RESEARCH

Nizar Ouarti[*??], Thomas Braud and Vivien Billaud

OpenMotion: An open source library for attitude estimation.

Abstract

The abstract goes here

Keywords: IMU; Sensor Fusion; Sensor Calibration

1 Introduction

2 Global Presentation of the Library

`OpenMotion` is a open source library mainly aimed for real-time attitude estimation. Our approach is based on data fusion technics using only the output of a IMU (composed by a 3D gyroscope, 3D accelerometer and a 3D magnetometer). Among several methods proposed by the library, the user has the possibilities to choose which methods fits the best to his expectation according to its performance. Here the 2 reference frame to remember.

- The North East Down (NED) frame $\{a\}$ system has its origin fixed at the (moving) object center of gravity. The z-axis points upward perpendicularly to the tangent plane of the ellipsoid, and the x-axis points towards true north (and not the magnetic north). The y-axis point towards east.
- The object-fixed reference frame $\{c\}$ corresponding to the IMU device. It is a moving and rotating coordinate frame.

For simplicity reasons, the library admit that all the components of the IMU belong to the same reference frame $\{c\}$ (ref figure 1). It is a cyclopean approximation [1]. Moreover, the user has to interface the sends of data from the device to the library because the library does not take care of it. `OpenMotion` provide an estimation of the orientation of $\{c\}$ relative to $\{a\}$. The choice of the output type (quaternion, rotation matrix, euler angle) is according to the user needs.

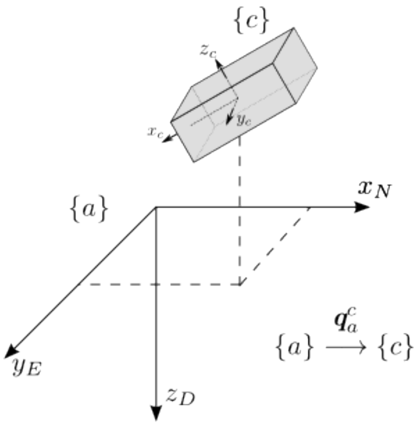


Figure 1: Definition of the scene with 2 frame: a fixed frame NED (North East Down) noted $\{a\}$ and a moving frame object noted $\{c\}$

We are also working on the establishment of a dynamic sensor calibration process on all component of the IMU. We insist on the dynamic aspect. Indeed, firstly, it allows the library to be adaptable to any kind of IMU, but also to improve the performance without requesting some effort from the user.

3 Calibration and initialization

3.1 Problematique de la calibration

Les systemes embarque comme les smartphones disposent pour la plupart de capteurs servant au systeme ou a l'utilisateur pour se renseigner sur des grandeurs utiles quant a l'appareil lui-meme ou bien a son environnement. Ces capteurs sont donc varies mais on peut notamment distinguer les capteurs constituant une centrale inertielle ou IMU pour Inertial Measurement Unit que sont les accelerometres et les gyroscopes 3 axes. Ces capteurs sont de plus en plus associes avec un magnetometre. Ainsi, ces capteurs peuvent constituer un AHRS pour Attitude and Heading Reference System ayant pour but de reperer l'appareil dans l'espace 3D. L'accelerometre sert ici a mesurer l'acceleration lineaire, le gyroscope la vitesse angulaire et le magnetometre le champ magnetique terrestre local et sert dans la plupart des cas de compas. Dans le cas d'un smartphone, ces capteurs sont low-cost et de type MEMS pour Micro-Electro-Mechanical Systems. Des lors, le signal utile etant faible, l'erreur induite peut-etre du meme ordre de grandeur et donc conduire a des imprecisions importantes, notamment dans le cas

[*] Correspondence: nizar.ouarti@ipal.cnrs.fr

Image Pervasive Access Lab (Sorbonne UPMC, CNRS, Astar, NUS, UJF, IMT), Singapore

Full list of author information is available at the end of the article

des AHRS qui font l'objet d'une fusion de donnees. Il convient donc d'eliminer ou tout du moins de reduire ces erreurs via une calibration initiale de l'ensemble des capteurs utilises.

3.2 Types et modele d'erreur

Tous les types de capteurs presentent des biais, facteurs d'echelle (scale factors), des erreurs d'intercouplage (misalignements et cross axis sensitivities) et dans une certaine mesure du bruit aleatoire. Des erreurs d'ordre superieur, des erreurs d'inter-couplage d'acceleration angulaire ou encore des erreurs specifiques comme les softs et irons distorsions pour le magnetometre peuvent egalement se produire. Pour l'ensemble des capteurs, un modele rendant compte des erreurs lors des differentes mesures peut-etre propose :

$$X_i = MS \cdot (X_{i,raw} - \mathbf{b}) + \mathbf{n}_i \tag{1}$$

ou X_i est la mesure du capteur comprenant les erreurs, $X_{i,raw}$ la mesure reelle du capteur, b le biais, S la matrice rendant compte des scales factor errors, M la matrice des misalignement entre les axes du capteur et $\mathbf{n}_i$ le bruit aleatoire avec $i \in \{x, y, z\}$. Les differents matrices sont alors

$$S = \begin{pmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{yx} & s_{yy} & s_{yz} \\ s_{zx} & s_{zy} & s_{zz} \end{pmatrix} \text{ et } M = \begin{pmatrix} 1 & -\alpha_{yz} & \alpha_{zy} \\ 0 & 1 & -\alpha_{zx} \\ 0 & 0 & 1 \end{pmatrix}$$

ou α_{ij} avec $i, j \in \{x, y, z\}$ est l'angle de misalignement entre l'axe i du repere du capteur avec celui de la plateforme selon l'axe j de la plateforme. Enfin le biais est represente par un vecteur $\mathbf{b} \in \mathbb{R}^3$

3.3 Methodes adoptees

Une premiere calibration d'usine est effectuee mais s'avere souvent imprecise ou obsolete du fait de l'usure des capteurs. Des lors, une nouvelle calibration s'avere necessaire.

La calibration des capteurs se fera dans un premier temps independamment les unes des autres et de maniere statique emphi. e. a l'initialisation. Il est a noter que nous ne prendrons pas en compte ici l'influence de la temperature puisqu'il est deja traite dans la calibration d'usine.

3.3.1 Accelerometre

Freescale propose une methode de calibration lineaire de l'accelerometre 3 axes. Elle repose sur le fait que la somme des amplitudes des composantes normalisees du champ gravitationnel terrestre sur les 3 axes du capteur doit equationetre egale a l'amplitude totale de celui-ci quand le capteur est dans un etat quasi statique. Elle consiste a placer l'appareil dans 8 positions donnees pour chaque axe puis d'utiliser une methode d'optimisation de moindres carres pour determiner les 12 parametres de calibration que sont les scale factors et le biais.

Cette methode ne permet toutefois pas de determiner les misalignements.

3.3.2 Gyroscope

Zhang propose une methode de calibration du gyroscope[2]. Elle repose sur le fait que la somme des mesures selon les 3 axes du gyroscope doit equationetre egale a la vitesse angulaire en entree.

2 etapes sont alors necessaires a la calibration du gyroscope. La premiere consiste en des rotations horaires et anti-horaires permettant d'eliminer la vitesse angulaire de rotation de la terre et le biais du gyroscope en prenant le carre de la norme de la difference des integrales des mesures dans les 2 sens puis en utilisant une methode de moindre carre. Les scales factors et les angles de misalignements sont ainsi determines. Ensuite, les angles de misalignements et les scales factors sont utilises pour estimer le biais via la difference des carres de l'equation de la somme des vitesses angulaires pour 2 positions differentes.

3.3.3 Magnetometre

Vasconcelos et al. [3] proposent un algorithme dit de maximum likelihood estimation (MLE) qui utilise l'estimation issue de "Two-Step" [4], *i.e.* une methode d'estimation de moindre carre suivie d'un filtrage etendu de Kalman (EKF), comme approximation initiale et qui traite le bruit de mesure comme une distribution normale.

Le probleme de minimisation la fonction de vraisemblance (likelihood) qui est quadratique revient alors a determiner les points de l'ellipsoide qui correspondent au mieux aux mesures du capteur.

Ce probleme est resolu via la methode du gradient et une methode de Newton descendante pour des espaces euclidiens [5] et la regle de Armijo pour la determination de la taille du pas. Le gradient et la matrice hessienne de la fonction de log-likelihood sont eux calcules analytiquement.

3.3.4 Calibration on-board

Une calibration on-board peut egalement s'averer necessaire du fait d'un besoin de precision accrue ou de l'obsolescence de la calibration d'usine due a l'usure des capteurs.

4 State of the art

4.1 Quaternion representation

A quaternion is a four-dimensional vector, defined as

$$\boldsymbol{q} = \begin{pmatrix} \mathbf{q} \\ q_w \end{pmatrix} \text{ with } \mathbf{q} = (q_x, q_y, q_z)^T.$$

For attitude representation, the quaternion must satisfies a single constraint given by $||\boldsymbol{q}|| = 1$. The attitude matrix $R \in SO(3)$ related to the quaternion $\boldsymbol{q}$ is given by :

$$R(\boldsymbol{q}) = (q_w^2 - |\mathbf{q}|^2)I + 2\mathbf{q}\mathbf{q}^T - 2q_w[\mathbf{q}\times] \tag{2}$$

Where $SO(3)$ is the special orthogonal group defined by:

$$SO(3) = \{R \in \mathbb{R}^{3\times3} | RR^T = R^T R = I, det(R) = 1\}$$

And $[\mathbf{v}\times] \in \mathbb{R}^{3\times3}$ is the skew-symetric matrix of the vector $\mathbf{v}$ such as.

$$[\mathbf{v}\times] = \begin{pmatrix} 0 & v_z & -v_y \\ -v_z & 0 & v_x \\ v_y & -v_x & 0 \end{pmatrix} \tag{3}$$

4.2 Most Common Fusion Methods

4.2.1 Kalman Filter

Introduced in 1960 by Kalman [6], the Kalman filter is an algorithm that uses a series of measurements observed over time, containing noise, in order to estimate unknown variables (state vector). Based on a model it is possible to obtain a more robust estimation of the state vector. Consider the following linear system, described by the difference equation and the observation model with additive noise:

$$\begin{cases} \mathbf{x}_{k+1} = F_k \mathbf{x}_k + B_k \mathbf{u}_k + G_k \mathbf{v}_k \\ \mathbf{z}_k = H_k \mathbf{x}_k + D_k \mathbf{w}_k \end{cases} \tag{4}$$

with,

$\mathbf{x}_k$	State vector at time k
$\mathbf{z}_k$	Measurement vector at time k
$\mathbf{u}_k$	Input vector (e.i.: from sensor) at time k
$\mathbf{v}_k$	State noise vector disturbing the system
$\mathbf{w}_k$	Observation noise disturbing the measurement

A brief presentation of the algorithm is provided here, see [7] for more details. The algorithm has 3 important steps :

- The first step is the **prediction** of state $\mathbf{x}_{k+1}^-$ at time $k+1$ and covariance P_{k+1}^- is given by :

$$\begin{cases} \mathbf{x}_{k+1}^- = & F_k \mathbf{x}_k^+ + B_k \mathbf{u}_k \\ P_{k+1}^- = & F_k P_k^+ F_k^T + G_k Q_k G_k^T \end{cases} \tag{5}$$

- The **innovation** is the difference between predicted measurement and real measurement. It measure up the deviation used by the Kalman gain.

$$\begin{cases} \boldsymbol{\nu}_{k+1} = & \mathbf{z}_{k+1} - H_{k+1} \mathbf{x}_{k+1}^- \\ \mathcal{S} = & R_{k+1} + H_{k+1} P_{k+1}^- H_{k+1}^T \end{cases} \tag{6}$$

- And finally, An **update** is processed to correct the estimate state $\mathbf{x}_{k+1}^+$ and his covariance P_{k+1}^+ with the innovation.

$$\begin{cases} \mathbf{x}_{k+1}^+ = & \mathbf{x}_{k+1}^- + \mathcal{K}_{k+1} \boldsymbol{\nu}_{k+1} \\ P_{k+1}^+ = & P_{k+1}^- - \mathcal{K}_{k+1} S_{k+1} \mathcal{K}_{k+1}^T \end{cases} \tag{7}$$

Where $\mathcal{K}_{k+1}$ is the Kalman gain representing relative importance of innovation ν_{k+1}.

$$\mathcal{K}_{k+1} = P_{k+1}^- H_{k+1}^T \mathcal{S}^{-1} \tag{8}$$

The Kalman filter provides an optimal estimation for linear systems. Nevertheless, most of studied dynamical systems are nonlinear (like attitude estimation problem). Thus, extended Kalman Filter approach was proposed for nonlinear systems [8, 9, 10].

Here, we have chosen to use quaternions for the attitude representation because of its singular-free specificity. But special care must be taken when designing a quaternion based extended Kalman filter. Because of the constraint on the quaternions, the covariance matrix P can be singular [11] which is problematic for the numerical stability. There exist several methods to deal with this issue. Multiplicative Extended Kalman Filter (MEKF)[12] and the Additive Extended Kalman Filter (AEKF) [13] are designed to tackle this problem.The difference between AEKF and MEKF is discussed in [14]. For the library, we choose to implement MEKF.

On the other hand, the Unscented Kalman Filter method was proposed by Julier and Uhlmann [15] in 1995 . It belongs to a class of filters called Sigma-Point Kalman Filters. Instead of using the Taylor series expension, UKF use an unscented transformation [16] to approximate the nonlinear projection of mean and covariance of a probability distribution. For the library, we choose to implement the filter proposed by Crassidis et al. called UnScented QUaternion Estimator (USQUE) [17], based on UKF. They used the same representation as MEKF.

4.2.2 Nonlinear Observer

The theory of the state observer was first introduced by Kalman and Bucy for a linear system in a stochastic environment. Then Luenberger [18] made a general theory of observers for deterministic linear systems, introducing the notions of observer and reduced minimum observer [19]. Mahony et al. [20] have proposed an nonlinear observer called Constant Gain Observer (CGO), termed the explicit complementary filter, that provides attitude estimates as well as gyro bias estimates. One problem of Kalman Filters is to find the initial values in the covariance (matrix Q) because it is difficult to relate it to real physical quantities. The covariance matrix is not involve in nonlinear observer theory. This is one of the reasons there has been an increasing interest for nonlinear observer design during the last decades. Another reason is their low computational need.

4.2.3 Methods Based on Solutions of the Wahba's Problem

The attitude determination problem for vector observation was first formulated as a least squares estimation problem by Wahba [21] in 1965 :

Given the two sets of n vectors $\{\mathbf{r}_1, ..., \mathbf{r}_n\}$ and $\{\mathbf{b}_1, ..., \mathbf{b}_n\}$, $n \geqslant 2$, where each pair $(\mathbf{r}_i, \mathbf{b}_i)$ corresponds to a generalised vector, $\mathbf{x}_i$, find the proper orthogonal matrix, A, which bring the first into the best least squares coincidence with the second. That is, find $A \in SO(3)$ which minimises the function J defined as:

$$J(A) = \frac{1}{2}\sum_{i=1}^{n} a_i ||b_i - Ar_i||^2 \tag{9}$$

The equation (9) can be written with a quaternion representation which gives:

$$J(\boldsymbol{q}) = \frac{1}{2}\sum_{i=1}^{n} a_i ||b_i - R(\boldsymbol{q})r_i||^2 \tag{10}$$

The matrix $R(\boldsymbol{q})$, is the rotation matrix that corresponds to the quaternion $\boldsymbol{q}$ (equation 2). The Quatenion Estimator (QUEST) algorithm propose a solution to the Wahba's problem. Let :

$$B = \sum_{i=1}^{N} a_i \mathbf{b}_i \mathbf{r}_i^T \tag{11}$$

Then, we can rewrite the equation (10) as follow

$$J(\boldsymbol{q}) = \frac{1}{2}\sum_{i=1}^{n} a_i - tr(R(\boldsymbol{q})B^T) \tag{12}$$

Due to the fact that the representation of the attitude matrix is a homogenous quadratic function of $\boldsymbol{q}$, we can say that

$$tr(R(\boldsymbol{q})B^T) = \boldsymbol{q}^T K \boldsymbol{q} \tag{13}$$

where K is the symmetric traceless matrix such as

$$K = \begin{pmatrix} S - tr(B)I_3 & \mathbf{z} \\ \mathbf{z}^T & tr(B) \end{pmatrix} \tag{14}$$

with

$$\begin{cases} S = B + B^T \\ \mathbf{z} = \sum_i a_i \mathbf{b}_i \times \mathbf{r}_i \end{cases} \tag{15}$$

The minimization problem from equation (12) is now a maximization problem. thus, the solution $\boldsymbol{q}_{opt}$ is the eingen vector correspond to the highest eigen value of K [22]. The main problem of QUEST is that it does not take into account of all the previous observations performed since the beginning. Then, a REcursive-QUEST algorithm (REQUEST) proposed by Bar-Itzhack[23] manages this problem. Contrary to other methods, solutions to Wahba's problem do not estimate the gyroscope bias. Which can make it less robust in some cases. There are also other methods based on Wahba's problem like the Davenport's q-method [24] proposed in 1971 considered more robust than QUEST. Or the method ESOQ proposed by Mortari [25] in 1997. However, for this library, we limit our investigation to REQUEST.

4.2.4 Particle Filtering Method

Particle filtering method is based on Monte-Carlo simulation[26]. The main idea is to use a large number of samples called particles to estimate a nonlinear function [27, 28]. The set of N particle with associated weight at time k are denoted by $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}$. Initially, the particle are drawn from a proposal distribution. The PF is divided into three step, prediction, update, resampling, that constitute a filter cycle:

- Prediction

The first step is to propagate all particles $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}$ to $\{\mathbf{x}_{k+1}^{(i)}, w_{k+1}^{(i)}\}$ with the nonlinear function f while the weight remain unchanged :

$$\mathbf{x}_{k+1}^{(i)} = f(\mathbf{x}_k^{(i)}, \mathbf{u}_k) + \mathbf{w}_k^{(i)} \tag{16}$$

where N sample $\mathbf{w}_k^{(i)}$ of the process noise (supposed gaussian in our case) are drawn.

- Update/Correction

At the update step, the weights associated with each particle is updated based on the likelihood function $p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}^{(i)})$.

$$\begin{cases} w_{k+1}^{(i)} = w_k^{(i)} p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}^{(i)}) \\ \tilde{w}_{k+1}^{(i)} = \frac{w_{k+1}^{(i)}}{\sum_i w_{k+1}^{(i)}} \end{cases} \tag{17}$$

The weight are normalised such as we have $\sum_i \tilde{w}_{k+1}^{(i)} = 1$. To finish, the mean is computed with the following equation:

$$\mathbf{x}_{k+1}^{+} = \sum_{i=1}^{N} \tilde{w}_{k+1}^{(i)} \mathbf{x}_{k+1}^{(i)} \tag{18}$$

- Resampling/Regularisation

A common problem with PF is the degeneracy phenomenon. After some iterations, more and more particles will have a weak weight, which make them negligible and useless. To overcome this problem, a regularisation step called resampling is processed. It is not needed to perform PF but it allows to maintain a good performance relative to degeneracy problem. The idea of resampling is to eliminate particles having feeble weight and focus on strong weight particles. In order to measure degeneracy, the number of effective samples N_{eff} is computed (equation 19). If N_{eff} is lower than a threshold, the resampling step is processed.

$$N_{eff} = \sum_{i=1}^{N} \frac{1}{(\tilde{w}_k^{(i)})^2} \tag{19}$$

The choice of the threshold is important. Indeed, a high threshold gives $N_{eff} \approx N$, but that does not mean necessarily we have a lot of efficient particles. On the contrary, it means that most of the particle are identical making large computations unnecessary. Moreover, with a low threshold, many particles are neglected making computations less efficient. In general, the threshold is fixed at $\frac{N}{2}$ or $\frac{N}{3}$. There are several types of resampling, a comparison is made in [29]. We took the resampling algorithm described in [30] because it is the most commun. Cheng, Yang and Crassidis have propose an attitute estimation with a particle filter method [31]. We have also considered the revisited algorithm of Cheng [32] based on bootstrap filter[33]. We have chosen to implement this one because they use the same approach as USQUE and MEKF. The particle filter does not assume that the measurement noise is Gaussian. Which makes it more adaptable in the context of a real applications. Nevertheless an important computation time is required.

5 Performance Evaluation

5.1 Score definition

We have access through the ground truth (simulated or from computer vision technics) to the true attitude $\boldsymbol{q}_k^{real}$ at time k of the reference frame $\{c\}$ related to $\{a\}$. By computation, each method returns an estimate attitude $\boldsymbol{q}_k^{est}$ at time k.

$$\delta\boldsymbol{q}_k = \boldsymbol{q}_k^{real} \otimes (\boldsymbol{q}_k^{est})^{-1} = \begin{pmatrix} \delta\mathbf{q}_k \\ \delta q_{w_k} \end{pmatrix} \tag{20}$$

The error-MRP vector related to $\delta\boldsymbol{q}_k$ is given by

$$\delta\mathbf{p}_k = 4[\frac{\delta\mathbf{q}_k}{(1+\delta q_{w_k})}] \tag{21}$$

Thus, we can calculate the attitude error given by the root mean square (RMS) of $\delta\mathbf{p}_k$:

$$\epsilon_k = RMS(\delta\mathbf{p}_k) = \sqrt{\frac{1}{3}(\delta p_{k_x}^2 + \delta p_{k_y}^2 + \delta p_{k_z}^2)} \tag{22}$$

The mean of the error is given by :

$$\bar{\epsilon} = \frac{1}{n}\sum_{i=1}^{n} \epsilon_i \tag{23}$$

With n the number of iteration at our disposal. We use this information (mean error) for our performance study. To differentiate the methods, we assigned some scores in order to highlight the behaviour of these methods in different configurations defined as:

$$s_1 = \bar{\epsilon} \tag{24}$$

$$s_2 = \frac{1}{\bar{\epsilon} \times \bar{\tau}} \tag{25}$$

With $\bar{\epsilon}$ average attitude error (see 22) et $\bar{\tau}$ the average computation time.

5.2 Simulated environment

We modelled our inertial sensors based on the Fossen's model [34]. The noise generated here is Gaussian. In this study, sensor bias are not considerate as constant. We proceed 30 consecutive simulations in order to get valid results with data performance specification described in table 1.

Sensor	Bias instability	Measurement noise
Gyroscope	0.316×10^{-4} μrad.s$^{-\frac{3}{2}}$	0.316 μrad.s$^{-\frac{1}{2}}$
Magnetometer	0.0023 mGauss/$\sqrt{Hz}$	0.05 mGauss
Accelerometer	0.002 m.s$^{-2}/\sqrt{Hz}$	0.02 m.s^{-2}

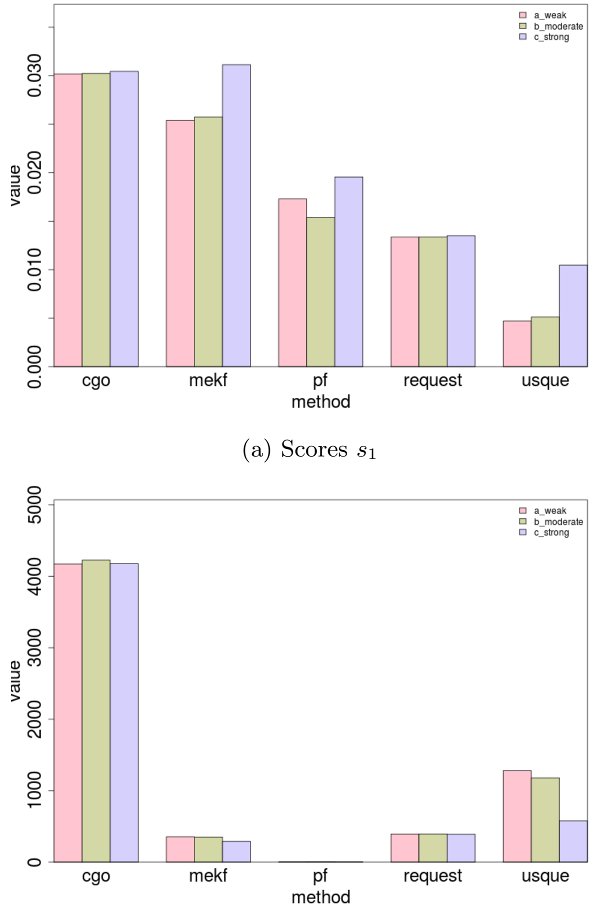
Table 1: Calibrated IMU data performance specification, Gyrometer, Accelerometer, Magnetometer

All simulations started with an initial attitude error $\delta \boldsymbol{q}_0$ randomly generated through normal distribution. The results of score s_1 (resp s_2) are shown in figure 2a (resp 2b). Mean computation times of all methods required for one iteration is given in table (2).

Method	Mean Computation time
CGO	$7.88 \times 10^{-6}s$
MEKF	$1.12 \times 10^{-4}s$
USQUE	$1.63 \times 10^{-4}s$
REQUEST	$1.91 \times 10^{-4}s$
PF	$4.5 \times 10^{-2}s$

Table 2: Mean computation times for one iteration according to the methods available in the library

The library has been entierely coded in C and all tests have been launched on a Dell Optiplex 9020 with Intel $Core^{TM}$ i7-4790 CPU at 3.60GHz × 8. All curves were generated with R and Matlab. No paralelization or code optimization have been done, our program turns entirely in sequential manner. Some optimization will come in the next versions.



(a) Scores s_1

(b) Scores s_2

Figure 2: Scores s_1 (accuracy) and s_2 (accuracy and computation time ratio) of methods where noise is AWGN with 3 strength (weak, moderate, strong)

5.3 Experimental validation

We have designed a system that can be easily repeatable in laboratory. We used a smartphone as our IMU and a calibrated webcam (see figure 4). With this system, we have defined our ground-truth as follows: Some markers are stuck on the smartphone and we calculate an estimate of the quaternion attitude $\boldsymbol{q}_c^w$ of the markers (which represents the smartphone) relative to the reference frame of the webcam $\{w\}$. To do so, we've used the library ArUco [35] as optical feedback. At the same time, we get the data from the IMU of the smartphone to proceed the attitude estimation with our different algorithms. At time $t = 0$, the camera and the smartphone have been set such as attitude quaternions $\boldsymbol{q}_w^a$, $\boldsymbol{q}_a^c$ et $\boldsymbol{q}_c^w$ are approximatively known. The camera being fixed, $\boldsymbol{q}_w^a$ does not vary. Moreover, from figure 4 we have:

$$\boldsymbol{q}_a^c = (\boldsymbol{q}_w^c \otimes \boldsymbol{q}_w^a)^{-1} \tag{26}$$

All methods give an approximation of $\boldsymbol{q}_a^c$, that are compared with the data from the camera. To estimate the error that can occur with the optical feedback. We were careful in the experimentation to avoid the occlusion of the marker that is required to estimate the orientation. During this experiment, we applied a rotation of 90 degree around the z_c on the smartphone in clockwise direction. Then, another rotation of 90 degree around the z_c on the smartphone but in anticlockwise direction to return to the initial state. In or-

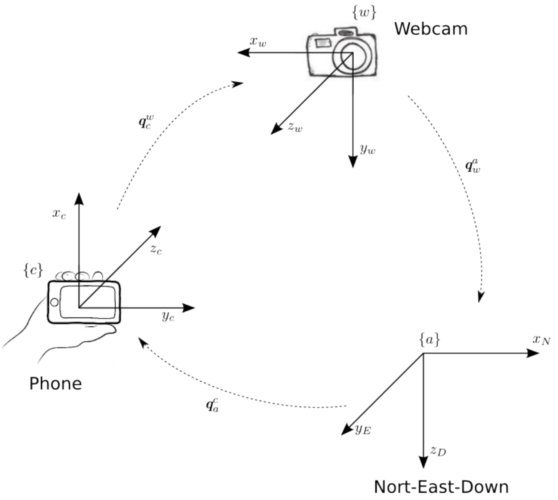
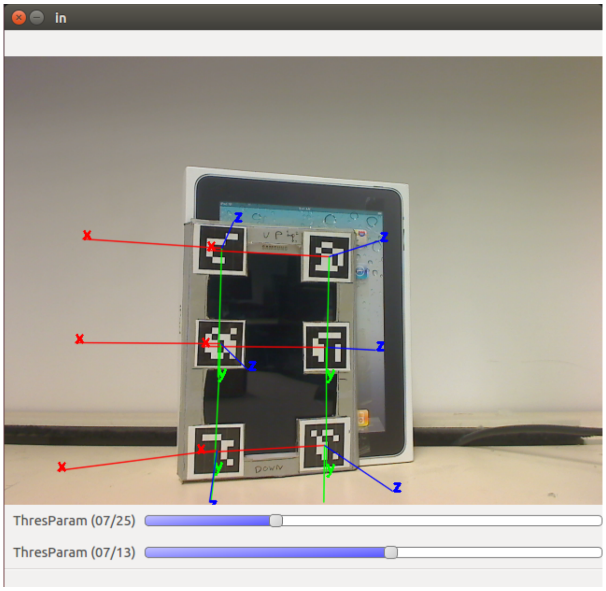


Figure 3: Representation of different reference frame used in this experiment. Here, only q_w^a does not change.



ThresParam (07/25)

ThresParam (07/13)

Figure 4: exemple of output of `ArUco` with 6 markers. Here the object-fixed reference frame $\{c\}$ is drawn

der to study possible divergence. The results is shown in figure ...

6 Conclusion

notre library est trop bien...

Competing interests

The authors declare that they have no competing interests.

Author's contributions

Text for this section ...

Acknowledgements

This work was supported by the ARAP program of ASTAR and by the university Sorbonne UPMC.

References

1. Ouarti, N., Berthoz, A.: Multimodal fusion in self-motion perception using wavelets and quaternions. In: Deuxième Conférence Française de Neurosciences Computationnelles," Neurocomp08" (2008)
2. Zhang, H., Wu, Y., Wu, W., Wu, M., Hu, X.: Improved multi-position calibration for inertial measurement units. Measurement Science and Technology **21**(1) (2009)
3. Vasconcelos, J.F., Elkaim, G., Silvstre, C., Oliveira, P., Cardeira, B.: Geometric approach to strapdown magnetometer calibration in sensor frame. IEEE Trans. Aerosp. Electron. Syst. **47**(2), 1293–1306 (2011)
4. Gebre-Egziabher, D., Elkaim, G.H., Powell, J.D., Parkinson, B.W.: A non-linear, two-step estimation algorithm for calibrating solid-state strapdown magnetometers. In: Proc. 8th Int. St. Petersburg Conf. Navig. Syst., pp. 290–297 (2001)
5. Bertsekas, D., 2nd edn. Athena Scientific (1999)
6. Kalman, R.E.: A new approach to linear filtering and prediction problems. Journal of Fluids Engineering **82**(1), 35–45 (1960). Accessed 2015-08-24
7. Terejanu, G.A.: Discrete kalman filter tutorial. University at Buffalo, Department of Computer Science and Engineering, NY **14260** (2013)
8. Larson, R.E., Dressler, R.M., Ratner, R.S.: Application of the extended kalman filter to ballistic trajectory estimation. Technical report, DTIC Document (1967)
9. Larson, R., Dressler, R., Ratner, R.: Precomputation of the weighting matrix in an extended kalman filter. In: Joint Automatic Control Conference, pp. 634–645 (1967)
10. Terejanu, G.A.: Extended kalman filter tutorial. Online]. Disponible: http://users. ices. utexas. edu/˜ terejanu/files/tutorialEKF. pdf (2008)
11. Vik, B.: Integrated satellite and inertial navigation systems. Department of Engineering Cybernetics, NTNU (2009)
12. Markley, F.L.: Attitude error representations for kalman filtering. Journal of guidance, control, and dynamics **26**(2), 311–317 (2003)
13. Bar-Itzhack, I.Y., Deutschmann, J., Markley, F.: Quaternion normalization in additive ekf for spacecraft attitude determination. AIAA paper, 91–2706 (1991)
14. Markley, F.L.: Multiplicative vs. additive filtering for spacecraft attitude determination. In: Proceedings of the 6th Conference on Dynamics and Control of Systems and Structures in Space (DCSSS), vol. 22 (2004)
15. Julier, S.J., Uhlmann, J.K., Durrant-Whyte, H.F.: A new approach for filtering nonlinear systems. In: American Control Conference, Proceedings of The 1995, vol. 3, pp. 1628–1632. IEEE, ??? (1995)
16. Uhlmann, J.K.: Dynamic map building and localization: New theoretical foundations. PhD thesis, University of Oxford (1995)
17. Crassidis, J.L., Markley, F.L.: Unscented filtering for spacecraft attitude estimation. Journal of guidance, control, and dynamics **26**(4), 536–542 (2003). Accessed 2015-07-02
18. David, G.: An introduction to observers. IEEE Transactions on automatic control **16**(6), 596–602 (1971)
19. Primbs, J.: Survey of nonlinear observer design techniques. Citeseer (1996)
20. Mahony, R., Hamel, T., Pflimlin, J.-M.: Nonlinear complementary filters on the special orthogonal group. Automatic Control, IEEE Transactions on **53**(5), 1203–1218 (2008). Accessed 2015-07-02
21. Wahba, G.: A least squares estimate of satellite attitude. SIAM review **7**(3), 409–409 (1965). Accessed 2015-09-03
22. Markley, F.L., Mortari, D.: How to estimate attitude from vector observations. In: Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, vol. 103, pp. 1979–1996 (1999)
23. Bar-Itzhack, I.Y.: REQUEST-A recursive QUEST algorithm for sequential attitude determination. Journal of Guidance, Control, and Dynamics **19**(5), 1034–1038 (1996). Accessed 2015-08-27
24. Davenport, P.B.: Attitude determination and sensor aligment via weighted least squares affine transformations (1971)
25. Mortari, D.: Esoq: A closed-form solution to the wahba problem. Journal of the Astronautical Sciences **45**(2), 195–204 (1997)

26. Metropolis, N., Ulam, S.: The monte carlo method. Journal of the American statistical association **44**(247), 335–341 (1949)
27. Chen, Z.: Bayesian filtering: From Kalman filters to particle filters, and beyond. Statistics **182**(1), 1–69 (2003). Accessed 2015-07-02
28. Terejanu, G.A.: Tutorial on monte carlo techniques. Department of Computer Science and Engineering. University at Buffalo, Buffalo, NY **14260** (2009)
29. Douc, R., Cappé, O.: Comparison of resampling schemes for particle filtering. In: Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium On, pp. 64–69 (2005). IEEE
30. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. Signal Processing, IEEE Transactions on **50**(2), 174–188 (2002)
31. Cheng, Y., Crassidis, J.L.: Particle filtering for attitude estimation using a minimal local-error representation. Journal of guidance, control, and dynamics **33**(4), 1305–1310 (2010). Accessed 2015-07-02
32. Chang, L.: Particle Filtering for Attitude Estimation Using a Minimal Local-Error Representation: A Revisit. arXiv preprint arXiv:1411.6127 (2014). Accessed 2015-07-02
33. Gordon, N.J., Salmond, D.J., Smith, A.F.: Novel approach to nonlinear/non-gaussian bayesian state estimation. In: IEE Proceedings F (Radar and Signal Processing), vol. 140, pp. 107–113 (1993). IET
34. Fossen, T.I.: Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley & Sons, ??? (2011)
35. Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F.J., Marín-Jiménez, M.J.: Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognition **47**(6), 2280–2292 (2014)