Image Registration Using Single Cluster PHD Methods

Mark Campbell[*1], **Isabel Schlangen**[1], **Emmanuel Delande**[1], and **Daniel Clark**[2]

[1]*School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh, United Kingdom*

[2]*Télécom SudParis, Institut Mines-Télécom, France*

Abstract

When telescopes are exploited for the observation of orbiting objects, images are often distorted by diurnal motion and also by the motion of the imaging apparatus during acquisition, causing a significant drift across image sequences. The drift of normally static objects, such as the stars in the background, can be exploited to correct the effect of the drift and recalibrate the sequence of images. This paper presents recent developments in multi-target detection and tracking techniques, exploiting the single cluster Probability Hypothesis Density (PHD) filter, in order to jointly estimate the static objects and the sensor drift. A comparison on the correction of the sensor drift is carried out between the PHD filter, the Second-Order Probability Hypothesis Density (SO-PHD) filter, the Cardinalized Probability Hypothesis Density (CPHD) filter, and the pattern matching, an established technique for the correction of image drift in astronomical data.

I Introduction

Over the last sixty years, the number of objects in the near-Earth space has increased substantially. There are over $22,000$ currently being tracked by US Strategic Command [1] although this represents a small amount of the population of objects of interest in orbit. The need for autonomous detection and tracking systems is of great importance as even the smallest debris can damage functional Resident Space Objects (RSOs) such as spacecraft and communication satellites. The hypothesized scenario, Kessler Syndrome [2], proposes that the density of objects in Low Earth Orbit (LEO) could rise to a point, where collisions between objects could cause a cascading effect with each collision increasing the probability of further collisions due to the release of more debris. Space Situational Awareness (SSA) involves the tracking and maintenance of the orbits of these RSOs to a high accuracy to mitigate the chance of collisions and ensure the safety of future missions. RSOs are usually observed using either electro-optical or radio-frequency sensors [3].

Single sensor calibration is an issue that plagues the SSA community, as sensors are imperfect pieces of equipment. In the case of optical sensors, when these sensors are used with amateur telescope mounting equipment jitter is often an unavoidable source of noise. This jitter [4] can be caused by a number of sources: mechanical imperfections in the mount, movement caused by the operator, atmospheric perturbations [5] and uncompensated diurnal motion to name a few. Minimizing this jitter is of significant importance in a astronomical target tracking scenario as even a small shift in the pointing direction of the imaging system can mean a large shift in the image scene. This would result in a large error when the astrometric coordinates of possible targets are measured. One of the most common methods for astronomical image registration employs pattern matching techniques. First presented in [6] it consists of matching points in two lists by constructing triangles between the detected objects in each list. Several extensions to this original method have been implemented since [7, 8]. This pattern matching method is widely used due to the fact that it is invariant with respect to scale, rotation and translation and can tolerate random errors and distortions.

With the increase of performance and decrease in cost of Complementary Metal Oxide Semiconductor (CMOS) sensors, their exploitation in the context of SSA has become increasingly popular. Consumer Off-The-Shelf (COTS) cameras and optics are inexpensive solutions and provide a picture of the sky with a relatively wide Field of View (FoV); several sky-survey systems exploiting a network of those exist as detailed in [9, 10, 11] with further examples in [12]. A central challenge in the exploitation of these sensors, however, is the extraction of

[*]This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/K014277/1, the MOD University Defence Research Collaboration (UDRC) in Signal Processing and UDRC Enabling contract ED TIN2-10.

multiple detections from a single image due to the wide FoV: some detections can stem from moving objects, some others from statics objects (e.g., stars), some of them may be artefacts from the extraction process or *false alarms*. The association between the objects and the detections is unknown, as are the number of objects, both static and moving, in the FoV. In these conditions, exploiting the output of a sensor in order to estimate the number and trajectories of the detected objects becomes a multi-object estimation problem.

This paper explores the exploitation of multi-object tracking algorithms developed from the Finite Set Statistics (FISST) framework [13] in order to jointly estimate the sensor drift and the objects' states in a sequence of image produced from an optical sensor. This approach aims at proposing a more robust framework for sensor drift correction than the pattern matching techniques, as it can handle missed detections, false alarms, and can be used in challenging situations with a restricted number of static objects on which to calibrate the sensor's state. The joint calibration and tracking method exploits a Simultaneous Localisation and Mapping (SLAM) based approach [14]. While previously proposed calibration methods [15, 14, 16] have used a single motion model for the objects, the proposed approach is specifically adapted to the context of SSA and distinguishes static objects (stars) from moving objects (satellites), the former being primarily used for the estimation of the sensor's drift across the sequence of images. This estimated sensor drift can be be applied to the images to obtain a registered dataset. An example of this image registration can be seen in [17], where it was used to register microscopy images. The first iteration of the work presented in this paper can be found in [18], which employed the PHD filter [19]. In this paper, the SO-PHD filter [20] and the CPHD filter [21] are employed as well. A more general form of the single cluster PHD employed here can be seen in [22].

This paper is organized as follows. Sec. II presents the principle of joint multi-object filtering and sensor state estimation, and introduced the three multi-object filters exploited in the paper. Sec. III presents the target and sensor models employed for the implementation of the algorithms. Sec. IV describes the simulation tests and discusses the results. Finally, Sec. V concludes.

II Joint Multi-Object Filtering and Sensor State Estimation

II.1 Hierarchical processes

The image registration method detailed in this section exploits a SLAM based approach [14], in which the sensor drift and the detected objects' states are jointly estimated from a sequence of images. There are three sources of uncertainty to be estimated, represented by two different *processes*:

- The sensor process Ψ estimates the sensor state: in this application, the relative drift of the images from the initial frame of the sequence,
- The static object process Φ estimates the number and states of static objects: in this application, the stars in the background of the image sequence,

The estimated sensor drift is reversely applied to the detection points extracted from the images to negate the effect of the drift, and thus influences the estimation of the objects' states. Conversely, the estimation of the "motion" of the static objects throughout the image sequence provides a lot of insight on the sensor drift. The joint estimator is then built as a single cluster point process [23], in which the parent/daughter hierarchy between the sensor process Ψ and the static object process Φ is depicted in Fig. 1, where both the sensor and multi-object state have been reduced to two dimensions for the ease of illustration.

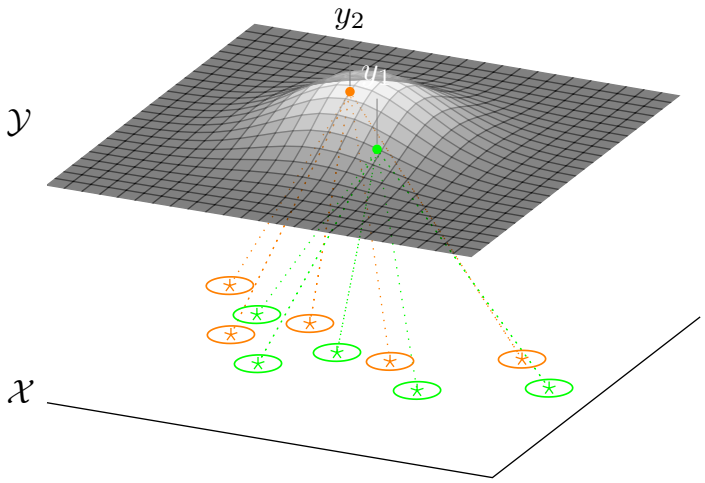


Figure 1: The hierarchical structure of the algorithm (illustrated in 2D). The sensor state space $\mathcal{Y}$ is overlaid by the probability distribution of the sensor position. A specific multi-object distribution of the static objects in the target state space $\mathcal{X}$ is maintained for each possible sensor position (green and orange colours for sensor positions y_1 and y_2, respectively). The asterisks and ellipses illustrate the means and covariances of the objects, extracted from the multi-object distributions.

Since the number and states of the objects (whether static or moving) is assumed unknown and possibly varying across the image sequence, the object process Φ is estimated through multi-object filters derived from the FISST framework: in the scope of this paper, the PHD filter [13], the SO-PHD filter [20] [24], and the CPHD filter [21]. These three solutions are explained in more details in Sec. II.2.

II.2 Multi Object Filtering

This section focuses on the estimation of the static object process Φ As seen in Sec. II.1, the estimation of the object process is conditioned upon the sensor state (i.e. drift in the image sequence). In practical terms, a multi-object filter estimating the static object process is maintained for each possible sensor state y. In this section, y_k denotes an arbitrary sensor state[1] at time k (i.e., in the k-th image of the sequence). The elements of the modelling phase fr the static object process can be seen in Sec. III. The following describes the approach for the object filtering. Since this algorithm can be extended to any number of motion models, the superscript (m) is used for quantities whose nature depends on the object process.

Each object is described by its state x in the *(single) target state space* $\mathcal{X} \subseteq \mathbb{R}^d$, describing the physical characteristics of the object. Static objects are described only with their position in the image frame, while moving objects are described also with their speed and heading in the image plane, but the dependency of the state space to the object process will be omitted from now on for the sake of simplicity. An object may enter or leave the sensor FoV at any time during the image sequence, and thus the number of objects at any time is unknown and needs to be estimated. The multi-object state is represented by a Random Finite Set (RFS) Φ_k, a random object whose size and elements are unknown, and whose realization is a set of target states $X_k = \{x_1, x_2, ..., x_{n_k}\}$ represents a specific *multi-object configuration* at time k [21]. The evolution of the objects' state between time step $k-1$ and k is supposed Markovian, and described by a Markov transition function $t_{k|k-1}$, obviously dependent on the object process since Φ , which describes static objects,. The number and states of newborn objects is described by a RFS $\Phi_{\mathrm{b},k}$ whose nature depends on the filter; its *first-order moment density* or *intensity* is denoted by $\mu_{\mathrm{b},k}(\cdot)$.

An observation collected from the sensor is described by a state z in the *observation space* $\mathcal{Z} \subseteq \mathbb{R}^d$. The set of collected observation is denoted by Z_k, which contains observations of all the targets in the scene, be they static or dynamic. The observation process is plagued by observation noise, missed detections, and false alarms. The observation noise associated to some collected observation z is characterized by the *likelihood* function $\ell_{z,k}(\cdot|y)$, while the probability of detection for each individual object in the sensor FoV is denoted by $p_{\mathrm{d},k}(\cdot|y)$. The number and states of false alarms is described by a RFS $\Phi_{\mathrm{c},k}$ whose nature depends on the filter; its intensity is denoted by $\mu_{\mathrm{c},k}(\cdot|y)$, and its spatial distribution is denoted by $s_{\mathrm{c},k}(\cdot|y)$.

We shall now describe the mechanisms of the filter estimating thestatic object process, depending on whether a PHD, SO-PHD, or CPHD filter is used. In all cases, the principle is to propagate some reduced information describing the object process through tractable filtering equations; the nature of the propagated information and

[1] the estimator associated to the sensor state will be presented later in Sec. II.5

the underlying assumptions characterize a specific filter. For the rest of the paper, the notation $\mu(\mathcal{X})$ will be used for various intensity functions μ to denote the integral $\int_{\mathcal{X}} \mu(x)\mathrm{d}x$.

II.2.1 Probability Hypothesis Density (PHD) Filter

The PHD filter [13] was designed as an inexpensive filtering solution in the the context of multi-object filtering, propagating only the intensity μ_k of the object process Φ_k. A single iteration of the PHD recursion is shown in Fig. 2.

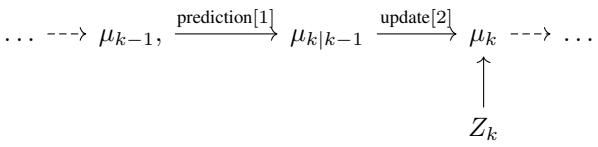


Figure 2: Data flow of the PHD filter at time k.

The key assumptions of the PHD filter is that both the predicted process $\Phi_{k|k-1}$ and false alarm process $\Phi_{\mathrm{c},k}$ are Poisson [13]. The prediction and update steps of the PHD filter are given by

$$\mu_{k|k-1}(x|y) = \mu_{\mathrm{b},k}(x) + \mu_{\mathrm{s},k|k-1}(x|y), \tag{1}$$

$$\mu_k(x|y) = \mu_{\phi,k}(x|y) + \sum_{z\in Z_k} \frac{\mu_{z,k}(x|y)}{\mu_{\mathrm{c},k}(z|y) + \mu_{z,k}(\mathcal{X}|y)}, \tag{2}$$

where the *survival* $\mu_{\mathrm{s},k|k-1}(x|y)$, *missed detection* $\mu_{\phi,k}(x|y)$, and *association* $\mu_{z,k}(x|y)$ terms are defined as

$$\mu_{\mathrm{s},k|k-1}(x|y) = \int p_{\mathrm{s},k}(\bar{x}) t_{k|k-1}(x|\bar{x}) \mu_{k-1}(\bar{x}|y)\mathrm{d}\bar{x}, \tag{3}$$

$$\mu_{\phi,k}(x|y) = (1 - p_{\mathrm{d},k}(x|y))\, \mu_{k|k-1}(x|y), \tag{4}$$

$$\mu_{z,k}(x|y) = p_{\mathrm{d},k}(x|y) \ell_{z,k}(x|y) \mu_{k|k-1}(x|y). \tag{5}$$

II.3 Second-Order Probability Hypothesis Density (SO-PHD) Filter

While the PHD filter is a popular solution to the multi-target tracking problem due to its low complexity and ability to estimate the states of targets in cluttered environments, it relies on Poisson assumptions which can be restrictive in more challenging scenarios involving high fluctuation in the number of objects and false alarms. Recently, a second-order version of the PHD filter was introduced in [20], propagating not only the intensity μ_k of the object process Φ_k, but also the variance $\mathrm{var}_k(\mathcal{X})$ of the number of objects in the whole target state space $\mathcal{X}$. A single recursion of the SO-PHD filter is shown in Fig. 3.

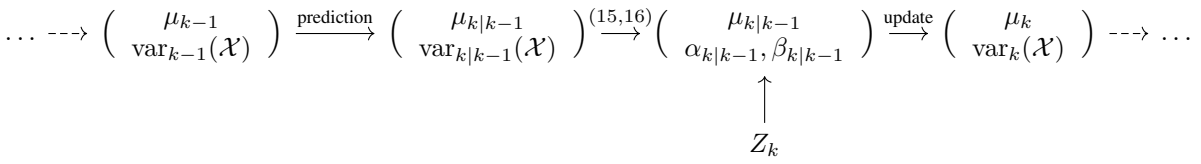


Figure 3: Data flow of the SO-PHD filter at time k. In addition to the intensity μ_k it propagates the scalar $\mathrm{var}_k(\mathcal{X})$, describing the variance in the estimated number of targets in the entire state space $\mathcal{X}$.

The SO-PHD filter substitutes Panjer assumptions to Poisson assumptions in the PHD filter, thus providing more flexibility in the description of the number of objects and false alarms. More specifically, the Panjer distribution [20] describes the Poisson, binomial and negative binomial distributions in a unified formulation involving two parameters α and β which stand in one-to-one correspondence with the mean and variance of the distribution. Using this property, the SO-PHD filter is able to propagate the variances in the number of objects and number of false alarms via the Panjer parameters of the corresponding distributions. From now on, the dependency of the Panjer parameters α, β to the object process (superscript (m)) and to the sensor state y will be omitted for the sake of simplicity.

Before stating the recursion of the SO-PHD filter, let us state some notations inspired by those introduced in [25] for the CPHD filter. The *Pochhammer symbol* or *rising factorial* $(\zeta)_n$ for any $\zeta \in \mathbb{R}$ and $n \in \mathbb{N}$ is defined with

$$(\zeta)_n := \zeta(\zeta+1)\cdots(\zeta+n-1), \quad (\zeta)_0 := 1. \tag{6}$$

Let $\alpha_{k|k-1}, \beta_{k|k-1}$ and $\alpha_{\mathrm{c},k}, \beta_{\mathrm{c},k}$ be the parameters of the predicted object process and clutter process at time k, respectively, and define the expression

$$Y_u^\flat[Z] := \sum_{j=0}^{|Z|} \frac{(\alpha_{k|k-1})_{j+u}}{(\beta_{k|k-1})^{j+u}} \frac{(\alpha_{\mathrm{c},k})_{|Z|-j}}{(\beta_{\mathrm{c},k}+1)^{|Z|-j}} F_{\mathrm{d}}^{-j-u} e_j^\flat(Z) \tag{7}$$

for any $Z \subseteq Z_k$ and for $u = 1, 2$, where F_{d} is the scalar

$$F_{\mathrm{d}} := \int \left[1 + \frac{p_{\mathrm{d},k}(x|y)}{\beta_{k|k-1}}\right] \mu_{k|k-1}(x|y)\mathrm{d}x, \tag{8}$$

and the so-called *elementary symmetric functions* [22] $e_j^\flat$ are given by

$$e_j^\flat(Z) := \sum_{\substack{Z' \subseteq Z \\ |Z'|=j}} \prod_{z \in Z'} \frac{\mu_{z,k}(\mathcal{X}|y)}{s_{\mathrm{c},k}(z|y)}. \tag{9}$$

Where $s_{c,k}$ denotes the spatial clutter distribution at time k. Again, the explicit dependency of the scalar F_{d}, the $Y_u^\flat$-term, and the elementary symmetric functions $e_j^\flat$, to the object process and sensor state have been omitted for the sake of simplicity. With the help of Eq. (7), define the *corrective terms* via

$$l_u^\flat(\phi|y) := \frac{Y_u^\flat[Z_k]}{Y_0[Z_k]} \quad \text{and} \quad l_u^\flat(z|y) := \frac{Y_u^\flat[Z_k \backslash \{z\}]}{Y_0^\flat[Z_k]} \tag{10}$$

for $u = 1, 2$, and

$$l_2^\flat(z, z'|y) := \begin{cases} \dfrac{Y_2^\flat[Z_k \backslash \{z, z'\}]}{Y_0^\flat[Z_k]} & \text{if} \quad z \neq z', \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Assuming that $p_{\mathrm{s},k}(x) = p_{\mathrm{s},k}$ is constant for all $x \in \mathcal{X}$ at time k, the prediction step of the SO-PHD filter is then given by

$$\mu_{k|k-1}(x|y) = \mu_{\mathrm{b},k}(x) + \mu_{\mathrm{s},k|k-1}(x|y), \tag{12}$$

$$\mathrm{var}_{k|k-1}(\mathcal{X}|y) = \mathrm{var}_{\mathrm{b},k}(\mathcal{X}) + \mathrm{var}_{\mathrm{s},k|k-1}(\mathcal{X}|y), \tag{13}$$

where the survival term for the variance $\mathrm{var}_{\mathrm{s},k|k-1}$ is given by

$$\mathrm{var}_{\mathrm{s},k|k-1}(\mathcal{X}|y) = (p_{\mathrm{s},k})^2 \mathrm{var}_{k-1}(\mathcal{X}|y) + p_{\mathrm{s},k}[1 - p_{\mathrm{s},k}]\mu_{k-1}(\mathcal{X}|y). \tag{14}$$

The variance in the number of newborn objects $\mathrm{var}_{\mathrm{b},k}$ is a parameter of the filter, and allows the operator to describe situations where the information on the number of objects entering the image frame.

The Panjer parameters of the predicted object process $\Phi_{k|k-1}$ are then given by [20]

$$\alpha_{k|k-1} = \frac{\mu_{k|k-1}(\mathcal{X}|y)^2}{\mathrm{var}_{k|k-1}(\mathcal{X}|y) - \mu_{k|k-1}(\mathcal{X}|y)}, \tag{15}$$

$$\beta_{k|k-1} = \frac{\mu_{k|k-1}(\mathcal{X}|y)}{\mathrm{var}_{k|k-1}(\mathcal{X}|y) - \mu_{k|k-1}(\mathcal{X}|y)}. \tag{16}$$

With these, the corrective terms $l_u^\flat$ (10), (11) can be computed, and the update step of the SO-PHD follows:

$$\mu_k(x|y) = \mu_{\phi,k}(x|y) l_1^\flat(\phi|y) + \sum_{z \in Z_k} \frac{\mu_{z,k}(x|y)}{s_{\mathrm{c},k}(z|y)} l_1^\flat(z|y). \tag{17}$$

$$\begin{aligned} \mathrm{var}_k(\mathcal{X}|y) = {} & \mu_k(\mathcal{X}|y) + \mu_{\phi,k}(\mathcal{X}|y)^2 \left[l_2^\flat(\phi|y) - l_1^\flat(\phi|y)^2\right] \\ & + 2\mu_{\phi,k}(\mathcal{X}|y) \sum_{z \in Z_k} \frac{\mu_{z,k}(\mathcal{X}|y)}{s_{\mathrm{c},k}(z|y)} \left[l_2^\flat(z|y) - l_1^\flat(\phi) l_1^\flat(z|y)\right] \\ & + \sum_{z, z' \in Z_k} \frac{\mu_{z,k}(\mathcal{X}|y)}{s_{\mathrm{c},k}(z|y)} \frac{\mu_{z',k}(\mathcal{X}|y)}{s_{\mathrm{c},k}(z'|y)} \left[l_2^\flat(z, z'|y) - l_1^\flat(z|y) l_1^\flat(z'|y)\right]. \end{aligned} \tag{18}$$

II.4 Cardinalized Probability Hypothesis Density (CPHD) Filter

The superscript $\sharp$ will be used in order to denote the quantities relevant to the CPHD filter. The CPHD filter was designed as a generalization of the PHD filter, in which the cardinality distribution $\rho_k^\sharp$ of the object process Φ_k is propagated alongside its intensity $\mu_k^\sharp$ [21]. The CPHD is therefore a generalization of the SO-PHD as well, propagating more information on the number of objects, though incurring a significant increase in the computational cost [20]. A single recursion of the CPHD filter is shown in Fig. 4.

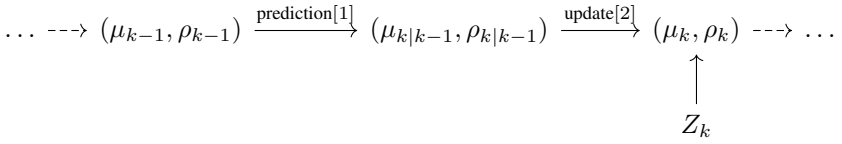


Figure 4: Data flow of the CPHD filter at time k. In addition to the intensity μ_k it propagates the cardinality distribution ρ_k, fully describing the number of targets in the entire state space $\mathcal{X}$.

Again, let us define some useful notation before stating the recursion of the CPHD filter, inspired by those introduced in [25]. Similar to the $Y^\flat$ terms (7) of the SO-PHD filter, define the $Y^\sharp$ terms for the SO-PHD filter as follows:

$$Y_u^\sharp[Z](n) = \sum_{j=0}^{\min(|Z|,n-u)} \frac{n!(|Z|-j)!}{(n-(j+u))!}\rho_{\mathrm{c},k}(|Z|-j|y)\frac{\mu_{\phi,k}(\mathcal{X}|y)^{n-(j+u)}}{\mu_{k|k-1}(\mathcal{X}|y)^n}e_j^\sharp(Z), \tag{19}$$

where $e_j^\sharp$ are the elementary symmetric functions [25] given by

$$e_j^\sharp(Z) := \sum_{\substack{Z'\subseteq Z \\ |Z'|=j}} \prod_{z\in Z'} \frac{\mu_{z,k}(\mathcal{X}|y)}{\mu_{\mathrm{c},k}(z|y)}. \tag{20}$$

for any $n \in \mathbb{N}$ and for $u = 1, 2$. Similarly to the corrective terms $l_u^\flat$ (10), (11) for the SO-PHD, we can defined the corrective terms for the CPHD filter as follows

$$l_1^\sharp(\phi|y) = \frac{\langle Y_1^\sharp[Z_k], \rho_{k|k-1}(\cdot|y)\rangle}{\langle Y_0^\sharp[Z_k], \rho_{k|k-1}(\cdot|y)\rangle}, \tag{21}$$

$$l_1^\sharp(z|y) = \frac{\langle Y_1^\sharp[Z_k \setminus \{z\}], \rho_{k|k-1}(\cdot|y)\rangle}{\langle Y_0^\sharp[Z_k], \rho_{k|k-1}(\cdot|y)\rangle}, \tag{22}$$

where $\langle f, g\rangle$ denotes the inner product $\sum_{n\geq 0} f(n)g(n)$.

The prediction step of the CPHD filter is then given by

$$\mu_{k|k-1}(x|y) = \mu_{\mathrm{b},k}(x) + \mu_{\mathrm{s},k|k-1}(x|y), \tag{23}$$

$$\rho_{k|k-1}(n|y) = \sum_{j=0}^{n} \rho_{\mathrm{b},k}(n-j)S[\mu_{k-1}(\cdot|y), \rho_{k-1}(\cdot|y)](j), \quad n \in \mathbb{N}, \tag{24}$$

where

$$S[\mu, \rho](j) = \sum_{l=j}^{\infty} \binom{l}{j} \frac{\langle p_{\mathrm{s},k}, \mu\rangle^j \langle (1-p_{\mathrm{s},k}), \mu\rangle^{l-j}}{\langle 1, \mu\rangle^l}\rho(l). \tag{25}$$

Finally, the update step of the CPHD filter is given by

$$\mu_k(x|y) = \mu_{\phi,k}(x|y)l_1^\sharp(\phi|y) + \sum_{z\in Z_k} \frac{\mu_{z,k}(x|y)}{s_{\mathrm{c},k}(z|y)}l_1^\sharp(z|y), \tag{26}$$

$$\rho_k(n|y) = \frac{Y_0^\sharp[Z_k](n)\rho_{k|k-1}(n|y)}{\langle Y_0^\sharp[Z_k], \rho_{k|k-1}(\cdot|y)\rangle}, \quad n \in \mathbb{N}. \tag{27}$$

II.5 Sensor State Estimation

This section focuses on the estimation of the sensor process Ψ. The state of the sensor is denoted by $y \in \mathcal{Y}$, the sensor space $\mathcal{Y}$ describing the relative position of an image frame with respect to the initial frame in the image sequence. Following the Sequential Monte Carlo (SMC) implementation in [23], the information regarding the state of the sensor at time k is described by a probability distribution p_k approximated through a set of weighted particles $\{w_k^i, y_k^i\}_{i=1}^N$, i.e.

$$p_k(\cdot) \simeq \sum_{i=1}^{N} w_k^i \delta_{y_k^i}(\cdot). \tag{28}$$

Note that the number of particles, N, is maintained constant throughout the scenario. Selecting an appropriate number of particles is a key consideration of the implementation of the filter. A large number of sensor particles provides a more robust estimator of the drift, but increases significantly the computational cost of the overall algorithm: recall from the hierarchical structure (see Sec. II.2) that a pair of multi-object filters – either PHD, SO-PHD, or CPHD filters – is maintained for each possible sensor state, i.e., for each particle state y_k^i.

By construction, the initial state y_0 of the sensor is the origin vector in $\mathcal{Y}$: the particle states y_0^i are all initialised as the origin vector and the particle weights are all initialised as the uniform weight N^{-1}.

At each time step, the prediction step of the sensor state follows a Markov transition model $h_{k|k-1}$ denoting the operator's knowledge in the nature of the sensor drift (i.e. jittering of the sensor base, linear drift, etc.). Each particle state is thus resampled according to the transition model, i.e.

$$y_k^i \sim h_{k|k-1}(\cdot|y_{k-1}^i). \tag{29}$$

The update step of the sensor state works as follows. Since the likelihood function $\ell_{z,k}(x|y)$ associating a collected observation z to an object with state x is dependent on the sensor state[2], the estimation of the static object process $\Phi^{(1)}$, is conditioned on the sensor state y. As it may be expected, the population of static objects (e.g. stars) is a remarkably stable feature for the estimation of the sensor drift; thus, the estimation of the static object process Φ is exploited to update the sensor state distribution as follows:

$$w_k^i \propto \mathcal{L}(Z_k|\Phi^{(1)}, y_k^i) w_{k-1}^i, \tag{30}$$

where the *multi-object likelihood* $\mathcal{L}(Z_k|\Phi^{(1)}, y_k^i)$ quantifies the match between the set of collected observations Z_k and the estimation of the static object process Φ conditioned on y_k^i. These multi-object likelihoods can be seen below in Sec.II.5.1-II.5.3

The sensor particles are then re-sampled using an importance re-sampling with progressive correction [26], whenever the effective sample size ratio $\left(\sum_{i=1}^N (w_k^i)^2\right)^{-1}$ drops beneath a threshold τ.

II.5.1 PHD Multi-Object Likelihood

If the static object process is estimated through a PHD filter (see Sec.II.2.1), the multi-object likelihood is given by [27]:

$$\mathcal{L}(Z_k|\Phi^{(1)}, y_k^i) = \frac{\prod_{z\in Z_k}\left[\mu_{\mathrm{c},k}(z|y_k^i) + \int p_{\mathrm{d},k}(x|y_k^i)\ell_k(z|x, y_k^i)\mu_{k|k-1}^{(1)}(x|y_k^i)\mathrm{d}x\right]}{\exp\left[\int \mu_{\mathrm{c},k}(z|y_k^i)\mathrm{d}z + \int p_{\mathrm{d},k}(x|y_k^i)\mu_{k|k-1}^{(1)}(x|y_k^i)\mathrm{d}x\right]}. \tag{31}$$

II.5.2 SO-PHD Multi-Object Likelihood

If the static object process is estimated through a SO-PHD filter (see Sec.II.3), the multi-object likelihood is given by

$$\mathcal{L}(Z_k|\Phi^{(1)}, y_k^i) = \sum_{j=0}^{|Z_k|} \frac{(\alpha_{k|k-1})_j}{(\beta_{k|k-1})^j} \frac{(\alpha_{\mathrm{c},k})_{|Z|-j}}{(\beta_{\mathrm{c},k}+1)^{|Z|-j}} \tilde{F}_{\mathrm{d}}^{-\alpha-j} F_{\mathrm{c}}^{-\alpha_{\mathrm{c},k}-|Z|-j} \sum_{\substack{Z'\subseteq Z_k \\ |Z'|=j}} \prod_{z\in Z'} \mu_{z,k}^{(1)}(\mathcal{X}|y_k^i) \prod_{z'\in Z_k\setminus Z'} \mu_{\mathrm{c},k}(z|y_k^i), \tag{32}$$

[2] the effect of the sensor drift y is reversely applied to the observation z to obtain the "true" observation, see Sec. III for more details

where

$$\tilde{F}_{\mathrm{d}} = 1 - \frac{1}{\beta_{k|k-1}} \int p_{\mathrm{d},k}(x|y_k^i) s_{k|k-1}(x|y_k^i) \mathrm{d}x, \tag{33}$$

$$F_{\mathrm{c}} = 1 + \frac{1}{\beta_{\mathrm{c}}}, \tag{34}$$

where $s_{k|k-1}(x|y_k^i) = \dfrac{\mu_{k|k-1}(x|y_k^i)}{\int \mu_{k|k-1}(\bar{x}|y_k^i)\mathrm{d}\bar{x}}$ is the spatial distribution associated to the intensity $\mu_{k|k-1}$.

II.5.3 CPHD Multi-Object Likelihood

If the static object process is estimated through a CPHD filter (see Sec.II.4), the multi-object likelihood is given by

$$\mathcal{L}(Z_k|\Phi^{(1)}, y_k^i) = \langle \tilde{Y}^{\sharp}[Z_k], \rho_{k|k-1}^{(1)}(\cdot|y_k^i) \rangle, \tag{35}$$

with

$$\tilde{Y}^{\sharp}[Z](n) = \sum_{j=0}^{\min(|Z|,n)} \frac{n!(|Z|-j)!}{(n-j)!} \rho_{\mathrm{c},k}(|Z|-j|y_k^i) \mu_{\phi,k}^{(1)}(\mathcal{X}|y_k^i)^{n-(j+d)} \sum_{\substack{Z' \subseteq Z \\ |Z'|=j}} \prod_{z \in Z'} \mu_{z,k}^{(1)}(\mathcal{X}|y_k^i) \prod_{z' \in Z \setminus Z'} \mu_{\mathrm{c},k}(z|y_k^i), \tag{36}$$

for any $n \in \mathbb{N}$.

III Target and Sensor Modelling

III.1 Target Motion Models

The static object process Φ accounts for near-static objects (stars) in the image. Static objects are described in terms of their x and y positions in a two-dimensional state space $\mathcal{X}^{(1)} \subseteq \mathbb{R}^2$, where specific states at time k are of the form

$$x_k = \begin{bmatrix} \mathrm{x}_k, & \mathrm{y}_k \end{bmatrix}^T, \tag{37}$$

and the Markov transition function for the objects is given by

$$t_k^{(1)}(x_k|x_{k-1}) = \mathcal{N}(x_k; x_{k-1}, Q_k^{(1)}), \tag{38}$$

where $Q_k^{(1)}$ is a covariance matrix of the form

$$Q_k^{(1)} = \sigma_{s,k}^2 \begin{bmatrix} \Delta_k & 0 \\ 0 & \Delta_k \end{bmatrix}, \tag{39}$$

where Δ_k is the time lapse between time steps $k-1$ and k, and $\sigma_{s,k}$ is the standard deviation of the noise on the position vector, set to a small value.

The probability of survival $p_{\mathrm{s},k}$ is assumed to be independent of the target motion model.

III.2 Sensor Motion Models

The motion of the sensor state (29) can be modelled using two motion models, Brownian motion and composite motion, each accounting for a different drift observed in common scenarios. The Brownian motion model, or random walk, describes the random jitter of the image frames, possibly caused by operator error or mount failure. The composite motion model combines a Brownian model with an Nearly Constant Velocity (NCV) model [28], which describes the linear drift of the image frames when the diurnal motion of the Earth has not been accounted for through the use of an equatorial mount or an equivalent solution.

Let us first focus on the Brownian drift model. In that case, the sensor state space $\mathcal{Y} \subseteq \mathbb{R}^2$ describes the x and y displacements with respect to the initial frame, i.e. a sensor state at time k can be written as

$$y_k = \begin{bmatrix} \mathrm{x}_k, & \mathrm{y}_k \end{bmatrix}^T. \tag{40}$$

The motion model is then given as follows:

$$h^{\mathrm{b}}_{k|k-1}(y_k|y_{k-1}) = \mathcal{N}(y_k; y_{k-1}, Q^{\mathrm{b}}_k), \tag{41}$$

where Q^{b}_t is a covariance matrix:

$$Q^{\mathrm{b}}_k = \sigma^2_{b,k} \begin{bmatrix} \Delta_k & 0 \\ 0 & \Delta_k, \end{bmatrix} \tag{42}$$

where $\sigma_{\mathrm{b},k}$ is the standard deviation of the Brownian motion. An example of this Brownian drift over 30 time steps is shown below:

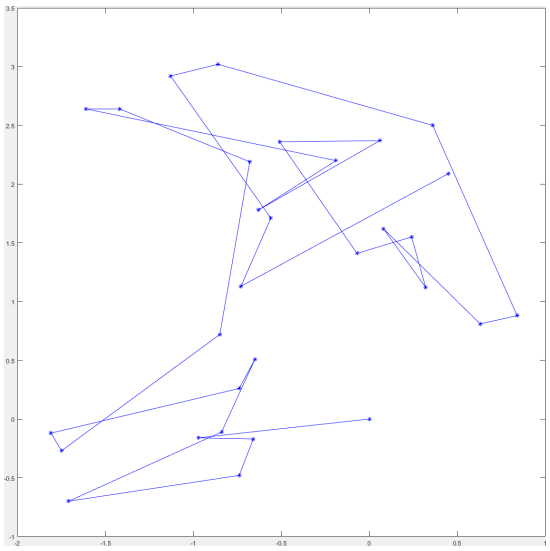


Figure 5: Example of the Brownian drift

Let us now focus on the composite drift model. In that case, the sensor state space $\mathcal{Y} \subseteq \mathbb{R}^4$ describes the displacements in x and y plus the displacement velocities $\dot{\mathrm{x}}$ and $\dot{\mathrm{y}}$ with respect to the initial frame, i.e. a sensor state at time k is written as

$$y_k = \begin{bmatrix} \mathrm{x}_k, & \dot{\mathrm{x}}_k, & \mathrm{y}_k, & \dot{\mathrm{y}}_k \end{bmatrix}^T \tag{43}$$

The motion model is as follows. First, an intermediate state $\hat{y}_k$ is predicted using a NCV model:

$$\hat{y}_k \sim \mathcal{N}(\cdot; A^{\mathrm{c}}_k y_{k-1}, Q^{\mathrm{c}}_k), \tag{44}$$

where A^{c}_k is the transition matrix:

$$A^{\mathrm{c}}_k = \begin{bmatrix} 1 & \Delta_k & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta_k \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{45}$$

and Q^{c}_k is a covariance matrix:

$$Q^{\mathrm{c}}_k = \sigma^2_{\mathrm{c},k} \begin{bmatrix} \frac{\Delta^4_k}{4} & \frac{\Delta^3_k}{2} & 0 & 0 \\ \frac{\Delta^3_k}{2} & \Delta^2_k & 0 & 0 \\ 0 & 0 & \frac{\Delta^4_k}{4} & \frac{\Delta^3_k}{2} \\ 0 & 0 & \frac{\Delta^3_k}{2} & \Delta^2_k \end{bmatrix}, \tag{46}$$

where $\sigma_{\mathrm{c},k}$ is the standard deviation of the NCV velocity. Brownian noise is then added to the intermediate state $\hat{y}_k$ as shown in Eq. (41). An example of this composite drift over 30 time steps is shown below:

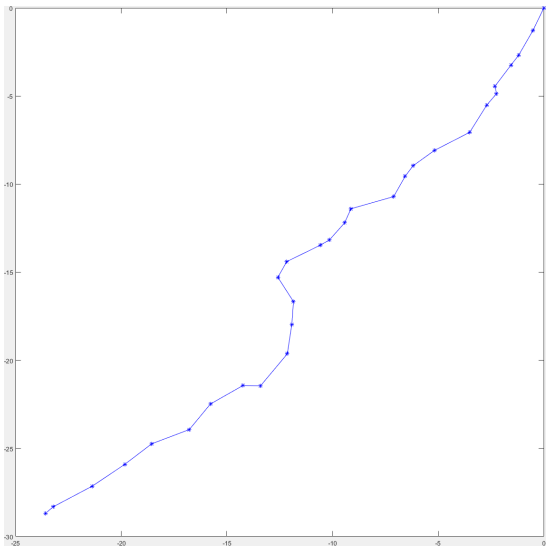


Figure 6: Example of the composite drift

III.3 Sensor Observation Model

The observation space $\mathcal{Z} \subseteq \mathbb{R}^2$ describes the position coordinates of the observed objects in the image frame. The single-object likelihood function $\ell_{z,k}(\cdot|y)$ depends on the sensor state y, as the drift estimated by y must be reversely applied to the collected observation z to recover the "drift-free" observation. This is can be written as follows:

$$\ell_{z,k}(x|y) = \mathcal{N}(z; H_t(x) - H_s(y_k^i), R), \tag{47}$$

where the transformation matrix H_s is

$$H_s^{\mathrm{b}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{48}$$

for the sensor Brownian motion, and is

$$H_s^{\mathrm{c}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{49}$$

for the sensor composite motion. The observation matrix H_t is

$$H_t^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{50}$$

for the static objects. In any case, the noise covariance matrix R is given by

$$R = \begin{bmatrix} \sigma_{\mathrm{x},k}^2 & 0 \\ 0 & \sigma_{\mathrm{y},k}^2 \end{bmatrix}, \tag{51}$$

where the standard deviations $\sigma_{\mathrm{x},k}$, $\sigma_{\dot{\mathrm{x}},k}$ are of the x, y positions.

The probability of detection $p_{\mathrm{d},k}$ and the clutter intensity $\mu_{\mathrm{c},k}$ are assumed to be independent of the sensor state, and the clutter is uniformly distributed across the sensor FoV.

IV Simulations

Three experiments are proposed here that will be carried out to assess several aspects of the single cluster PHD filter using simulated data. The aspects under assessment are:

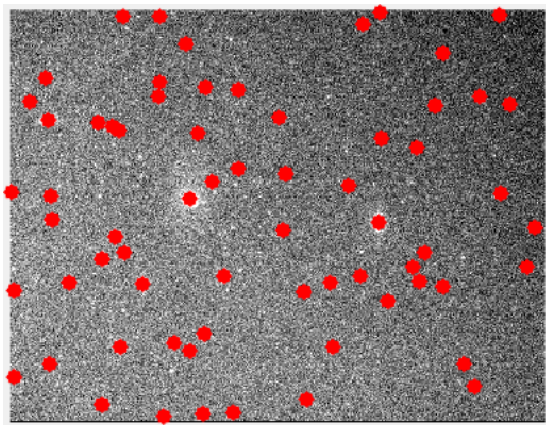
- Sensor Drift Accuracy: How accurately can the filter estimate the sensor drift?
- Execution Time: How long does the filter take to process each time step?

In order to provide a comparison, an implementation of the pattern matching will be used, see below in Sec.IV.1. To measure the sensor drift accuracy, the Root Mean Square Error (RMSE) of both methods' sensor position estimates will be taken and compared. The execution time of the methods for each time step will be also measured. Since the parent process is implemented using a SMC particle filter approach, a preset number of particles shall be used - 50 Monte Carlo (MC) particles. Also the presented results are the averaged results over 20 MC runs. All of the multi-object filters used in the experiments are implemented using a Gaussian Mixture (GM) approach following [29], [20] and [25].

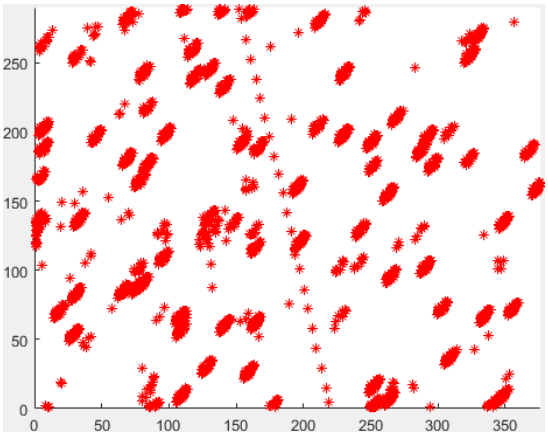
The first experiment (Sec.IV.2) represents a perfect scenario, where no false alarms, missed detections or target death occur. There is a guaranteed sufficient number of static objects (> 10) for the triangle matching method to perform correctly. The outcomes of this experiment are to shows that both methods function correctly and that the single cluster PHD filter has similar accuracy to the triangle matching method. The second experiment (Sec.IV.3) is similar to the first experiment, the difference being that the number of static objects has been reduced to the minimum needed for the triangle matching is function correctly (3). Again this is to demonstrate the robustness of the single cluster PHDmethod, that it can calibrate the sensor off a small number of static objects. The final experiment (Sec.IV.4) is meant to test the single cluster PHD filter to its fullest extent. It consists of a challenging scenario where false alarms, missed detections and target death occur along with a low number of static objects. Each experiment will be performed for the two sensor motion model types.

To generate the data needed to the test the methods the following simulator will be used. It simulates a typical astronomy scenario, see Fig.7 for a real example, but instead of an image it produces point measurements.

(a) Example Image



(b) Example detections shown in red



(c) All detections produced through a dataset

Figure 7: An example of real data

This simulator performs several functions:

- Static Target Generation
- Dynamic Target Generation

- False Alarm and Missed Detection Modelling
- Sensor Drift Modelling

For a given number of dynamic targets M and static targets N, random initial x, y positions within the state space $\mathcal{X}$ are generated. Each target is modeled with a Nearly Constant Heading (NCH) motion model with a random velocity v and inclination θ. These random velocity values are drawn from a Gaussian distribution $\mathcal{N}_v$:

$$\mathcal{N}_v = \mathcal{N}(\cdot; 0, \sigma_v), \tag{52}$$

Where σ_v is the standard deviation of the dynamic targets' velocity. The inclination values are drawn randomly from a range $0 \leq \theta \leq 360$

Each dynamic target is initially inactive and can be activated or "birthed" at any time with probability p_b, this birth can only happen once. The static targets, stars, are present from the beginning of the simulation. At each time step the dynamic targets are moved along according to their respective velocities with added Gaussian noise and detected with probability p_d. False alarms are added to these detections, drawing from a Poisson process with a rate λ_{fa} with random states within the state space. Targets may also be removed from the state space or "die" with probability $1 - p_s$. An example of this simulation can be seen below.

If the sensor drift motion model is set as Brownian motion, it will be modelled at each time step by drawing from a Gaussian distribution:

$$\mathcal{N}(\cdot; 0, \sigma_s), \tag{53}$$

Where σ_s is the standard deviation of the Brownian motion velocity.

Else if the composite motion model is used then a random constant x and y velocity will be generated by drawing from two Gaussian distributions $\mathcal{N}(\cdot; 0, \sigma_x)$ and $\mathcal{N}(\cdot; 0, \sigma_y)$. Where σ_x and σ_y are the standard deviation of the NCV motion velocity in the x, y directions respectively. Random noise will then be added to the drift at each time step to produce the desired motion using the Brownian motion Gaussian above. This drift will be added to every measurement produced to fully replicate the sensor behavior. An example of a simulated dataset can be seen below:

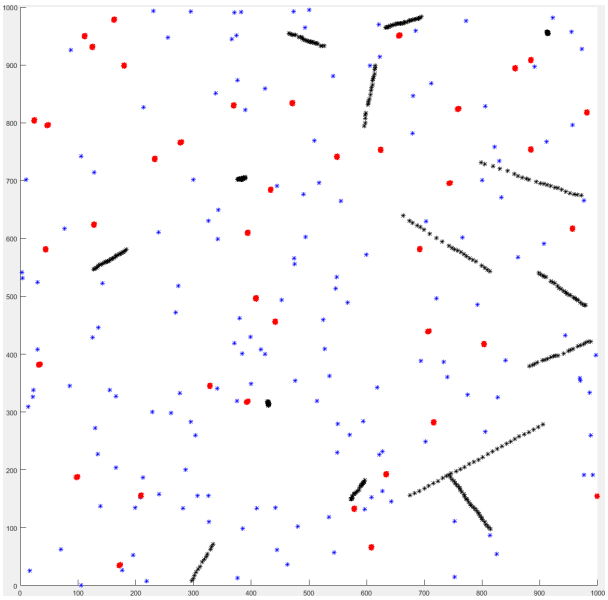


Figure 8: Example of a simulated dataset. Dynamic targets shown in black, static targets shown in red and false alarms shown in blue.

IV.1 Pattern Matching

The initial step in the pattern matching technique is the creation of the lists of triangles. In order to construct the triangles, the distances between the M brightest objects need to be calculated, where M is a user supplied variable. Instead of repeatedly calculating three distances for every new triangle, the squared distances between the objects are calculated once then stored in a table. It has to be noted that the squared distance is used, since finding the square root adds an unnecessary computational complexity.

Once all of the distances are computed the triangle can start to be constructed. The algorithm iterates through the object list and chooses successive trios of objects. The three distances are then sorted in descending order and the ratio between the longest side a and the second longest side b is calculated as in Eq. (54). If value x_t is found to be greater than 0.9, the potential triangle is discarded as triangles with two similar length sides lead to either equilateral triangles or very slim triangles, both of which are not useful for matching. If the triangle passes this test, the other triangle space coordinate y_t is found by the ratio between b and c Eq. (54). These two coordinates are added to an array which also contains the location of the three vertices in the object list. This implementation is almost identical to that in [8] with the addition of the pre-calculated distances and triangle rejection mentioned in [7].

$$x_t = a/b \quad y_t = b/c \tag{54}$$

Once the reference image and another image have had their respective triangle list computed, these lists can be compared. This is done by iterating through the reference list, selecting one triangle at a time. A two dimensional distance comparison is then carried out using this triangle and all triangles in the current image. If any triangles fall within the circle of radius ϵ centered on the reference triangle, they can be considered to be matching the reference triangle. If multiple matching triangles are found, the closest one is used for the voting process.

The voting process is carried out between a pair of matching triangles, where each of the corresponding vertices (objects) are used to increment a value in the corresponding $[i, j]$ element in a matrix, of dimensions $M \times N$, where M and N are the number of objects in the reference and current lists respectively.

Following [8] a differential voting method was implemented as to reduce the number of false matches found in the vote matrix. Since a one to one match can be expected between objects, any object that has a similar number of votes for various objects cannot be trusted. By taking the difference between the original vote matrix $v(m, n)$ and the maximum value of the pairs that share a member:

$$v'(m,n) = v(m,n) - max(\underset{k\neq m}{max}(v(k,n)),\ \underset{l\neq n}{max}(v(m,l))) \tag{55}$$

The coordinate transform for drift estimation can then be derived from list of matched pairs. Several methods exist to calculate this transform, [7] proposes an iterative sigma clipping method or to use a least-squares fitting algorithm [30] to estimate the transform and [8] uses the matrix method for coordinate transform. The technique used in this paper is that of the Maximum Likelihood Estimation SAmple Consensus (MLESAC) algorithm, a robust estimator which can be used in a wide variety of tasks. A detailed explanation of the algorithm can be seen in [31].

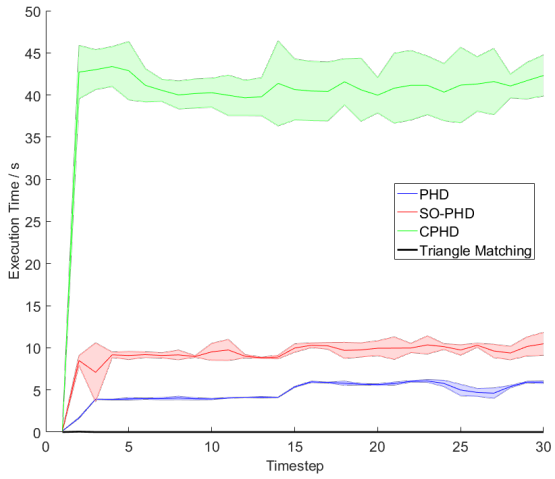
IV.2 Experiment 1

This experiment aims to analyse the performance of the calibration methods in a perfect scenario. The simulation parameters can be seen below:

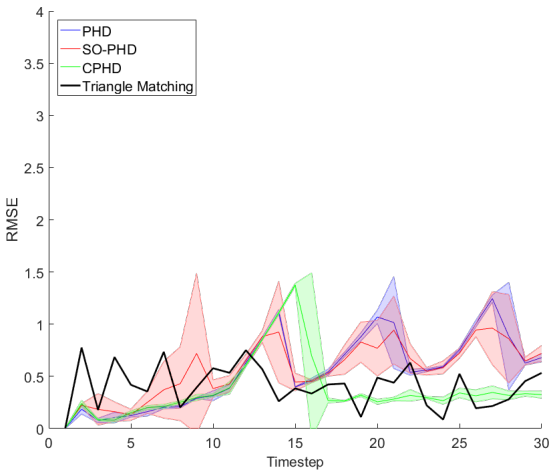
Number of Time Steps	30
Number of Dynamic Targets M	5
Probability of Dynamic Target Birth p_b	0.7
Standard Deviation of Dynamic Target Velocity σ_v	3 pixels/frame
Number of Static Targets N	10
Survival Probability p_s	1
Detection Probability p_d	1
False Alarm Rate λ_{fa}	0 / frame
Standard Deviation of Sensor Brownian Drift σ_s	0.4 pixels/frame
Standard Deviation of Sensor Linear Drift in X σ_x	2 pixels/frame
Standard Deviation of Sensor Linear Drift in Y σ_y	2 pixels/frame
State Space Dimensions ($X \times Y$)	1000 pixels $\times$ 1000 pixels

Table 1: Simulation parameters used in Experiment 1

IV.2.1 Brownian Model



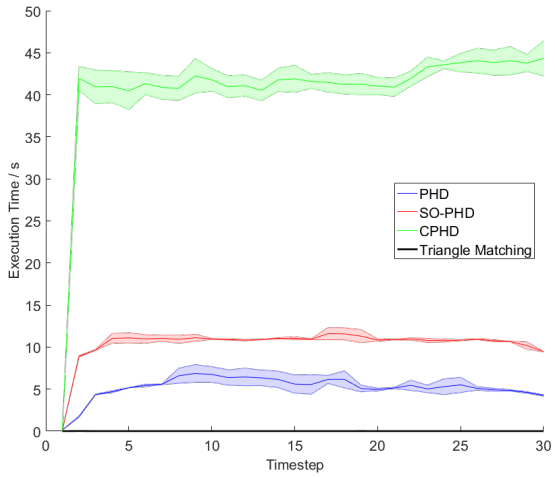
(a) Graph showing the execution times of the methods per time step



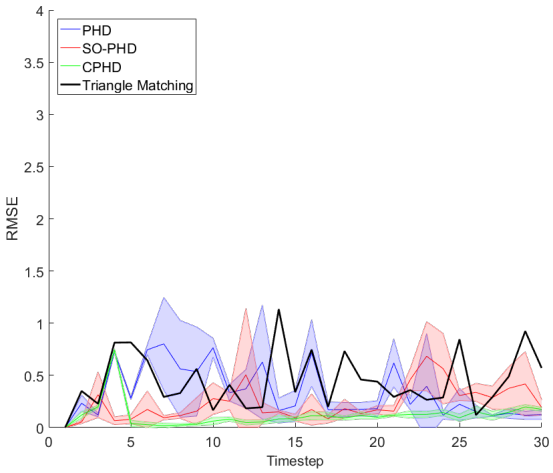
(b) Graph showing the RMSE results of the sensor position estimates

Figure 9: Results for Experiment 1 for the Brownian motion model

IV.2.2 Composite Model



(a) Graph showing the execution times of the methods per time step



(b) Graph showing the RMSE results of the sensor position estimates

Figure 10: Results for Experiment 1 for the composite motion model

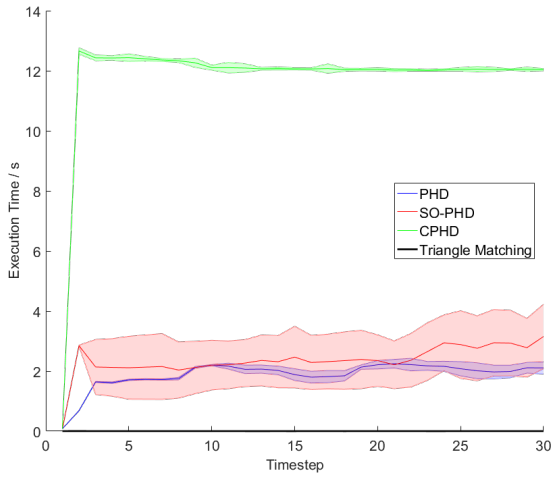
IV.3 Experiment 2

This experiment is similar to the first experiment with the difference being that the number of static objects has been reduced near to the minimum needed for the triangle matching is function correctly (3). The simulation parameters can be seen below:

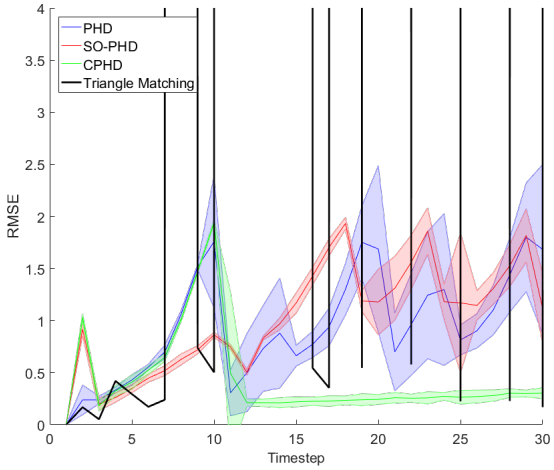
Number of Time Steps	30
Number of Dynamic Targets M	5
Probability of Dynamic Target Birth p_b	0.7
Standard Deviation of Dynamic Target Velocity σ_v	3 pixels/frame
Number of Static Targets N	3
Survival Probability p_s	1
Detection Probability p_d	1
False Alarm Rate λ_{fa}	0 / frame
Standard Deviation of Sensor Brownian Drift σ_s	0.4 pixels/frame
Standard Deviation of Sensor Linear Drift in X σ_x	2 pixels/frame
Standard Deviation of Sensor Linear Drift in Y σ_y	2 pixels/frame
State Space Dimensions ($X \times Y$)	1000 pixels $\times$ 1000 pixels

Table 2: Simulation parameters used in Experiment 2

IV.3.1 Brownian Model



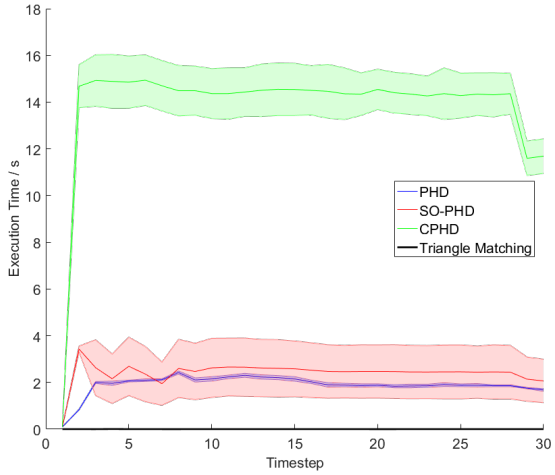
(a) Graph showing the execution times of the methods per time step



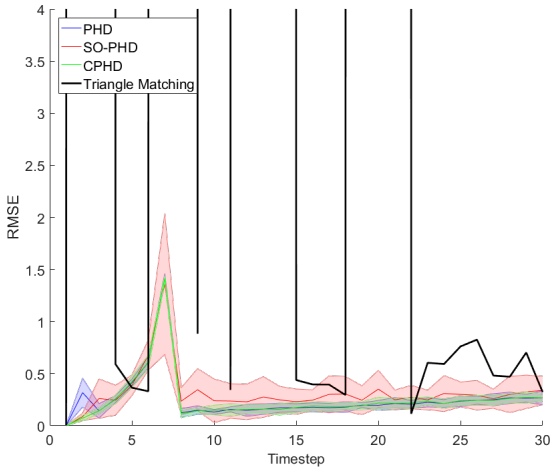
(b) Graph showing the RMSE results of the sensor position estimates

Figure 11: Results for Experiment 2 for the Brownian motion model

IV.3.2 Composite Model



(a) Graph showing the execution times of the methods per time step



(b) Graph showing the RMSE results of the sensor position estimates

Figure 12: Results for Experiment 2 for the composite motion model

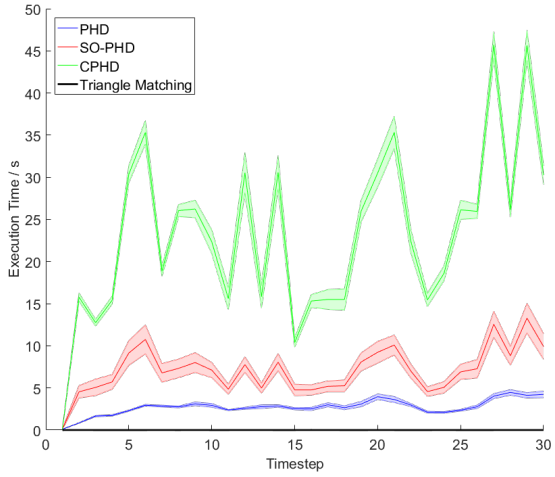
IV.4 Experiment 3

This experiment is meant to test the single cluster PHD filter to its fullest extent. It consists of a challenging scenario where false alarms, missed detections and target death occur along with a low number of static objects. The simulation parameters can be seen below:

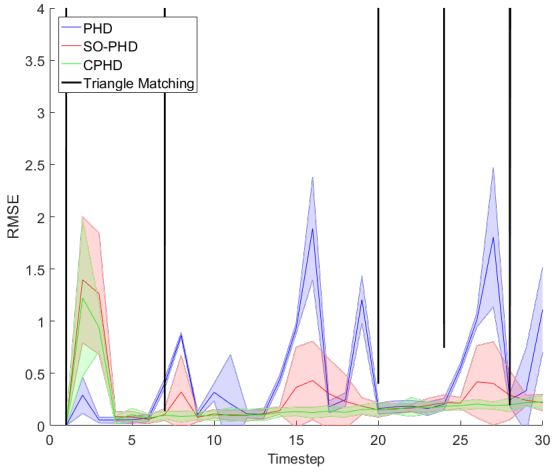
Number of Time Steps	30
Number of Dynamic Targets M	5
Probability of Dynamic Target Birth p_b	0.7
Standard Deviation of Dynamic Target Velocity σ_v	3 pixels/frame
Number of Static Targets N	3
Survival Probability p_s	0.9
Detection Probability p_d	0.8
False Alarm Rate λ_{fa}	5 / frame
Standard Deviation of Sensor Brownian Drift σ_s	0.4 pixels/frame
Standard Deviation of Sensor Linear Drift in X σ_x	2 pixels/frame
Standard Deviation of Sensor Linear Drift in Y σ_y	2 pixels/frame
State Space Dimensions ($X \times Y$)	1000 pixels $\times$ 1000 pixels

Table 3: Simulation parameters used in Experiment 3

IV.4.1 Brownian Model



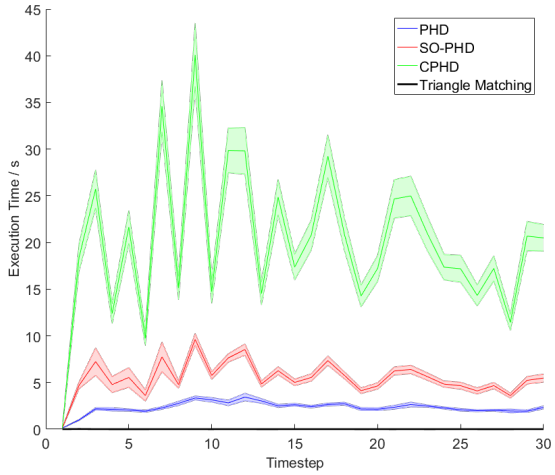
(a) Graph showing the execution times of the methods per time step



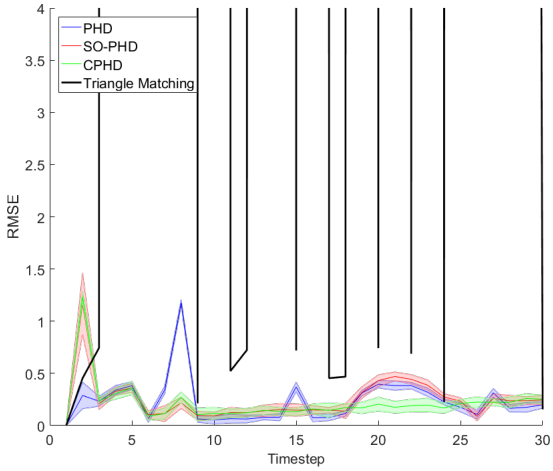
(b) Graph showing the RMSE results of the sensor position estimates

Figure 13: Results for Experiment 3 for the Brownian motion model

IV.4.2 Composite Model



(a) Graph showing the execution times of the methods per time step



(b) Graph showing the RMSE results of the sensor position estimates

Figure 14: Results for Experiment 3 for the composite motion model

IV.5 Analysis

As seen in Experiment 1 (Figs.9 & 10), both the single cluster PHD filter and the triangle matching method show stellar performance for both of the sensor motion models (peak RMSE ≈ 0.45). Experiment 2 (Figs.11 & 12) was designed to display one of the main weaknesses of the triangle matching method, that it needs at least three static objects to properly calibrate an image. The results from Experiment 2 confirm that the triangle matching method starts to fail regularly for both sensor motion models, when there is a low number of static objects to calibrate from, whereas the single cluster PHD filter performs well for both motion models. In Experiment 3 (Figs.13 & 14), where a low number of static objects is combined with a realistic observation scenario, as in Experiment 2, the triangle matching method shows almost complete failure. The single cluster PHD filter on the other hand not only manages to calibrate every image, but manages to maintain a acceptable error on the estimate of the sensor position for both motion models.

The execution time of the two methods, as seen above, is vastly different with the triangle matching method being significantly faster. This is due to the vastly more complex single cluster PHD filter. However note that the tracking of the dynamic and static objects would still need to take place after using the triangle matching method. Out of the single cluster PHD implementations, the PHD is the fastest, followed by the SO-PHD and then the CPHD in all experiments. This is an expected result as they propagate different levels of complexity: The PHD filter propagates the first order moment of the target process, the SO-PHD propagates the the first order moment of the target process and its variance whilst the CPHD propagates the first order moment of the target process and the full cardinality distribution.

V Conclusions

In this paper a refined version of the method first presented in [18] is developed. The preliminary work that was performed in [18] is built upon through the comparison of three different implementations of the single cluster PHD methods [22] and the current state-of-the-art triangle matching method. It was hypothesized that the single cluster PHD filter could provide a similar level of calibration accuracy to the triangle matching method. the presented approach uses a SLAM-based approach that uses a parent-daughter process, where the daughter process is conditioned by the state of the parent process. This parent process consists of a particle filter, which estimates the current sensor position in relation to its initial starting position. The daughter process estimates the time-varying multi-object configuration.

From the results and analysis shown, it can be concluded that single cluster PHD filter offers a almost identical degree of accuracy (sub-pixel) to the triangle matching method when used in an ideal environment. The single cluster PHD filter is also able to cope with more challenging scenarios, Experiments 2 and 3.

The satellites and other dynamic targets in a dataset could be tracked using an additional object process $\Phi_k^{(2)}$. When satellites and debris are observed using telescope, a "streak" is left behind in the image due to the long exposure times needed. The length of this streak can provide an estimate of the target's velocity v and the inclination (or heading) θ of the streak can also be measured.

Figure 15: Example of a satellite streak. Note how the streak length and inclination can be easily measured

This allows the use of the NCH motion model [32]. It provides a better representation of the target dynamics than a regular NCV motion model since the targets move along a fairly fixed line. Dynamic targets are described via their x and y position, the speed v and the inclination θ using a four-dimensional state space $\mathcal{X}^{(2)} \subseteq \mathbb{R}^4$, where specific states at time k are of the form

$$x_k = \begin{bmatrix} \mathrm{x}_k, & \mathrm{y}_k, & v_k, & \theta_k \end{bmatrix}^T \tag{56}$$

The motion model for the dynamic objects is non-linear so an Extended Kalman Filter (EKF) [33] is used to propagate through the time steps. It is given as follows:

$$t_k^{(2)}(x_k|x_{k-1}) = \mathcal{N}(x_k; \hat{x}_k, Q_k^{(2)}), \tag{57}$$

where the intermediate state $\hat{x}_k$ is obtained with

$$\hat{x}_k = x_{k-1} + \Delta_k v_{k-1} \begin{bmatrix} \cos(\theta_{k-1}), & \sin(\theta_{k-1}), & 0, & 0 \end{bmatrix}^T, \tag{58}$$

and where $Q_k^{(2)}$ is a covariance matrix of the form:

$$Q_k^{(2)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{v,k}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\theta,k}^2 \end{bmatrix}, \tag{59}$$

where $\sigma_{v,k}$ and $\sigma_{\theta,k}$ are the standard deviations of the velocity and inclination, respectively.

Another improvement for the multi-object tracking can be seen in [14], where a joint update step for the PHD filter is used to discriminate better between the different target populations. At the time of writing this joint update step has not been implemented for the SO-PHD and CPHD filters. As mentioned before this technique could be adapted to be applied to estimate any unknown sensor parameters with any number of target motion models [22].

Instead of attempting to solve the single-sensor calibration problem in the image plane (x, y coordinates), the method could be extended to the physical spherical plane (right ascension α and declination δ). This would allow a more meaningful physical interpretation of the results to occur and an easier integration with common orbital determination methods such as Gauss' method, double R iteration [34] or Gooding's method [35]. This would also mean that the dynamical motion models would need to be adapted accordingly.

References

[1] U. S. Command, "U.S. Strategic Command," http://www.stratocom.mil, 2017, [Online; accessed May-2017]. [Online]. Available: http://www.stratocom.mil

[2] D. Kessler and G. Gleghorn, "Limiting Future Collision Risk to Spacecraft: An Assessment of NASA's Meteoroid and Orbital Debris Programs," 2011.

[3] A. Rossi, "The Earth Orbiting Space Debris," *Serbian Astronomical Journal*, vol. 170, pp. 1–12, 2005.

[4] J. Cheng, *The Principles of Astronomical Telescope Design*. Springer, 2009.

[5] P. Bely, *The Design and Construction of Large Optical Telescopes*. Springer Science & Business Media, 2003.

[6] E. Groth, "A Pattern-Matching Algorithm for Two-Dimensional Coordinate Lists," *Astronomical Journal*, vol. 91, pp. 1244–1248, May 1986.

[7] F. Valdes, L. Campusano, J. Velasquez, and P. Stetson, "FOCAS Automatic Catalog Matching Algorithms," *Publications of the Astronomical Society of the Pacific*, vol. 107, pp. 1119–1128, Nov. 1995.

[8] M. Marszaek and P. Rokita, "Pattern Matching with Differential Voting and Median Transformation Derivation," 2004, pp. 1002–1007.

[9] W. T. Vestrand, K. N. Borozdin, S. P. Brumby, D. E. Casperson, E. E. Fenimore, M. C. Galassi, K. McGowan, S. J. Perkins, W. C. Priedhorsky, D. Starr, R. White, P. Wozniak, and J. A. Wren, "The RAPTOR Experiment: A System For Monitoring The Optical Sky In Real Time," in *Advanced Global Communications Technologies for Astronomy II*, vol. 4845, Nov. 2002, pp. 126–136.

[10] J. R. Shell, "Optimizing Orbital Debris Monitoring With Optical Telescopes," DTIC Document, Tech. Rep., 2010.

[11] R. G. Abraham and P. G. van Dokkum, "UltraLow Surface Brightness Imaging with the Dragonfly Telephoto Array," *Publications of the Astronomical Society of the Pacific*, vol. 126, p. 55, Jan. 2014.

[12] M. Ackermann, D. Cox, J. McGraw, and P. Zimmer, "Lens and Camera Arrays for Sky Surveys and Space Surveillance," in *Advanced Maui Optical and Space Surveillance Technologies Conference*, 2016.

[13] R. P. S. Mahler, "Multitarget Bayes Filtering via First-Order Multitarget Moments," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 39, no. 4, pp. 1152–1178, 2003.

[14] C. S. Lee, D. E. Clark, and J. Salvi, "SLAM with Dynamic Targets via Single-Cluster PHD Filtering," *IEEE Journal for Selected Topics in Signal Processing (Special Issue on Multi-Target Tracking)*, 2013.

[15] B. Ristic, D. E. Clark, and N. Gordon, "Calibration of Multi-Target Tracking Algorithms Using Non-Cooperative Targets," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 7, no. 3, pp. 390–398, 2013.

[16] J. Houssineau, D. E. Clark, S. Ivekovic, C. S. Lee, and J. Franco, "A Unified Approach for Multi-Object Triangulation, Tracking and Camera Calibration," *Signal Processing, IEEE Transactions on*, vol. 64, no. 11, pp. 2934–2948, Oct. 2016.

[17] I. Schlangen, J. Franco, J. Houssineau, W. T. E. Pitkeathly, D. E. Clark, I. Smal, and C. Rickman, "Marker-Less Stage Drift Correction in Super-Resolution Microscopy Using the Single-Cluster PHD Filter," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 10, no. 1, pp. 193–202, Feb. 2016.

[18] O. Hagen, J. Houssineau, I. Schlangen, E. D. Delande, J. Franco, and D. E. Clark, "Joint Estimation of Telescope Drift and Space Object Tracking," in *2016 IEEE Aerospace Conference*, 2016.

[19] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Artech House, 2007.

[20] I. Schlangen, "A Second-Order PHD Filter with Mean and Variance in Target Number," *Accepted to: Transactions on Signal Processing, IEEE Journal of*, 2016, arXiv:1704.02084.

[21] R. P. S. Mahler, "PHD Filters of Higher Order in Target Number," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 43, no. 4, pp. 1523–1543, 2007.

[22] I. Schlangen, D. E. Clark, and E. D. Delande, "Single-cluster PHD filter Methods For Joint Multi-Object Filtering and Parameter Estimation," May 2017, arXiv:1705.05312.

[23] C. S. Lee, D. E. Clark, and J. Salvi, "SLAM with Single Cluster PHD Filters," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 2096–2101.

[24] I. Schlangen, "Multi-object filtering with second-order moment statistics," Ph.D. dissertation, Heriot-Watt University, 2017, Ph.D. thesis.

[25] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Analytic Implementations of the Cardinalized Probability Hypothesis Density Filter," *Signal Processing, IEEE Transactions on*, vol. 55, no. 7, pp. 3553–3567, 2007.

[26] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, ser. Statistics for Engineering and Information Science. Springer, 2001.

[27] A. Swain, "Group and Extended Target Tracking with the Probability Hypothesis Density Filter," Ph.D. dissertation, Heriot-Watt University, 2013.

[28] X. Li Rong and V. P. Jilkov, "A Survey of maneuvering Target Tracking. Part I: Dynamic Models," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.

[29] B.-N. Vo and W.-K. Ma, "The Gaussian Mixture Probability Hypothesis Density Filter," *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4091–4104, 2006.

[30] C. Lawson and R. Hanson, *Solving Least Squares Problems*. Prentice-Hall, 1974.

[31] P. Torr and A. Zisserman, "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry," *Computer Vision and Image Understanding*, vol. 78, pp. 138–156, 2000.

[32] P. A. Kountouriotis and S. Maskell, "Maneuvering Target Tracking Using an Unbiased Nearly Constant Heading Model," in *2012 15th International Conference on Information Fusion*, July 2012, pp. 2249–2255.

[33] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Prentice-Hall, 1979.

[34] P. R. Escobal, "Methods of orbit determination," *New York: Wiley, 1965*, 1965.

[35] R. Gooding, "A new procedure for the solution of the classical problem of minimal orbit determination from three lines of sight," *Celestial Mechanics and Dynamical Astronomy*, vol. 66, no. 4, pp. 387–423, 1996.