Journal of Applied Science and Engineering A,
Volume 3, Issue 1, 2022, pp.01-10.
https://doi.org/10.5564/jasea.v3i1.2454

Article

# Motion and structure from the linear and non-linear algorithms

Tserennadmid Tumurbaatar[1]

[1]Department of Information and Computer Sciences, National University of Mongolia, Ulaanbaatar, Mongolia

*Corresponding author: tserennadmid@seas.num.edu.mn

**ABSTRACT.** This study presents the estimations of the 3D motion of a moving object in an image sequence taken from a monocular camera through linear and non-linear equations and determines the differences between linear and non-linear algorithms in terms of theoretical level and estimation accuracy with noisy point correspondences. Firstly, we investigated linear and non-linear algorithms for determining 3D motion at the theoretical level. Second, we estimated the 3D motion of the moving object in an image frame at two different instants of time with feature point correspondences in real time. Finally, we implemented an accuracy analysis of the results from the linear and non-linear estimations. We showed that the non-linear approach produced more accurate results than the linear approach from noisy point correspondences.

**Keywords:** image sequences, motion parameters, motion estimations, linear and non-linear equations

## 1 Introduction

Determining three-dimensional (3D) motion parameters from image sequences has been a challenging task in various applications for a long time. Motion parameters are determined through linear and non-linear algorithms by establishing point correspondences extracted from two or more views. Generally, non-linear algorithms solve the problem of non-linear least squares iteratively. Iterative methods may converge to a local but not global minimum with a good initial guess or may diverge at all. In a number of cases, linear algorithms have been formulated with eight or more correspondences. Linear algorithms solve linear equations that give a unique solution except in degenerate cases. In all practical situations, non-linear and linear algorithms fail to find the unique solution for many problems such as initialization, degenerate spatial configuration, noisy data, any relative motion between the camera and the scene.

Received: 03 March 2022;
Revised: 18 May 2022;
Accepted: 10 June 2022.

Various solutions and analyses of results from estimation of motion using the point correspondences have introduced sensitivity of their solutions to image noise, their robustness, computation efficiency, and accuracy against state-of-the-art methods for synthetic and real image data.

Among them, the linear algorithm for determining motion and structure for a planar surface using point correspondences between two images [1] [2] and minimal non-iterative relative pose solvers under planar motion constraint [3] were discussed. The effectiveness of the proposed algorithms was demonstrated with an assessment of the accuracy of the solutions with the synthetic data and real image sequences. They also indicated degenerate configurations in the presence of noise. Robustness of pose estimation related to measuring error of image coordinates [4] and sensitivity of linear solutions of the 3D motion estimation problem to image noise [5] were analyzed. Iterative algorithms for camera pose estimation problem [6] [7], robust registration of 2D to 3D points sets by using non-linear optimization [8], the efficient algorithmic solution to the classical five-point relative pose by using numerical algorithm [9], iterative pose estimation using coplanar feature points [10] and a two-step robust direct method for six-dimensional pose estimation [11] were developed for various applications. A new algebraic method to solve the perspective-three-point (P3P) problem [12], the numerical stability of the P3P estimation problem [13], decomposition algorithms to solve the P3P problem [14], a novel closed-form solution to the P3P [15], and a non-iterative solution to the perspective-n-point (PnP) problem for estimation of the pose of a calibrated camera from 3D-to-2D point correspondences [16] were presented. Analyses and results of the methods demonstrated that the precision and accuracy were comparable to the state-of-the-art methods. A closed-form solution to the least-squares problem of absolute orientation [17] and linear algorithms by decomposition [18] [19] was introduced. Experiments were carried out on comparing the performance of the algorithms with several existing algebraic and linear methods. The estimation errors were increased to the translation direction if the object's translation is small along the optical axis or the target object moves closer to the camera [20] [21]. The method to estimate 3D motion through non-linear least square equations for 2D to 3D point correspondences [23], a comparison of linear methods in computer vision and non-linear methods in photogrammetry for 2D to 2D point correspondences [25] were published by the authors of this study.

We have seen that no more results for investigating the differences in estimation accuracy between the linear approach using 2D to 2D point correspondences and the non-linear approach using 2D to 3D point correspondences are available from the previous analyses. In this paper, we analyzed how errors in the linear and non-linear estimation of motion parameters are related to the noisy correspondences. We used a single camera to estimate motion parameters since using a single camera has arisen for users in consumer electronics. Motion models are mathematically expressed with an epipolar constraint for all solutions under perspective geometry. First, we present mathematical formulations and general principles for linear and non-linear models. Next, we introduce the implementation steps for estimating motion parameters from image sequences. Third, we point out the differences in the solutions with the test

datasets at the experimental level and analyze the relationships between errors and noisy correspondences.

The paper is organized as follows. Motion models of the proposed solutions are presented in Section 2. Processing techniques to implement estimations are described in Section 3. The comparison results with test datasets are discussed in Section 4. The conclusions are summarized in Section 5.

## 2 Investigation of models for linear and non-linear algorithms

Consider an object viewed by a camera. Fig.1 shows a basic perspective geometry for the camera and object. Object space coordinate, $p = (x, y, z)$ at time $t1$ moves to $p' = (x', y', z')$ at time $t2$. Image space coordinates are denoted by $P = (X, Y, 1)$ and $P' = (X', Y', 1)$. The points $p$ and $p'$ on the surface of an object are projected at the points $P$ and $P'$ under perspective projection, respectively. Thus, the image point $p_i$ is at $P_i'$ whose coordinates are given by [24]

$$X_i' = \frac{x_i'}{z_i'} \text{ and } Y_i' = \frac{y_i'}{z_i'} \tag{1}$$

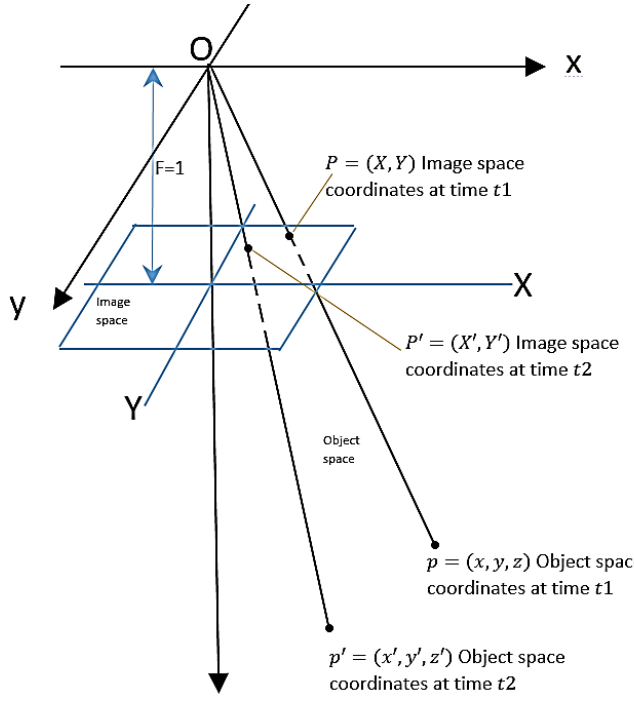


**Figure 1.** Perspective geometry for imaging.

Then, 3D coordinates $p_i'$ are related to $p_i$ by

$$\boldsymbol{p_i' = Rp_i + t} \tag{2}$$

Our goal is to estimate rotation $R$ and translation $t$ from the $N$ point correspondences. Firstly, combining (1) and (2), we get

$$\varepsilon = \sum_{i=1}^{N} \left\{ \left( X_i' - \frac{r_{11}x_i + r_{12}y_i + r_{13}z_i + t_x}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_z} \right)^2 + \left( Y_i' - \frac{r_{21}x_i + r_{22}y_i + r_{23}z_i + t_y}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_z} \right)^2 \right\} \tag{3}$$

Eq.(3) is nonlinear in the nine unknowns represented by three unknowns of the rotation matrix (ω, φ, κ), three unknowns of the translation ($t_x$, $t_y$, $t_z$) and three unknowns of 3D coordinates ($x, y, z$). This non-linear least square problem can be solved iteratively. To converge to the right solution, the estimation of these non-linear equations should start with a good initial guess. To overcome ill-conditioned convergence problem of these non-linear equations, good initial values were provided from a para-perspective projection model using 2D to 2D point correspondences developed by Tomasi and Kanade [22]. After estimation of initialization, the initial values of six motion parameters and 3D object-space coordinates ($x, y$, and $z$) are provided, then equation (3) will be solved to find motion parameters as presented in detail [23]. Secondly, given eight or more points correspondences, a linear algorithm is devised by reformulating Eq.(2) under coplanarity conditions [24].

$$(P')^T EP = 0 \text{ and } \begin{bmatrix} X' & Y' & 1 \end{bmatrix} E \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = 0 \tag{4}$$

where $E$ is a $3x3$ matrix defined as

$$E = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} \tag{5}$$

Eq.(4) is linear and homogenous in the nine unknowns. Once $E$ is determined, $R$ and $t$ can be determined uniquely. In practice, since the point correspondences may be inaccurate and noisy, RANSAC (Random sample consensus) can be weed out these outliers to obtain a good solution. In practice, when the point correspondences are noisy, various least-squares techniques can be used.

## 3 Methodology

The implementation steps of the non-linear motion model are described in Fig.2. We assume N feature points in a template region extracted from the first frame and track them to the next each F frame ($p_{f1} \dots p_{fn} | f = 1, \dots, F, n = 1, \dots, N$). In non-linear case, we need at least three or more frames for determining 3D motion of the moving object due to the factorization process [22]. In other words, motion parameters are estimated by using the corresponding points between a template region at time $t1$ and the fifth frame at time $t2$.

Firstly, we extract the whole or parts of the moving object as a template region. Secondly, SIFT feature extractor can be used to extract the N feature points for the template region. Thirdly, once capturing a new frame, we compute the feature points for this frame. The corresponding points ($p_{f1} \dots p_{fn}$) between the new frame and the template region are calculated with a Brute Force matcher. The outlier corresponding points are eliminated by RANSAC based robust method. The steps as mentioned before can be repeated for five consecutive frames.

We used the corresponding points for entries of the $2F \times N$ measurement matrix W to calculate initial values of the Eq.(3) through the para-perspective factorization method [23]. The overdetermined $2N$ equations for six unknown variables ($t_{xf}, t_{yf}, t_{zf}, \omega, \varphi, \kappa$) are solved from the given corresponding points in the camera coordinate system for each frame.

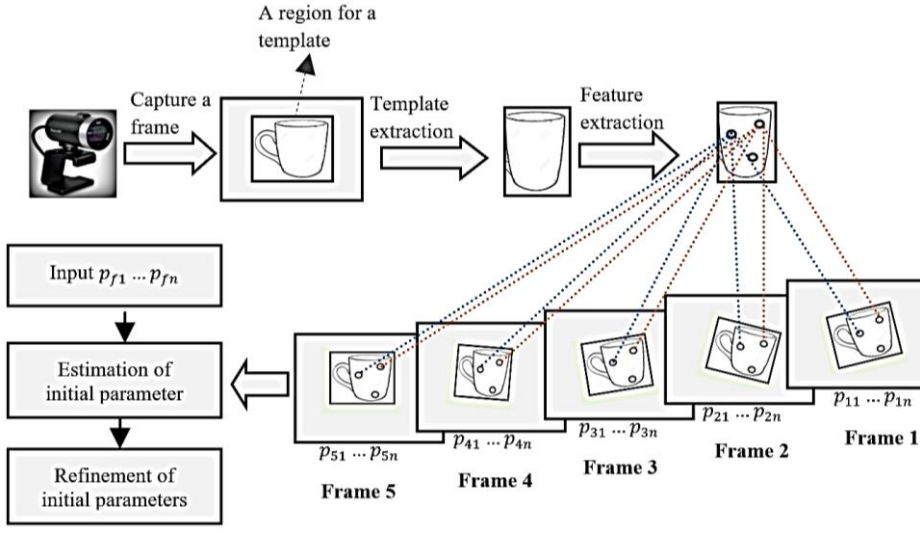


**Figure 2.** Process flow of the non-linear motion model

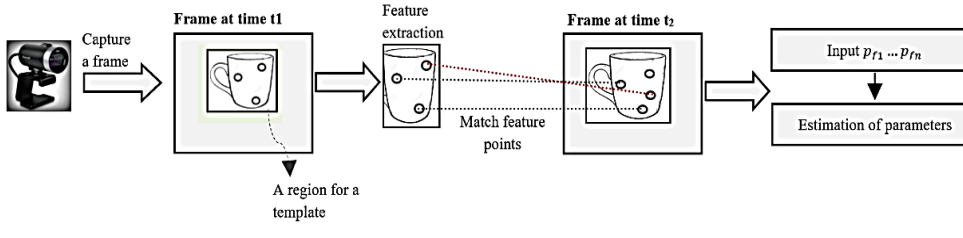


**Figure 3.** Process flow of the linear motion model [25]

Now, we present the implementation steps of the linear motion model for determining 3D motion parameters as described in Fig.3. In the linear case, motion parameters are determined by using corresponding points between a template region at time $t1$ and the second frame at time $t2$.

Firstly, after extracting the template region from the first frame, the feature points for the template are computed by using SIFT feature extractor. Secondly, once a new frame is captured, the feature points is extracted for this frame. The corresponding points between the template and the new frame are calculated with a Brute Force matcher by eliminating wrong matches with the RANSAC based method. Thirdly, motion parameters are estimated after accumulating all processing steps for two consecutive frames.

## 4 Experiment analysis

In this section, we discuss the performances and results of the 3D motion estimation from the non-linear and linear algorithms. We performed the experiments with the Intel(R) Core (TM) i7, CPU 3.0 GHz, 12MB RAM computer, and a Microsoft LifeCam. We used the C++ programming language with Visual Studio programming tool, OpenCV library, and OpenGL graphic library. The 320x240 video image sequences are taken from the single calibrated camera. We examined motion results for a real dataset created from real image sequences and a synthetic dataset created from the OpenGL library. The corresponding points from the synthetic sequences were prepared with known motion parameters.

Firstly, we draw a cube in virtual space by choosing the largest focal length for keeping it in the field of view throughout sequences under perspective projection. The edge length of the cube is ten. We created the synthetic sequences by rotating the cube at 20 degrees around the $x$ and $y$ axes. The cube's rotation around the $z$ axis is varied up to 90 degrees. We randomly generated 25 points within cube without noise.

We estimated the motion parameters through Eq.(3) (non-linear approach) and Eq.(4) (linear approach) for each synthetic image sequence. The synthetic dataset consists of thousands of frames. We compared the estimated rotation parameters around the $x$, $y$, and $z$ axes with known rotation parameters to check the accuracy of the results. Also, we computed their minimum and maximum errors, Absolute Mean Error (ME) and Root Mean Square Error (RMSE), as summarized in Table 1.

Secondly, while changing the object's position in front of the static camera, we captured thousands of image sequences for preparing a real dataset. The object's position is fixed arbitrarily in the first frame of the moving object. The object moves $300\,mm$ closer to the camera and $600\,mm$ away from the camera in the z axis direction. The object is translated by up to $200\,mm$ in the x and y axes direction. We rotated the object around the z-axis by 90 degrees and the x and y axes by 20 degrees.

We compared the estimated rotation parameters around the $x$, $y$, and $z$ axes with known rotation parameters to check the accuracy of the results. Also, we computed their minimum and maximum errors, Absolute Mean Error (ME) and Root Mean Square Error (RMSE), as summarized in Table 2.

**Table 1.** Comparison of error analysis for rotations around $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ axis for the synthetic dataset

| Method | Non-linear | Linear |
|---|---|---|
| **Comparison of Maximum Error /degrees/** | | |
| $\omega = 1^{\circ}$ *to* $20^{\circ}$ | 1.647 | 1.615 |
| $\varphi = 1^{\circ}$ *to* $20^{\circ}$ | 1.986 | 1.891 |
| $\kappa = 1^{\circ}$ *to* $90^{\circ}$ | 1.937 | 1.828 |
| **Comparison of Minimum Error /degrees/** | | |
| $\omega = 1^{\circ}$ *to* $20^{\circ}$ | 0.015 | 0.009 |
| $\varphi = 1^{\circ}$ *to* $20^{\circ}$ | 0.005 | 0.01 |
| $\kappa = 1^{\circ}$ *to* $90^{\circ}$ | 0.014 | 0.016 |
| **Comparison of Mean Error /degrees/** | | |
| $\omega = 1^{\circ}$ *to* $20^{\circ}$ | 0.549 | 0.498 |
| $\varphi = 1^{\circ}$ *to* $20^{\circ}$ | 0.648 | 0.647 |
| $\kappa = 1^{\circ}$ *to* $90^{\circ}$ | 0.684 | 0.556 |
| **Comparison of RMS Error /degrees/** | | |
| $\omega = 1^{\circ}$ *to* $20^{\circ}$ | 0.68 | 0.632 |
| $\varphi = 1^{\circ}$ *to* $20^{\circ}$ | 0.818 | 0.801 |
| $\kappa = 1^{\circ}$ *to* $90^{\circ}$ | 0.811 | 0.645 |

**Table 2.** Comparison of error analysis for rotations around $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ axis for real scene

| Method | Non-linear | Linear |
|---|---|---|
| **Comparison of Maximum Error /degrees/** | | |
| $\omega = 1^{\circ}$ *to* $20^{\circ}$ | 1.867 | 1.993 |
| $\varphi = 1^{\circ}$ *to* $20^{\circ}$ | 1.59623 | 1.731 |
| $\kappa = 1^{\circ}$ *to* $90^{\circ}$ | 0.892 | 0.757 |
| **Comparison of Minimum Error /degrees/** | | |
| $\omega = 1^{\circ}$ *to* $20^{\circ}$ | 0.00014 | 0.000 |
| $\varphi = 1^{\circ}$ *to* $20^{\circ}$ | 0.00045 | 0.001 |
| $\kappa = 1^{\circ}$ *to* $90^{\circ}$ | 0.0051 | 0.00 |
| **Comparison of Mean Error /degrees/** | | |
| $\omega = 1^{\circ}$ *to* $20^{\circ}$ | 0.538 | 0.564 |
| $\varphi = 1^{\circ}$ *to* $20^{\circ}$ | 0.346 | 0.403 |
| $\kappa = 1^{\circ}$ *to* $90^{\circ}$ | 0.391 | 0.312 |
| **Comparison of RMS Error /degrees/** | | |
| $\omega = 1^{\circ}$ *to* $20^{\circ}$ | 0.685 | 0.717 |
| $\varphi = 1^{\circ}$ *to* $20^{\circ}$ | 0.346 | 0.403 |
| $\kappa = 1^{\circ}$ *to* $90^{\circ}$ | 0.434 | 0.364 |

As we see in Table 1, the linear approach is more accurate than the non-linear approach for accurate feature correspondences. Also, we can observe that the linear approach is more sensitive for noisy measurements of feature correspondences than the non-linear approach, as shown in the most results in Table 2.

Generally, we see in Table 1-2 that the two estimation methods produced negligible errors in both synthetic and real dataset. It is also proved that large perspective changes don't affect the accuracy of the estimated motion parameters for all proposed methods with accurate point correspondences. Finally, we can conclude that the non-linear

approach produces more accurate results for real dataset. The linear approach produces more accurate results for the 3D object in synthetic dataset.

## 5 Conclusion

In this study, we showed two approaches to determine the motion parameters of the moving object from point correspondences. We estimated the motion parameters by using corresponding points between the template region and subsequent frames. We can try the linear algorithms if we have eight or more correspondences. Solving non-linear equations is viable if we have a good initial guess solution. In the case of six or more correspondences, the non-linear solution is generally unique. Good initial values of these non-linear equations are estimated from the para-perspective projection model. The results of both approaches were accurate in large changes of the translation and rotation. In particular, the non-linear approach is produced negligible error with noisy point correspondences. The linear approach is more robust to estimate the motion parameters from accurate point correspondences.

## Acknowledgments

The work in this paper was supported by the National University of Mongolia (No. P2019-3714 ).

## References

1. Weng Juyang, N. Ahuja and T. S. Huang, Motion and structure from point correspondences: a robust algorithm for planar case with error estimation, 9th International Conference on Pattern Recognition, 1 (1988), pp. 247-251, https://doi.org/10.1109/ICPR.1988.28215
2. J. Weng, N. Ahuja and T. S. Huang, Motion and structure from point correspondences with error estimation: planar surfaces, in IEEE Transactions on Signal Processing, 39-12, (1991) pp. 2691-2717, https://doi.org/10.1109/78.107418
3. Sunglok Choi, Jong-Hwan Kim, Fast and reliable minimal relative pose estimation under planar motion, Image and Vision Computing, 69 (2018), pp. 103-112, https://doi.org/10.1016/j.imavis.2017.08.007
4. Yingming Hao, Feng Zhu, Jinjun Ou, Qingxiao Wu, J. Zhou and Shuangfei Fu, Robust analysis of P3P pose estimation, IEEE International Conference on Robotics and Biomimetics (ROBIO), Sanya, (2007), pp. 222-226, https://doi.org/10.1109/ROBIO.2007.4522164
5. A. Ansar and K. Daniilidis, Linear pose estimation from points or lines, in IEEE Transactions on Pattern Analysis and Machine Intelligence, 25-5, (2003), pp. 578-589, https://doi.org/10.1109/TPAMI.2003.1195992
6. Shiqiang Hu, Minzhe Li, Single view based nonlinear vision pose estimation from coplanar points, Optik, 2020.

7. Maoteng Zheng, Shiguang Wang, Xiaodong Xiong, Junfeng Zhu, A fast and accurate iterative method for the camera pose estimation problem, Image and Vision Computing, 94, (2020), https://doi.org/10.1016/j.imavis.2019.103860
8. A.W. Fitzgibbon, Robust registration of 2D and 3D point sets, Image and Vision Computing, 21 (2003), pp.1145–1153, https://doi.org/10.1016/j.imavis.2003.09.004
9. D. Nister, An efficient solution to the five-point relative pose problem, IEEE Transactions on Pattern Analysis and Machine Intelligence, 26-6 (2004), pp. 756–770, https://doi.org/10.1109/TPAMI.2004.17
10. D. Oberkampf, D.F. Dementhon, and L.S. Davis, Iterative pose estimation using coplanar feature points, Computer Vision and Image Understanding, 63-3 (1996), pp. 495–511, https://doi.org/10.1006/cviu.1996.0037
11. Po-Chen Wu, Hung-Yu Tseng, Ming-Hsuan Yang, Shao-Yi Chien, Direct pose estimation for planar objects, Computer Vision and Image Understanding, 172 (2018), pp. 50-66, https://doi.org/10.1016/j.cviu.2018.03.006
12. Ping Wang, Guili Xu, Zhengsheng Wang, Yuehua Cheng, An efficient solution to the perspective-three-point pose problem, Computer Vision and Image Understanding, 166 (2018), pp.81-87, https://doi.org/10.1016/j.cviu.2017.10.005
13. B.M. Haralick, CN. Lee, K. Ottenberg, et al., Review and analysis of solutions of the three-point perspective pose estimation problem, International Journal of Computer Vision, 13 (1994), pp. 331-356, https://doi.org/10.1007/BF02028352
14. Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang and Hang-Fei Cheng, Complete solution classification for the perspective-three-point problem, in IEEE Transactions on Pattern Analysis and Machine Intelligence, 25-8, (2003), pp. 930-943, https://doi.org/10.1109/TPAMI.2003.1217599
15. 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, In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, pp. 2969–2976, https://doi.org/10.1109/CVPR.2011.5995464
16. V. Lepetit, F. Moreno-Noguer, and P. Fua, EPnP: An accurate O(n) solution to the PnP problem, International Journal of Computer Vision, 81-2 (2009), pp. 155– 166, https://doi.org/10.1007/s11263-008-0152-6
17. B. Horn, Closed-form solution of absolute orientation using unit quaternions, Journal of the Optical Society of America, JOSA A, 4-4 (1987), pp. 629– 642, https://doi.org/10.1364/JOSAA.4.000629
18. M.A. Ameller, B. Triggs, and L. Quan, Camera pose revisited-new linear algorithms, In European Conference on Computer Vision, 2000.
19. R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2001.
20. J. Weng, T. S. Huang and N. Ahuja, Motion and structure from two perspective views: algorithms, error analysis, and error estimation, in IEEE Transactions on Pattern Analysis and Machine Intelligence, 11-5 (1989), pp. 451-476, https://doi.org/10.1109/34.24779
21. C.P. Lu, G.D. Hager, and E. Mjolsness, Fast and globally convergent pose estimation from video images, IEEE Trans. on Pattern Analysis and Machine Intelligence, 22 (2000), pp. 610–622, https://doi.org/10.1109/34.862199
22. C. J. Poelman and T. Kanade, A para-perspective factorization method for shape and motion recovery, in IEEE Transactions on Pattern Analysis and Machine Intelligence, 19-3 (1997), pp. 206-218, https://doi.org/10.1109/34.584098
23. T. Tumurbaatar and T. Kim, Development of real-time object motion estimation from single camera, Spat. Inf. Res., 25(2017), pp. 647-656, https://doi.org/10.1007/s41324-017-0130-6.

24. T. S. Huang and A. N. Netravali, Motion and structure from feature correspondences: A review, in Proceedings of the IEEE, 82-2 (1994), pp. 252-268, https://doi.org/10.1109/5.265351
25. Tumurbaatar Tserennadmid, Kim Taejung, Comparative Study of Relative-Pose Estimations from a Monocular Image Sequence in Computer Vision and Photogrammetry. Sensors, 19-8, (2019), pp.1905, https://doi.org/10.3390/s19081905