Fast and Reliable Minimal Relative Pose Estimation under Planar Motion

Sunglok Choi\(^{a,b}\), Jong-Hwan Kim\(^{c,*}\)

\(^{a}\)Intelligent Robotics Research Division, ETRI, Daejeon, South Korea
\(^{b}\)Robotics Program, KAIST, Daejeon, South Korea
\(^{c}\)School of Electrical Engineering, KAIST, Daejeon, South Korea

Abstract

This paper proposes fast, reliable, and minimal non-iterative relative pose solvers under planar motion constraint. Relative pose estimation is popularly utilized in many important problems such as visual odometry and SLAM, and planar motion is common for mobile robots and vehicles on floors and roads. We transform the original problem formulation of finding intersections of two ellipses into more accessible form of finding intersections of a line and unit circle. Such transformation leads to a non-iterative and closed-form solver, which enables significant speed-up compared to previous methods. The proposed algorithm is almost 9 times faster than the previous minimal solver with planar motion and around 90 times faster than the previous minimal solver with general motion. In addition, our algorithms provide reliable relative pose in degeneracy of the previous minimal planar solvers. Effectiveness of the proposed algorithms is demonstrated with two types of experiments: relative pose estimation with synthetic data and monocular visual odometry with real image sequences.

Keywords: relative pose estimation, planar motion, epipolar geometry, essential matrix, 2-point algorithm.

1. Introduction

Relative pose estimation between two images is one of the most fundamental problems in computer vision and robotics. It is a common and popular tool to reject incorrect visual correspondence (outliers) and also provides initial rotation and translation between two image coordinates. It has contributed to solve many important problems such as visual odometry, visual SLAM, and structure-from-motion (SfM), which has been further applied to visual navigation, 3-D reconstruction, scene synthesis, and augmented reality. Due to its importance and usefulness, relative pose problem has been investigated across computer vision and robotics. In computer vision, full degree-of-freedom (DoF) relative pose problem had been initially tackled and made an elegant mathematical basis, epipolar geometry \[^{[1]}\].

On the other hand, in robotics, more simplification and practical approaches have been made based on prior kinematic constraints. Simplified epipolar geometry \[^{[2]}\] was coined to describe those formulations and approaches in robotics. Mobile robots and ground vehicles usually operate on floors (indoor) and roads (outdoor) so that their motions are almost or locally planar. The planar motion constraint can simplify the 6-DoF relative pose problem as a 3-DoF one, which naturally enables less computation but higher accuracy. Above all, planar motion mostly leads to simpler mathematical equations than general motion. As an extreme example, there is a simple closed-form solver available under planar and circular motion \[^{[3]}\]. Planar motion estimation also requires less number of visual correspondences. The number of necessary correspondences highly affects computing time of outlier rejection. For example, RANSAC’s number of iterations is exponentially increased as more correspondences are used \[^{[4]}\]. That’s why minimal solvers are important in real applications. From the viewpoint of accuracy, less DoF formulation is usually more robust to noise because extra variables in higher DoF are additionally affected from noise. For example, 6-DoF visual odometry on a plane suffers from vertical drift, but 3-DoF one does not. There have been many researches and applications to demonstrate better performance with the planar motion constraint, which are briefly reviewed in Section \[^{[2]}\].

This paper proposes two versions of minimal and non-iterative relative pose solvers with planar motion constraint. The proposed algorithms are derived from simplified epipolar geometry under planar motion, whose problem is originally formulated as finding intersections of two zero-centered arbitrary ellipses. The problem is transformed into more accessible forms, finding intersections of a line and unit circle, so that the solvers can find solutions significantly faster and more reliable than the previous works such as an iterative minimal solver \[^{[5]}\] and non-iterative 2-point algorithm \[^{[6,7]}\]. The proposed algorithms do not rely on iterative local optimization so that they can find globally optimal solutions with less computation. Moreover, our novel problem formulation, a line and unit circle, provides a completely closed-form solution at its minimal case, which accelerates its computing time around
9 and 4 times faster than the previous non-iterative solver [6, 7], respectively. Additionally, the proposed algorithms can reliably recover failure cases (e.g., no intersection of two ellipses) for which the previous non-iterative solvers [6, 7] fail. In our experiments, the previous non-iterative algorithms [6, 7] failed around 10% of overall trials with mild noise configuration. The mathematical details about the proposed and previous minimal solvers are described in Section 3 and two types of experiments with synthetic and real data are presented in Section 4 to demonstrate effectiveness of the proposed algorithms.

2. Related Works

Relative pose estimation between two cameras is one of classical geometric problems in computer vision. Generally, 6-DoF relative pose is estimated through 8-point algorithm, which is common and standard in textbooks [11]. There has been a minimal solver, 5-point algorithm [8], which uses the least number of correspondences but needs more computation. The minimal solver was improved based on Gröbner basis [9, 10], which becomes a popular approach to generate minimal solvers in many problems. Recently, there was a tutorial, The Art of Solving Minimal Problems [11], about designing and implementing minimal solvers due to their technical usefulness.

In robotics, less-DoF relative pose problem has been inspired because mobile robots and vehicles usually move on floors or roads. There have been many meaningful researches and applications based on simplified relative pose problems.

Planar Circular Motion: Relative pose extremely becomes 1-DoF (except scale) when a camera undergoes planar and circular motion. The motion assumption, planar and circular, is usually valid when a camera is installed at the rear axle of vehicles. Sometimes arbitrary planar motion can be approximated to locally circular motion during a very short period of time. Scaramuzza et al. [3] proposed significantly fast 1-point RANSAC based on a minimal solver using only a single correspondence. Choi et al. [12] improved 1-point algorithm by normalization. The normalization enabled higher accuracy in least-square situations (more than one point on the same plane) and led to a computationally simple geometric error function. Civera et al. [13] utilized 1-point RANSAC incorporated with extended Kalman filtering for real-time visual odometry and structure-from-motion. Interestingly, the motion assumption can resolve scale ambiguity [14] when a camera is not on the rear axle. When a vehicle undergoes planar and circular motion and its camera does not (due to its miss installation), such disparity provides a clue for motion scale, which is fundamentally unknown in a single camera configuration.

Planar Motion: A camera generally follows planar motion when it moves on floors and roads. Such 2-DoF (except scale) motion problem was initially tackled by Ortin and Montiel [5]. They suggested two solvers: one is iterative 2-point algorithm based on Newton method, and the other is 3-point algorithm formulated by a linear equation. The iterative minimal solver was popularly applied to monocular visual odometry [3, 12] incorporated with 1-point algorithm. Recently, Chou and Wang [6] proposed non-iterative 2-point algorithm by formulating the problem as finding intersections of two ellipses, and Hong et al. [7] also suggested similar approach with less computation. Such idea was also briefly introduced in Choi et al. [15]. Their mathematical formulations are described in Section 3 in detail.

Since planar motion is quite common for many applications, there have been further related researches with additional sensors and practical configurations. Troiani et al. [16] proposed 2-point RANSAC for their camera-IMU system installed on micro aerial vehicles (MAVs), and Lee et al. [17] devised a novel 2-point algorithm for generalized (multiple) cameras for commercial cars. Choi and Park [18] exploited planar motion estimation with RGB-D sensors.

General Small Motion: The simplified motion assumption was effectively applied to full-DoF motion if some of motion elements are small. Since most mobile robots, vehicles, and MAVs move smoothly, its inter-frame camera motion is mostly quite small as shown in Table 1 and Fig. 4 in [2]. Therefore, full-DoF algorithms can be accelerated and simplified by linearizing such small elements, not ignoring them. Choi et al. [2] proposed approximated 5-point algorithm based on iterative Newton method and applied it to on-road monocular visual odometry. They linearized roll and pitch of 3-DoF rotation matrix and found relative pose using the Newton method. Similarly, Im et al. [19] linearized all elements of rotation matrix for structure-from-small-motion and applied it to acquire a dense depth image during taking a photo using commercial cellular phones.

3. Relative Pose Estimation under Planar Motion

3.1. Epipolar Geometry under Planar Motion

Epipolar geometry explains the geometric relationship between two images taken from different point of views. The geometric relationship is usually represented by an essential matrix (or fundamental matrix) and point correspondence between two images. An essential matrix is defined as a 3-by-3 matrix derived from relative pose between two images. When a 3-D point is projected on two images (normalized image planes) as \( x \) and \( x' \), respectively, an essential matrix satisfies

\[
x^T E x = 0 \quad (E = [t]_x R)
\]

where \( R \) and \( t \) are rotation and translation of the 1st camera coordinate from the 2nd camera coordinate, and \([t]_x\) is a skew-symmetric matrix to convert cross product to
matrix multiplication. Motion from the 1st camera to 2nd camera is noted as $R'$ and $t'$ which hold $R' = R^T$ and $t' = -R^T t$.

An essential matrix with planar motion is simply derived from its definition $[1]$. As shown in Fig. 1, planar motion is represented as

$$E = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \phi & 0 & -\sin \phi \\ 0 & 1 & 0 \\ \sin \phi & 0 & \cos \phi \end{bmatrix} = \begin{bmatrix} x' \cos \phi - y' \sin \phi \\ x' \sin \phi + y' \cos \phi \end{bmatrix}.$$  \tag{3}$$

and its epipolar constraint $[1]$ becomes

$$xy' \cos \phi - zy' \sin \phi - x'y \cos(\theta - \phi) - z'y \sin(\theta - \phi) = 0. \tag{4}$$

3.2. Iterative 2-point Algorithm

As a basic approach, nonlinear programming (NLP) can be applied to find relative pose with the minimal situation ($N \geq 2$). Ortin and Montiel $[5]$ utilized the Newton method to find two unknowns, $x = [\phi, \theta]^T$, constrained with epipolar constraint $[4]$. The Newton method gradually reaches a locally optimal solution by a series of iterations of

$$x_{k+1} = x_k - J^T(x_k) f(x_k), \tag{5}$$

where $f$ is a cost function derived from epipolar constraint $[1]$ and $J^T$ is pseudo-inverse of its Jacobian matrix. Even though this approach can converge to one of solutions with the minimal number of points, it has fundamental drawbacks. It only pursues one of solutions near its initial value and cannot guarantee global optimality. Moreover, it spends a lot of computing time due to a series of iterations for convergence.

3.3. Linear 3-point Algorithm

Direct linear transform (DLT) $[11]$ can find relative pose from the given point correspondence. DLT is a process to solve a set of unknown variables constrained with a set of similarity relations. The relative pose problem is formulated as a set of homogeneous equations derived from epipolar constraint $[4]$ as follows:

$$Ab = 0 \text{ such that } b = \begin{bmatrix} \cos \phi \\ \sin \phi \\ \cos(\theta - \phi) \\ \sin(\theta - \phi) \end{bmatrix}, \tag{6}$$

where the $i$-th row of $A$ is

$$A_{(i)} = \begin{bmatrix} x_i y_i' - z_i y_i \\ -x_i y_i' - z_i y_i' \\ x_i' y_i - z_i y_i' \end{bmatrix}. \tag{7}$$

We can find $b$, a null vector of $A$, through singular value decomposition (SVD) with more than two points ($N \geq 3$). However, it is not a minimal solver because it does not utilize additional constraints in $b$,

$$b^2(1) + b^2(2) = 1 \text{ and } b^2(3) + b^2(4) = 1. \tag{8}$$

Intersections of Two Ellipses (Ellipse Version):

Non-iterative and minimal solvers had been proposed while utilizing such additional constraints. Chou and Wang $[6]$ suggested to model the constraints as two ellipses. From two correspondences, the rank of matrix $A$ is $2$ so that it has two null vectors, $v_1$ and $v_2$, which can be calculated by SVD. The solution $b$ is a linear combination of the two null vectors as follows:

$$b = \lambda_1 v_1 + \lambda_2 v_2 = \mathbf{V} \lambda, \tag{9}$$

where the matrix $V$ is a horizontal concatenation of the two null vectors, $[v_1, v_2]$, and the vector $\lambda$ is a vertical concatenation of the two variables, $[\lambda_1, \lambda_2]^T$. Therefore, the additional constraints become a pair of zero-centered ellipses in the $\lambda$ domain as follows:

$$\mathbf{V} \lambda = 1 \text{ and } \mathbf{V} \lambda^T \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{V} \lambda = 1. \tag{10}$$

Chou and Wang $[6]$ found intersections $\lambda$ using an opensource geometry library, Geometric Tools $[20]$. On the other hand, Hong et al. $[7]$ found the ratio of two variables, $\lambda_2/\lambda_1$, using the following transformed constraint:

$$\begin{bmatrix} b(1) + b(3) \\ b(2) + b(4) \\ b(1) - b(3) \\ b(2) - b(4) \end{bmatrix} = -1, \tag{11}$$

which leads a single quadratic equation of $\lambda_2/\lambda_1$ so that it is solved in a closed-form. The relative pose $b$ is finally derived from Eq. (6). However, both methods suffer from degenerate cases such as no intersection of two ellipses and no solution of the quadratic equation. Both also need to calculate two null vectors of the $N$-by-4 matrix $A$ using SVD which inevitably works in an iterative manner.
3.4. Proposed 2-point Algorithms

Now we propose two versions of more reliable and faster non-iterative algorithms working with minimal point correspondence ($N \geq 2$). Compared to the iterative 2-point algorithm, they provide globally optimal solutions without iterative local optimization. In contrast to the linear 3-point algorithm, they work with two correspondences by exploiting the missing constraints $\|$. Compared to the previous non-iterative 2-point algorithms \cite{6, 7}, they always provide results without degeneracy. Especially, our algorithms work completely in a closed-form (without SVD to find null vectors) at the minimal case ($N = 2$) so that its computing time is significantly less than the others. Table \ref{table} briefly summarizes their computing time and major operations.

The epipolar constraint \cite{4} can be represented as another linear equation,

$$\mathbf{Aa} = \mathbf{Bb} \text{ such that } \mathbf{a} = \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \cos(\theta - \phi) \\ \sin(\theta - \phi) \end{bmatrix},$$ \quad (12)

where $i$-th rows of $\mathbf{A}$ and $\mathbf{B}$ are respectively

$$\mathbf{A}_i = [x_i y_i' - z_i y_i'] \quad \text{and} \quad \mathbf{B}_i = [x_i' y_i z_i' y_i] .$$ \quad (13)

The proposed two different transformations and their solvers are presented in the following.

Intersections of Unit Circle and Ellipse (Circle Version):

The linear equation \cite{12} is arranged as

$$\mathbf{b} = \mathbf{B}^\dagger \mathbf{Aa} = \mathbf{Ca} ,$$ \quad (14)

so that we can derive the unit circle and ellipse, which are mathematically written by a vector $\mathbf{a}$ as follows:

$$\mathbf{a}^\top \mathbf{a} = 1 \quad \text{and} \quad \mathbf{a}^\top \mathbf{C}^\top \mathbf{Ca} = 1 .$$ \quad (15)

Therefore, to find the vector $\mathbf{a}$, we need to find intersections between the unit circle and zero-centered ellipse.

SVD is possible to make an oblique ellipse axis-aligned so that the intersections can be simply found. SVD decomposes a symmetric matrix $\mathbf{C}^\top \mathbf{C}$ into $\mathbf{US}^\top$ where $\mathbf{S}$ is a diagonal matrix. Instead of the vector $\mathbf{a}$, two ellipses are represented by $\mathbf{y} = \mathbf{U}^\top \mathbf{a}$ as follows:

$$\mathbf{y}^\top \mathbf{y} = 1 \quad \text{and} \quad \mathbf{y}^\top \mathbf{Sy} = 1 ,$$ \quad (16)

which is depicted in Fig. \ref{fig:2}. The relative pose problem becomes finding intersections between the unit circle and axis-aligned ellipse, whose solutions are simply derived as

$$\mathbf{y}(1) = \pm \sqrt{\frac{1 - s_2}{s_1 - s_2}} \quad \text{and} \quad \mathbf{y}(2) = \pm \sqrt{\frac{s_1 - 1}{s_1 - s_2}} ,$$ \quad (17)

where $s_1$ and $s_2$ are the first and second diagonal elements of $\mathbf{S}$ ($s_1 \geq s_2 \geq 0$), respectively. Finally, $\mathbf{a}$ and $\mathbf{b}$ come from

$$\mathbf{a} = \mathbf{Uy} \quad \text{and} \quad \mathbf{b} = \mathbf{Ca} .$$ \quad (18)

Four solutions are derived from a combination of $\mathbf{y}(1)$ and $\mathbf{y}(2)$.

The transformed formulation is also quite useful to recover two degenerate cases. The two failed cases, $s_1 < 1$ or $s_2 > 1$, are depicted in Fig. \ref{fig:3} From our geometric intuition, if there is no intersection, we can select the nearest points of the unit circle and ellipse as follows:

$$\mathbf{y} = \begin{cases} \begin{bmatrix} \pm 1 \\ 0 \end{bmatrix}^\top & \text{if } s_1 < 1 \\ \begin{bmatrix} 0 \\ \pm 1 \end{bmatrix}^\top & \text{if } s_2 > 1 \end{cases} .$$ \quad (19)

Two solutions are derived from positive and negative poles of the ellipse, respectively. The solutions break the second condition of Eqs. \cite{15} and \cite{16}, but they mathematically mean

$$\mathbf{y} = \arg \min_{\mathbf{y} = 1} [\mathbf{y}^\top \mathbf{Sy} - 1]^2 .$$ \quad (20)

From Eqs. \cite{14} and \cite{18}, the optimization is transformed to

$$\mathbf{a} = \arg \min_{\mathbf{a}^\top \mathbf{a} = 1} \left| \mathbf{b}^\top \mathbf{B}^\dagger (\mathbf{Aa} - \mathbf{Bb}) \right|^2 \quad \text{such that} \quad \mathbf{b}^\top = \mathbf{Ca} ,$$ \quad (21)

which means that our solutions in degeneracy is optimal in the sense of epipolar constraint \cite{4} with multiplication of $\mathbf{b}^\top \mathbf{B}^\dagger$. We observed that degeneracy occurred when planarity of motion was seriously broken. In our experiments,
larger correspondence noise generated more frequent failure (as shown in Fig. 6c), but small amount of non-planar motion noise made high and steady failure rate (as shown in Fig. 7c). We also observed that the solutions in degeneracy were quite effective rather than accepting failures. As shown in Fig. 8, the degenerated solutions had reasonable rotation and translation error compared to normal solutions.

Intersections of Line and Unit Circle (Line Version):

Interestingly, trigonometric identities enable to transform an ellipse to a line so that solutions can be achieved in a simpler form without SVD. The ellipse, $a^T C^T C a = 1$, is represented as

$$\alpha \cos^2 \phi + 2\beta \cos \phi \sin \phi + \gamma \sin^2 \phi = 1. \quad (22)$$

where each coefficient comes from $C^T C = \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix}$. This ellipse is transformed to a line,

$$(\alpha - \gamma) \cos 2\phi + 2\beta \sin 2\phi + (\alpha + \gamma - 2) = 0, \quad (23)$$

due to three trigonometric identities as follows:

$$\cos^2 \phi = \frac{1 + \cos 2\phi}{2}, \quad \sin^2 \phi = \frac{1 - \cos 2\phi}{2}, \quad \cos \phi \sin \phi = \frac{\sin 2\phi}{2}. \quad (24)$$

Finally, the problem is re-formulated as finding intersections between a line (23) and unit circle, $\cos^2 2\phi + \sin^2 2\phi = 1$, which is depicted in Fig. 4. Due to simplicity of a line and unit circle, their closed-form intersections are also available as follows:

$$\begin{bmatrix} \cos 2\phi \\ \sin 2\phi \end{bmatrix} = \frac{1}{\sqrt{\alpha'^2 + \beta'^2}} \begin{bmatrix} -\alpha' \gamma' + \beta \Delta \\ -\beta \gamma' + \alpha' \Delta \end{bmatrix}, \quad (25)$$

where $\alpha' = \alpha - \gamma, \gamma' = \alpha + \gamma - 2, \text{ and } \Delta = \sqrt{\alpha'^2 + \beta'^2 - \gamma'^2}$. Therefore, translation $a$ and rotation $b$ are calculated as

$$a = \frac{\pm 1}{\sqrt{2}} \begin{bmatrix} \sqrt{1 + \cos 2\phi} \\ \beta \sin 2\phi \end{bmatrix} \text{ and } b = Ca, \quad (26)$$

where $s$ is whether $\cos \phi$ and $\sin \phi$ have the same sign (+1) or not (−1). The value $s$ is simply calculated from $s = \text{sgn}(\sin 2\phi)$ due to the trigonometric relation \[24\], where $\text{sgn}(x)$ is a function to return the sign of vector $x$ among $\{+1, 0, -1\}$. Finally, four solutions are derived from a pair of $[\cos 2\phi, \sin 2\phi]^T$.

In case of degeneracy, the nearest points can be found as follows:

$$\begin{bmatrix} \cos 2\phi \\ \sin 2\phi \end{bmatrix} = \frac{\text{sgn}(\gamma')}{\sqrt{\alpha'^2 + \beta'^2}} \begin{bmatrix} -\alpha' \\ -\beta \end{bmatrix}. \quad (27)$$

Similarly, two solutions are derived from Eq. \[26\].

The proposed transformations and derived relative pose solvers are meaningful in two aspects. First, they are computationally inexpensive. They do not require two null vectors (calculated from SVD) and does not find intersections of two arbitrary ellipse. Moreover, as shown in Table \[1\] only necessity, $2$-by-$2$ matrix inversion, can simply be done in a closed-form at the minimal case ($N = 2$). Second, they are reliable against degeneracy caused by non-planar motion noise. They can easily find alternative solutions, the nearest points of two constraints, which minimize epipolar constraint \[1\] with multiplication of $b^T B^\dagger$. We also experimentally showed that the alternative solutions are reasonable rather than accepting failure.

4. Experiments

Two kinds of experiments were performed to verify the effectiveness of the proposed 2-point algorithms. The first experiment was relative pose estimation using synthetic data with ground truth. Since we generated image observation artificially, we could control the amount of data, noise, and camera motion so that we could analyze their performance with respect to each control variable. The second experiment was about one of major applications, monocular visual odometry, with real image sequences. From the second experiment, we could verify that the 2-point algorithms were meaningful not only in synthetic data, but also in real data and application when the planar motion assumption is often violated.

4.1. Relative Pose Estimation

4.1.1. Configuration

Relative pose estimation was performed to evaluate accuracy and computing time of the proposed 2-point algorithms. The two proposed 2-point algorithms (circle and line version) were compared with six previous algorithms. Four previous planar algorithms (iterative 2-point, linear 3-point, elliptic 2-point algorithms, and null-vector 2-point) are briefly reviewed in Section \[3\] respectively, and the rest two general algorithms (5-point and 8-point algorithms) are from recent and standard approaches \[9 \& 11\].

Synthetically generated images were utilized to evaluate seven algorithms for control variables such as the number of points and the magnitude of noise. Two virtual
camera observed randomly generated 10,000 3-D points around them, whose focal length was 500 and resolution was 640 x 480. We modeled that the point observation had additive non-biased Gaussian noise, \(N(0, \sigma^2)\), whose standard deviation \(\sigma\) was defined as the magnitude of image noise. Relative pose between two cameras was written in \([\rho^*, \phi^*, \theta^*]^T\), which was the ground truth for relative pose estimation. In the experiments, the true relative pose \((\phi^*, \theta^*)\) was \((8, 16)\) and \((16, 32)\) degrees with constant baseline, \(\rho^* = 1\) m. We also considered non-planar motion noise which was modeled as uniform distribution, \(U(-\epsilon, \epsilon)\), whose scale \(\epsilon\) was defined as the magnitude of non-planar motion noise. The randomly generated non-planar motion noise was applied to non-planar parts of 6-DoF relative pose (XZ-plane translation, X-axis rotation, and Z-axis rotation) as follows:

\[
\phi_{xz} = \phi^*, \quad \phi_{yz} = U(-\epsilon, \epsilon), \quad \theta_x = U(-\epsilon, \epsilon), \quad \theta_y = \theta^*, \quad \text{and} \quad \theta_z = U(-\epsilon, \epsilon).
\]

(28)

We quantified accuracy in the view of rotational error, translational error, and failure rate. The rotational and translational errors are defined as

\[
e_r(\theta; \theta^*) = |\theta - \theta^*| \quad \text{and} \quad e_t(\phi; \phi^*) = |\phi - \phi^*|.
\]

(29)

The failure rate is the ratio of failures to the overall trials. All algorithms were implemented in C++ with OpenCV v3.2 whose computing time was measured by OpenCV API on Intel Core i7-4700MQ 2.4GHz (using single core). For statistically meaningful results, we executed each algorithm on 10,000 different sets of point correspondences. Among 10,000 values, we selected their median as their representative value in order to deal with failure of algorithms.

4.1.2. Results and Discussion

The experimental results were recorded while varying three control variables, the number of point correspondence \(N\), the magnitude of image noise \(\sigma\), and the magnitude of non-planar motion noise \(\epsilon\). They are graphically presented in Figs. [4] [6] and [7]. Especially, Table 1 highlights computing time and major operations.

In the view of accuracy, four non-iterative 2-point algorithms (ellipse, null, circle, and line versions) had almost the same rotational and translation accuracy. However, ellipse and null versions had around 10% of failure rate as presented in Figs. [6] and [7], which was increasing as noise became more serious. The iterative 2-point algorithm had worst accuracy, especially rotation, because it is local optimization from the given initial value. The linear 3-point algorithm had the best least-square performance when the given point correspondence was more than five, but it is not a minimal solver. Two general algorithms, 5-point and 8-point algorithms, were usually worse than other planar algorithms because they do not utilize an additional prior, planar motion. Nevertheless, as we expect, they were stronger than planar algorithms against non-planar motion noise as shown in Fig. [7]. Similarly to planar algorithms, their rotational and translational errors were linearly increasing as motion noise was higher because their estimation contained the motion noise as relative pose.

More detailed analysis was performed to investigate the accuracy of the proposed algorithms in ellipse version’s degenerate situations (no intersection of two ellipses). We distinguished accuracy records with respect to success and failure of the ellipse version. Fig. [5] presents line version’s error distributions while varying the number of point correspondences, \(N\), whose results are shown in Fig. [5]. We observed that the proposed algorithms gave reasonable solutions with slightly lower accuracy in spite of the ellipse version’s failure. Two previous versions, ellipse and null, failed at the same datasets, and two proposed versions, circle and line, had the same accuracy in case of degeneracy.

Additional experiments were performed to observe effect of degeneracy in conjunction with RANSAC framework. The given point correspondence had 70% of outliers which generated by random association of point pairs. Similarly to the above experiments, inliers were also contaminated by additive Gaussian noise whose standard deviation is 0.1 pixels. Around 50 iterations of random sampling was sufficient for 70% of outliers to guarantee 0.99 success rate of selecting two inliers [4]. We performed 2-point algorithms with RANSAC while varying its iterations from 4 to 60 as shown in Fig. [9]. As a result, two previous algorithms, ellipse and null, failed almost 7% of RANSAC iterations due to their degeneracy. Such loss of iterations significantly degraded accuracy when the number of iterations was fixed and much less than minimum necessity. In our experiments, when the number of RANSAC iterations was less than 15, two previous methods had worse accuracy than the proposed methods. In addition, such

<table>
<thead>
<tr>
<th>DoF</th>
<th>Algorithms</th>
<th>(N^*)</th>
<th>Time [(\mu\text{sec})]</th>
<th>Scaled Time</th>
<th>Major Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planar (3 DoF)</td>
<td>2pt_ellipse [6]</td>
<td>2</td>
<td>7.698</td>
<td>9.004</td>
<td>(\text{Nx4SVD, 2x2inverse})</td>
</tr>
<tr>
<td></td>
<td>2pt_null [7]</td>
<td>2</td>
<td>2.994</td>
<td>3.502</td>
<td>(\text{Nx4SVD})</td>
</tr>
<tr>
<td></td>
<td>2pt_circle</td>
<td>2</td>
<td>5.987</td>
<td>7.002</td>
<td>(\text{Nx9SVD, 2x2inverse})</td>
</tr>
<tr>
<td></td>
<td>2pt_line</td>
<td>2</td>
<td>8.055</td>
<td>9.535</td>
<td>(\text{Nx9SVD, 2x2inverse})</td>
</tr>
<tr>
<td></td>
<td>2pt_iterative [5]</td>
<td>3</td>
<td>2.994</td>
<td>3.502</td>
<td>(\text{Nx9SVD, 2x2inverse x 1 iterations})</td>
</tr>
<tr>
<td></td>
<td>3pt_linear [5]</td>
<td>3</td>
<td>2.994</td>
<td>3.502</td>
<td>(\text{Nx9SVD, 2x2inverse})</td>
</tr>
<tr>
<td></td>
<td>3pt [9]</td>
<td>5</td>
<td>60.827</td>
<td>74.535</td>
<td>(\text{Nx9SVD, 10x9inverse, 10x9eigen_decomp})</td>
</tr>
<tr>
<td></td>
<td>8pt [1]</td>
<td>8</td>
<td>20.100</td>
<td>23.509</td>
<td>(\text{Nx9SVD})</td>
</tr>
</tbody>
</table>
Figure 5: Seven algorithms were executed while varying the number of point correspondences, $N$. The other variables were fixed, $\sigma = 0.1$ and $\epsilon = 0$. The computing time (c) is log-scaled to visualize small-scale and large-scale in detail.

Figure 6: Seven algorithms were executed while varying the magnitude of image noise, $\sigma$. The number of point correspondence $N$ was minimal for each algorithm, and the other variable $\epsilon$ was constant as zero.

Figure 7: Seven algorithms were executed while varying the magnitude of non-planar motion noise, $\epsilon$. The number of point correspondence $N$ was minimal for each algorithm, and the other variable $\sigma$ was constant as 0.1.
loss of iterations also caused around more RANSAC iterations when RANSAC decided its termination adaptively. As shown in Table 2, adaptive RANSAC with the previous 2-point method (ellipse) had 10% more iteration. More detail configuration and results are described in Section 4.2.2.

In comparison of computing time, the proposed 2-point algorithm (line version) was the best. It was almost 9 times faster than ellipse version, 4 times faster than null, and about 90 times faster than the recent 5-point algorithm at their minimal situations. More details were presented in Table 1 and Fig. 5c with the increasing number of correspondence, N. In Fig. 5c, line version had significantly less computing time at its minimal case (N = 2). Such speed-up was possible because N-by-2 matrix inversion became a closed-form when N = 2.

4.2. Monocular Visual Odometry

4.2.1. Configuration

The nine algorithms were applied to monocular visual odometry to evaluate their performance in real data and application. Basically, our monocular visual odometry was based on an open-source library, LIBVISO2 [21], which includes monocular version so-called VISO2-M. It is composed of four steps: feature extraction, outlier rejection, pose refinement, and scale estimation. In feature extraction, it retrieves locally salient points by 5-by-5 kernel responses and matches them with features on the previous image using sum-of-absolute-difference (SAD) of 11-by-11 windows. To exclude wrong feature correspondence, outlier rejection is performed by RANSAC in conjunction with normalized 8-point algorithm and Sampson distance. Again normalized 8-point algorithm is applied to get refined relative pose only from correctly matched features. Since monocular configuration does not provide motion scale, \( \rho \), ground plane fitting and its interval scaling is used to find a proper inter-frame distance.

In our outlier rejection, the eight algorithms were incorporated with adaptive version of RANSAC [22] which can terminate its iterations based on online inlier ratio estimate. Especially, five planar algorithms was applied with two-step RANSAC similar to [13, 14, 12]. The planar algorithms rejected wrong matches with larger threshold (4) in the first RANSAC, and approximated 5-point algorithm [2] refined results with smaller threshold (0.3) in the second RANSAC. In our experiments, we also included the original single-step RANSAC with the approximated 5-point algorithm as a reference. Additionally, in the scale estimation, we also used asymmetric kernel fitting [23] to find more reliable motion scale from ground planes.

We tested monocular visual odometry on KITTI odometry dataset [24], real on-road image sequences over 11 sets (No. 0 to 10) and more than 23,000 image frames. To quantify odometry accuracy, we adopted rotational and translation error defined in KITTI Visual Odometry and SLAM Evaluation 2012 [24]. All algorithms and visual odometry were implemented in C++ and executed on the same computing configuration in Section 4.2.2.

4.2.2. Results and Discussion

Odometry accuracy was presented with respect to path length and speed as shown in Fig. 10. The eight algorithms except VISO2-M had similar accuracy, but 5-point and 8-point algorithms were slightly worse than the rest of five planar algorithms. Two general algorithms, especially 8-point algorithm, became worse under high speed more than 50 km/h (Fig. 10). Even though the KITTI odometry dataset contained non-planar motion of its vehicle due to vibration and bumps, the five planar algorithms worked effectively as the first RANSAC filtering. An additional experiment with synthetic data explains robustness of the planar algorithms against non-planar motion noise, which is described in Fig. 11. According to its result, the planar algorithms can easily distinguish inliers and outliers until 6° of non-planar motion noise. Odometry with the approximated 5-point algorithm had similar accuracy to the five planar algorithms. In contrast, VISO2-M was worst in view of rotational and translational errors because the others equipped with Sampson distance on image planes.\(^1\)}

\(^1\)VISO2-M calculates Sampson distance on the normalized image coordinate, but the others calculate it on image planes. For more details, the readers is referred to Table 3 in [2].
more reliable scale estimation [23], respectively. Examples of trajectories by planar and general motion models are depicted in Fig. 12.

In computing time of outlier rejection by RANSAC, the planar algorithms with two-step RANSAC reduced computing time significantly as presented in Table 2. They were almost 30 times less than the 5-point algorithm and 8 times less than the 8-point algorithm. However, their respective difference was minor because the planar algorithms were executed almost 10 times during RANSAC’s first step. Even though a few iterations, they were quite effective in rejecting wrong feature correspondence so that they could reduce necessary iterations (around 20% less) in RANSAC’s second step and overall computing time (almost 28% less) compared to approximated 5-point algorithm with single-step RANSAC.

5. Conclusion

In this paper, we proposed two kinds of non-iterative 2-point algorithms to estimate relative pose under planar motion. We transformed the problem into finding intersections of a line and the unit circle instead of the original elliptic formulation. The transformed formulation enabled non-iterative and closed-form solvers so that the proposed algorithms could be faster and more reliable than the previous minimal algorithms. In our experiments with synthetic data, our proposed algorithm of the line version was almost 9 times faster than the elliptic formulation and around 90 times faster than the recent 5-point minimal solver. Even though the previous 2-point algorithms (ellipse and null version) had about 10% of failures with mild noise, our proposed algorithms effectively recovered such degenerate cases. We also demonstrated effectiveness of our proposed algorithms through our additional experiments of monocular visual odometry. The proposed algorithms were effectively applied to outlier rejection in the monocular visual odometry and presented significantly faster computing time with similar accuracy. It is expected that the proposed algorithms can be applied to not only visual odometry/SLAM for mobile robots and vehicles but also many others such as 3-D reconstruction on turntables and visual place recognition for personal mobile devices [6].

Acknowledgment

This work was supported by the R&D program of the Korea Ministry of Land, Infrastructure and Transport (MOLIT) and the Korea Agency for Infrastructure Technology Advancement (KAIA), Train Station Service Standardization and the Development of Safety Management Automation Technology (14RTRP-B091404-01).

References

Figure 10: Rotational and translational errors were presented for seven algorithms and two references with respect to path length and speed.

Figure 11: Distribution of Sampson distances by the proposed algorithm are derived from inliers and outliers of synthetic data with varying the magnitude of non-planar motion noise. The inliers had 0.1 pixel of image noise and the outliers were generated from random association of point correspondences.


Figure 12: Trajectories were generated by monocular visual odometry. Each representative in planar and general algorithms was selected and included ground truth and VISO2-M for reference.