Combined Vision – Inertial Navigation with Improved Outlier Robustness

Francesco Di Corato, Mario Innocenti, and Lorenzo Pollini

University of Pisa, Pisa, 56122, Italy
{dicorato,minnoce,lpollini}@dsea.unipi.it

Abstract. This paper describes a loosely coupled approach for the improvement of state estimation in autonomous inertial navigation, using image-based relative motion estimation for augmentation. The augmentation system uses a recently proposed pose estimation technique based on a *Entropy-Like* cost function, which was proven to be robust to the presence of noise and outliers in the visual features. Experimental evidence of its performance is given and compared to a state-of-the-art algorithm. Vision-inertial integrated navigation is achieved using an Indirect Kalman Navigation Filter in the framework of stochastic cloning, and the proposed robust relative pose estimation technique is used to feed a relative position fix to the navigation filter. Simulation and Experimental results are presented and compared with the results obtained via the classical RANSAC – based Direct Linear Transform approach.

1 Introduction

Inertial navigation suffers from drifts due to several factors, in particular inertial sensor errors. As a matter of fact, usually additional sensors like GPS, air data sensors or Doppler speedometers are employed to provide corrections to the navigation system. A viable augmentation alternative is the adoption of a vision system; these were employed in the past for air and land vehicle automation, like car driving [1], obstacle avoidance ([2], [3]) or formation flight ([4], [5], [6], [7]). More recently, mainly due to the increased computational power available, they are receiving more interest in the field of navigation. The use of vision for navigation is often referred to as visual odometry, which core tool is the estimation of the pose of the vision system with respect to the observed scene. Pose estimation is often the concluding step in a sequence of different phases including: detection of significant features in the scene from camera images, and tracking of them between successive frames. The presence of noise and outliers in the acquired data represents the main, in the sense of most challenging, issue in solving the Pose Estimation problem. The presence of outliers depends mainly on inaccurate key points matching and/or tracking between left and right images, in the stereo vision case, and in successive time instants. The outliers rejection problem is often solved via linear/nonlinear minimization techniques (L_2,L_∞, etc) ([8]) or via iterative refinements ([9], [10], [11]), that is via images pre/post-processing

D. Choukroun et al. (eds.), *Advances in Estimation, Navigation, and Spacecraft Control*,
DOI: 10.1007/978-3-662-44785-7_16

techniques. Well-known robust approaches in estimating camera pose are RANSAC-type algorithms [9], [10], which have no guarantee of optimality. Almost all outlier rejection schemes proposed in the literature act in a pre/post-processing phase, and most of them perform the pose estimation algorithm by minimizing a squared norm of the estimation error.

The concept of Entropy is not new in the field of estimation; it has already been applied in the last decade in the field of autonomous navigation and robotics, and the most well-known and recent works in such direction can be found in [12], where the concept of alignment via maximization of the Mutual Information is used to perform robust visual servoing and autonomous guidance tasks, in a previously visited scenario. Recently, integrated vision-inertial navigation systems are appearing in the literature; they differ mainly in the adopted coupling approach between vision and inertial measurements. Two large family exists: in the tightly coupled approach [13], [14], [15], each collected key point is added to the navigation filter state, its position is refined over time and cooperates to the estimation phase. The second large family is the loosely coupled approach [16] [17], in which the navigation filter is provided with position fixes computed by the vision system, used in this case as an external aiding sensor like it happens with GPS or altimeters. In [16] the stochastic cloning approach is introduced and used and the relative pose estimation is computed via a classical Least Square minimization. [17] uses a similar approach, but the filter is provided with relative pose measurements, which are obtained via a robust 2-norm minimization, using the Huber cost function [18], in a framework of M-estimation. The work in [19] instead, reverses the point of view and uses the stereo vision system as the main navigation sensor, while the processed IMU measurements are used to feed attitude corrections to an EKF.

In the present paper, we propose a loosely coupled approach, which uses a Stereo Vision system and an Inertial Measurement Unit. The relative pose estimations given by the vision system are computed using an *Entropy-Like* cost function, which is robust by nature with respect to the outliers in the data. The estimated pose is then used to give relative position fixes to the Indirect Kalman Navigation Filter in the framework of the stochastic cloning [20] [16]. The main contribution of the paper is showing that the adoption of the proposed robust pose estimation algorithm, which is robust to a large class of disturbances, provides a net improvement to the navigation accuracy and that there is still room for further improvements that better exploit the peculiar characteristics of the proposed pose estimation algorithm.

The paper is organized as follows: Section 2 introduces the adopted notation and the necessary perturbed inertial navigation background; Section 3 describes the application of the proposed Entropy-based cost function to pose estimation and Section 4 presents a static comparison of performance with a state-of- the-art pose estimation algorithm. Section 5 describes an error-state Extended Kalman Filter for integration of the proposed pose estimation algorithm with inertial navigation; finally Section 6 presents experimental results performed with a ground vehicle.

2 Background on Perturbed Inertial Navigation Dynamics

This paper proposes a vision-inertial integrated navigation system that, as common in precise inertial navigation, makes use of an error-state formulation where navigation errors, rather than navigation states are estimated by the filter[21]. The adopted notation is very common in the Inertial Navigation Literature: define χ as generic motion/sensitivity variable, then $\hat{\chi}$ indicates the estimated value of the true value χ, and $\tilde{\chi}$ indicates the measured value. Thus, the relationship between true values and their measurements is defined as follows:

$$\begin{aligned} \tilde{\chi} &= \chi + \nu_\chi, \\ \chi &= \hat{\chi} - \delta\chi \end{aligned} \tag{1}$$

where $\delta\chi$ is the actual navigation error, and ν_χ is the measurement error. In this work, the measurement errors is modeled as a zero-mean Gaussian process with variance $\mathrm{E}[\nu_\chi^T \nu_\chi]$. With the above notation, it is possible to write a set of perturbed navigation equations for attitude (represented here by the direction cosine matrix R_b^n), velocity in some navigation frame (we used the NED reference frame for filter implementation but any geodetic frame may be used) V^n , and position in ECEF frame r^e as:

$$\begin{cases} \hat{R}_b^n = (I - \delta\gamma \wedge) R_b^n \\ \hat{V}^n = V^n + \delta V^n \\ \hat{r}^e = r^e + \delta r^e \\ \tilde{\omega}_{ib}^b = \omega_{ib}^b + \delta\omega_{ib,b}^b + \nu_\omega \\ \tilde{f}_{ib}^b = f_{ib}^b + \delta f_{ib,b}^b + \nu_f \end{cases} \tag{2}$$

where $\delta\gamma \wedge$ denotes the skew symmetric matrix whose elements are the components of the errors vector $\delta\gamma$, which are functions of the attitude error [21]. Moreover, $\delta\omega_{ib,b}^b$ and $\delta f_{ib,b}^b$ are the bias terms in the measurements of gyroscopes and accelerometers, while ν_ω and ν_f are gyroscope and accelerometer noises, represented here as zero-mean Gaussian processes with variances $E[\nu_\omega \nu_\omega^T] = Q_\omega$ and $E[\nu_f \nu_f^T] = Q_f$. Finally δV^n and δr^e are velocity and (global) position errors respectively.

Given the definition above of the navigation error variables, the continuous time e error dynamics of the navigation equations resolved in the navigation frame [21] can be locally approximated by a compact Linear Parameter Varying (LPV) system, as in Eq. (3) :

$$\delta\dot{x}(t) = F(t)\delta x(t) + G(t)u_c(t) \tag{3}$$

The state vector $\delta x(t)$ and the input vector $u_c(t)$ are defined respectively as (we dropped the function of time for compactness of notation):

$$\delta x(t) = [\delta\gamma^{nT} \quad \delta V^{nT} \quad \delta r^{eT} \quad \delta\omega_{ib,b}^{b\,T} \quad \delta f_{ib,b}^{b\,T}]^T \tag{4}$$

$$u_c(t) = [\nu_\omega^T \quad \nu_f^T \quad \nu_{r\omega}^T \quad \nu_{rf}^T]^T \tag{5}$$

The system matrices $F(t)$ and $G(t)$ in Equation (3) come from linearization of the error dynamics, thus they change with the selected navigation frame, and locally relate the evolution of the state i to the current estimation of the state j. The reader interested in the derivation of the above equations can find all the details in [21]. The IMU biases dynamics in Eq. 3 were modeled as Brownian motions, with trivial dynamics:

$$\begin{cases} \delta\dot{\omega}_{ib,b}^{b} = \nu_{r\omega} \\ \delta\dot{f}_{ib,b}^{b} = \nu_{rf} \end{cases} \tag{6}$$

where $\mathrm{E}[\nu_{r\omega}\nu_{r\omega}^{T}] = Q_{r\omega}$ and $\mathrm{E}[\nu_{rf}\nu_{rf}^{T}] = Q_{rf}$. The covariance matrix of the Gaussian process noise $u_c(t)$, considering the sensors' noises uncorrelated and having the same noise characteristics, is given by:

$$\mathrm{E}[u_c(t)u_c(t)^{T}] = \begin{bmatrix} Q_f & 0 & 0 & 0 \\ 0 & Q_\omega & 0 & 0 \\ 0 & 0 & Q_{r\omega} & 0 \\ 0 & 0 & 0 & Q_{rf} \end{bmatrix} \delta(t-\tau) \tag{7}$$

In order to implement the filter dynamics in real-time, it is necessary to discretize the continuous time dynamics; in the remainder we will consider a time-discretized version of the above dynamics using the Euler integration method, with sample time ΔT. The final discrete-time form of the LPV perturbed system of Equation (3) can then be written as:

$$\delta x_{k+1} = \Psi_k \delta x_k + \Gamma_k u_{c,k} \tag{8}$$

where:

$$\begin{aligned} \Psi_k &= (I - F(t_k)\Delta T) \\ \Gamma_k &= G(t_k)\Delta T \end{aligned} \tag{9}$$

3 Least-Entropy Like Pose Estimation

Loosely coupled vision-aided inertial navigation with relative measurements requires the estimation of the camera motion in between successive frames. This section presents first a general framework for pose estimation, then cast this problem into the framework of Least-Entropy Like (LEL) estimation[22][23], finally presents an analysis of performance using static images.

3.1 Stereo Vision and Pose Estimation

In a stereo vision system, each camera acquires an image, relevant 2-dimensional features (points in the image plane) $\{p_{i,k}\}$ are automatically extracted from the images (for the purpose of this work we used the SIFT algorithm), identical features, that is image points belonging to the same object in the observed scene, are searched for in

the right and left images, and finally a cloud of N 3D keypoints $\{P_{i,k}\}$ is obtained by triangulation of the two corresponding sets (one for the left and one for the right images) of N 2D features $\{p_{i,k}\}$. Several techniques exist for selection and tracking of image features[5][4]; the feature selection and tracking approach used in the later simulations use stereo vision and the Scale Invariant Feature Transform (SIFT) algorithm[24][3], that easily allows both to detect, and to match features for successive triangulation. The stereo matching of features between left and right images is performed by comparing the squared distance between the SIFT descriptors of each feature in the two images, and selecting the couple with the lowest distance. Only those features that are both in the left and right images are considered valid for triangulation and tracking. With the same distance-based approach it is possible to track the features that are present in the current and past images; this makes the selection of 3D keypoints $\mathrm{P_{i,k}}$ and $P_{i,k+h}$ possible. Figure 1 shows a sample of two images with matched features (red circles), unmatched features (blue circles) and green lines representing left-right matches.

Tracking of 2D features in two successive time instants t_k and t_{k+h} produces two clouds of 3D keypoints $P_{i,k}$ and $P_{i,k+h}$ that are related by a rigid motion relationship. This relationship represents, essentially, the camera motion, in terms of translation T_k^{k+h} and rotation R_k^{k+h}, between times t_k and t_{k+h}. Thus the following relationship holds:

$$P_{i,k+h} = R_k^{k+h} P_{i,k} + T_k^{k+h} = g_k^{k+h} P_{i,k} \tag{10}$$

where $g_k^{k+h} = \left\{R_k^{k+h},\ T_k^{k+h}\right\} \in SE(3)$ is the transformation mapping the pose of the camera at the time t_k in the pose of the camera at the time t_{k+h}. The notation $g_k^{k+h} P_{i,k}$ is not actually a vector or matrix multiplication but denotes the *application* of the translation and rotation transformations $R_k^{k+h},\ T_k^{k+h}$ to the point $P_{i,k}$, as described in Eq. (10). Given any 3 dimensional parameterization of the rotation matrix, the transformation matrix in Eq. (10) can be written as: $g_k^{k+h}(\theta_p), \theta_p \in \Theta_p \subset \mathbb{R}^6$, being Θ_p the set of all possible motion parameters (angular displacements and translations). The Pose Estimation problem then becomes the estimation of the unknown motion parameters vector θ_p, given two clouds of N features at the time t_k and t_{k+h}. The solution of the problem can be found by using a minimization approach (either linear or non-linear) over the estimation residuals $E_{i,k+h}$:

$$E_{i,k+h} = P_{i,k+h} - g_k^{k+h}(\theta_p) P_{i,k} \tag{11}$$

that is:

$$\hat{\theta}_p = arg\,min_{\theta_p} \sum_{i=1}^{N} \mathcal{L}\left\{P_{i,k+h} - g_k^{k+h}(\theta_p) P_{i,k}\right\} \tag{12}$$

where $\mathcal{L}\{\cdot\}$ is a suitable cost function built upon the pose estimation residual; common choices for $\mathcal{L}\{\cdot\}$ are the 2-norm or the infinity-norm. Due to triangulation and calibration errors a number $N \geq 4$ of non-aligned points, tracked along the camera motion, are necessary for the problem to have a solution.

3.2 Robust Camera Pose Estimation Using LEL

A very relevant and desirable behavior for any feature detector and tracker is its ability to recognize features in different images even if they were taken from viewpoints distant one from the other (this means the capability to track features during camera motion for long time). When the camera moves and rotates, the same objects of the pictured scene produce different images on the camera plane: deformations and warping happens due to camera motion, change of the point of view, and perspective projection. Thus a good feature detector and tracker must be able to recognize exactly the same warped image regions. In order to achieve this property, covariant feature detectors, such as SIFT, are designed to mod-out the effects of transformations belonging to some group [25]. Such characteristic induces a certain amount of loss of information in the detected features, thus some ambiguities could raise. Figure 1 shows one example where this information loss leads to a mismatch. As a result, the whole set of features collected during the acquisition, matching and tracking phases may be affected by a certain amount of outliers. In the following, a technique which is able to give a measure of the degree of dispersion of the data will be used to design a robust pose estimator.

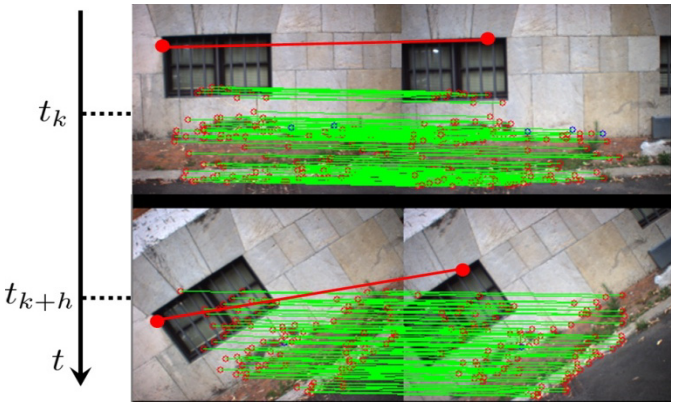

Fig. 1. Example of left-right matched features (red circles connected by the green lines) and an example of a possible matching ambiguity that may happen with the use of co-variant feature detectors (e.g. with SIFT). The matching ambiguity contaminates the data used for pose estimation with outliers (the large red dots).

A robust nonlinear alternative to Least-Square estimation was recently proposed [22]. The aim of such estimator is to give a representation of the dispersion of the residuals; such function is built on the concept of Gibbs' entropy ([26]): this is the reason why such estimator was named *Least-Entropy Like* (*LEL*) estimator. Given a reference model, which allows to match given inputs with measured outputs, minimizing the *LEL* metric of the residuals means to drive the solution toward such directions in which such residuals are in one *configuration* where not all the points have the same probability to belong to the chosen model. In [22], [27] and [23] it is shown that this *selectivity* turns out to be very important in such cases in which data are (heavily) corrupted by noise and outliers. All the implementation considerations

regarding the parameterization and minimization of the *Entropy-Like* cost function are described in detail in the references cited above.

The robust solution $\hat{\theta}_{LEL}$ to the problem of the stereo camera pose estimation between two consecutive acquisitions can be solved by minimizing the *Normalized Entropy-Like* function:

$$\hat{\theta}_{LEL} = \underset{\theta}{argmin}\left(-\frac{1}{log\, N}\sum_{i=1}^{N}\pi_i\, log\, \pi_i\right) \tag{13}$$

where:

$$\pi_i = \frac{\psi(r_i)}{\sum_{j=1}^{N}\psi(r_j)} \tag{14}$$

$$r_i = p_{i,k+h} - K_c\, proj\left\{g_k^{k+h}(\theta_p)P_{i,k}\right\} \tag{15}$$

Notice that this approach uses the re-projected pose estimation residuals r_i in 2D, instead of the pose estimation residual in 3D as in the most general form of equation (11). The re-projection error r_i involves image coordinates only that are invariant to changes in depth [28]; this leads to a better estimation accuracy. The adopted pin-hole camera model is represented, as common in computer vision, by the calibration matrix K_c and the perspective projection operator $proj\{\cdot\}$: given a generic 3D point P with coordinates P_x, P_y, P_z, the perspective projection operator is defined as:

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \Rightarrow proj\{P\} = \begin{bmatrix} P_x/P_z \\ P_y/P_z \\ 1 \end{bmatrix} \tag{16}$$

In addition, this formulation of the *Entropy-Like* function employs the Huber-like [18] function $\psi(\cdot)$ to reduce the risk of incurring into a local minimum during solution of Eq. (13).

$$\psi(a) = \begin{cases} \|a\|^2, & if\ \|a\| \le d \\ d^2, & otherwise \end{cases} \tag{17}$$

The employment of the Huber-like function allows avoiding bad conditionings of the *Entropy-Like* cost function by limiting the upper bound of the denominator in (14), and thus by avoiding the uncontrolled growth of the sum of the residuals norm due to numerical sensitivities.

Numerical solution of the optimization problem in Eq. (13) can be done in several ways. The simulations and experiments presented in this paper adopted the Levenberg-Marquardt as in [23][28]. As explained in [22] and [27], the *Entropy-Like* penalty function is nonlinear and multiple local minima may exist. Thus, the minimization must be computed with particular attention to the initial conditions. The scope of this work is such that we expect to have an acceptable local estimate of the motion given by inertial mechanization alone performed over a short period of time (between two

successive frames). Therefore, it is possible to initialize the nonlinear estimation with the parameter $\hat{\theta}_{p,0}$ that can be extracted by the best available estimate of the relative transformation:

$$\hat{g}_k^{k+h} = (\hat{g}_{k+h}^-)^{-1}\hat{g}_k^+ \rightarrow \hat{\theta}_{p,0} \tag{18}$$

where $\hat{g}_k^+$ is the best (corrected by the filter in the past) estimate of the navigation at time t_k, $\hat{g}_{k+h}^-$ is the navigation prediction at the current time t_{k+h}. The arrow symbol in Eq. (18) means that the value of $\theta_{p,0}$ is extracted by the transformation $\hat{g}_k^{k+h}$.

3.3 Experimental Results for Pose Estimation Only

LEL has already been shown to perform better than ICP [27], and a Monte Carlo Analysis have shown that it can outperform the RANSAC-based Direct Linear Transform (DLT) [29], with nonlinear refinement via Bundle Adjustment [30]. The main results are summarized here for completeness. Tests were performed both with simulated features and various level of image noise, and with real imagery; experiments were performed outdoor with a hand-held fire wire stereo camera system at a resolution of 516×388 pixels (a good trade-off between speed of image processing and accuracy of features selection and matching). An industrial 1.6 GHz PC with 1 GB RAM was used to collect the test videos; then, the video frames were processed off – line, together with the estimation algorithm.

Figure 2 show a sample image pair from an outdoor experiment; the green dots are the matched SIFT features, the red circles are the re-projected features by using the LEL pose estimation result. Figure 3 shows the sorted 2-norm of the re-projection residuals:

$$\left\|e_{r,i}\right\|^2 = \left\|p_{i,2} - K_c \,\mathrm{proj}\left\{g_1^2\left(\hat{\theta}_p^{LEL,DLT}\right)P_{i,1}\right\}\right\|^2 \tag{19}$$

computed using the motion parameters $\hat{\theta}_p$ estimated by the two methods, LEL and robust DLT. The camera calibration matrix $\mathrm{K_c}$ was determined experimentally, $\mathrm{P_{i,1}}$ are the 3D keypoints triangulated in the first position of the camera (at time $\mathrm{t_1}$), and $\mathrm{p_{i,2}}$ are the image-space coordinates of the corresponding features on the image plane of the image acquired in the final position of the camera (at time $\mathrm{t_2}$). The measurement unit of points $\mathrm{p_{i,2}}$ is pixels. The features re-projections (red circle in Fig. 3) were computed as:

$$\hat{p}_{i,2} = K_c \,\mathrm{proj}\left\{g_1^2\left(\hat{\theta}_p^{LEL}\right)P_{i,1}\right\} \tag{20}$$

In addition, Figure 3 highlights the mismatching between the measured and estimated projection, once the optimal transformation is applied to an outlier (marked with two red 'x' connected via the red line). It should be noticed that the robust DLT algorithm tries to exclude the outliers and the noisiest points from the dataset before solving the pose estimation problem, while LEL performs both pose estimation and outlier rejection in one step. Furthermore, it can be stated that LEL (which is run on the whole dataset) is able to perform as good as a 2-norm approach like DLT (that needs the

dataset to be purged by outliers and ambiguous points) [22]. Although no analytical guarantee is available yet, the LEL algorithm performs in general as good as the robust DLT algorithm with nonlinear refinement (tuned with our best efforts), which was used as benchmark. In some particular experiments the accuracy of the methods cannot be stated in an absolute fashion, since no ground truth was available in order to compare algorithms.

Finally, a Monte Carlo analysis was performed to assess the robustness of the proposed algorithm to outliers; LEL provided less re-projection error then DLT for all the tested percentage of presence of outliers [28].

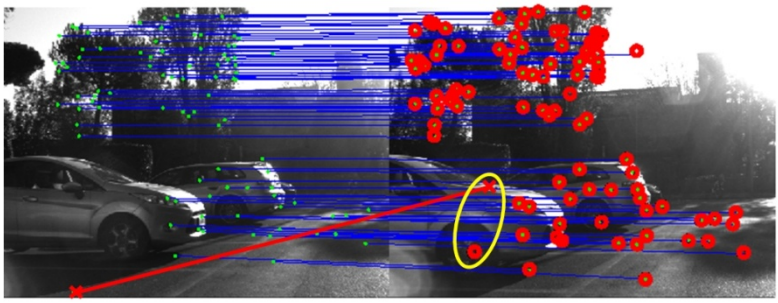

Fig. 2. Outdoor experiment. Image pair with points correspondences and estimation results. The green dots are the matched SIFT features. The red circles are the re-projected features by using the LEL pose estimation result.

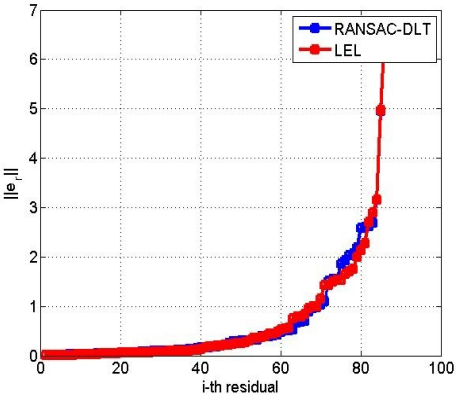

Fig. 3. Outdoor experiment. Sorted re-projection errors.

4 Navigation and Kalman Filtering with Relative Pose Measurements

Usually all aiding sensors produce absolute measures (with respect to a known and fixed reference) of the estimated variables (e.g. GPS measures $\tilde{r}_e$ and $\tilde{V}_e$) while, in the case of visual odometry, the motion measurements are relative only (i.e. only the relative displacement between two successive images is measured). This section

summarizes the equations used for the fusion of the relative motion measurements, given from the pose estimation algorithm, and the inertial data.

4.1 Definition of the Relative Pose Pseudo-Measurement Error

The camera pose at time $t_k$can be related, with respect to initial position (at time t_0), to the inertial mechanization states (position and attitude) as:

$$g_k^0 = \{R_k^0,\ T_k^0\} \tag{21}$$

where $R_k^0 = R_{b,k}^n = R_b^n(t_k)$ and T_k^0 represents the position of the origin of the Navigation/Body frame at time t_k seen from the initial Navigation frame (at time t_0). For the purposes of this paper, we assumed that the relative displacement (latitude and longitude) between successive images is small enough so that the navigation frame (NED) can be considered orientation-invariant (with respect to the ECEF frame); thus a simple planar projection can be used (approximation of flat surface), to approximate motion in the neighborhood of starting point. Thus the camera position can be obtained with Eq. (22):

$$T_k^0 = \xi(r^e(t_k)) = \begin{bmatrix} P_n(\phi_k - \phi_0) \\ P_n \cos\phi_k\,(\lambda_k - \lambda_0) \\ h_k \end{bmatrix} \tag{22}$$

where $r^e(0) = [\phi_0 \quad \lambda_0 \quad 0]$ represents the vector of coordinates in the ECEF frame (latitude, longitude, altitude) corresponding to the initial position of the vehicle, when the navigation task began its execution (at time t_0). P_n is the radius of curvature normal to the ellipsoid surface at the point of tangency at the given latitude ϕ [21].

Given two pairs of successive images at time t_k and t_{k+h}, the relative motion, that must be computed by the vision system, g_k^{k+h} is related to the absolute poses at time t_k and t_{k+h} by:

$$g_k^{k+h} = \left\{R_k^{k+h}, T_k^{k+h}\right\} = \{R_0^{k+h}R_k^0\,, R_0^{k+h}(T_k^0 - T_{k+h}^0)\} \tag{23}$$

It is now necessary to define a filter output that can be used to construct a measurement residual with the vision system output. Thus, first we construct a navigation position error estimate using the planar projection operator $\xi(\cdot)$ and an attitude error:

$$\delta y_k = H_k \delta x_k \triangleq \begin{bmatrix} \delta T_k^0 \\ \delta \gamma_k^0 \end{bmatrix} \tag{24}$$

where:

$$H_k = \begin{bmatrix} 0 & 0 & R_{n,k}^b \left.\frac{\partial T_k^0}{\partial r_k^e}\right|_{\hat{r}_k^e} & 0 & 0 \\ \left.\frac{\partial R_{n,k}^b}{\partial \varphi_k}\right|_{\hat{\varphi}_k} & 0 & 0 & 0 & \left.\frac{\partial R_{n,k}^b}{\partial \omega_{ib,b,k}^b}\right|_{\hat{\omega}_{ib,b,k}^b, \hat{\varphi}_k} \end{bmatrix} \tag{25}$$

It is worth to highlight that $\left.\frac{\partial T_k^0}{\partial r_k^e}\right|_{\hat{r}_k^e}$ is the Jacobian of the function $\xi(\cdot)$ with respect to the estimated position in ECEF frame around $\hat{\mathrm{r}}_{\mathrm{k}}^{\mathrm{e}}$. Then we can estimate the relative navigation error $\Delta\delta y_{k+h}$ between time t_{k+h} and t_k as:

$$\Delta\delta y_{k+h} = H_{k+h}\delta x_{k+h} - H_k \delta x_k \tag{26}$$

Note that H_{k+h} takes the same form of H_{k} except that it is computed with respect to the state at time $k+h$. $\Delta\delta\mathrm{y}_{\mathrm{k+h}}$ is the estimate of the navigation error of the value $\hat{\mathrm{T}}_{\mathrm{k}}^{\mathrm{k+h}}$ computed by the inertial mechanization.

Since the vision system actually measures $\tilde{g}_k^{k+h} = \left\{\tilde{R}_k^{k+h}, \tilde{T}_k^{k+h}\right\}$, it is possible to compute a pseudo-measure of the relative pose error from the measured relative pose $\tilde{\mathrm{g}}_{\mathrm{k}}^{\mathrm{k+h}}$ and its estimation $\hat{\mathrm{g}}_{\mathrm{k}}^{\mathrm{k+h}}$ reconstructed from the navigation equations, as a function of the filter state. In our case, such error can be written as:

$$\Delta\delta y_{k+h}^* = \begin{bmatrix} \tilde{T}_k^{k+h} - \hat{T}_k^{k+h} \\ \tilde{\varphi}_k^{k+h} - \hat{\varphi}_k^{k+h} \end{bmatrix} = \begin{bmatrix} \Delta\delta y_{T,k+h}^* \\ \Delta\delta y_{R,k+h}^* \end{bmatrix} \tag{27}$$

We aim at writing the pseudo-measure of relative pose error $\Delta\delta y_{k+h}^*$ as a function of the filter state.

The estimated relative translation is:

$$\hat{T}_k^{k+h} = \hat{R}_0^{k+h}\left(\hat{T}_k^0 - \hat{T}_{k+h}^0\right) \tag{28}$$

while, the measured relative translation is, by definition, equal to the actual data corrupted by noise ν_k:

$$\begin{aligned} \tilde{T}_k^{k+h} &= T_k^{k+h} + \nu_{T,k+h} \\ &= R_0^{k+h}(T_k^0 - T_{k+h}^0) + \nu_{T,k+h} \\ &= R_0^{k+h}\left(\hat{T}_k^0 - \delta T_k^0 - \hat{T}_{k+h}^0 - \delta T_{k+h}^0\right) + \nu_{T,k+h} \\ &= R_0^{k+h}\left(\hat{T}_k^0 - \hat{T}_{k+h}^0\right) - R_0^{k+h}(\delta T_k^0 - \delta T_{k+h}^0) + \nu_{T,k+h} \end{aligned} \tag{29}$$

The pseudo-measure of the relative translation error can be rewritten as a function of the states (current and of the past) of the indirect Kalman Filter only:

$$\begin{aligned} \Delta\delta y_{T,k+h}^* &= \tilde{T}_k^{k+h} - \hat{T}_k^{k+h} \\ &= R_0^{k+h}\left(\hat{T}_k^0 - \hat{T}_{k+h}^0\right) - R_0^{k+h}\left(\delta T_{k+h}^0 - \delta T_k^0\right) + \nu_{T,k+h} \\ &\quad - \hat{R}_0^{k+h}\left(\hat{T}_k^0 - \hat{T}_{k+h}^0\right) \\ &\approx -\hat{R}_0^{k+h}\left(\hat{T}_k^0 - \hat{T}_{k+h}^0\right) \wedge \delta\gamma_{k+h} - \hat{R}_0^{k+h}\left(\delta T_{k+h}^0 - \delta T_k^0\right) \\ &\quad + \nu_{T,k+h} \end{aligned} \tag{30}$$

where $\delta\gamma_{k+h}$ is the attitude error at the time step $k+h$. The previous equation was obtained by neglecting the cross products between error terms and by using the fact that the attitude error $\delta\gamma_{k+h}$ is defined for the matrix R_0^{k+h}, via Equation (2). Thus, by taking the transpose, we have:

$$R_0^{k+h} = (R_{k+h}^0)^T = \left(\hat{R}_{k+h}^0\right)^T (I - \delta\gamma_{k+h}\wedge)^T = \hat{R}_0^{k+h}(I + \delta\gamma_{k+h}\wedge) \tag{31}$$

It is not straightforward to obtain in the same manner, i.e. algebraically, the pseudo-measure of the relative rotation error $\Delta\delta y_{R,k+h}^*$ as a function of the filter state. It is convenient to derive the equation relative to $\Delta\delta y_{R,k+h}^*$ via partial derivatives instead, that is:

$$\Delta\delta y_{R,k+h}^* \approx \left.\frac{\partial R_k^{k+h}}{\partial \gamma_k}\right|_{\hat{R}_k^0, \hat{R}_{k+h}^0} \delta\gamma_k + \left.\frac{\partial R_k^{k+h}}{\partial \gamma_{k+h}}\right|_{\hat{R}_k^0, \hat{R}_{k+h}^0} \delta\gamma_{k+h} + \nu_{R,k+h} \tag{32}$$

The relative pose measurement error do depend on the motion variables corresponding to the current time (via $\hat{r}_{k+h}^e$ and $\hat{R}_0^{k+h}$) and to some steps in the past (via $\hat{r}_k^e$ and $\hat{R}_0^k$). Thus it is necessary to augment the filter state with a memory of the past; this allow to keep track of the cross covariance of estimated navigation between the two time instants [20][23].

The state of the error navigation filter is augmented with one exact copy $\delta\breve{x}_k$ of itself when a reference frame is acquired. Suppose a new reference frame arrives at time t_k, the state of the Kalman Filter will be set to:

$$\delta\bar{x}_k = \begin{bmatrix} \delta\breve{x}_k \\ \delta x_k \end{bmatrix} \tag{33}$$

and the state covariance matrix is set to:

$$\bar{P}_k = \begin{bmatrix} P_k & P_k \\ P_k & P_k \end{bmatrix} \tag{34}$$

being $P_k = E\{\delta x_k \delta x_k^T\}$. The state copy $\delta\breve{x}_k$ is initialized to δx_k and is kept constant during the filter propagation, whereas the state vector δx_k is propagated according to error dynamics.

4.2 Kalman Filter Prediction Step

At each time step inertial mechanization is performed to obtain a new estimate of the vehicle state (position, velocity and accelerometer biases):

$$\hat{x}_{k+1}^- = f(\hat{x}_k, u_k) \tag{35}$$

where $f(\cdot)$ represents the discretized version of the standard INS mechanization [21], which maps corrected navigation states on the states at the next time step; variable

$\hat{x}_{k+1}^{-}$ is the estimation of the vehicle position and velocity (at the time t_{k+1}), before the corrections, if any, produced by Kalman Filter are applied (i.e. the a priori estimate).

According to the above discussion, the indirect Kalman Filter prediction step is performed using:

$$\delta\bar{x}_{k+1} = \begin{bmatrix}\delta\bar{x}_{k+1}\\ \delta x_{k+1}^{-}\end{bmatrix} = \begin{bmatrix} I & 0\\ 0 & \Psi_k\end{bmatrix}\begin{bmatrix}\delta\bar{x}_k\\ \delta x_k\end{bmatrix} + \begin{bmatrix}0\\ \Gamma_k\end{bmatrix} w_k = \bar{\Psi}_k\delta\bar{x}_k + \bar{\Gamma}_k w_k \tag{36}$$

The propagation equation for the covariance matrix is, like for standard Kalman filtering:

$$\bar{P}_{k+1} = \bar{\Psi}_k\bar{P}_k\bar{\Psi}_k^T + \bar{\Gamma}_k Q\bar{\Gamma}_k^T \tag{37}$$

where Q is the process noise covariance matrix. After h steps (the time span needed to obtain the second image) the covariance matrix becomes:

$$\bar{P}_{k+h} = \begin{bmatrix} P_k & P_k\left(\prod_{i=1}^{h}\Psi_{k+i}\right)^T\\ P_k\prod_{i=1}^{h}\Psi_{k+i} & P_{k+h}\end{bmatrix} \tag{38}$$

Note the off-diagonal blocks that represent the cross-correlation between the navigation errors at the time t_k and t_{k+h}.

4.3 Kalman Filter Correction Step

When the vision system provides a new relative pose measurement, the update step is performed, as follows:

$$\begin{aligned}
S_{k+h} &= [H_k \quad H_{k+h}]\bar{P}_{k+h}^{-}\begin{bmatrix}H_k^T\\ H_{k+h}^T\end{bmatrix} + R_k\\
K_{k+h} &= \bar{P}_{k+h}^{-}\begin{bmatrix}H_k^T\\ H_{k+h}^T\end{bmatrix}S_{k+h}^{-1}\\
\Delta\delta y_{k+h}^{*} &= \hat{T}_k^{k+h} - \hat{T}_k^{k+h}\\
\delta x_{k+h}^{+} &= \delta x_{k+h}^{-} + K_{k+h}|_{\delta x_{k+h}}(\Delta\delta y_{k+h}^{*} - \Delta\delta y_{k+h})\\
\hat{x}_{k+h}^{+} &= \hat{x}_{k+h}^{-} + \delta x_{k+h}^{+}\\
\bar{P}_{k+h}^{+} &= \bar{P}_{k+h}^{-} - K_{k+h}[H_k \quad H_{k+h}]\bar{P}_{k+h}^{-}
\end{aligned} \tag{39}$$

where R_k is the measurement noise covariance matrix. Variable $\hat{x}_{k+h}^{+}$ is the estimation of the vehicle position and velocity (at the time t_{k+h}), given the corrections produced by Kalman Filter (i.e. the a posteriori estimate).

5 Experimental Results

Simulation results with a comparison of the proposed navigation filter with the RANSAC-based Direct Linear Transform, with nonlinear refinement via Bundle

Adjustment demonstrated already the viability of the LEL approach [28] where an inertial grade gyroscope unit was assumed available, and only vision-estimated translational motion was used to correct filter state. The simulations were performed by generating sample (noisy) accelerations and clean angular velocities. The result was a sample camera trajectory in 6DOF. The accelerations and angular velocities movements were generated by using a VTOL quad rotor aircraft simulator, and, in order to emulate the presence of outliers in the data, random numbers were added to the image-space 2D coordinates. Both algorithms produced small errors (few centimeters) but the DLT visual odometry solution resulted to be noisier.

This section presents a sample experiment performed outdoor in the Univ. of Pisa Faculty of Engineering parking lot using a wheeled ground vehicle. The cameras and hardware used was the same of the static experiments. A snapshot of about 80 seconds, where recognition of the actual travelled path was easier, was extracted from a longer recording. The filter state was initially coarse aligned with gravity to estimate initial roll and pitch angles of the camera-IMU system; then motion began and the vehicle was driven along a straight path, followed by a 180 degrees turn, and a successive almost straight path that brought the vehicle back to its initial position.

Figure 4 shows the time histories of the estimated position, velocity and attitude angles during the motion of the vehicle. It appears clearly that the navigation filter produces smooth estimation with minimal drift. The expected drift in pure inertial navigation (i.e. without any aiding), according to the characteristics of the low-cost inertial sensor suite used, would be of several tens of meters in the same time range.
Figure 5 shows the estimated vehicle trajectory in the local geodetic frame. Three trajectories are shown: the output of the integrated vision-inertial system, the result of running the visual odometry algorithm (integration of relative position fixes only, and no inertial data) on the pose estimation results provided by DLT and LEL. By knowing the actual path followed by the vehicle, it appears clearly that the best estimate in terms of navigation accuracy is given by the integrated visual-inertial navigation: the path starts and returns to the same point. The result of visual odometry for both DLT and LEL show instead a relevant drift in the position estimation. Nevertheless the integrated navigation filter succeeds in filtering out these drifts.

Figure 6 shows a comparison of the estimates of relative camera motion performed by: DLT algorithm using visual features only, LEL algorithm using visual features only, inertial mechanization. The latter represents the translation and rotation parameters that are actually estimated by the filter just before a new image is acquired, and that are used to initialize the solver for the LEL minimization problem. The figure proposes selections of the time range where the differences between the three are large. It appears that LEL and DLT pose estimation solutions are often very near to each other, even if LEL is often less noisy then DLT. In addition, the smoothing effect performed by the Kalman filter on the noisy visual measurements is noticeable throughout the entire time range of the experiment.

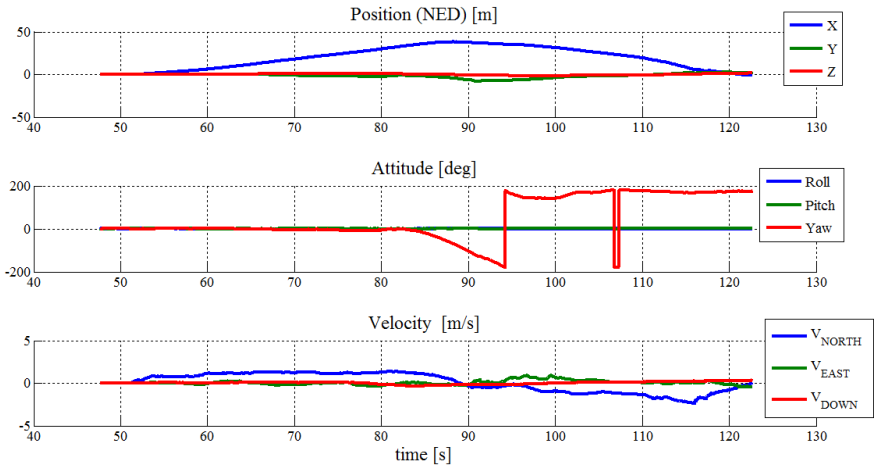

Fig. 4. Time histories of vehicle position (meters from a geodetic fixed reference frame), attitude and velocity in NED

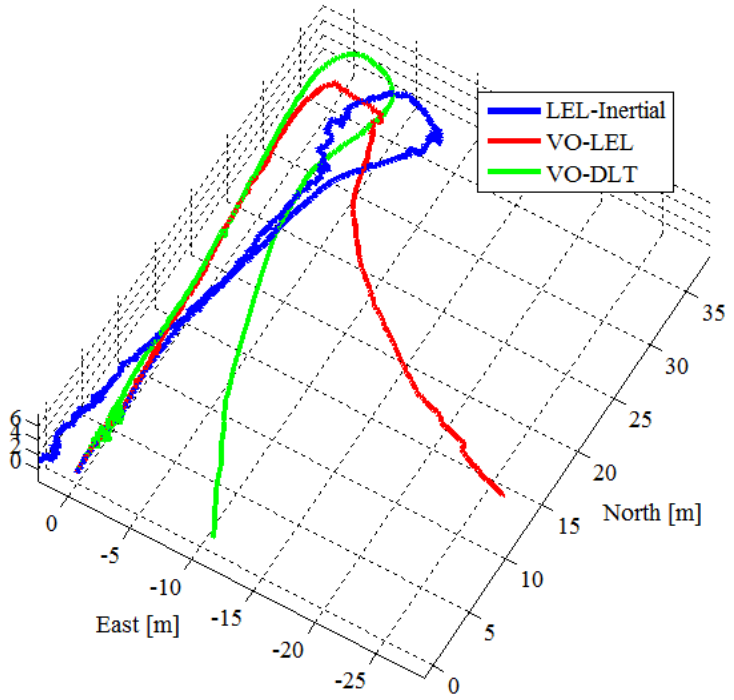

Fig. 5. Trajectories in the local geodetic frame. Comparison of the output of the vision-inertial navigation filter, with visual odometry (VO) performed integrating only the LEL and DLT relative position fixes.

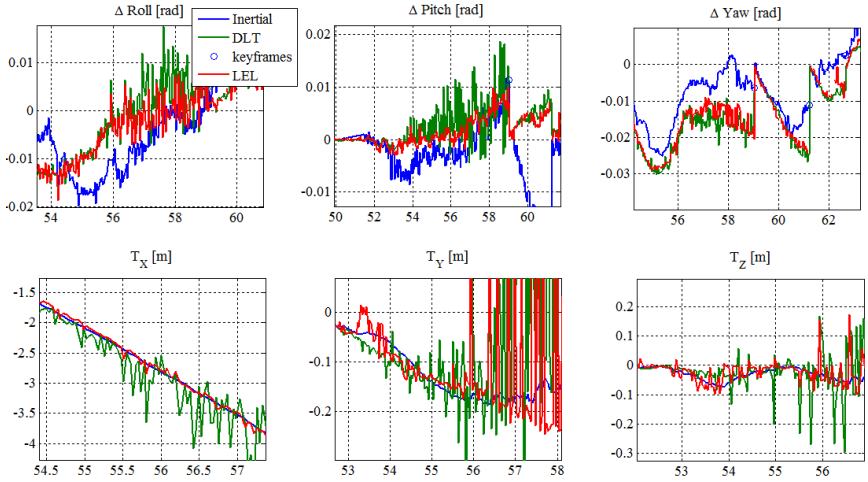

Fig. 6. Comparison of Inertial, DLT, and LEL estimates of relative camera motion

6 Conclusions

A robust loose-coupling approach to vision-augmented inertial navigation, which makes use of a novel cost function, the Entropy of relative squared residuals, was proposed. The LEL algorithm was shown with simulations and experimental tests to be robust to the presence of noise and outliers in the visual features. An error-state Kalman filter was designed and experimental results were presented; these show that using the LEL approach for pose estimation, although may produce noisy estimates, allows to reduce the navigation drift, with respect to a robust technique based on 2-norm minimization plus nonlinear refinement via Bundle Adjustment.

Acknowledgments. Support for the work of the first author was provided by Northrop Grumman Italia Spa.

References

1. Broggi, A.: Robust Real-Time Lane and Road Detection in Critical Shadow Conditions. In: Proceedings of IEEE International Symposium on Computer Vision, Coral Gables, Florida (1995)
2. Watanabe, Y., Calise, A.J., Johnson, E.N.: Vision-Based Obstacle Avoidance for UAVs. In: AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, South Carolina (2007)
3. Pollini, L., Greco, F., Mati, R., Innocenti, M., Tortelli, A.: Stereo Vision Obstacle Detection based on Scale Invariant Feature Transform Algorithm. In: AIAA Guidance Navigation and Control Conference, Hilton Head, South Carolina (2007)

4. Innocenti, M., Mati, R., Pollini, L.: Vision Algorithms for Formation Flight and Aerial Refueling with Optimal Marker Labeling. In: AIAA Modeling and Simulation Technologies Conference, vol. 1, pp. 1–15 (2005)
5. Campa, G., Mammarella, M., Napolitano, M.R., Fravolini, M.L., Pollini, L., Stolarik, B.: A comparison of Pose Estimation algorithms for Machine Vision based Aerial Refueling for UAVs. In: Mediterranean Control Conference 2006, vol. 1, pp. 1–6 (2006)
6. Giulietti, F., Pollini, L., Innocenti, M., Napolitano, M.: Dynamic and control issues of formation flight. Aerospace Science and Technology 9(1), 65–71 (2005)
7. Pollini, L., Innocenti, M., Giulietti, F.: Formation Flight: a Behavioral Approach. In: AIAA Guidance, Navigation and Control Conference, Montreal, Canada, vol. 1 (2001)
8. Hartley, R.I., Kahl, F.: Optimal algorithms in multiview geometry. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part I. LNCS, vol. 4843, pp. 13–34. Springer, Heidelberg (2007)
9. Milella, A., Siegwart, R.: Stereo-Based Ego-Motion Estimation Using Pixel Tracking and Iterative Closest Point. In: Proceedings of the Fourth IEEE International Conference on Computer Vision Systems (2006)
10. Nistér, D.: Preemptive RANSAC for Live Structure and Motion Estimation. In: IEEE International Conference on Computer Vision (2003)
11. Olson, C.F., Matthies, L.H., Schoppers, M., Maimone, M.W.: Rover navigation using stereo ego-motion. Robotics and Autonomous Systems 43, 215–229 (2003)
12. Dame, A., Marchand, E.: Entropy-based visual servoing. In: IEEE International Conference on Robotics and Automation, Kobe, Japan (2009)
13. Jones, E., Soatto, S.: Visual-Inertial Navigation, Mapping and Localization: A Scalable Real-Time Causal Approach. International Journal of Robotics Research (2010)
14. Mourikis, A., Trawny, N., Roumeliotis, S., Johnson, A., Ansar, A., Matthies, L.: Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing. IEEE Transactions on Robotics 25(2), 264–280 (2009)
15. Bryson, M., Reid, A., Ramos, F., Sukkarieh, S.: Airborne vision-based mapping and classification of large farmland environments. J. Field Robot. 27(5), 632–655 (2010)
16. Roumeliotis, S., Johnson, A., Montgomery, J.: Augmenting inertial navigation with image-based motion estimation. In: Proceedings of IEEE International Conference on Robotics and Automation (2002)
17. Tardif, J.-P., George, M., Laverne, M., Kelly, A., Stentz, A.: A new approach to vision-aided inertial navigation. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2010)
18. Huber, P.J., Ronchetti, E.M.: Robust Statistics, 2nd edn. John Wiley & Sons, Inc. (2009)
19. Konolige, K., Agrawal, M., Solà, J.: Large-Scale Visual Odometry for Rough Terrain. Robotics Research; Springer Tracts in Advanced Robotics 66, 201–212 (2011)
20. Roumeliotis, S., Burdick, J.: Stochastic Cloning: A generalized framework for processing relative state measurements. In: Proceedings of IEEE International Conference on Robotics and Automation (2002)
21. Rogers, R.: Applied Mathematics in Integrated Navigation Systems. American Institute of Aeronautics and Astronautics (2007)
22. Lowe, D.: Object Recognition from Local Scale-Invariant Features. In: Proc. of the International Conference on Computer Vision (ICCV) (1999)
23. Le, H., Kendall, D.G.: The Riemannian Structure of Euclidean Shape Spaces: A Novel Environment for Statistics. The Annals of Statistics 21(3), 1225–1271 (1993)
24. Indiveri, G.: An Entropy-Like Estimator for Robust Parameter Identification. Entropy 11, 560–585 (2009)

25. Chakrabarti, C., De, K.: Boltzmann-Gibbs entropy: axiomatic characterization and application. Journal of Mathematics and Mathematical Sciences 23(4), 243–251 (2000)
26. Di Corato, F., Innocenti, M., Indiveri, G., Pollini, L.: An Entropy-Like Approach to Vision Based Autonomous Navigation. In: The Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China (2011)
27. Di Corato, F., Innocenti, M., Pollini, L.: An Entropy-Like Approach to Vision-Aided Inertial Navigation. In: The Proceedings of the 18th IFAC World Congress, Milan, Italy (2011)
28. Di Corato, F., Innocenti, M., Pollini, L.: Robust Vision-Aided Inertial Navigation via Entropy-Like Relative Pose Estimation. Journal of Gyroscopy and Navigation 4(1), 1–13 (2013)
29. Nistér, D., Naroditsky, O., Bergen, J.: Visual Odometry. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (2004)
30. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
31. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment – A modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) Vision Algorithms 1999. LNCS, vol. 1883, pp. 298–372. Springer, Heidelberg (2000)
32. Vedaldi, A., Fulkerson, B.: Vlfeat: an open and portable library of computer vision algorithms. In: Proceedings of the International Conference on Multimedia (MM 2010) (2010)
33. Lee, T.: Vision Lab Geometry Library. UCLA VisionLab (2008), `http://vision.ucla.edu/vlg/`
34. Innocenti, M., Pollini, L.: A Synthetic Environment for Dynamic Systems Control and Distributed Simulation. IEEE Control Systems Magazine 20(2), 49–61 (2000)
35. Pollini, L., Greco, F., Mati, R., Innocenti, M.: Stereo Vision Obstacle Detection based on Scale Invariant Feature Transform Algorithm. In: Guidance Navigation and Control Conference (2007)
36. Hornegger, J., Tomasi, C.: Representation issues in the ML estimation of camera motion. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision (1999)
37. Schmidt, J., Niemann, H.: Using Quaternions for Parametrizing 3-D Rotations in Unconstrained Nonlinear Optimization. In: Proceedings of the Vision Modeling and Visualization Conference (2001)