



Instantaneous 3D motion from image derivatives using the Least Trimmed Square regression

Fadi Dornaika^{a,*}, Angel Sappa^{b,1}

^a Institut Géographique National, 2 Avenue Pasteur, 94165 Saint Mandé, France

^b Computer Vision Center, Edifici O Campus UAB, 08193 Bellaterra, Barcelona, Spain

ARTICLE INFO

Article history:

Received 22 November 2007

Received in revised form 15 October 2008

Available online 24 December 2008

Communicated by H.H.S. Ip

Keywords:

Instantaneous 3D motion

Image derivatives

Robust statistics

Least Trimmed Square regression

ABSTRACT

This paper presents a new technique to the instantaneous 3D motion estimation. The main contributions are as follows. First, we show that the 3D camera or scene velocity can be retrieved from image derivatives only assuming that the scene contains a dominant plane. Second, we propose a new robust algorithm that simultaneously provides the Least Trimmed Square solution and the percentage of inliers—the non-contaminated data. Experiments on both synthetic and real image sequences demonstrated the effectiveness of the developed method. Those experiments show that the new robust approach can outperform classical robust schemes.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Computing object and camera motions from 2D image sequences has been a central problem in computer vision for many years (Hartley and Zisserman, 2000; Jasinschi et al., 2000; Weng et al., 1993; Zucchelli et al., 2002). More especially, computing the 3D velocity of either the camera or the scene is of particular interest to a wide variety of applications in computer vision and robotics such as calibration (Malm and Heyden, 2002), visual servoing, ego-motion estimation, detecting independently moving objects, to mention a few. One of the main tasks in computer vision is the reconstruction of the structure of a scene in a process known as structure from motion (SFM). The classic approach to SFM, which attracted considerable attention in the literature, is based on the extraction and matching of image features throughout the image sequence.

Many algorithms have been proposed for estimating the 3D relative camera motions (discrete case) (Lourakis and Argyros, 2004) and the 3D velocity (differential case) (Baumela et al., 2000; Brooks et al., 1997; Rother and Carlsson, 2002).

In (Jasinschi et al., 2000), the authors describe a method for extracting the camera velocity. This method is a combination of

the eight-point method in structure-from-motion with a statistical technique to automatically select feature points in the image, irrespective of 3D content information. In (Kim et al., 1997), the authors estimate the camera motion parameters from image sequences using a linear motion parameter equation and the Kalman filtering method.

Very few non-correspondence motion estimation algorithms have been proposed (Dellaert et al., 2000; Boughorbel et al., 2003). One can notice that although these ones circumvent the need for establishing direct correspondences, they still need to perform feature extraction in the images. In (Boughorbel et al., 2003), the authors propose a method for estimating the relative motion of a camera from two successive frames. The method relies on a structure saliency measure, and does not require any previous knowledge of point correspondences between the images. They presented two different such metrics. The first metric was simple and based on measuring the scattering of the structure points. The second metric used the tensor voting approach and was more robust. In (Baumela et al., 2000), the authors derived the continuous analogue of the discrete epipolar equation, given a geometric interpretation of it, and a practical algorithm for computing camera's motion parameters from closely spaced views. The input were given by the optical flow field.

While the discrete case requires feature matching and tracking across the images, the differential case requires the computation of the optical flow field (2D velocity field) (Barron et al., 1994; Irani, 1999; Fleet et al., 2000). All these problems are generally ill-conditioned.

* Corresponding author. Tel.: +33 143988429; fax: +33 143988581.

E-mail addresses: fadi.dornaika@ign.fr (F. Dornaika), sappa@cvc.uab.es (A. Sappa).

¹ Tel.: +34 935813036; fax: +34 935811670.

In our work, we assume that the scene is far from the camera or it contains a dominant planar structure. The use of image derivatives has been exploited in (Brody and Fermuller, 2002) to make camera intrinsic calibration. In (Malm and Heyden, 2000), image derivatives have been used to perform hand-eye calibration with constrained camera motions obtained by controlling the motion of the robot hand. The current paper has two main contributions. First, we introduce a novel technique to the unconstrained 3D velocity estimation using image derivatives alone, therefore feature extraction and tracking are not required. Second, we propose a robust method that combines the Least Trimmed Square regression and the Golden Section Search algorithm where the number of inliers is not known *a priori*. Our robust method simultaneously estimates the inlier percentage and the robust solution. On the other hand, existing LTS regression methods (e.g., Rousseeuw and Driessen, 2002) assume that the percentage of inliers is known in advance.

The first contribution concerns the instantaneous 3D motion estimation from image data, which can be useful for many applications in vision and robotics such as extrinsic calibration (Dornaika and Chung, 2008), visual servoing (Horaud et al., 1998), video indexing (Jasinschi et al., 2000), space robot localization (Johnson et al., 2007), and augmented reality (Lourakis and Argyros, 2004). What differentiates our work from existing ones is the use of image derivatives alone and not the optical flow field with a novel robust statistics solution.

The second contribution is within the field of robust linear regression (Maronna et al., 2006).

In our study, we deal with the estimation of the 3D velocity of the camera or the scene from image derivatives where the motion is not constrained. Our proposed approach lends itself nicely to all applications in which the camera motion is not controlled (e.g.; when using a hand-held camera in indoor environments). The paper is organized as follows. Section 2 describes the problem we are focusing on. Section 3 describes the proposed methods. Experimental results on both synthetic and real image sequences are given in Section 4.

2. Problem formulation

Throughout this paper we represent the coordinates of a point in the image plane by small letters (x, y) and the object coordinates in the camera coordinate frame by capital letters (X, Y, Z) . In our work, we use the perspective camera model as our projection model. Thus, the projection is governed by the following equation where the coordinates are expressed in homogeneous form,

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & s & x_c & 0 \\ 0 & rf & y_c & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (1)$$

Here, f denotes the focal length in pixels, r and s the aspect ratio and the skew and (x_c, y_c) the principal point. These are called the intrinsic parameters. In this study, we assume that the camera is calibrated, i.e., the intrinsic parameters are known. For the sake of presentation simplicity, we assume that the image coordinates have been corrected for the principal point and the aspect ratio. This means that the camera equation can be written as in (1) with $r = 1$, and $(x_c, y_c) = (0, 0)$. Also, we assume that the skew is zero ($s = 0$). With these parameters the projection simply becomes

$$x = f \frac{X}{Z} \quad \text{and} \quad y = f \frac{Y}{Z} \quad (2)$$

Let $I(x, y, t)$ be the intensity at pixel (x, y) in the image plane at time t . Let $u(x, y)$ and $v(x, y)$ denote components of the motion field in the x and y directions, respectively. This motion field is caused by the translational and rotational camera velocities $(\mathbf{V}, \boldsymbol{\Omega}) = (V_x, V_y, V_z,$

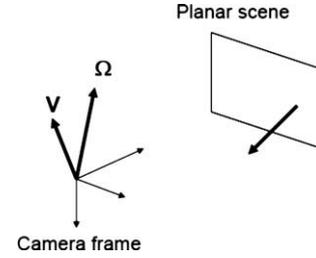


Fig. 1. The goal is to compute the 3D velocity from image derivatives.

$\Omega_x, \Omega_y, \Omega_z$) (see Fig. 1). Using the constraint that the gray-level intensity is locally invariant to the viewing angle and distance we obtain the well-known optical flow constraint equation:

$$I_x u + I_y v + I_t = 0 \quad (3)$$

where $u = \frac{\partial x}{\partial t}$ and $v = \frac{\partial y}{\partial t}$ denote the motion field. $I_x = \frac{\partial I}{\partial x}$ and $I_y = \frac{\partial I}{\partial y}$ denote the components of the spatial image gradient. They can be computed by convolution with derivatives of a 2D Gaussian kernel. The temporal derivative $I_t = \frac{\partial I}{\partial t}$ can be computed by convolution between the derivative of a 1D Gaussian and the image sequence.

We assume that the perspective camera observes a planar scene² described in the camera coordinate system by $Z = \alpha X + \beta Y + \gamma$.

The derivation of the motion field (optical flow field) associated with a planar scene (Eqs. (6)–(8)) has been already established in the literature of SFM (e.g., see Kanatani, 1993). However, to the best of our knowledge, extracting the 3D motion velocity parameters from the spatio-temporal image derivatives has not been addressed (see Section 3). In the following, we sketch out the derivation of the motion field for completeness purposes. Let (u, v) be the 2D velocity (the instantaneous motion) of a 2D point (x, y) , this is given by the temporal derivative of x and y , thus

$$\dot{x} = \frac{f}{Z^2} (\dot{X}Z - X\dot{Z}) \quad (4)$$

$$\dot{y} = \frac{f}{Z^2} (\dot{Y}Z - Y\dot{Z}) \quad (5)$$

The velocity of any 3D point $(X, Y, Z)^T$ is given by

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} + \begin{pmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} \times \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

where \times denotes the cross product. If we insert the above expressions for \dot{X}, \dot{Y} , and \dot{Z} into Eqs. (4) and (5), and use $Z = \alpha X + \beta Y + \gamma$ together with Eq. (2), we get the 8-parameter motion field

$$u(x, y) = \dot{x} = a_1 + a_2 x + a_3 y + a_7 x^2 + a_8 xy \quad (6)$$

$$v(x, y) = \dot{y} = a_4 + a_5 x + a_6 y + a_7 xy + a_8 y^2 \quad (7)$$

where the coefficients are given by:

$$\begin{cases} a_1 = -f \left(\frac{V_x}{\gamma} + \Omega_y \right) \\ a_2 = \left(\frac{V_x}{\gamma} \alpha + \frac{V_z}{\gamma} \right) \\ a_3 = \frac{V_x}{\gamma} \beta + \Omega_z \\ a_4 = -f \left(\frac{V_y}{\gamma} - \Omega_x \right) \\ a_5 = \left(\frac{V_y}{\gamma} \alpha - \Omega_z \right) \\ a_6 = \left(\frac{V_y}{\gamma} \beta + \frac{V_z}{\gamma} \right) \\ a_7 = \frac{-1}{f} \left(\frac{V_z}{\gamma} \alpha + \Omega_y \right) \\ a_8 = \frac{-1}{f} \left(\frac{V_z}{\gamma} \beta - \Omega_x \right) \end{cases} \quad (8)$$

² Since we use robust statistics, the proposed framework can handle the case where the scene contains a dominant planar structure.

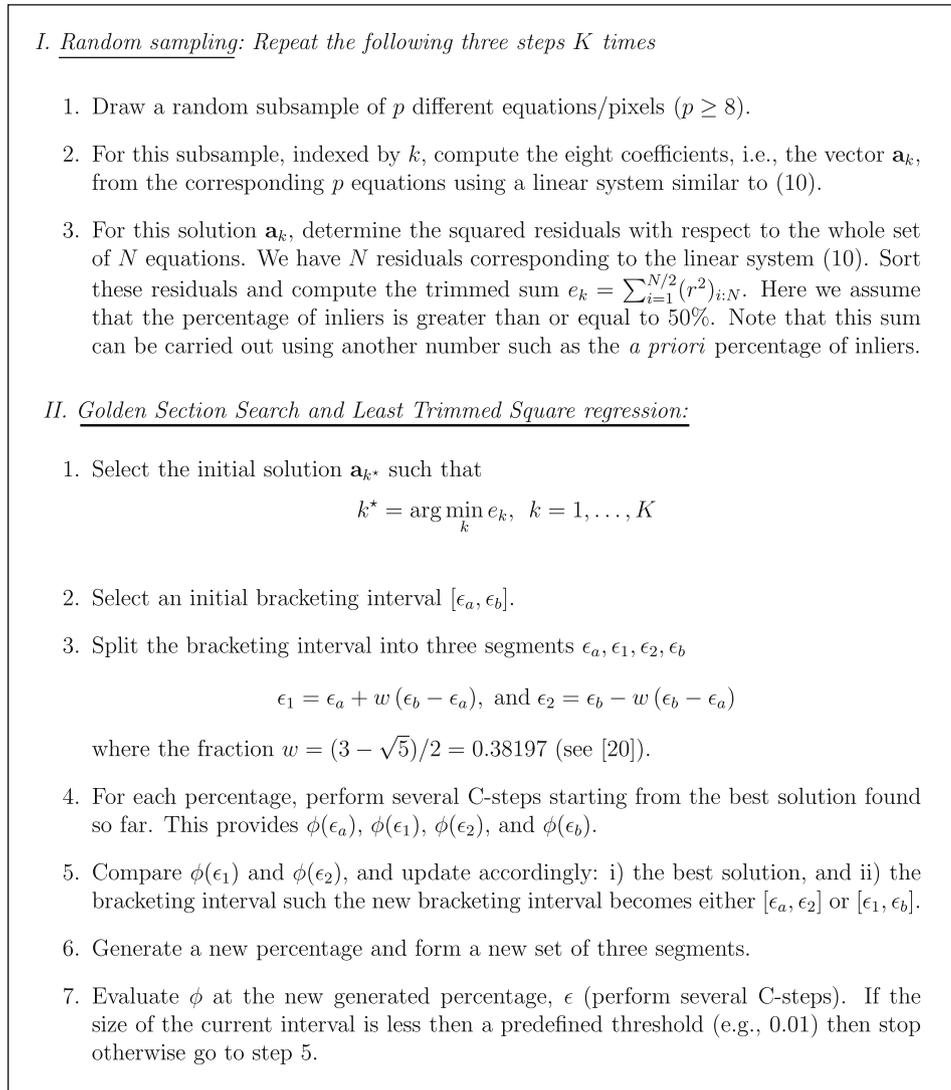


Fig. 3. Estimating the 8 coefficients using the LTS regression and the Golden Section Search algorithm.

Rousseeuw proposed an algorithm that attempts to reach the global minimum. In brief, the algorithm can be summarized as follows.

- Draw random samples where each sample is composed of p equations where $p \ll h < N$.
- For each random sample compute the corresponding solution and then refine it using several C-steps (see above).
- Report the solution that provides the smallest objective function.

3.2. The proposed approach and the eight coefficients

As we have mentioned earlier, the algorithm proposed by Rousseeuw assumes that the size of the subset, h , is known in advance. In our case, however, h is not known. We propose an algorithm that simultaneously provides the LTS solution and the percentage of inliers. Our problem consists in solving the 8-vector \mathbf{a} using the over-constrained linear system (10). When the inlier percentage $\epsilon = \frac{h}{N} \in [0, 1]$ is unknown, we compute it by minimizing the following objective function

$$\phi(\epsilon) = \frac{e(\epsilon)}{\epsilon^\lambda} \quad (11)$$

where λ is a predefined parameter (in all our tests described in the sequel, we used $\lambda = 6$). The above objective function $\phi(\epsilon)$ minimizes

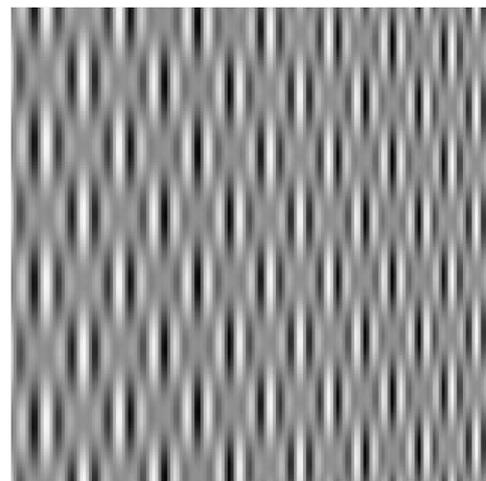


Fig. 4. A computer generated image of a 3D plane that is rotated about 60° about an axis perpendicular to the optical axis.

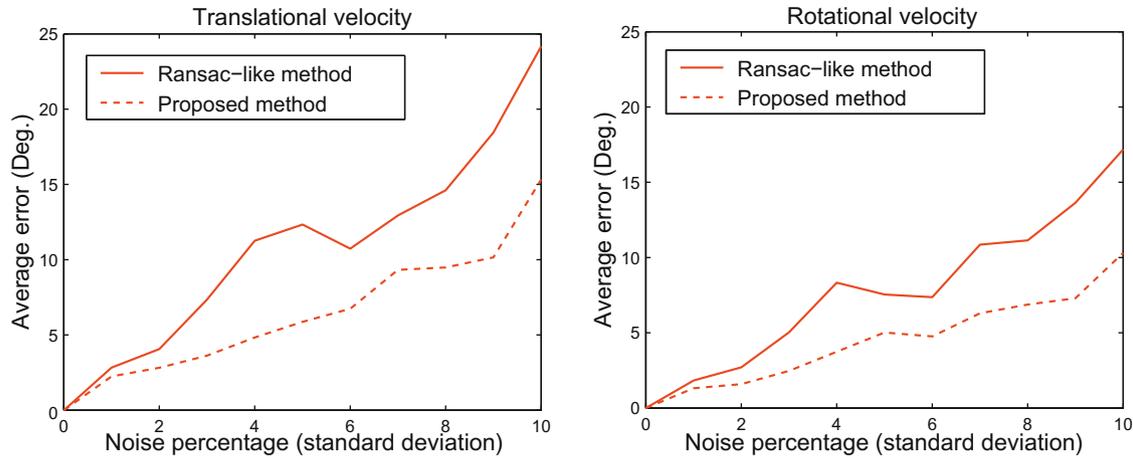


Fig. 5. Average errors obtained with a Gaussian noise and 10% of outliers.

the trimmed error $e(\epsilon)$ while trying to use as many equations/pixels as possible. The use of the denominator has the effect that cost function $\phi(\epsilon)$ will have a unique minimum. Indeed, the function $\phi(\epsilon)$ is the product of two functions: $\frac{1}{\epsilon}$ and $e(\epsilon)$. The former one is a monotonically decreasing function and the latter one is a monotonically increasing function. Recall that the error $e(\epsilon)$ is the trimmed sum of squared residuals whose plot versus the inlier percentage will be very similar to an exponential function whenever data contains outliers.

Note that λ can slightly affect the location of the minimum, that is, the optimum inlier percentage.

The minimization procedure is given a search interval $[\epsilon_a, \epsilon_b]$. It assumes that in this interval the function has a single minimum and locates the minimum by iterative bracketing with the Golden Section Search algorithm (Press et al., 1992). Fig. 3 summarizes the proposed approach that estimates the vector \mathbf{a} by combining the LTS principles and the Golden Section Search algorithm.

By default, the minimum of ϕ is searched in the interval $[0.5, 1.0]$ assuming that the inlier percentage is at least 50%. Specifying the interval more strictly improves the computational efficiency of the method. In our case, for an initial bracketing of 10%, about six iterations are sufficient to locate the minimum of $\phi(\epsilon)$ with an acceptable precision of 0.01, i.e., the interval becomes less than 1%. Note that even though the inlier percentage is not known in advance the bracketing interval can be known in advance.

3.3. The 3D velocity and the plane orientation

Once the vector $\mathbf{a} = (a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8)^T$ is recovered, the 3D velocity and the plane parameters, i.e., $\frac{V_x}{\gamma}$, $\frac{V_y}{\gamma}$, $\frac{V_z}{\gamma}$, Ω_x , Ω_y , Ω_z , α and β , can be recovered by solving the eight non-linear Eq. (8). This is carried out using the Levenberg–Marquardt technique (Press et al., 1992). Note that the translational velocity is recovered up to a scale, which is consistent with the scale ambiguity of the general structure from motion problem.

In order to get an initial solution one can adopt assumptions for which Eq. (8) can be solved in a linear fashion. Alternatively, when tracking a video sequence the estimated velocity at the previous frame can be used as an initial solution for the current frame.

In practice, one of the following two assumptions can be used for making Eq. (8) linear: (i) a negligible translational velocity along the optical axis, i.e., $V_z = 0$, and (ii) a pure translation velocity, i.e., $\Omega = \mathbf{0}$. The obtained rough solution is then handed over to the Levenberg–Marquardt technique. Notice that even the initialization assumes a particular 3D motion for the camera, the Levenberg–Marquardt technique estimates the six degrees of freedom

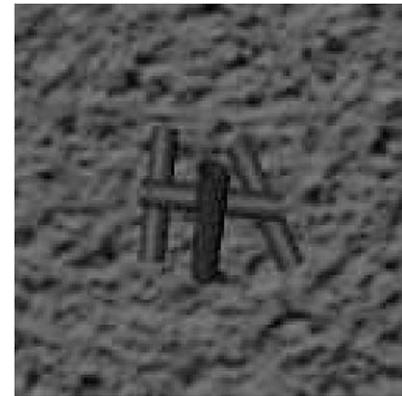


Fig. 6. The current image for which the camera motion parameters are computed. The temporal derivatives are computed using nine subsequent images.

associated with the general camera motion up to a translational scale.

4. Experimental results

Our experiments have been carried out on synthetic and real images.

4.1. Synthetic images

A synthetic planar scene was built. Its texture is described by $g(X_o, Y_o) \propto \cos(6X_o)(\sin(1.5X_o) + \sin(1.5Y_o))$

where X_o and Y_o are the 3D coordinates expressed in the plane coordinate system, see Fig. 4. The resolution of the synthesized images was 160×160 pixels. The 3D plane was placed at 100 cm from the camera whose focal length is set to 1000 pixels. A synthesized image sequence of the above planar scene was generated according to a nominal camera velocity (\mathbf{V}_n, Ω_n) . A reference image for which we like to compute the camera velocity was then fixed. The associated image derivatives can be computed or set to their theoretical values. Since we use synthetic data, the ground-truth values for the image derivatives as well as for the camera velocity are known. The nominal velocity $(\mathbf{V}_n(\text{cm/s}), \Omega_n(\text{rad/s}))$ was set to $(10, 10, 1, 0.1, 0.15, 0.1)^T$. The corresponding linear system (10) was then corrupted by adding Gaussian noise and outliers to the spatio-temporal derivatives associated with each pixel. Our approach

Table 1
Camera motion associated with the frame of Fig. 6 using the RANSAC method (first row) and the LTS (second row). The first column depicts the estimated translation axis (a unit vector), the second column depicts the estimated rotation axis (a unit vector), and the third column depicts the obtained number of inliers.

	Translation axis	Rotation axis	Inlier percentage (%)
RANSAC	(-0.0100, 0.0800, 0.9900)	(0.0700, 0.0000, -0.9900)	88.2
LTS	(0.0200, -0.00815, 0.9997)	(-0.0202, -0.0137, -0.9996)	96.6
Ground-truth motion	(0.0, 0.0, 1.0)	(0.0, 0.0, -1.0)	



Frame 5



Frame 104

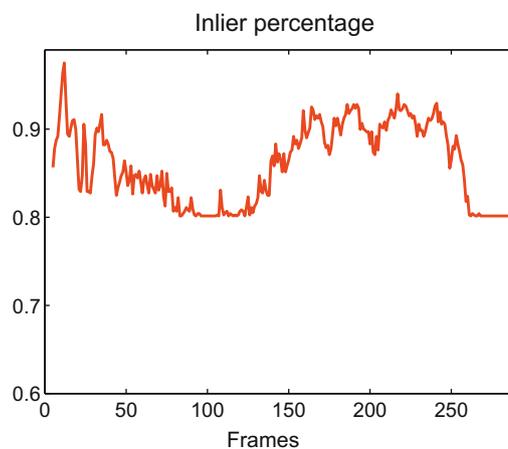
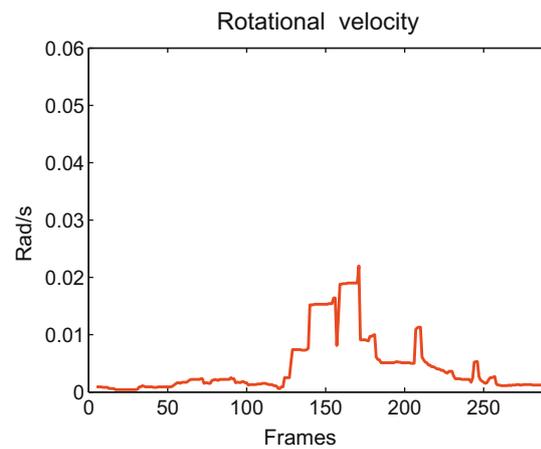
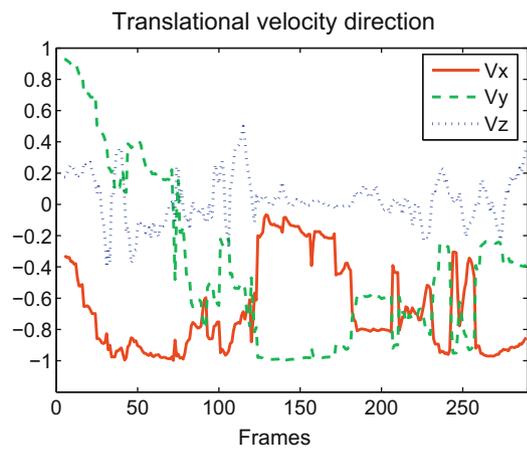


Fig. 7. Top: Two frames from the used video sequence (frames 5 and 104). Middle: The estimated translational and rotational velocities as a function of the frame number. Bottom: The estimated inlier percentage.

was then invoked to estimate the camera velocity. The discrepancies between the estimated parameters and their ground truth were then computed. In our case, the camera velocity was given by two vectors: (i) the scaled translational velocity, and (ii) the rotational velocity. Thus, the accuracy of the estimated parameters can be summarized by the angle between the direction of the estimated vector and its ground truth direction.

Fig. 5 illustrates the obtained average errors associated with the camera velocity as a function of the Gaussian noise standard deviation. Here the grey-level of the images has 256 values. The solid curve corresponds to a RANSAC-like approach adopting a robust

and automatic threshold (Rousseeuw and Leroy, 1987) (Eq. (10)), and the dashed curve to our proposed robust solution (Sections 3.2 and 3.3). Each average error was computed with 50 random trials. In this experiment the percentage of outliers was set to 10%. The location of these outliers in the image plane was chosen using a uniform distribution. The values were set as a function of the derivatives maximum.

As can be seen, unlike the RANSAC technique, our proposed method has provided more accurate 3D camera velocity. Since the translational and rotational 3D velocities are inferred from the 8 parameters (the 8-parameter motion field) using the same

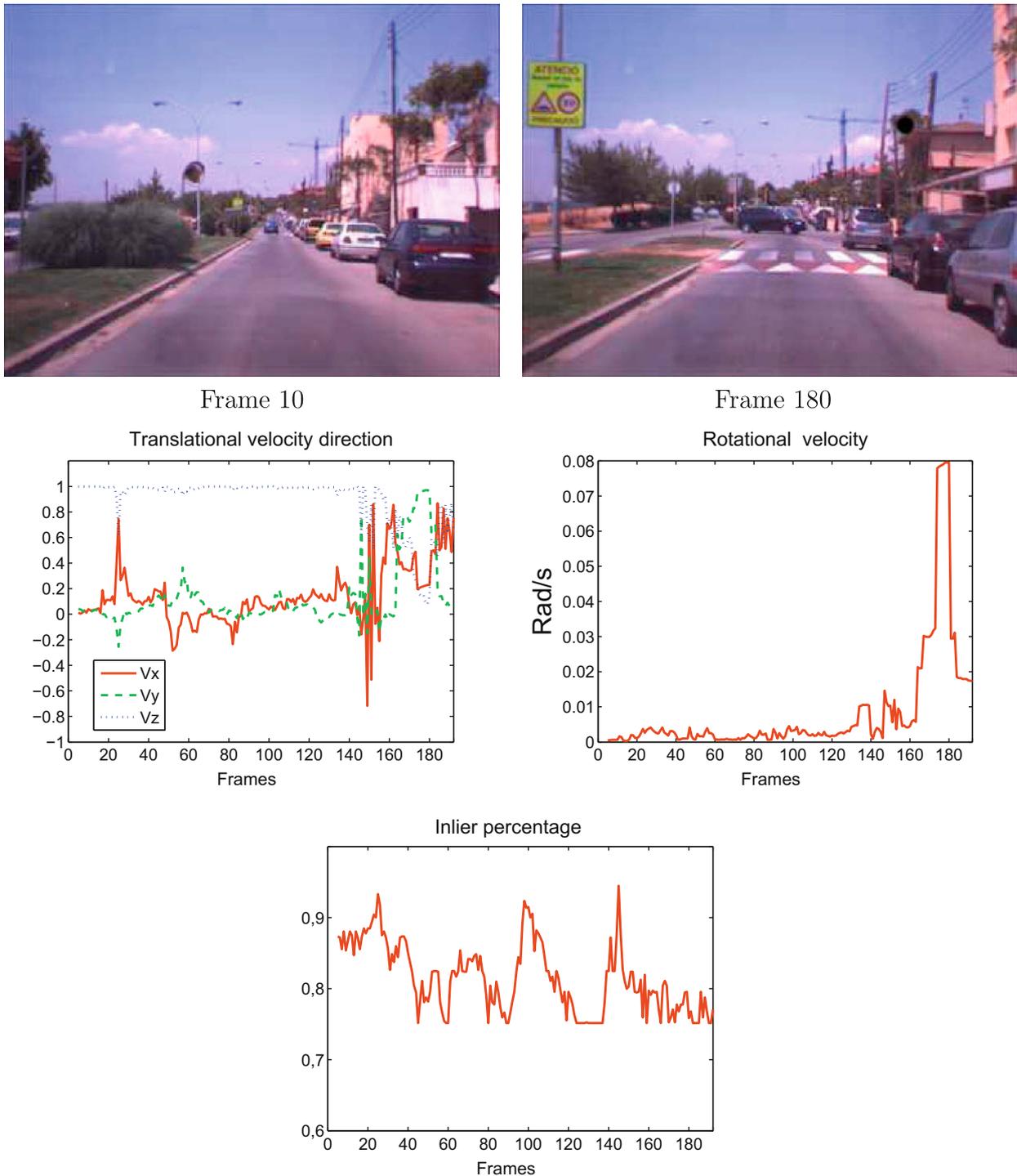


Fig. 8. Top: Two frames from the used video sequence (frames 5 and 104). Middle: The estimated translational and rotational velocities as a function of the frame number. Bottom: The estimated inlier percentage.

non-linear algorithm, it follows that the 8-parameter motion field estimated by the proposed algorithm (Section 3.2) is more accurate than the one estimated by a RANSAC-like technique. Moreover, we stress the fact that it is extremely challenging to manually select the threshold for the RANSAC technique since the residual errors are not measured in a metric space such as the Euclidean space or the image plane, instead the residual error depicts temporal derivative of the grey-levels. Thus, an adaptive threshold is better suited for the RANSAC technique.

We applied the proposed approach on the synthesized nine frame sequence depicted in Fig. 6 (only frame 5 is shown). The sequence was retrieved from the department of computer and electrical engineering, at Heriot-Watt University, UK.³ The obtained results are summarized in Table 1. We have used the RANSAC technique (first row) and the proposed approach (second row). The third row illustrates the ground-truth camera velocity which is a rotation about the optical axis combined with a translation about the same axis. The first column depicts the estimated translation axis (a unit vector), the second column depicts the estimated rotation axis (a unit vector), and the third column depicts the obtained number of inliers. As can be seen, both approaches succeeded to estimate the camera velocity with our approach being more accurate.

The CPU time associated with an image of 1400 pixels was about 3 s. As can be seen, the proposed method is slower than most of classical LS techniques. This is due to the use of the Golden Section optimization in which a complete Least Trimmed Square process is invoked.

4.2. Real images

The spatial derivatives associated with an input image are calculated by convolution with derivatives of 2D Gaussian kernels. The temporal derivatives associated with the current image are calculated using difference approximation involving a temporal window centered on the current image. The weights of the images are taken from the derivatives of a 1D Gaussian kernel.

The following two experiments used the monocular images of a commercial stereo vision system (Bumblebee from Point Grey).⁴

The first experiment was conducted on a 300-frame video sequence of a moving newspaper captured by a steady-camera, see Fig. 7. The resolution is 160×120 pixels. We used 9 consecutive images to compute the temporal derivatives. The upper part shows two different frames. The middle-left and middle-right show the estimated 3D translational velocity direction (a unit vector) and the rotational velocity $\|\Omega\|$, respectively. The lower part depicts the estimated inlier percentage. Although, the ground-truth is not known, we have found that the estimated 3D motion was very consistent with the 3D motion captured by the video. Notice that the estimated inlier percentage was high (about 90%) whenever the newspaper was the dominant object in the image. For example, its value was 90% for frame 5 and 80% for frame 104 (see Fig. 7).

The second experiment was conducted on another video sequence acquired with an on-board camera, see Fig. 8. The resolution is 320×240 pixels. For every image, only the lower part corresponding to one third of the image height is used. We used 9 consecutive images to compute the temporal derivatives. The upper part of Fig. 8 shows two different frames. The middle-left and middle-right show the estimated 3D translational velocity direction (a unit vector) and the rotational velocity $\|\Omega\|$, respectively. The lower part depicts the estimated inlier percentage. As can be seen, the instantaneous camera motion is essentially along

the optical axis for most of the frames. At the end of the sequence (from frame 160 to frame 192), the camera motion is coupled since the translational motion is no more along the optical axis alone and since there is a slight pitching. This is due to some deceleration in the car motion.

5. Conclusion

This paper presented an approach to the 3D velocity estimation from spatio-temporal image derivatives alone. What differentiates the presented work from existing ones is the use of image derivatives alone and not the optical flow field with a novel robust statistics solution. Despite the fact that the developed approach does not rely on the computation of the optical flow, the latter one is a by-product of the approach. This paper had two main contributions. First, we introduced a novel technique to the unconstrained 3D velocity estimation using image derivatives only, therefore feature extraction and tracking are not required. Second, we propose a robust method that combines the Least Trimmed Square regression and the Golden Section Search algorithm where the number of inliers is not known *a priori*. Our robust method simultaneously estimates the inlier percentage and the robust solution. On the other hand, existing LTS regression methods (e.g., Rousseeuw and Driessen, 2002) assume that the percentage of inliers is known in advance. As can be seen, the first contribution concerns the instantaneous 3D motion estimation from image data. The second contribution is within the field of robust statistics (robust linear regression). It is worth noting that motion estimation from visual data and robust statistics are two sub-fields of pattern recognition.

The conducted experiments tend to confirm that the proposed approach is more accurate than a RANSAC-like approach.

Acknowledgements

This work was partially supported by the Government of Spain under MEC Project TRA2007-62526/AUT and research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

References

- Barron, J.L., Fleet, D.J., Beauchemin, S.S., 1994. Performance of optical flow techniques. *Internat. J. Comput. Vision* 12 (1), 43–77.
- Baumela, L., Agapito, L., Bustos, P., Reid, I., 2000. Motion estimation using the differential epipolar equation. In: *IEEE Internat. Conf. on Pattern Recognition*.
- Boughorbel, F., Crilly, P., Koschan, A., Abidi, M., 2003. Estimating 3D camera motion without correspondences using a search for the best structure. *Pattern Recognition Lett.* 24, 327–337.
- Brodsky, T., Fermuller, C., 2002. Self-calibration from image derivatives. *Internat. J. Comput. Vision* 48 (2), 91–114.
- Brooks, M.J., Chojnacki, W., Baumela, L., 1997. Determining the egomotion of an uncalibrated camera from instantaneous optical flow. *J. Optical Soc. Amer. A* 14 (10), 2670–2677.
- Dellaert, F., Seitz, S., Thorpe, C., Thrun, S., 2000. Structure from motion without correspondence. In: *IEEE Conf. on Computer Vision and Pattern Recognition*.
- Dornaika, F., Chung, R., 2008. Stereo geometry from 3D ego-motion streams. *IEEE Trans. Systems Man Cybernet., Part B* 33 (2), 308–323.
- Fleet, D.J., Black, M.J., Yacoob, Y., Jepson, A.D., 2000. Design and use of linear models for image motion analysis. *Internat. J. Comput. Vision* 36 (3), 171–193.
- Hartley, R., Zisserman, A., 2000. *Multiple View Geometry in Computer Vision*. Cambridge University Press.
- Horaud, R., Dornaika, F., Espiau, B., 1998. Visually guided object grasping. *IEEE Trans. Robotics Automation* 14 (4), 525–532.
- Irani, M., 1999. Multi-frame optical flow estimation using subspace constraints. In: *IEEE Conf. on Computer Vision*.
- Jasinschi, R.S., Naveen, T., Tabatabai, A.J., Babic-Vovk, P., 2000. Apparent 3D camera velocity—Extraction and applications. *IEEE Trans. Circuits Systems Video Technol.* 10 (7), 1185–1191.
- Johnson, A., Willson, R., Cheng, Y., Goguen, J., Leger, C., Sanmartin, M., Matthies, L., 2007. Design through operation of an image-based velocity estimation system for Mars landing. *Internat. J. Comput. Vision* 74 (3), 319–341.
- Kanatani, K., 1993. *Geometric Computation for Machine Vision*. Clarendon Press.

³ <http://www.cee.hw.ac.uk/~mtc/sfda>.

⁴ www.ptgrey.com.

- Kim, E., Han, J., Kim, H., 1997. A Kalman-filtering method for 3D camera motion estimation from image sequences.
- Lourakis, M.I.A., Argyros, A.A., 2004. Vision-based camera motion recovery for augmented reality. In: *Computer Graphics Internat.*
- Malm, H., Heyden, A., 2000. A new approach to hand-eye calibration. In: *IEEE Internat. Conf. on Pattern Recognition.*
- Malm, H., Heyden, A., 2002. Self-calibration from image derivatives for active vision systems. In: *IEEE Internat. Conf. on Control, Automation, Robotics and Vision.*
- Maronna, R., Martin, D., Yohai, V., 2006. *Robust statistics: Theory and methods.* Eyrolles.
- Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992. *Numerical Recipes in C.* Cambridge University Press.
- Rodehorst, V., Hellwich, O., 2006. Genetic algorithm sample consensus (GASAC) – A parallel strategy for robust parameter estimation. In: *25 Years of RANSAC Workshop in Conjunction with the IEEE Conf. on Computer Vision and Pattern Recognition.*
- Rother, C., Carlsson, S., 2002. Linear multi view reconstruction and camera recovery using a reference plane. *Internat. J. Comput. Vision* 49 (2/3), 117–142.
- Rousseeuw, P.J., Driessen, K.V., 2002. Computing LTS regression for large data sets. *Estadistica* 54, 163–190.
- Rousseeuw, P.J., Leroy, A.M., 1987. *Robust Regression and Outlier Detection.* John Wiley and Sons, New York.
- Subbarao, R., Meer, P., 2006. Beyond RANSAC: User independent robust regression. In: *25 Years of RANSAC Workshop in Conjunction with the IEEE Conf. on Computer Vision and Pattern Recognition.*
- Weng, J., Huang, T.S., Ahuja, N., 1993. *Motion and Structure from Image Sequences.* Springer-Verlag, Berlin.
- Zucchelli, M., Jose, S., Christensen, H., 2002. Multiple plane segmentation using optical flow. In: *British Machine Vision Conf.*