

On the Second Order Statistics of Essential Matrix Elements

M. Hossein Mirabdollah^(✉) and Bärbel Mertsching

GET Lab, University of Paderborn, Pohlwegstr 47-49, 33098 Paderborn, Germany
mirabdollah@get.upb.de

Abstract. In this paper, we investigate the second order statistics of essential matrix elements. Using the Taylor expansion for a rotation matrix up to second order terms and considering relatively high uncertainties for the rotation angles and translation parameters, a covariance matrix is obtained which includes the second order statistics of essential matrix elements. The covariance matrix is utilized along with the coplanarity equations and acts as a regularization term. Using the regularization term brings considerable improvements in the recovery of camera motion which will be proven based on simulation and different real image sequences.

Keywords: Essential matrix · Second order statistics

1 Introduction

Relative monocular camera motion (ego-motion) estimation based on the coplanarity constraint for calibrated cameras was initially addressed by Higgins with the well-known 8-point method [12]. In this method, eight matched points between two frames are used to obtain a 3×3 matrix known as *essential matrix*. Consequently, based on an essential matrix, the rotation matrix and the translation vector which define the motion of the camera (up to a scale) can be obtained. For uncalibrated cameras, in [8] an 8-point method is proposed to extract a 3×3 matrix known as *fundamental matrix* to recover camera motion and camera focal length simultaneously, if the principal point of the camera is known and the horizontal and vertical focal lengths are the same. Two main applications for the ego-motion estimation can be named: visual odometry [16, 19] and optical flow calculation [18].

Although the 8-point methods proposed by Higgins and Hartley were simple, they had poor performances in the presence of measurement noise, especially for the uncalibrated cameras. Therefore, different nonlinear optimization methods were proposed to estimate the essential and fundamental matrices iteratively [13]. Clearly, the iterative methods require good initial guesses to converge to correct solutions; otherwise, they would get stuck in local minima. On the other hand, Hartley in [9] claimed that if matched points are transformed such that their centroids become zero and their average distances from the centroids become $\sqrt{2}$, the performance of the 8-point method would improve noticeably. Nevertheless,

by comparing the results presented in the mentioned work, we can see that this method also performs poorly for measurement noise greater than 0.15 pixel.

One reason for the poor performance of the 8-point method is ignoring the dependencies of essential matrix elements. Hartley et. al proposed a 7-point method which used rank deficiency of fundamental matrices leading to enhancement in the estimation of camera motions [10]. Considering the fact that ego-motion is defined up to a scale, it can be defined with three rotation angles and two translation parameters. Thus, an essential matrix can also be determined with only five points. In this regard, David Nister in [15] introduced an algebraic solution using five matched points. He used rank deficiency and trace equations which hold for any essential matrix and obtained a polynomial equation of the order of ten, of which real roots yielded different valid solutions for the essential matrix. In [11], the authors proposed a more direct solution to reach the polynomial of the order of ten, which is the determinant of a 10×10 matrix of a variable. Obviously, implementations of both algorithms require symbolic processing, which makes their implementations inconvenient. Additionally, they would be very slow. Both 5-point methods are ad-hoc as there are 3 unknowns (x , y and z) but 10 equations; thus, none of them can guarantee that the solutions minimize all 10 equations. In Nister's method four equations are ignored and in the second method, we may confront contradictory elementary equations for an unknown in the forms of first, second or third order equations. In [14], the authors assumed large uncertainties for the motion parameters and obtained a variance for each coplanarity equation depending on the coordinates of the matching point. Consequently, the 8-point method was modified and solved based on a Mahalanobis criterion and the variance of each coplanarity equation to come up with a method which is more robust against the measurement noise.

In this paper, we demonstrate that in addition to the variance for each coplanarity equation, a covariance matrix for the nine elements of an essential matrix can be found, which encodes the second order statistics and dependencies of the elements. Then, in the essential matrix recovery, the covariance matrix will be augmented as a regularization term to direct the final solution in a feasible region defined by the physical constraints of cameras.

This paper is structured as follows: in Sect. 2, the 8-point, 7-point and 5-point methods are briefly introduced. The derivation of our proposed method is discussed in Sect. 3. Through the simulation in Sect. 4 the proposed method will be evaluated in two different cases. The evaluation of the proposed method in comparison with the 8-point, 7-point and 5-point methods based on the KITTI benchmark sequences is done in Sect. 5. Section 6 concludes this paper.

2 Essential Matrix Estimation

It is known that for a calibrated camera for each pair of matched points between two frames such as (x_1, y_1) and (x_2, y_2) , the following equation (coplanarity constraint) is valid:

$$\mathbf{p}_2^T E \mathbf{p}_1 = 0 \quad (1)$$

where $\mathbf{p}_1 = [x_1 \ y_1 \ 1]^T$, $\mathbf{p}_2 = [x_2 \ y_2 \ 1]^T$ and E is an essential matrix. An essential matrix is a 3×3 matrix with nine elements:

$$E = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} \tag{2}$$

If an essential matrix can be determined using a set of matched points between two frames, the relative camera motion can be estimated up to a scale factor. In the 8-point method, given eight matched points, a homogeneous equation system of nine unknown elements, consisting of eight equations, is formed. Thus, the essential matrix elements will be the null space of the matrix containing the coefficients of the equation system. Since the dependencies of essential matrix elements are ignored in this method, the solution could be sensitive to the measurement noise. Based on [4, 17], the dependencies of essential matrix elements can be formulated with the following equations:

$$\det(E) = 0 \tag{3}$$

$$(EE^T)E - \frac{1}{2}\text{trace}(EE^T)E = 0 \tag{4}$$

Using Eq. 3, the 7-point method is obtained. To utilize Eq. 3, a homogeneous equation system including seven coplanarity equations is formed and the coefficients of the equation system are stacked in a matrix. The matrix has a null space spanned by two vectors such as \mathbf{y} and \mathbf{z} . Thus, we have:

$$\mathbf{e} = \mathbf{y}\mathbf{y} + \mathbf{z} \tag{5}$$

where $\mathbf{e} = [e_1 \dots e_9]^T$. Plugging Eq. 5 in Eq. 3 results in a third order polynomial equation of y , which may have up to three real roots. It means that three valid essential matrices could explain the camera motion. However, if Eq. 4 is used, the solution which minimizes Eq. 4 can be selected as the best essential matrix. For the 5-point method, five coplanarity equations are used and therefore, we have $\mathbf{e} = \mathbf{w} + x\mathbf{x} + y\mathbf{y} + z\mathbf{z}$. Using the 10 equations obtained from Eqs. 3 and 4, a polynomial equation of the order of ten of the variable z is obtained. The real roots of the equation lead to different valid essential matrices. Generally, the true solution can be found based on the multiple observations of the matched points in multiple frames.

3 Second Order Statistics of an Essential Matrix

In this section, we find the mean and the covariance matrix for essential matrix elements and use them as regularization terms in different original N-point methods. In this regard, we first find the means and variances of motion parameters based on physical or equation constraints, and then the mean and covariance matrix of essential matrix elements will be calculated.

Concerning the translation vector, it can be verified that, if we find the null space vectors under the condition: $\mathbf{e}_1^2 + \dots + \mathbf{e}_9^2 = 1$ (for instance using singular value decomposition (SVD)), the following equation will hold:

$$t_x^2 + t_y^2 + t_z^2 = \frac{1}{2} \quad (6)$$

It means that $\mathbf{t} = [t_x, t_y, t_z]^T$ has a uniform distribution over a sphere with the radius $\frac{\sqrt{2}}{2}$. To obtain means, variances and correlations of the translation elements, we may need to marginalize out one and two variables from their joint probability distribution function: $p([t_x, t_y, t_z]^T)$. However, due to the symmetry of the distribution, it can be simply proven that mean values are zero: $\mu_{\mathbf{t}} = \mathcal{E}(\mathbf{t}) = [0 \ 0 \ 0]^T$ (\mathcal{E} is expectation operator) and also covariance between each two translation elements are zero: $\sigma_{t_x t_y} = \sigma_{t_x t_z} = \sigma_{t_y t_z} = 0$. Consequently, Eq. 6 results in $\sigma_{t_x}^2 = \sigma_{t_y}^2 = \sigma_{t_z}^2 = \frac{1}{6}$.

On the other hand, by encoding a rotation matrix using the three elementary rotations about the X , Y and Z axis: $R = R_Z(\psi)R_Y(-\theta)R_X(\phi)$ and using the Taylor expansion for the $\sin(\cdot)$ and $\cos(\cdot)$ up to desired orders, the rotation matrix can be written in the form of polynomials of ϕ , θ and ψ which makes the expectation operations much simpler. It should be mentioned that physical constraints of cameras and limitation of feature tracking methods force the angles to hardly exceed 30° . As a result, the terms with more than second order can be ignored. Thus, the following approximation for the rotation matrix holds:

$$R = \begin{bmatrix} 1 - \frac{\theta^2}{2} - \frac{\psi^2}{2} & -\psi - \phi\theta & -\theta + \phi\psi \\ \psi & 1 - \frac{\phi^2}{2} - \frac{\psi^2}{2} & -\phi - \theta\psi \\ \theta & \phi & 1 - \frac{\phi^2}{2} - \frac{\theta^2}{2} \end{bmatrix} \quad (7)$$

Obviously, for larger rotation angles higher order terms should be included, but the rest of the algorithm will be similar. Now, by considering maximum possible deviations from zero for the angles such as σ_ϕ , σ_θ and σ_ψ , we can model the angles as Gaussian random variables as: $\phi \sim \mathcal{N}(0, \sigma_\phi^2)$, $\theta \sim \mathcal{N}(0, \sigma_\theta^2)$ and $\psi \sim \mathcal{N}(0, \sigma_\psi^2)$. Additionally, we know that:

$$E = RT; \quad T = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \quad (8)$$

Using Eqs. 7 and 8, we can obtain the vector of essential matrix elements $\mathbf{e} = [e_1, e_2, \dots, e_9]^T$ as nonlinear functions of the motion parameters:

$$\begin{aligned} e_1 &= -t_z(\psi + \phi\theta) + t_y(\theta - \phi\psi), \quad e_2 = t_x(\phi\psi - \theta) + t_z(\frac{\psi^2 + \theta^2}{2} - 1), \\ e_3 &= t_y(1 - \frac{\psi^2 + \theta^2}{2}) + t_x(\psi + \phi\theta), \quad e_4 = -t_z(\frac{\phi^2 + \psi^2}{2} - 1) + t_y(\phi + \psi\theta), \\ e_5 &= -t_z\psi - t_x(\phi + \psi\theta), \quad e_6 = \psi t_y + t_x(\frac{\phi^2 + \psi^2}{2} - 1), \quad e_7 = t_z\phi + t_y(\frac{\phi^2 + \theta^2}{2} - 1), \\ e_8 &= -\theta t_z - t_x(\frac{\phi^2 + \theta^2}{2} - 1) \quad \text{and} \quad e_9 = -\phi t_x + \theta t_y. \end{aligned}$$

Using the predefined Gaussian distributions for the motion parameters, we can now calculate the first and second order statistics of \mathbf{e} . For the first order

statistics (mean vector), we can simply verify that all the nine elements include first order terms of the motion parameter and therefore we have:

$$\mu_{\mathbf{e}} = \mathcal{E}(\mathbf{e}) = \mathbf{0} \tag{9}$$

Since $\mu_{\mathbf{e}} = \mathbf{0}$, the second order statistics can be calculated as follows:

$$P_{\mathbf{e}} = \mathcal{E}(\mathbf{e}\mathbf{e}^T) \tag{10}$$

The calculation of the above covariance matrix is tedious and should be done by using a symbolic math package once. The covariance matrix looks as follows:

$$P_{\mathbf{e}} = \begin{bmatrix} \sigma_{e_1}^2 & 0 & 0 & 0 & \sigma_{e_1, e_5} & 0 & 0 & 0 & \sigma_{e_1, e_9} \\ 0 & \sigma_{e_2}^2 & 0 & \sigma_{e_2, e_4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{e_3}^2 & 0 & 0 & 0 & \sigma_{e_3, e_7} & 0 & 0 \\ 0 & \sigma_{e_2, e_4} & 0 & \sigma_{e_4}^2 & 0 & 0 & 0 & 0 & 0 \\ \sigma_{e_1, e_5} & 0 & 0 & 0 & \sigma_{e_5}^2 & 0 & 0 & 0 & \sigma_{e_5, e_9} \\ 0 & 0 & 0 & 0 & 0 & \sigma_{e_6}^2 & 0 & \sigma_{e_6, e_8} & 0 \\ 0 & 0 & \sigma_{e_3, e_7} & 0 & 0 & 0 & \sigma_{e_7}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{e_6, e_8} & 0 & \sigma_{e_8}^2 & 0 \\ \sigma_{e_1, e_9} & 0 & 0 & 0 & \sigma_{e_5, e_9} & 0 & 0 & 0 & \sigma_{e_9}^2 \end{bmatrix}$$

where

$$\begin{aligned} \sigma_{e_1}^2 &= \sigma_{\phi}^2 \sigma_{\psi}^2 \sigma_{t_y}^2 + \sigma_{\phi}^2 \sigma_{\theta}^2 \sigma_{t_z}^2 + \sigma_{\psi}^2 \sigma_{t_x}^2 + \sigma_{\theta}^2 \sigma_{t_y}^2, \\ \sigma_{e_1 e_5} &= \sigma_{\psi}^2 \sigma_{t_x}^2, \quad \sigma_{e_1 e_9} = \sigma_{\theta}^2 \sigma_{t_y}^2, \\ \sigma_{e_2}^2 &= \sigma_{\phi}^2 \sigma_{\psi}^2 \sigma_{t_x}^2 + \frac{3}{4} \sigma_{\psi}^4 \sigma_{t_x}^2 + \frac{1}{2} \sigma_{\psi}^2 \sigma_{\theta}^2 \sigma_{t_x}^2 - \sigma_{\psi}^2 \sigma_{t_x}^2 + \frac{3}{4} \sigma_{\theta}^4 \sigma_{t_x}^2 + \sigma_{\theta}^2 \sigma_{t_x}^2 - \sigma_{\theta}^2 \sigma_{t_x}^2 + \sigma_{t_x}^2, \\ \sigma_{e_2 e_4} &= -\frac{1}{4} \sigma_{\phi}^2 \sigma_{\psi}^2 \sigma_{t_x}^2 - \frac{1}{4} \sigma_{\phi}^2 \sigma_{\theta}^2 \sigma_{t_x}^2 + \frac{1}{2} \sigma_{\phi}^2 \sigma_{t_x}^2 - \frac{3}{4} \sigma_{\psi}^4 \sigma_{t_x}^2 - \frac{1}{4} \sigma_{\psi}^2 \sigma_{\theta}^2 \sigma_{t_x}^2 + \\ &\quad \sigma_{\psi}^2 \sigma_{t_x}^2 + \frac{1}{2} \sigma_{\theta}^2 \sigma_{t_x}^2 - \sigma_{t_x}^2, \\ \sigma_{e_3}^2 &= \sigma_{\phi}^2 \sigma_{\theta}^2 \sigma_{t_x}^2 + \frac{3}{4} \sigma_{\psi}^4 \sigma_{t_x}^2 + \frac{1}{2} \sigma_{\psi}^2 \sigma_{\theta}^2 \sigma_{t_x}^2 + \sigma_{\psi}^2 \sigma_{t_x}^2 - \sigma_{\psi}^2 \sigma_{t_x}^2 + \frac{3}{4} \sigma_{\theta}^4 \sigma_{t_x}^2 - \sigma_{\theta}^2 \sigma_{t_x}^2 + \sigma_{t_x}^2, \\ \sigma_{e_3 e_7} &= -\frac{1}{4} \sigma_{\phi}^2 \sigma_{\psi}^2 \sigma_{t_x}^2 - \frac{1}{4} \sigma_{\phi}^2 \sigma_{\theta}^2 \sigma_{t_x}^2 + \frac{1}{2} \sigma_{\phi}^2 \sigma_{t_x}^2 - \frac{1}{4} \sigma_{\psi}^2 \sigma_{\theta}^2 \sigma_{t_x}^2 + \frac{1}{2} \sigma_{\psi}^2 \sigma_{t_x}^2 - \frac{3}{4} \sigma_{\theta}^4 \sigma_{t_x}^2 + \\ &\quad \sigma_{\theta}^2 \sigma_{t_x}^2 - \sigma_{t_x}^2, \\ \sigma_{e_4}^2 &= \frac{3}{4} \sigma_{\phi}^4 \sigma_{t_x}^2 + \frac{1}{2} \sigma_{\phi}^2 \sigma_{\psi}^2 \sigma_{t_x}^2 + \sigma_{\phi}^2 \sigma_{t_x}^2 - \sigma_{\phi}^2 \sigma_{t_x}^2 + \frac{3}{4} \sigma_{\psi}^4 \sigma_{t_x}^2 + \sigma_{\psi}^2 \sigma_{\theta}^2 \sigma_{t_x}^2 - \sigma_{\psi}^2 \sigma_{t_x}^2 + \sigma_{t_x}^2, \\ \sigma_{e_5}^2 &= \sigma_{\phi}^2 \sigma_{t_x}^2 + \sigma_{\psi}^2 \sigma_{\theta}^2 \sigma_{t_x}^2 + \sigma_{\psi}^2 \sigma_{t_x}^2, \quad \sigma_{e_5 e_9} = \sigma_{\phi}^2 \sigma_{t_x}^2, \\ \sigma_{e_6}^2 &= \frac{3}{4} \sigma_{\phi}^4 \sigma_{t_x}^2 + \frac{1}{2} \sigma_{\phi}^2 \sigma_{\psi}^2 \sigma_{t_x}^2 - \sigma_{\phi}^2 \sigma_{t_x}^2 + \frac{3}{4} \sigma_{\psi}^4 \sigma_{t_x}^2 - \sigma_{\psi}^2 \sigma_{t_x}^2 + \sigma_{\psi}^2 \sigma_{t_x}^2 + \sigma_{t_x}^2, \\ \sigma_{e_6, e_8} &= -\frac{3}{4} \sigma_{\phi}^4 \sigma_{t_x}^2 - \frac{1}{4} \sigma_{\phi}^2 \sigma_{\psi}^2 \sigma_{t_x}^2 - \frac{1}{4} \sigma_{\phi}^2 \sigma_{\theta}^2 \sigma_{t_x}^2 + \sigma_{\phi}^2 \sigma_{t_x}^2 - \frac{1}{4} \sigma_{\psi}^2 \sigma_{\theta}^2 \sigma_{t_x}^2 + \frac{1}{2} \sigma_{\psi}^2 \sigma_{t_x}^2 + \\ &\quad \frac{1}{2} \sigma_{\theta}^2 \sigma_{t_x}^2 - \sigma_{t_x}^2, \\ \sigma_{e_7}^2 &= \frac{3}{4} \sigma_{\phi}^4 \sigma_{t_x}^2 + \frac{1}{2} \sigma_{\phi}^2 \sigma_{\theta}^2 \sigma_{t_x}^2 - \sigma_{\phi}^2 \sigma_{t_x}^2 + \sigma_{\phi}^2 \sigma_{t_x}^2 + \frac{3}{4} \sigma_{\theta}^4 \sigma_{t_x}^2 - \sigma_{\theta}^2 \sigma_{t_x}^2 + \sigma_{t_x}^2, \\ \sigma_{e_8}^2 &= \frac{3}{4} \sigma_{\phi}^4 \sigma_{t_x}^2 + \frac{1}{2} \sigma_{\phi}^2 \sigma_{\theta}^2 \sigma_{t_x}^2 - \sigma_{\phi}^2 \sigma_{t_x}^2 + \frac{3}{4} \sigma_{\theta}^4 \sigma_{t_x}^2 - \sigma_{\theta}^2 \sigma_{t_x}^2 + \sigma_{\theta}^2 \sigma_{t_x}^2 + \sigma_{t_x}^2 \text{ and} \\ \sigma_{e_9}^2 &= \sigma_{\phi}^2 \sigma_{t_x}^2 + \sigma_{\theta}^2 \sigma_{t_x}^2. \end{aligned}$$

Interestingly, the dependencies of the essential elements appeared as non-zero off-diagonal elements. Inspired by the work of [3], we applied a smoothing method for the essential matrix estimation by minimizing a cost function consisting of the

coplanarity equations as data terms and the covariance matrix as a smoothness term. The cost function will be:

$$C = \sum_{i=0}^N \frac{1}{\sigma_{c_i}} (A_i \mathbf{e})^T (A_i \mathbf{e}) + \mathbf{e}^T P_e^{-1} \mathbf{e} \tag{11}$$

where N is the number of matched points, A_i is a row vector of i^{th} coplanarity coefficients and σ_{c_i} is the standard deviation for the i^{th} coplanarity equation.

In the cost function, σ_{c_i} s play important roles. Assuming that the true motion of a camera and consequently the essential matrix is given, we can see in the presence of the measurement noise, the coplanarity equations will not hold exactly. In this case, each coplanarity equation has an almost zero mean Gaussian distribution if the measurement noises are also Gaussian. The variance of the distributions can be calculated as a function of the matched points and measurement noise variances. However, since the essential matrix is not known initially, considering large uncertainties for the motion parameters, we can obtain a relative measure which shows how much each coplanarity equation is allowed to deviate from zero. For the calculation of this measure, we assume matched points (x_1, y_1) and (x_2, y_2) have Gaussian distributions as:

$$x_1 = \mathcal{N}(\bar{x}_1, \sigma_p^2), y_1 = \mathcal{N}(\bar{y}_1, \sigma_p^2), x_2 = \mathcal{N}(\bar{x}_2, \sigma_p^2), y_2 = \mathcal{N}(\bar{y}_2, \sigma_p^2) \tag{12}$$

where $\bar{x}_1, \bar{y}_1, \bar{x}_2, \bar{y}_2$ are the measured values and σ_p is the standard deviation for the measurement noise. Using a similar procedure proposed in [14], and since we can always say $\sigma_{t_x}^2 = \sigma_{t_y}^2 = \sigma_{t_z}^2 = \frac{1}{6}$ (as already discussed), we have:

$$\sigma_c^2 = \frac{\sigma_p^2}{6} (4 + \sigma_\phi^2 + \sigma_\theta^2 + 2\sigma_\psi^2 + (1 + \sigma_\phi^2 + \sigma_\theta^2 + \sigma_\psi^2)(\bar{x}_1^2 + \bar{y}_1^2 + \bar{x}_2^2 + \bar{y}_2^2)) \tag{13}$$

The inverse of σ_p weights the importance of the data term: the larger σ_p , the more estimations of essential matrix elements are affected by the regularization term. Therefore, σ_p should be selected to achieve a balance between the data and regularization terms such that the method can work well for a wide range of measurement noise. Based on simulation and experimental results, we found $\sigma_p = 0.5$ pixel a proper selection.

For the minimization of the cost function Eq. 11, we reform it as follows:

$$C = (B\mathbf{e})^T P^{-1} (B\mathbf{e}) \tag{14}$$

where

$$B = \begin{bmatrix} A_1 \\ \vdots \\ A_N \\ I_{9 \times 9} \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} \sigma_{c,1}^2 & 0 & 0 & 0 \\ \vdots & \ddots & \dots & 0 \\ 0 & 0 & \sigma_{c,N}^2 & 0 \\ 0 & 0 & 0 & P_e \end{bmatrix} \tag{15}$$

Now we should find the vectors which minimize the cost function. It can be done using the SVD. We only need to calculate the SVD of the matrix $P^{-1/2}B = U\Sigma V^T$. Then the columns of V^T which are associated to the smallest eigenvalues can form the basis for different N-point methods.

4 Simulation

We have simulated a camera with the resolution 2000×2000 [pixels²] and a focal length of 1000 pixels. The camera was moved based on random translation and rotation parameters (ϕ, θ and ψ had the means 0.3 rad and standard deviations 0.1 rad) for two different types of motions: dominant forward and dominant side translations. During the motion, the camera could observe spatial points randomly distributed at depths from 10 to 20 m. The projection of the spatial points on the camera screen were added by zero mean Gaussian noises with varying standard deviation as measurement noise. The original 8-, 7- and 5-point methods were compared with the smoothed versions (named as 8-, 7- and 5- point-S). The standard deviations of the rotation angles ($\sigma_\phi, \sigma_\theta$ and σ_ψ) for the regularization constraint were set to 0.5 rad. For the evaluation, two different measures were used: the mean of magnitudes of errors between the estimated translation and true translation vectors (MME_t) and the mean of magnitudes of errors between the estimated rotation angles and the true angles (MME_a). Assuming the estimated translation vector at time k is $\hat{\mathbf{t}}_k$ and the ground truth is \mathbf{t}_k , the error between two vectors will be $\epsilon_{\mathbf{t},k} = \hat{\mathbf{t}}_k - \mathbf{t}_k$. Then, we have: $MME_t = \frac{1}{K} \sum_{k=1}^K \sqrt{\epsilon_{\mathbf{t},k}^T \epsilon_{\mathbf{t},k}}$, where K is the number of frames. If the estimated rotation matrix at time k is \hat{R}_k and the ground truth is R_k the rotation matrix error will be $R_{e,k} = R_k^T \hat{R}_k$. Consequently, using Eq. 8, we can calculate the error angles as : $\epsilon_{a,k} = \frac{180}{\pi} \sqrt{3 - trace(R_{e,k})}$. The mean square error for angles will be: $MME_a = \frac{1}{K} \sum_{k=1}^K \epsilon_{a,k}$. It is good to mention that since the regularization term is centered on the origin; not surprisingly, the proposed method could perform well if the rotation parameters were also distributed about the origin. Therefore, we selected random rotation angles with non-zero means to evaluate the method in a more challenging case.

The results can be seen in Tables 1 and 2. We can see that the regularization constraint improved the translation estimation slightly; however, it resulted in noticeable improvements for the rotation estimation in case of forward translation and good improvements for side translation. Surprisingly, we see that the 5-point method had a poor performance in rotation estimation (possibly due to

Table 1. Mean of magnitudes of errors for dominant forward translations.

σ_p [pixel]	0.1	0.5	1.0	1.5	2.0	0.1	0.5	1.0	1.5	2.0
	MME_t [m/frame]					MME_a [deg/frame]				
5-point-S	0.03	0.11	0.14	0.17	0.18	0.13	0.62	0.92	1.23	1.39
5-point	0.03	0.13	0.16	0.19	0.20	0.15	0.80	1.42	1.53	1.57
7-point-S	0.03	0.14	0.15	0.17	0.17	0.11	0.66	0.98	1.28	1.40
7-point	0.03	0.2	0.22	0.22	0.23	0.14	0.99	1.49	2.03	2.48
8-point-S	0.03	0.14	0.17	0.22	0.26	0.12	0.67	0.99	1.32	1.74
8-point [14]	0.03	0.20	0.26	0.32	0.38	0.14	1.10	1.48	2.04	2.38
8-point [9]	0.03	0.20	0.26	0.33	0.40	0.15	1.10	1.62	2.38	3.10

Table 2. Mean of magnitudes of errors for dominant side translations.

σ_p [pixel]	0.1	0.5	1.0	1.5	2.0	0.1	0.5	1.0	1.5	2.0
	MME_t [m/frame]					MME_a [deg/frame]				
5-point-S	0.05	0.16	0.24	0.29	0.32	1.86	5.89	9.73	10.92	14.19
5-point	0.06	0.17	0.25	0.29	0.33	2.06	6.09	10.28	11.86	14.54
7-point-S	0.10	0.35	0.55	0.63	0.68	0.70	2.15	3.38	4.44	5.10
7-point	0.11	0.36	0.59	0.67	0.72	0.75	2.81	4.51	5.92	6.77
8-point-S	0.11	0.33	0.53	0.66	0.74	0.71	2.19	3.93	5.76	6.88
8-point[14]	0.11	0.35	0.58	0.71	0.79	0.76	2.84	4.82	6.32	7.23
8-point[9]	0.11	0.35	0.57	0.71	0.79	0.76	2.84	5.04	6.34	7.27

numerical errors) in case of side motion. As a result, we can conclude that the 7-point-S method provides more reliable estimations if the camera experiences different types of motions.

5 Experimental Results

We used the KITTI dataset for visual odometry [6] to evaluate the performance of the proposed method. We applied the proposed regularization constraint for the 8-point, 7-point and 5-point methods and compared them with the original methods. The methods were implemented using C++ and ran on a computer with an Intel(R) Core(TM)2 Duo 3.33GHz CPU. All the methods were run in the context of a random sample consensus algorithm (RANSAC) [5], in which several essential matrices based on randomly selected N-matched points were calculated and then the best essential matrix which defined the flow of all points in the best way was selected.

As monocular ego-motion can generally be calculated only up to a scale factor, and the scale drift detection is not within the scope of this work, we removed the scale factor from the provided ground truths of the first eleven sequences. For the submitted data (for 7-point-S [1]), we have only used a fixed scale factor and obviously the submitted results were not meant for the ranking evaluations due to the large scale drifts in some of the sequences (for instance sequence 13) and also most of the other methods have used stereo or laser data which make the comparison unfair. What could be compared is the mean error angle. It can be seen that the proposed method (named as RMCPE) outperforms even some of the stereo based algorithms and also obviously has an error 2.7 times less than the 8-point based method (VISO2-M [7]). In this benchmark, the errors are calculated only based on a few frames, while in case of all frames, the differences will be much more as shown in Tables 3 and 4.

Since apparently instantaneous motion parameters of ground truth had drifts, we used two types of accumulating errors as evaluation measures: first, the mean of magnitudes of errors between all estimated and ground truth camera poses: MME_p , and second, the mean of magnitudes of errors between the estimated

Table 3. Mean of magnitudes of errors for estimated camera poses.

sequence	00	01	02	03	04	05	06	07	08	09	10
	MME_p [m/frame]										
7-point-S	23.4	25.0	12.8	2.4	0.9	22.9	5.5	12.2	58.7	11.2	8.3
7-point	32.3	139.3	30.0	4.7	1.5	27.4	10.5	14.9	61.5	25.6	12.0
5-point-S	9.2	53.5	24.0	3.0	1.1	30.1	8.3	10.5	58.0	12.0	9.2
5-point	9.4	78.7	36.3	3.6	1.3	30.1	8.8	10.8	67.3	16.6	9.4
8-point-S	25.3	40.7	47.3	6.2	1.6	28.6	13.00	13.8	67.6	16.2	14.8
8-point [14]	70.3	126.2	93.3	11.2	1.6	67.3	10.2	22.8	82.9	44.0	24.3
8-point[9]	70.0	443.5	97.4	23.6	5.3	66.5	27.3	22.7	124.9	42.0	38.6

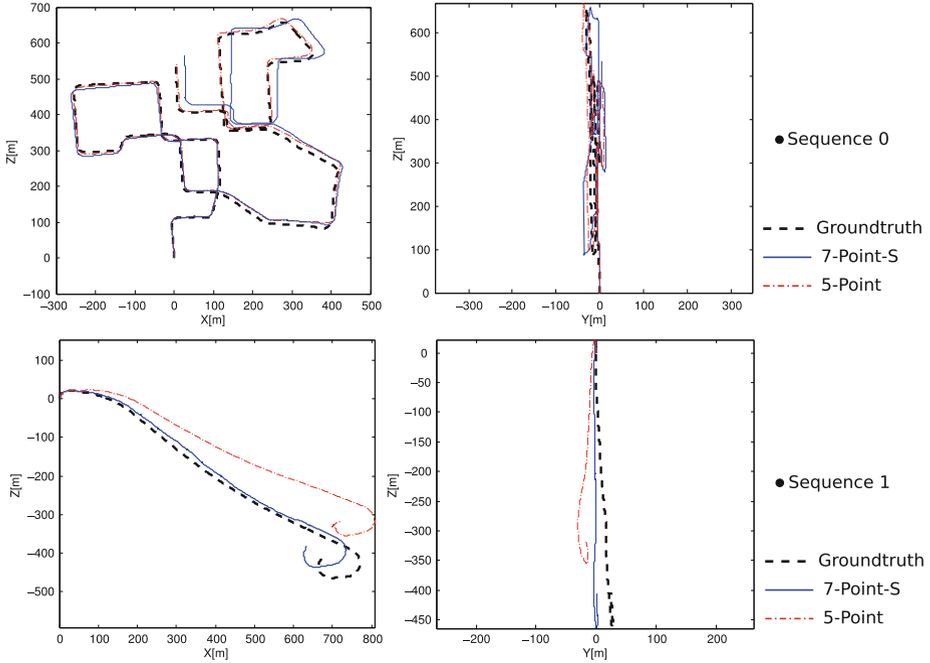
Table 4. Mean of magnitudes of errors for estimated camera angles.

sequence	00	01	02	03	04	05	06	07	08	09	10
	MME_a [deg/frame]										
7-point-S	0.53	0.59	0.42	0.22	0.03	0.18	0.27	0.17	0.40	0.34	0.22
7-point	1.50	4.28	1.37	0.20	0.14	0.86	1.08	1.05	0.64	0.93	1.26
5-point-S	0.31	1.19	0.80	0.23	0.07	0.94	0.53	0.60	0.31	0.43	0.30
5-point	0.33	1.75	1.37	0.24	0.1	1.03	0.54	0.72	0.30	0.47	0.30
8-point-S	1.51	1.63	1.69	0.37	0.07	0.64	1.10	0.86	0.73	0.71	1.09
8-point [14]	1.72	4.12	2.86	0.57	0.07	2.77	0.89	1.65	1.31	2.00	0.74
8-point [9]	2.09	10.92	2.98	0.59	1.23	2.58	1.14	1.63	2.12	1.90	1.61

and ground truth angles at all poses: MME_a . For corner feature tracking, we used pyramid Lucas-Kanade optical flow (implemented in opencv) [2], which worked very well in real-time. Nevertheless, the percentage of outliers and the amount of measurement noise were relatively high. The comparison results can be seen in Table 3 (MME_p), Table 4 (MME_a) and Table 5 (elapsed time). The regularization constraint brings noticeable improvement to the original 8- and 7- point methods and in some sequences sound improvement for the 5-point method. In average, the 7-point-S method gave the best result. To understand the results better, it should be mentioned that in the estimation of essential matrices the most challenging case is when the base line is small and the rotations are high. In this case, measurement noise can destructively affect the recovery of camera motion, especially the rotation matrix. This situation occurs very often in the KITTI dataset when the car is driven trough sharp bends. Consequently, the 8-point methods had very poor performance almost in all of the sequences. On the contrary, applying the regularization constraint made the 8-point-S method have almost the same performance as the 7-point method at a much lower elapsed time which shows that it could be a proper option for real-time applications. The 5-point(-S) method, as expected, had a better performance than the 7-point method and even for the sequence 0, it outperformed 7-point-S (Fig. 1-top) but for the other sequences, it performed either similar to or worse than the 7-point-S (for instance Fig. 1-bottom). We analyzed the

Table 5. Average elapsed time (E. T.) per frame for different methods.

Method	5-point-S	5-point	7-point-S	7-point	8-point-S	8-point [14]	8-point [9]
E. T. [s/frame]	4.82	4.12	0.39	0.32	0.21	0.12	0.13

**Fig. 1.** The estimated and ground truth paths for Seq. 0 (top) and Seq.1 (bottom). To avoid visual confusion for the Seq. 0, the path was plotted by the frame 3650.

sequences more precisely and noticed that in the sequences in which the outlier ratio is high, the 5-point method works more robust than other methods. The reason lies in RANSAC part, in which N -matched points are selected randomly. Clearly, the less N matched points are selected, the more it is possible that the matched points do not contain any outliers. On the other hand, based on simulation results, the 5-point(-S) method does not perform well to estimate rotation matrices in case of dominant side motions (occurred in sharp bends) which explains why it cannot outperform the 7-point-S method.

6 Conclusion

A new method was developed to include prior knowledge about the ranges of the motion parameter in the form of a covariance matrix for the estimation of essential matrices. By applying the method for the original 8- and 7-point

method, great improvements were obtained which have been proven based on simulated and real datasets. Future work could be applying the same analysis to the case of uncalibrated cameras.

References

1. Relative Monocular Camera Pose Estimation (RMCPE). http://www.cvlibs.net/datasets/kitti/eval_odometry.php
2. Bouguet, J.Y.: Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm (2000). http://robots.stanford.edu/cs223b04/algorithm_tracking.pdf
3. Dellaert, F., Kaess, M.: Square root SAM: simultaneous localization and mapping via square root information smoothing. *Int. J. Rob. Res.* **25**(12), 1181–1203 (2006)
4. Faugeras, O.: *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge (1993)
5. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), 381–395 (1981)
6. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: *Conference on Computer Vision and Pattern Recognition* (2012)
7. Geiger, A., Ziegler, J., Stiller, C.: StereoScan: dense 3D reconstruction in real-time. In: *Intelligent Vehicles Symposium*, pp. 963–968 (2011)
8. Hartley, R.I.: Estimation of relative camera positions for uncalibrated cameras. In: Sandini, G. (ed.) *ECCV 1992*. LNCS, vol. 588, pp. 579–587. Springer, Heidelberg (1992)
9. Hartley, R.I.: In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(6), 580–593 (1997)
10. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge (2004). ISBN: 0521540518
11. Li, H., Hartley, R.I.: Five-point motion estimation made easy. In: *18th International Conference on Pattern Recognition*, pp. 630–633 (2006)
12. Longuet-Higgins, H.C.: A computer algorithm for reconstructing a scene from two projections. *Nature* **293**, 133–135 (1981)
13. Luong, Q., Deriche, R., Faugeras, O., Papadopoulos, T.: On determining the fundamental matrix: analysis of different methods and experimental results. Technical report 1894, INRIA (1993)
14. Mirabdollah, M.H., Mertsching, B.: Single camera motion estimation: modification of the 8-point method. In: Lee, J., Lee, M.C., Liu, H., Ryu, J.-H. (eds.) *ICIRA 2013, Part I*. LNCS, vol. 8102, pp. 117–128. Springer, Heidelberg (2013)
15. Nistér, D.: An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(6), 756–777 (2004)
16. Strasdat, H., Montiel, J.M.M., Davison, A.J.: Scale drift-aware large scale monocular SLAM. In: *Proceedings of Robotics: Science and Systems* (2010)
17. Tsai, R.Y., Huang, T.S.: Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**(1), 13–27 (1984)
18. Yamaguchi, K., McAllester, D., Urtasun, R.: Robust monocular epipolar flow estimation. In: *CVPR*, pp. 1862–1869 (2013)
19. Zhao, L., Huang, S., Yan, L., Dissanayake, G.: Parallax angle parametrization for monocular SLAM. In: *IEEE International Conference on Robotics and Automation*, pp. 3117–3124 (2011)