A New 2-Point Absolute Pose Estimation Algorithm under Plannar Motion

Sung-In Choi and Soon-Yong Park
School of Computer Science & Engineering, Kyungpook National University,
1370 Sankyuk-Dong, Puk-gu, Daegu, 702-701, Korea

Keywords: Visual Odometry, Simultaneous Localization and Mapping, Robot Motion.

Abstract: Several motion estimation algorithms, such as *n*-point and perspective *n*-point (P*n*P) have been introduced over the last few decades to solve relative and absolute pose estimation problems. Since the *n*-point algorithms cannot decide the real scale of robot motion, the P*n*P algorithms are often addressed to find the absolute scale of motion. This paper introduces a new P*n*P algorithm which uses only two 3D-2D correspondences by considering only planar motion. Experiment results prove that the proposed algorithm solves the absolute motion in real scale with high accuracy and less computational time compared to previous algorithms.

1 INTRODUCTION

Pose estimation using visual features is one of the interesting problems in computer vision research. The motion information of a moving robot can be determined using feature correspondences between two camera images. Using 2D-2D feature correspondences, the relative pose can be estimated with an unknown translation scale. Using 3D-2D or 3D-3D feature correspondences, the absolute pose can be estimated with a known scale. Depending on the number of correspondences, many different *n*-point algorithms which range from 1-point to 8-point are introduced. The 8-point or 7-point algorithm (Hartley et al., 2003) can estimate the fundamental matrix of the two-view geometry to solve the 6DoF (Degree of Freedom) relative pose of an un-calibrated camera. If the camera is fully calibrated, 6-point (Stewenius et al., 2008) or 5-point (Nister, 2004) algorithms can be used to estimate the essential matrix. Most feature based techniques use iterative methods such as RANdom SAmple Consensus (RANSAC) (Fischler et al., 1981) to obtain the best solution from a set of correspondences which contain both inliers and outliers. The performance and accuracy of an *n*-point algorithm largely depends on the number of feature correspondences, feature quality, and number of iterations. Due to this reason, some algorithms such as 8-point or 5-point require long computational time to get an optimal solution. In

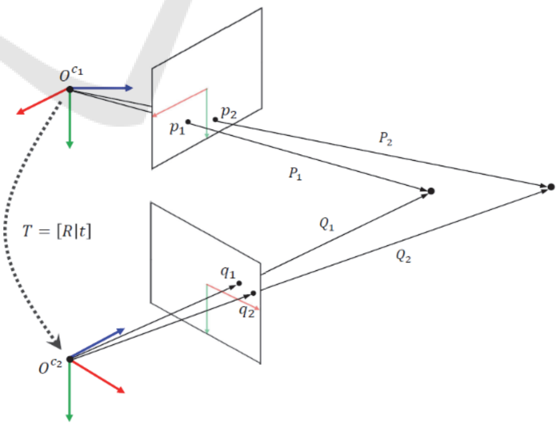


Figure 1: A geometrical model of the proposed method.

visual odometry, fast pose estimation is an important issue because the performance of localization and map building largely depends on estimation speed. In this regard, some pose estimation algorithms use less number of feature correspondences to reduce the number of iterations. Currently the 5-point algorithm is known as the minimal solution to solve the 6DoF problem using calibrated cameras.

To reduce the number of correspondences, some investigations employ extra motion sensors (E.g. IMU, Odometer and GPS/INS) to obtain the angle information of rotational motion. For example, 4-point and 3-point algorithms introduced in (Li et al.,

Choi S. and Park S..
A New 2-Point Absolute Pose Estimation Algorithm under Plannar Motion.
DOI: 10.5220/0005360406430647
In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications* (VISAPP-2015), pages 643-647
ISBN: 978-989-758-091-8

2013) and (Fraundorfer et al., 2010) use extra sensors to obtain one or more rotational motion information. The 1-point algorithm in (Scaramuzza, 2011) and 2-point algorithm in (Ortin et al., 2001) solve the relative pose with an assumption of only 3DoF planar motion. Due to the low computational complexity, many 1-point and 2-point algorithms are especially applied for the visual odometry of mobile robots which are equipped with low computing resources.

The relative pose estimation algorithm cannot obtain the scale of motion because of the inherent Epipolar geometry of two-view motion. This fact is considered as the main drawback of the 2D-2D feature based motion estimation approaches. To overcome this issue, 3D-2D correspondences are often employed in some *n*-point algorithms, which are known as the *perspective from n points* (P*n*P) algorithms. The P*n*P algorithm mostly needs less number of features than the relative *n*-point algorithm. In addition, the metric scale of robot motion can be obtained without using extra motion sensors. In conventional P*n*P algorithms, at least three 3D-2D correspondences are needed for the 6DoF pose estimation. However, the number of correspondences can be reduced in visual odometry with a planar motion constraint.

In this paper, we assume a mobile robot is restricted in 3DoF planar motion. With this assumption, we introduce a new 2-point P*n*P algorithm to reduce computation complexity and fast visual odometry. Using only two 3D-2D correspondences obtained from a RGB-D camera, we derive a linear equation to solve the 3DoF motion problem. Figure 1 shows a basic geometrical constraint of the proposed approach.

2 A NEW 2-POINT ALGORITHM

The main goal of this proposed approach is to find the rigid-body transformation matrix T by employing perspective projection model. Let us assume that a 3D point $Q = [X, Y, Z]^{\top}$in a world coordinate system and its corresponding 2D point $p = [x, y]^{\top}$ in the camera's image plane are given. Then the matrix T can be found by minimizing (1):

$$T \leftarrow \underset{T}{\operatorname{argmin}} \sum_{i=1}^{n} \|p_i - KTQ_i\|, \tag{1}$$

where n represents the number of 3D-2D feature correspondences.

In 3D Euclidean space, the rigid-body transformation matrix T has 6 degrees of freedom; 3 for rotation and 3 for translation. If the motion of the camera is restricted in a planar space, then the degrees of freedom of T can be reduced to 3; 1 for rotation and 2 for translation. When the camera is moving on the *X-Z* plane, the perspective projection matrix M can be defined as:

$$M = \begin{bmatrix} f_x \cos\theta - c_x \sin\theta & 0 & f_x \sin\theta + c_x \cos\theta & f_x t_x + c_x t_z \\ -c_y \sin\theta & f_y & c_y \cos\theta & c_y t_z \\ -\sin\theta & 0 & \cos\theta & t_z \end{bmatrix}, \tag{2}$$

where $[f_x, \ f_y]^{\top}$ represents the focal lengths of the camera in x and y directions, and $[c_x, \ c_y]^{\top}$ represents the principal point (position of the optical center of the image). Based on the definition of the perspective projection matrix of planar motion; the relation between Q and p can be represented with projection matrix M as follows:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} \cong \begin{bmatrix} f_x \cos\theta - c_x \sin\theta & 0 & f_x \sin\theta + c_x \cos\theta & f_x t_x + c_x t_z \\ -c_y \sin\theta & f_y & c_y \cos\theta & c_y t_z \\ -\sin\theta & 0 & \cos\theta & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{3}$$

In (3), $[u, v, w]^{\top}$ is represented in homogeneous coordinate system and it satisfies $p = [x, y, 1]^{\top} \cong [u/w, v/w, 1]^{\top}$. To calculate the rigid-body transformation between the two camera coordinate systems using least square minimization, (3) is converted to a linear system $AW = B$ and the result is as follows:

$$AW = B \ \rightarrow$$
$$\begin{bmatrix} f_x Z_i + (x_i - c_x) X_i & f_x X_i + (c_x - x_i) Z_i & f_x & c_x - x_i \\ (y_i - c_y) X_i & (c_y - y_i) Z_i & 0 & c_y - y_i \end{bmatrix} \begin{bmatrix} \sin\theta \\ \cos\theta \\ t_x \\ t_z \end{bmatrix} = \begin{bmatrix} 0 \\ -f_y Y_i \end{bmatrix}, \tag{4}$$
$$(i = 1, \ldots, n)$$

where n represents the number of 3D-2D correspondences, A and B represent $2n \times 4$ and $2n \times 1$ matrices respectively. Since W has 4 unknowns, at least two 3D-2D corrspondences ($n \geq 2$) are required to calculate W.

The rotation of the planar motion is represented by two variables; $\sin\theta$, $\cos\theta$ in (4). The calculated values for these two variables using (4) generally do not satisfy the Pythagorean Theorem: $\sin^2\theta + \cos^2\theta = 1$. Consequently the rotation matrix R does not satisfy the orthonormality. Moreover, $\sin\theta$ and $\cos\theta$ are different representations of the single rotational angle θ. Therefore it is preferred to reduce the unknowns of W by either removing $\sin\theta$ or $\cos\theta$, or combining $\sin\theta$ and $\cos\theta$ using the linear equation.

Let us convert $AW = B$ in (4) to $A'W' = B'$ by using the following trigonometric functions.

$$\begin{cases} \sigma \sin\theta + \beta \cos\theta = \sqrt{\sigma^2 + \beta^2} \sin(\theta + \varphi) \\ \sigma \cos\theta + \beta \sin\theta = \sqrt{\sigma^2 + \beta^2} \cos(\theta - \varphi) \end{cases}, \tag{5}$$

where φ is:

$$\varphi = \tan^{-1}\frac{\beta}{\alpha} + \begin{cases} 0, & if\ \alpha \geq 0 \\ \pi, & if\ \alpha < 0 \end{cases} \tag{6}$$

Throughout this paper, we only consider the $\sin(\theta + \varphi)$ function because $\cos(\theta + \varphi)$ function generates an ambiguity within $[-\pi, \pi]$ angle range due to its symmetry. Let $a_{ij} \in A$, $\beta_i \in B_{(i=1,\ldots,2n, j=1,\ldots,2n)}$ and $a_{ij} \neq 0$, $b_i \neq 0$ are satisfied, then (4) can be rewritten as follows when $n = 2$.

$$U\widetilde{W} = V \rightarrow \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \\ a_{41} & a_{42} \end{bmatrix} \begin{bmatrix} \sin\theta \\ \cos\theta \end{bmatrix} = \begin{bmatrix} b_1 - a_{13}t_x - a_{14}t_z \\ b_2 - a_{24}t_z \\ b_3 - a_{33}t_x - a_{34}t_z \\ b_4 - a_{44}t_z \end{bmatrix} \tag{7}$$

In order to apply the trigonometric function in (5), a rank constraint of U matrix has to satisfy the following:

$$rank(U) = 1 \tag{8}$$

For this purpose, (7) is rearranged with a proportional relation such that $w_a(c_1 f_2 - c_2 f_1) = w_b(c_2 g_1 - c_1 g_2)$. Rearranged function is as follows:

$$U'\widetilde{W} = V' \rightarrow$$
$$\begin{bmatrix} a_{11} & a_{12} \\ \gamma_{13}a_{21} - \gamma_{12}a_{31} & \gamma_{13}a_{22} - \gamma_{12}a_{32} \\ \gamma_{14}a_{21} - \gamma_{12}a_{41} & \gamma_{14}a_{22} - \gamma_{12}a_{42} \\ \gamma_{14}a_{31} - \gamma_{13}a_{41} & \gamma_{14}a_{32} - \gamma_{13}a_{42} \end{bmatrix} \begin{bmatrix} \sin\theta \\ \cos\theta \end{bmatrix} = \begin{bmatrix} v_1 \\ \gamma_{13}v_2 - \gamma_{12}v_3 \\ \gamma_{14}v_2 - \gamma_{12}v_4 \\ \gamma_{14}v_3 - \gamma_{13}v_4 \end{bmatrix} \tag{9}$$

where $\gamma_{ij} = a_{i2}a_{j1} - a_{i1}a_{j2}$ and v_i represents the i^{th} row vector of the matrix V.

The linear equation in (9) now satisfies the $rank(U') = 1$. When u'_i is set to be the i^{th} row vector of U', a matrix S which satisfies $\delta_i u'_i = u'_1$ $_{(i\,=\,1,\,\ldots\,,\,4)}$ can be defined as (10):

$$\begin{aligned} S &= [\delta_1, \delta_2, \delta_3, \delta_4] \\ &= \left[\frac{a_{11}}{a_{11}}, \frac{a_{11}}{\gamma_{13}a_{21} - \gamma_{12}a_{31}}, \frac{a_{11}}{\gamma_{14}a_{21} - \gamma_{12}a_{41}}, \frac{a_{11}}{\gamma_{14}a_{31} - \gamma_{13}a_{41}}\right] \\ &= \left[\frac{a_{12}}{a_{12}}, \frac{a_{12}}{\gamma_{13}a_{22} - \gamma_{12}a_{32}}, \frac{a_{12}}{\gamma_{14}a_{22} - \gamma_{12}a_{42}}, \frac{a_{12}}{\gamma_{14}a_{32} - \gamma_{13}a_{42}}\right] \\ &= \left[1, \frac{1}{\gamma_{23}}, \frac{1}{\gamma_{24}}, \frac{1}{\gamma_{34}}\right] \end{aligned} \tag{10}$$

A trigonometric function combining $\sin\theta$, $\cos\theta$ can be finally derived when left and right sides of the linear equation $U'\widetilde{W} = V'$in (9) are multiplied with S in (10). The resultant equation is derived as:

$$\left[\sqrt{a_{11}{}^2 + a_{12}{}^2}\sin(\theta + \varphi)\right] = \begin{bmatrix} v_1 \\ (\gamma_{13}v_2 - \gamma_{12}v_3)/\gamma_{23} \\ (\gamma_{14}v_2 - \gamma_{12}v_4)/\gamma_{24} \\ (\gamma_{14}v_3 - \gamma_{13}v_4)/\gamma_{34} \end{bmatrix} \tag{11}$$

Finally a linear equation $A'W' = B'$ can be derived from (11) and the result is shown in (12).

The inverse matrix of A' does not exist as A' is not a square matrix. Assuming A' always satisfies the condition $det(A'A'^{\intercal}) \neq 0$, W' can be easily calculated by the *pseudo-inverse matrix* $A'^{+} = \left(A'^{\intercal}A'\right)^{-1}A'$ of A'.

When $a_{ij} \in A'_{(i\,=\,1,\,\ldots\,,3,\ \ j\,=\,1,\,\ldots\,,3)}$ and $w'_i \in W'$ $_{(i\,=\,1,\,\ldots\,,3)}$ are derived, the translation vector of motion is defined as $t = [w'_2, 0, w'_3]^{\intercal}$ and the rotation angle θ is calculated as follows:

$$\varphi = \tan^{-1}\frac{a_{12}}{a_{11}} + \begin{cases} 0, & if\ a_{11} \geq 0 \\ \pi, & if\ a_{11} < 0 \end{cases} \tag{13}$$

and

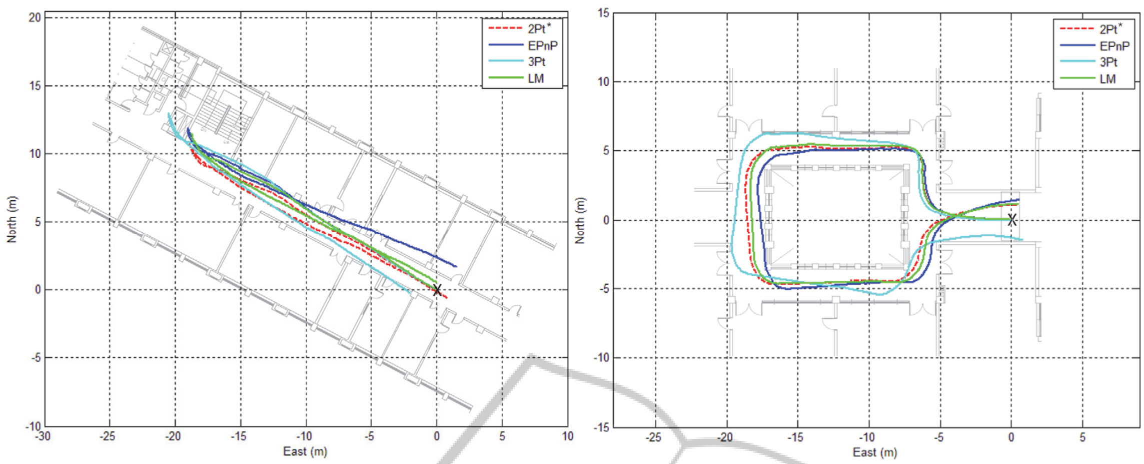
$$\theta = \sin^{-1} x'_1 - \varphi \tag{14}$$

3 EXPERIMENTS

The performance is compared with three different P*n*P methods which are considered as current state-of-arts in visual odometry. Table 1 shows the list of algorithms used for the performance comparison. Each data set consists of a sequence of RGB-D image frames. All RGB images and depth frames are captured by a RGB-D camera - *ASUS XTionPro Live* - with a 640×480 resolution. In every frame the proposed algorithm determines the motion of a mobile robot with a moving speed of 0.5m/s. The RGB-D camera is mounted on the mobile robot to capture RGB-D data in the forward direction and its position is adjusted precisely to make sure the *X-Z* plane of the camera coordinate system is parallel with the moving plane of the robot.

$$A' = \begin{bmatrix} \sqrt{a_{11}{}^2 + a_{12}{}^2} & a_{13} & a_{14} \\ (a_{22}a_{31} - a_{21}a_{32})\sqrt{a_{11}{}^2 + a_{12}{}^2} & a_{11}a_{22}a_{33} - a_{12}a_{21}a_{33} & a_{11}a_{22}a_{34} - a_{11}a_{24}a_{32} - a_{12}a_{21}a_{34} + a_{12}a_{24}a_{31} \\ (a_{22}a_{41} - a_{21}a_{42})\sqrt{a_{11}{}^2 + a_{12}{}^2} & 0 & a_{11}a_{22}a_{44} - a_{11}a_{24}a_{42} - a_{12}a_{21}a_{44} + a_{12}a_{24}a_{41} \\ (a_{32}a_{41} - a_{31}a_{42})\sqrt{a_{11}{}^2 + a_{12}{}^2} & -a_{11}a_{33}a_{42} + a_{12}a_{33}a_{41} & a_{11}a_{32}a_{44} - a_{11}a_{34}a_{42} - a_{12}a_{31}a_{44} + a_{12}a_{34}a_{41} \end{bmatrix},$$

$$B' = \begin{bmatrix} b_1 \\ a_{11}a_{22}b_3 - a_{11}a_{32}b_2 - a_{12}a_{21}b_3 + a_{12}a_{31}b_2 \\ a_{11}a_{22}b_4 - a_{11}a_{42}b_2 - a_{12}a_{21}b_4 + a_{12}a_{41}b_2 \\ a_{11}a_{32}b_4 - a_{11}a_{42}b_3 - a_{12}a_{31}b_4 + a_{12}a_{41}b_3 \end{bmatrix}, \quad X' = \begin{bmatrix} \sin(\theta + \psi) \\ t_x \\ t_z \end{bmatrix}. \tag{12}$$



(a) (b)

Figure 2: The estimated trajectories of all the methods for two data sets. (a): Corridor, (b): Hall. Coloured lines on the graphs represent different methods while the red dashed line represents the proposed method.

Table 1: Methods for performance analysis.

	Methods		
3 DoF	2pt*		
6 DoF	EPnP (Lepetit et al., 2009)	3pt (Gao et al., 2003)	LM (Levenberg, 1944)

Table 2: Information of two experimental sequences.

Sequences	Travel Distance (m)	# of frames
Corridor	49.5	3,042
Hall	53.9	3,471

All experiments are done in indoor. For more precise analysis, the closed-loop translation and rotation errors are measured in all two test sequences. Table 2 shows details of our test sets.

3.1 Implementation

We have implemented the proposed algorithm in C++ using the OpenCV library. The SIFT algorithm (Lowe, 2004) is first applied to obtain the 2D-2D correspondences between two consecutive RGB images. The matching pairs between two keypoint sets are obtained by comparing the keypoint descriptors with FLANN library (Muja et al., 2009). For robust implementation, all the P*n*P algorithms are supported by RANSAC scheme to remove the outliers of the matching pairs.

3.2 Results

When the mobile robot has returned back to the starting point after traveling, the differences of both the heading direction and the position between first and last frames are measured.

Table 3 shows the closed-loop rotation and translation errors of each estimation algorithm for two indoor sequences. The top ranks in each sequence are printed in boldface with green shades. Note that the proposed method yields the lowest errors in the translation of two test sequences. The motion trajectory results of the mobile robot in corridor and hall environments are shown in Figure 2(a) and 2(b) respectively. The trajectory of the proposed method is expressed in red dashed lines and the starting position with a black cross mark.

Pose estimation time is also compared as shown in Figure 3. The proposed approach is very fast compared to other motion estimation algorithms. The average processing time per frame is less than 40ms (including RANSAC process), which assures that the proposed algorithm runs at least 10 times faster than LM based algorithms. The processing time mainly depends on the total number of features, which could result in a time delay in RANSAC process. With the advantage of fast processing time and absolute motion estimation, the proposed algorithm can be applied very usefully in localization and map building of mobile robots.

4 CONCLUSIONS

In this paper, we proposed a new P2P algorithm to estimate the absolute pose between two different pose of a RGB-D camera. The proposed algorithm requires less computational time compared to other conventional pose estimation algorithms because it needs only two 3D-2D correspondences. Compared to other 3-DoF pose estimation algorithms, such as 1-point and 2-point algorithms, the proposed algorithm has the advantage of directly obtaining the real metric scale for the translation motion. Through several localization experiments, we showed that the proposed algorithm can achieve very accurate results in fast computational time.

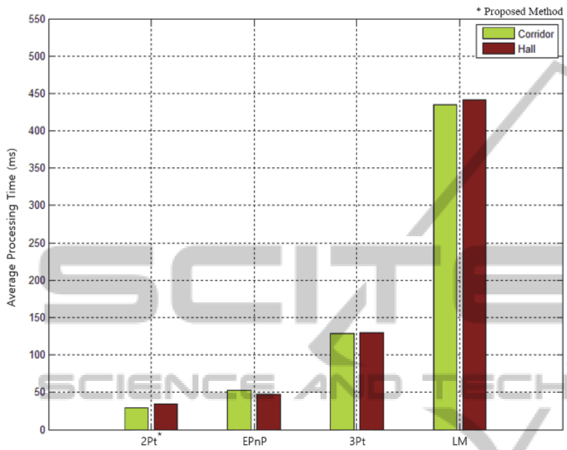


Figure 3: Computational time comparison.

Table 3: Position error of closed-loop test.

Methods	Corridor Translation (m)	Corridor Rotation (degree)	Hall Translation (m)	Hall Rotation (degree)
2Pt.*	**0.87**	-3.74	**1.14**	4.17
EPnP	2.34	-4.75	1.55	4.92
3Pt.	2.01	-8.33	1.27	**3.04**
LM	1.06	**-2.82**	1.24	-4.2

ACKNOWLEDGEMENTS

This work was supported in part from UTRC (Unmanned Technology Research Center) at KAIST, originally funded by DAPA, ADD and in part by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the C-ITRC support program (NIPA-2014-H0401-13-1004) supervised by the NIPA (National IT Industry Promotion Agency)

REFERENCES

Hartley, R. and Zisserman, A., 2003. *Multiple view geometry in computer vision*, Cambridge University Press, 2nd edition.

Stewenius, H., Nister, D., Kahl, F. and Schaffalitzky, F., 2008. A minimal solution for relative pose with unknown focal length. *Image and Vision Computing*, 26(7):871–877.

Nister, D., 2004. An efficient solution to the five-point relative pose problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(6):756–770.

Fischler, M. A. and 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.

Li, B., Heng, L., Lee, G. H. and Pollefeys, M., 2013. A 4-point algorithm for relative pose estimation of a calibrated camera with a known relative rotation angle. *In Proc. Of the IEEE Int'l Conf. on Intelligent Robots and Systems*, 1595–1601.

Fraundorfer, F., Tanskanen, P. and Pollefeys, M., 2010. A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. *In: ECCV*, 269–282.

Scaramuzza, D., 2011. 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International Journal of Computer Vision*, 95(1):74–85.

Ortin, D. and Montiel, J., 2001. Indoor robot motion based on monocular images. *Robotica*, 19(03):331–342.

Lepetit, V., Moreno. N. F. and Fua, P., 2009. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155–166.

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

Levenberg, K., 1944. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics*, 2(2):164–168.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Muja, M. and Lowe, D. G., 2009. Fast approximate nearest neighbors with automatic algorithm configuration. In: VISAPP, 331–340.