PERGAMON

Pattern Recognition 36 (2003) 2927–2944

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

A review on egomotion by means of differential epipolar geometry applied to the movement of a mobile robot

Xavier Armangué[a,1], Helder Araújo[b], Joaquim Salvi[a,*]

[a]*Institut de Informàtica i Aplicacions, Universitat de Girona, Av. Lluís Santaló, s/n, E-17071 Girona, Spain*
[b]*Instituto de Sistemas e Robótica, Universidade de Coimbra, Polo II, 3030 Coimbra, Portugal*

Received 24 April 2002; received in revised form 19 February 2003; accepted 28 April 2003

Abstract

The estimation of camera egomotion is an old problem in computer vision. Since the 1980s, many approaches based on both the discrete and the differential epipolar constraint have been proposed. The discrete case is used mainly in self-calibrated stereoscopic systems, whereas the differential case deals with a single moving camera. This article surveys several methods for 3D motion estimation unifying the mathematics convention which are then adapted to the common case of a mobile robot moving on a plane. Experimental results are given on synthetic data covering more than 0.5 million estimations. These surveyed algorithms have been programmed and are available on the Internet.
© 2003 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.

Keywords: Differential epipolar constraint; Egomotion; Optical flow; Computer vision

1. Introduction

Camera calibration is the first step toward computational computer vision. The use of precisely calibrated cameras makes the measurement of distances in a metric world from their projections on the image plane possible. The camera model is a mathematical description of the geometric relationship between the 3D geometric entities and their 2D projections on the image plane consisting of a set of internal camera parameters which describes the internal geometry and optics of the camera, and a set of extrinsic parameters which describes the position and orientation of the camera in the scene. Perspective cameras can be represented by several models depending on the desired level of accuracy.

Given a 3D point $\mathbf{p}$ in metric coordinates with respect to the world coordinate system $\{W\}$, its projection $\mathbf{m}$ in pixels with respect to the image coordinate system $\{I\}$ can be computed through some linear (sometimes non-linear) equations.[2] This set of equations encapsulates several transformations which are broken down into four steps (see Fig. 1).

1. First the coordinates of point $\mathbf{p}$ in the world coordinate system are transformed into the camera coordinate system by using an Euclidean transformation.
2. Next, it is necessary to carry out the projection of point $\mathbf{p}$ onto the image plane by using a projective transformation, obtaining point $\mathbf{q}$.
3. The third step models lens distortion causing a disparity of the real projection on the image plane. Then, point $\mathbf{q}$ is transformed into the real projection $\mathbf{m}$.
4. Finally, the last step consists of transforming the $\mathbf{m}$ point from the metric coordinate system of the camera into the image coordinate system of the computer in pixels.

Small variations in the definition of the geometric transformations used imply the use of different camera models,

[*] Corresponding author. Tel.: +34-972-41-8474; fax: +34-972-41-8098.
E-mail addresses: armangue@eia.udg.es (X. Armangué), helder@isr.uc.pt (H. Araújo), qsalvi@eia.udg.es (J. Salvi).
[1] This research is partly supported by Spanish project CICYT-TAP99-0443-C05-01.

[2] See Appendix A for the notation.

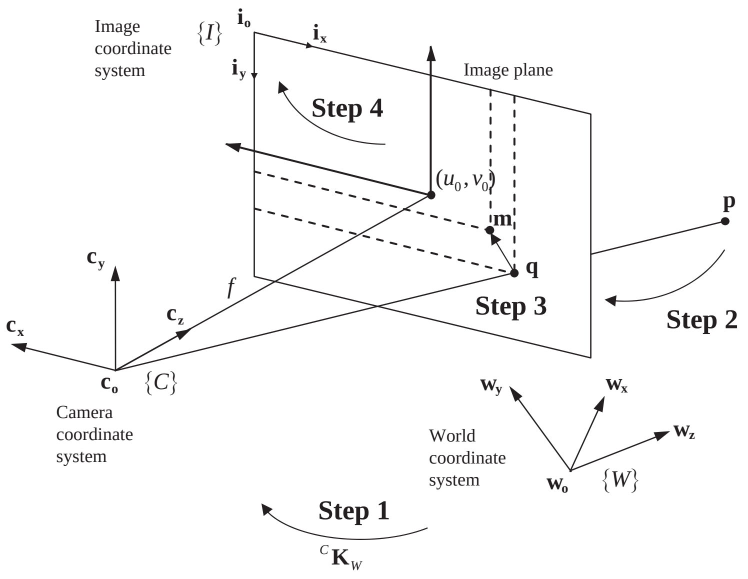


Fig. 1. The geometric relation between a 3D point and its 2D projection.

resulting in different calibration techniques. For instance, the technique proposed by Hall [1] in 1982 is based on an implicit linear camera calibration by computing the 3×4 transformation matrix which relates 3D object points with their 2D image projections. The latter work of Faugeras [2], proposed in 1986, was based on extracting the physical parameters of the camera from this sort of transformation technique. Some years later, work proposed by Faugeras was adaptated to include radial lens distortion [3]. Two other interesting contributions are the widely used method proposed by Tsai [4], is based on a *two-step technique* modelling only radial lens distortion, and the complete model of Weng [5], proposed in 1992, which includes three different types of lens distortion. Research efforts are still being carried out on obtaining new camera models to improve both accuracy in computing the optical ray and in extracting the camera parameters which best model reality. For additional details concerning camera calibration methods, please check the recent calibration survey [6].

When we get into the binocular case (that is two views from a stereoscopic system or two different views from a single moving camera) another interesting relationship is defined in the so-called epipolar geometry. This information is contained in the fundamental matrix which includes the intrinsic parameters of both cameras and the position and orientation of one camera with respect to the other. The fundamental matrix can be used to simplify the matching process between the viewpoints and to get the camera parameters in active systems where optical and geometrical characteristics might change dynamically depending on the imaging scene. In this case, the camera parameters can be extracted by using Kruppa equations [7]. Moreover, the epipolar geometry can be considered from both a continuous and a discrete point of view.

Probably the most well-known viewpoint is the discrete epipolar constraint formulated by Longuet–Higgins [8], Huang [9] and Faugeras [10]. In this case the relative 3D displacement between both views is recovered by the epipolar constraint from a set of correspondences in both image planes. Then, given an object point $\mathbf{p}$ with respect to one of the two camera coordinate system and its 2D projections $\mathbf{q}$ and $\mathbf{q}'$ on both image planes (in metric coordinates), the 3 points define a plane Π which intersects both image planes at the epipolar lines $\mathbf{l}_{\mathbf{q}'}$ and $\mathbf{l}'_{\mathbf{q}}$, respectively, as shown in Fig. 2. Note that the same plane Π can be computed using both focal points $\mathbf{c_o}$ and $\mathbf{c}'_{\mathbf{o}}$ and a single 2D projection, which is the principle used to reduce the correspondence problem to a single search along the epipolar line. Moreover, the intersection of all the epipolar lines defines an epipole on both image planes, which can also be extracted by intersecting the line defined by both focal points $\mathbf{c_o}$ and $\mathbf{c}'_{\mathbf{o}}$ on both image planes. All the epipolar geometry is contained in the so-called Fundamental matrix [8] as shown in Eq. (1)

$$\mathbf{m}^{\mathrm{T}}\mathbf{F}\mathbf{m}' = 0, \tag{1}$$

where the fundamental matrix depends on the intrinsic parameters of both cameras and the rigid transformation between them

$$\mathbf{F} = \mathbf{A}^{-\mathrm{T}}\mathbf{R}^{\mathrm{T}}\hat{\mathbf{t}}\mathbf{A}'^{-1}. \tag{2}$$

When the intrinsic camera parameters are known, it is possible to simplify Eqs. (1) and (2), obtaining,

$$\mathbf{q}^{\mathrm{T}}\mathbf{E}\mathbf{q}' = 0, \tag{3}$$

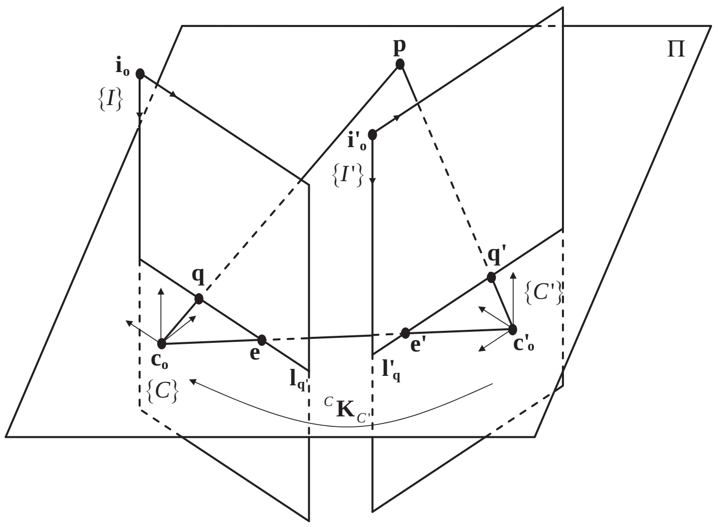


Fig. 2. Discrete epipolar case.

where $\mathbf{q} = \mathbf{A}^{-1}\mathbf{m}$, $\mathbf{E} = \mathbf{R}^{\mathrm{T}}\hat{\mathbf{t}}$, $\mathbf{q}' = \mathbf{A}'^{-1}\mathbf{m}'$. The matrix $\mathbf{E}$ is called essential [9].

Many papers describe different methods to estimate the fundamental matrix [11–14].

The differential case is the infinitesimal version of the discrete case, in which both views are always given from a single moving camera. If the velocity of the camera is low enough and the frame rate is very high, the relative displacement between two consecutive images becomes very small. The 2D displacement of image points can then be obtained from an image sequence using the optical flow. In this case, the 3D camera motion is described by a rigid motion using a rotation matrix and a translation vector, as in (Fig. 3)

$$\mathbf{p}(t) = \mathbf{R}(t)\mathbf{p}(0) + \mathbf{t}(t), \tag{4}$$

where differentiating

$$\dot{\mathbf{p}}(t) = \dot{\mathbf{R}}(t)\mathbf{p}(0) + \dot{\mathbf{t}}(t). \tag{5}$$

Then, replacing the parameter $\mathbf{p}(0)$ to $\mathbf{R}^{-1}(t)[\mathbf{p}(t) - \mathbf{t}(t)]$ in Eq. (5), the following equation is obtained:

$$\dot{\mathbf{p}}(t) = \dot{\mathbf{R}}(t)\mathbf{R}^{-1}(t)\mathbf{p}(t) + \dot{\mathbf{t}}(t) - \dot{\mathbf{R}}(t)\mathbf{R}^{-1}(t)\mathbf{t}(t), \tag{6}$$

which leads to the following differential epipolar constraint:

$$\mathbf{q}^{\mathrm{T}}\hat{\upsilon}\dot{\mathbf{q}} + \mathbf{q}^{\mathrm{T}}\hat{\omega}\hat{\upsilon}\mathbf{q} = 0. \tag{7}$$

$\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)^{\mathrm{T}}$ is the angular velocity of the camera and $\boldsymbol{\upsilon} = (\upsilon_1, \upsilon_2, \upsilon_3)^{\mathrm{T}}$ is the linear velocity of the camera. By projecting $\mathbf{p}$ and $\dot{\mathbf{p}}$ in the image plane, $\mathbf{q}$ in camera coordinates and its corresponding optical flow $\dot{\mathbf{q}}$ are obtained.

For a complete demonstration, the reader is directed to Haralick's book, Chapter 15 [15], where the movement of a rigid body related to a camera is explained. In our case, the demonstration is used to describe the movement of a camera related to a static object, in which only the sign of the obtained velocities differs from the previous one. Nevertheless, Eq. (7) can also be demonstrated in different ways as explained in Viéville [16] and Brooks [17]. Also, another equivalent form of Eq. (7) is shown in Eq. (8). In this case, since matrix $\mathbf{s}$ is symmetric, the number of unknowns is reduced to six

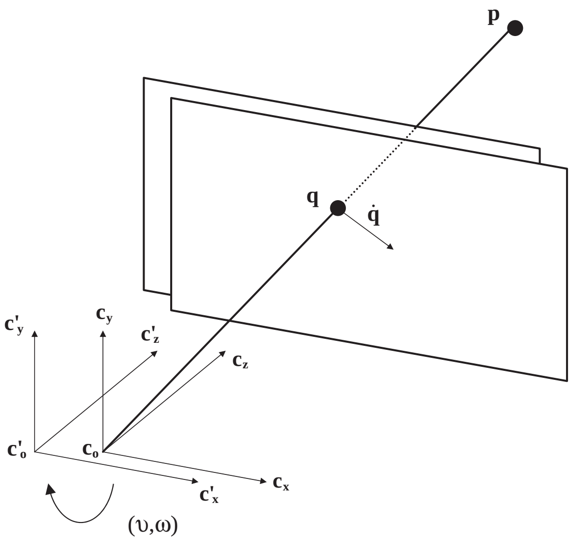


Fig. 3. Differential epipolar case.

$$\mathbf{q}^{\mathrm{T}}\hat{\upsilon}\dot{\mathbf{q}} + \mathbf{q}^{\mathrm{T}}\mathbf{S}\,\mathbf{q} = 0, \tag{8}$$

where

$$\mathbf{S} = \tfrac{1}{2}\,(\hat{\omega}\hat{\upsilon} + \hat{\upsilon}\hat{\omega}). \tag{9}$$

Table 1
Motion recovery methods

Discrete case	Differential case
Linear techniques	
Longuet–Higgins [8]	Zhuang, Huang, Ahuja, Haralick [21,22][a]
Tsai and Huang [23]	Heeger and Jepson [24–26]
Toscani and Faugeras [27]	Kanatani [28,29][a]
Tomasi and Kanade [30]	Tomasi and Shi [31,32]
	Brooks, Chojnacki, Hengel and Baumela [33][a]
	Seven points
	Least squares
	Iteratively reweighted least squares
	Modified iteratively reweighted least squares
	Least median squares
	Ma, Košecká and Sastry [18,34][a]
	Baumela, Agapito, Bustos and Reid [35][a]
Nonlinear techniques	
Horn [36]	Prazdny [37,38]
Weng, Ahuja and Huang [39]	Bruss and Horn [40]
Taylor and Kriegman [41]	Zhang and Tomasi [42]
Soatto and Brockett [43]	
Ma, Košecká and Sastry [44]	

[a]These methods are based on the Differential Epipolar Constraint.

The existence of two forms indicates that a redundancy exists in Eq. (7) (for a demonstration see Viéville [16], Brooks [17] and Ma [18]). Several books describe the optical flow such as Trucco and Verri [19], and the article published by Barron et al. [20] gives a state-of-the-art in optical flow estimation.

When comparing the discrete and differential methods, the discrete epipolar equation incorporates a single matrix, whereas the differential epipolar equation incorporates two matrices. These matrices encode information about the linear and angular velocities of the camera [15].

Approaches to motion estimation can be classified into discrete and differential methods depending on whether they use a set of point correspondences or optical flow. Another possible classification takes into account the estimation techniques used for motion recovery (linear or non-linear techniques). In Table 1, the algorithms are summarized and classified in terms of their nature (discrete and differential case), and estimation method (linear and non-linear technique).

This article analyzes several different algorithms for camera motion estimation based on differential image motion. The surveyed methods have been compared with them and experimental results are given. Moreover, this article analyzes the adaptation of general methods used in free 3D movement to planar motion which corresponds to the common case of a robot moving on a plane with the aim of studying how much accuracy improves by constraining the camera movement. Hence, this article focuses on linear techniques, as the motion has to be recovered in real-time.

This article is structured as follows. Section 2 describes up to 12 algorithms for 3D motion estimation based on optical flow. Section 3 focuses on the estimation of planar motion by constraining the free movement explained in the previous section. Then, Section 4 deals with the experimental results obtained. The article ends with conclusions.

2. Overview of 3D motion estimation

In this section, we detail some methods used for the recovery of every 6-DOF[3] motion parameter from optical flow, providing insights into the complexity of the problem. The surveyed methods have been classified considering whether they are based on the Differential Epipolar Constraint or not.

2.1. Methods based on the Differential Epipolar Constraint

The methods based on the Differential Epipolar Constraint deal with the minimization of the following criteria:

$$\min_{\upsilon,\mathbf{S}} \sum_{i=1}^{n} (\mathbf{q}_i^{\mathrm{T}} \hat{\upsilon} \dot{\mathbf{q}}_i + \mathbf{q}_i^{\mathrm{T}} \mathbf{S}\, \mathbf{q}_i)^2. \tag{10}$$

Rewriting Eq. (10) in matrix form

$$\min_{\boldsymbol{\theta}} \|\mathbf{U}\,\boldsymbol{\theta}\|^2, \tag{11}$$

[3] Degrees of Freedom.

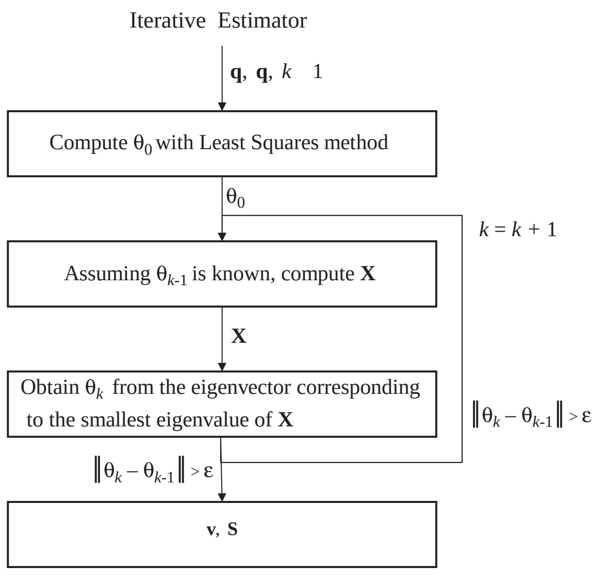

Iterative Estimator
q, q, k 1
Compute θ0 with Least Squares method
θ0
k = k + 1
Assuming θk-1 is known, compute X
X
Obtain θk from the eigenvector corresponding to the smallest eigenvalue of X
‖θk – θk-1‖ > ε
‖θk – θk-1‖ > ε
v, S

Fig. 4. Iterative estimator procedure.

where

$$\boldsymbol{\theta} = (\upsilon_1, \upsilon_2, \upsilon_3, s_{11}, s_{12}, s_{13}, s_{22}, s_{23}, s_{33})^{\mathrm{T}}, \tag{12}$$

$$\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n)^{\mathrm{T}}, \tag{13}$$

in which

$$\mathbf{u}_i = (\dot{q}_{i_2}q_{i_3} - \dot{q}_{i_3}q_{i_2}, \dot{q}_{i_3}q_{i_1} - \dot{q}_{i_1}q_{i_3}, \dot{q}_{i_1}q_{i_2} - \dot{q}_{i_2}q_{i_1}, q_{i_1}^2, 2q_{i_1}q_{i_2}, 2q_{i_1}q_{i_3}, q_{i_2}^2, 2q_{i_2}q_{i_3}, q_{i_3}^2)^{\mathrm{T}}. \tag{14}$$

Hereafter, 7 different methods based on such minimization criteria are consecutively explained in the following paragraphs.

Seven points estimator: This method was proposed by Brooks [33] but not entirely, since a similar method to estimate the fundamental matrix has been used by other authors [11,33].

Using seven points, through singular value decomposition ($\mathbf{U} = \mathbf{V_1 D V_2^T}$), vectors $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are obtained from the two last columns of $\mathbf{V_2}$. The solution is a linear combination of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, $\boldsymbol{\theta} = \alpha\boldsymbol{\theta}_1 + (1-\alpha)\boldsymbol{\theta}_2$ which corresponds to $\boldsymbol{\upsilon}$ and $\mathbf{S}$. Substituting $\boldsymbol{\theta}$ into $\boldsymbol{\upsilon}^{\mathrm{T}}\mathbf{S}\,\boldsymbol{\upsilon} = 0$ (for details see Ref. [17]) a cubic equation on α is obtained. This equation could have one or three solutions for $\boldsymbol{\upsilon}$ and $\mathbf{S}$.

The main advantage of this method is that $\boldsymbol{\upsilon}$ and $\mathbf{S}$ can be estimated using only seven points, but this can also be a drawback when some of the points are corrupted by noise. Moreover, the 7-points method cannot be applied in the presence of redundancy. Hence, it can not be applied using n points where $n > 7$.

Least-squares estimator using eigen analysis: When there are n points, where $n \geqslant 8$, Eq. (10) goes into redundancy, in which $\boldsymbol{\theta} = 0$ is the trivial solution. Then some constraints have to be established in order to avoid the null solution. A general constraint is to fix $\|\boldsymbol{\theta}\| = 1$. Then, Eq. (11) is rewritten as

$$\min_{\|\boldsymbol{\theta}\|=1} \boldsymbol{\theta}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\mathbf{U}\boldsymbol{\theta}. \tag{15}$$

Introducing the Lagrange multiplier λ,

$$\mathscr{L}(\boldsymbol{\theta}) = \boldsymbol{\theta}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\mathbf{U}\boldsymbol{\theta} - \lambda(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\theta} - 1). \tag{16}$$

Eq. (15) is equivalent to minimize

$$\min_{\|\boldsymbol{\theta}\|=1} \mathbf{U}^{\mathrm{T}}\mathbf{U}\boldsymbol{\theta} - \lambda\boldsymbol{\theta}. \tag{17}$$

Thus, the solution $\boldsymbol{\theta}$ must be the eigenvector of the 9×9 matrix $\mathbf{X} = \mathbf{U}^{\mathrm{T}}\mathbf{U}$ corresponding to the smallest eigenvalue λ_0 (more details in Ref. [19] Appendix A.6).

Iteratively reweighted least-squares estimator: This method is an evolution of the previous Seven-Point Estimator proposed by Brooks [33]. In this case, $\boldsymbol{\theta}$ is computed by using an iterative approximation method where all points are reweighted in each iteration. The equation to minimize is

$$\min_{\upsilon,\mathbf{S}} \sum_{i=1}^{n} \left(\frac{|\mathbf{q}_i^{\mathrm{T}}\hat{\boldsymbol{\upsilon}}\dot{\mathbf{q}}_i + \mathbf{q}_i^{\mathrm{T}}\mathbf{S}\,\mathbf{q}_i|^2}{\sqrt{\|2\mathbf{S}\,\mathbf{q}_i + \hat{\boldsymbol{\upsilon}}\dot{\mathbf{q}}_i\|^2 + \|\hat{\boldsymbol{\upsilon}}\mathbf{q}_i\|^2}} \right)^2, \tag{18}$$

which can be rewritten in matrix form obtaining

$$\min_{\|\boldsymbol{\theta}\|=1} \boldsymbol{\theta}^{\mathrm{T}}\mathbf{w}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\mathbf{U}\boldsymbol{\theta}, \tag{19}$$

$$\min_{\|\boldsymbol{\theta}\|=1} \mathbf{w}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\mathbf{U}\boldsymbol{\theta} - \lambda\boldsymbol{\theta}, \tag{20}$$

where $\mathbf{w} = (w_1, \ldots, w_n)^{\mathrm{T}}$ and $w_i = (\|2\mathbf{S}\,\mathbf{q}_i + \hat{\boldsymbol{\upsilon}}\dot{\mathbf{q}}_i\|^2 + \|\hat{\boldsymbol{\upsilon}}\mathbf{q}_i\|^2)^{-1}$. Employing Lagrange multipliers, as in the previous method, the solution $\boldsymbol{\theta}$ is the eigenvector corresponding to the smallest eigenvalue of $\mathbf{X} = \mathbf{w}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\mathbf{U}$. Then, the proposed iterative method is shown in Fig. 4.

Modified iteratively reweighted least-squares estimator: This technique, surveyed by Brooks [33], is similar to the precedent method but is based on a different minimization of Eq. (18). Starting from the reweighted equation w_i, we can express it in matrix form as shown in Eq. (21).

$$w_i^{-1} = \|2\mathbf{S}\,\mathbf{q}_i + \hat{\boldsymbol{\upsilon}}\dot{\mathbf{q}}_i\|^2 + \|\hat{\boldsymbol{\upsilon}}\mathbf{q}_i\|^2 = \boldsymbol{\theta}^{\mathrm{T}}\mathbf{N}_i\boldsymbol{\theta}, \tag{21}$$

in which

$$\mathbf{N}_i = 4\sum_{\alpha=1}^{3}\boldsymbol{\rho}_\alpha^{\mathrm{T}}\mathbf{q}_i\mathbf{q}_i^{\mathrm{T}}\boldsymbol{\rho}_\alpha + 4\sum_{\alpha=1}^{3}\boldsymbol{\rho}_\alpha^{\mathrm{T}}\mathbf{q}_i\dot{\mathbf{q}}_i^{\mathrm{T}}\boldsymbol{\sigma}_\alpha - \sum_{\alpha=1}^{3}\boldsymbol{\sigma}_\alpha^{\mathrm{T}}\dot{\mathbf{q}}_i\dot{\mathbf{q}}_i^{\mathrm{T}}\boldsymbol{\sigma}_\alpha$$
$$-\sum_{\alpha=1}^{3}\boldsymbol{\sigma}_\alpha^{\mathrm{T}}\mathbf{q}_i\mathbf{q}_i^{\mathrm{T}}\boldsymbol{\sigma}_\alpha, \tag{22}$$

where $\boldsymbol{\rho}_i$ and $\boldsymbol{\sigma}_i$ are 3×9 matrices explained in Ref. [33].

After that, Eq. (19) is reorganized as

$$\min_{\|\boldsymbol{\theta}\|=1} 2\mathbf{X}\boldsymbol{\theta}, \tag{23}$$

where

$$\mathbf{X} = \sum_{i=1}^{n}\frac{\mathbf{M}_i}{\boldsymbol{\theta}^{\mathrm{T}}\mathbf{N}_i\boldsymbol{\theta}} - \sum_{i=1}^{n}\frac{\boldsymbol{\theta}^{\mathrm{T}}\mathbf{M}_i\boldsymbol{\theta}}{(\boldsymbol{\theta}^{\mathrm{T}}\mathbf{N}_i\boldsymbol{\theta})^2}\mathbf{N}_i, \tag{24}$$

$$\mathbf{M}_i = \mathbf{u}_i\mathbf{u}_i^{\mathrm{T}}. \tag{25}$$

By using eigen analysis, a solution of $\boldsymbol{\theta}$ is obtained. The procedure proposed is shown in Fig. 4.

Least median squares estimator: The previous methods assume that image points can only present a gaussian noise in its localization in the image plane. LMedS is considered a robust estimator as the method does not use all data because it assumes that a set of points could present a matching error with its correspondent. The points with an erroneous matching are called outliers and the rest are called inliers. LMedS is able to recognize the outliers and keep them out of the computation. LMedS is considered a statistical method which was first proposed by Rousseeuw [45] but later used in several other applications, such as Fundamental matrix estimation [11].

First, this technique is based on selecting a number of sets made up of seven random points which are used to compute approximations of $\boldsymbol{\theta}$ by using the 7-points method. It is important to assure that sets are made up of points evenly spread throughout the image. The number of sets considered depends on the estimated outliers ratio and the probability that at least one set is free of outliers. This probability is given by $P = 1-(1-(1-\varepsilon)^s)^p$, where ε is the maximum outlier ratio, s is the number of elements in each set sample (seven in our case) and p is the number of set samples. Arranging the terms, we obtain

$$p = \left\lceil \frac{\log(1-P)}{\log(1-(1-\varepsilon)^s)} \right\rceil. \tag{26}$$

The LMedS method calculates the median of algebraic residual for each $\boldsymbol{\theta}$ using all data, where the chosen $\boldsymbol{\theta}$ has to minimize this median. This $\boldsymbol{\theta}$ is used to identify the outliers. Finally, when the outliers are removed, a non-robust method using all the remaining points is used to estimate the best solution.

The complete algorithm is:

1. Choose p sets of seven points evenly spread throughout the image.
2. For each set, obtain an estimation of $\boldsymbol{\theta}$ using the 7-Point method. We obtain $\boldsymbol{\upsilon}_j$ and $\mathbf{S}_j$, where $j = 1,\ldots,p$.
3. For each estimation, compute the algebraic residual and determine the median as

$$m_j = \operatorname*{med}_{i=1,\ldots,n}\left(\frac{|\mathbf{q}_i^{\mathrm{T}}\hat{\boldsymbol{\upsilon}}_j\dot{\mathbf{q}}_i + \mathbf{q}_i^{\mathrm{T}}\mathbf{S}_j\mathbf{q}_i|^2}{\sqrt{\|2\mathbf{S}_j\,\mathbf{q}_i + \hat{\boldsymbol{\upsilon}}_j\dot{\mathbf{q}}_i\|^2 + \|\hat{\boldsymbol{\upsilon}}_j\mathbf{q}_i\|^2}}\right)^2. \tag{27}$$

4. From every estimation m_j, take the m_k in which the median is the minimum: $m_k = \min_{j=1,\ldots,p} m_j$.
5. Compute the robust standard deviation $\hat{\sigma} = 1.4826(1 + (5/(n-7))\sqrt{m_k})$.
6. Then, $\mathbf{q}_i$ is considered an outlier only if $(|\mathbf{q}_i^{\mathrm{T}}\hat{\boldsymbol{\upsilon}}_k\dot{\mathbf{q}}_i + \mathbf{q}_i^{\mathrm{T}}\mathbf{S}_k\mathbf{q}_i|^2/\sqrt{\|2\mathbf{S}_k\mathbf{q}_i + \hat{\boldsymbol{\upsilon}}_k\dot{\mathbf{q}}_i\|^2 + \|\hat{\boldsymbol{\upsilon}}_k\mathbf{q}_i\|^2})^2 > (2.5\hat{\sigma})^2$.
7. Finally, recompute $\boldsymbol{\upsilon}$ and $\mathbf{S}$ considering only the inliers by using one of the previous methods.

Ma, Košecká and Sastry estimator: The estimator proposed by Ma et al. [18,34] estimates not only the linear velocity $\boldsymbol{\upsilon}$ and the symmetric matrix $\mathbf{S}$ like the previous methods, but also the angular velocity $\boldsymbol{\omega}$ which is computed from $\mathbf{S}$.

The computation of the 3D velocity $(\boldsymbol{\upsilon},\boldsymbol{\omega})$ is divided into four steps:

1. Estimate $\boldsymbol{\upsilon}_0$ and $\mathbf{S}_0$. Obtain $\boldsymbol{\upsilon}_0$ and $\mathbf{S}_0$, minimizing the error, function using least-squares estimator using eigen analysis.

$$\min_{\boldsymbol{\theta}_0}\|\mathbf{U}\boldsymbol{\theta}_0\|^2. \tag{28}$$

2. Recover the special symmetric matrix. The symmetric matrix $\mathbf{S}$ obtained in the previous step probably does not have the form $\mathbf{S} = 1/2(\hat{\boldsymbol{\omega}}\hat{\boldsymbol{\upsilon}} + \hat{\boldsymbol{\upsilon}}\hat{\boldsymbol{\omega}})$ called *special symmetric matrix* by the authors. Therefore, diagonalize the symmetric matrix $\mathbf{s}_0$ using eigenvalue decomposition:

$$\mathbf{S}_0 = \mathbf{V}_1\,\mathrm{diag}\{\lambda_1,\lambda_2,\lambda_3\}\mathbf{V}_1^{\mathrm{T}} \tag{29}$$

with $\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3$. Project the symmetric matrix onto the special symmetric matrix.

$$\mathbf{S} = \mathbf{V}_1\,\mathrm{diag}\{\sigma_1,\sigma_2,\sigma_3\}\mathbf{V}_1^{\mathrm{T}}, \tag{30}$$

where

$$\sigma_1 = \frac{2\lambda_1+\lambda_2-\lambda_3}{3},\quad \sigma_2 = \frac{\lambda_1+2\lambda_2+\lambda_3}{3}$$
$$\text{and}\quad \sigma_3 = \frac{2\lambda_3+\lambda_2-\lambda_1}{3}. \tag{31}$$

3. Recover the linear and angular velocities from the special symmetric matrix, obtaining four solutions:

$$\begin{cases} \hat{\omega}_{1,2} = \mathbf{U}_2\mathbf{R_Z}(\pm\frac{\pi}{2})\,\mathrm{diag}\{\lambda,\lambda,0\}\mathbf{U}_2^{\mathrm{T}}, \\ \quad \hat{\upsilon}_{1,2} = \mathbf{V}_2\mathbf{R_Z}(\pm\frac{\pi}{2})\,\mathrm{diag}\{1,1,0\}\mathbf{V}_2^{\mathrm{T}}, \\ \hat{\omega}_{3,4} = \mathbf{V}_2\mathbf{R_Z}(\pm\frac{\pi}{2})\,\mathrm{diag}\{\lambda,\lambda,0\}\mathbf{V}_2^{\mathrm{T}}, \\ \quad \hat{\upsilon}_{3,4} = \mathbf{U}_2\mathbf{R_Z}(\pm\frac{\pi}{2})\,\mathrm{diag}\{1,1,0\}\mathbf{U}_2^{\mathrm{T}}, \end{cases} \tag{32}$$

where $\mathbf{U}_2 = -\mathbf{V}_2\mathbf{R_Y}(\psi)$, $\mathbf{V}_2 = \mathbf{V}_1\mathbf{R}_\mathbf{Y}^{\mathrm{T}}(\psi/2 - \pi/2)$, $\lambda = \sigma_1 - \sigma_3 \geqslant 0$ and $\psi = \arccos(-\sigma_2/\lambda) \in [0,\pi]$.
4. Recover the linear velocity. It is necessary to choose one velocity from the four available. Choose the one which accomplishes

$$\boldsymbol{\upsilon}_k^{\mathrm{T}}\boldsymbol{\upsilon}_0 = \max_{i=1,\dots,4} \boldsymbol{\upsilon}_i^{\mathrm{T}}\boldsymbol{\upsilon}_0. \tag{33}$$

Then the 3D velocity estimated is $\boldsymbol{\upsilon} = \boldsymbol{\upsilon}_0$ and $\boldsymbol{\omega} = \boldsymbol{\omega}_k$.

Baumela, Agapito, Bustos and Reid estimator: This motion estimator was proposed by Baumela et al. [35] and is based on using information from the uncertainty of the optical flow. This algorithm is based on the assumption that the optical flow estimation produce exact values for q and noisy estimations for $\dot{\mathbf{q}}$.

By starting from Eq. (8) and knowing that $\mathbf{q} = (q_1, q_2, 1)^{\mathrm{T}}$ and $\dot{\mathbf{q}} = (\dot{q}_1, \dot{q}_2, 0)^{\mathrm{T}}$, we obtain

$$a\dot{q}_1 + b\dot{q}_2 + c = 0, \tag{34}$$

where $a = \upsilon_3 q_2 - \upsilon_2$, $b = \upsilon_1 - \upsilon_3 q_1$ and $c = s_{11}q_1^2 + 2s_{12}q_1q_2 + 2s_{13}q_1 + s_{22}q_2^2 + 2s_{23}q_2 + s_{33}$. Then the algorithm is based on minimizing the distance between Eq. (34) and the optical flow for all the points

$$\min_{\upsilon,\mathbf{S}} \sum_{i=1}^{n} \frac{(a_i\dot{q}_{i_1} + b_i\dot{q}_{i_2} + c_i)^2}{a_i^2 + b_i^2}. \tag{35}$$

The uncertainty of each flow measurement can be expressed by the covariance matrix $\boldsymbol{\Sigma}_{\dot{\mathbf{q}}_i}$, consequently Eq. (35) becomes

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \frac{\boldsymbol{\theta}^{\mathrm{T}}\mathbf{f}_i\mathbf{f}_i^{\mathrm{T}}\boldsymbol{\theta}}{\boldsymbol{\upsilon}^{\mathrm{T}}\mathbf{H}_i\boldsymbol{\Sigma}_{\dot{\mathbf{q}}_i}\mathbf{H}_i^{\mathrm{T}}\boldsymbol{\upsilon}}, \tag{36}$$

where

$$\mathbf{f}_i = (\dot{q}_{i_2}, -\dot{q}_{i_1}, \dot{q}_{i_1}q_{i_2} - \dot{q}_{i_2}q_{i_1}, q_{i_1}^2, 2q_{i_1}q_{i_2}, 2q_{i_1}, q_{i_2}^2, 2q_{i_2}, 1)^{\mathrm{T}}, \tag{37}$$

$$\mathbf{H}_i^{\mathrm{T}} = \begin{pmatrix} 1 & 0 & -q_{i_2} \\ 0 & -1 & q_{i_1} \end{pmatrix}. \tag{38}$$

It is necessary to impose the constraint $\boldsymbol{\upsilon}^{\mathrm{T}}\mathbf{S}\boldsymbol{\upsilon} = 0$ in the minimization of Eq. (36). The possibility proposed by the authors of the method is to substitute s_{33} to the explicit value and employ an iterative minimization method. In this case, Eq. (36) is rewritten in the following way:

$$\min_{\boldsymbol{\delta}} \sum_{i=1}^{n} \frac{\boldsymbol{\delta}^{\mathrm{T}}\mathbf{g}_i\mathbf{g}_i^{\mathrm{T}}\boldsymbol{\delta}}{\boldsymbol{\upsilon}^{\mathrm{T}}\mathbf{H}_i\boldsymbol{\Sigma}_{\dot{\mathbf{q}}_i}\mathbf{H}_i^{\mathrm{T}}\boldsymbol{\upsilon}}, \tag{39}$$

where

$$\mathbf{g}_i = \left(\dot{q}_{i_2}, -\dot{q}_{i_1}, \dot{q}_{i_1}q_{i_2} - \dot{q}_{i_2}q_{i_1}, q_{i_1}^2 - \frac{\upsilon_1^2}{\upsilon_3^2}, 2\left(q_{i_1}q_{i_2} - \frac{\upsilon_1\upsilon_2}{\upsilon_3^2}\right), \times 2\left(q_{i_1} - \frac{\upsilon_1}{\upsilon_3}\right), q_{i_2}^2 - \frac{\upsilon_2^2}{\upsilon_3^2}, 2\left(q_{i_2} - \frac{\upsilon_2}{\upsilon_3}\right)\right)^{\mathrm{T}}, \tag{40}$$

$$\boldsymbol{\delta} = (\upsilon_1, \upsilon_2, \upsilon_3, s_{11}, s_{12}, s_{13}, s_{22}, s_{23})^{\mathrm{T}} \tag{41}$$

and

$$s_6 = -\frac{s_{11}\upsilon_1^2 + 2s_{12}\upsilon_1\upsilon_2 + s_{22}\upsilon_2^2}{\upsilon_3^2} - \frac{2s_{13}\upsilon_1 + 2s_{23}\upsilon_2}{\upsilon_3}. \tag{42}$$

2.2. Methods directly based on the optical flow

Hereafter, 5 different methods which are based directly on the optical flow instead of using the Differential Epipolar constraint are explained. Such methods have been included in the article with the aim of providing experimental results covering the whole field of differential egomotion estimation.

Bruss and Horn estimator: Bruss and Horn [40] proposed a method not based on the differential epipolar principle as are the previously described methods. Instead, their method is based on the following equation:

$$\dot{\mathbf{q}} = \begin{pmatrix} 1 & 0 & -q_1 \\ 0 & 1 & -q_2 \end{pmatrix} \left(\frac{\boldsymbol{\upsilon}}{Z(\mathbf{q})} + \boldsymbol{\omega} \times \mathbf{q}\right). \tag{43}$$

This equation relates the camera velocity with respect to the static scene in which $\mathbf{q}$ is the image point, $\dot{\mathbf{q}}$ is the point velocity, $Z(\mathbf{q})$ is the point depth related to the camera coordinate system and, finally, $\boldsymbol{\upsilon}$ and $\boldsymbol{\omega}$ are the linear and angular camera velocities, respectively.

Then a bilinear constraint can be imposed on the linear $\boldsymbol{\upsilon}$ and angular $\boldsymbol{\omega}$ velocities of every pixel in order to remove point depth $Z(\mathbf{q})$ by means of a few algebraic transformations in Eq. (43) described in the following equation:

$$\boldsymbol{\upsilon}^{\mathrm{T}}(\mathbf{q} \times \dot{\mathbf{q}}) + (\boldsymbol{\upsilon} \times \mathbf{q})^{\mathrm{T}}(\mathbf{q} \times \boldsymbol{\omega}) = 0. \tag{44}$$

However, the following stages are required. First, an initial guess of the translation which can be estimated by using least-squares technique is needed. Second, a non-linear minimization of Eq. (44) is applied to every image pixel with the constraint $\|\boldsymbol{\upsilon}\| = 1$ with the aim of obtaining $\boldsymbol{\upsilon}$. Finally, the rotation velocity can be extracted linearly from Eq. (44).

Prazdny estimator: The method proposed by Prazdny [38] was one of the first to estimate camera movement from optical flow, so it is not based on the epipolar constraint either. This method differs from the previous one because it estimates the rotating velocity first instead of computing the linear velocity. Then a triplet of image points can be obtained from Eq. (43) and by using some algebraic transformations the following equation is given:

$$\mathbf{n}_3(\mathbf{n}_1 \times \mathbf{n}_2) = 0, \tag{45}$$

where $\mathbf{n}_i = (\omega \times \mathbf{q}_i + \dot{\mathbf{q}}_i) \times \mathbf{q}_i = 0$. Once the rotating velocity w is computed from Eq. (45), the linear velocity is given by Eq. (44).

Hegger and Jepson estimator: Hegger and Jepson proposed the so-called linear subspace method [24,25]. Given the optical flow of a set of n image points, the following relationship can be formulated:

$$\boldsymbol{\tau}_i = \sum_{k=1}^{n} \mathbf{c}_{ik}(\dot{\mathbf{q}}_k \times \mathbf{q}_k), \tag{46}$$

where vector $\boldsymbol{\tau}_i$ is orthogonal to $\boldsymbol{\upsilon}$. Moreover, $\mathbf{c}_i = (\mathbf{c}_{i1}, \ldots, \mathbf{c}_{in})$ has to be chosen so that it is orthogonal to both quadratic polynomiums q_{k_1} and q_{k_2} with the aim of removing the rotating velocity from the images. Note that given n points, $n-6$ vectors $\boldsymbol{\tau}_i$ are generated. Then υ corresponds to the eigenvector associated to the smallest eigenvalue of $\sum \boldsymbol{\tau}_i \boldsymbol{\tau}_i^{\mathrm{T}}$.

Tomasi and Shi estimator: The method proposed by Tomasi and Shi [31,32] estimates v from image deformations. Their method estimates translation from image deformations, defined as the change $\dot{\alpha}$ in the angular distance $\alpha = \arccos(\mathbf{q}_i\, \mathbf{q}_j)$ between two image points due to camera movement. Image deformations do not depend on the camera rotating movement so the following bilinear equation can be extracted. Note that this equation is only a function of $\boldsymbol{\upsilon}$ and both depth points $Z(\mathbf{q}_i)$ and $Z(\mathbf{q}_j)$,

$$\dot{\alpha} = \sin\alpha (Z(\mathbf{q}_j), Z(\mathbf{q}_i), 0)(\mathbf{q}_i, \mathbf{q}_j, \mathbf{w}_{ij})^{-\mathrm{T}} \boldsymbol{\upsilon}, \tag{47}$$

where $\mathbf{w}_{ij} = (\mathbf{q}_i \times \mathbf{q}_j)/\|\mathbf{q}_i \times \mathbf{q}_j\|$. Eq. (47) is then minimized by using the variable projection method [46] from a given set of correspondences with the aim of obtaining $\boldsymbol{\upsilon}$ forcing $\|\boldsymbol{\upsilon}\| = 1$. This minimization leads to estimates of three parameters of the linear velocity $\boldsymbol{\upsilon}$ and the n depth parameters of every point. Hence, computing time depends considerably on the number of point correspondences.

Kanatani estimator: The last of the surveyed methods considered in this article is that proposed by Kanatani [28,29] in 1993. One of the first methods to estimate camera movement based on the epipolar geometry was described by Zhuang et al. [22] in 1998. Later, in 1993, Kanatani reformulated this method. He described the image sphere representation and solved the differential epipolar equation written in terms of the essential parameters and twisted optical flow.

The first approach proposed by Kanatani was statistically biased. In order to remove the bias, he proposed an algorithm called renormalization, also in 1993. This second method automatically adjusts the bias and removes the image noise.

3. Adaptation to a mobile robots

The aim of this work is to estimate the motion of a mobile robot. Due to the fact that the permitted movements of a robot are limited, it is possible to establish some modifications in the differential epipolar equation by applying new constraints. With these modifications, the number of potential solutions is reduced so the obtained results improve considerably.

Our robot (see Fig. 5) is constrained to only two independent movements: a translation along $\mathbf{r_x}$ axis and a rotation around $\mathbf{r_z}$ axis.

$${}^{R}\boldsymbol{\upsilon}_{\boldsymbol{r}} = (\upsilon_{r_1}, 0, 0)^{\mathrm{T}}, \qquad {}^{R}\boldsymbol{\omega}_{\boldsymbol{r}} = (0, 0, \omega_{r_3})^{\mathrm{T}}. \tag{48}$$

This limitation implies that the camera placed on and above the robot cannot move freely, so the camera velocity is the same as the robot velocity ($\boldsymbol{\upsilon}_r = \boldsymbol{\upsilon}_c$ and $\boldsymbol{\omega}_r = \boldsymbol{\omega}_c$). The motion with respect to the camera coordinate system depends on its position. The camera is placed in the vertical of the robot at height h and the $\mathbf{c_x}$ axis is parallel to $\mathbf{r}_y$ and perpendicular to $\mathbf{r_x}$. There is a known angle α between $\mathbf{r_x}$ and $\mathbf{c_z}$ axis (see Fig. 5). With this configuration the matrices which relate camera and robot coordinate systems are:

$$\begin{aligned}
&{}^{R}\mathbf{R}_C = \mathbf{R_Z}\left(-\frac{\pi}{2}\right) \mathbf{R_X}\left(-\frac{\pi}{2} - \alpha\right), \qquad {}^{R}\mathbf{t}_C = (0, 0, h)^{\mathrm{T}}, \\
&{}^{C}\mathbf{R}_R = \mathbf{R_X}\left(\frac{\pi}{2} + \alpha\right) \mathbf{R_Z}\left(\frac{\pi}{2}\right), \\
&\quad {}^{C}\mathbf{t}_R = (0, -h\cos\alpha, h\sin\alpha)^{\mathrm{T}}.
\end{aligned} \tag{49}$$

By transforming the velocities to camera coordinate system [47]

$${}^{C}\boldsymbol{\upsilon}_{\boldsymbol{r}} = {}^{C}\mathbf{R}_R\, {}^{R}\boldsymbol{\upsilon}_{\boldsymbol{r}} - {}^{C}\mathbf{R}_R\, {}^{R}\boldsymbol{\omega}_{\boldsymbol{r}} \times {}^{C}\mathbf{t}_R, \tag{50}$$

$${}^{C}\boldsymbol{\omega}_{\boldsymbol{r}} = {}^{C}\mathbf{R}_R\, {}^{R}\boldsymbol{\omega}_{\boldsymbol{r}} \tag{51}$$

we obtain

$${}^{C}\boldsymbol{\upsilon}_{\boldsymbol{r}} = (0, \upsilon_{r_1}\sin\alpha, \upsilon_{r_1}\cos\alpha)^{\mathrm{T}}, \tag{52}$$

$${}^{C}\boldsymbol{\omega}_{\boldsymbol{r}} = (0, \omega_{r_3}\cos\alpha, -\omega_{r_3}\sin\alpha)^{\mathrm{T}}. \tag{53}$$

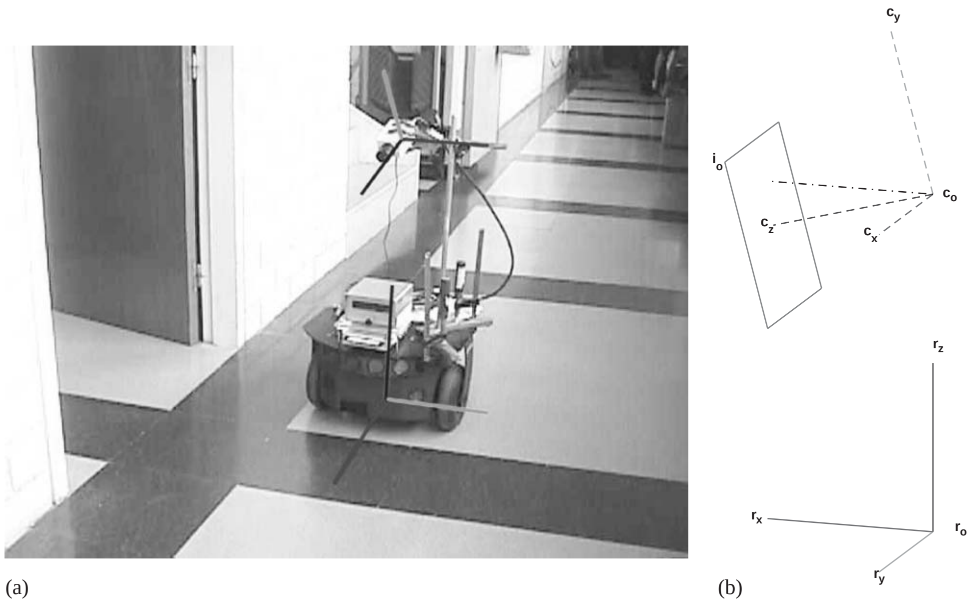


Fig. 5. (a) Robot for test the algorithms; (b) robot and camera coordinate systems.

The camera motion is independent of h and depends only on three unknowns: υ_{r_1}, w_{r_3} and the angle α. With this information it is possible to simplify the symmetric matrix $\mathbf{S}_r$ as

$$\mathbf{S}_r = \frac{1}{2}(\hat{\boldsymbol{\omega}}_r\hat{\boldsymbol{\upsilon}}_r + \hat{\boldsymbol{\upsilon}}_r\hat{\boldsymbol{\omega}}_r) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \omega_{r_3}\upsilon_{r_1}\sin\alpha\cos\alpha & \frac{1}{2}\omega_{r_3}\upsilon_{r_1}(\cos^2\alpha - \sin^2\alpha) \\ 0 & \frac{1}{2}\omega_{r_3}\upsilon_{r_1}(\cos^2\alpha - \sin^2\alpha) & -\omega_{r_3}\upsilon_{r_1}\sin\alpha\cos\alpha \end{pmatrix}, \tag{54}$$

where $s_{11} = s_{12} = s_{13} = 0$, $s_{22} = \frac{1}{2}\omega_{r_3}\upsilon_{r_1}\sin(2\alpha)$, $s_{23} = \frac{1}{2}\omega_{r_3}\upsilon_{r_1}\cos(2\alpha)$ and $s_{33} = -s_{22}$.

The following subsections concern the adaptation of the methods described in Section 2.1 to the movement of a mobile robot by means of the differential epipolar constraint.

Least-squares simplified estimator using eigen analysis: This method is a simplified version of the method described in Section 2.1. Starting from the differential epipolar equation

$$\min_{\upsilon_{r_1}, s_{22}, s_{23}} \sum_{i=1}^{n} (\mathbf{q}_i^{\mathrm{T}}\hat{\boldsymbol{\upsilon}}_r\dot{\mathbf{q}}_i + \mathbf{q}_i^{\mathrm{T}}\mathbf{S}_r\mathbf{q}_i)^2 \tag{55}$$

and knowing the camera motion constraint (Eqs. (52)–(54)), it is possible to rewrite Eq. (55) in matrix form as

$$\min_{\boldsymbol{\theta}'} \|\mathbf{U}'\boldsymbol{\theta}'\|^2, \tag{56}$$

where

$$\boldsymbol{\theta}' = (\upsilon_{r_1}, s_{22}, s_{23})^{\mathrm{T}}, \tag{57}$$

$$\mathbf{U}' = (\mathbf{u}'_1, \mathbf{u}'_2, \ldots, \mathbf{u}'_n)^{\mathrm{T}} \tag{58}$$

and

$$\mathbf{u}'_i = ((\dot{q}_{i_1}q_{i_2} - q_{i_1}\dot{q}_{i_2})\cos\alpha + (q_{i_1}\dot{q}_{i_3} - \dot{q}_{i_1}q_{i_3})\sin\alpha, q_{i_2}^2 - q_{i_3}^2, 2q_{i_2}q_{i_3})^{\mathrm{T}}. \tag{59}$$

When there are n points and $n \geqslant 8$, the solution of $\boldsymbol{\theta}'$ is the eigenvector corresponding to the smallest eigenvalue of the square matrix $\mathbf{X}' = \mathbf{U}'^{\mathrm{T}}\mathbf{U}'$.

Iteratively reweighted least-squares simplified estimator: This simplified estimation reduces the number of parameters and the weight function. By rewriting Eq. (18) in matrix form and only using three parameters (υ_{r_1}, s_{22} and s_{23}) we obtain

$$\min_{\|\boldsymbol{\theta}'\|=1} \boldsymbol{\theta}'^{\mathrm{T}}\mathbf{w}'^{\mathrm{T}}\mathbf{U}'^{\mathrm{T}}\mathbf{U}'\boldsymbol{\theta}', \tag{60}$$

$$\min_{\|\boldsymbol{\theta}'\|=1} \mathbf{w}'^{\mathrm{T}}\mathbf{U}'^{\mathrm{T}}\mathbf{U}'\boldsymbol{\theta}' - \lambda\boldsymbol{\theta}', \tag{61}$$

where $\boldsymbol{\theta}'$ is defined in Eq. (57), $\mathbf{U}'$ is defined in Eq. (58), $\mathbf{w}' = (\mathbf{w}'_1, \ldots, \mathbf{w}'_n)^{\mathrm{T}}$ and $\mathbf{w}'_i = (\|2\mathbf{S}_r\mathbf{q}_i + \hat{\boldsymbol{\upsilon}}_r\dot{\mathbf{q}}_i\|^2 + \|\hat{\boldsymbol{\upsilon}}_r\mathbf{q}_i\|^2)^{-1}$. It is necessary to use an iterative algorithm (see Fig. 4) to solve Eq. (60), which is actually the same, already applied in the general method, but uses the simplified equation $\mathbf{X}' = \mathbf{w}'^{\mathrm{T}}\mathbf{U}'^{\mathrm{T}}\mathbf{U}'$.

Modified iteratively reweighted least-squares simplified estimator: This method is the adapted version to a mobile robot of the method used to estimate camera motion proposed by Brooks [33] and described in Section 2.1. When considering Eq. (18), the reduced number of parameters to estimate leads to a simplified equation as follows:

$$\mathbf{w}_{i'}^{-1} = \|2\mathbf{S}_r\mathbf{q}_i + \hat{\boldsymbol{\upsilon}}_r\dot{\mathbf{q}}_i\|^2 + \|\hat{\boldsymbol{\upsilon}}_r\mathbf{q}_i\|^2 = \boldsymbol{\theta}'^{\mathrm{T}}\mathbf{N}_i'\boldsymbol{\theta}', \tag{62}$$

where

$$\mathbf{N}_i' = 4\sum_{\alpha=1}^{3}\boldsymbol{\rho}_{\alpha'}^{\mathrm{T}}\mathbf{q}_i\mathbf{q}_i^{\mathrm{T}}\boldsymbol{\rho}_\alpha' + 4\sum_{\alpha=1}^{3}\boldsymbol{\rho}_{\alpha'}^{\mathrm{T}}\mathbf{q}_i\dot{\mathbf{q}}_i^{\mathrm{T}}\boldsymbol{\sigma}_\alpha' - \sum_{\alpha=1}^{3}\boldsymbol{\sigma}_{\alpha'}^{\mathrm{T}}\dot{\mathbf{q}}_i\dot{\mathbf{q}}_i^{\mathrm{T}}\boldsymbol{\sigma}_\alpha' - \sum_{\alpha=1}^{3}\boldsymbol{\sigma}_{\alpha'}^{\mathrm{T}}\mathbf{q}_i\mathbf{q}_i^{\mathrm{T}}\boldsymbol{\sigma}_\alpha' \tag{63}$$

and

$$\boldsymbol{\rho}_1' = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{\rho}_2' = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\boldsymbol{\rho}_3' = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}, \quad \boldsymbol{\sigma}_1' = \begin{pmatrix} 0 & 0 & 0 \\ -\cos(\alpha) & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$\boldsymbol{\sigma}_2' = \begin{pmatrix} \cos(\alpha) & 0 & 0 \\ 0 & 0 & 0 \\ -\sin(\alpha) & 0 & 0 \end{pmatrix}, \quad \boldsymbol{\sigma}_3' = \begin{pmatrix} 0 & 0 & 0 \\ \sin(\alpha) & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{64}$$

Then Eq. (18) is rewritten in matrix form considering the parameters to estimate the robot movement obtaining

$$\min_{\|\boldsymbol{\theta}'\|=1} 2\mathbf{X}'\boldsymbol{\theta}', \tag{65}$$

where

$$\mathbf{X}' = \sum_{i=1}^{n}\frac{\mathbf{M}_i'}{\boldsymbol{\theta}'^{\mathrm{T}}\mathbf{N}_i'\boldsymbol{\theta}'} - \sum_{i=1}^{n}\frac{\boldsymbol{\theta}'^{\mathrm{T}}\mathbf{M}_i'\theta'}{(\boldsymbol{\theta}'^{\mathrm{T}}\mathbf{N}_i'\boldsymbol{\theta}')^2}\mathbf{N}_i', \tag{66}$$

$$\mathbf{M}_i' = \mathbf{u}_i'\mathbf{u}_{i'}^{\mathrm{T}} \tag{67}$$

using $\mathbf{u}_i'$ which was previously described in Eq. (59).

Finally, $\boldsymbol{\theta}'$ is obtained, minimizing $\mathbf{X}'\boldsymbol{\theta}' = 0$ by using eigen analysis and the algorithm shown in Fig. 5.

Least median squares simplified estimator: The adaptation of LMedS to robot movement estimation forces the following considerations. First, instead of using the 7-points method to measure of the velocity for each group of points, we will use the simplified least-squares method described in this section. In the adaptation to a mobile robot, the movement is constrained to only three unknowns; that is straight forward, translation and rotation. Moreover, the number of sets to generate randomly also changes as it depends on the number of set points. Eq. (26) relates such a relationship.

The second modification of this method takes place in the last step of the algorithm once all the outliers have been removed, and $\boldsymbol{\upsilon}_r$ and $\mathbf{S}_r$ are recalculated using the modified iteratively reweighted least-square simplified estimator instead of the general method.

Ma, Košecká and Sastry simplified estimator: The method proposed by Ma et al. [18,34] is not affected by the reduction of the number of parameters. Moreover, only the first step of the method has to be modified to adapt the algorithm to the case of a robot moving on a plane, that is υ_0 and $\mathbf{s}$. The equation to minimize is the following:

$$\min_{\boldsymbol{\theta}'}\|\mathbf{U}'\boldsymbol{\theta}'\|^2. \tag{68}$$

Eq. (68) computes $\boldsymbol{\upsilon}_{r_0}$ and $\mathbf{S}_r$, these results are used in the further steps of the algorithm already detailed in Section 2.1. Then the steps of the algorithm are the following:

1. Estimate $\boldsymbol{\upsilon}_{r_0}$ and $\mathbf{S}_r$.
2. Recover the special symmetric matrix.
3. Recover velocities form the special symmetric matrix.
4. Recover velocity.

Baumela, Agapito, Bustos and Reid simplified estimator: The adaptation of the general method proposed by Baumela et al. [35] described in Section 2.1 to the simplified case of the robot movement is based on the following modifications. Considering Eq. (8) and using the simplified camera movement described in Eqs. (52) and (54), the following linear equation is obtained:

$$a'\dot{q}_1 + b'\dot{q}_2 + c' = 0, \tag{69}$$

where $a' = \upsilon_{r_1}(q_2\cos\alpha - \sin\alpha)$, $b' = -\upsilon_{r_1}q_1\cos\alpha$ and $c' = s_{22}(q_2^2 - 1) + 2s_{23}q_2$.

This algorithm minimizes the distance between Eq. (69) and the optical flow of every point by using the following equation:

$$\min_{\upsilon_{r_1}, s_{22}, s_{23}}\sum_{i=1}^{n}\frac{(a_i'\dot{q}_{i_1} + b_i'\dot{q}_{i_2} + c_i')^2}{a_{i'}^2 + b_{i'}^2}. \tag{70}$$

Considering that the measures of the optical flow present a given discrepancy, Eq. (70) is rewritten in matrix form including the covariance matrix of the optical flow $\boldsymbol{\Sigma}_{\dot{\mathbf{q}}_i}$, obtaining

$$\min_{\boldsymbol{\theta}'}\sum_{i=1}^{n}\frac{\boldsymbol{\theta}'^{\mathrm{T}}\mathbf{f}_i'\mathbf{f}_i'^{\mathrm{T}}\boldsymbol{\theta}'}{\boldsymbol{\upsilon}_r^{\mathrm{T}}\mathbf{H}_i\boldsymbol{\Sigma}_{\dot{\mathbf{q}}_i}\mathbf{H}_i^{\mathrm{T}}\boldsymbol{\upsilon}_r}, \tag{71}$$

where $\mathbf{f}_i' = ((\dot{q}_{i_1}q_{i_2} - q_{i_1}\dot{q}_{i_2})\cos\alpha - \dot{q}_{i_1}\sin\alpha, q_{i_2}^2 - 1, 2q_{i_2})^{\mathrm{T}}$ and $\mathbf{H}_i^{\mathrm{T}}$ is defined in Eq. (38).

However, the constraint $\boldsymbol{\upsilon}_r^{\mathrm{T}}\mathbf{S}_r\boldsymbol{\upsilon}_r = 0$ has to be forced to Eq. (71) in minimization. Hence, the parameter s_{22} is extracted and substituted in Eq. (71), obtaining

$$\min_{\boldsymbol{\delta}'} \sum_{i=1}^{n} \frac{\boldsymbol{\delta}'^{\mathrm{T}} \mathbf{g}'_i \mathbf{g}'^{\mathrm{T}}_i \boldsymbol{\delta}'}{\boldsymbol{\upsilon}_r^{\mathrm{T}} \mathbf{H}_i \boldsymbol{\Sigma}_{\dot{\mathbf{q}}_i} \mathbf{H}_i^{\mathrm{T}} \boldsymbol{\upsilon}_r}, \tag{72}$$

where

$$\mathbf{g}'_i = ((\dot{q}_{i_1} q_{i_2} - q_{i_1} \dot{q}_{i_2}) \cos\alpha - \dot{q}_{i_1} \sin\alpha, 2q_{i_2} - \tan(2\alpha)(q_{i_2}^2 - 1))^{\mathrm{T}}, \tag{73}$$

$$\boldsymbol{\delta}' = (\upsilon_{r_1}, s_{23})^{\mathrm{T}}, \tag{74}$$

and

$$s_{22} = -s_{23} \tan(2\alpha). \tag{75}$$

4. Experimental results

All the methods surveyed have been programmed and tested under the same conditions of image noise with the aim of giving an exhaustive comparison of most of 6-DOF motion estimation methods. Hence, Section 4.1 compares the twelve surveyed methods of 3D motion estimation and Section 4.2 deals with the six proposed adaptations to a 2-DOF mobile robot movement estimation. Section 4.3 shows results in real image sequences.

4.1. Results on 3D motion estimation

The surveyed methods based on the differential epipolar constraint explained in Sections 2.1 have been programmed in MATLAB®. The others, that is the ones explained in Section 2.2 have been taken from the comparative survey and MATLAB® toolbox given by Tian et al. [48]. The partial use of a previous toolbox permits us to validate the programmed methods and compare the obtained results.

Several tests were done using synthetic data with the goal of comparing the robustness of the methods in the presence of image noise. We have used a methodology similar to the one proposed by Tian et al. [48] and Ma et al. [18]. Moreover, the camera movement is estimated from a cloud of 50 3D points located in front of the camera and distributed throughout the field of view image (we considered a field of view varying between 30° and 90°). Next, the optical flow of every point is computed. Once the optical flow of the 50 points is computed, gaussian noise is added to every velocity component with a standard deviation varying from 0.05 pixels up to 0.5 pixels. With the aim of studying the robustness of every method in any potential camera movement, all the potential camera orientations and translations are considered in ranges of 22.5°. 10 movement estimations are carried out for every camera pose.

Then the optical flow of every point is computed by using Eq. (43), in which $\dot{\mathbf{q}}$ is the velocity of the image point $\mathbf{q} = (q_1, q_2, 1)$ on the image plane; $\boldsymbol{\upsilon}$ and $\boldsymbol{\omega}$ are the camera's linear and angular velocities; and $Z(\mathbf{q})$ is the depth of every pixel $\mathbf{q}$.

In the following experiments the angular velocity has been considered fixed and equal to 0.23°/frame, while the coefficient of linear/angular velocity varies from 1 up to 10. Once the optical flow of the 50 points is computed, a gaussian noise is added to every velocity component varying its standard deviation from 0.05 up to 0.5 pixels. With the aim of studying the robustness of every method in any potential camera movement, all the camera orientations and translations have been considered in ranges of 22.5° and 10° movement estimations are carried out for every camera pose. Hence, given an image field of view of 30°, 60° and 90°, a linear/angular coefficient of 1, 5 and 10 and a gaussian noise of 0.05 up to 0.5 pixels, an amount of 655,360 movement estimations have been computed for each surveyed method. Every estimation has been compared to the real movement where the discrepancy in the linear velocity estimation is the angle between the real movement vector $\boldsymbol{\upsilon}$ and the estimated $\boldsymbol{\upsilon}_{\mathrm{est}}$ which is computed using the following equation:

$$\mathrm{error}_{\mathrm{lineal}} = \cos^{-1}(\boldsymbol{\upsilon} \cdot \boldsymbol{\upsilon}_{\mathrm{est}}). \tag{76}$$

The discrepancy between the rotation matrix of the real angular movement $\mathbf{R}$ with respect to the estimated rotation matrix $\mathbf{R}_{\mathrm{est}}$ obtained from the vector of angular velocities $\boldsymbol{\omega}_{\mathrm{est}}$ is used to compute the angular velocity error. Then, the difference rotation matrix is defined as follows, $\Delta\mathbf{R} = \mathbf{R}^{\mathrm{T}}\mathbf{R}_{\mathrm{est}}$. The matrix $\Delta\mathbf{R}$ is defined by a rotation axis and an angle. The measuring error presented in this angle is computed by using the following equation:

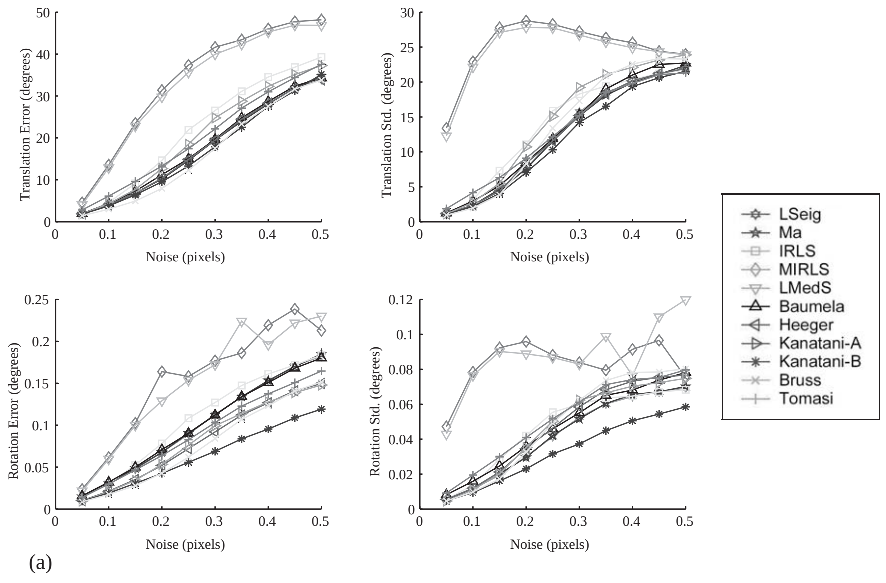
$$\mathrm{error}_{\mathrm{angular}} = \cos^{-1}\left(\frac{\mathrm{Tr}(\Delta\mathbf{R}) - 1}{2}\right), \tag{77}$$

where $\mathrm{Tr}(\Delta\mathbf{R})$ is the trace of the matrix.

Fig. 6a shows the obtained results of every method considering a linear/angular velocity coefficient equal to unity, an image field of view of 90° and a gaussian noise varying from 0.05 up to 0.5 pixels. This is a worst case, which leads us to compare the robustness of every method. The best results are obtained when the linear/angular velocity coefficient is bigger than the unity and the image field of view smaller than 90°.

The 7-points method and the method proposed by Prazdny are not shown in Fig. 6a due to the poor results obtained which condition the illustrative comparison of the figure. The figure shows the errors in the linear velocity angle and its standard deviation and the errors in the angular velocity angle and its standard deviation.

Summarizing, the surveyed methods which present the worst results are the modified iteratively reweighted least squares (MIRLS) and the least median squares (LMedS) due to the lack of convergence in minimization given by MIRLS. Our implementation of LMedS optimizes the solution, once the outliers have been removed, by using MIRLS minimization, showing a poor estimate of the camera movement. The other surveyed methods obtain similar results, especially in the estimation of the translation movement, where an angle error of around 20° is obtained with a gaussian noise



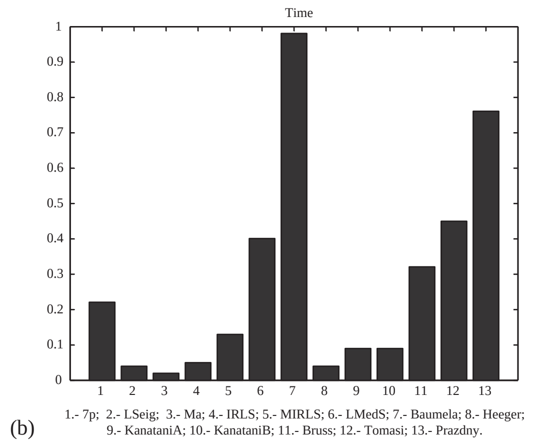


Fig. 6. (a) Estimation results and (b) computing time results of general methods with synthetic dates.

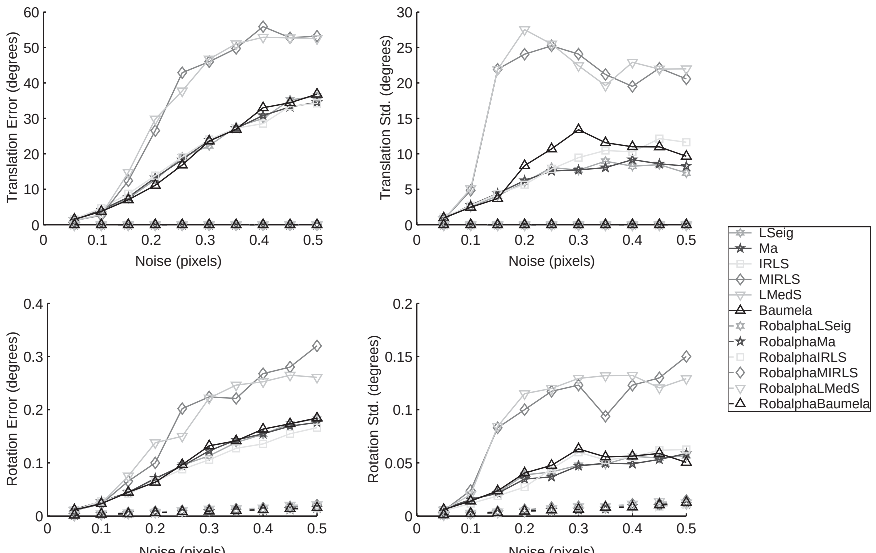
of only 0.3 pixels. Note that we are comparing these methods in the most unfavorable conditions. These conditions can be improved by reducing the image field of view of the camera or increasing the linear/angular velocity coefficient. Nevertheless, the results indicate that in different conditions the results obtained by the surveyed methods are comparatively similar. However, Fig. 6a shows that the method proposed by Kanatani with a previous data normalization obtains by far the best results in angular velocity estimation even with increasing gaussian noise. Finally, comparison of the execution time of every method in MATLAB® running on a Pentium® III Computer at 800 MHz is shown in Fig. 6b. The figure shows that while almost half the methods obtain a solution in less than 0.1 s; eleven, such as 7-points method, Bruss and Horn, and Tomasi and Shi, obtain a movement estimation in less than 0.5 s, and the two others, Prazdny, and Baumela, Agapito, Bustos and Reid spend more than 0.5 s.

4.2. *Results on mobile robot motion estimation*

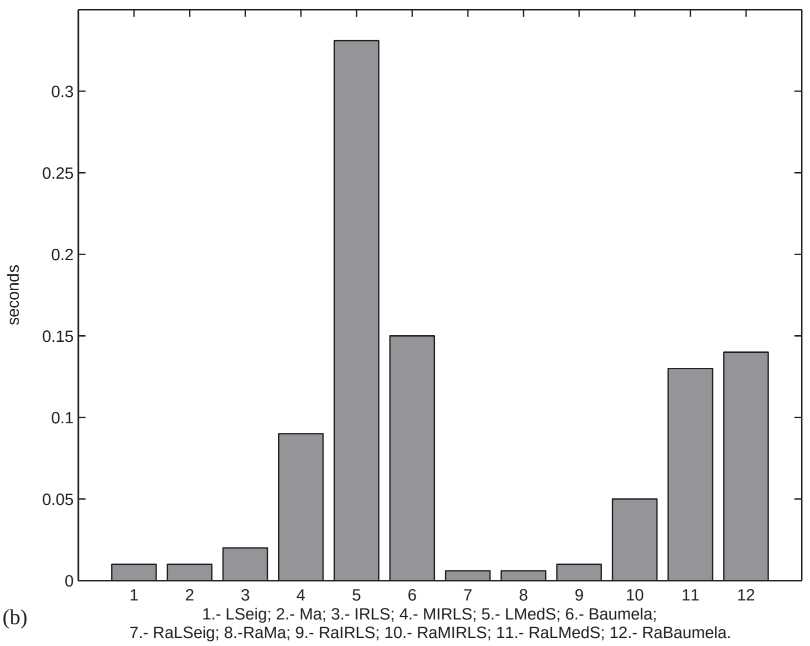
In order to compare the methods adapted to estimate mobile robot movement, tests are based on the same settings used in the previous section but constrained to the common case of a mobile robot. A new parameter α, corresponding to the angle between the optical axis of the camera and the ground plane has been considered. Tests were done for several values of α, that is: $0°$, $15°$, $30°$, $45°$, $60°$, $75°$ and $90°$. It has been observed that the movement estimation presents a slight error at $\alpha = 45°$, which was the worst case.

Fig. 7a shows the results obtained by the movement estimation methods for the case of a mobile robot with a field of view of $90°$, a coefficient linear/angular velocity equal to unity considering the worst case $\alpha = 45°$.

The methods adapted to robot motion do not present an error in the linear velocity estimation because the methods



(a)



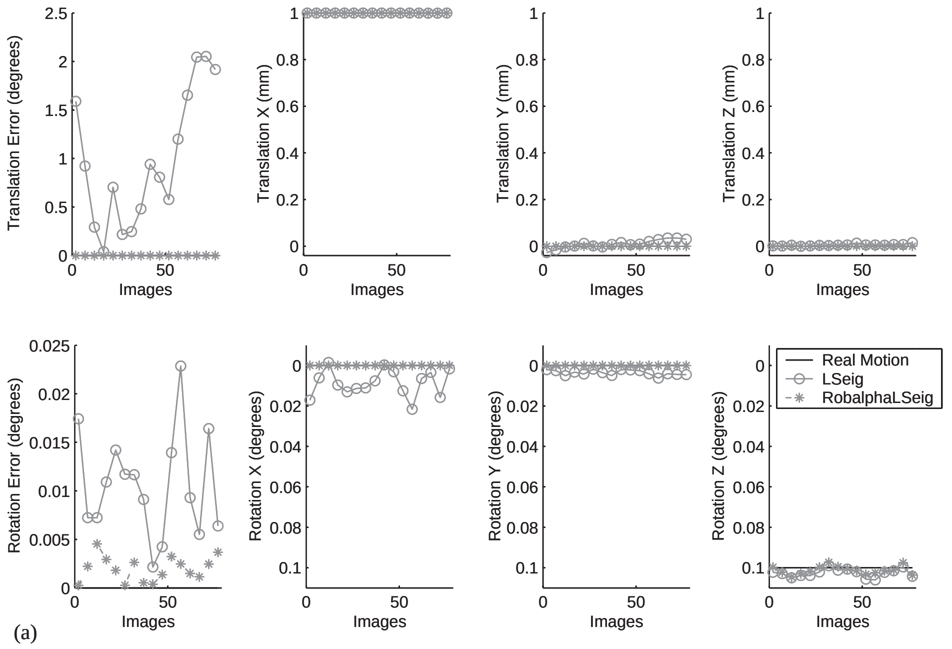
(b)

Fig. 7. (a) Estimation results and (b) computing time results of general and simplified methods with synthetic images.

intrinsically fix its direction. Hence, the error in the translation estimate for every method shown in Fig. 7a is zero. Actually, this fact implies that the error presented in the estimation of the angular velocity decreases considerably. Results on rotation estimation show that the adapted methods are more robust in the presence of image noise than their general versions (i.e. including all the 6-DOF). Fig. 7b shows the computation times obtained by using MATLAB® and a PC Pentium® III at 800 MHz, showing that seven of the 12 methods yield an estimate in 0.05 s or less, permitting their use in real-time applications.

4.3. Results on mobile robot motion with real images

The results obtained with real images are also quite accurate. Fig. 8 compare the results given by LS and its adaptation to the mobile robot (RobalphaLSeig), considering up to 80 test images where the camera has a tilt angle of



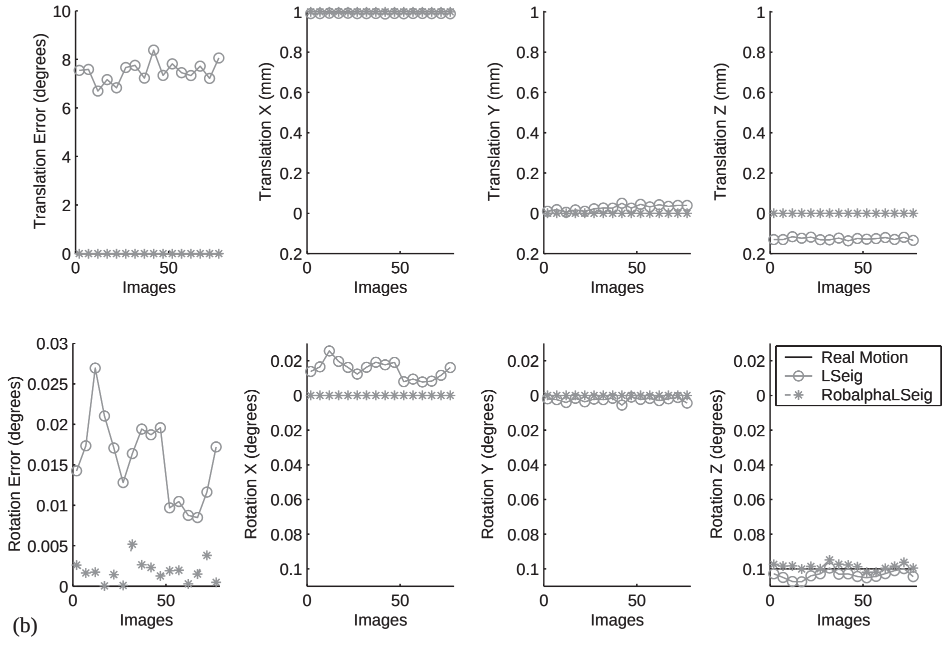


Fig. 8. Example of motion estimation with real images of 6-DOF (LSeig) and its adaptation to 2-DOF (RobalphaLSeig) with (a) $\alpha = 0°$; (b) $\alpha = 10°$; and (c) $\alpha = 20°$.

$0°$, $10°$ and $20°$, respectively, and the robot progresses and rotates with an angle of $-0.1°$ in every two consecutive images. Figure show the accuracy on rotation and translation estimation and the vectors obtained. The error on translation estimation is zero in the adapted method while the general method (6-DOF estimation) gives an error in the Z-axis, since, in that case, it is difficult to distinguish between a camera rotation around Y-axis and a camera translation along X-axis. The error increases for α equal to $45°$ being minimum at $0°$ and $90°$. Finally, the rotation estimation is also more accurate using the adapted RaLS than the general method.

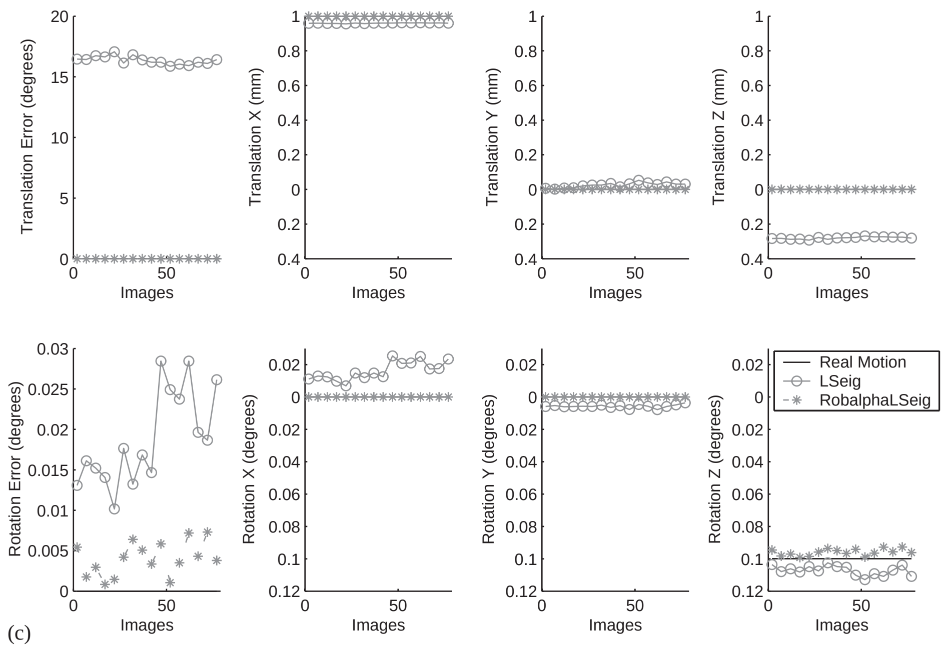


Fig. 8. (*Continued*)

The same example described in the previous paragraph was tested for all the surveyed methods. The results obtained are presented in Fig. 9, which shows that the adapted 2-DOF methods are always more accurate than the general 6-DOF.

5. Conclusions

This article presents an up-to-date classification of the methods and techniques used to estimate the movement of a single camera. A survey of several motion recovering methods is done and experimental results are given with synthetic data considering both gaussian noise and outliers.

The general methods to estimate a 6-DOF movement have been adapted to the common case of a mobile robot moving on a plane obtaining better results and stability even under important noise conditions.

This article is presented to give a better understanding of the relative performances of the 6-DOF camera movement estimators, especially in the common 2-DOF case of a mobile robot. Several methods were described after analyzing their differences with respect to the use of the epipolar geometry in both the discrete case and the differential case.

In summarizing, the 6-DOF movement estimator methods are quite sensitive to noise. Hence, these methods should be adapted constraining the number of DOF with the aim of reducing the error. In this article, the 2-DOF common case of a mobile robot has been considered and results show better movement estimation and stability due to a fixed direction of the translation movement constrained by the structure of the mobile robot.

The contributions of this article are the following:

- Comparison of discrete versus differential epipolar constraint.
- A state-of-the-art motion estimation.
- Proposal of some new methods to compute 2-DOF mobile robot egomotion.
- Experimental results are presented considering synthetic data with both gaussian noise and outliers, and real images obtained by a camera mounted on a mobile robot.

Acknowledgements

We greatly appreciate Dr. Tina Y. Tian, Dr. Carlo Tomasi and Dr. David J. Heeger who implemented the methods explained in Section 2.2 which have been compared with the rest of methods explained in the article, specially Dr. Heeger who gave us insightful information and the source code of such methods.

Appendix A. Mathematics convention

The mathematical convention used in this article is the following:

- Vectors are in boldface and lowercase i.e. $\mathbf{v}$ is a vector; matrices are in boldface and uppercase i.e. $\mathbf{M}$ is a matrix; and scalars are in lowercase i.e. s is a scalar.
- World coordinate system: $\{W\} = \{\mathbf{w_o}, \mathbf{w_x}, \mathbf{w_y}, \mathbf{w_z}\}$.

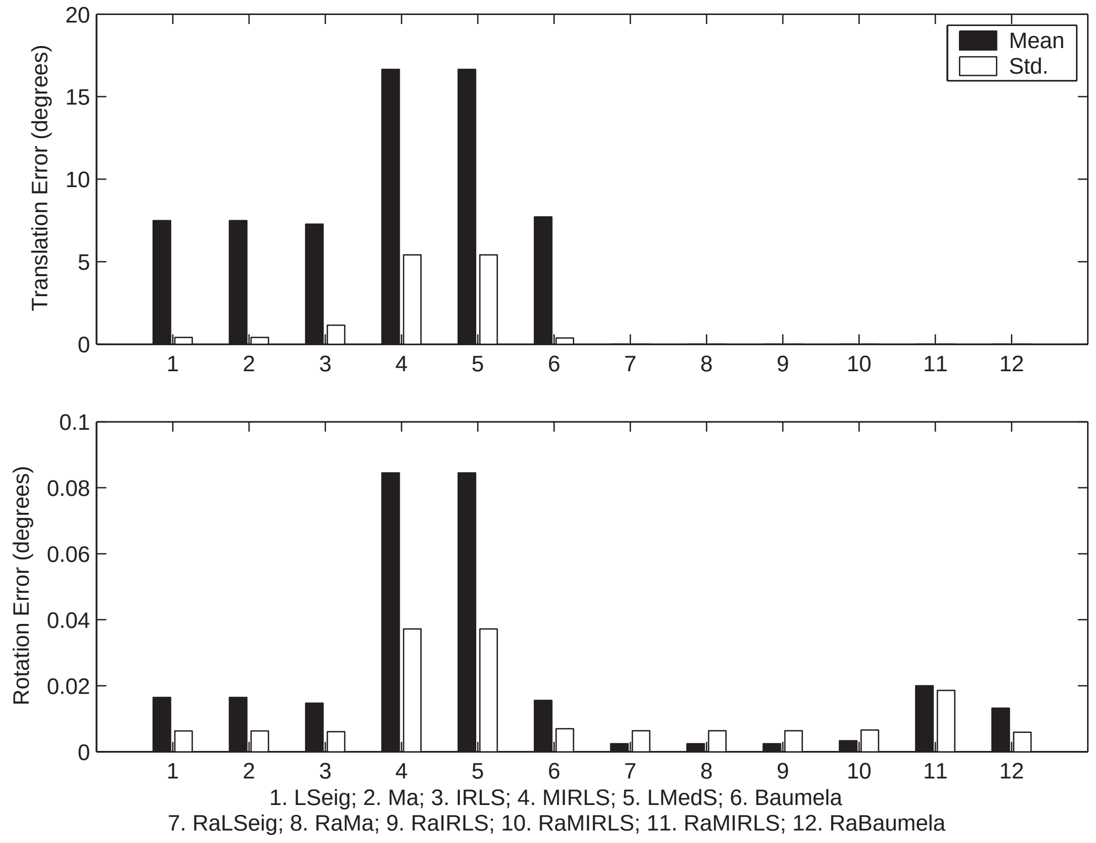


Fig. 9. Results of general (1–6) and adapted (7–12) methods with real images.

- Camera coordinate system: $\{C\} = \{\mathbf{c_o}, \mathbf{c_x}, \mathbf{c_y}, \mathbf{c_z}\}$.
- Robot coordinate system: $\{R\} = \{\mathbf{r_o}, \mathbf{r_x}, \mathbf{r_y}, \mathbf{r_z}\}$.
- Image coordinate system: $\{I\} = \{\mathbf{i_o}, \mathbf{i_x}, \mathbf{i_y}\}$.
- Left uppercase superscript relates the reference coordinate system, i.e. ${}^{J}\mathbf{v}$ relates the vector $\mathbf{v}$ with respect to the coordinate system $\{J\}$.
- A rigid transformation between a two coordinate system: ${}^{J}\mathbf{K}_{H}$ expresses the coordinate system $\{H\}$ with respect to $\{J\}$
- 3D point: $\mathbf{p} = (p_1, p_2, p_3)^{\mathrm{T}} \in \mathbb{R}^3$
- 2D image point: $\mathbf{m} = (m_1, m_2)^{\mathrm{T}} \in \mathbb{Z}^2$
- Translation vector: $\mathbf{t} = (t_1, t_2, t_3)^{\mathrm{T}} \in \mathbb{R}^3$.
- Rotation matrix: $\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \in \mathbb{R}^{3\times 3}$ and orthonormal.
- Linear velocity: $\boldsymbol{\upsilon} = (\upsilon_1, \upsilon_2, \upsilon_3)^{\mathrm{T}} \in \mathbb{R}^3$.
- Angular velocity: $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)^{\mathrm{T}} \in \mathbb{R}^3$.
- Symmetric matrix: $\mathbf{S} = \begin{pmatrix} s_{11} & s_{12} & s_{13} \\ s_{12} & s_{22} & s_{23} \\ s_{13} & s_{23} & s_{33} \end{pmatrix} \in \mathbb{R}^{3\times 3}$ where $\mathbf{S} = \mathbf{S}^{\mathrm{T}}$.
- Skew symmetric matrix associated with $\boldsymbol{\upsilon}$: $\hat{\boldsymbol{\upsilon}} = \begin{pmatrix} 0 & -\upsilon_3 & \upsilon_2 \\ \upsilon_3 & 0 & -\upsilon_1 \\ -\upsilon_2 & \upsilon_1 & 0 \end{pmatrix}$ where $\boldsymbol{\upsilon} \times \boldsymbol{u} = \hat{\boldsymbol{\upsilon}}\boldsymbol{u}$.

References

[1] E.L. Hall, J.B.K. Tio, C.A. McPherson, F.A. Sadjadi, Measuring curved surfaces for robot vision, Comput. J. 15 (12) (1982) 42–54.

[2] O.D. Faugeras, G. Toscani, The calibration problem for stereo, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Los Alamitos, CA, 1986, pp. 15–20.

[3] J. Salvi, J. Batlle, E.M. Mouaddib, A robust-coded pattern projection for dynamic 3D scene measurement, Pattern Recognition Lett. 19 (11) (1998) 1055–1065.

[4] R.Y. Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, IEEE Int. J. Robotics Automat. RA-3 (4) (1987) 323–344.

[5] J. Weng, P. Cohen, M. Herniou, Camera calibration with distortion models and accuracy evaluation, IEEE Trans. Pattern Anal. Mach. Intell. 14 (10) (1992) 965–980.

[6] J. Salvi, X. Armangué, J. Batlle, A comparative review of camera calibrating methods with accuracy evaluation, Pattern Recognition 35 (7) (2002) 1617–1635.

[7] R.I. Hartley, Kruppa's equations derived from the fundamental matrix, Pattern Anal. Mach. Intell. 19 (2) (1997) 133–135.

[8] H.C. Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, Nature 293 (1981) 133–135.

[9] T.S. Huang, O.D. Faugeras, Some properties of the E matrix in two-view motion estimation, IEEE Trans. Pattern Anal. Mach. Intell. 11 (12) (1989) 1310–1312.

[10] O.D. Faugeras, Three-Dimensional Computer Vision, The MIT Press, Cambridge, MA, 1993.

[11] Z. Zhang, Determining the epipolar geometry and its uncertainty: a review, Int. J. Comput. Vision 27 (2) (1998) 161–198.

[12] Q.-T. Luong, O.D. Faugeras, The fundamental matrix: theory, algorithms, and stability analysis, Int. J. Comput. Vision 17 (1) (1996) 43–75.

[13] P.H.S. Torr, D.W. Murray, The development and comparison of robust methods for estimating the fundamental matrix, Int. J. Comput. Vision 24 (3) (1997) 271–300.

[14] X. Armangué, J. Salvi, Overall view regarding fundamental matrix estimation, Image Vision Comput. 21 (2) (2003) 205–220.

[15] R.M. Haralick, L.G. Shapiro, Computer and Robot Vision, Vol. 2, Addison-Wesley Publishing Company, Reading, MA, 1992.

[16] T. Viéville, O.D. Faugeras, The first order expansion of motion equations in the uncalibrated case, Comput. Vision Graphics Image Process.: Image Understanding 64 (1) (1996) 128–146.

[17] M.J. Brooks, W. Chojnacki, L. Baumela, Determining the ego-motion of an uncalibrated camera from instantaneous optical flow, J. Opt. Soc. Am. 10 (1997) 2670–2677.

[18] Y. Ma, J. Košecká, S. Sastry, Linear differential algorithm for motion recovery: a geometric approach, Int. J. Comput. Vision 36 (1) (2000) 71–89.

[19] E. Trucco, A. Verri, Introductory Techniques for 3D Computer Vision, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1998.

[20] J. Barron, D. Fleet, S. Beauchemin, T. Burkitt, Performance of optical flow techniques, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Champaigne, IL, USA, 1992, pp. 236–242.

[21] X. Zhuang, R.M. Haralick, Rigid body motion on optical flow image, in: Proceedings of the First International Conference on Artificial Intelligence Applications, Denver, CO, 1984, pp. 366–375.

[22] X. Zhuang, T.S. Huang, N. Ahuja, R.M. Haralick, A simplified linear optic flow-motion algorithm, Comput. Vision Graphics Image Process. 42 (1988) 334–344.

[23] R.Y. Tsai, T.S. Huang, Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1) (1984) 13–27.

[24] D.J. Hegger, A.D. Jepson, Subspace methods for recovering rigid motion I: algorithm and implementation, Int. J. Comput. Vision 7 (2) (1992) 95–318.

[25] A.D. Jepson, D.J. Heeger, A fast subspace algorithm for recovering rigid motion, in: Proceedings of the IEEE Workshop on Visual Motion, Princeton, NJ, 1991, pp. 124–131.

[26] A.D. Jepson, D.J. Heeger, Linear subspace methods for recovering translation direction, in: L. Harris, M. Jenkin (Eds.), Spatial Vision in Humans and Robots, Cambridge University Press (1993) 39–62.

[27] G. Toscani, O.D. Faugeras, Structure and motion from two noisy perspective images, in: Proceedings of the IEEE Conference on Robotics and Automation, San Francisco, CA, 1986, pp. 221–227.

[28] K. Kanatani, Unbiased estimation and statistical analysis of 3-D rigid form two views, IEEE Trans. Pattern Anal. Mach. Intell. 15 (1) (1993) 37–50.

[29] K. Kanatani, 3-D interpretation of optical flow by renormalization, Int. J. Comput. Vision 11 (3) (1993) 267–282.

[30] C. Tomasi, T. Kanade, Shape and motion from image streams under orthography, Int. J. Comput. Vision 9 (2) (1992) 137–154.

[31] C. Tomasi, J. Shi, Direction of heading from image deformations, in: Proceedings of Computer Vision and Pattern Recognition, New York City, NY, 1993, pp. 422–427.

[32] J. Shi, C. Tomasi, Good features to track, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, 1994.

[33] M.J. Brooks, W. Chojnacki, A. Van Den Hengel, L. Baumela, Robust techniques for the estimation of structure from motion in the uncalibrated case, in: Proceedings of the Fifth European Conference on Computer Vision, Vol. 1, Freiburg, Germany, 1998, pp. 281–295.

[34] Y. Ma, J. Košecká, S. Sastry, Motion recovery form images sequences: Discrete viewpoint vs. differential viewpoint, in: Proceedings of the 15th European Conference on Computer Vision, Freiburg, Germany, 1998.

[35] L. Baumela, L. Agapito, P. Bustos, I. Reid, Motion estimation using the differential epipolar equation, Proceedings of the Fifteenth International Conference on Pattern Recognition, Vol. 3, Barcelona, Spain, 2000, pp. 848–851.

[36] B.K. Horn, Relative orientation, Int. J. Comput. Vision 4 (1990) 59–78.

[37] K. Prazdny, Egomotion and relative depth map from optical flow, Biol. Cybernet. 36 (1980) 87–102.

[38] K. Prazdny, Determining the instantaneous direction of motion from optical flow generated by a curvilinearly moving observer, Comput. Graphics Image Process. 17 (1981) 238–248.

[39] J. Weng, N. Ahuja, T.S. Huang, Matching two prespective views, IEEE Trans. Pattern Anal. Mach. Intell. 14 (1992) 806–825.

[40] A.R. Bruss, B.K. Horn, Passive navigation, Comput. Vision Graphics Image Process. 21 (1983) 3–20.

[41] C.J. Taylor, D.J. Kriegman, Structure and motion from line segments in multiple images, IEEE Trans. Pattern Anal. Mach. Intell. 17 (11) (1995) 1021–1032.

[42] T. Zhang, C. Tomasi, Fast, robust, and consistent camera motion estimation, in: Proceedings of the IEEE Computer Vision and Pattern Recognition, Ft. Collins, CO, Vol. 1, 1999, pp. 164–170.

[43] S. Soatto, R. Brockett, Optimal and suboptimal structure from motion, in: Proceedings of the IEEE International Conference on Computer Vision, Santa Barbara, CA, 1998, pp. 282–288.

[44] Y. Ma, J. Košecká, S. Sastry, Optimal motion from image sequences: A riemannian viewpoint, in: Proceedings of the Conference on Mathematical Theory of Networks and Systems, Padova, Italy, 1998.
[45] P.J. Rousseeuw, A.M. Leroy, Robust Regression and Outlier Detection, Wiley, New York, 1987.
[46] A. Ruhe, P.A. Wedin, Algorithms for separable nonlinear least squares problems, SIAM Rev. 22 (3) (1980) 318–337.
[47] S. Hutchinson, G.D. Hager, P.I. Corke, A tutorial on visual servo control, IEEE Trans. Robotics Automat. 12 (5) (1996) 651–670.
[48] T.Y. Tian, C. Tomasi, D.J. Heeger, Comparison of approaches to egomotion computation, in: Proceedings of Computer Vision and Pattern Recognition, San Francisco, CA, 1996, pp. 315–320.

About the Author—XAVIER ARMANGUÉ received the B.S. degree in Computer Science in the University of Girona in 1999 before joining the Computer Vision and Robotics Group. At present he is involved in the study of stereovision systems for mobile robotics and he is working for his Ph.D. in the Computer Vision and Robotics Group in the University of Girona and in the Institute of Systems and Robotics in the University of Coimbra.

About the Author—HELDER ARAÚJO is currently Associate Professor at the Department of Electrical and Computer Engineering of the University of Coimbra. He is Deputy Director of the Institute for Systems and Robotics, Coimbra. His main research interests are computer vision and mobile robotics. He has been working in vision and robotics for the last 13 years.

About the Author—JOAQUIM SALVI graduated in Computer Science in the Polytechnical University of Catalunya in 1993. He joined the Computer Vision and Robotics Group in the University of Girona, where he received the M.S. degree in Computer Science in July 1996 and the Ph.D in Industrial Engineering in January 1998. He received the best thesis award in Industrial Engineering of the University of Girona. At present, he is an associate professor in the Electronics, Computer Engineering and Automation Department of the University of Girona. His current interest are in the field of computer vision and mobile robotics, focusing on structured light, stereovision and camera calibration.