Combining Kinect and PnP for Camera Pose Estimation

Shu Zhang[1, 2], Hui Yu* [2] , Junyu Dong[1] , Ting Wang[1, 2] , Lin Qi[1] , Honghai Liu[2]
[1] Ocean University of China, Qingdao, China
[2] University of Portsmouth, Portsmouth, UK
hi.shu.z@outlook.com
hui.yu@port.ac.uk

***Abstract*—This paper presents a novel method to conduct camera pose estimation though combining Kinect and Perspective-n-points algorithms. Most existing camera pose estimation methods suffer from the errors caused by inevitable outliers between 2D-3D correspondences. To this end, we propose to use a random down sampling process to deal with outliers in this paper. The proposed method is divided into two main steps, which are 2D-3D correspondences generation and pose estimation. The method has been tested in a real project, and the experiment has shown encouraging results compared to the ground truth.**

Keywords—pose estimation; Kinect; PnP; feature matching

I. INTRODUCTION

Camera pose estimation is a widely used technique for multi-camera related applications. The main objective of the estimation is to recover the Rotation and Translation of one camera in a certain coordinate system from its image. There are three kinds of implementation for this estimation process, which are the ones based on direct linear transformation (DLT), perspective-n-points (PnP), and a priori information estimator. Due to its higher robustness, PnP has wider users than others. However, to achieve accurate estimation results, PnP requires a set of high accurate 2D-3D correspondences between 2D points in camera image and 3D points in the space, which would be obtained with inevitable outliers that could leads to potential errors.

In this paper, a robust method for camera pose estimation is presented for dealing with outliers. The proposed method consists of a robust 2D-3D correspondence generator, iterated camera pose estimator. We employ Kinect into implementation for 3D points sensing, and perspective-n-points as the core of the estimator. Through experiments, the proposed method is tested to be robust for different situations.

II. BACKGROUND

A. Kinect

In recent years, 3D depth camera draws more and more attentions to researchers due to its versatile applications in computer vision, such as stereo cameras and Time of Flight (TOF) cameras [1]. One of the best in these 3D depth sensors is Microsoft's Kinect [2]. Kinect was designed for human machine interaction in a game environment in the first place. However, the characteristics of the data captured by Kinect, especially 3D depth information acquired, have also attracted the researchers in computer vision community. Kinect is actually an RGB-D sensor which provides synchronized RGB color and depth images. The RGB color image is captured by a normal camera built in to the Kinect, while the ability of depth sensing is achieved by an infrared laser projector and an infrared video camera mounted within the Kinect, as shown in Fig. 1. The system uses the infrared camera to detect a speckle pattern projected onto objects in the Kinect's field of view (FOV). By measuring deformations in the reference speckle pattern, Kinect can recover the 3D depth information of objects [3]. The experimental results have shown that Kinect is more accurate than the TOF depth sensor, and close to a medium-resolution stereo camera [4].

Though the RGB image and Depth image are captured simultaneously, there is a spatial shift between two captured images by the normal camera and the infrared camera due to their location difference [3]. And moreover the raw depth data is noisy, and usually contains zero depth (holes). Therefore, many Kinect based systems begin with a preprocessing that conducts RGB and depth spatial alignment, or depth data filtering.

Fig. 1. The sensor of Kinect 360.

B. *Perspective-N-Points*

The main objective of camera pose estimation based on perspective-n-points, referring to as PnP problem, is to recover the relative position between camera and the origin of a certain coordinate system from n known correspondences of 3D points in space and 2D points in image. There is a number of implementation for the PnP estimation problem. A distance based definition was first proposed by Fisher in 1981 [5], and Horaud, Conio, and Leboulleux in 1989 [6] presented the transformation based definition of PnP problem. The later PnP problem solutions are mostly based on these two kinds of definitions. The minimal corresponding points' number is 3, which make the minimal PnP problem to be P3P problem. Gao [7] and Kneip [8] discussed the P3P solutions in their literatures respectively. In practical, P3P often suffers from the instability due to the outliers. As a common solution, most of existing researches of PnP using redundancy of the 2D-3D correspondence to improve the accuracy.

Among the whole range of PnP implementations, a solution called Efficient Perspective-n-Points (EPnP) was presented by Lepetit, Moreno-Noguer and Fua [9] in 2009. They speeded up the estimation process with a computational complexity of O (n) for n ($n \geq 4$) points of 2D-3D correspondences. The efficiency of the implementation is achieved by representing the 3D points in space as a weighted sum of m ($m \leq 4$) actual 3D points in the space, and process all following calculation only on these sum weighted 3D points. By solve the parameterized quadratic equations to obtain the estimated solution within a linear time consumption over a number of 2D-3D corresponding point pairs. This linear algebra techniques based estimation method has made a trade-off between speed and accuracy.

III. PROPOSED METHOD

In this paper, a camera pose estimation method is proposed by combining the Kinect and EPnP algorithm. We choose Kinect because it could provide us with 3D point cloud of the scene, which will be useful for the calculation of the Perspective-n-Points algorithm. Our method is tested in the real project which tracks human's face orientation across different cameras. The experiments with encouraging results is showed in the section V.

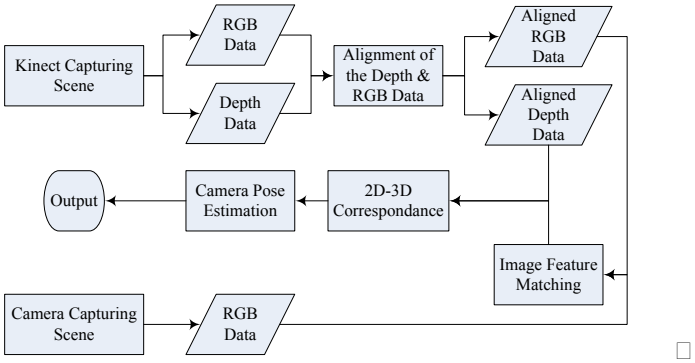


Fig. 2. The illustration of work flow of the proposed method.

A. *Preparation of the Corresponding 3D Coordinates*

The first step for camera pose estimation is to find the 2D-3D correspondence between the 2D points in the camera image and the 3D points in the space. Because Kinect can generate both RGB image and depth image, the 2D-3D correspondence can be done through an intermediate step of 2D-2D correspondence between camera RGB image and Kinect RGB image. And then the

relationship between points in Kinect RGB image and Kinect Depth image will provide the 2D-3D correspondence mentioned above.

In general, 2D-2D correspondence can be achieved by feature matching. In this paper, the feature matching algorithm by Orb [10] is employed as both feature detector and descriptor due to its fast calculation speed. And brute force matching is used for feature matching with distance determined by Hamming distance, which is chosen also because of its computational timesaving. However, there would be uncertainties of mismatched features in the matching results, which present errors to the following-up process. Therefore, we introduce Fundamental Matrix into our method to do the filtering work for outliers of the matching.

Fundamental Matrix is a 3 by 3 matrix that refers to the projection relationship between two images that captured by two cameras with different poses. Every pair of 2D points in two images respectively that projected from same one 3D point in the space should be matched as a corresponding pair. And Fundamental Matrix implies the fact that this pair of 2D points should lie in a same plane, which is called epipolar plane. The intersections of this plane and two image planes are called epipolar lines, which are represented by the Fundamental Matrix. Therefore, the estimation of the Fundamental Matrix will guide the feature matching with a better accuracy. However, this guidance cannot exactly provide the rule of truly matching due to that the Fundamental Matrix is also estimated based on the pre-matched points. Therefore, we applied a randomly down sampling of the matched point-pairs filtered by estimated Fundamental Matrix. And another Fundamental Matrix estimated based on the previous filtered point-pairs is used to perform further filtering.

As a result, the whole 2D-2D feature matching process is divided into three steps: (a) feature extraction using ORB as both detector and descriptor, and matching using Hamming distance; (b) filtering the matching results by the fundamental matrix calculated using the results obtained in step (a); (c) Randomly down sampling the matched feature pairs from the results in step (b) to a predefined number of feature pairs count, and filtering them again with fundamental matrix calculated by the random samples. After the step (c), a feature matching results with much better accuracy can be achieved, as shown in Fig. 3.

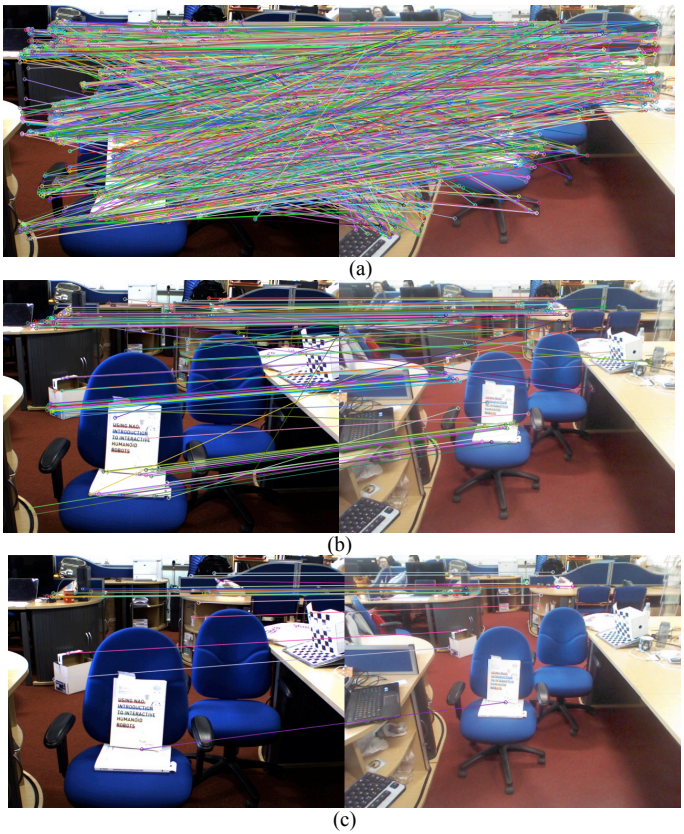


Fig. 3. The illustration of the feature matching filtering process. (a) The results of using ORB feature matching alone; (b) The results that filtered by Fundamental Matrix for the first time from the results in (a). They are more accurate than the results in (a); (c) The results that filtered by Fundamental Matrix for the second time from the random samples of the results in (b). They have the best accuracy in all steps.

In the meanwhile, the process of alignment between the RGB image and the depth image both generated from Kinect is carried out. As we mentioned in section II, the shift of the location of the different sensors causes a shift between RGB image and Depth Image. This presents an obstacle for searching from 3D points in space to 2D points in the camera image, which has 2D-2D correspondence to Kinect RGB image. This could be solved by taking into account of the constant distance between the RGB sensor and the Infrared sensor in the Kinect device. With the knowledge of field of view (FOV) of the Kinect, we can modify every

pixel in the depth image accordingly to make them aligned with the pixel in RGB image. After alignment, for every coordinate of 2D point in RGB image, we can retrieve the corresponding 2D coordinate in Depth image. Then coordinate of 3D point in the space can be obtained with the (1).

$$\frac{x_p}{u-u_0} = \frac{y_p}{v-v_0} = \frac{z_p}{f} \tag{1}$$

Where u_0, v_0 being the depth image center of the Kinect, f being the focal length of the infrared camera. Then x_p, y_p, z_p will be the 3D coordinate of a point in the space corresponding to the 2D point of (u, v) in the depth image. The alignment

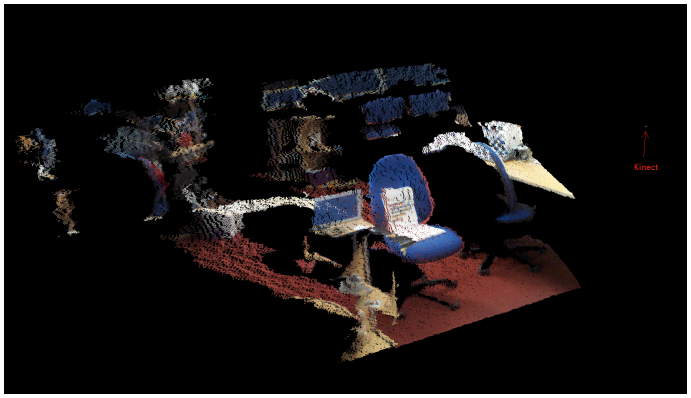


Fig. 4. The illustration of the point cloud obtain with Kinect after RGB image and Depth image aligned. The Kinect is in front of the chair.

result of RGB image and Depth image is illustrated in the Fig. 4.

B. Camera Pose Estimation

When 2D-3D correspondence is found, the next process is the estimation of the camera pose. In our method, this process is mainly based on an iterative process. In every loop of iterations, a Perspective-n-Points (PnP) algorithm is applied along with the 2D-3D correspondence calculated by the previous process. There a range of PnP algorithm implementations in the community. We choose Efficient Perspective-n-Points (EPnP) according to its high efficiency in calculation. EPnP algorithm is an O (n) non-iterative process in the first place. We put it into a sequence of loops because that the main process of the PnP algorithm is about parameterization and quadratic equations solving, which will also bring in errors when outliers are inputted. To minimize this, in each loop of the iteration, we firstly apply the EPnP algorithm with the 2D-3D correspondences. And then a projection process from every 3D point in space to 2D points is conducted with the estimated camera rotation and translation in current loop. By comparing the projected 2D points and the true 2D points in the camera image, the outliers of the 2D-3D pairs can be counted. If the number of outliers is above a threshold as a predefined value, such as the 40% of the total number of the point-pairs in our experiment, then randomly down sample the 2D-3D point pairs to a predefined number of count, such as the 60% of the total number of the point-pairs in our experiment. After randomly down sampling, begin the next loop. If the outliers is below the threshold, or the total count of the loop is above a predefined number, the iteration should end, and the final results of the camera pose can be outputted.

IV. EXPERIMENTS

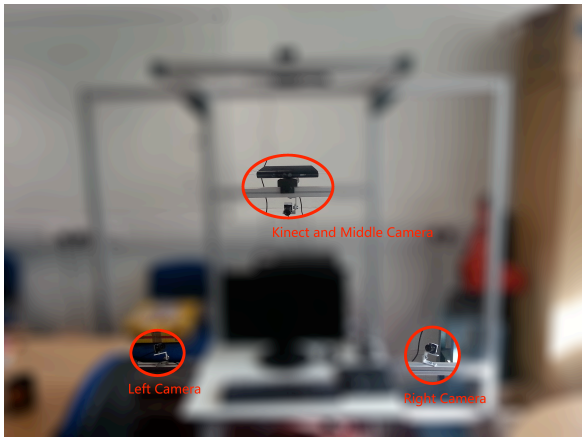
The proposed method is tested in a real application project which consists of multiple cameras and Kinect. The origin of the global coordinate is located in the principle point of the Kinect. It's a right hand coordinate system while the z axis comes out from Kinect face. The project requires the knowledge of every camera's pose in the global coordinate. However, every time the devices are assembled together, the related pose of every camera varies. In order to obtain every pose of the camera after assembling, method proposed in this paper is applied.

In practical, our method has successful calculated the cameras' poses in different situations. As shown in Fig. 5, different camera has different relative pose to the Kinect. The left camera is located in the left bottom of the Kinect. The right camera is mounted in the right bottom of the Kinect. And the middle camera is seated in the near bottom of the Kinect. The estimations show encouraging results compared to the ground truth.

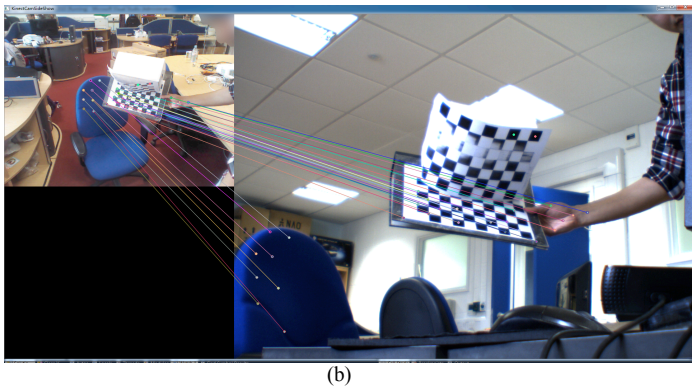
TABLE I. THE RESULTS OF PROPOSED METHOD

	Rotation Matrix	Translation Matrix
Proposed Method		
Right Camera	$\begin{bmatrix} -0.7708 & -0.2553 & -0.5837 \\ -0.1778 & -0.7936 & 0.5819 \\ -0.6118 & 0.5522 & 0.5663 \end{bmatrix}$	$\begin{bmatrix} 538.5903 \\ -608.0502 \\ 525.1226 \end{bmatrix}$
Middle Camera	$\begin{bmatrix} -0.9997 & 0.0191 & -0.0172 \\ -0.0210 & -0.9927 & 0.1188 \\ -0.0148 & 0.1192 & 0.9928 \end{bmatrix}$	$\begin{bmatrix} 13.1757 \\ -124.8214 \\ -145.3288 \end{bmatrix}$
Left Camera	$\begin{bmatrix} -0.7742 & 0.3490 & 0.5279 \\ 0.2347 & -0.6163 & 0.7517 \\ 0.5877 & 0.7059 & 0.3953 \end{bmatrix}$	$\begin{bmatrix} -386.8028 \\ -632.9952 \\ 725.2192 \end{bmatrix}$
Ground Truth [a]		
Right Camera	$\begin{bmatrix} -0.7823 & -0.2098 & -0.5865 \\ -0.2038 & -0.8036 & 0.5592 \\ -0.5887 & 0.5570 & 0.5859 \end{bmatrix}$	$\begin{bmatrix} 542.5367 \\ -612.5463 \\ 520.4527 \end{bmatrix}$
Middle Camera	$\begin{bmatrix} -0.9911 & 0.1318 & -0.0192 \\ -0.1331 & -0.9845 & 0.1144 \\ -0.0038 & 0.1159 & 0.9932 \end{bmatrix}$	$\begin{bmatrix} 11.4364 \\ -120.6453 \\ -139.8563 \end{bmatrix}$
Left Camera	$\begin{bmatrix} -0.7663 & 0.3017 & 0.5672 \\ 0.3048 & -0.6064 & 0.7344 \\ 0.5655 & 0.7357 & 0.3727 \end{bmatrix}$	$\begin{bmatrix} -391.5437 \\ -629.2484 \\ 728.2386 \end{bmatrix}$

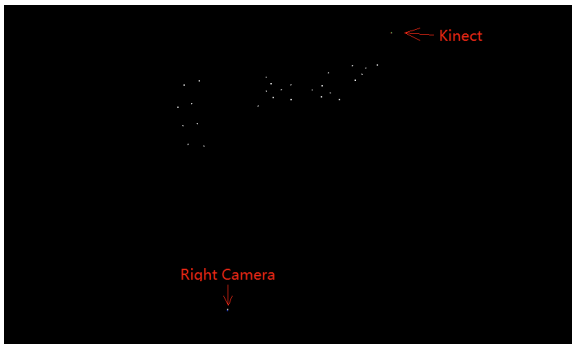
[a.] The ground truth is calculated by handy measurements of the devices along three axes, and Rotation matrix is obtained with Rodrigues transformation of the measurements afterwards.



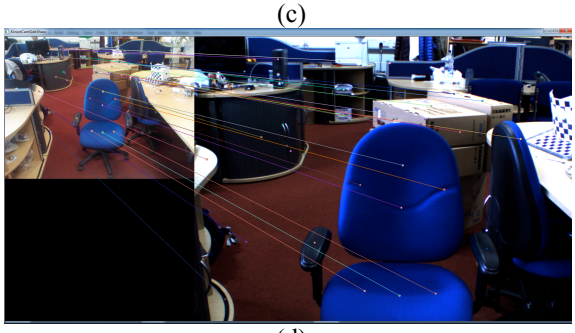
(a)



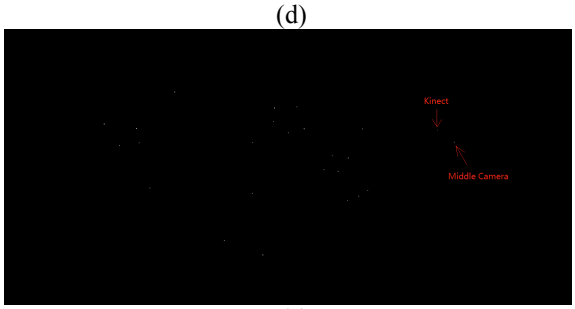
(b)



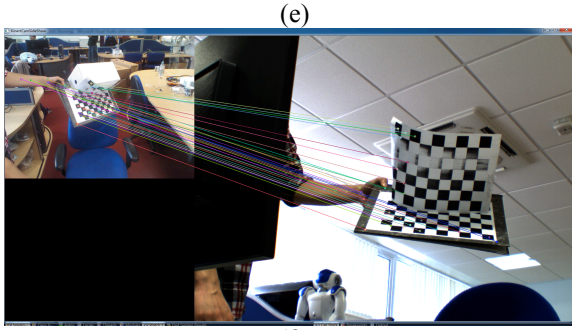
(c)



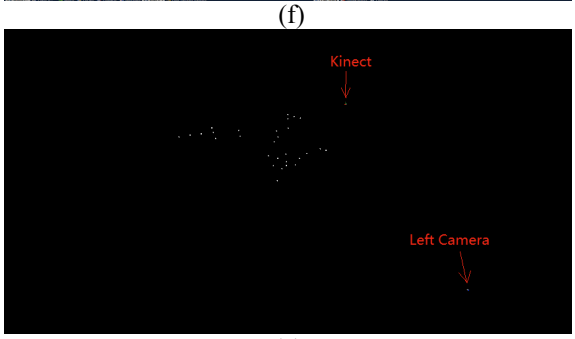
(d)



(e)



(f)



(g)

Fig. 5. The experiments results with proposed method, where (b) and (c) is for Right Camera in (a); (d) and (e) is for Middle Camera in (a); (f) and (g) is for Left Camera in (a).

According to Table 1, three cameras' rotations and translations calculated by proposed method are very close to the ground truth, which is obtained by manually measuring the positions and orientations for three cameras in the global coordinate system respectively. As shown in Fig. 5, recovered relationships between three cameras and Kinect could be a good representation of the true situation from the reality.

Since the estimation process is achieved by optimization, it only results in local minima. The errors in Table 1 are mainly caused by the input of the optimization, which is the 2D-3D correspondence in our method. The better the input provides to the optimization process, the closer the initial position is to the global minima in the optimization process, which will transfer the local minima to the global minima to achieve a higher accuracy. Therefore, high accurate 2D-3D correspondence is crucial to the estimation process. The iterations with Randomly Down Sampling process discussed in this paper are aimed at removing outliers in 2D-3D correspondence as many as possible to provide better input to the optimization process. It thusleads to lower errors in the outputs of the camera pose estimation.

V. Conclusion

In this paper, we have presented a novel method for camera pose estimation including multi-steps feature matching for 2D-3D correspondence, and an iterated estimation process with randomly down sampling. In order to minimize the number of outliers in the 2D-3D correspondences, a multi-step with randomly down sampling and fundamental matrix guided filtering is applied to the 2D-2D matching process. With the alignment of the RGB image and Depth image of the Kinect, 2D-3D correspondence can be obtained with as less outliers as possible. The iterated process is invited with PnP estimation and re-projection check combined to minimize the error brought in by outliers. The experiment results show encouraging outputs for very different poses of cameras comparing to ground truth.

Acknowledgment

This work was supported by EU seventh framework programme under grant agreement No. 611391, Development of Robot-Enhanced Therapy for Children with Autism Spectrum Disorders (DREAM); International Science & Technology Cooperation Program of China (ISTCP) (N0. 2014DFA10410); and the Ph.D. Program Foundation Of Ministry Of Education Of China (No. 20120132110018).

References

[1] Gokturk, S. B., Yalcin, H., & Bamji, C. (2004, June). A time-of-flight depth sensor-system description, issues and solutions. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on* (pp. 35-35). IEEE

[2] Zhang, Z. (2012). Microsoft kinect sensor and its effect. *MultiMedia, IEEE*, *19*(2), 4-10.

[3] Han, J., Shao, L., Xu, D., & Shotton, J. (2013). Enhanced computer vision with microsoft kinect sensor: A review. *Cybernetics, IEEE Transactions on*, *43*(5), 1318-1334.

[4] Smisek, J., Jancosek, M., & Pajdla, T. (2013). 3D with Kinect. In *Consumer Depth Cameras for Computer Vision* (pp. 3-25). Springer London.

[5] Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381-395.

[6] Horaud, R., Conio, B., Leboulleux, O., & Lacolle, B. (1989, June). An analytic solution for the perspective 4-point problem. In *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR'89., IEEE Computer Society Conference on* (pp. 500-507). IEEE.

[7] Gao, X. S., Hou, X. R., Tang, J., & Cheng, H. F. (2003). Complete solution classification for the perspective-three-point problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *25*(8), 930-943.

[8] Kneip, L., Scaramuzza, D., & Siegwart, R. (2011, June). A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (pp. 2969-2976). IEEE.

[9] Lepetit, V., Moreno-Noguer, F., & Fua, P. (2009). Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, *81*(2), 155-166.

[10] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, November). ORB: an efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (pp. 2564-2571). IEEE.