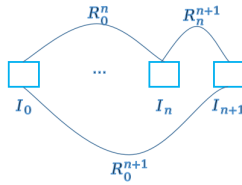


Fig. 6. Calculating the relative rotation between the  $n + 1$ th image and the first image

### III. EXPERIMENTS

Because GPS cannot obtain all information about the pose of the camera, sensors such as IMU and gyroscope have errors such as temperature drift and zero drift, so the experimental part uses the simulated rotation data set generated by matlab. This section conducted 2 simulation experiments, where 2 data sets corresponding to two different scenes. The size of the generated image is  $480 \times 640$ , the parameters of camera are  $k = \begin{bmatrix} 800 & 0 & 240 \\ 0 & 800 & 320 \end{bmatrix}$ ,  $\xi = 1.8$ . And selected number of iterations is 6.

The scene and the angular velocity of the rotation are different in two simulation. The rotation angles on the three coordinate axes  $[x, y, z]$  were recorded while generating the rotated pictures to compare the experimental results.

The processor of computer configuration used for the experiment is AMD R5 3550H CPU.

#### A. Simulation 1

In the first experiment, the angular velocities in  $[x, y, z]$  direction are considered as:

$$w_x = 0.1rad/s$$

$$w_y = 0.4rad/s$$

$$w_z = 0.5rad/s$$

Fig. 7 shows six images from the used sequences in simulation 1. The red circle indicates the camera's field of view. As shown in the figure, a part of the rotated picture ran out of the field of camera view, when calculating the rotation of the camera based on the spherical moments, this part becomes an error.

The total time spent processing each picture is  $50ms$ , where the time spent by the added optimization module is  $10ms$ . The simulation was conducted using Matlab. Therefore, there is still much room for improvement in the speed of the program.

The obtained results are shown on Fig. 8, where x axis is the serial number of the picture, and y axis is the angle on the corresponding coordinate axis. The red line is the real value, the blue line is the result calculated with optimization module, and the black line indicates the result without removing non-common part of two images.

It can be seen that at the beginning and end of the x-axis, the estimated value using the newly proposed algorithm completely coincides with the true value, and the estimated

value without removing the non-common part has a large deviation from the true value. Especially in the position of the  $50^{th}$  picture and the  $230^{th}$  to  $280^{th}$  images in the images sequence, it is most obvious. On the y-axis, the estimated value and the real value of the newly proposed algorithm almost all coincide, and the original old algorithm still has a large error in the position of the  $40^{th}$  to  $80^{th}$  pictures. In the z-axis direction, at the beginning, the estimated value using the newly proposed algorithm has almost no error, and the error of the algorithm that does not remove the non-common part reaches about 30%. The two algorithms perform similarly, but after 250 images, the error of the new algorithm is only about a quarter of the error of the original algorithm. This shows the experimental results using the added optimization module are better.

#### B. Simulation 2

In the second experiment, the configuration of the computer, the parameters of the selected camera, and the number of iterations are the same as in simulation 1. The difference from the previous experiment is that the used scene is different as shown in Fig. 9, and the following larger angular velocities were considered :

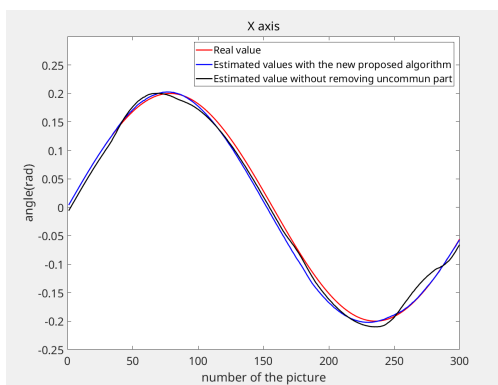
$$w_x = 0.4rad/s$$

$$w_y = 0.7rad/s$$

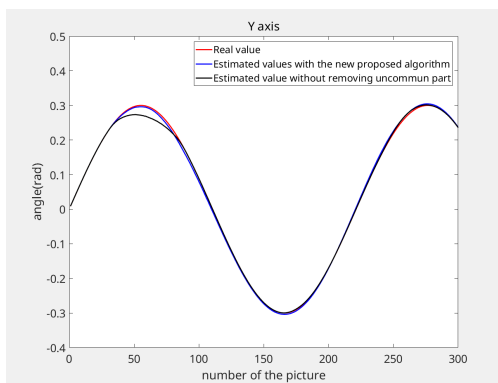
$$w_z = 1.1rad/s$$



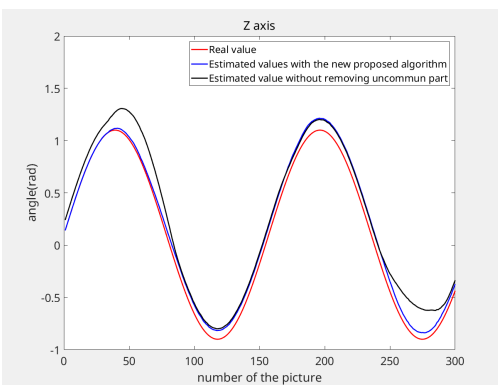
Fig. 7. Images of image sequence used in simulation 1



(a) X axis



(b) Y axis



(c) Z axis

Fig. 8. The result of simulation 1 with 300 images

The test results show that the processing time for each picture and the time consumed by the added optimization module are also basically the same. As can be seen from the illustrations shown in the Fig. 10, at the beginning of the estimation in the x-axis direction, the new method produced good results, and the new algorithm had advantages up to the 200th picture, but after about 250th picture, the error of the new method Obviously become larger. We will discuss this part in the next section. In the y-axis direction, before the 200th image, the new algorithm performs well, with almost no errors, and the old algorithm has obvious errors on the 50th to 150th images. On the entire y-axis, the new algorithm has advantages over the new whole. In the z-axis direction,

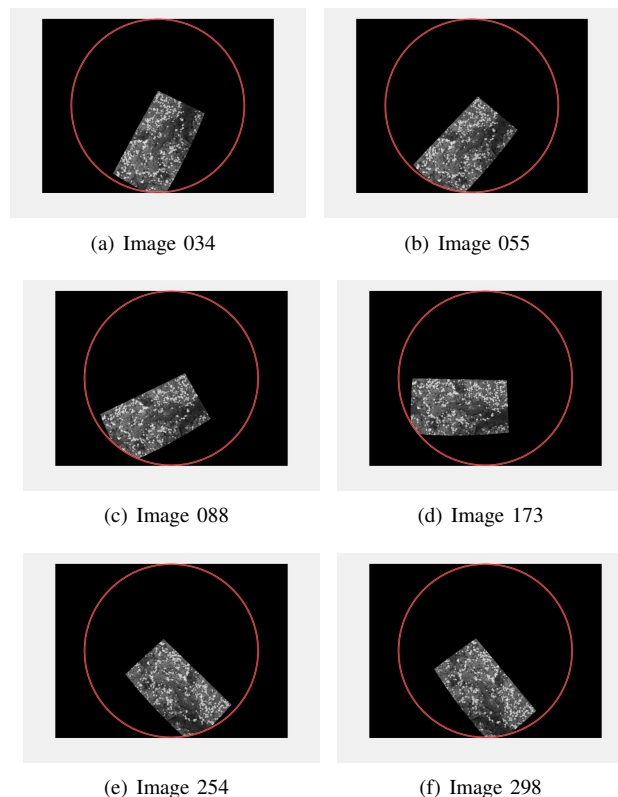


Fig. 9. Images of image sequence used in simulation 2

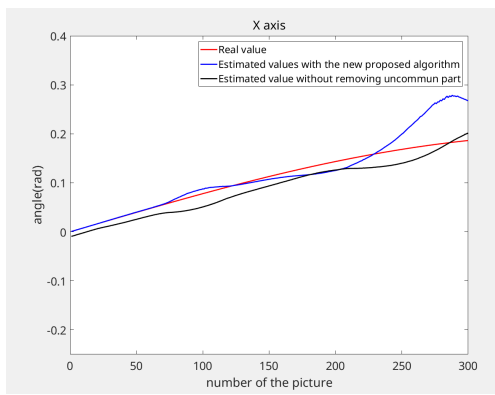
this advantage is more obvious. The first half of the new algorithm almost completely coincides with the real rotation, while the old algorithm has obvious errors. The second half of the new algorithm produces errors, but it is always smaller than the old algorithm. In the results in the entire z-axis direction, the new algorithm has greatly improved compared to the original algorithm.

#### IV. DISCUSSION

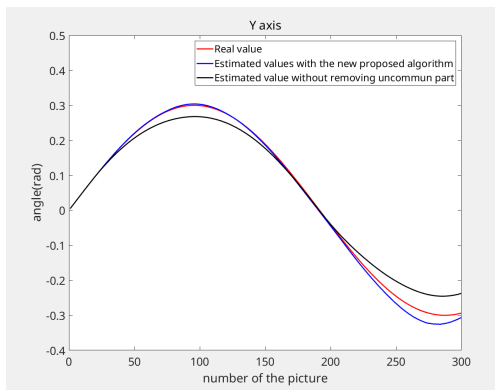
The spherical moment provides a tool for calculating the rotation of the camera, thus a quick introduction method is obtained. However, when calculating the spherical moment, there are non-common parts in the two images, which causes the inherent error of the algorithm itself. This paper improves it, calculates the rotation estimate, removes the non-common parts of the image, and iterates, greatly reducing or even eliminating this error. However, there are still some questions in the experiment that have not been answered. For example, in simulation 2, the final error of the x-axis is larger than the original old algorithm. The reason may be that the deviation between the rotation estimation and the real value in the first step is large, resulting in the subsequent iterative algorithm failing to converge to the real value.

#### V. CONCLUSION

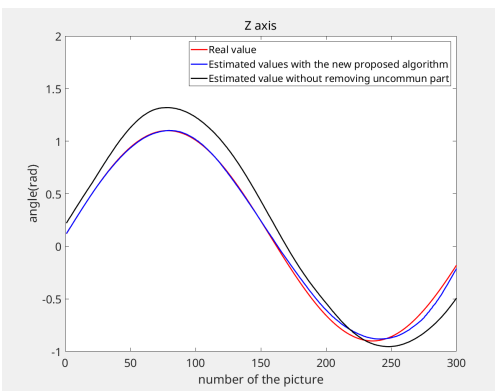
In this paper, an added optimization module of rotation estimation with spherical moment is proposed. Two simulations are carried. The results show that after removing



(a) X axis



(b) Y axis



(c) Z axis

Fig. 10. The result of simulation 2 with 300 images

the non-common parts of the two pictures, the error of the experimental results is significantly reduced, and the optimization module in this paper is efficient. And the execution speed of the algorithm is very fast, since matlab is used in the simulation instead of the C, there is much room for improvement in the speed of the program, which will meet the real-time requirements in most cases. However, there were still some errors. The future work will analyze the cause of the error, and further optimize and accelerate the algorithm.

## REFERENCES

- [1] DiSalvo C F, Gemperle F, Forlizzi J, et al. All robots are not created equal: the design and perception of humanoid robot heads[C]//Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques. 2002: 321-326.
- [2] Chatterji G B, Menon P K, Sridhar B. GPS/machine vision navigation system for aircraft[J]. IEEE Transactions on Aerospace and Electronic Systems, 1997, 33(3): 1012-1025.
- [3] Muad A M, Hussain A, Samad S A, et al. Implementation of inverse perspective mapping algorithm for the development of an automatic lane tracking system[C]//2004 IEEE Region 10 Conference TENCON 2004. IEEE, 2004: 207-210.
- [4] Hasegawa O, Kanade T. Type classification, color estimation, and specific target detection of moving targets on public streets[J]. Machine Vision and Applications, 2005, 16(2): 116-121.
- [5] Shen Y F, Rahman Z U, Krusienski D, et al. A vision-based automatic safe landing-site detection system[J]. IEEE Transactions on Aerospace and Electronic Systems, 2013, 49(1): 294-311.
- [6] Valette F, Ruffier F, Viollet S, et al. Biomimetic optical flow sensing applied to a lunar landing scenario[C]//2010 IEEE International Conference on Robotics and Automation. IEEE, 2010: 2253-2260.
- [7] Malamas E N, Petrakis E G M, Zervakis M, et al. A survey on industrial vision systems, applications and tools[J]. Image and vision computing, 2003, 21(2): 171-188.
- [8] Merino L, Wiklund J, Caballero F, et al. Vision-based multi-UAV position estimation[J]. IEEE robotics automation magazine, 2006, 13(3): 53-62.
- [9] De Castro, E.; Morandi, C. Registration of Translated and Rotated Images Using Finite Fourier Transforms. IEEE Trans. Pattern Anal. Mach. Intell. 1987, PAMI-9, 700-703.
- [10] Torr P H S, Zisserman A. Feature based methods for structure and motion estimation[C]//International workshop on vision algorithms. Springer, Berlin, Heidelberg, 1999: 278-294.
- [11] Funayama R, Jeong M H, Osugi M. Image processing apparatus for estimating motion of predetermined feature point of 3D object: U.S. Patent 7,809,166[P]. 2010-10-5.
- [12] Lee M S. Apparatus for estimating motion vectors for feature points of a video signal: U.S. Patent 5,760,846[P]. 1998-6-2.
- [13] Brox T, Malik J. Large displacement optical flow: descriptor matching in variational motion estimation[J]. IEEE transactions on pattern analysis and machine intelligence, 2010, 33(3): 500-513.
- [14] Xu L, Jia J, Matsushita Y. Motion detail preserving optical flow estimation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 34(9): 1744-1757.
- [15] Sun D, Roth S, Black M J. Secrets of optical flow estimation and their principles[C]//2010 IEEE computer society conference on computer vision and pattern recognition.
- [16] David G. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, pages 91-110, 2004.
- [17] Deng, X., Zhang, Z., Sintov, A., Huang, J., Bretl, T. (2018). Feature-constrained Active Visual SLAM for Mobile Robot Navigation. Proceedings - IEEE International Conference on Robotics and Automation.
- [18] Argyriou V, Vlachos T. Sub-pixel motion estimation using gradient cross-correlation[C]//Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings. IEEE, 2003, 2: 215-218.
- [19] Horn, B.k.;Schunck, B.G. Determining Optical Flow. Technical report, 1908.
- [20] Zhetao Zhang and Wanggen Wan. Dovo: Mixed visual odometry based on direct method and orb feature. 2018 International Conference on Audio, Language and Image Processing (ICALIP), pages 344-348, 2018.
- [21] Tahri, O. Application des Moments à L'asservissement Visuel et au Calcul de Pose. Ph.D. Thesis, Université de Rennes, Rennes, France, 2004.
- [22] Hadj-Abdelkader, H., Tahri, O., Benseddik, H. E. (2019). Rotation estimation: A closed-form solution using spherical moments. Sensors (Switzerland), 19(22).