

## Article

# Comparative Study of Relative-Pose Estimations from a Monocular Image Sequence in Computer Vision and Photogrammetry <sup>†</sup>

Tserennadmid Tumurbaatar <sup>1,\*</sup>  and Taejung Kim <sup>2</sup><sup>1</sup> Department of Information and Computer Sciences, National University of Mongolia, Ulaanbaatar 14200, Mongolia<sup>2</sup> Department of Geoinformatic Engineering, Inha University, 100 Inharo, Michuhol-Gu, Incheon 22212, Korea; tezid@inha.ac.kr

\* Correspondence: tserennadmid@seas.num.edu.mn; Tel.: +976-75754400-3410

<sup>†</sup> This paper is an extended version of conference paper published in 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), Chongqing, China, 27–29 June 2018.

Received: 18 February 2019; Accepted: 17 April 2019; Published: 22 April 2019



**Abstract:** Techniques for measuring the position and orientation of an object from corresponding images are based on the principles of epipolar geometry in the computer vision and photogrammetric fields. Contributing to their importance, many different approaches have been developed in computer vision, increasing the automation of the pure photogrammetric processes. The aim of this paper is to evaluate the main differences between photogrammetric and computer vision approaches for the pose estimation of an object from image sequences, and how these have to be considered in the choice of processing technique when using a single camera. The use of a single camera in consumer electronics has enormously increased, even though most 3D user interfaces require additional devices to sense 3D motion for their input. In this regard, using a monocular camera to determine 3D motion is unique. However, we argue that relative pose estimations from monocular image sequences have not been studied thoroughly by comparing both photogrammetry and computer vision methods. To estimate motion parameters characterized by 3D rotation and 3D translations, estimation methods developed in the computer vision and photogrammetric fields are implemented. This paper describes a mathematical motion model for the proposed approaches, by differentiating their geometric properties and estimations of the motion parameters. A precision analysis is conducted to investigate the main characteristics of the methods in both fields. The results of the comparison indicate the differences between the estimations in both fields, in terms of accuracy and the test dataset. We show that homography-based approaches are more accurate than essential-matrix or relative orientation-based approaches under noisy conditions.

**Keywords:** object; pose estimation; motion parameters; computer vision; photogrammetry; single camera

## 1. Introduction

The three-dimensional (3D) spatial context is becoming an integral part of 3D user interfaces in everyday life, such as modeling applications, virtual and augmented reality, and gaming systems. The 3D user interfaces are all characterized by a user input that involves 3D position ( $x$ ,  $y$ ,  $z$ ) or orientation (yaw, pitch, roll), and 3D tracking is a key technology to recovering the 3D position and orientation of an object relative to the camera or, equivalently, the 3D position and orientation of the camera relative to the object in physical 3D space. Recovering the position and orientation of an object from images is becoming an important task in the fields of computer vision and photogrammetry. A necessity has arisen for users in both fields to know more about the differences in, and behavior

of, the approaches developed by the computer vision and photogrammetry communities. In both fields, mathematical motion models are expressed with an epipolar constraint under perspective geometry. Linear approaches have mainly been developed by the computer vision community, while the photogrammetric community has generally considered non-linear solutions to recover 3D motion parameters.

Especially in computer vision, determining the 3D motion of an object from image sequences starts from image matching, as the establishment of point correspondences extracted from two or more images. The automatic relative orientation of image sequences, with assumptions of calibrated or uncalibrated cameras (unknown intrinsic parameters), has been widely investigated. An essential matrix is defined as the set of linear homogeneous equations found by establishing eight point correspondences. By decomposing the essential matrix, the relative pose parameters of the two perspective views were computed [1–3]. Pose estimations based on decomposition of the essential matrix using fewer than eight point correspondences were developed for various applications, such as visual servoing and robot control [4–7]. Moreover, the pose parameters of the camera relative to a planar object can be estimated by decomposing a homography matrix through point correspondences. Numerical and analytical methods for pose estimation based on homography decomposition were introduced, in detail, in [8]. Other authors introduced non-linear and linear solutions for determining the pose parameters based on the decomposition of a homography matrix in augmented reality and robot control applications [9–14].

In photogrammetry, the determination of 3D pose from a set of 3D points was originally known as resectioning. To determine the position and orientation (extrinsic parameters) of the right image, relative to the left image, from a sufficient set of tie-points is well known as a relative orientation process. In this photogrammetric task, the intrinsic parameters are assumed to be known, instead the analysis of the images, to discover the corresponding points between the images, but no ground truth is assumed. Mathematically, relative orientation parameters, as pose parameters between frames in a sequence, can be determined by collinearity or coplanarity equations [15–19].

Various solutions for pose estimation from the point correspondences have been published, introducing their accuracy to image noise and their computation speed against state-of-the-art methods for both simulated and real image data. The non-linear and the linear approaches to solve the perspective- $n$ -point (PnP) problem for determining the position and orientation of the camera based on the 3D reference points and their 2D projections have been developed for various applications, and quantitative comparisons of their results with the state-of-the-art approaches were carried out. The authors in [20] developed an algorithm to solve the PnP problem using the 3D collinearity model. The experimental results on simulated and real data were compared with the efficient PnP (EPnP) algorithm [21], the orthogonal iterative (OI) approach [22], and a non-linear solution to relative position estimation [23]. An algebraic method was developed, in [24], to solve the perspective-three-point (P3P) problem of computing the rotation and position of a camera. The experimental results demonstrated that the accuracy and precision of the method was comparable to the existing state-of-the-art methods, such as the first complete analytical solution to the P3P problem [25], the perspective similar triangle (PST) method [26], and the closed-form solution to the P3P problem [27]. The authors in [28] proposed a method for the PnP problem, for determining the position and orientation of a calibrated camera from known reference points. The authors investigated the performance of the proposed method and compared the accuracy with the leading PnP methods, such as the non-linear least-squares solution [29], the efficient PnP (EPnP) algorithm [21], the robust non-iterative method (RPnP) [26], the direct least-squares (DLS) method [30], and the optimal solution (OPnP) [31]. A novel set of linear solutions to the pose estimation problem from an image of  $n$  points and  $n$  lines was presented in [32]. Their results were compared to other recent linear algorithms, such as the non-linear least-squares solutions [29,33], the direct recovery and decomposition of the full projection matrix [34], and the  $n$  point linear algorithms [35,36]. Combining two fields, a evaluation of epipolar resampling methods developed for image sequences with an intension of stereo applications was reported in [37].

A theoretical comparison of the 3D metric reconstruction in the cultural heritage field and its accuracy assessment with the results from commercial software were discussed in [38]. The relationship between photogrammetry and computer vision was examined in [39]. However, the evaluations, comparisons, and the differences in the theoretical and practical perspectives of pose estimation methods based on the epipolar constraint developed in each field, using a common dataset, are missing. On the other hand, even if the processing steps for the two techniques seem to be different approaches, some computer vision techniques were implemented for increasing the workflow automation of the photogrammetric techniques. Moreover, an additional investigation of the automatic techniques that regulate the processing steps is needed.

The aim of this paper is to investigate the approaches to determining the 3D motion of a moving object in real-time image sequences taken by a monocular camera. First, we determined a novel automatic technique to perform motion estimations with a single camera. Common processing steps were implemented in the estimations. We try to use a web camera instead of other specialized devices, such as accelerometers, gyroscopes, and global positioning systems (GPS), to sense 3D motion, as the web camera is cheap and widely available. Furthermore, we believe a single camera is more suitable for real-time computation, because we can maintain one processing chain. Second, we analyzed and compared the linear approaches with the non-linear approaches. We used four estimation methods for recovering the 3D motion parameters, based on tracked point correspondences. Two of them were developed with linear solutions in computer vision, and two of them were developed with non-linear solutions in photogrammetry. We argue that most previous investigations for pose estimations have not thoroughly compared the methods developed in both fields. It is, at present, difficult to define the border between the two methodologies. Moreover, a number of applications have recently been developed by linking the techniques developed in both fields. It is, thus, important to show the differences in the methods and to confirm which one is the most suitable technique to be used, based on the needs of the given application. We aim to identify the differences and better understand how those differences may influence outcomes. First, we review general principles and mathematical formulations for motion models in linear and non-linear solutions. Next, we explain the implementation steps of pose estimations, regarding image sequences from a single camera. Third, we point out the practical differences at the experimental level, with the test datasets, and analyze the results. This paper is organized as follows. Mathematical models of the proposed methods in computer vision and photogrammetry are described in Section 2. Processing techniques to implement pose estimations in both fields, with test datasets, are presented in Section 3. The comparison results are discussed in Section 4. A discussion and the conclusion are presented in Section 5.

## 2. General Motion Model

We reviewed the basic geometry of the proposed motion models to determine the relative pose of a moving object in image sequences, once the correspondence points have been established. We assumed that a stationary camera had taken an image sequence of the moving object through its field of view. The camera coordinate system is fixed with its origin,  $O$ , at the optical center. The  $z$  axis is pointing in the direction of the view, coinciding with the optical axis. The image plane is located at a distance equal to the focal length, which is considered to be unity.

Consider point  $P_1$  on a rigid object at time  $t_1$  moving to point  $P_2$  on a rigid object at time  $t_2$ , with respect to the camera coordinate system (as shown in Figure 1). Point  $P_1$  is projected at point  $p_1$  on the image plane under perspective projection. Similarly, point  $P_2$  is projected at point  $p_2$  on the image plane. The object-space coordinates of point  $P_1$  are  $X_1 \in R^3$ , and the image-space coordinates are defined as  $x_1 \in R^3$ . The object-space coordinates of point  $P_2$  are  $X_2 \in R^3$ , and the image-space coordinates are defined as  $x_2 \in R^3$ .

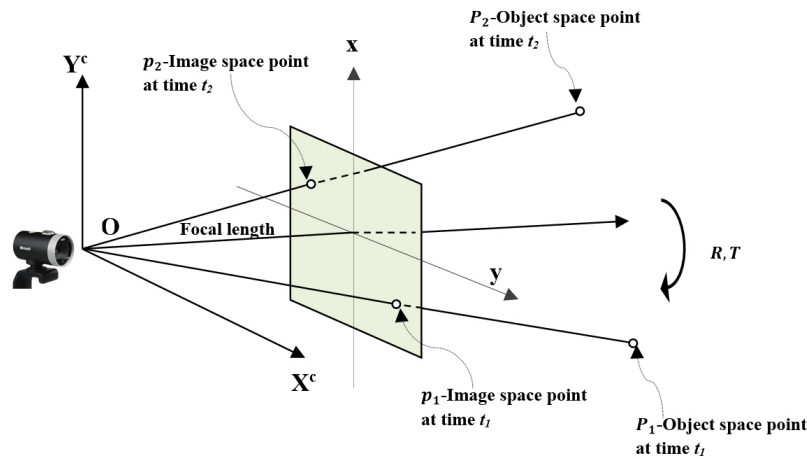


Figure 1. Geometry of the imaging system.

To summarize, our problem is as follows:

Given two image views with correspondences  $(p_1, p_2)$ , find 3D rotations and 3D translations.

We have that  $P_1$  and  $P_2$  are related by the rotation matrix  $R$  and translational vector  $T$ , due to the rigidity constraint of the object motion:

$$X_2 = RX_1 + T. \quad (1)$$

It is obvious, from the geometry of Figure 1, that  $Rx_1$ ,  $x_2$ , and  $T$  are coplanar, which can be written in matrix form as

$$x_2^T \hat{T} R x_1 = 0 \quad (2)$$

This is called coplanarity, or an epipolar constraint, written in a triple product of vectors.

## 2.1. Recovering Motion Parameters from an Essential Matrix

We can reformulate Equation (2) as follows:

$$x_2^T E x_1 = 0, \quad (3)$$

where

$$E = \hat{T}R \in R^{3 \times 3} \quad \text{and} \quad \hat{T} = \begin{pmatrix} 0 & -B_z & B_y \\ B_z & 0 & -B_x \\ B_y & B_x & 0 \end{pmatrix}.$$

Equation (3) is linear and homogeneous in the nine unknowns. Given  $N$  point correspondences, we can write Equation (3) in the form:

$$A_N [e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9]^T = 0. \quad (4)$$

The relative pose between two views can be found from matrix Equation (3), encoded by a well-known essential matrix. Due to the theorem in [40],  $E$  has singular value decomposition (SVD), defined as:

$$E = U \Sigma V^T, \quad (5)$$

where  $U$ ,  $V$ , and  $\Sigma$  are chosen such that  $\det(U) > 0$ ,  $\det(V) > 0$ , and  $\Sigma = \{1, 1, 0\}$ . Furthermore, the following formulas give the two distinct solutions for the rotation and translation vectors from the essential matrix.

$$R = U \begin{bmatrix} 0 & \mp 1 & 0 \\ \pm 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^T, \hat{T} = U \begin{bmatrix} 0 & \mp 1 & 0 \\ \pm 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} U^T. \quad (6)$$

One of the four possible solutions corresponds to the true solution for Equation (6). The correct solution can be chosen by enforcing a constraint called the cheirality test [41]. The cheirality test basically means that the triangulated scene points should have positive depth, and scene points should be in front of the camera.

## 2.2. Recovering Motion Parameters from a Homography Matrix

Consider the points  $P_i, i = 1, 2$  to be on a 2D plane,  $\pi$ , in 3D space. The plane  $\pi$  has unit normal vector  $n = [n_1, n_2, n_3]$ , and  $d$  ( $d > 0$ ) denotes the distance from plane  $\pi$  to the optical center of the camera. Suppose the optical center of the camera never passes through plane  $\pi$ , as shown in Figure 2.

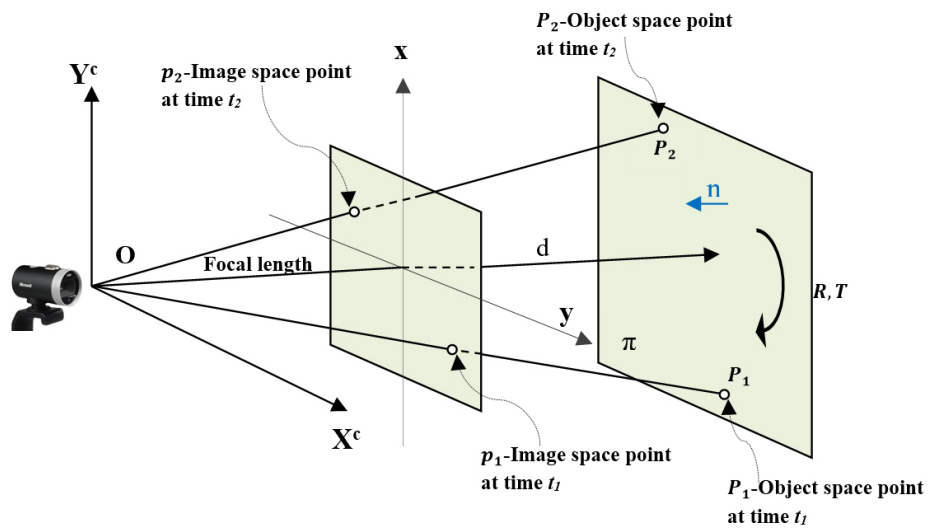


Figure 2. Geometry of a planar object in an image sequence.

Then, we formulate the following equation, normalizing the translational vector  $T$  by the plane depth  $d$  from Equation (1):

$$X_2 = RX_1 + T = (R + \frac{1}{d}Tn^T)X_1 = HX_1 \quad (7)$$

$$H = (R + \frac{1}{d}Tn^T) \in R^{3 \times 3}$$

We call the matrix  $H$  a planar homography matrix, as it depends on motion parameters  $\{R, T\}$  and structure parameters  $\{n, d\}$ . We have a homography mapping induced by the plane  $\pi$ , since it denotes a linear transformation from  $X_1 \in R^3$  to  $X_2 \in R^3$ , due to the scale ambiguity in the term  $\frac{1}{d}T$  in Equation (7):

$$x_2 \sim Hx_1. \quad (8)$$

Equation (8) is linear and homogeneous in the nine unknowns. Given  $N$  point correspondences, we can write Equation (8) in the form:

$$B_N[h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9]^T = 0. \quad (9)$$

After we have recovered the  $H$  matrix, using at least four point correspondences, we can decompose the matrix into its motion and structure parameters by SVD ([8,42–44]):

$$H = U\Sigma V^T, \quad (10)$$

where  $U$  and  $V$  are orthogonal matrices, and  $\Sigma$  is a diagonal matrix which contains a singular value for  $H$ . Then, we also obtain four solutions: Two completely different solutions, and their opposites for decomposing the  $H$  matrix. In order to reduce the number of physically possible solutions, we impose a positive depth constraint, having  $n^T e > 0, e = [0, 0, 1]$ , as the camera can see only points in front of it.

### 2.3. Recovering Motion Parameters from Relative Orientation

In photogrammetry, determining the relative position and orientation of the first view of the frame, with respect to the next view of the frame in a sequence, is well-known as a relative orientation process. In general, the geometric relationship between a ground point  $P$  and its corresponding image points,  $p_1$  and  $p_2$ , at two time instants, is formulated by the coplanarity constraint from Equation (2). As shown in Equation (2), the essential matrix determined in computer vision is mathematically identical to a coplanarity condition equation in photogrammetry, which has been well-confirmed [19,37].

By equivalently reformulating the non-linear equations of the coplanarity condition in [17] into Equation (2), we have the following:

$$\begin{bmatrix} D_1 \\ E_1 \\ F_1 \end{bmatrix} = \begin{bmatrix} ix_1 \\ jx_1 \\ kx_1 \end{bmatrix}, \quad \begin{bmatrix} D_2 \\ E_2 \\ F_2 \end{bmatrix} = \begin{bmatrix} ix_2 \\ jx_2 \\ kx_2 \end{bmatrix}, \quad (11)$$

$$b = \begin{bmatrix} B_x & B_y & B_z \end{bmatrix}, \quad R = \begin{bmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{bmatrix},$$

where

$$i_x = \cos \alpha * \cos \kappa; \quad j_x = -\cos \phi * \cos \kappa;$$

$$k_x = \sin \phi;$$

$$i_y = \sin \omega * \sin \phi * \cos \kappa + \cos \omega * \sin \kappa; \quad j_y = -\sin \omega * \sin \phi * \sin \kappa + \cos \omega * \cos \kappa;$$

$$k_y = -\sin \omega * \cos \phi;$$

$$i_z = -\cos \omega * \sin \phi * \cos \kappa + \sin \omega * \sin \kappa; \quad j_z = \cos \omega * \sin \phi * \sin \kappa + \sin \omega * \cos \kappa; \text{ and}$$

$$k_z = \cos \omega * \cos \phi.$$

Then, the triple-scalar product of the three vectors is as follows:

$$B_x \cdot (E_1 F_2 - E_2 F_1) + B_y \cdot (F_1 D_2 - F_2 D_1) + B_z \cdot (D_1 E_2 - D_2 E_1) = 0. \quad (12)$$

From this non-linear equation, the unknowns can be obtained by Taylor linearization in a least-squares solution. The three parameters of the rotation matrix  $R$ , and two components of the base vector, are estimated. In Figure 3,  $\omega$ ,  $\phi$ , and  $\kappa$  are rotations about the  $x$ ,  $y$ , and  $z$  axes, respectively; and  $B_x$ ,  $B_y$ , and  $B_z$  are translations about the  $x$ ,  $y$ , and  $z$  axes, respectively. The iterative method for solving a non-linear equation requires an initial estimate for each unknown parameter for convergence to the correct answer. We set the position and orientation angles of the first-view by making all six variables equal to zero ( $\omega_1 = \phi_1 = \kappa_1 = 0^\circ$ ,  $B_x = B_y = 0$ ), because we considered no movement of the object in the first-view. The second-view orientation is set as  $\omega_2 = \phi_2 = \kappa_2 = 0^\circ$ , by assuming a constant fixed value for  $B_x$  or  $B_y$  based on simple parallax differences between the two views, as is illustrated in Figure 3.



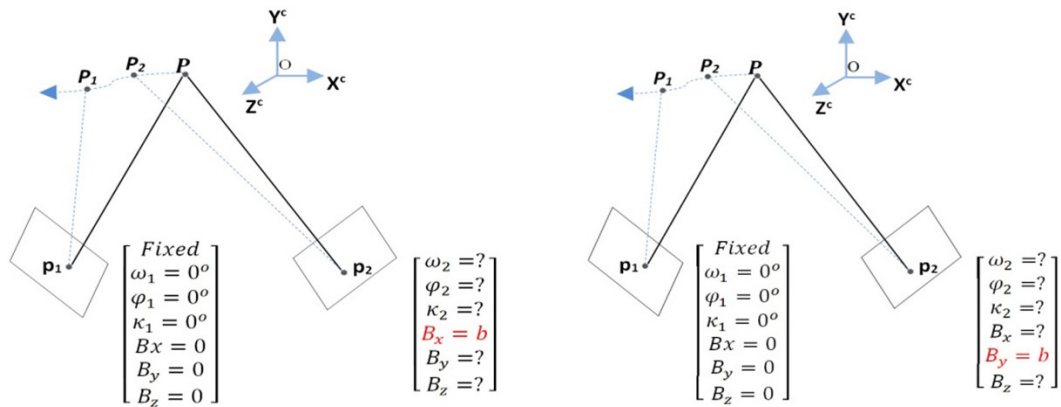


Figure 3. Parameter setup configurations of relative orientation-based estimations.

#### 2.4. Recovering Motion Parameters from Homography-Based Relative Orientation

By reformulating matrix Equation (8), we can obtain the motion and structure parameters from the following non-linear equation, as mapping from the first to the next image is given by homography:

$$x_2 \cong (R + \frac{1}{d}Tn^T)x_1. \quad (13)$$

The seven unknown parameters of this equation can be described as five parameters ( $\omega$ ,  $\phi$ ,  $\kappa$ ,  $B_x$  or  $B_y$ , and  $B_z$ ) for the relative orientation, and two parameters ( $n_1, n_2$ ) for the plane in object-space. Similarly, this equation can be solved by Taylor linearization in a least-squares solution, with an a priori fixed value for  $B_x$  or  $B_y$ , as illustrated in Figure 3. We set the initial value for the variable  $n_3$  as 1.

#### 2.5. Summary of Approaches

We have reviewed estimations for determining the motion of a rigid body from 2D to 2D point correspondences. If the rank of  $A_N$  in Equation (4) is 8,  $E$  can be determined uniquely, within a scale factor, and with an eight point correspondence. Once  $E$  is determined,  $R$  and  $T$  can be determined uniquely. If the rank of  $A_N$  is less than 8, then either we have a pure rotation case or the surface assumption of eight points is violated. However, the surface assumption (position) of the given eight points is very important. It can be easily shown that, if these points form a degenerate configuration, the estimation will fail [44]. This approach is sensitive to noise. Pure rotation of the object in the images generates numerical ill-conditioning, so enough translation in the motion of the object is needed in the images for the estimation (given in Section 2.1) to operate correctly. With a four point correspondence, the rank of  $B_N$  in Equation (9) is 8 when the 3D points lie on a plane [45]. In this case, a pure rotation can be handled by the planar approach. Then, we have a unique solution for  $H$ , within a scale factor. Once  $H$  is determined,  $R$  and  $T$  can be determined uniquely. If the rank of  $B_N$  is more than 8, it is considered that the 3D points do not lie on a plane. As both  $E$  and  $H$  give four possible solutions to  $(R, T)$ , the depths of the 3D points being observed by the camera are all positive. Therefore, one of the four solutions will be chosen, based on the positive depth constraint. To find a least-squares solution of the non-linear equations in (12) and (13), using the iterative method is not computationally expensive; however, we need a good initial value for its convergence to the correct solution. We set the initial values for  $B_x$  or  $B_y$  differently, depending on the experimental settings. However, we expect that the solution for non-linear approaches is generally unique with six or more point correspondences for the five motion parameters.

### 3. Implementation Steps for Estimations and Datasets

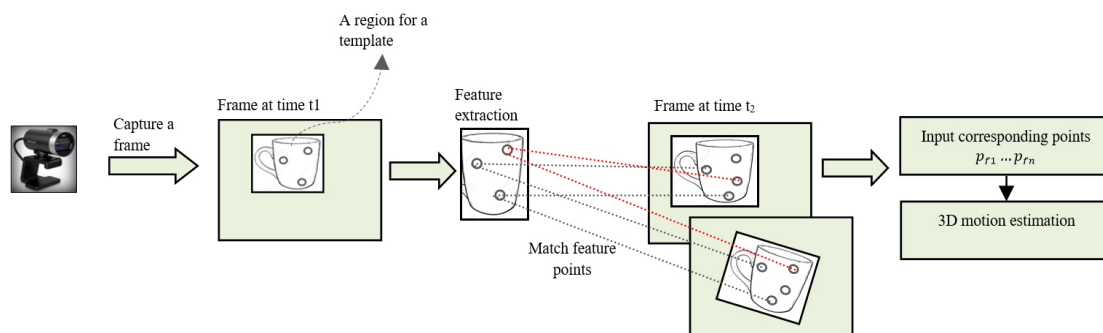
#### 3.1. Methodology

In this section, we explain the processing steps for implementing the four estimations and how these have to be considered with a single camera. Figure 4 provides the process flow for implementing the estimations. We estimate the position and orientation of the moving object from two views, such as between a template region and the next consecutive frames, using point correspondences.

First, an initial frame is captured at time  $t_1$ , and a template region is extracted from it. Then, feature points for the extracted template region are computed by using a scale-invariant feature transform (SIFT) feature extractor. The template region is a rectangle, in which the extracted points are circles in a frame at time  $t_1$ , as shown in Figure 4. SIFT features are robust to perspective changes. This advantage is preserved in the matching process, as well. Here, we assume that  $n$  feature points are extracted to the template region and tracked through each of the next  $f$  frames. In other words, template extraction could be done automatically by analyzing moving parts of the object between frames through these feature points.

Secondly, after capturing the next frame of the moving object at time  $t_2$ , feature points of the next frame are extracted. Feature points in the next frame are matched with feature points in the template, as described in Figure 4. The best matches for corresponding points between the two views are found by a brute-force matcher. Outliers among the matched points are eliminated by a random sample consensus (RANSAC) method [46] before estimation of the motion parameters. We used different RANSAC methods for each of the four proposed approaches. Homography-based RANSAC was applied before estimation of the motion parameters from homography-based methods in both fields. Essential matrix-based RANSAC was applied before estimation of the motion parameters from an essential matrix. Relative orientation-based RANSAC was applied before estimation of the motion parameters from the relative-orientation method.

Third, when all processing steps are accumulated, as explained above, the proposed estimations (as defined in Section 2) are used to estimate the motion parameters.



**Figure 4.** Process flow of the implementation performing the estimations.

#### 3.2. Test Datasets

We implemented the estimations in both fields with Visual C++, Open Source Computer Vision Library (OpenCV) 2.4.9, and Open Graphics Library (OpenGL) on a PC with an Intel Core i5 CPU at 3.0 GHz with 4096 MB of RAM and a Microsoft LifeCam. The camera's intrinsic parameters, including focal length, principle point, and lens distortion coefficients, were determined with the GML Camera Calibration Toolbox [47].










We experimentally examined and compared the performance of estimations with a real dataset (created from real scenes) and a simulated dataset (created from an OpenGL library). Video sequences for the moving object were captured at a resolution of  $640 \times 360$  pixels.

For creation of the real dataset, we captured video sequences while changing object positions along each axis and rotating the object around each axis in front of a static camera. Note that the



object position in the initial frame was arbitrarily fixed before changing it. Translation of the object along the  $x$  and  $y$  axes varied by up to 200 mm. Translation of the object along the  $z$  axis varied between 300–800 mm from the camera. Rotations of the object around the  $x$  and  $y$  axes varied by up to 20 degrees, and rotation of the object around the  $z$  axis varied by up to 90 degrees. We used thousands of image sequences, comprised of different textured 3D objects and planar objects, to check the accuracy of estimations of the motion parameters. Each object template was composed of approximately one hundred frames. Object templates used in the experiments and their feature descriptions are listed in Table 1.

**Table 1.** Datasets used in the experiments.

Dataset for Real Scenes									
Dataset ID	Dataset for 3D Objects					Dataset for Planar Objects			
	F_ID21	F_ID22	F_ID23	F_ID24	F_ID25	F_ID31	F_ID32	F_ID33	F_ID34
Template									
Feature points	115	104	78	151	159	62	86	189	126
Size					150 × 120				
Dataset for Simulated Scenes									
Dataset ID	S_ID1	S_ID2	S_ID3	S_ID4	S_ID5	S_ID6	S_ID7	S_ID8	
3D object	polygon	cube	pyramid	polygon	cube	polygon	cube	pyramid	
Edge length	15	15	15	13	13	10	10	10	
Feature points	15	25	25	13	25	21	15	16	

For creation of the simulated dataset, we used different 3D objects, such as polygons, pyramids, and cubes with different edge lengths, by creating them in Euclidean space. We manually measured up to 25 feature points on the 3D objects. The simulated sequences for moving 3D objects were created through perspective projection by keeping the object in the field of view throughout the sequences. Descriptions of the simulated dataset are summarized in Table 1. Translations of the 3D objects along the  $x$ ,  $y$ , and  $z$  axes varied by up to 15 units. Rotations of the 3D objects around the  $x$  and  $y$  axes varied by up to 30 degrees. Rotation around the  $z$  axis varied by up to 90 degrees. We used thousands of simulated sequences for the experiments. Each 3D object template was composed of approximately one hundred frames.

#### 4. Performance of 3D Motion Estimation

##### 4.1. Performance Analysis for Estimations with a Real Dataset

After estimating the motion parameters, we analyzed the accuracy of the proposed four estimations in Section 2. We renamed the estimation methods to simplify the notation in the experimental results. Decomposition of the essential matrix is notated as (CV\_E). Decomposition of the homography matrix is notated as (CV\_H). Relative orientation is notated as (PM\_RO). Homography-based relative orientation is notated as (PM\_H).

First, we estimated the motion parameters for the real dataset in two different cases. We checked the accuracy of the estimated rotation parameters by comparing them with true (known) rotation parameters. For a true reference value, we manually measured the corresponding points between the template and the next consecutive frames of the object. Using the measured corresponding points, the precise 3D motion was estimated for each frame. In the first case, the object in the image sequences was rotated along only one of the  $x$ ,  $y$ , or  $z$  axes separately, and translated along one of the axes. In the second case, we estimated the motion parameters of an object from image sequences rotated by a combination of rotations around the  $x$ ,  $y$ , and  $z$  axes, and translated along an  $x$ ,  $y$ , or  $z$  axis differently. For example, the object was rotated around the  $y$  axis by  $5^\circ$ , and simultaneously rotated around the  $z$  axis by up to  $25^\circ$ .

For both cases, we analyzed the mean, maximum (Max), and minimum (Min) errors for the four estimations, as summarized in Table 2. We also checked the accuracy of the estimated rotation parameters around each of the three axes by comparing them with true (known) rotation parameters and analyzing the root mean square error (RMSE). The RMSEs for rotation around the  $x$ ,  $y$ , and  $z$  axes are shown in Tables 3–5, respectively; which are summarized for each dataset used in the experiments. A graphical representation of the comparisons is shown in Figure 5.

**Table 2.** Comparison of error analysis for rotations around  $x$ ,  $y$ , and  $z$  axes for real scenes.

	Comparison of 3D Objects /Degrees/				Comparison of Planar Objects /Degrees/			
	PM_H	CV_H	CV_E	PM_RO	PM_H	CV_H	CV_E	PM_RO
<b>Max</b>	1.867	1.993	1.997	1.863	1.828	1.837	1.801	1.983
<b>Min</b>	0.00045	0.001	0.005	0.0004	0.0001	0.00014	0.013	0.0009
<b>Mean</b>	0.538	0.564	1.096	0.527	0.312	0.391	0.677	0.408

**Table 3.** Comparison of error analysis for rotation around  $x$  axis for real scenes. RMSE, root mean square error.

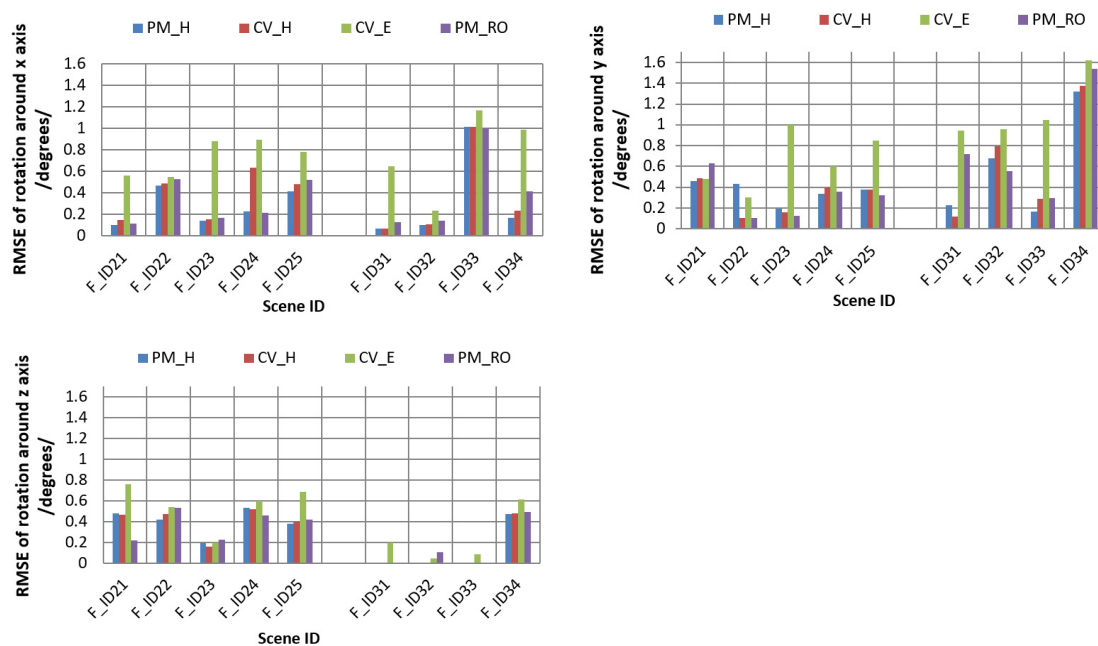
RMSEs of the Estimated Rotation around the $x$ axis ( $\omega = 1^\circ \sim 20^\circ$ Degrees)					
	Dataset	PM_H	CV_H	CV_E	PM_RO
<b>F_ID21</b>	$\omega = 5^\circ, B_y = -5$ cm	0.103	0.146	0.560	0.112
<b>F_ID22</b>	$\phi = 6^\circ, B_y = 20$ cm	0.465	0.488	0.551	0.530
<b>F_ID23</b>	$\omega = 10^\circ, B_x = -10$ cm	0.144	0.155	0.878	0.168
<b>F_ID24</b>	$\phi = 13^\circ, B_x = 15$ cm	0.228	0.636	0.896	0.215
<b>F_ID25</b>	$\omega = 4^\circ, B_z = 8$ cm	0.411	0.484	0.778	0.522
<b>F_ID31</b>	$\phi = 15^\circ, B_y = -13$ cm	0.065	0.068	0.647	0.127
<b>F_ID32</b>	$\kappa = 20^\circ, B_x = -10$ cm	0.100	0.108	0.233	0.143
<b>F_ID33</b>	$\phi = 15^\circ, B_z = -15$ cm	1.012	1.013	1.167	1.003
<b>F_ID34</b>	$\kappa = 20^\circ, B_x = 10$ cm	0.165	0.233	0.991	0.415

**Table 4.** Comparison of error analysis for rotation around the  $y$  axis for real scenes.

RMSEs of the Estimated Rotation around the $y$ axis ( $\phi = 1^\circ \sim 20^\circ$ Degrees)					
	Dataset	PM_H	CV_H	CV_E	PM_RO
<b>F_ID21</b>	$\phi = 5^\circ, B_z = -8$ cm	0.458	0.488	0.477	0.627
<b>F_ID22</b>	$\omega = 6^\circ, B_y = 9$ cm	0.429	0.106	0.302	0.101
<b>F_ID23</b>	$\omega = 10^\circ, B_x = -13$ cm	0.195	0.161	1.001	0.128
<b>F_ID24</b>	$\phi = 13^\circ, B_x = 18$ cm	0.335	0.401	0.601	0.358
<b>F_ID25</b>	$\omega = 15^\circ, B_z = 5$ cm	0.381	0.381	0.847	0.323
<b>F_ID31</b>	$\kappa = 15^\circ, B_z = -7$ cm	0.224	0.116	0.947	0.716
<b>F_ID32</b>	$\omega = 20^\circ, B_y = -6$ cm	0.681	0.790	0.954	0.552
<b>F_ID33</b>	$\kappa = 15^\circ, B_x = 15$ cm	0.163	0.287	1.048	0.298
<b>F_ID34</b>	$\phi = 20^\circ, B_z = -20$ cm	1.323	1.377	1.62	1.535

**Table 5.** Comparison of error analysis for rotations around the  $z$  axis for real scenes.

RMSEs of the Estimated Rotation around the $z$ axis ( $\kappa = 1^\circ \sim 90^\circ$ Degrees)					
	Dataset	PM_H	CV_H	CV_E	PM_RO
<b>F_ID21</b>	$\omega = 5^\circ, B_z = 6$ cm	0.477	0.468	0.758	0.219
<b>F_ID22</b>	$\omega = 6^\circ, B_y = -4$ cm	0.419	0.475	0.537	0.532
<b>F_ID23</b>	$\kappa = 10^\circ, B_x = 16$ cm	0.195	0.161	0.201	0.228
<b>F_ID24</b>	$\omega = 13^\circ, B_z = -7$ cm	0.535	0.521	0.6002	0.458
<b>F_ID25</b>	$\kappa = 15^\circ, B_x = -17$ cm	0.381	0.401	0.684	0.423
<b>F_ID31</b>	$\kappa = 15^\circ, B_y = -4$ cm	0.0001	0.0001	0.198	0.0001
<b>F_ID32</b>	$\phi = 20^\circ, B_x = -10$ cm	0.002	0.0006	0.046	0.103
<b>F_ID33</b>	$\kappa = 10^\circ, B_z = 5$ cm	0.004	0.005	0.085	0.003
<b>F_ID34</b>	$\phi = 20^\circ, B_z = -10$ cm	0.472	0.482	0.615	0.495

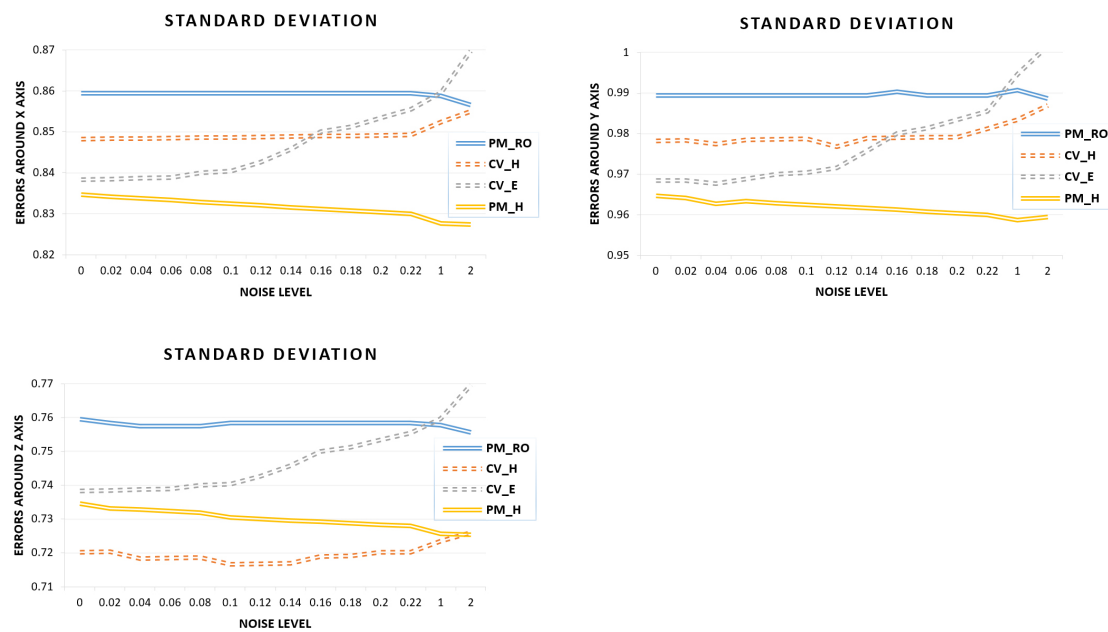


**Figure 5.** Comparisons of error analyses for combinations of rotations around the  $x$ ,  $y$ , and  $z$  axes for real scenes.

As we can see, from Tables 2–5, the RMSEs were small, and the rotation results were accurate for each of the four approaches in both of the test cases. In particular, the homography-based methods PM\_H and CV\_H produced more accurate results for image sequences of the moving planar object, since the planar pattern is dominant in the test datasets F\_ID31 to F\_ID34. On the other hand, PM\_H and CV\_H produced more negligible and comparable errors, among the four methods with noisy correspondences for the datasets of both 3D and planar objects. Among them, the motion parameters from PM\_H estimations are especially accurate. We observe that the estimation errors for the motion parameters slightly increased in the dataset for cases F\_ID33 in Table 3 and F\_ID34 in Table 4, due to noisy feature correspondences. For these datasets, the object was translated close to the camera along the  $z$  axis and was rotated around the  $x$  or  $y$  axis to a large degree. Object rotation around the  $x$  or  $y$  axis at large degrees creates the side-effect of scaling. In this case, matched feature correspondences are noisy and unstable in their matched positions. Generally, we can see that the CV\_E estimation method was sensitive to noisy measurements in feature correspondences, as is seen in most of the results in Tables 2–5. Another interesting result is that the motion parameters estimated by PM\_RO were the most accurate, compared to the other three approaches for the 3D object datasets F\_ID21 to F\_ID25.

#### Stability Analysis for Estimations with Varying Noises

This experiment examines the stability of the all estimations with the real dataset. We chose  $n = 50$  random point correspondences as noise, nearing each of the six numbers:  $\sigma_1 = 0.06$ ,  $\sigma_2 = 0.1$ ,  $\sigma_3 = 0.16$ ,  $\sigma_4 = 0.2$ ,  $\sigma_5 = 1$ , and  $\sigma_6 = 2$ , and computed the rotations around the  $x$ ,  $y$ , and  $z$  axes. The experimental results, as illustrated in Figure 6, give the standard deviation of the rotation errors around the  $x$ ,  $y$ , and  $z$  axes. From Figure 6, we observe that the non-linear approaches (PM\_H and PM\_RO) were more stable with increasing noises, as compared to the linear approaches (CV\_H and CV\_E). We see that the linear approaches were critical and showed accuracy degradation with increasing noise. The experimental results strongly support the fact that the linear approaches are very sensitive to noise. The computation of CV\_E became unstable when reaching the middle of the noise level range. We also observed that the errors from the computation of CV\_H were stable at the low noise levels but, as the noise level increased, the stability decreased.



**Figure 6.** Error analysis of rotations around the  $x$ ,  $y$ , and  $z$  axes for varying noises.

#### 4.2. Performance Analysis for Estimations with a Simulated Dataset

We checked the accuracy of the estimated motion parameters for simulated datasets in two different test cases. We checked the accuracy of the estimated rotation parameters by comparing them with true (known) rotation parameters. For a true reference value, we created synthetic corresponding points between the template and the next consecutive frames of the 3D object. Using the created corresponding points, a precise 3D motion was estimated for each frame. In the first case, the 3D object in the simulated sequences was rotated along one of the  $x$ ,  $y$ , or  $z$  axes separately, and translated along an  $x$ ,  $y$ , or  $z$  axis. In the second case, the 3D object was rotated by a combination of rotation degrees around the  $x$ ,  $y$ , and  $z$  axes, and translated by 15 units along one of the  $x$ ,  $y$ , or  $z$  axes.

For both cases, we checked the accuracy of the estimated rotation parameters through motion estimations. To check the performance of the four estimations, we compared the estimated rotation parameters with the known rotation parameters. We analyzed the mean, Max, and Min errors in the results of the estimations and summarize them in Table 6. We also computed the RMSEs of the estimated rotation results by comparing them against the true rotation parameters for each of the four estimations. The dataset summaries of RMSEs for rotations around the  $x$ ,  $y$ , and  $z$  axes are given in Tables 7–9, respectively. A graphical representation of the comparisons is shown in Figure 7.

**Table 6.** Comparison of error analysis for rotations around the  $x$ ,  $y$ , and  $z$  axes for the simulated dataset.

Error Comparison for 3D Objects /Degrees/				
	PM_H	CV_H	CV_E	PM_RO
<b>Max</b>	1.547	1.515	1.631	1.489
<b>Min</b>	0.015	0.009	0.013	0.003
<b>Mean</b>	0.549	0.498	0.762	0.461

**Table 7.** Comparison of error analysis for rotation around the  $x$  axis for the simulated dataset.

RMSEs of the Estimated Rotations around the $x$ axis ( $\omega = 1^\circ \sim 30^\circ$ Degrees)					
	Dataset	PM_H	CV_H	CV_E	PM_RO
S_ID1	$\omega = 5^\circ, B_x = -13$	0.471	0.338	0.365	0.282
S_ID2	$\omega = 10^\circ, B_y = 8$	0.546	0.750	0.528	0.402
S_ID3	$\omega = 15^\circ, B_y = -5$	0.290	0.173	0.506	0.141
S_ID4	$\omega = 25^\circ, B_x = 10$	1.369	1.385	1.296	1.390
S_ID5	$\phi = 14^\circ, B_z = -3$	0.724	0.397	0.598	0.296
S_ID6	$\phi = 15^\circ, B_z = -5$	0.484	0.527	1.256	0.396
S_ID7	$\kappa = 25^\circ, B_z = 7$	0.740	0.990	0.823	0.711
S_ID8	$\phi = 20^\circ, B_z = 12$	0.569	0.575	0.675	0.52

**Table 8.** Comparison of error analysis for rotation around the  $y$  axis for the simulated dataset.

RMSEs of the Estimated Rotations around the $y$ axis ( $\phi = 1^\circ \sim 30^\circ$ Degrees)					
	Dataset	PM_H	CV_H	CV_E	PM_RO
S_ID1	$\phi = 8^\circ, B_x = -15$	0.929	0.881	0.743	0.581
S_ID2	$\phi = 15^\circ, B_y = -3$	0.517	0.577	0.725	0.616
S_ID3	$\phi = 22^\circ, B_y = 3$	0.702	0.723	0.377	0.692
S_ID4	$\phi = 28^\circ, B_x = 8$	1.442	1.481	1.492	0.497
S_ID5	$\omega = 25^\circ, B_z = -5$	0.639	1.059	0.851	0.637
S_ID6	$\kappa = 13^\circ, B_z = -5$	0.681	0.513	0.221	0.493
S_ID7	$\kappa = 18^\circ, B_z = 10$	0.344	0.313	0.568	0.245
S_ID8	$\kappa = 15^\circ, B_z = 15$	0.548	0.687	0.425	0.549

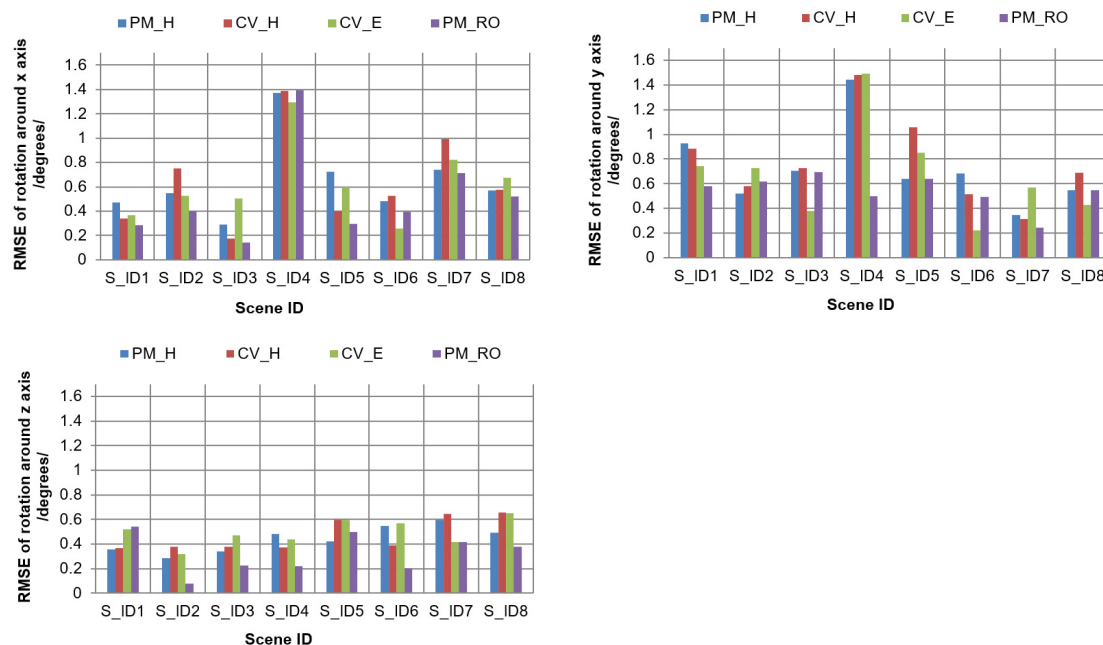
**Table 9.** Comparison of error analysis for rotations around the  $z$  axis for the simulated dataset.

RMSEs of the Estimated Rotations around the $z$ axis ( $\kappa = 1^\circ \sim 90^\circ$ Degrees)					
	Dataset	PM_H	CV_H	CV_E	PM_RO
S_ID1	$\kappa = 5^\circ, B_x = -12$	0.353	0.369	0.522	0.542
S_ID2	$\kappa = 6^\circ, B_y = -2$	0.283	0.375	0.316	0.077
S_ID3	$\kappa = 10^\circ, B_y = 5$	0.341	0.377	0.468	0.226
S_ID4	$\kappa = 13^\circ, B_x = 13$	0.484	0.374	0.439	0.221
S_ID5	$\kappa = 20^\circ, B_z = -5$	0.424	0.597	0.598	0.496
S_ID6	$\phi = 15^\circ, B_z = -13$	0.549	0.387	0.571	0.205
S_ID7	$\phi = 20^\circ, B_z = 6$	0.594	0.643	0.418	0.415
S_ID8	$\phi = 15^\circ, B_z = -15$	0.495	0.655	0.651	0.380

As we see in Tables 6–9, the four estimations in photogrammetry and computer vision produced small errors in both test cases of the simulated datasets. The success rates of the estimations stayed within the range for large motions in the moving 3D object; this means that large perspective changes did not affect estimation accuracy. Moreover, this implies that all four estimations worked successfully when favorable corresponding points were provided. Specifically, the motion parameters obtained by PM\_RO were the most accurate, compared to the other three approaches. It is confirmed, again, that PM\_RO outperformed the other three approaches for datasets of 3D objects.

Generally, relative orientation-based approaches are formulated in non-linear problems, requiring an initial guess for each unknown parameter; however, these approaches are more robust to unique solutions for motion parameters with noisy point correspondences. Moreover, these approaches do not require additional computation to choose a correct solution for the motion parameters and estimate the rotation and translation parameters directly, compared with the linear approaches. Combining the two cases of test datasets, we observed that the PM\_H approach produced more accurate results for planar objects in real-image sequences, and the PM\_RO approach produced more accurate results for 3D objects in real and simulated sequences. It was shown that the approach based on relative orientation produced the most accurate results. On the other hand, the results with a real dataset are very interesting. Regardless of linear or non-linear approaches, homography-based methods

outperformed other (essential matrix or relative orientation-based) methods under noisy situations. In particular, we observed that the homography-based non-linear approach worked better than the relative orientation-based non-linear approach. In other words, the homography-based non-linear approach supports the motivation for linking the techniques developed in photogrammetry and computer vision.



**Figure 7.** Comparisons of error analyses for combinations of rotations around the  $x$ ,  $y$ , and  $z$  axes for simulated scenes.

#### 4.3. Qualification of Fast Processing

Once we define the template region in the initial frame, real-time processing is started with all computational steps. We extracted feature points by CPU-based or GPU-based parallel processing with a GeForce GTX 550 Ti graphics card.

To assess real-time performance, we measured the processing time for SIFT feature extraction with different numbers of extracted feature points. The speed of feature extraction was almost independent of the number of feature points, due to parallel processing. The processing time in SIFT feature extraction speeds up with a large number of feature points, by 0.2 s. Moreover, processing time can speed up with more powerful CPUs and graphics cards.

We also measured the processing time of four motion estimations, by including RANSAC-based elimination of outliers for different numbers of correspondences. For this, the comparison results are summarized in Table 10.

As we can see in Table 10, the processing times of all four methods were very fast with large numbers of feature correspondences. Generally, the processing speed of the PM\_H estimation method is slower than the other three methods, due to its non-linear estimation. The processing speeds of the PM\_RO and CV\_H functions are the fastest. The total processing time could be defined as the sum of processing times for feature extraction and estimation of motion parameters.



**Table 10.** Comparison of processing times for 3D motion estimation.

Feature Number	Processing Time /s/			
	PM_H	CV_H	CV_E	PM_RO
227	0.002	0.0001	0.001	0.001
190	0.004	0.001	0.002	0.001
172	0.003	0.001	0.002	0.00001
156	0.003	0.001	0.002	0.001
110	0.004	0.0001	0.002	0.001
89	0.003	0.001	0.001	0.00001
46	0.003	0.001	0.001	0.00001
22	0.004	0.0001	0.0001	0.00001
9	0.019	0.0001	0.0001	0.00001

## 5. Conclusions

In this study, we investigated and compared pose estimations in computer vision and photogrammetry. All estimations were implemented with common datasets and processing steps, which are suitable to use with a single camera. The results from the comparisons demonstrated the main differences in, and computational behavior of, the approaches developed by the computer vision and photogrammetry communities. In order to check the performance of these methods, we estimated the 3D motion of moving planar and 3D objects, at the experimental level, by using corresponding points between a template region and subsequent frames. Outlier corresponding points were eliminated by different RANSAC-based methods, which were adapted for each estimation. The experimental results were evaluated for the measured corresponding points, which were measured manually for both synthetic and real images. The results of the estimations in both fields were accurate, even with high variations of translation and rotation changes, with decent point correspondences. For noisy situations, the methods based on homography produced smaller errors. Comparisons of both fields highlight the robust performance of the estimations and 3D applications in each field. The processing speed was close to real-time. Due to the motion of the moving objects, the estimations diverged or converged to the correct solution. In further research, we plan to determine which method is more suitable in estimating the motions in a moving object.

**Author Contributions:** T.T. designed the work, performed the experiments, and wrote the manuscript; T.K. supervised the entire process.

**Funding:** The work in this paper was supported by “Cooperative Research Program for Agriculture Science and Technology Development (No. PJ01350003)” of Rural Development Administration, Republic of Korea, by the National University of Mongolia (No. P2017-2469) and by an MJEED Grant (No. JR14B16).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Weng, J.; Huang, T.; Ahuja, N. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **1989**, *11*, 451–476. [[CrossRef](#)]
2. Hartley, R.; Zisserman, A. *Multiple View in Computer Vision*; Cambridge University Press: Cambridge, UK, 2000.
3. Chesi, G. Estimation of the camera pose from image point correspondences through the essential matrix and convex optimization. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009.
4. Fathian, K.; Gans, N.R. A new approach for solving the five-point relative pose problem for vision-based estimation and control. In Proceedings of the 2014 American Control Conference, Portland, OR, USA, 4–6 June 2014.
5. Sarkis, M.; Diepold, K.; Huper, K. A Fast and Robust Solution to the Five-Point Relative Pose Problem using Gauss-Newton Optimization on a Manifold. In Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing, Honolulu, HI, USA, 15–20 April 2007.

6. Batra, B.; Nabbe, D.; Hebert, M. An alternative formulation for five point relative pose problem. In Proceedings of the 2007 IEEE Workshop on Motion and Video Computing, Austin, TX, USA, 23–24 February 2007.
7. Chesi, G.; Hashimoto, K. Camera pose estimation from less than eight points in visual servoing. In Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, 26 April–1 May 2004.
8. Malis, E.; Vargas, M. Deeper understanding of the homography decomposition for vision-based control. In *Research Report 6303*; INRIA: Sophia Antipolis Cedex, France, 2007.
9. Kim, K.; Lepetit, V.; Woo, W. Scalable real-time planar targets tracking for digilog books. *Vis. Comput.* **2010**, *26*, 1145–1154. [[CrossRef](#)]
10. Bazargani, H.; Bilaniuk, O.; Laganie're, R. A fast and robust homography scheme for real-time planar target detection. *J. Real-Time Image Proc.* **2015**, *15*, 739–758. [[CrossRef](#)]
11. Mae, Y.; Choi, J.; Takahashi, H.; Ohara, K.; Takubo, T.; Arai, T. Interoperable vision component for object detection and 3D pose estimation for modularized robot control. *Mechatronics* **2011**, *21*, 983–992. [[CrossRef](#)]
12. Marchand, E.; Uchiyama, H.; Spindler, F. Pose Estimation for Augmented Reality: A Hands-On Survey. *IEEE Trans. Vis. Comput. Graph.* **2016**, *22*, 2633–2651. [[CrossRef](#)] [[PubMed](#)]
13. Zhu, B. Self-Recalibration of the Camera Pose via Homography. In Proceedings of the 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems, Xiamen, China, 29–31 October 2010.
14. Pirchheim, C.; Reitmayr, G. Homography-based planar mapping and tracking for mobile phones. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, Basel, Switzerland, 26–29 October 2011.
15. Horn, B.K.P. Relative Orientation. *Int. J. Comput. Vis.* **1990**, *4*, 59–78. [[CrossRef](#)]
16. Schenk, T. *Digital Photogrammetry*; Terra Science: Laurelville, OH, USA, 1999.
17. Wolf, P.; DeWitt, B.; Wilkinson, B.E. *Elements of Photogrammetry with Applications in GIS*, 4th ed.; McGraw-Hill Science: New York, NY, USA, 2014.
18. Kim, J.; Kim, H.; Lee, T.; Kim, T. Photogrammetric Approach for Precise Correction of Camera Misalignment for 3D Image Generation. In Proceedings of the IEEE International Conference on Consumer Electronics, Las Vegas, NV, USA, 13–16 January 2012; pp. 396–397.
19. McGlone, J.C.; Mikhail, E.M.; Bethel, J. *Manual of Photogrammetry*, 5th ed.; American Society of Photogrammetry and Remote Sensing: Bethesda, MD, USA, 2004.
20. Fan, B.; Du, Y.; Cong, Y. Robust and accurate online pose estimation algorithm via efficient three-dimensional collinearity model. *IET Comput. Vis.* **2013**, *7*, 382–393. [[CrossRef](#)]
21. Lepetit, V.; Moreno-Noguer, F.; Fua, P. Epnp: An accurate  $O(n)$  solution to the pnp problem. *Int. J. Comput. Vis.* **2009**, *81*, 155–166. [[CrossRef](#)]
22. DeMenthon, D.; Davis, L.S. Model-based object pose in 25 lines of code. *Int. J. Comput. Vis.* **1995**, *15*, 123–141. [[CrossRef](#)]
23. Weng, J.; Ahuja, N.; Huang, T.S. Optimal motion and structure estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **1993**, *15*, 864–884. [[CrossRef](#)]
24. Wang, P.; Xu, G.; Wang, Z.; Cheng, Y. An efficient solution to the perspective-three-point pose problem. *Comput. Vis. Image Underst.* **2018**, *166*, 81–87. [[CrossRef](#)]
25. Gao, X.S.; Hou, X.R.; Tang, J.L. Complete solution classification for the perspective-three-point problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 930–943.
26. Li, S.; Xu, C.; Xie, M. A robust  $O(n)$  solution to the perspective-n-point problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 383–390. [[CrossRef](#)] [[PubMed](#)]
27. Kneip, L.; Scaramuzza, D.; Siegwart, R. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR2011), Colorado Springs, CO, USA, 20–25 June 2011; pp. 2969–2976. [[CrossRef](#)]
28. Wang, P.; Xu, G.; Cheng, Y.; Yu, O. A simple, robust and fast method for the perspective-n-point Problem. *Pattern Recognit. Lett.* **2018**, *108*, 31–37. [[CrossRef](#)]
29. Lu, C.P.; Hager, G.D.; Mjolsness, E. Fast and globally convergent pose estimation from video images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *6*, 610–622. [[CrossRef](#)]
30. Hesch, J.A.; Roumeliotis, S.I. A direct least-squares (dls) method for pnp. In Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 383–390.

31. Zheng, Y.; Kuang, Y.; Sugimoto, S.; Astrom, K.; Okutomi, M. Revisiting the pnp problem: A fast, general and optimal solution. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 2344–2351.
32. Ansar, A.; Daniilidis, A. Linear pose estimation from points or lines. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 578–589. [[CrossRef](#)]
33. Kumar, R.; Hanson, A.R. Robust Methods for Estimating Pose and a Sensitivity Analysis. *Comput. Vis. Image Underst.* **1994**, *60*, 313–342. [[CrossRef](#)]
34. Faugeras, O. *Three-Dimensional Computer Vision*; MIT Press: Cambridge, MA, USA, 1993.
35. Fiore, P.D. Efficient Linear Solution of Exterior Orientation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 140–148. [[CrossRef](#)]
36. Quan, L.; Lan, Z. Linear N-Point Camera Pose Determination. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 774–780. [[CrossRef](#)]
37. Kim, J.; Kim, T. Comparison of Computer Vision and Photogrammetric Approaches for Epipolar Resampling of Image Sequence. *Sensors* **2016**, *16*, 412. [[CrossRef](#)]
38. Aicardi, I.; Chiabrando, F.; Lingua, A.; Noardo, F. Recent trends in cultural heritage 3D survey: The photogrammetric computer vision approach. *J. Cult. Herit.* **2018**, *32*, 257–266. [[CrossRef](#)]
39. Hartley, R.I.; Mundy, J.L. Relationship between photogrammetry and computer vision. In Proceedings of the SPIE 1944, Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision, Orlando, FL, USA, 24 September 1993.
40. Huang, T.S.; Faugeras, O.D. Some properties of the E matrix in two-view motion estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **1989**, *11*, 1310–1312. [[CrossRef](#)]
41. Nister, D. An efficient solution to the five points relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 756–770. [[CrossRef](#)] [[PubMed](#)]
42. Faugeras, O.; Lustman, F. Motion and structure from motion in a piecewise planar environment. *Int. J. Pattern Recognit. Artif. Intell.* **1988**, *2*, 485–508. [[CrossRef](#)]
43. Zhang, Z.; Hanson, A.R. 3D Reconstruction based on homography mapping. In Proceedings of the Advanced Research Project Agency (ARPA) Image Understanding Workshop, Palm Springs, CA, USA, 12–15 February 1996; pp. 1007–1012.
44. Ma, Y.; Soatto, S.; Kosecka, J.; Sastry, S.S. *An Invitation to 3D Vision: From Images to Geometric Models*; Springer: Berlin, Germany, 2004.
45. Huang, T.S.; Netravali, A.N. Motion and structure from feature correspondences: A review. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *82*, 252–268. [[CrossRef](#)]
46. Martin A. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395.
47. Vezhnevets, V.; Velizhev, A.; Yakubenko, A.; Chetverikov, N. GML C++ Camera Calibration Toolbox. 2013. Available online: <http://graphics.cs.msu.ru/en/node/909> (accessed on 2 February 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).