On Feature Parameterization for EKF-based Monocular SLAM

Simone Ceriani * **Daniele Marzorati** ** **Matteo Matteucci** *
Davide Migliore *** **Domenico G. Sorrenti** ****

* *Politecnico di Milano, Italy (e-mail: {ceriani,matteucci}@elet.polimi.it)*
** *Info Solution S.p.A., Italy (e-mail: d.marzorati@infosolution.it)*
*** *IDSIA, Galleria 1, Manno, Switzerland (e-mail: davide@idsia.ch)*
**** *Univ. di Milano - Bicocca, Italy (e-mail: sorrenti@disco.unimib.it)*

Abstract: In the last years, the Monocular SLAM problem was widely studied, to allow the simultaneous reconstruction of the environment and the localization of the observer, by using a single camera. As for other SLAM problems, a frequently used feature for the representation of the world, is the 3D point. Differently from other SLAM problems, because of the perspective model of the camera, in Monocular SLAM, features cannot be completely perceived and initialized from a single measurement. To solve this issue, different parameterizations have been proposed in the literature, which try to solve also another problem in Monocular SLAM, i.e., the distortion of the Gaussian uncertainty in depth estimation that takes place because of the non-linear measurement model. In this paper, we start from recent results in consistency analysis for these parameterizations to propose a novel approach to improve EKF-based Monocular SLAM even further. Our claims are sustained by an extended validation on simulated and real data.

Keywords: Robot vision, Robot navigation, Computer vision, Parametrization, Extended Kalman filters.

1. INTRODUCTION

In the last years, one of the most studied problems in Robotics has been the navigation of a mobile robot in an unknown environment, this is a consequence of the limited number of practical situation that can be tackled by means of structuring the environment, so to ease navigation. The problem is called Simultaneous Localization And Mapping (SLAM), to represent the case where the observer, at each time, uses the information obtained from one or more sensors to estimate simultaneously the environment map and the robot position. More recently, beginning with the work of Davison (2003), we assisted to the diffusion of approaches that were trying to solve the problem by basing only on one single camera; this problem is known as Monocular SLAM. The importance of such problem stems from the fact that a camera is a lightweight and cheap device, both in cost and power consumption, which is a critical aspect in mobile systems.

The monocular SLAM solution in Davison (2003) demonstrated that it is possible to solve the SLAM problem in real time, using only the sequence of images acquired by an hand-held camera, and subsequent works showed quite impressive results also when applied to related fields such as augmented reality (Castle et al., 2010) or 3D dense reconstruction (Newcombe and Davison, 2010) and (Mei et al., 2010). Notwithstanding such advancements, open issues remain, and one particularly requires, in our opinion, a more detailed analysis: 3D feature parametrization and their corresponding initialization. The problem is due to the fact that, from a single image, we cannot estimate the 3D position of one point, as it is possible to perceive only its bearing. In this situation, which is typical of Monocular SLAM, the 3D feature estimate is initially affected by uniform uncertainty in depth, i.e., 3D features are equally likely on the entire viewing ray.

In the realm of Bayesian Filtering, Davison (2003) proposed to use a Kalman filter (in its Extended version) in combination with a non-parametric representation of the uncertainty on the depth of the 3D feature. That proposal required to measure a given feature a certain number of times, so to allow enough parallax because of the observer motion, to modify depth uncertainty into a quasi-normal distribution. Subsequently, the initialization of the 3D feature in the EKF can take place. This approach leads to a delayed initialization of the feature and a delayed use of this information, exactly when it is mostly useful.

To overcome the delayed initialization problem, Solà et al. (2005) proposed a different approach, introducing an undelayed initialization based on the idea of representing depth uncertainty with a Mixture of Gaussians. The solution needs to maintain in the filter multiple depth hypoteses, which affects the performance of the EKF. Subsequently Montiel et al. (2006) demonstrated that it is possible to represent this uncertainty using a single Gaussian by changing the feature representation and adopting an inverse-depth parametrization.

Nowadays, to the best of our knowledge, the problem of undelayed single feature initialization and parametrization in EKF-based Monocular SLAM has three main solutions, all related to how the 3D points are represented in the filter state: the Unified Inverse Depth (UID) representation (Montiel et al., 2006), the Inverse Scaling (IS) representation (Marzorati et al., 2009), and the Anchored Homogeneous Point (AHP) representation (Solà, 2010). All these parametrizations allow to represent depth uncertainty skewed and extending up to infinity (Hartley and Zisserman, 2004) by using a standard Gaussian distributions, and, at the same time, reducing the non-linearity of the measurement equation. The aim of this paper is to review this state of the art and present a fourth solution to this problem, namely the Framed Homogenous Point (FHP), that extends the AHP parametrization and allows a better consistency of the EKF used in Monocular SLAM.

The rest of this article is organized as follow: in Section 2 the EKF-based Monocular SLAM is briefly resumed, in Section 3 the three known parametrization are presented toghether with the new one. In Section 4 experimental results obtained with a simulator and real images will be presented, and in Section 5 conclusion and future works are presented.

2. EKF-SLAM WITH A SINGLE CAMERA

The Extended Kalman Filter is a well known method used to solve the SLAM problem. The joint distribution of the camera pose and 3D features is recursively estimated in the form of a multidimensional Gaussian distribution. In general, the filter state could be subdivided in three parts:

- the camera frame (i.e., position and orientation) in the world [1] reference frame $\mathbf{\Gamma}_t$;
- (optionally) the parameters of motion $\mathbf{\Lambda}_t$ (e.g., tangential and rotational speed);
- the map of the environment $\mathbf{Y}_t$.

The camera frame $\mathbf{\Gamma}_t$ is composed by a translational part and a rotational part, with the angles coded with quaternions [2]:

$$\mathbf{t}_t = \left[\, t_{x_t} \;\; t_{y_t} \;\; t_{z_t} \,\right]^T \quad \mathbf{q}_t = \left[\, q_{w_t} \; q_{x_t} \; q_{y_t} \; q_{z_t} \,\right]^T. \qquad (1)$$

Without loss of generality, we can consider as camera frame another reference frame that is linked to the real camera frame by a known transformation; e.g., this might apply on a mobile robot, where the camera frame is in a known pose w.r.t. the robot odometric frame.

The map of the environment is represented by a set of features, which are measurable elements in the environment: $\mathbf{Y}_t = \{\mathbf{y}_i\}$. Each feature is represented with a parametrization that is the main topic of the following section.

The state is thus represented as:

$$\mathbf{X}_t = \left[\, \mathbf{\Gamma}_t^T \;\; \mathbf{\Lambda}_t^T \;\; \mathbf{y}_{1t}^T \;\; \mathbf{y}_{2t}^T \;\; \dots \;\; \mathbf{y}_{nt}^T \,\right]^T. \qquad (2)$$

The prediction step of the Kalman filter is $\hat{\mathbf{X}}_{t+1} = f(\mathbf{X}_t, \mathbf{u}_t)$, and aims at the prediction of the next system state, using prior state values and an optional input $\mathbf{u}$. As features are fixed element in the environment, we can restrict the prediction to: $\hat{\mathbf{\Gamma}}_{t+1} = f(\mathbf{\Gamma_t}, \mathbf{\Lambda_t}, \mathbf{u}_t)$, i.e., the feature prediction is the identity: $\hat{\mathbf{y}}_{i_{t+1}} = \mathbf{y}_{i_t}$.

In Monocular SLAM, for each 3D point observation $\mathbf{o_i}$ (represented in image coordinates as $[u_i, v_i]^T$), only the correspnding viewing ray can be estimated, not its complete positionin the world. The viewing ray, expressed in the camera reference frame C, is coded by direction vector:

$$\mathbf{r}_i^C = \left[\, \frac{u_0 - u_i}{f_x} \;\; \frac{u_0 - u_i}{f_y} \;\; 1 \,\right]^T, \qquad (3)$$

where $[u_0, v_0]^T$ is the principal point image, f_x and f_y are the focal lengths on the two axes. So we have only a partial observation of the feature. A new feature, thus, has to be created as a function of the actual camera pose, the available observation, and some unknowns $\mathbf{U}$ which are represented, in the Gaussian based EKF, as a multidimensional Gaussian variable with a default mean and variance. The new feature is then created by:

$$y_{new} = g(\mathbf{\Gamma}_t, \mathbf{o}_i, \mathbf{U}). \qquad (4)$$

After the feature creation, the state becomes:

$$\mathbf{X}_{new} = \left[\, \mathbf{X}^T \;\; \mathbf{y}_{new}^T \,\right]^T, \qquad (5)$$

and we can compute its covariance by exploiting the Jacobians of the g function.

When a feature, that is already in the filter state, is perceived in a new frame, we model two aspects:

(1) the feature is transformed from the world to the camera frame;
(2) the feature is then projected, using the camera parameters.

This is summarized in the measurement equation:

$$[u_i, v_i]^T = h_i(\mathbf{\Gamma}_t, \mathbf{y}_{i_t}). \qquad (6)$$

The update step of the EKF is performed by collecting all the features that have been observed at each time in one vector $\mathbf{h}_t(\mathbf{\Gamma}_t, \{\mathbf{y}_{i_t}\}) = \{h_i(\mathbf{\Gamma}_t, \mathbf{y}_{i_t})\}$.

3. FEATURE PARAMETRIZATION

The simplest way to represent a 3D point is by means of its Euclidean coordinates: $\mathbf{y}_i^E = [X_i Y_i Z_i]^T$. However, Euclidean points are not suited for undelayed initialization and their measurement function is non linear, as extensively demonstrated in (Solà et al. (2005), Montiel et al. (2006), Marzorati et al. (2009)); in the following, alternative solutions are presented.

3.1 Unified inverse depth (UID)

In the Unified Inverse Depth parametrization a 3D scene point $\mathbf{y}_i^{UID}$ is defined by a vector:

$$\mathbf{y}_i^{UID} = \left[\, \mathbf{t}_i^T \;\; \vartheta_i \;\; \varphi_i \;\; \varrho_i \,\right]^T, \qquad (7)$$

[1] In this paper we focus on world centric SLAM systems; in (Martinez-Cantin and Castellanos, 2006) the robocentric approach has proven able to reduce linearization errors thus, improving filter consistency. Our analysis is here performed in the world reference frame, due to the reduced computational complexity and the paper length limit.

[2] Several representations for rotations are possible (Funda and Paul, 1988); in this paper, being this marginal w.r.t. the main issue of feature representation, we use quaternions to allow comparison with EKF-based SLAM systems publicly available.

which represents a 3D point located at:

$$\mathbf{t}_i + \frac{1}{\varrho_i}\mathbf{m}(\vartheta_i, \varphi_i). \tag{8}$$

where $\mathbf{t}_i^T$ is the camera position (i.e., the position of its projection center) when the 3D point was first observed; ϑ_i and φ_i are respectively the azimuth and the elevation (in the world reference frame) of the line

$$\mathbf{m}(\vartheta_i, \varphi_i) = [\cos(\varphi_i)\sin(\vartheta_i), -\sin(\varphi_i), \cos(\varphi_i)\cos(\vartheta_i)]^T, \tag{9}$$

and $\varrho_i = 1/d_i$ is the inverse of the point depth along this line (see Montiel et al. (2006) for more details). The initialization is performed by:

$$\begin{aligned}\mathbf{y}_i^{UID} &= g^{UID}(\mathbf{\Gamma}_t, \mathbf{o}_i, \varrho_{init}) = \\ &\left[\, \mathbf{t}_t^T \;\; \theta(\mathbf{r}_i^W) \;\; \phi(\mathbf{r}_i^W) \;\; \varrho_{init} \,\right]^T,\end{aligned} \tag{10}$$

where

$$\mathbf{r}_i^W = \mathbf{R}(\mathbf{q}_t) \cdot \mathbf{r}_i^{C_t}, \tag{11}$$

is the viewing ray of $[u_{i_t}, v_{i_t}]^T$ in a frame that is oriented as the world reference frame ($\mathbf{R}(\mathbf{q})$ convert the quaternion $\mathbf{q}$ to a 3x3 rotation matrix), and

$$\theta(\mathbf{r}) = \operatorname{atan2}(\mathbf{r}_x, \mathbf{r}_z); \quad \phi(\mathbf{r}) = \operatorname{atan2}(-\mathbf{r}_y, \sqrt{\mathbf{r}_x^2 + \mathbf{r}_z^2}). \tag{12}$$

Using this representation for each feature i, we obtain the following measurement equation:

$$h_i^{UID}(\mathbf{\Gamma}_t, \mathbf{y}_{i_t}) = K\left(\mathbf{R}(\mathbf{q}_t)^T\left(\mathbf{t}_i - \mathbf{t}_t + \frac{1}{\varrho_i}\mathbf{m}(\vartheta_i, \varphi_i)\right)\right) \tag{13}$$

where K is here intended as a function that, beside computing the projection of the viewing ray in the image frame, also transforms from homogeneous coordinates to euclidean. This equation can be written, to avoid division by zero when the feature is at infinity, as:

$$h_i^{UID}(\mathbf{\Gamma}_t, \mathbf{y}_{i_t}) = K\left(\mathbf{R}(\mathbf{q}_t)^T\left(\varrho_i\left(\mathbf{t}_i - \mathbf{t}_t\right) + \mathbf{m}(\vartheta_i, \varphi_i)\right)\right) \tag{14}$$

This representation requires the storage of six parameters in the state vector for each map feature.

3.2 Inverse scaling (IS)

In Marzorati et al. (2009) the Inverse Scaling parametrization was presented. It does not use polar coordinates, while it uses the homogeneous representation of a 3D ponit:

$$\mathbf{y}_i^{IS} = \left[\, \mathbf{t}_i^T \;\; \omega_i \,\right]^T. \tag{15}$$

The transformation of a point from IS to euclidean coordinates is:

$$\frac{\mathbf{t}_i}{\omega_i}. \tag{16}$$

The initialization of a feature is as follows:

$$\mathbf{y}_i^{IS} = g^{IS}(\mathbf{\Gamma}_t, \mathbf{o}_i, \omega_{init}) = \left[\, \left(\mathbf{r}_i^W + \mathbf{t}_t\right)^T \;\; \omega_{init} \,\right]^T. \tag{17}$$

The new parametrization changes the measurement equation into the following:

$$h_i^{IS}(\mathbf{\Gamma}_t, \mathbf{y}_{i_t}) = K\left(\mathbf{R}(\mathbf{q}_t)^T\left(\mathbf{t}_i - \omega_i \mathbf{t}_t\right)\right). \tag{18}$$

3.3 Anchored Homogeneous Point (AHP)

The Anchored Homogeneous Point parametrization was proposed in Solà (2010), and combined UID and IS. In this parametrization a point is represented by a 7-vector: from UID it took the idea of representing the Euclidean optical center at the initialization time, a 3-vector; from IS it took the 3D point expressed in homogeneous coordinates, i.e., a 4-vector, to encode both the viewing ray direction and the distance information. An AHP point is hence coded as:

$$\mathbf{y}_i^{AHP} = \left[\, \mathbf{t}_i^T \;\; \mathbf{m}_i^T \;\; \omega_i \,\right]^T, \tag{19}$$

where $\mathbf{t}_i$ is the 3-vector, i.e., the position of the camera, and $[\mathbf{m}_i \;\; \omega_i]^T$ is the 4-vector, where $\mathbf{m}_i$ is applied in the camera pose and points towards the feature, and ω_i is the inverse of the scale of $\mathbf{m}_i$. The transformation into a 3D point is thus described by the sum of two vectors:

$$\mathbf{t}_i + \frac{\mathbf{m}_i}{\omega_i}. \tag{20}$$

The initialization is performed by:

$$\begin{aligned}\mathbf{y}_i^{AHP} &= g^{AHP}(\mathbf{\Gamma}_t, \mathbf{o}_i, \omega_{init}) = \\ &\left[\, \mathbf{t}_t^T \;\; \mathbf{r}_i^{W\,T} \;\; \omega_{init} \,\right]^T,\end{aligned} \tag{21}$$

and the measurement equations of the feature is:

$$h_i^{AHP}(\mathbf{\Gamma}_t, \mathbf{y}_{i_t}) = K\left(\mathbf{R}(\mathbf{q}_t)^T\left(\omega_i \;\; (\mathbf{t}_i - \mathbf{t}_t) + \mathbf{m}_i\right)\right). \tag{22}$$

3.4 Framed homogeneous point (FHP)

We propose hereafter a new parametrization, in a sense similar to anchoring a point represented in Inverse Scaling. Differently from AHP, we substitute the anchor point with the complete frame of the camera. By doing this, we obtain an "anchor frame" represented by the position and the orientation of the camera when the feature was initialized. Since the camera frame is completely specified for the feature, we can represent the viewing ray $\mathbf{r}^C$ using only its first two elements, as the third is 1 in the camera frame. The last parameter of framed point parametrization is the inverse distance along the viewing ray.

The complete parametrization results in a 10-dimensional vector [3]:

$$\mathbf{y}_i^{FHP} = \left[\, \mathbf{t}_i^T \;\; \mathbf{q}_i^T \;\; u_i \;\; v_i \;\; \omega_i \,\right]^T, \tag{23}$$

being $\mathbf{t}_i$ and $\mathbf{q}_i$ the camera position and orientation when the feature was created, u_i and v_i the first two elements of a viewing ray while ω_i encodes the inverse scaling factor for the viewing ray. The transformation into a 3D point is:

$$\mathbf{t}_i + \frac{1}{\omega_i}\,\mathbf{R}({\mathbf{q}^*}_i) \cdot \left[\, u_i \;\; v_i \;\; 1 \,\right]^T, \tag{24}$$

where

$${\mathbf{q}^*}_i = \frac{\mathbf{q}_i}{\|\mathbf{q}_i\|} \tag{25}$$

is the normalized quaternion. This operation is needed because the EKF update step does not guarantee that quaternions remain normalized.

[3] Our implementation is based on quaternions. In the case of an Euler angles based implementation, the FHP parametrization would require 9 elements for each feature.

The initialization is done through:

$$\begin{aligned} \mathbf{y}_i^{FHP} &= g^{FHP}(\mathbf{\Gamma}_t, \mathbf{o}_i, \omega_{init}) = \\ &\left[\mathbf{t}_t^T \;\; \mathbf{q}_t^T \;\; \mathbf{r_x}_i^{C_t} \;\; \mathbf{r_y}_i^{C_t} \;\; \omega_{init} \right]^T \end{aligned} \tag{26}$$

while the measurement equation of the feature is:

$$\begin{aligned} & h_i^{FHP}(\mathbf{\Gamma}_t, \mathbf{y}_{i_t}) = \\ K & \left(\mathbf{R}(\mathbf{q}_t)^T \left(\omega_i \left(\mathbf{t}_i - \mathbf{t}_t\right) + \mathbf{R}(\mathbf{q^*}_i) \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \right) \right). \end{aligned} \tag{27}$$

3.5 Framed vs. anchored parametrization

The advantages of an anchored parametrization are maily due to two factors.

The first factor is related to feature initialization. When a new feature is initialized, uncertainty in camera position becomes correlated to the anchor point, while the uncertainties in feature measurement and representation are taken into account in the viewing ray description. Consequently, when an update is performed, these correlations, together with a more detailed measurement equations, allow to distribute errors in a more effective way. This implies that the more details are dropped in the parametrization, the worst is the results on tracking the feature position.

In our opinion, this is the main reason that explains why the inverse scaling parametrization performs worse than others parametrization, as shown in (Solà, 2010), and the reason to expect a better performance from the FHP parametrization. In UID and AHP parametrizations, the anchor point codes the position of the camera when the feature is initialized, while the viewing ray carries mixed information about the the attitude of the camera, when the feature was initialized, and the viewing ray in the camera frame. The FHP parametrization has the advantage of keeping these sources of information separated.

The previous reasoning is better explained in the following comparison. Let's consider the initialization of a landmark in the four different parametrizations (Equations 10, 17, 21, and 26). UID, AHP, and IS parametrizations are initialized by a non linear function (i.e., the rotation of the viewing ray in the world reference frame implies a linearization), while FHP parametrization has a linear initialization. This implies that when a new feature is initialized, no information on uncertainty is dropped due to linearization.

The second factor is related to the gains that can be attained in the measurement equations by using rich parametrization of the features. As an example, let's consider the part of the feature that codes the orientation of the camera in the FHP parametrization. These elements correlate with the current state of the camera without involving the viewing ray, and this is not possible in other parametrization, because of the mixed information of frame and viewing ray. For these reasons, we expect FHP to be more consistent than other parametrization; although it requires more space in the filter.

4. EXPERIMENTAL RESULTS

The experimental evaluation of our proposal is conducted in this section through experiments on simulated and real data. For the simulated data, we use the Matlab software presented in Solà (2010), while real data are taken from RAWSEEDS[4] (Ceriani et al., 2009) and are processed by using a C++ framework for SLAM developed in our research group.

4.1 Evaluation on simulated data

In the experiments on simulated data, we are interested in evaluationg the filter consistency through the average Normalized Estimation Error Squared (NEES) (Bailey et al., 2006) as done also by Solà (2010). Knowing the groudtruth about some variable $\mathbf{x}_t$, the NEES value at time t is computed as:

$$\epsilon_t = (\mathbf{x}_t - \hat{\mathbf{x}}_t)^T \mathbf{P}_t^{-1} (\mathbf{x}_t - \hat{\mathbf{x}}_t) \tag{28}$$

where $\hat{\mathbf{x}}_t$ is the estimated value of $\mathbf{x}_t$ and $\mathbf{P}_t$ its covariance.

The simulator runs N trials of the same problem with a random noise and checks SLAM consistency using the 95% confidence interval of the average NEES for camera position and attitude. The filter is considered too conservative if the NEES value is below the lower bound of the confidence iterval or, on the contrary, too optimistic (i.e., the errors in camera pose are underestimated and the final trajectory and map badly recostructed) if it is above the upper bound of the confidence interval.

The Monocular SLAM algorithm implemented by the simulator performs an active-search-based SLAM (Davison et al., 2007). It initializes 10 landmarks in the first frame, then for each frame the 10 more informative landmarks are updated and a new landmark is added (whenever available). Inconsistent and unstable landmarks are removed from the map.

Two different setups for the experiments are used. In the first setup a camera is simulated pointing in the direction of the robot movement. The camera has a 640x480px CCD sensor with a 90° field of view. The robot moves on a planar circular trajectory inside an area of 12x12 meters populated with 72 landmarks whose overall shape resembles a cloister[5]. Experiments 1 to 4 of Table 1 refer to this setup. In the second setup the same camera is moved according to a full 6DoF motion, and, to guarantee that the robot is able to perceive landmarks in the environment a denser cloister is created with 180 landmarks lying on five different planes. The main difference, and challenge, of this second setup with respect to the previous one is the 6DoF motion of the camera. Experiment 5 of Table 1 refers to this setup.

In the experiments of the first setup, we use two different configurations for the camera movement: in the first configuration the camera increments its position of $0.08m$ along robot x axis (which points forward) and 0.9° around z axis (that points up); a complete turn of the cloister is completed in 400 frames. In the second configuration

[4] http://www.rawseeds.org

[5] For a detailed description of the simulated environment refer to (Solà, 2010).

#	Noises mm deg	init m^{-1}	σ m^{-1}	Consistency IS	 UID	 AHP	 FHP
$\Delta X = 0.08m, \Delta\Psi = 0.9°$							
1.1	2.5, 0.025	1	1	no	no	no	no†
1.2		0.01	0.5	no	yes	yes	yes
2.1	1.25, 0.0125	1	1	no	no	no	yes
2.2		0.01	0.5	no	yes	yes	yes
$\Delta X = 0.04m, \Delta\Psi = 0.45°$							
3.1	2.5, 0.025	1	1	no	yes	yes	yes
3.2		0.01	0.5	no	yes	yes	yes
4.1	5, 0.05	1	1	no	no	no	no‡
4.2		0.01	0.5	no	no	no	no
$\Delta_{XYZ} = [0.08, 0.02, -0.02]\, m, \Delta_{\Theta\Phi\Psi} = [0.2, -0.45, 0.9]°$							
5.1	1.25, 0.0125	1	1	no	no	no	yes
5.2		0.01	0.5	no	yes	yes	yes

Table 1. Summary of Results. In experiment 2.1, and 5.1 FHP is the only parametrization reaching filter consistency (see Figure 1 and Figure 2). † FHP is not consistent but it is close to (see Figure 3). ‡ All parametrizations are above NEES upper bound, but FHP performs better (see Figure 4).

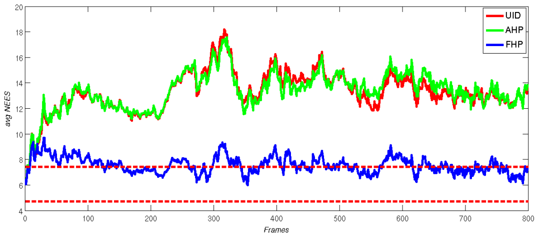

Fig. 1. NEES results for experiment 5.1. FHP shows consistency for the entire trajectory, while AHP and UID do not.

the camera moves slower (increments are halved) and the complete turn needs 800 frames. In the second setup, motion increments are on and around all axes.

A zero mean Gaussian random noise is added to the control variables (i.e., to the linear increments along the three axis and to the angular increments around the three axis) and we try two different values for standard deviation. The unobservable part of the feature (i.e., feature depth) is initialized with two different values: in one case we use $1\mathtt{m}^{-1}$ for the mean of the inverse distance (or scale) and $1\mathtt{m}^{-1}$ for its standard deviation; in the second case $0.01\mathtt{m}^{-1}$ is used for the mean and $0.5\mathtt{m}^{-1}$ for its standard deviation (that implies a very far initialization for the point).

In all experiments FHP shows performance at least equal to AHP and UID. Notice that AHP and UID show similar performances [6] , while IS is always worse (and this confirms the relevance of the anchor point/frame). Looking at Table 1, in two cases FHP maintains consistency where others parametrization cannot (Figure 1 and Figure 2); and in two cases FHP is not consistent, but performs better than other parametrizations anyway (Figure 3 and Figure 4).

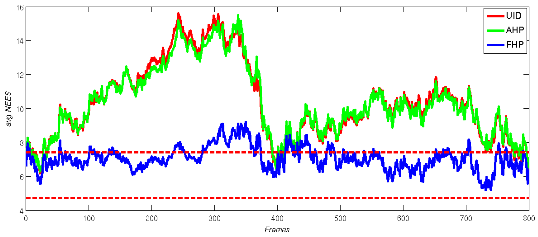

Fig. 2. NEES results for experiment 2.1. FHP is consistent for the most of time. At frame 400 a loop is closed.

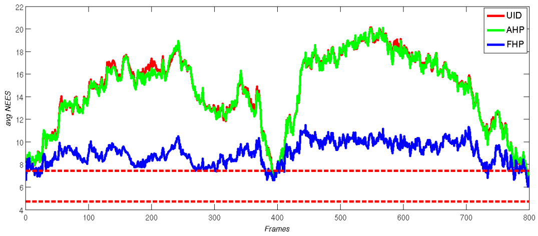

Fig. 3. NEES results for experiment 1.1. FHP is close to consistency. At frame 400 a loop is closed.

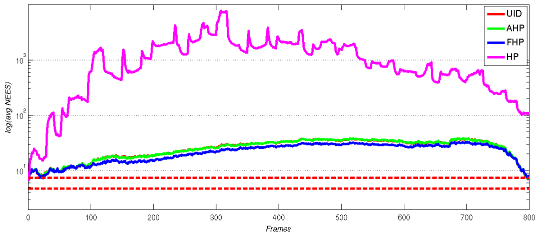

Fig. 4. NEES results for experiment 4.1. FHP is slightly better than AHP and UID. IS is shown only in this picture, but its performance is consistently worse in all experiments. Vertical axis is in logarithmic scale.

4.2 Evaluation on real data

The evaluation on real data is done on a small subset of an indoor dataset taken from RAWSEEDS [7] (Ceriani et al., 2009); in particular, we used the stream from the lowcost color camera, a Unibrain Fire-i (320x240px resolution with a focal length of about 200px), from the dataset Bicocca_2009-02-26a, starting at timestamp 1235647818.221857 and ending at 1235647843.517272 (760 frames at 30fps). The filter state includes the position and the attitude of the camera, and no information on speed is estimated by the filter. The new state is predicted using the odometric information of the robot.

As for the simulated experiments, the algorithm performs active-search-based SLAM. It initializes at most 32 landmarks in the first frame, then, for each frame, it matches landmarks by cross-correlation and performs update with all the matched landmarks. New landmarks are added if less than 16 landmarks are reprojected to the current frame. No landmark deletion is applied. We are aware that this is a rather poor Monocular SLAM system compared to those that use compatibility test for update selection, e.g., like in Civera et al. (2010) or submapping techniques like in Piniés et al. (2010); however, we expect that a more consistent parametrization could improve the filter

[6] See also the fix to Solà's toolbox, published on 2010/09/04 at http://homepages.laas.fr/jsola/JoanSola/eng/toolbox.html

[7] http://www.rawseeds.org

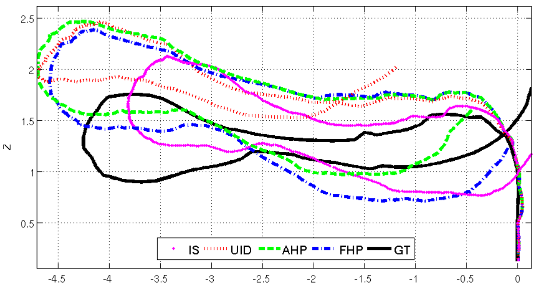

Fig. 5. Estimated trajectory with real data compared to the extended groundtruth.

behavior in simple implementations as well as in more sophisticated ones.

Figure 5 shows the estimated trajectory results using IS, UID, AHP and FHP compared with the available "extended ground truth" (i.e., a ground truth obtained by scan-matching laser range finder data) that is available in this dataset. The initial inverse depth, or inverse scale, is set to $0.25m^{-1}$ with a standard deviation of $0.5m^{-1}$ for all parametrizations.

All the parametrizations fail to estimate correctly the trajectory but, from the figure, we can observe that FHP gives more chances to detect the loop closure, resulting more robust to orientation changes.

5. CONCLUSIONS AND FUTURE WORKS

We have shown that a framed parametrization could be useful to achieve a more consistent SLAM filter. We can consider this parametrization as a "complete" parametrization, because it describes a point perceived by a camera using all the components available (i.e., the complete pose of the camera and the homogeneous viewing ray). The disadvantages of this solution are in the larger memory space required for the storing the map; this affects the performance of the SLAM algorithm, e.g., the EKF computational complexity depends by the state dimension. However, this is issue could be faced with the conversion of features in Euclidean coordinates every time a linearity test is satisfied as proposed in (Civera et al., 2007).

To obtain a reduction in the filter state dimension, we are now working on a solution which shares the frame parameters between features that are created at the same time. By doing this way, when N landmarks are created, the state augmentation is of $7 + 3N$ elements instead of $10N$ elements. This should improve the consistency of the filter even further, being the anchor frame exactly the same for features added simultaneously; this has been inspired by a similar proposal, (Imre et al., 2009), for UID.

REFERENCES

T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot. Consistency of the ekf-slam algorithm. In *Proc. of IEEE Intern. Conf. on Intelligent Robots and Systems*, pages 3562–3568, 2006.

R. O. Castle, G. Klein, and D. W. Murray. Combining monoSLAM with object recognition for scene augmentation using a wearable camera. *Journ. of Image and Vision Computing*, 28(11):1548 – 1556, 2010.

S. Ceriani, G. Fontana, A. Giusti, D. Marzorati, M. Matteucci, D. Migliore, D. Rizzi, D. G. Sorrenti, and P. Taddei. Rawseeds ground truth collection systems for indoor self-localization and mapping. *Autonomous Robots*, 27 (4):353–371, 2009.

J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse depth to depth conversion for monocular slam. In *Proc. of IEEE Intern. Conf. on Robotics and Automation*, pages 2778–2783, 2007.

J. Civera, O. Grasa, A.J. Davison., and J. M. M. Montiel. 1-point RANSAC for Extended Kalman filtering: Application to real-time structure from motion and visual odometry. *Journ. of Field Robotics*, 27, 2010.

A. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. of IEEE Intern. Conf. on Computer Vision*, 2003.

A. J. Davison, I. D. Reid, N.D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. on Patt Anal. and Machine Intellig.*, 29(6):1052–1067, 2007.

J. Funda and R.P. Paul. A comparison of transforms and quaternions in robotics. In *Proc. of the 1988 IEEE Intern. Conf. on Robotics and Automation*, volume 2, pages 886–891, 1988.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, second edition, 2004.

E. Imre, M.O. Berger, and N. Noury. Improved inverse-depth parameterization for monocular simultaneous localization and mapping. In *Proc. of IEEE Intern. Conf. on Robotics and Automation*, pages 381–386, 2009.

R. Martinez-Cantin and J. Castellanos. Bounding uncertainty in EKF-SLAM: The robocentric local approach. In *Proc. of IEEE Intern. Conf. on Robotics and Automation*, 2006.

D. Marzorati, M. Matteucci, D. Migliore, and D. G. Sorrenti. On the use of inverse scaling in monocular slam. In *Proc. of IEEE Intern. Conf. on Robotics and Automation*, pages 2030–2036, 2009.

C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. Rslam: A system for large-scale mapping in constant-time using stereo. *Intern. Journ. of Computer Vision*, pages 1–17, 2010.

J. Montiel, J. Civera, and A. J. Davison. Unified inverse depth parametrization for monocular slam. In *Proc. of Robotics: Science and Systems*, August 2006.

R.A. Newcombe and A.J. Davison. Live dense reconstruction with a single moving camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conf. on*, pages 1498 –1505, jun. 2010.

P. Piniés, L. Maria Paz, D. Gálvez-López, and J. D. Tardós. CI-Graph simultaneous localization and mapping for three-dimensional reconstruction of large and complex environments using a multicamera system. *Journ. of Field Robotics*, 27(5), 2010.

J. Solà. Consistency of the monocular EKF-SLAM algorithm for three different landmark parametrizations. In *Robotics and Automation (ICRA), 2010 IEEE Intern. Conf. on*, pages 3513–3518. IEEE, 2010.

J. Solà, A. Monin, M. Devy, and T. Lemaire. Undelayed initialization in bearing only slam. In *Proc. of Intern. Conf. on Intelligent Robots and Systems*, pages 2499–2504, 2005.