Probabilistic Egomotion for Stereo Visual Odometry

H. Silva · A. Bernardino · E. Silva

Abstract We present a novel approach of Stereo Visual Odometry for vehicles equipped with calibrated stereo cameras. We combine a dense probabilistic 5D egomotion estimation method with a sparse keypoint based stereo approach to provide high quality estimates of vehicle's angular and linear velocities. To validate our approach, we perform two sets of experiments with a well known benchmarking dataset. First, we assess the quality of the raw velocity estimates in comparison to classical pose estimation algorithms. Second, we added to our method's instantaneous velocity estimates a Kalman Filter and compare its performance with a well known open source stereo Visual Odometry library. The presented results compare favorably with state-of-the-art approaches, mainly in the estimation of the angular velocities, where significant improvements are achieved.

Keywords Stereo vision · Visual Odometry · Egomotion · Visual Navigation

1 Introduction

Visual Navigation systems [2], have been subject of important developments by the robotics research community in the last decade. The use of low-cost visual sensors (cameras) together with Inertial Measurement Units (IMU) are becoming ubiquitous on today's modern mobile robots and pushing research on high-performance algorithms for robot navigation.

The use of vision based methods in navigation systems is justified by their ability to ground perception to static features in the environment and measure the robot relative displacement with respect to those features. Therefore, vision based methods are, in principle, less prone to bias and drifts common in other navigation sensory modalities like IMU's and wheel odometers.

In [30] VO was defined as the process of estimating a vehicle's egomotion by using vision cameras (Fig. 1). Cameras work as linear and angular velocity sensors but, because they rely on the observation of fixed points in the environment, they typically provide measurements with less drift than IMU's and wheel odometers. Ultimately, the linear and angular velocities obtained from the egomotion estimation process are integrated along time to provide the relative pose of the robot with respect to some inertial frame. In

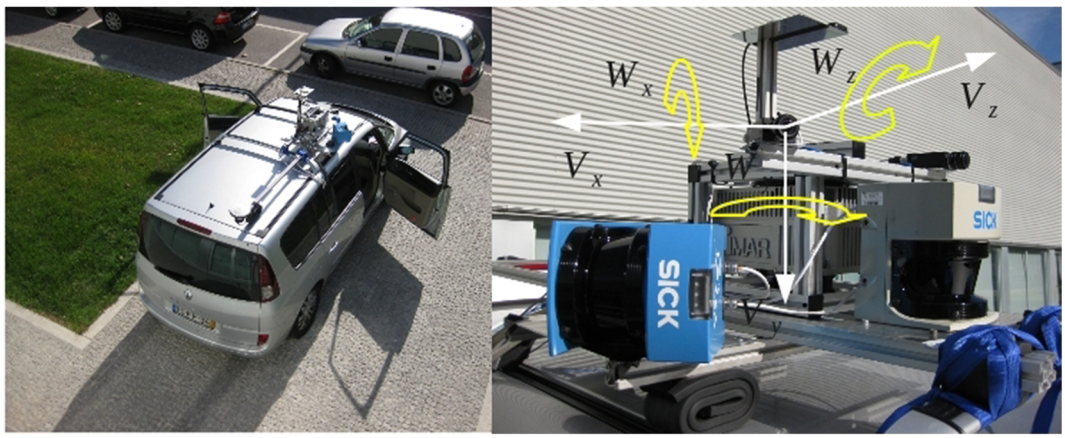


Fig. 1 The INESC TEC camera calibrated setup and inertial navigation system. Illustration of a vehicle-like robot where the methods of this paper could be applied (not actually used in the experiments)

this paper we focus on the visual egomotion estimation process, since it is the most critical component of a VO system.

Most of the research on VO employ sparse feature based methods. These methods have the advantage of being fast, since only a subset of image points is processed, but depend critically on the features to track between adjacent time frames, and are often sensitive to noise and outliers. On the contrary, dense (pixel based) methods combined with probabilistic approaches have demonstrated higher robustness to those source of errors. Domke and Aloimonos [5] proposed a dense probabilistic egomotion estimation method based upon epipolar geometry for describing the motion of a camera. The method does not commit feature matches between two images on adjacent time frames, but instead computes a probability distribution over all possible correspondences. By exploiting a larger amount of data, a better performance is achieved under noisy measurements. However, Domke method is more computationally expensive than standard feature based methods and only computes the direction of the linear motion, but not the translation scale factor (i.e., the amplitude of the linear motion).

To overcome such limitations, we use a dense probabilistic method such as the one developed by Domke and Aloimonos [5] but with three important contributions. First we add a sparse feature based method that provides stereo vision information needed to compute the translation scale factor. Second we implement a fast correspondence method based on recursive Zero-Mean Normalized Cross Correlation (ZNCC) scheme for computational efficiency. Third we integrate the obtained velocity estimates in an Kalman Filter able to reduce the noise present in the instantaneous measurements. Our proposed approach to perform stereo egomotion combines a deterministic sparse feature based method for obtaining depth estimation, with a dense probabilistic egomotion approach that allows to recover camera rotation (R) and translation ($\tilde{t}$) up to a scale factor (α). For recovering the missing translation scale factor (α) we use a Procrustes Absolute Orientation method, that takes registered 3D point information from two adjacent time frames.

This paper is an extension of the work conducted by Silva et al. [31], with the addition of scale invariant feature detectors (SIFT) and temporal filtering via Standard Kalman Filter (KF) that increase the accuracy of our Stereo Visual Odometry method (here on denoted as 6DP), as well as an extended comparison with other well known state-of-the-art methods e.g LIBVISO [14]. We compare our mixed deterministic and probabilistic egomotion estimation approach (6DP) against two well known state-of-the art egomotion estimation methods. First we evaluate 6DP linear and angular velocities raw estimates (without any type of filtering) against a 5-point algorithm. The use of a dense probabilistic approach allows to obtain better estimates of the rotation and translation up to scale factor when compared to the 5-point implementation, but exhibits an unfavorable performance in the translation scale factor (α) estimation. Afterwards we have implemented a filtering approach on top of the 6DP estimator and compared it with

LIBVISO Visual Odometry Library [14], using a standard dataset from this library. The dataset also provides ground-truth information from the fusion of IMU and GPS measurements. Results show that our method presents significant improvements in the estimation of angular velocities and a similar performance for linear velocities.

This paper is organized as follows: In Section 2 related work regarding stereo VO is presented. The 6DP algorithm implementation is detailed in Section 3. Finally Section 4 and Section 5 contain the experimental results and conclusions with final remarks.

2 Related Work

Stereo VO consists of performing egomotion estimation using as input the sequence of images acquired from a stereo camera rig rigidly attached to the vehicle or robot. One of the advantages of performing VO estimation with a stereo camera configuration is the ability to recover translation motion scale. Classical stereo VO algorithms estimate the 3D position of observed image point features by using triangulation between the left and right images. Then, relative camera motion can be calculated through the alignment of 3D feature's position between consecutive image frames.

Most of the work on stereo visual odometry methods was driven by Matthies et al. [18, 19] outstanding work on the famous Mars Rover Project. Their system was able to determine all 6-DOF of the rover (x,y,z,roll,pitch,yaw) by tracking the motion of 2D image keypoints between stereo image pairs, as well as their 3D world coordinates. Afterwards, a maximum likelihood estimation method was used to compute motion between consecutive image frames. Their method exploited robust techniques for outlier rejection such as RANSAC [6]. The Stereo VO work performed on Mars Rover Project was somewhat inspired by Olson et al. [27]. The method was developed as a replacement for wheel odometry dead reckoning methods that were not able to correctly estimate robot motion over long distances. In order to avoid large drift in robot position over time, Olson's method combined stereo egomotion estimation with absolute orientation sensor information.

Among the different approaches to compute stereo VO, two main categories have emerged in the literature, either based on their feature detection scheme or by the way motion estimation is performed. Usually, motion estimation is computed using 3D Absolute Orientation (AO) methods or Perspective-n-Point (PnP) methods. Alismail et al. [1] conducted a study on evaluating both AO and PnP methods for achieving robot pose estimation using only stereo visual odometry, and concluded that PnP methods are more accurate than AO methods. The AO methods consist on 3D triangulated points estimation for every stereo pair. Then motion estimation is solved by using point alignment algorithms like the Procrustes method [7] or Iterative-Closest- Point (ICP) method [29], such as the one used by Milella and Siegwart [20] for estimating motion of an all-terrain rover. Nister et al. [25], were one of the first to develop a PnP algorithm (3D-2D camera pose estimation), that could be computed in real-time with an outlier rejection scheme. The authors argue that minimizing the re-projection error would benefit stereo VO method accuracy. Nister et al. [24] also developed a Visual Odometry system, based on a 5-point algorithm, that became the standard algorithm for comparison of Visual Odometry techniques. This algorithm can be used either in stereo or monocular vision approaches and consists on the use of several visual processing techniques, namely: feature detection and matching, tracking, stereo triangulation and RANSAC for pose estimation with iterative refinement.

Most of stereo VO methods differ on the way stereo information is acquired and computed: sparse or dense approaches. One of the most relevant dense stereo VO applications was developed by Howard [10] for ground vehicle applications. The method does not assume prior knowledge over camera motion and so can handle very large image translations. However, due to the absence of feature detectors invariant to rotation and scaling, only works on low-speed applications and with high frame-rate, since large motions around the optical axis result in poor performance. In [21] a sparse stereo VO method is presented. A closed form solution is derived for the incremental movement of the cameras and combines distinctive features invariant to rotation and scale (SIFT)[16] with sparse optical flow (KLT) [17]. Some other authors like Ni et al. [12], minimize dependencies on feature matching and tracking algorithms by simultaneously using an

algorithm that computes feature displacement in both cameras, together with a quadrifocal setting within a RANSAC framework. Later on, the same authors [23], decoupled the rotation and translation recovery into two different estimation problems. Instead of using the three-point method, they used a RANSAC two-point algorithm for rotation recovery and a one-point method for the translation recovery.

More recently the application focus of stereo VO methods has moved from planetary rover application to the development of novel intelligent vehicles by automotive industry. Obdrzalek et al. [26] developed a voting scheme strategy for egomotion estimation, where 6-DOF problem was divided into a four dimensions problems and then decomposed in two sub-problems for rotation and translation estimation. Another influential work, is the one developed by Kitt et al. [14]. Their method, is available as an open-source visual odometry library named LIBVISO. Stereo egomotion estimation is based on image triples and the online estimation of the trifocal tensor [9]. It uses rectified stereo image sequences and produces an output 6D vector with estimated linear and angular velocities. Comport et al. [3] also develop a stereo VO method based on a different geometry estimation solution, the quadrifocal tensor. By using tensor notation, the authors can compute motion using 2D-2D image pixels matches, thus yielding a more precise motion estimation.

Stereo VO can be combined with other absolute sensor information. Rehder et al. [28] developed a stereo visual odometry method that combined visual data with GPS and IMU information. The proposed method consistently fused stereo visual odometry information with inertial measurements and sparse GPS information into a single pose estimate in real-time. Kneip et al. [15] also proposed an alternative tightly coupled approach with vision and IMU information. Their strategy for continuous robust pose computation is based on the triangulation of frame to frame point clouds when there is sufficient disparity among them.

More recently Kazik et al. [13] developed a framework that performed 6-DOF absolute scale motion with a stereo setup that copes with non-overlapping fields of view in indoor environments. It estimates monocular VO from each camera and afterwards scale is recovered by imposing the known stereo rig transformation between both cameras.

3 A Mixed Approach To Stereo Visual Odometry: Combining Sparse And Dense Methods

In this paper we propose a method to estimate the linear and angular velocities (V, W) of a vehicle equipped with a calibrated stereo vision setup. Let the images acquired by the left and right cameras of the stereo vision system in consecutive time instants be represented as the 4-tuple $\mathbf{I_{k+1}} = (I_k^L, I_k^R, I_{k+1}^L, I_{k+1}^R)$, where the subscripts k and $k+1$ denote time, and the superscripts R and L denote the right and left cameras, respectively. From point correspondences between the observations in $\mathbf{I_{k+1}}$ we can compute the rigid transformation describing the incremental motion of the setup and, thus, estimate its velocity at instant k, (V_k, W_k). Our method, denoted 6DP, combines sparse feature based methods and dense probabilistic methods [31] to compute the point correspondences between the 4-tuple of images. While feature based methods are less computational expensive and are used in real-time applications, dense correlation methods tend to be computational intensive and used in more complex applications. However, when combined with probabilistic approaches, dense methods are usually more robust and tend to produce more precise results. Therefore we developed a solution that tries to exploit the advantages of both methods.

Our 6DP method, as schematically illustrated in Fig. 2, can be roughly divided into three main steps:

- **Dense Correspondence and Egomotion estimation** In order to be able to estimate egomotion, first there is the need to compute correspondence information between images I_k and I_{k+1}, where k and $k+1$ are consecutive time instants. For egomotion estimation a variant of the dense probabilistic egomotion estimation method of [5] is used. By doing so, we establish a probabilistic correspondence between the left images at consecutive time steps, I_k^L and I_{k+1}^L, and estimate camera rotation (R) and translation ($\tilde{\mathbf{t}}$) up to a scale factor (α), thus obtaining the Essential Matrix (E) [9].
- **Sparse Keypoint and Stereo Matching** The sparse keypoint detection consists on obtaining salient features in both images a time k (I_k^L, I_k^R). To obtain the keypoints a feature detector such as the Harris corner [8] or a SIFT detector [16] is

used. The result is a set of feature points F_k^L, F_k^R that will be used in a stereo matching procedure to obtain point correspondence $P2_k$ at time k, and together with the Essential Matrix (E), correspondences $P2_{k+1}$ at time $k+1$.

– **Scale Estimation** The missing translation scale factor (α), is obtained by stereo triangulation with the point correspondences at time k and $k+1$, ($P2_k$, $P2_{k+1}$), thus obtaining corresponding point clouds $P3_k$ and $P3_{k+1}$ with point match information. Afterwards, we use an AO method like the Procrustes method [7] to obtain the best alignment between the two sets of points and determine the value of the translation scale factor (α). A RANSAC algorithm [6] is used to discard outliers in the 3D point cloud matches.
– **Kalman Filtering** To achieve a more robust egomotion estimation, we use a standard Kalman Filter approach for the linear and angular velocity estimates.

3.1 Probabilistic Correspondence

The key to the proposed method relies on a robust probabilistic computation of the epipolar geometry relating the camera's relative pose on consecutive time steps. This will speed-up and simplify the search for 3D matches on the subsequent phases of the algorithm. Given two images taken at different times, I_k and I_{k+1}, the probabilistic correspondence between point $\mathbf{x} \in R^2$ in image I_k and point $\mathbf{x}' \in R^2$ in image I_{k+1}, is defined as a belief:

$$\rho_{\mathbf{x}}(\mathbf{x}') = \text{match}(\mathbf{x}, \mathbf{x}' | I_k, I_{k+1}) \tag{1}$$

where the function match$(\cdot)$ outputs a value between 0 and 1 expressing similarity in the appearance of the two points in local neighborhoods.

Thus, all points $\mathbf{x}'$ in image 2 are candidates for matching with point $\mathbf{x}$ in image 1 with a likelihood proportional to $\rho_{\mathbf{x}}(\mathbf{x}')$. One can consider $\rho_{\mathbf{x}}$ as images (one per each pixel in image 1) whose value in $\mathbf{x}'$ is proportional to the likelihood of $\mathbf{x}'$ matching with $\mathbf{x}$. In Fig. 4, we can observe the correspondence likelihood of a point $\mathbf{x}$ in image I_k^L with all matching candidates $\mathbf{x}'$ in I_{k+1}^L. For the sake of computational cost, likelihoods are not computed for the whole range in image 2 but just on windows around $\mathbf{x}$, or suitable predictions based on prior information (see Figs. 3 and 4).

Algorithm 1 6DP Method

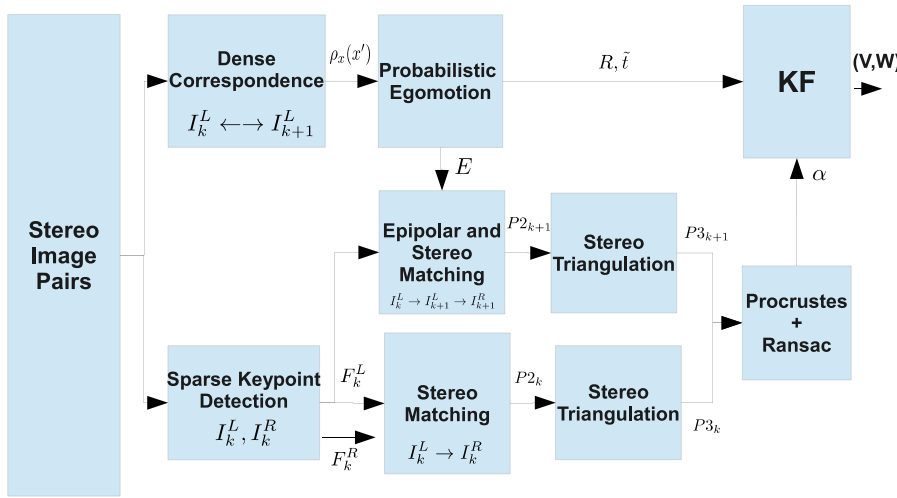
Input: 2 stereo Image pairs $\left(I_k^L, I_k^R\right)$ and $\left(I_{k+1}^L, I_{k+1}^R\right)$, E_{rig} (stereo calibration)
Output: (Velocities) V, W

Step 1. **Compute the probabilistic correspondences between images I_k^L and I_{k+1}^L, $\rho_x(x')$. Eqs. (1), (2), (3).**
Step 2. **Compute probabilistic egomotion, E. Eqs. (7), (8), (9), (10)**
Step 3. **Compute sparse keypoints in images I_k^L and I_k^R, F_k^L and F_k^R respectively. We conducted experiments using both Harris corners and Scale Invariant Features (SIFT)**
Step 4. **Perform stereo matching in between features F_k^L and F_k^R to obtain matches $P2_k$.**
Step 5. **Perform epipolar and stereo matching between images I_k^L, I_{k+1}^L and I_{k+1}^L , I_{k+1}^R, respectively, to obtain point matches $P2_{k+1}$.**
Step 6. **Stereo triangulate matches $P2_k$ and $P2_{k+1}$ to obtain corresponding point clouds $P3_k$ and $P3_{k+1}$, respectively.**
Step 7. **Perform Translation scale estimation using an Absolute Orientation method (Procrustes) to align point clouds $P3_k$ and $P3_{k+1}$. Use RANSAC to reject outliers. Eqs. (11), (12), (13).**
Step 8. **Estimate Linear and Angular Velocities , V and W Eqs. (14), (15), (16)**
Step 9. **Constant Velocity Kalman Filtering Eqs. (17) and (18)**

In [5] the probabilistic correspondence images was computed via the differences between the angle of a bank of Gabor filter responses in $\mathbf{x}$ and $\mathbf{x}'$. The motivation for using a Gabor filter bank is its robustness to changes in the brightness and contrast of the image. However, it demands a significant computational effort, thus we propose to perform the computations with the well known Zero Mean Normalized Cross Correlation function (ZNCC):

$$C_{x,y}(u,v) = \frac{\sum_{x,y \in N_W} (f(x,y) - \bar{f})(g(x+u, y+v) - \bar{g})}{\sqrt{\sum_{x,y \in N_W} (f(x,y) - \bar{f})^2} \sqrt{\sum_{x,y \in N_W} (g(x+u, y+v) - \bar{g})^2}} \tag{2}$$

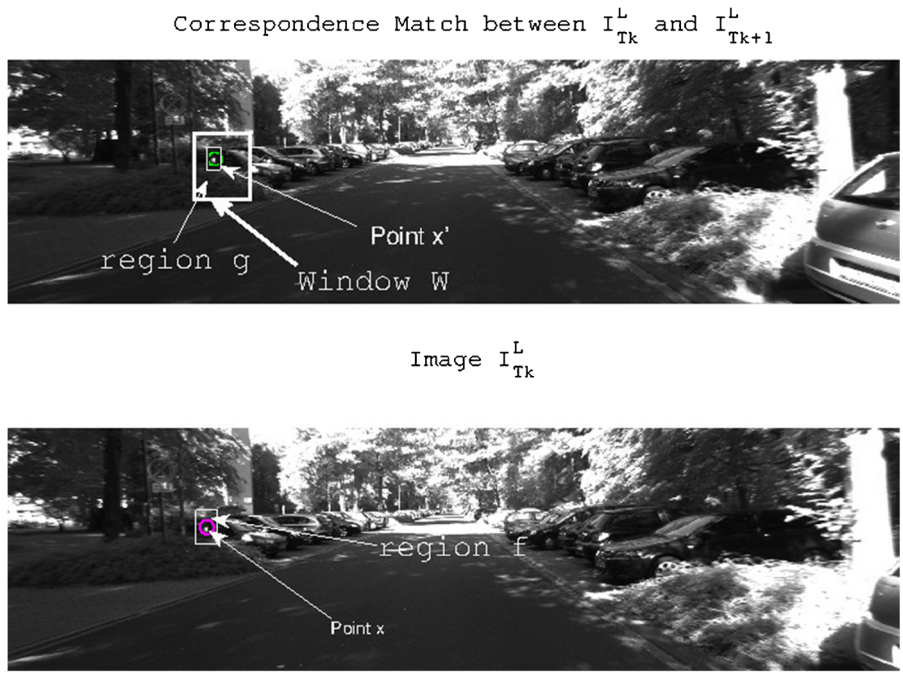
Fig. 2 6DP architecture



The ZNCC method allows to compute the correlation factor $C_{x,y}(u, v)$ between regions of two images f and g by using a correlation window around pixel $\mathbf{x} = (x, y)$ in image f and pixel **x'** = **x**+(u,v) in image g, being the correlation window size $N_W = 20$. The value $N_W = 20$ is a compromise between match quality and computational cost that we found adequate for this problem through our empirical studies, $\bar{f}$ and $\bar{g}$ are the mean values of the images in the regions delimited by the window size. This correlation factor is then transformed into a likelihood match between **x** and **x'**.

$$\rho_{\mathbf{x}}(\mathbf{x}') = \frac{C_{x,y}(u, v)}{2} + 0.5 \tag{3}$$

Fig. 3 Image feature point correspondence for ZNCC matching, with window size N_W between points **x** and $\mathbf{x}'$ represented in *red* and *green* respectively



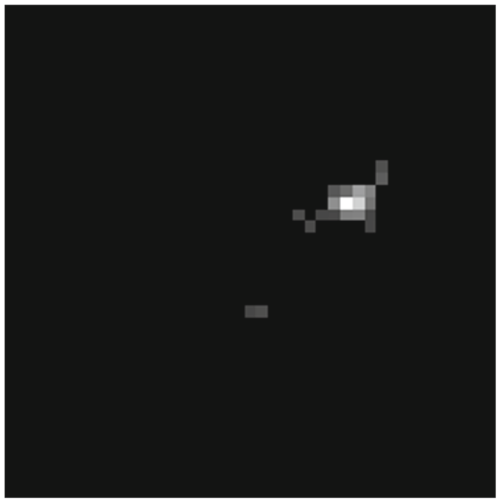


Fig. 4 Likelihood of a point $\mathbf{x}$ in image I_k^L with all matching candidates $\mathbf{x}'$ in I_{k+1}^L, for the case of Fig. 3. Points with high likelihood are represented in lighter colour

The ZNCC function is known to be robust to brightness and contrast changes and recent efficient recursive schemes developed by Huang et al. [11] render it suitable to real-time implementations. The method is faster to compute and yields similar results to the implemented by Domke [5].

3.2 Probabilistic Egomotion Estimation

From two images of the same camera, one can recover its motion up to the translation scale factor. Given the camera motion, image motion can be represented by the epipolar constraint which, in homogeneous normalized coordinates, can be written as:

$$(\tilde{\mathbf{x}}')^T E\tilde{\mathbf{x}} = 0 \tag{4}$$

where E is the so called Essential Matrix [9], a $3X3$ matrix with rank 2 and 5 degrees-of-freedom and $\tilde{\mathbf{x}}$, $\tilde{\mathbf{x}}'$ the homogeneous coordinate representations of points $\mathbf{x}$ and $\mathbf{x}'$. Given a point $\tilde{\mathbf{x}}$ in image 1, this expression constraints the points $\tilde{\mathbf{x}}'$ in image 2 to lie on line $E\tilde{\mathbf{x}}$, thus it expresses the loci in image 2 that should be searched for matches of points in image 1. It can be factored by:

$$E = R\left[\tilde{\mathbf{t}}\right]_\times \tag{5}$$

where R and $\tilde{\mathbf{t}}$ are, respectively, the rotation and translation direction of the camera between the two frames, with $\tilde{\mathbf{t}}_\times$ the skew symmetric representation of $\tilde{\mathbf{t}}$, as defined in the following expression:

$$\tilde{t}_\times = \begin{bmatrix} 0 & -\tilde{t_z} & \tilde{t_y} \\ \tilde{t_z} & 0 & -\tilde{t_x} \\ -\tilde{t_y} & \tilde{t_x} & 0 \end{bmatrix} \tag{6}$$

To obtain the Essential matrix from the probabilistic correspondences, [5] proposes the computation of a probability distribution over the 5-dimensional space of essential matrices. Each dimension of the space is discretized in 10 bins, thus leading to 100000 hypotheses E_i. For each point $\mathbf{x}$ the likelihood of these hypotheses is evaluated by:

$$\rho(E_i|\mathbf{x}) \propto \max_{(\tilde{\mathbf{x}}')^T E_i \tilde{\mathbf{x}}=0} \rho_{\mathbf{x}}(\mathbf{x}') \tag{7}$$

Intuitively, for a single point $\mathbf{x}$ in image 1, the likelihood of a motion hypothesis is proportional to the likelihood of the best match obtained along the epipolar line generated by the essential matrix. After the dense correspondence probability distribution has been computed for all points, the method [5] computes a probability distribution over motion hypotheses represented by the epipolar constraint. Assuming statistical independence between the measurements obtained at each point the overall likelihood of a motion hypothesis is proportional to the product of the likelihoods for all points:

$$\rho(E_i) \propto \prod_{\mathbf{x}} \rho(E_i|\mathbf{x}) \tag{8}$$

Finally, having computed all the motion hypotheses, a Nelder-Mead simplex method [22] is used to refine the motion estimate around the highest scoring samples E_i. The Nelder-Mead simplex method is a local search method for problems whose derivatives are not known. The method was already applied in [5] to search for the local maxima of likelihood around the top ranked motion hypotheses:

$$E_i^* = arg \max_{E_i+\delta E} \rho(E_i + \delta E) \tag{9}$$

where δE are perturbations to the initial solution E_i computed by the Nelder-Mead optimization procedure.

Then, the output of the algorithm is the solution with the highest likelihood

$$E^* = \max_i E_i^* \tag{10}$$

3.3 Scale Estimation

By using the previous method, we compute the 5D transformation $(R, \tilde{\mathbf{t}})$ between the camera frames at times k and $k+1$. However, $\tilde{\mathbf{t}}$ does not contain translation scale information. This type of information, will be calculated by an Absolute Orientation(AO) method like the Procrustes method.

Once the essential matrix between images I_k^L and I_{k+1}^L has been computed by the method described in the previous section, we search along the epipolar lines for matches F_{k+1}^L in I_{k+1}^L to the features F_k^L computed in I_k^L, as displayed in Fig. 5.

Then, these matches are propagated to I_{k+1}^R by searching along horizontal stereo epipolar lines for matches F_{k+1}^R. From stereo triangulation we compute 3D point clouds at instant k and $k+1$, respectively $P3_k$ and $P3_{k+1}$, with known point correspondence. Points whose matches are unreliable or were not found are discarded from the point clouds.

3.3.1 Procrustes Analysis and Scale Factor Recovery

The Procrustes method allows to recover rigid body motion between frames through the use of 3D point matches, obtained in the previous steps

$$\mathbf{P3^i_{k+1}} = R'\mathbf{P3^i_k} + \mathbf{t}' \tag{11}$$

where i is a point cloud element.

In order to estimate the motion $[R', \mathbf{t}']$, a cost function that measures the sum of squared distances between corresponding points is used.

$$c^2 = \sum_i^n \left\| \mathbf{P3_{k+1}} - (R'\mathbf{P3_k} + \mathbf{t}') \right\|^2 \tag{12}$$

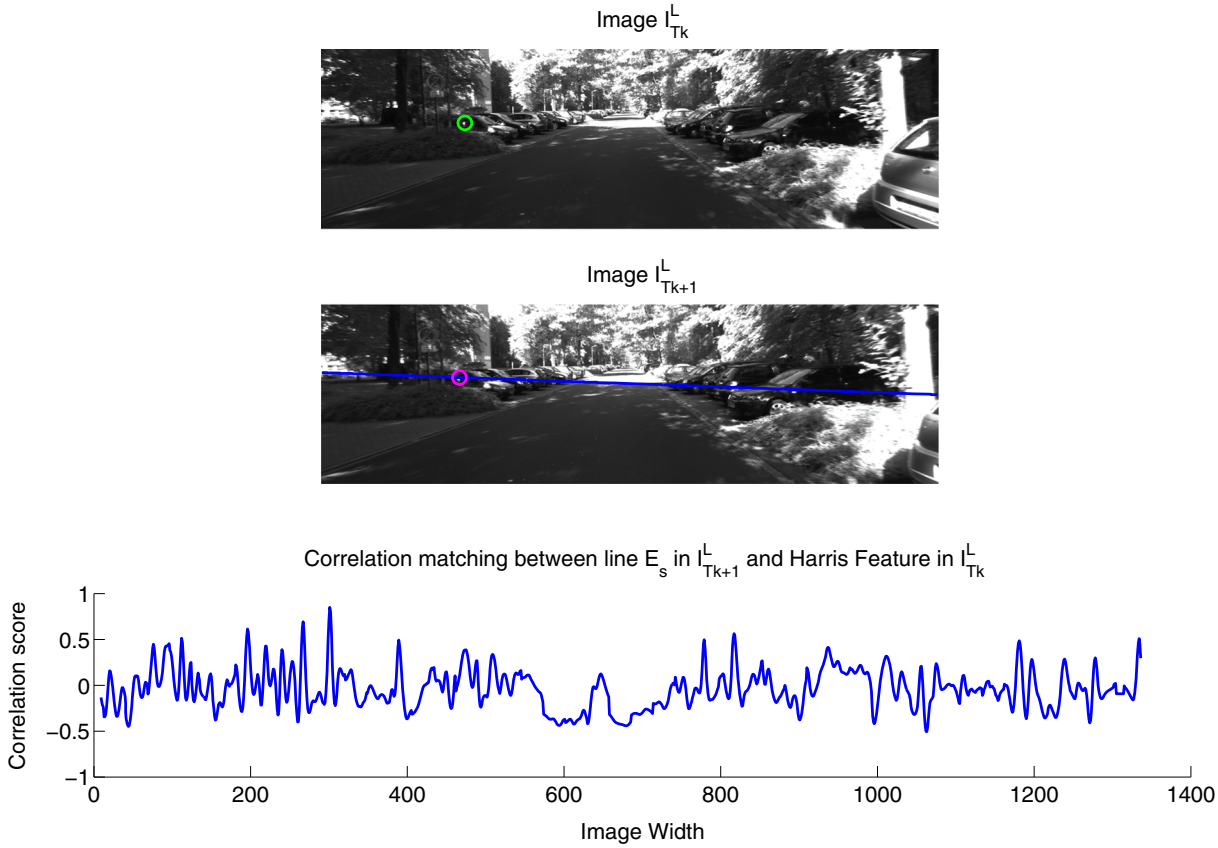


Fig. 5 Image feature point marked in colour green in image I_k^L lies in the epipolar line (blue) estimated between I_k to I_{k+1}. The point with higher correlation score, marked in red in image I_{k+1}^L is chosen as the matching feature point

Performing minimization of Eq. (12) is possible to estimate $[R', \mathbf{t}']$. However these estimates are only used to obtain the missing translation scale factor α, since rotation (R) and translation direction $(\tilde{\mathbf{t}})$ were already obtained by the probabilistic method. Although conceptually simple, some aspects regarding the practical implementation of the Procrustes method must be taken into consideration. Namely, since this method is very sensible to data noise, obtained results tend to vary in the presence of outliers. To overcome this difficulty, RANSAC [6] is used to discard possible outliers within the set of matching points.

3.3.2 Bucketing

For a correct motion scale estimation, it is necessary to have a proper spatial feature distribution through out the image. For instance, if the Procrustes method uses all obtained image feature points without having their image spatial distribution into consideration, the obtained motion estimation $[R', \mathbf{t}']$ between two consecutive images could turn out biased. To avoid having biased samples in the RANSAC phase of the algorithm a bucketing technique [32] is implemented to assure a balanced image feature distribution sample. In Fig. 6 a possible division of the image is displayed. The image region is divided into $L_x \times L_y$ buckets, based on minimum and maximum coordinates of the feature points. Afterwards, image feature points are classified as belonging to one of the buckets. In case a bucket does not contain any feature, it will be disregarded. The bucket size must be previously defined: in our case we divided the image into a 8×8 buckets. Assuming we have l buckets, the interval between [0...1] is divided into l intervals such that the width (i^{th}) of each interval is defined as $n_i / \sum_i n_i$, where n_i is the number of matches assigned to the i^{th} bucket. The bucket selection procedure, consists on retrieving a number using a uniform random generator in the interval [0...1]. The number that falls in the i^{th} interval, gives origin to the i^{th} bucket being selected. Finally, we select a random point of the selected i^{th} bucket.

3.4 Linear and Angular Velocity Estimation

To sum up the foregoing, we determine camera motion up to a scale factor using a probabilistic method, and by adding stereo vision combined with Procrustes estimation method, we are able to determine the missing motion scale α:

$$\alpha = \frac{\|\mathbf{t}'\|}{\|\tilde{\mathbf{t}}\|} \tag{13}$$

Then, the instantaneous linear velocity is given by:

$$V = \frac{\alpha \tilde{\mathbf{t}}}{\Delta T} \tag{14}$$

where ΔT is the sampling interval:

$$\Delta T = T_{k+1} - T_k \tag{15}$$

Likewise, the angular velocity is computed by:

$$W = \frac{r}{\Delta T} \tag{16}$$

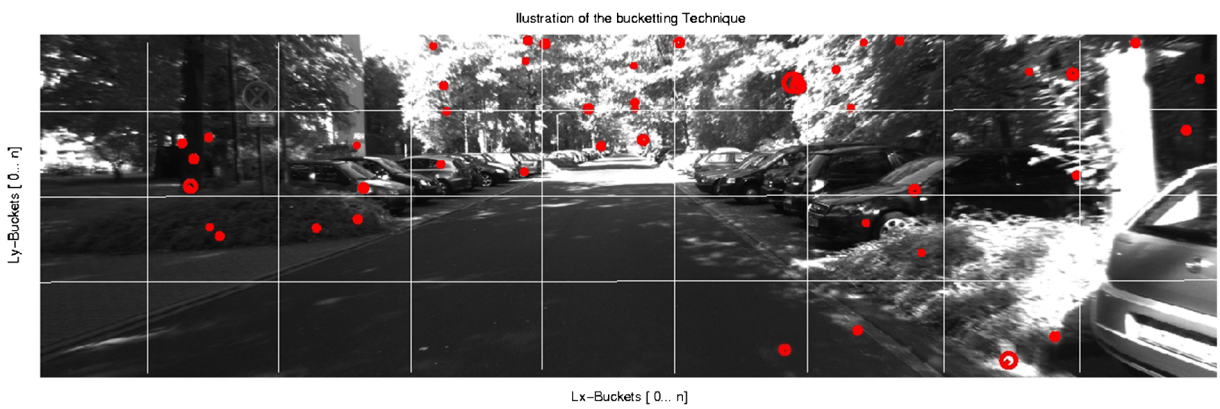


Fig. 6 Feature detection bucketing technique used to avoid biased samples in the RANSAC method stage. The image is divided in buckets where feature points are assigned to and pulled according to the bucket probability

where $r = \theta u$, the angle-axis representation of the incremental rotation R [4].

Thus, using motion scale information given by the Procrustes method, we can estimate vehicle linear velocity between instants k and $k+1$. The AO orientation method is only used for linear velocity estimation (motion scale). For the angular velocity estimation we use the rotation matrix R calculated by Domke's probabilistic method, that is more accurate than the rotation obtained by the AO method.

3.5 Kalman Filter

In order to achieve a more robust estimation, we also use a Kalman filter used to filter the linear and angular velocity estimates having state equation $X = [V, W]^T$, where V is the vehicle linear velocity, W is the vehicle angular velocity. The constant velocity Kalman filter [7] considers a state transition model with zero-mean stochastic acceleration:

$$X_k = F X_{k-1} + \xi_k \tag{17}$$

where the state transition matrix is the identity matrix, $F = I_{6x6}$, and the stochastic acceleration vector ξ_k is distributed according to a multivariate zero-mean Gaussian distribution with covariance matrix Q, $\xi_k \sim \mathcal{N}(0,Q)$. The observation model considers state observations with additive noise:

$$Y_k = H X_k + \eta_k \tag{18}$$

where the observation matrix H is identity, $H = I_{6x6}$, and the η_k measurement noise is zero-mean Gaussian with covariance R.

We set the covariance matrices Q and R empirically, according to our experiences, to:

$$Q = \text{diag}(q_1, \cdots, q_6) \tag{19}$$

$$R = \text{diag}(r_1, \cdots, r_6) \tag{20}$$

where $q_i = 10^{-3}, i = 1, \cdots, 6$, $r_3 = 10^{-3}$ and $r_i = 10^{-4}, i \neq 3$.

The r_3 differs from the other (r) measurement noises values, due to the fact that it corresponds to the translation on the z axis which is inherently noisier due to the uncertainty of the t_z estimates in the stereo triangulation step.

4 Results

In this section, we present results of 3 implementations of the 6DP method. The first experiment compares 6DP raw estimates using the Harris corner detector [8] as the sparse feature detector, here on denoted as 6DP-raw-Harris and compares it against a native 5-point implementation. Afterwards, we present results of the other 2 implementations: (i) 6DP-raw-SIFT where we replaced the Harris corner for a more robust and invariant to scale detector (SIFT)[16]; (ii) 6DP-KF that also uses SIFT features but this time integrated in a Kalman Filter framework. The results of both implementations are compared with the state-of-the art visual odometry estimation method LIBVISO [14] using their dataset reference (2009-09-08-drive-0021).

4.1 Computational Implementation

The code used to compute 6DP was written in MATLAB as a proof of concept, without using any kind of code optimization. The experiments were performed using an Intel I5 Dual Core 3.2 GHz. For the evaluation we used a section of the dataset [14] reference (2009-09-08-drive-0021), which has been used for benchmarking visual odometry methods in other works against which we compare our method. During our experiments several parts of the dataset were tried and results were consistent across the dataset. The dataset images have resolution of 1344 $\times$ 391, which consumes a considerable amount of computational and memory resources ($\sim$ 0.5MB per point) making unfeasible the computation of all image points using the Matlab implementation on standard CPU hardware. Thus, the results shown in this paper were obtained using 1000 randomly selected points in image I_k^L. The method takes about 12 sec per image pair. Most of time is consumed in the first stage of the implementation, with the dense probabilistic correspondences and the motion up to a scale factor estimates. The recursive ZNCC approach allowed to reduce Domke Gabor Filter processing time by 20 %.

Even so, the approach is feasible and can be implemented in real-time for use on mobile robotics applications. The main option is to develop a GPGPU version of the method. Since the method deals with multiple hypothesis of correspondence, and motion, it is suitable to be implemented into parallel hardware.

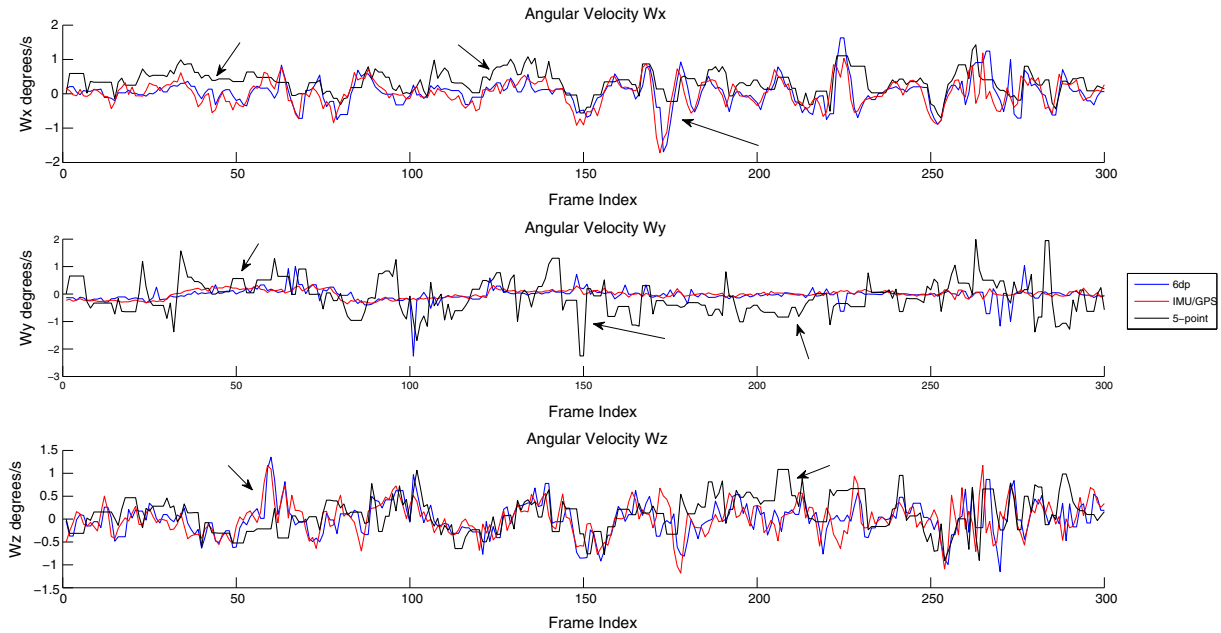


Fig. 7 Comparison of angular velocity estimation results between IMS/GPU(*red*), 6DP-raw-Harris measurements (*blue*) and a native 5-point implementation (*black*). The obtained 6DP-raw-Harris measurements are similar to the data estimated by the IMU/GPS, contrary to the 5-point implementation that has some periods of large errors (e.g. the regions indicated with arrows in the plots)

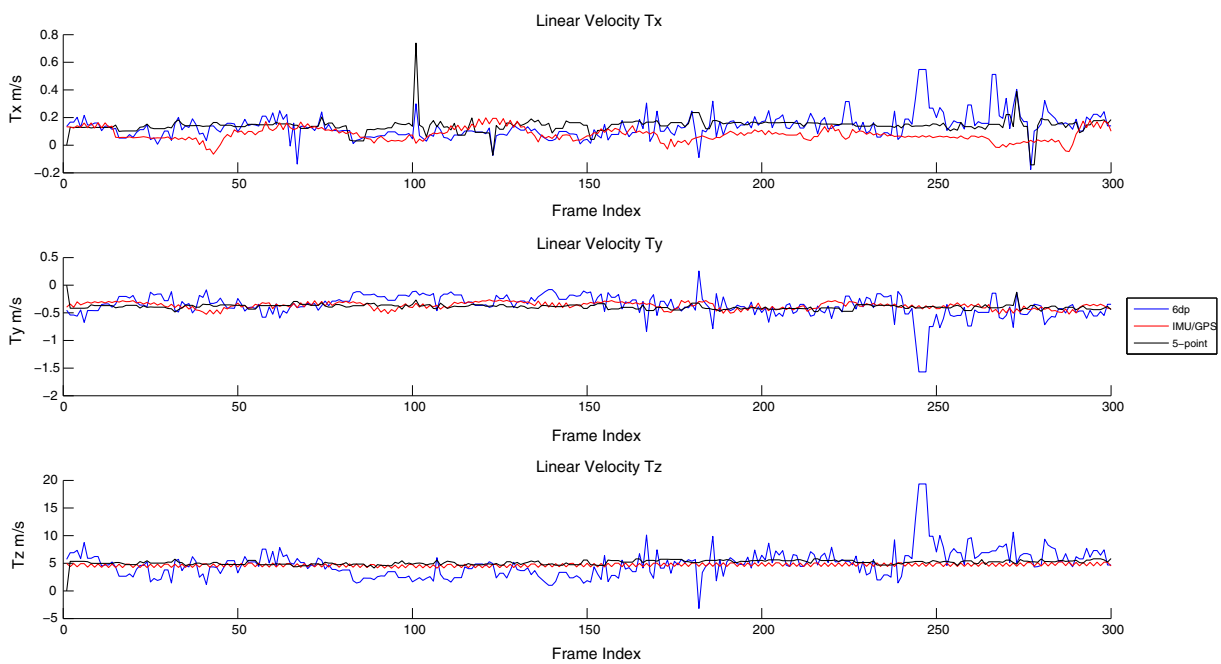


Fig. 8 Comparison of linear velocity estimation results, where the 5-point implementation (*black*) exhibits a closer match to the IMU/GPS information (*red*). The 6DP-raw-Harris method (*blue*) displays some highlighted outliers due to the use of the Harris feature detection matching in the sparse method stage

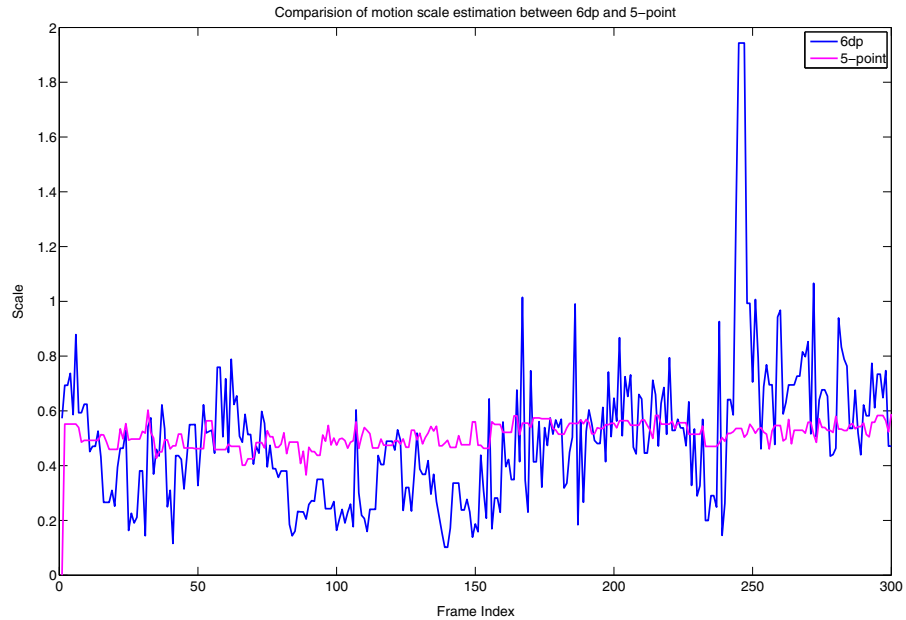


Fig. 9 Translation scale factor comparison between 5-point and 6DP-raw-Harris, where the 5-point method exhibits a more constant behavior for the translation scale factor estimation

4.2 6DP-Raw-Harris vs 5-Point

In this section, one can observe results comparing our approach versus the 5-point RANSAC algorithm [24]. Linear and angular velocities estimation results are presented in the camera reference frame.

In Fig. 7, one can observe the angular velocity estimation of the 6DP method, IMU/GPS information and the 5-point RANSAC. We also show the Inertial Navigation System data (IMU/GPS OXTS RT 3003), which is considered as "ground-truth" information. The displayed results demonstrate a high degree of

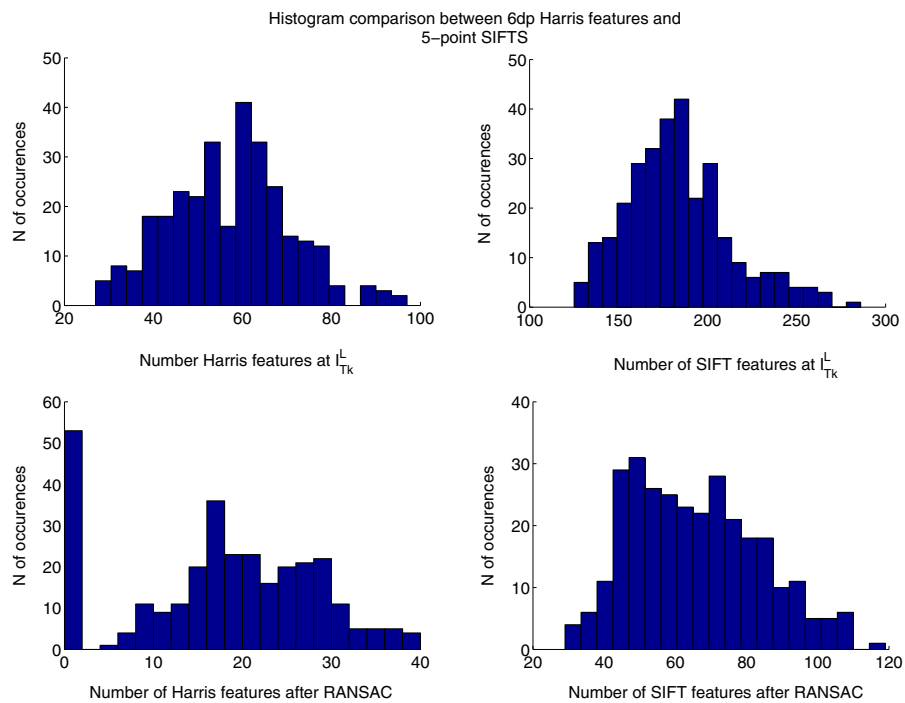


Fig. 10 Number of Features at different steps of 6DP-raw-Harris and 5-point. SIFT features display a more robust matching behavior between images. Contrary to Harris Corners, most of the SIFTS are not eliminated in the RANSAC stage

similarity between performance obtained using 6DP and IMU/GPS information. Results obtained by 6DP were performed without using any type of filtering technique, thus the display of one or two clear outliers. Most importantly, when it comes to angular velocities estimation, the 6DP method performance is better than the performance exhibited by the 5-point RANSAC algorithm.

However, for linear velocities as displayed in Fig. 8, the 5-point RANSAC algorithm implementation performance is smoother than our 6DP approach, especially on the Z axis. As shown in Fig. 10, the 5-point algorithm contains more image features when performing Procrustes Absolute Orientation method (after RANSAC) which may also explain the higher robustness in motion scale estimation in Fig. 9, where the 5-point algorithm displays a constant translation scale value.

The results demonstrate complementary performances, one more suitable for linear motion estimation and the other more suitable for angular motion estimation.

4.3 6DP-Raw-Harris vs 6DP-Raw-SIFT

The obtained results using 6DP-raw-Harris in the translation scale (α) estimation were not sufficiently accurate, mostly due to the use of the Harris corner

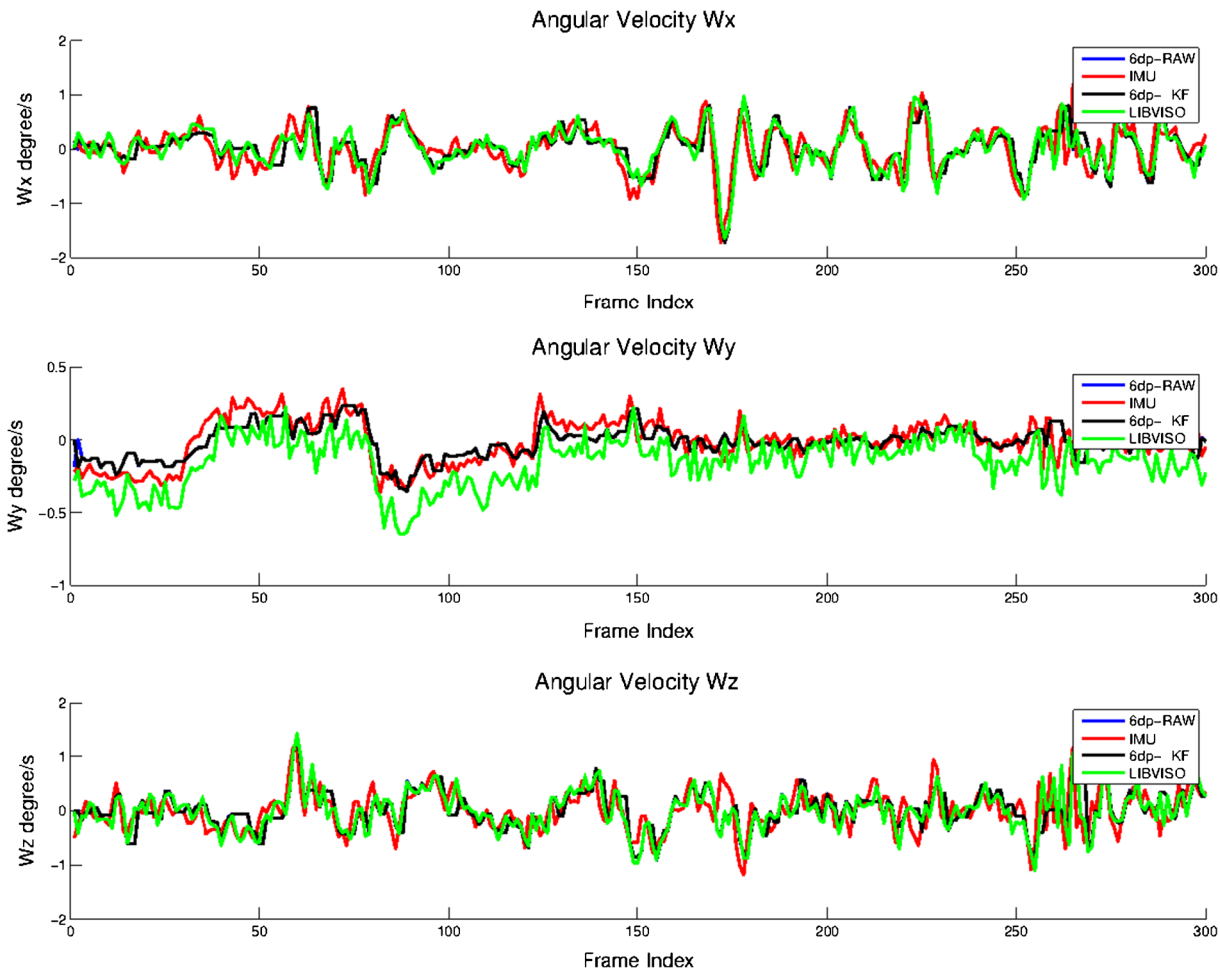


Fig. 11 Results for angular velocities estimation between IMU/GPS information (*red*), raw 6DP measurements 6DP-raw-SIFTS (*blue*), filtered 6DP measurements 6DP-KF (*black*), and 6D Visual Odometry Library LIBVISO (*green*). Even though all exhibit similar behaviors the filtered implementation 6DP-KF is the one which is closer to the "ground truth" IMU/GPS measurements (see also Table 1)

Table 1 Standard Mean Squared Error between IMU and Visual Odometry (LIBVISO and 6DP-KF)

	V_x	V_y	V_z	W_x	W_y	W_z	$\|\|V\|\|$	$\|\|W\|\|$
LIBVISO	0.0674	0.7353	0.3186	0.0127	0.0059	0.0117	1.1213	0.0303
6DP-KF	0.0884	0.0748	0.7789	0.0049	0.0021	0.0056	0.9421	0.0126

The displayed results show a significant improvement of the 6DP-KF method performance specially in the angular velocities estimation case

detector. We modified the 6DP method, by replacing the Harris corner feature detector [8] for the more robust and invariant to rotation and scale SIFT detector [16]. We can observe in Fig. 10 that SIFT features are more stable after the RANSAC step when compared to the Harris corner approach, and thus can provide more accurate point correspondence between I_k^L and I_{k+1}^L.

4.4 6DP-KF vs LIBVISO

To illustrate the performance of the 6DP-KF method, we compared our system performance against LIBVISO [14], which is a standard library for computing 6-DOF visual Odometry. We also compared our performance against IMU/GPS acting as ground truth

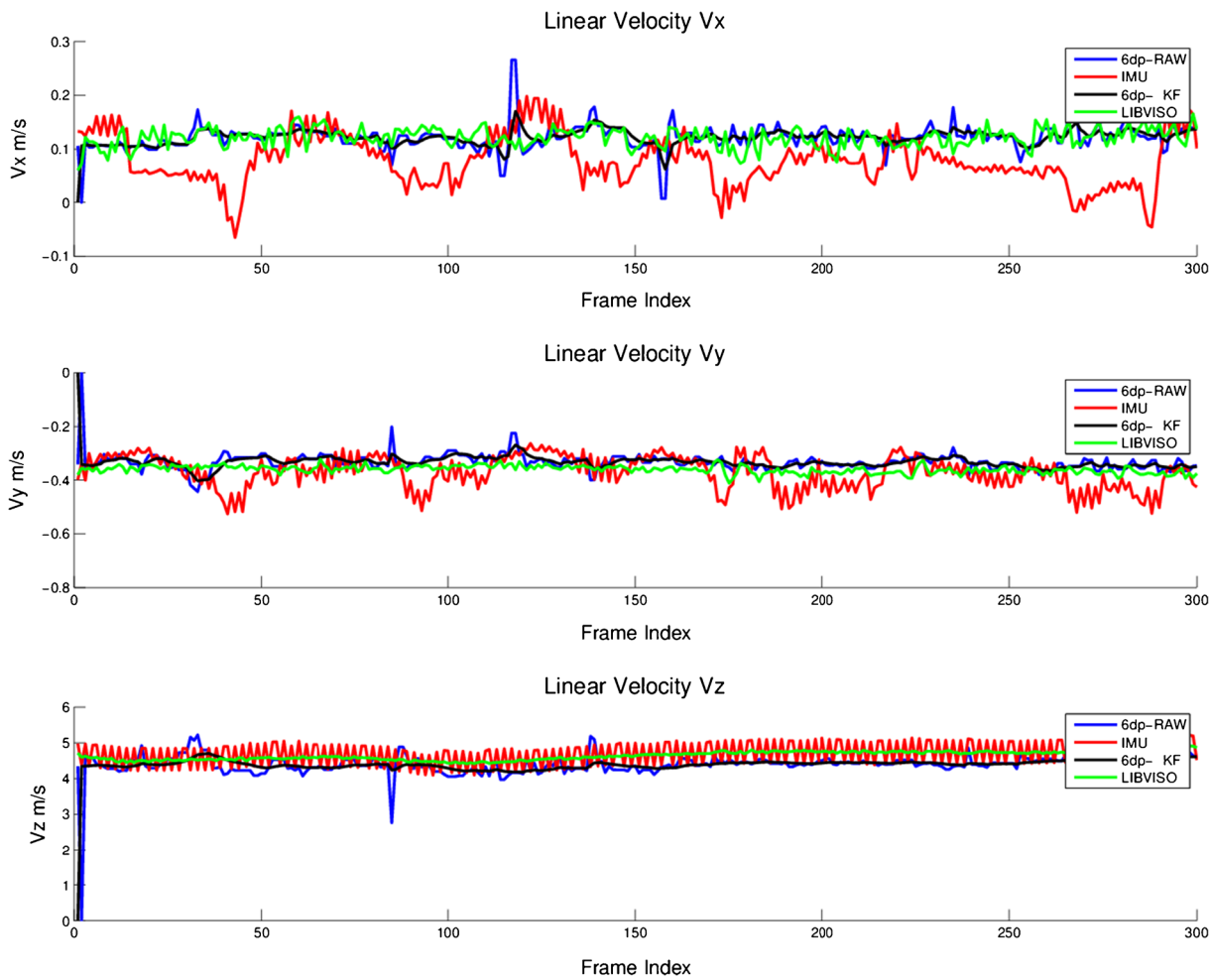


Fig. 12 Results for linear velocities estimation, where the LIBVISO implementation and 6DP-KF display similar performance when compared to IMU/GPS performance

information using the same Kitt et al. [14] Karlsruhe dataset sequences.

In Fig. 11 one can observe angular velocity estimation from both IMU/GPS and LIBVISO, together with 6DP-raw-SIFT and 6DP-KF filtered measurements. All approaches obtained results consistent with the IMU/GPS, but the 6DP-KF displays a better performance in what respects the angular velocities. These results are stated in Table 1, where root mean square error between IMU/GPS, LIBVISO and 6DP-KF estimation are displayed. The 6DP-KF method shows 50 % lower error than LIBVISO for the angular velocities estimation.

Although not as good as for the angular velocities, the 6DP-KF method also displays a better performance in obtaining linear velocity estimates as displayed in Fig. 12 and in Table 1. Overall, our 6DP-KF shows an important precision improvement over LIBVISO.

5 Conclusions and Future Work

In this work, we developed a novel method of stereo visual odometry using sparse and dense egomotion estimation methods. We utilized dense egomotion estimation methods for estimating the rotation and translation up to scale and then complement the method with the use of a sparse feature approach for recovering the scale factor.

First, we compared the raw estimates of our 6DP-raw-Harris algorithm against a native 5-point implementation without any type of filtering. The results obtained proved that 6DP-raw-Harris performed better in the angular velocities estimation but compared unfavorably in the linear velocities estimation due to lack of robustness in the translation scale factor(α) estimation. On a second implementation, we replaced the Harris feature detector with the more robust SIFT detector, implemented a Kalman filter on top of the raw estimates and tested the proposed algorithm against a state-of-the-art 6D visual Odometry Library such as LIBVISO. The presented results demonstrate that 6DP-KF performs more accurately when compared to other techniques for stereo VO estimation, yielding robust motion estimation results, most notably in the angular velocities.

The benefits of using dense probabilistic approaches are thus tested and validated in a real world scenario with practical significance. Despite more computational intensive, dense methods produce more accurate results than feature based methods and are a competitive alternative to stereo egomotion computation.

To overcome increased computational cost one should, in future work, explore their potential implementation in parallel hardware such as a GPU.

Acknowledgments This work is financed by Project "NORTE-07-0124-FEDER-000060" of the North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia within project FCT project [PEst-OE/EEI/LA0009/2013]" and under grant SFRH / BD / 47468 / 2008

References

1. Alismail, H., Browning, B., Dias, M.B.: Evaluating pose estimation methods for stereo visual odometry on robots. In: Proceedings of the 11th International Conference on Intelligent Autonomous Systems (IAS-11) (2010)
2. Bonin-Font, F., Ortiz, A., Oliver, G.: Visual navigation for mobile robots: a survey. J. Intell. Robot. Syst. **53**, 263–296 (2008)
3. Comport, A., Malis, E., Rives, P.: Real-time quadrifocal visual odometry. Int. J. Robot. Res. **29**(2–3), 245–266 (2010)
4. Craig, J.J.: Introduction to Robotics: Mechanics and Control, 2nd edn. Addison-Wesley Longman Publishing Co., Inc., Boston (1989)
5. Domke, J., Aloimonos, Y.: A probabilistic notion of correspondence and the epipolar constraint. In: 3rd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06), pp. 41–48. IEEE (2006)
6. Fischler, M.A., Bolles, R.C.: Random sample consensus a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
7. Goodall, C.: Procrustes methods in the statistical analysis of shape. J. R. Stat. Soc. Ser. B Methodol. **53**(2), 285–339 (1991)
8. Harris, C., Stephens, M.: A combined corner and edge detection. In: Proceedings of the 4th Alvey Vision Conference, pp. 147–151 (1988)
9. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press. ISBN: 0521540518 (2004)
10. Howard, A.: Real-time stereo visual odometry for autonomous ground vehicles. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008, pp. 3946–3952. IEEE (2008)

11. Huang, J., Zhu, T., Pan, X., Qin, L., Peng, X., Xiong, C., Fang, J.: A high-efficiency digital image correlation method based on a fast recursive scheme. Meas. Sci. Technol. **21**(3) (2011)
12. Kai, N., Dellaert, F.: Stereo tracking and three-point/one-point algorithms - a robust approach. In: Visual Odometry, International Conference on Image Processing (ICIP), pp. 2777–2780 (2006)
13. Kazik, T., Kneip, L., Nikolic, J., Pollefeys, M., Siegwart, R.: Real-time 6d stereo visual odometry with non-overlapping fields of view. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1529–1536 (2012)
14. Kitt, B., Geiger, A., Lategahn, H.: Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In: IEEE Intelligent Vehicles Symposium (IV), pp. 486–492. IEEE (2010)
15. Kneip, L., Chli, M., Siegwart, R.: Robust real-time visual odometry with a single camera and an imu. In: Proceedings of the British Machine Vision Conference (BMVC) (2011)
16. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
17. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. pp. 674–679 (1981)
18. Maimone, M., Matthies, L., Cheng, Y.: Visual odometry on the Mars exploration rovers. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 903–910. IEEE (2005)
19. Maimone, M., Matthies, L., Cheng, Y.: Two years of visual odometry on the mars exploration rovers: Field reports. J. Field Robot. **24**(3) (2007)
20. Milella, A., Siegwart, R.: Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In: IEEE International Conference on Computer Vision Systems, pp. 21 (2006)
21. Moreno, F., Blanco, J., González, J.: An efficient closed-form solution to probabilistic 6D visual odometry for a stereo camera. In: Proceedings of the 9th International Conference on Advanced Concepts for Intelligent Vision Systems, pp. 932–942. Springer-Verlag (2007)
22. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**(4), 308–313 (1965). doi:10.1093/comjnl/7.4.308
23. Ni, K., Dellaert, F., Kaess, M.: Flow separation for fast and robust stereo odometry. In: IEEE International Conference on Robotics and Automation, ICRA 2009, vol. 1, pp. 3539–3544 (2009)
24. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Mach. Intell. **26**, 756–777 (2004)
25. Nistér, D., Naroditsky, O., Bergen, J.: Visual odometry for ground vehicle applications. J. Field Robot. **23**(1), 3–20 (2006)
26. Obdrzalek, S., Matas, J.: A voting strategy for visual ego-motion from stereo. In: 2010 IEEE Intelligent Vehicles Symposium, pp. 382–387
27. Olson, C., Matthies, L., Schoppers, M.: Maimone, M.: Rover navigation using stereo ego-motion. Robot. Auton. Syst. **43**, 215–229 (2003)
28. Rehder, J., Gupta, K., Nuske, S.T., Singh, S.: Global pose estimation with limited gps and long range visual odometry. In: IEEE Conference on Robotics and Automation (2012)
29. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: 3rd International Conference on 3D Digital Imaging and Modeling (3DIM) (2001)
30. Scaramuzza, D., Fraundorfer, F.: Visual odometry tutorial, part i. Robot. Autom. Mag. IEEE **18**(4), 80–92 (2011)
31. Silva, H., Bernardino, A., Silva, E.: Combining sparse and dense methods for 6d visual odometry. In: 13th IEEE International Conference on Autonomous Robot Systems and Competitions. Lisbon (2013)
32. Zhang, Z., Deriche, R., Faugeras, O., Luong, Q.T.: A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Artif. Intell. Spec. Vol. Comp. Vis. **78**(2), 87–119 (1995)