3D Model-based estimation for UAV tracking

Nuno Pessanha Santos & Victor Lobo
Portuguese Navy Research Center (CINAV)
Portuguese Navy
2810-001, Almada, Portugal
nuno.pessanha.santos@marinha.pt; vlobo@novaims.unl.pt

Alexandre Bernardino
Institute for Systems and Robotics (ISR)
Instituto Superior Técnico (IST)
1049-001, Lisboa, Portugal
alex@isr.ist.utl.pt

***Abstract*—This work presents a system to perform autonomous landing of a small size fixed-wing Unmanned Aerial Vehicle (UAV) on a Fast Patrol Boat (FPB). We propose a ground-based vision system with the camera, image capture and processing equipment installed in the ship, thus reducing the UAV size, weight and power requirements. The system observes the UAV and computes the control commands to send to the UAV via radio. This approach makes it also possible to use standard UAVs equipped with commercial autopilots. The developed system uses the captured image as input and a Particle Filter (PF) structure to estimate the UAV trajectory. It is also used an Unscented Kalman Filter (UKF) for the translational motion filtering and an Unscented Bingham Filter (UBiF) for the rotational motion filtering. This filtering structure is reminiscent of the Unscented Particle Filter (UPF). The obtained tracking error is compatible with automatic landing requirements.**

***Index Terms*—Computer Vision, Model Based-Pose Estimation, Model-Based Tracking, Autonomous Vehicles.**

I. Introduction

Over the past years, the Unmanned Aerial Vehicle (UAV) field has grown significantly and nowadays several Commercial-Off-The-Shelf (COTS) UAVs are available. This evolution makes the technology available to all. UAVs can significantly reduce the operational costs on tasks that traditionally require full-size vehicles such as pipeline inspection and aerial surveillance.

The current developments on UAV technology have unlocked a valuable opportunity to extend military operational capability in maritime environments being able to perform maritime reconnaissance, surveillance, patrolling, and search and rescue. The capability of performing autonomous take-off and landing is critical since it increases the system reliability and decreases the needed crew training and specialization. The landing area in a Fast Patrol Boat (FPB) is usually a small area located on the stern section with 5×6 meters. In our tests, we are using a small-size fixed-wing UAV with 5 kg Maximum Take-Off Weight (MTOW), 180 cm of wingspan and 150 cm in length compatible with the available landing area.

Our primary objective is to perform an autonomous landing on a ship using a ground-based vision system (Figure 1). It is used a monocular Red, Green and Blue (RGB) camera located on the ship with a processing station to perform the needed Computer Vision (CV) processing tasks. The use of a ground-based system makes it possible to use simple COTS UAVs with an autopilot capable of performing simple trajectories. The outdoor environment itself is very challenging especially in a maritime environment that is more susceptible to the weather conditions (e.g. wave and the wind).

Fig. 1. Landing area illustration.

The majority of the research made in this field is based on UAV onboard sensors and processors, using markers on the platform to facilitate CV [1]–[4]. Ground-based vision systems are not often considered. In our previous work [5]–[7] a vision system based on a standard RGB camera was used to track a UAV landing aboard a ship. In [5] is used a combination of a Boosted classifier for UAV detection, a fast sampler of pose hypothesis and a Particle Filter (PF) for pose estimation in a single frame. In [6] used a PF for pose estimation and an Unscented Kalman Filter (UKF) for filtering while in [7] used a novel resampling step based on the evolution strategies found in the genetic algorithms.

In a periodic domain like the manifold of orientations in a 3D space, the Gaussian approach is not a good approximation. To overcome this, in this paper we extend our previous work by using the Bingham (Bi) distribution [8] defined in the orientation hypersphere ($SO3$). In our problem, rotational filtering is performed combining a PF with an Unscented Bingham Filter (UBiF) [9], [10].

The main contribution of this paper is, thus, the use of directional statistic distributions in the UAV tracking, to improve the obtained orientation estimation. Until now, no other UAV tracking system addressed the use of these distributions in a filtering structure.

This article is organized as follows. In Section II the overall system description is made. In Section III, is detailed the target detection stage. In Section IV, is described the pose initialization architecture using a pre-trained database. In Section V, the UAV motion models and the adopted filtering structure are detailed. In Section VII we present some experimental results. Finally, in Section VIII we present the conclusions of

978-1-5386-4814-8/18/$31.00 ©2018 IEEE

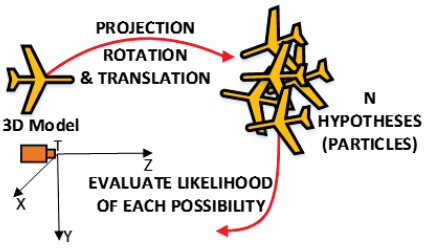

Fig. 2. Multiple pose hypotheses illustration.

Fig. 3. UAV CAD model.

the article and provide directions for further research work.

II. System description

The developed system is based on a PF architecture [11]–[18] using multiple hypotheses (particles) for pose estimation (Figure 2). The particle set is given by:

$$\tilde{\mathbf{x}}_i = [\mathbf{T}_i, \mathbf{O}_i] \quad \text{with} \quad \mathbf{T}_i = [\mathbf{p}_i, \mathbf{v}_i] \text{ and } \mathbf{O}_i = [\mathbf{q}_i, \boldsymbol{\omega}_i] \quad (1)$$

where $\mathbf{p}_i = [X^i, Y^i, Z^i]$ is the linear position, $\mathbf{v}_i = [v_x^i, v_y^i, v_z^i]$ is the linear velocity, $\mathbf{q}_i = [q_1^i, q_2^i, q_3^i, q_4^i]$ is the orientation quaternion and $\boldsymbol{\omega}_i = [\omega_x^i, \omega_y^i, \omega_z^i]$ is the angular velocity (Figure 9).

When we project the 3D Computer-Aided Design (CAD) model (Figure 3) of a particle in the image frame, we obtain a hypothesis of the appearance of the UAV in the acquired image. This is done by generating a synthetic image of the UAV using a computer graphics rendering engine. To check the quality of this hypothesis, we will compare the synthetic image with the real image using a likelihood function. The used likelihood metric was the color likelihood as in [5]–[7]. The particle with the highest likelihood will also be used to update the actual filters measurement (Figure 4). Combining a PF with other filtering techniques for motion filtering is a known technique to improve the estimation using a better proposal distribution [12], [13], [19]–[22].

The proposed method is divided into four major parts (Figure 4): target detection (Section III), appearance-based pose sampler (Section IV), filtering (Section V) and image-based resampling (Section VI).

III. Target detection

The target detection phase consists of searching in the image for Regions of interest (ROIs). A ROI represents an image area that may contain an object classified as the UAV to land (Figure 5). This detection is made using a trained Multi-scale Block Local Binary Pattern (MB-LBP) cascade classifier [23], ensuring that we have a region of interest given by the classifier with small error percentage [5], [6].

Fig. 5. Obtained target detection examples.

IV. Appearance-based pose sampler

The appearance-based pose sampling relies on the Oriented Bounding Box (OBB) that best fits UAV corner points obtained by Features from Accelerated Segment Test (FAST) detector [24] (Figure 6). Then the OBB is compared with a pre-trained database of UAV OBBs in multiple poses [5]–[7]. The difference between the angle and the Aspect Ratio (AR) of the observation and database is calculated online using the *Euclidean* distance, and all the poses with a satisfactory score ($score < threshold$) will be used as pose samples (Figure 7). In the first iteration of the filter, all the particles come from the database. After the first filter iteration, the particles coming from the database replaces 25% of the particles with lower likelihood value (Figure 4).

Fig. 6. Obtained FAST features (*white*) and OBB (*red*) example.

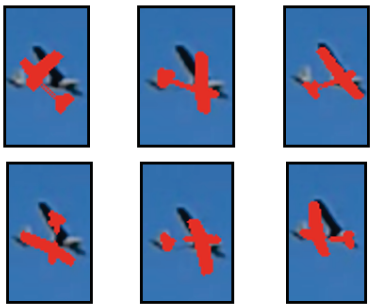

Fig. 7. Example of obtained poses from the best database matches.

V. Filtering

We have considered that the UAV follows a constant velocity model and that linear and angular motions are independent, i.e. it suffers small accelerations between two consecutive time steps t and $t+1$ [18]. We will use a UKF [13], [25]–[30] for the translational motion filtering (Section V-B) and a UBiF [9], [10], [31] for the rotational motion filtering (Section V-D).

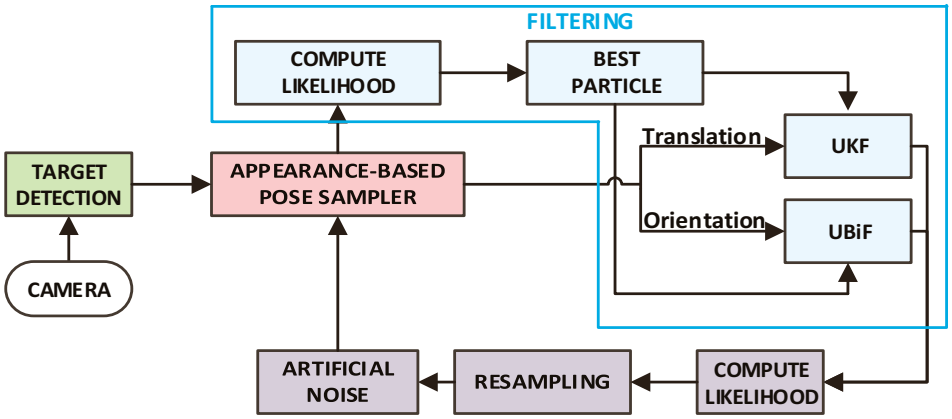

Fig. 4. System architecture.

A. Translational model

Given the stated assumptions, the linear motion state vector is defined as:

$$\mathbf{x}^l = \mathbf{T} = [\mathbf{p}, \mathbf{v}] \tag{2}$$

where $\mathbf{p} = [X, Y, Z]$ and $\mathbf{v} = [v_x, v_y, v_z]$ are respectively the linear position and velocities of the UAV. The time evolution of the state is:

$$\mathbf{x}^l_{t+1} = \mathbf{F}^l(\mathbf{x}^l_t, \boldsymbol{\xi}^l_t) = \mathbf{A}^l\mathbf{x}^l_t + \boldsymbol{\xi}^l_t \tag{3}$$

where $\boldsymbol{\xi}^l_t$ is a Gaussian noise random variable with zero mean and covariance $\mathbf{Q}^l_t$. $\mathbf{A}^l$ is the linear dynamics matrix:

$$\mathbf{A}^l = \begin{bmatrix} \mathbf{I}_{3\times3} & \Delta t \cdot \mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix} \tag{4}$$

The observation model is the following:

$$\mathbf{y}^l_t = \mathbf{H}^l\left(\mathbf{x}^l_t, \boldsymbol{\eta}^l_t\right) = \mathbf{C}^l\mathbf{x}^l_t + \boldsymbol{\eta}^l_t \tag{5}$$

where $\boldsymbol{\eta}^l_t$ is a Gaussian noise random variable with zero mean and covariance matrix $\mathbf{R}_t$. Since we only measure the position and not the velocity we have that $\mathbf{C}^l = [\mathbf{I}_{3\times3}, \mathbf{0}_{3\times3}]$.

As we are using a linear model for translation, a simple Kalman Filter (KF) could be applied, but since the UKF is simple to implement and to study better its implementation, we decide to use it [6].

B. Unscented Kalman Filter

In the UKF, the Gaussian distributions of the state and process noise are used to generate a set of points (sigma points) that are sufficient to represent their statistics using an Unscented Transform (UT) [19], [27], [32] approach. Then, these points are propagated through the process and measurement models to obtain new sets of sigma points from which the mean and covariance of the state and measurement predictions are obtained, assuming Gaussian approximations.

The process noise covariance $\mathbf{Q}_t$ (n dimensions vector) and the state covariance $\mathbf{P}_t$ ($n \times n$ matrix) are transformed into a $2n$ set of points $\delta\mathbf{x}_t(i)$ that represent perturbations to the current state according to:

$$\delta\mathbf{x}_t(i) = columns\left(\pm\sqrt{2n \cdot \iota \cdot (\mathbf{P}_t + \mathbf{Q}_t)}\right) \tag{6}$$

where $i = 1, ..., 2n$ and the scaling parameter ι is given by:

$$\iota = \alpha^2(n + k) \tag{7}$$

where α is a positive real ($0 \leq \alpha \leq 1$) parameter that controls the high order effects resulting from the existing nonlinearity, k is another real parameter ($k \geq 0$) that will control the distance between the sigma points and their average [13], [33], [34]. The matrix $\mathbf{P}_t$ is symmetric and positive definite, so it is possible to use the *Cholesky* decomposition to compute $\sqrt{\iota \cdot (\mathbf{P}_t + \mathbf{Q}_t)}$ [35]. The computation of the sigma points, is now done by adding directly $\delta\mathbf{x}_t(i)$ to the state vector $\mathbf{x}_t$ according to:

$$\mathcal{X}_i = \mathbf{x}_t + \delta\mathbf{x}_t(i) \qquad i = 1, ..., 2n \text{ and } \mathcal{X}_0 = \mathbf{x}_t \tag{8}$$

The process model $\mathbf{F}(.)$ is then applied to the obtained sigma points, generating the transformed sigma points:

$$\mathcal{X}'_i = \mathbf{F}(\mathcal{X}_i, 0) \qquad i = 0, ..., 2n \tag{9}$$

No additional noise is considered at this step because the noise was already added at the sigma point's creation step (Equation 8). The *a priori* state estimate is obtained by calculating the mean of the transformed sigma points $\mathcal{X}'_i$ according to:

$$\bar{\mathbf{x}}_{t+1} = \sum_{i=0}^{2n} \mathbf{w}_i\mathcal{X}'_i \tag{10}$$

The weights $\mathbf{w}_i$ come from the UT [19], [20], [25], [27], [32] and are given by ($i = 1, ..., 2n$):

$$\mathbf{w}_0 = \frac{\lambda}{n + \lambda} \qquad \text{and} \qquad \mathbf{w}_i = \frac{1}{2(n + \lambda)} \tag{11}$$

with λ given by:

$$\lambda = \alpha^2(n+k) - n \tag{12}$$

To estimate the *a priori* state covariance each propagated sigma point is removed from its mean to create the set of error vectors:

$$\delta\bar{\mathbf{x}}_{t+1}(i) = \mathcal{X'}_i - \bar{\mathbf{x}}_{t+1} \tag{13}$$

then:

$$\mathbf{P}^{xx}_{t+1} = \sum_{i=0}^{2n} \mathbf{w}_i \, \delta\bar{\mathbf{x}}_{t+1}(i) \, \delta\bar{\mathbf{x}}_{t+1}(i)^T \tag{14}$$

where the scaling weights $\mathbf{w}_i$ are given by Equation 11, with exception of $\mathbf{w}_0$ alternatively given by [20], [25]:

$$\mathbf{w}_0 = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta) \tag{15}$$

where β is a non-negative term which incorporates knowledge of the higher order moments of the distribution [13], [33], [34]. The chosen α and β determine the accuracy of third and higher order moments for non-Gaussian inputs [20].

The transformed sigma points are now projected into the measurement space according to:

$$\mathcal{Y}_i = \mathbf{H}(\mathcal{X'}_i, 0) \tag{16}$$

The measurement expected value is computed as:

$$\bar{\mathbf{y}}_{t+1} = \sum_{i=1}^{2n} \mathbf{w}_i \mathcal{Y}_i \tag{17}$$

and the measurement covariance estimative $\mathbf{P}^{yy}_{t+1}$ is given by:

$$\mathbf{P}^{yy}_{t+1} = \sum_{i=0}^{2n} \mathbf{w}_i \left[\mathcal{Y}_i - \bar{\mathbf{y}}_{t+1}\right] \left[\mathcal{Y}_i - \bar{\mathbf{y}}_{t+1}\right]^T \tag{18}$$

The innovation vector $\boldsymbol{\nu}_{t+1}$ is obtained comparing the actual measurement $\mathbf{y}_{t+1}$ to the measurement estimate $\bar{\mathbf{y}}_{t+1}$:

$$\boldsymbol{\nu}_{t+1} = \mathbf{y}_{t+1} - \bar{\mathbf{y}}_{t+1} \tag{19}$$

The innovation covariance $\mathbf{P}^{\nu\nu}_{t+1}$ is obtained adding the measurement noise $\mathbf{R}_{t+1}$ to the measurement covariance $\mathbf{P}^{yy}_{t+1}$:

$$\mathbf{P}^{\nu\nu}_{t+1} = \mathbf{P}^{yy}_{t+1} + \mathbf{R}_{t+1} \tag{20}$$

The cross correlation matrix $\mathbf{P}^{xy}_{t+1}$ is obtained from $\mathcal{Y}_i$ and $\mathcal{X'}_i$, according to:

$$\mathbf{P}^{xy}_{t+1} = \sum_{i=0}^{2n} \mathbf{w}_i \left[\mathcal{X'}_i - \bar{\mathbf{x}}_{t+1}\right] \left[\mathcal{Y}_i - \bar{\mathbf{y}}_{t+1}\right]^T \tag{21}$$

The Kalman gain is then computed from:

$$\mathbf{K}_{t+1} = \mathbf{P}^{xy}_{t+1} (\mathbf{P}^{\nu\nu}_{t+1})^{-1} \tag{22}$$

Finally, the *a posteriori* state estimate is obtained according to:

$$\mathbf{x}_{t+1} = \bar{\mathbf{x}}_{t+1} + \mathbf{K}_{t+1}\boldsymbol{\nu}_{t+1} \tag{23}$$

and the state covariance $\mathbf{P}_{t+1}$ is given by:

$$\mathbf{P}_{t+1} = \mathbf{P}^{xx}_{t+1} - \mathbf{K}_{t+1}\mathbf{P}^{\nu\nu}_{t+1}\mathbf{K}^T_{t+1} \tag{24}$$

C. Rotational model

Representing a distribution directly in the space of *Euler* angles is difficult because of the existing singularities (*gimbal lock*). To deal with this, we use a unit quaternion:

$$\mathbf{q} = [\boldsymbol{\varrho}, q_4] \tag{25}$$

where $\mathbf{q} \in S^3 \subset \mathbb{R}^4 : ||\,\mathbf{q}\,|| = 1$ with:

$$\boldsymbol{\varrho} = [q_1, q_2, q_3] = \hat{\mathbf{e}} \sin\left(\frac{\zeta}{2}\right) \quad \text{and} \quad q_4 = \cos\left(\frac{\zeta}{2}\right) \tag{26}$$

where $\hat{\mathbf{e}}$ is the axis rotation and ζ is the angle of rotation. The time evolution of the orientation is obtained by [28]:

$$\mathbf{q}_{t+1} = \mathbf{q}_t \otimes \delta\mathbf{q}^{\boldsymbol{\omega}}_t \otimes \delta\mathbf{q}^r_t \tag{27}$$

where $\otimes$ represents composition of orientations and $\delta\mathbf{q}^{\boldsymbol{\omega}}_t$ and $\delta\mathbf{q}^r_t$ are quaternions representing the integration of the effect of the angular velocity and rotation noise, assumed constant during a sampling interval Δt:

$$\delta\mathbf{q}^{\boldsymbol{\omega}}_t = \mathbf{Q}(\boldsymbol{\omega}_t) \quad \text{and} \quad \delta\mathbf{q}^r_t = \mathbf{Q}(\boldsymbol{\xi}^{\boldsymbol{\omega}}_t) \tag{28}$$

with:

$$\mathbf{Q}(\mathbf{x}) = \left[\frac{\mathbf{x}}{|\,\mathbf{x}\,|} \sin\left(\frac{|\,\mathbf{x}\,|\,\Delta t}{2}\right) \quad \cos\left(\frac{|\,\mathbf{x}\,|\,\Delta t}{2}\right)\right] \tag{29}$$

Thus, the effect of the noise vector $\boldsymbol{\xi}^{\boldsymbol{\omega}}_t$ is considered as a random angular velocity disturbance and has a 3×3 covariance matrix $\mathbf{Q}^r_t$.

D. Unscented Bingham Filter

We apply a UBiF to the orientation part of the state vector. The angular velocities will be obtained from the orientation difference between iterations. For the rotation case, the system model is given by:

$$\mathbf{q}_{t+1} = \mathbf{F}(\mathbf{q}_t) \otimes \boldsymbol{\Phi}_t \tag{30}$$

where $\mathbf{q}_t$ is the orientation at time t, $\boldsymbol{\Phi}_t \sim P_B(\mathbf{M}^{\boldsymbol{\Phi}}_t, \mathbf{Z}^{\boldsymbol{\Phi}}_t)$ is the Bi (the used Bi distribution properties are described in Appendix A) distributed system noise and $\mathbf{F} : \mathbb{S}^3 \mapsto \mathbb{S}^3$ respecting the existing antipodal symmetry. The measurement model is represented as:

$$\mathbf{r}_t = \mathbf{H}(\mathbf{q}_t) \otimes \boldsymbol{\Lambda}_t \tag{31}$$

where $\mathbf{r}_t \in \mathbb{S}^3$ is the measurement at time t and $\boldsymbol{\Lambda}_t \sim P_B\left(\mathbf{M}^{\boldsymbol{\Lambda}}_t, \mathbf{Z}^{\boldsymbol{\Lambda}}_t\right)$ is the Bi distributed measurement noise. Function $\mathbf{H}$ relates the measurement $\mathbf{r}_t$ to the values of the

orientation $\mathbf{q}_t$ (identity function in our study case). Choosing $\mathbf{M}_t^{\mathbf{\Phi}} = \mathbf{M}_t^{\mathbf{\Lambda}} = \mathbf{I}_{4\times4}$ is equivalent to the concept of zero-mean noise in the *Euclidean* space [8], [9].

The implemented UBiF framework is separated into prediction and measurement update [9], [10], [31]. The prediction step is composed of the following phases:

- Use deterministic sampling to generate sigma points $\mathbf{q}_t^i$ based on the UT [36], [37] to approximate the current system state $\mathbf{q_t} \sim P_B\left(\mathbf{M}_t^e, \mathbf{Z}_t^e\right)$ (Appendix B);
- Propagate each one of the sigma points $\mathbf{q}_t^i$ according to the system dynamic $\mathbf{F}(.)$ (Equation 27);
- Compute the covariance matrix $\mathbf{C}_{\mathbf{F}(\mathbf{q}_t)} = Cov(\mathbf{F}(\mathbf{q}_t))$ (Equation 38) from the obtained sigma points after propagation $(\bar{\mathbf{q}}_{t+1}^i)$ and the covariance matrix $\mathbf{C}_{\mathbf{\Phi}} = Cov(\mathbf{\Phi}_t)$ from the respective Bi distribution (Equation 36);
- Use the obtained covariance matrices to compute $\mathbf{C}_{\bar{\mathbf{q}}_{t+1}} = Cov(\mathbf{F}(\mathbf{q}_t) \otimes \mathbf{\Phi}_t)$ utilizing the method of moments [10]. The covariance of the composition is the directional analog to the addition of random vectors in linear space;
- Estimate $\bar{\mathbf{M}}_{t+1}$ and $\bar{\mathbf{Z}}_{t+1}$ from the obtained covariance matrix $\mathbf{C}_{\bar{\mathbf{q}}_{t+1}}$ using a Maximum Likelihood Estimation (MLE) [8] (Appendix A).

The update step is composed of the following phases:

- Rotate noise according to measurement $P_B(\bar{\mathbf{q}}_{t+1}; diag(-1,-1,-1,1) \cdot \mathbf{M}_{t+1}^{\mathbf{\Lambda}} \otimes \mathbf{r}_{t+1}, \mathbf{Z}_{t+1}^{\mathbf{\Lambda}}) = P_B(\check{\mathbf{M}}, \mathbf{Z}_{t+1}^{\mathbf{\Lambda}})$ (Appendix C);
- The measurement update step is obtained from the *Bayes theorem* $P(\bar{\mathbf{q}}_{t+1}|\mathbf{r}_{t+1}) = \mathrm{C} \cdot P(\mathbf{r}_{t+1}|\bar{\mathbf{q}}_{t+1}) \cdot P(\bar{\mathbf{q}}_{t+1})$ (where C is a normalization constant). We have that $P(\mathbf{r}_{t+1}|\bar{\mathbf{q}}_{t+1}) = P_B(\check{\mathbf{M}}, \mathbf{Z}_{t+1}^{\mathbf{\Lambda}})$ and $P(\bar{\mathbf{q}}_{t+1}) = P_B(\bar{\mathbf{M}}_{t+1}, \bar{\mathbf{Z}}_{t+1})$;
- Obtain the estimate Bi distribution $P(\bar{\mathbf{q}}_{t+1}|\mathbf{r}_{t+1}) = P_B(\mathbf{M}_{t+1}^e, \mathbf{Z}_{t+1}^e) = P_B(\check{\mathbf{M}}, \mathbf{Z}_{t+1}^{\mathbf{\Lambda}}) \cdot P_B(\bar{\mathbf{M}}_{t+1}, \bar{\mathbf{Z}}_{t+1})$ (Equation 34).

The update quaternion $\mathbf{q}_{t+1}$ is obtained from the $P_B(\mathbf{M}_{t+1}^e, \mathbf{Z}_{t+1}^e)$ mode. The quaternion error is obtained by multiplying the previous quaternion $(\mathbf{q}_t)$ with the conjugate of the estimated one $(\bar{\mathbf{q}}_{t+1})$. The angular velocities are them obtained converting to the angle-axis representation.

VI. Image-based resampling

One of the most critical steps in the PF is the resampling [15] that eliminates particles with low importance weights (based on the image information) and replicates particles having high importance weights (Figure 4). It is important to take into account that during resampling the particle diversity is reduced, and this could lead to a local minimum estimate [7], [18]. To overcome this, we also use the pre-trained database (Section IV) to be able to use the most recent frame information and add diversity to the particle set on each filter iteration similar to that used in the Boosted PF [38].

The applied resampling strategy was the resampling reallocation. As described in [7], the resampling reallocation is the traditional resampling strategy that obtained better performance in the problem at hand. The resampling step is always applied to the entire particle set on each filter iteration. It is also used a prediction (move) step (artificial noise as described in Figure 4) to increase the diversity of the particles after the resampling step [20], [39]. This step can be performed by adding Gaussian noise to the particle state [18], [40], [41].

Fig. 8. Created synthetic video environment.

VII. Experimental results

To test the developed system (Figure 4), we have created a synthetic video landing sequence to be able to generate ground-truth data for quantitative analysis of the results (Figure 8). The method was implemented in C++ on a 3.70 GHz Intel i7-8700K Central Processing Unit (CPU) and NVIDIA Quadro P5000 Graphics Processing Unit (GPU). All the results presented in this section refer to this platform.

During landing, the UAV is approaching the ship so we will use the database (Section IV) trained in the front hemisphere $(U\left(-90°, 90°\right))$. In the first filter iteration, all the used 100 particles come from the database. The velocities are randomly initialized with values near zero. The particles are evaluated using the color likelihood metric as described in Section II. Every time we capture a new frame, we update the particle vector with 25 new particles coming from the database to be able to use the most recent observation and to maintain particle diversity. The rest of the particles of the set are obtained from the last iteration using a resampling step (Section VI). The added noise (artificial noise as described in Figure 4) is Gaussian of zero mean and covariance 0.1 meters for the translation in X and Y, 0.2 meters for the translation in Z and 2,62 rad/sec for the angular velocity (Equation 28). In the performed tests, we use a Bi process noise $P_B^{\mathbf{\Phi}}$ with $\mathbf{M}_t^{\mathbf{\Phi}} = \mathbf{I}_{4\times4}$ and $\mathbf{Z}_t^{\mathbf{\Phi}} = diag$(-250,-250,-250,0) and a Bi measurement noise $P_B^{\mathbf{\Lambda}}$ with $\mathbf{M}_t^{\mathbf{\Lambda}} = \mathbf{I}_{4\times4}$ and $\mathbf{Z}_t^{\mathbf{\Lambda}} = diag(-800,-800,-800,0)$. We filter each one of the particles on each iteration. On each iteration, the state expectation is given by the particle with the highest weight. The time between iterations is $\Delta t = 0.034\ s/iteration$.

The translation error was obtained by the coordinate difference, and the orientation error was obtained by calculating the quaternion error $\mathbf{q}_e$ according to:

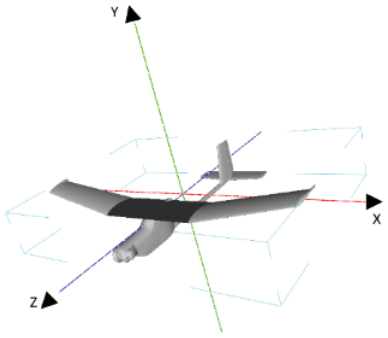

Fig. 9. UAV model axis representation.

TABLE I
OBTAINED TRANSLATION (METERS) AND ROTATION (DEGREES) ERROR.

	X	Y	Z	α	β	γ
5% Percentile	-0.01	-0.01	-0.07	-7.45	-5.03	-2.63
25% Percentile	0.05	0.01	0.85	-2.87	0.81	-0.82
Median	0.16	0.03	1.91	-0.77	6.08	-0.12
75% Percentile	0.26	0.06	3.58	1.32	11.16	0.46
95% Percentile	0.40	0.12	6.50	4.75	19.12	2.77
MAE	0.18	0.04	2.61	2.88	8.01	1.19
RMSE	0.23	0.06	4.07	3.87	10.32	2.19

$$\mathbf{q}_e = \mathbf{q}_g \otimes \bar{\mathbf{q}}_r \tag{32}$$

where $\mathbf{q}_g$ corresponds to our ground-truth quaternion and $\bar{\mathbf{q}}_r$ corresponds to the conjugate of the obtained hypothesis quaternion. Each quaternion error $\mathbf{q}_e$ is then converted to *Euler* angles to analyze the error on each angle independently (α represents the rotation around X, β represents the rotation around Y, and γ represents the rotation around Z as described in Figure 9). We have also obtained the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) [42] for each coordinate and angle independently (Table I).

The resulting translation estimates (Figure 10) present good results with 90% of the error between $[-0.01; 0.40]$ meters for X, $[-0.01; 0.12]$ meters for Y and $[-0.07; 6.50]$ meters for Z (Table I). The higher Z error is mainly due to the large initial tracking error as seen in Figure 10. As the UAV approaches the landing area, the error decreases and becomes close to zero. Given the available landing area (5×6 meters), the obtained Z error does not compromise a successful landing maneuver.

The resulting rotation estimates (Figure 11) present good results with 90% of the error between $[-7.45; 1.32]$ degrees for α, $[-5.03; 19.12]$ degrees for β and $[-2.63; 2.77]$ degrees for γ (Table I). The worst estimate is given by the β estimative (Figure 11), but the obtained error does not compromise the control process during landing. The attitude control is essential since the UAV should make a predefined landing trajectory to deal with the wind vortices created by the FPB superstructure.

In this work, we have not introduced coupling between the dynamics in the translation and rotational models yet because the dynamics of the UAV is currently more thoroughly analyzed. These constraints would further improve the dynamics model and thus the precision of the estimates.

VIII. Conclusions and Future Work

A method was introduced and tested for tracking of a known UAV for automatic landing. The presented algorithms feature a UAV detection method based on a cascade classifier, a pose initialization methodology with a pre-trained database and a PF with motion filtering performed using a UKF and a UBiF.

We consider the achieved precision levels suitable for automated landing. In the following stages of the work, we will focus on the improvement of the used UAV motion model and in the pose estimation using Deep Neural Networks (DNNs).

Appendix A
Bingham Distribution

The Bi distribution is an antipodally symmetric distribution (opposite points on S have equal probability) that represents a zero-mean Gaussian distribution in R^d projected on the unit hypersphere S^{d-1} [8], [9], [43]. The Probability Density Function (PDF) for the Bi distribution is obtained by [8]:

$$P_B(\mathbf{q}; \mathbf{M}, \mathbf{Z}) = \frac{1}{F(\mathbf{Z})} \exp^{(\mathbf{q}^T \mathbf{M} \mathbf{Z} \mathbf{M}^T \mathbf{q})} \tag{33}$$

where $\mathbf{q} \in S^{d-1} \subset \mathbb{R}^d : ||\mathbf{q}|| = 1$ is a unit vector (when using quaternions $d = 4$), $\mathbf{M} \in \mathbb{R}^{d \times d}$ is an orthogonal matrix describing the mean orientation of the distribution, $F(\mathbf{Z})$ is the normalization constant and $\mathbf{Z} = diag(z_1, z_2, ..., z_{d-1}, 0)$ with nondecreasing negative diagonal elements is the concentration matrix. We conveniently force the last entry to be zero to be possible to swap the columns of $\mathbf{M}$ without changing the corresponding distribution and to simplify the needed computation [31]. The parameters of $\mathbf{Z}$ control the spread of the Bi distribution around its mean direction, and the orientation matrix $\mathbf{M}$ control the mean direction itself. Adding a multiple of the identity matrix to $\mathbf{Z}$ or changing the order of a column of $\mathbf{M}$ and the according z value does not change the distribution. As referred before, we enforce the last entry of $\mathbf{Z}$ to be zero, and because of this, the mode of the distribution is simply obtained from the last column of $\mathbf{M}$ [8], [31], [44]. The PDF represented in Equation 33 is antipodally symmetric since $P_B(\mathbf{q}) = P_B(-\mathbf{q})$. This property is suitable for the use in the unit quaternion space since $\mathbf{q}$ and $-\mathbf{q}$ represent the same orientation. The main difficulty in the utilization of the Bi distribution consists of the computation of the normalization constant since the distribution must integrate to one over its domain. Since we are using this distribution in a real-time approach, we choose to interpolate tabulated values from a precomputed lookup table for computational efficiency.

The product of two given Bi distributions is closed under multiplication after renormalization and is given by [45]:

$$P_{B_1} \cdot P_{B_2} = \frac{1}{F(\mathbf{Z})} \exp^{(\mathbf{q}^T \mathbf{M} \mathbf{Z} \mathbf{M}^T \mathbf{q})} \tag{34}$$

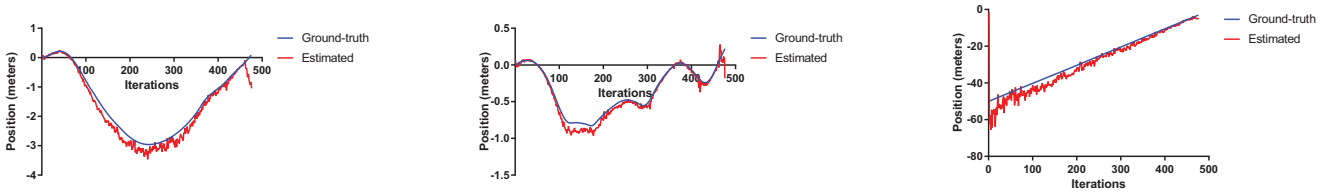

Fig. 10. Obtained translation error (meters): X (*left*), Y (*center*) and Z (*right*).

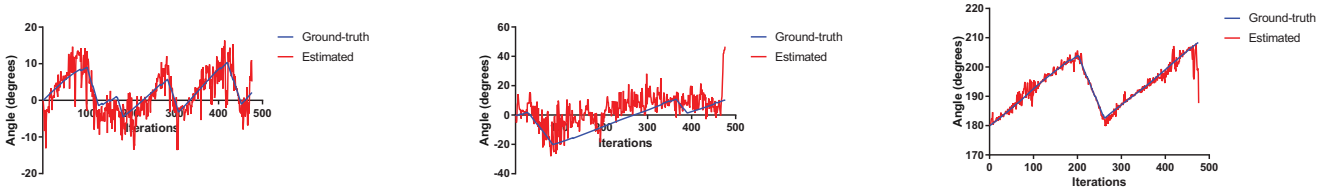

Fig. 11. Obtained rotation error (degrees): α (*left*), β (*center*) and γ (*right*).

where $F(\mathbf{Z})$ is the new normalization constant, $\mathbf{M}$ are the unit eigenvectors of $(\mathbf{M}_1\mathbf{Z}_1\mathbf{M}_1^T + \mathbf{M}_2\mathbf{Z}_2\mathbf{M}_2^T)$, $\mathbf{Z} = \mathbf{D} - \lambda_{dd}\mathbf{I}_{d\times d}$ where $\mathbf{D}$ has the eigenvalues of $\mathbf{M}$ on the diagonal in ascending order and λ_{dd} is the largest eigenvalue.

It is possible to change the orientation of a Bi distribution $\mathbf{q} \sim P_B(\mathbf{M}, \mathbf{Z})$ by a fixed quaternion $\mathbf{g} \in S^3$ according to [9], [10], [31]:

$$\mathbf{r} \sim P_B(\mathbf{M} \otimes \mathbf{g}, \mathbf{Z}) \quad \text{when} \quad \mathbf{r} = \mathbf{q} \otimes \mathbf{g} \tag{35}$$

where $\mathbf{M} \otimes \mathbf{g} \equiv [\mathbf{m}_1 \otimes \mathbf{g}, \mathbf{m}_2 \otimes \mathbf{g}, \mathbf{m}_3 \otimes \mathbf{g}, \mathbf{m}_4 \otimes \mathbf{g}]$ ($\mathbf{m}$ are the eigenvectors of $\mathbf{M}$). Since the quaternion multiplication is not commutative, we have that $\mathbf{r} \sim P_B(\mathbf{g} \otimes \mathbf{M}, \mathbf{Z})$ when $\mathbf{r} = \mathbf{g} \otimes \mathbf{q}$.

The covariance of the Bi PDF is given by:

$$Cov(\mathbf{q}) = E(\mathbf{q}\mathbf{q}^T) - (E(\mathbf{q}))^2 = E(\mathbf{q}\mathbf{q}^T) \tag{36}$$

where $(E(\mathbf{q}))^2 = 0$ is a consequence of the antipodal symmetry and $E(\mathbf{q}\mathbf{q}^T)$ is given by:

$$E(\mathbf{q}\mathbf{q}^T) = \mathbf{M} \cdot diag\left(\frac{\frac{d}{dz_1}F(\mathbf{Z})}{F(\mathbf{Z})}, \ldots, \frac{\frac{d}{dz_d}F(\mathbf{Z})}{F(\mathbf{Z})}\right) \cdot \mathbf{M}^T \tag{37}$$

where the values of the gradient of F with respect to $\mathbf{Z}$ are precomputed and accessed by interpolation as made for the normalization constant. The covariance of the composition (directional analog to the addition of random vectors in linear space) of two Bi distributions can be obtained using the method of moments [10]. The covariance is a sufficient statistics (no other statistic that can be calculated provides any additional information) for the Bi distribution since the Bi distribution is the maximum entropy distribution on the hypersphere which matches the sample inertia matrix [46].

It is important to be able to estimate the parameters of a Bi distribution which approximates a set of samples [8]. The inertia matrix for a set of N samples $\mathbf{q} = [\mathbf{q}_1, ..., \mathbf{q}_N]$ is given by [8]:

$$\mathbf{S} = E[\mathbf{q}\mathbf{q}^T] = Cov(\mathbf{q}) = \frac{1}{N}\sum_i q_i q_i^T \tag{38}$$

The MLE $\hat{\mathbf{M}}$ for a set of samples is an eigenvalue problem since the columns of $\hat{\mathbf{M}}$ are equal to the eigenvectors of $\mathbf{S}$. The MLE $\hat{\mathbf{Z}}$ can be found setting the partial log-likelihood function on $\mathbf{Z}$ to zero. This calculation is made using the Constrained Optimization BY Linear Approximations (COBYLA) algorithm [47]. This algorithm constructs successive linear approximations of the objective function and constraints via a simplex of $n+1$ points (in n dimensions) and optimizes these approximations in a trust region at each step.

APPENDIX B
BINGHAM FILTER SIGMA POINTS CREATION

The created sigma points follow the same principle as used in the UT applied in the UKF for the translation case (Section V-B). We need to use $4d-2$ samples that correspond in this case to fourteen samples ($d = 4$). Since the distribution is antipodally symmetric, it is sufficient to consider only one pole (adapting the respective weights). The canonical sigma points are given by [9], [37]:

$$\tilde{\mathbf{q}}^{1,2} = [\pm\sin\alpha_1\, 0\, 0\, \cos\alpha_1] \tag{39}$$

$$\tilde{\mathbf{q}}^{3,4} = [0\, \pm\sin\alpha_2\, 0\, \cos\alpha_2] \tag{40}$$

$$\tilde{\mathbf{q}}^{5,6} = [0\, 0\, \pm\sin\alpha_3\, \cos\alpha_3] \tag{41}$$

$$\tilde{\mathbf{q}}^{7} = [0\, 0\, 0\, 1] \tag{42}$$

where $\tilde{\mathbf{q}}^7$ is the sample located on the pole (identity quaternion). The canonical distribution will be employed since it simplifies the needed mathematical approach because the parameters will be dimensionless [36], [37]. For example, if

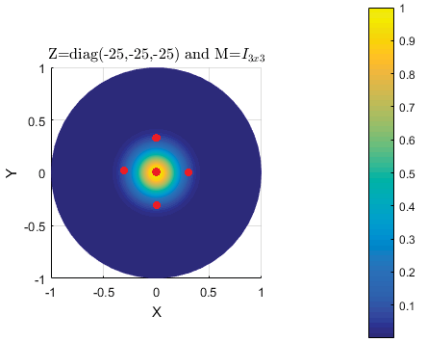

Fig. 12. An example of deterministic sampling with $d = 3$.

we have $d = 3$ we will need ten sigma points to approximate our Bi distribution, and the half corresponding to one pole will be located as shown in Figure 12.

The covariance of the estimated Bi distribution is obtained by (Equation 37):

$$E_{P_B}\left\{\mathbf{x}_t\mathbf{x}_t^T\right\} = \mathbf{M}\cdot diag(f_1, f_2, f_3, f_4)\cdot\mathbf{M}^T \tag{43}$$

The deviation for each one of the canonical sigma points is obtained from α_i:

$$\alpha_i = \sin^{-1}\left(\sqrt{\frac{f_i}{w_{B_i}}}\right) = \sin^{-1}\sqrt{\left(\frac{3f_i}{3f_i + \left(1-\frac{1}{N}\right)f_4}\right)} \tag{44}$$

The weights of the sigma points are given by:

$$w^{1,2} = \frac{w_{B_1}}{4} = \frac{f_1 + \frac{1-\frac{1}{N}f_4}{3}}{4} \tag{45}$$

$$w^{3,4} = \frac{w_{B_2}}{4} = \frac{f_2 + \frac{1-\frac{1}{N}f_4}{3}}{4} \tag{46}$$

$$w^{5,6} = \frac{w_{B_3}}{4} = \frac{f_3 + \frac{1-\frac{1}{N}f_4}{3}}{4} \tag{47}$$

where N is equal to the number of used sigma points. The weight for the central sigma point is obtained by:

$$w^7 = \frac{f_4}{N} \tag{48}$$

Each canonical sigma point $\tilde{\mathbf{q}}$ is multiplied by $\mathbf{M}$ (in the UBiF we use $\mathbf{M}_t^e\tilde{\mathbf{q}}$) originating the set of sigma points $\mathbf{q}$ that represent our P_B. The sigma points propagation is made adding a quaternion motion based on the angular velocities with some added noise to each one of the sigma points.

Appendix C
Rotate Bingham noise according to measurement

Given the actual measurement $\mathbf{r}_{t+1}$, the noise is rotated according to the measurement so that the Bi distributed measurement noise $P_B\left(\mathbf{M}_{t+1}^{\mathbf{\Lambda}}, \mathbf{Z}_{t+1}^{\mathbf{\Lambda}}\right)$ peak is aligned with the obtained measurement with spread controlled by $\mathbf{Z}_{t+1}^{\mathbf{\Lambda}}$ (Section V-D). Taking into account that:

$$P(\mathbf{r}_{t+1}\,|\,\bar{\mathbf{q}}_{t+1}) = P_B(\bar{\mathbf{q}}_{t+1}^{-1}\otimes\mathbf{r}_{t+1}; \mathbf{M}_{t+1}^{\mathbf{\Lambda}}, \mathbf{Z}_{t+1}^{\mathbf{\Lambda}}) \tag{49}$$

is equal to:

$$P_B(diag(-1,-1,-1,1)\cdot\bar{\mathbf{q}}_{t+1}\otimes\mathbf{r}_{t+1}; \mathbf{M}_{t+1}^{\mathbf{\Lambda}}, \mathbf{Z}_{t+1}^{\mathbf{\Lambda}}) \tag{50}$$

we obtain (Equation 35):

$$P_B(\bar{\mathbf{q}}_{t+1}; diag(-1,-1,-1,1)\cdot\mathbf{M}_{t+1}^{\mathbf{\Lambda}}\otimes\mathbf{r}_{t+1}, \mathbf{Z}_{t+1}^{\mathbf{\Lambda}}) \tag{51}$$

Acknowledgment

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Quadro P5000 GPU used for this research.

References

[1] S. Saripalli, "Vision-based autonomous landing of an helicopter on a moving target," in *Proceedings of AIAA Guidance, Navigation, and Control Conference, Chicago, USA*, 2009.

[2] K. E. Wenzel, A. Masselli, and A. Zell, "Automatic take off, tracking and landing of a miniature uav on a moving carrier vehicle," *Journal of intelligent and robotic systems*, vol. 61, no. 1-4, pp. 221–238, 2011.

[3] W. Xiang, Y. Cao, and Z. Wang, "Automatic take-off and landing of a quad-rotor flying robot," in *Control and Decision Conference (CCDC), 2012 24th Chinese*. IEEE, 2012, Conference Proceedings, pp. 1251–1255.

[4] Y. J. Zhao and H. L. Pei, "Improved vision-based algorithm for unmanned aerial vehicles autonomous landing," *Applied Mechanics and Materials*, vol. 273, pp. 560–565, 2013.

[5] N. Pessanha Santos, F. Melicio, V. Lobo, and A. Bernardino, "A ground-based vision system for uav pose estimation," *International Journal of Mechatronics and Robotics (IJMR) - UNSYSdigital International Journals*, vol. 1, no. 4, p. 7, 2014. [Online]. Available: http://ojs.unsysdigital.com/index.php/ijrm/article/view/180

[6] N. Pessanha Santos, V. Lobo, and A. Bernardino, "A ground-based vision system for uav tracking," in *OCEANS 2015 - Genova*, 2015, Conference Proceedings.

[7] ——, "Particle filtering based optimization applied to 3d model-based estimation for uav pose estimation," in *OCEANS 2017 - Aberdeen*, 2017, Conference Proceedings.

[8] C. Bingham, "An antipodally symmetric distribution on the sphere," *The Annals of Statistics*, pp. 1201–1225, 1974.

[9] I. Gilitschenski, G. Kurz, S. J. Julier, and U. D. Hanebeck, "Unscented orientation estimation based on the bingham distribution," *IEEE Transactions on Automatic Control*, vol. 61, no. 1, pp. 172–177, 2016.

[10] J. Glover and L. P. Kaelbling, "Tracking 3-d rotations with the quaternion bingham filter," *Computer Science and Artificial Intelligence Laboratory - Technical Report*, 2013.

[11] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[12] D. Wang, F. Yang, K.-L. Tsui, Q. Zhou, and S. J. Bae, "Remaining useful life prediction of lithium-ion batteries based on spherical cubature particle filter," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 6, pp. 1282–1291, 2016.

[13] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. Wan, "The unscented particle filter," in *Advances in neural information processing systems*, 2001, Conference Proceedings, pp. 584–590.

[14] Z. Wang, X. Zhao, and X. Qian, "Unscented particle filter with systematic resampling localization algorithm based on rss for mobile wireless sensor networks," in *Mobile Ad-hoc and Sensor Networks (MSN), 2012 Eighth International Conference on*. IEEE, 2012, Conference Proceedings, pp. 169–176.

[15] M. Boli, P. M. Djuri, and S. Hong, "Resampling algorithms for particle filters: a computational complexity perspective," *EURASIP J. Appl. Signal Process.*, vol. 2004, pp. 2267–2277, 2004.

[16] Y.-M. Chan, S.-S. Huang, L.-C. Fu, P.-Y. Hsiao, and M.-F. Lo, "Vehicle detection and tracking under various lighting conditions using a particle filter," *IET intelligent transport systems*, vol. 6, no. 1, pp. 1–8, 2012.

[17] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *Radar and Signal Processing, IEE Proceedings F*, vol. 140. IET, 1993, Conference Proceedings, pp. 107–113.

[18] A. J. Haug, *Bayesian Estimation and Tracking: A Practical Guide*. John Wiley and Sons, 2012.

[19] Y. Rui and Y. Chen, "Better proposal distributions: Object tracking using unscented particle filter," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2. IEEE, 2001, Conference Proceedings, pp. II–II.

[20] S. S. Haykin *et al.*, *Kalman filtering and neural networks*. Wiley Online Library, 2001.

[21] J. Zhou, S. Knedlik, and O. Loffeld, "Ins/gps tightly-coupled integration using adaptive unscented particle filter," *The Journal of Navigation*, vol. 63, no. 3, pp. 491–511, 2010.

[22] M. Birsan, "Unscented particle filter for tracking a magnetic dipole target," in *OCEANS, 2005. Proceedings of MTS/IEEE*. IEEE, 2005, pp. 1656–1659.

[23] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li, "Learning multi-scale block local binary patterns for face recognition," in *International Conference on Biometrics*. Springer, 2007, Conference Proceedings, pp. 828–837.

[24] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 1, pp. 105–119, 2010.

[25] Y.-J. Cheon and J.-H. Kim, "Unscented filtering in a unit quaternion space for spacecraft attitude estimation," in *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*. IEEE, 2007, Conference Proceedings, pp. 66–71.

[26] J. L. Crassidis and F. L. Markley, "Unscented filtering for spacecraft attitude estimation," *Journal of guidance, control, and dynamics*, vol. 26, no. 4, pp. 536–542, 2003.

[27] S. J. Julier, "The scaled unscented transformation," in *American Control Conference, 2002. Proceedings of the*, vol. 6. IEEE, 2002, Conference Proceedings, pp. 4555–4559.

[28] E. Kraft, "A quaternion-based unscented kalman filter for orientation tracking," in *Proceedings of the Sixth International Conference of Information Fusion*, vol. 1, 2003, Conference Proceedings, pp. 47–54.

[29] R. D. Turner, "Gaussian processes for state space models and change point detection," phdthesis, University of Cambridge, 2012.

[30] J. Zhou, Y. Yang, J. Zhang, and E. Edwan, "Applying quaternion-based unscented particle filter on ins/gps with field experiments," *Proceedings of the ION GNSS, Portland*, pp. 1–14, 2011.

[31] G. Kurz, I. Gilitschenski, S. Julier, and U. D. Hanebeck, "Recursive bingham filter for directional estimation involving 180 degree symmetry," *Journal of Advances in Information Fusion*, vol. 9, no. 2, pp. 90–105, 2014.

[32] P. Li, T. Zhang, and A. E. Pece, "Visual contour tracking based on particle filters," *Image and Vision Computing*, vol. 21, no. 1, pp. 111–123, 2003.

[33] A. Doucet, N. De Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2000, Conference Proceedings, pp. 176–183.

[34] A. Doucet, N. De Freitas, and N. Gordon, *An introduction to sequential Monte Carlo methods*. Springer, 2001, pp. 3–14.

[35] N. J. Higham, "Analysis of the cholesky decomposition of a semi-definite matrix," *Manchester Institute for Mathematical Sciences School of Mathematics*, 1990.

[36] J. E. Darling and K. J. DeMars, "Analysis of the gauss-bingham distribution for attitude uncertainty propagation," in *AAS/AIAA Space Flight Mechanics Meeting, No. AAS*, 2015, Conference Proceedings, pp. 15–605.

[37] ——, "Rigid body attitude uncertainty propagation using the gauss-bingham distribution," in *AAS/AIAA Space Flight Mechanics Meeting, No. AAS*, 2015, Conference Proceedings, pp. 15–347.

[38] K. Okuma, A. Taleghani, N. d. Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," *Computer Vision-ECCV 2004*, pp. 28–39, 2004.

[39] R. Havangi, M. A. Nekoui, H. D. Taghirad, and M. Teshnehlab, "An intelligent ufastslam with mcmc move step," *Advanced Robotics*, vol. 27, no. 5, pp. 311–324, 2013.

[40] J. H. Kotecha and P. M. Djuric, "Gaussian particle filtering," *IEEE Transactions on signal processing*, vol. 51, no. 10, pp. 2602–2612, 2003.

[41] ——, "Gaussian sum particle filtering for dynamic state space models," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 6. IEEE, 2001, pp. 3465–3468.

[42] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.

[43] C. J. Fallaize and T. Kypraios, "Exact bayesian inference for the bingham distribution," *Statistics and Computing*, vol. 26, no. 1-2, pp. 349–360, 2016.

[44] G. Kurz, I. Gilitschenski, and U. D. Hanebeck, "Recursive nonlinear filtering for angular data based on circular distributions," in *American Control Conference (ACC), 2013*. IEEE, 2013, Conference Proceedings, pp. 5439–5445.

[45] J. Glover and L. P. Kaelbling, "Tracking the spin on a ping pong ball with the quaternion bingham filter," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, Conference Proceedings, pp. 4133–4140.

[46] K. Mardia, *Characterizations of directional distributions*. Springer, 1975, pp. 365–385.

[47] M. Powell, "Direct search algorithms for optimization calculations," *Acta numerica*, pp. 287–336, 1998.